LVM2 • Lector eMMC USB 3.0 • UPS ODROID-HC2 • Proyecto Yocto

ODROID Magazine

Año Cinco
Num. #53
May 2018

Android AUTO

NUEVOS HORIZONTES PARA TI Y TU EXPERIENCIA CON EL MUNDO DE LOS COCHES

SCRIPTS PARA BACKUP:
MANTEN A SALVO TODOS TUS DATOS PARA TU TRANQUILIDAD. HOY Y MAÑANA

ANDROID OREO:
LOGRA QUE TU ODROID-XU4 EJECUTE LA ULTIMA VERSION DE ANDROID A DIA DE HOY

Scripts de Backup: Manten a Salvo tus Datos para tu Tranquilidad

May 1, 2018

¡Para evitar futuros problemas, siempre es bueno hacer una copia de seguridad!

Android Auto: Lleva tu ODROID de Viaje

May 1, 2018

Android Auto es una aplicación de Google que permite que un ODROID-C2/C1+ funcione como un ordenador de a bordo con soporte para navegación, audio y manos libres.

Centro Multimedia Kodi ODROID-C2: Monta tu Propio Sistema de Entretenimiento con una Carcasa LED Personalizada

May 1, 2018

Soy una persona a la que le gusta ver películas, y también me gustar mecaniquear y montar cosas, así que he unido las dos para conseguir de un modo fácil reproducir todo lo que incluye mi colección de películas/música. Tenía que ser fácil de usar, fiable y que también se ▶

NAS Doméstico y Reproductor Multimedia: Montando el Perfecto Sistema de Entretenimiento

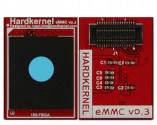
May 1, 2018

Si tienes edad suficiente, puedes recordar e incluso identificarte. Imagínate esto: principios de la década de 2000; DivX y más tarde, su rival XviD en términos de software, y Pentium 4 y Athlons en términos de hardware han hecho finalmente que la compresión de video se convierta en algo; no ▶

Administrador de Volúmenes Lógicos de Linux (LVM2)

May 1, 2018

La gestión de volúmenes Lógicos de Linux (LVM) es un software diseñado para añadir una capa entre los discos reales y la vista que hace el sistema operativo de los mismos para facilitar su administración, reemplazo y ampliación. Suele usarse en los centros de datos que usan hardware de actualización ▶



Lector eMMC USB 3.0

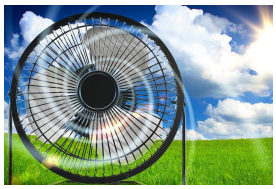
May 1, 2018

La plataforma ODROID de Hardkernel tiene una clara ventaja sobre el resto de ordenadores de placa reducida (SBCs) similares, permite que el módulo eMMC sea retirado y vuelva a grabarse con un adaptador USB externo. Todos los módulos eMMC de Hardkernel se entregan con un adaptador de tarjeta SD que ▶

Económica Solución UPS: Asegúrate de que tu ODROID-HC2 Siempre Esté Funcionado al 100%

May 1, 2018

Muchos sistemas NAS cuentan con una fuente de alimentación ininterrumpida (UPS) para proteger sus valiosos datos de la corrupción accidental provocada por la pérdida del suministro eléctrico principal. Este artículo te ayudará a montar un sistema UPS para el ODROID-HC2 utilizando algunos componentes estándar. Está basado en un mini UPS DC ▶️



Control del ventilador: Adapta el ODROID-XU4 a tu Entorno Perfecto

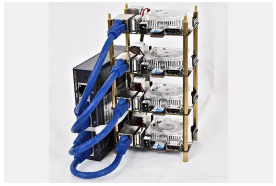
May 1, 2018

El ODROID-XU4 soporta 3 niveles de refrigeración para ajustar el control térmico, y en este artículo aprenderás a adaptarlo a tus necesidades de refrigeración y ruido.

Cliente Minecraft en ODROID

May 1, 2018

¡Ahora se puede jugar al Minecraft en el ODROID! La instalación es bastante sencilla, gracias a las habilidades de empaquetado de Tobias también conocido como @meveric.



Clúster ODROID-XU4

May 1, 2018

En los últimos años, los temas de big data y ciencia de los datos se han convertido en una corriente dominante en innumerables industrias. Las empresas de alta tecnología de Silicon Valley ya no son las únicas proveedoras de temas como Hadoop, regresión logística y aprendizaje de máquinas. Estar familiarizado ▶️

Conceptos Básicos de BASH: Introducción a BASH

May 1, 2018

Esta guía es una introducción amigable para los principiantes del shell BASH

Android Oreo: Hazte con la Última Versión de Android para tu ODROID-XU4

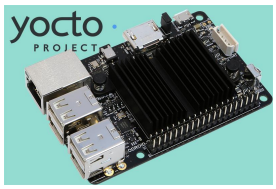
May 1, 2018

El usuario de los foros ODROID voodik ha estado exportando Android 8.1 (basado en LineageOS 15.1) para ODROID-XU4 desde el pasado mes de octubre. Recientemente ha lanzado la primera versión alfa con fines de depuración por parte de la comunidad.

Buscadores, Mineros y 49 – Parte 3: Operación y Mantenimiento de Sistemas de Minería de Cripto-Monedas

May 1, 2018

En los dos últimos artículos sobre buscadores, Mineros y 49, me centré en la minería dual CPU/GPU con sgminer-arm-5.5.6-RC y analicé brevemente la evolución térmica del sistema y la puesta a punto de la GPU. En este tercer artículo, echaremos un vistazo a los problemas operacionales más graves de la ▶️



El Proyecto Yocto: Instalado y Funcionando en el ODROID-C2

May 1, 2018

Este artículo describe los componentes fundamentales y el proceso para compilar una imagen Linux ODROID-C2 personalizada. Estos mismos pasos se pueden usar en otras máquinas ODROID. Yocto es la herramienta estándar de la industria para la creación de sistemas embebidos Linux personalizados y complejos que utilizan las últimas tecnologías de ▶️

Conociendo a un ODROIDian: Matthew Kinderwater (WebClaw)

May 1, 2018

Soy el Director de Servicios TI en una empresa llamada iCube Development que tiene su sede en Calgary, Alberta, Canadá. Entre mis funciones principales se encuentran la recuperación de datos, trabajar en un laboratorio realizando tareas como el reemplazo de cabezales, reparaciones eléctricas y recuperar datos de volúmenes RAID.

Scripts de Backup: Manten a Salvo tus Datos para tu Tranquilidad

May 1, 2018 By Adrian Popa Linux, Tutoriales

Has trabajado muy duro para darle forma a tu sistema y has solucionado todos los errores, pero sabes que las cosas no duran para siempre, y es posible que recibas una actualización con un navegador corrupto, o que quieras experimentar con un nuevo kernel o paquete beta. Para evitar futuros problemas, siempre es bueno hacer una copia de seguridad o backup. Sin embargo, las copias de seguridad suelen ser algo que llevan a confusión en los foros, y muchos usuarios novatos tienen dificultades a la hora de hacerlas. En este artículo, vamos a aprender qué debemos hacer para que nuestro sistema esté siempre seguro.

Discos, particiones y sistemas de archivos

Los usuarios informáticos experimentados no sulene tener problemas a la hora distinguir entre discos, particiones y sistemas de archivos. No obstante, para que todos partamos de un mismo punto vamos a analizarlos. Un disco generalmente es un dispositivo físico que almacena datos en bloques accesibles al azar. En configuraciones más complejas, es posible combinar múltiples discos físicos como matrices RAID utilizando para ello un determinado hardware o software, y que son presentados al sistema operativo como discos virtuales. Las particiones son secciones dentro de los discos que generalmente tienen un sistema de archivos. Los sistemas de archivos gestionan cómo se almacenan los archivos y los datos para poder localizarlos más tarde. Para realizar una copia de seguridad de tu sistema ODROID, deberá conservar la información el contenido de las particiones.

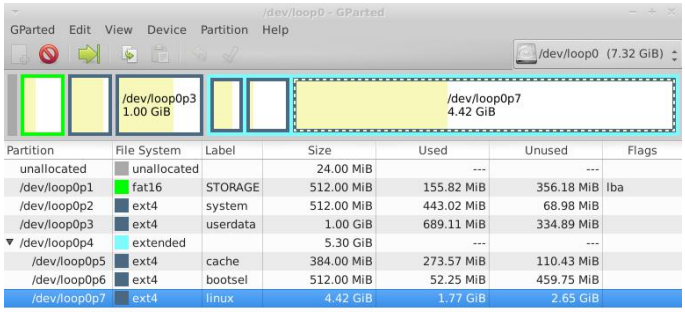


Figura 1 – Distribución de particiones de una triple imagen de arranque de ODROID-C1

Todos los discos empiezan con un bloque de datos de 512 bytes que generalmente contiene el gestor de arranque (para sistemas x86, 446 bytes) y el Registro Maestro de Inicio (MBR) 64B, el cual se explica en http://bit.ly/2bMCTUh. El MBR es una tabla con el offset inicial, la longitud y los tipos de particiones de tus 4 particiones primarias. Estas son las particiones asignadas del 1 al 4 en el kernel Linux (por ejemplo, sda1, sda2, sda3, sda4 para un disco llamado sda). El MBR es una estructura de datos antigua, introducida en 1983, de modo que tiene algunas limitaciones. La necesidad de utilizar discos cada vez más grandes (> 2 TB) dio lugar a la introducción de la tabla de particiones GUID (GPT) que reemplaza al MBR en los nuevos sistemas, tienes más información en http://bit.ly/2bvb4oL. Los ODROID pueden usar tanto MBR como GPT, pero el sistema de arranque está diseñado como un volumen MBR debido a su tamaño relativamente pequeño y simplicidad.

Aunque, tal y como se puede apreciar en la Figura 1, un disco puede tener más de 4 particiones. Esto se consigue recurriendo a un truco: una partición primaria se marca como "extendida" y puede contener tantas particiones lógicas como se quiera. Linux las representa con números desde el 5 en adelante (es decir, sda5, sda6, etc.). La información de las particiones lógicas se almacena en estructuras similares al MBR llamadas Registro de Arranque Extendido (EBR) tal y como se explica en http://bit.ly/2bw47Re, que se muestra como una lista vinculada tal y como se puede apreciar en la Figura 2, y que precede a la partición real del disco.

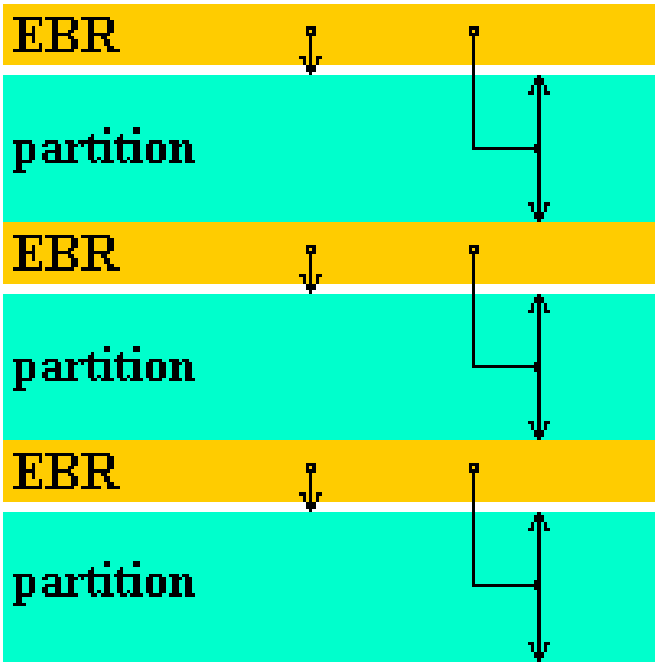


Figura 2 – Posición del EBR en el disco

Las particiones que generalmente verás en los ODROIDs son FAT16/FAT32 (se ven como VFAT bajo el comando mount) y Ext2/3/4. Existen otros tipos de particiones compatibles con Linux, como NTFS, XFS y ZFS, pero por lo general no son esenciales para el proceso de arranque, de modo que las dejaremos a un lado. Existen herramientas para hacer copias de seguridad como BackupPC (http://bit.ly/2bx3J6R) o Clonezilla (http://bit.ly/1Iq2mN7), que admiten más tipos de particiones o que hacen copias de seguridad a nivel de archivos. Estas mismas herramientas deberías utilizarlas para hacer copias de seguridad de tus datos personales, como archivos, imágenes o música. Tampoco está de más antes de empezar a hacer una backup llevar a cabo una "limpieza a fondo" para eliminar las cosas que ya no necesites, como son archivos temporales o descargas, con el objetivo de reducir el tiempo que se necesita para hacer la copia de seguridad y el tamaño del archivo de backup. Por ejemplo, puedes eliminar la memoria caché de los paquetes apt descargados con el siguiente comando:

```
$ sudo apt-get clean
```

Estrategias para hacer Backup

Hay unas cuantas formas de hacer una copia de seguridad de tu tarjeta SD/eMMC. La más simple de implementar es hacer una copia binaria 1:1 de los datos en un archivo de imagen. Para esta tarea,

puede usar una herramienta como dd o Win32DiskImager. Ten en cuenta que todos los comandos que introduzcas a continuación necesitan tener la variable $backupDir reemplazada por la ruta del directorio de la copia de seguridad, que no puede estar en la misma partición de la que intentas hacer una copia de seguridad por razones obvias.

```
$ sudo dd if=/dev/mmcblk0
of=$backupDir/backup.img bs=1M
```

En el comando anterior, "if" representa "el archivo de entrada" y debe apuntar al dispositivo del bloque que representa tu disco, como /dev/mmcblk0, y "of" representa "el archivo de salida" donde se deben escribir los datos. El parámetro "bs" representa "el tamaño del bloque", que viene a representar la cantidad de datos que se leen y escriben al mismo tiempo. Una variante del comando dd que muestra el progreso es el comando "pv" (visor de información):

```
# apt-get install pv
# dd if=/dev/mmcblk0 bs=1M | pv | dd
of=$backupDir/backup.img
```

Restaurar los datos es igual de fácil: simplemente reemplaza los valores de "if" y "of":

```
$ sudo dd if=$backupDir/backup.img
of=/dev/mmcblk0 bs=1M
```

Ten en cuenta que dd hace una copia binaria de tu disco. Esto significa que también copiará el espacio disponible de tu disco. El archivo de salida por defecto será tan grande como el disco, lo cual significa que copiar una tarjeta SD de 64GB en su mayor parte vacía te llevará bastante tiempo y ocupará mucho espacio. La ventaja que tiene es que luego puedes ejecutar herramientas como PhotoRec (http://bit.ly/1jwXElB) sobre el espacio libre y posiblemente, recuperar archivos eliminados, lo cual es muy útil cuando se realizan análisis forenses de datos o se intentar recuperar soportes defectuosos. La desventaja pasa por tener una imagen muy grande que será lenta de copiar. Puedes usar dd junto con gzip para reducir un poco el tamaño de la imagen antes de escribirla, pero no ahorrarás tiempo:

```
# dd if=/dev/mmcblk0 bs=1M | gzip -c >
$backupDir/backup.img.gz
# gunzip -c $backupDir/backup.img.gz | dd
of=/dev/mmcblk0 bs=1M
```

También ten en cuenta que, en teoría, puede hacer una copia de seguridad con dd de un sistema activo copiándolo mientras están montadas las particiones, pero existe el riesgo de que aparezcan inconsistencias si los archivos son modificados durante el proceso de copia de seguridad. Lo mejor es hacer una copia de seguridad de un sistema desconectado extrayendo la tarjeta eMMC/SD, conectar ésta a un sistema diferente y hacer la copia de seguridad sin tener particiones montadas. Existe una desventaja añadida cuando se hacen copias de soportes con tamaños ligeramente diferentes. Como no todas las tarjetas de 16 GB tienen exactamente el mismo tamaño, puedes acabar con una partición truncada en destino

La utilidad "dd" tiene la ventaja de que es fácil de usar, pero para lograr una cierta velocidad en el proceso de backup/restauración y minimizar el espacio de copia de seguridad necesario, debes descomponer la operación de copia de seguridad en varios pasos evitando así hacer una backup del espacio libre. Para ello, necesitarás hacer una copia de seguridad de MBR + EBR, del gestor de arranque y de las particiones individuales.

Aun así, puedes hacer un poco chapucero y usar dd si usas gparted para reducir tu última/mayor partición a solo el tamaño utilizado, dd hasta ese tamaño, luego redimensionar la partición al tamaño original tras restaurarla, pero esto implica un poco de trabajo manual.

Copia de seguridad y restauración del MBR

El MBR y EBR son estructuras de datos pequeñas y se pueden respaldar fácilmente con dd. Pero dado que la posición de EBR en el disco puede variar, debes valerte de una herramienta de partición para extraer y restaurar los datos del MBR/EBR. Dicha herramienta es sfdisk:

```
$ sudo apt-get install sfdisk
$ sudo sfdisk -d /dev/mmcblk0 >
$backupDir/partition_table.txt
```

Para restaurarlos más tarde, necesitarás poner el archivo guardado a disposición de sfdisk del siguiente modo:

```
$ sudo sfdisk /dev/mmcblk0 <
$backupDir/partition_table.txt
```

Ten en cuenta que sobrescribir el MBR en un disco con particiones existentes equivale a eliminar las particiones, puesto que el sistema operativo ya no podrá encontrar los offsets de las anteriores particiones, así que recurre a la restauración con extremo cuidado. Esta copia de seguridad se puede realizar en un sistema activo sin riesgos ya que las tablas de partición generalmente no se modifican durante el tiempo de ejecución.

Copia de Seguridad y rastauración del Gestor de Arranque

Los ODROID utilizan U-Boot como gestor de arranque, tal y como se detalla en el número de noviembre de 2015 de ODROID Magazine (**http://bit.ly/2bA3P9g**). U-Boot almacena su código y los datos en el espacio no asignado después del MBR y al comienzo de la primera partición. También hay un poco de código de arranque en los primeros 446 bytes del primer sector, antes de la tabla de particiones. Puesto que el tamaño y la estructura del U-Boot pueden variar entre los diferentes modelos ODROID, lo más seguro es realizar una copia de seguridad binaria de este espacio no asignado con dd. En primer lugar, debes averiguar el sector de inicio de la primera partición con sfdisk:

```
$ sudo sfdisk -l /dev/mmcblk0
```

```
adrianp@frost:~/temp/odroid/c1$ sudo sfdisk -l /dev/loop0
Disk /dev/loop0: 7.3 GiB, 7864320000 bytes, 15360000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x550ede21

Device       Boot   Start      End  Sectors  Size Id Type
/dev/loop0p1        49152  1097727  1048576  512M  c W95 FAT32 (LBA)
/dev/loop0p2      1097728  2146303  1048576  512M 83 Linux
/dev/loop0p3      2146304  4243455  2097152    1G 83 Linux
/dev/loop0p4      4243456 15359999 11116544  5.3G  5 Extended
/dev/loop0p5      4245504  5031935   786432  384M 83 Linux
/dev/loop0p6      5033984  6082559  1048576  512M 83 Linux
/dev/loop0p7      6084608 15359999  9275392  4.4G 83 Linux
```

Figura 3 – Identificando el sector de inicio de la primera partición con sfdisk y tamaño del sector

Tal y como se indica en la Figura 3, la primera partición (loop0p1) empieza en el offset 49152, de modo que necesitaremos copiarlo todo hasta el sector 49151, incluido éste. El parámetro bs (tamaño de bloque) debe coincidir con el que reporta sfdisk en la línea “Units”: line:

```
$ sudo dd if=/dev/mmcblk0
of=$backupDir/bootloader.bin bs=512
count=49151
```

Ten en cuenta que el comando dd también copiará sobre el MBR, que es el sector 0). Para restaurar el gestor de arranque omitiendo la tabla de particiones, puedes usar el siguiente comando:

```
$ sudo dd if=$backupDir/bootloader.bin
of=/dev/mmcblk0 bs=512 skip=1 seek=1
```

También debes restaurar el código de arranque desde el primer sector:

```
$ sudo dd if=$backupDir/bootloader.bin
of=/dev/mmcblk0 bs=446 count=1
```

Para restaurar la tabla de particiones igualmente, no añadas los parámetros skip y seek. Esto también se puede hacer sobre un sistema activo ya que los datos son mayoritariamente de solo lectura.

Copia de seguridad y restauración de particiones FAT

Por defecto, las imágenes de Hardkernel vienen con una partición FAT16/32 montada en /media/boot que contiene los archivos del kernel, initrd, árbol del dispositivo y boot.ini. Todos estos son cruciales para el arranque del sistema. Los sistemas Android muestran esta partición como almacenamiento “sdcard”.

Existen varias herramientas para Linux que permiten respaldar particiones FAT. Yo solía usar partimage, pero no era posible verificar la suma de comprobación de las particiones en el C2, así que cambié a partclone. Partclone puede hacer una copia de seguridad en bloque de las particiones FAT conservando los datos en los mismos offsets y, además, permite omitir el espacio vacío.

```
$ sudo apt-get install partclone
$ sudo partclone.vfat -c -s /dev/mmcblk0p1 -O
$backupDir/partition_1.img
```

El “-c” especifica “clonar”, “-s” es la partición de origen, que es la primera partición en nuestro caso, y “-O” es el archivo de salida, que se sobrescribirá si existe. Ten en cuenta que partclone no puede funcionar con sistemas de archivos montados y finalizará con un error. Para realizar una copia de seguridad desde un ODROID en ejecución, deberás desmontar /media/boot, realizar la copia de seguridad y volver a montarlo.

Para restaurar una partición FAT, puedes ejecutar el siguiente comando:

```
$ sudo partclone.restore -s
$backupDir/partition_1.img -o /dev/mmcblk0p1
```

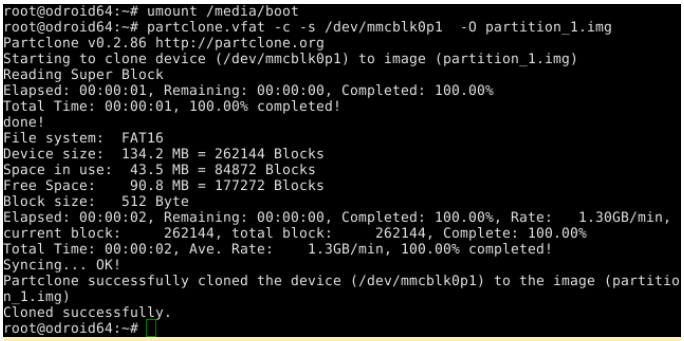


Figura 4 – Copia de seguridad Partclone desmontando previamente /media/boot

Desafortunadamente, PartClone no te permitirá restaurar una partición a una partición de destino más pequeña o más grande, de modo que cualquier ajuste de tamaño que quieras realizar tendrás que llevarlo a cabo tras la restauración. En realidad, puede restaurar a una partición más grande, pero necesitarás aumentarla manualmente para usar el espacio adicional.

Eopia de seguridad y restauración de particiones Ext2/3/4

Para hacer una copia de seguridad y restaurar los sistemas de archivos Ext2/3/4, necesitaremos usar una herramienta diferente llamada FSArchiver. A diferencia de PartClone, FSArchiver crea una copia de seguridad a nivel de archivo y reconstruye el sistema de archivos al restaurar. Desafortunadamente, debido a ciertas particularidades de los sistemas FAT donde los archivos de arranque de Windows deben estar en offsets específicos, el autor de fsarchiver no admite copias de seguridad de los sistemas de archivos FAT, de modo que estamos obligados a utilizar dos herramientas. Pero con la ayuda de paquetes externos, fsarchiver puede soportar otros sistemas de archivos, como XFS, ReiserFS, JFS, BTRFS y NTFS. Por lo general, realiza copias de seguridad de sistemas de archivos no montados, aunque también se puede utilizar en sistemas de archivos activos con el indicador “-A”, que no seimpre funciona. FSArchiver tiene la ventaja de que puede restaurar un sistema de archivos en una partición de destino más grande o más pequeña al mismo tiempo que conserva los UUID. Para hacer una copia de seguridad de la segunda partición, puede ejecutar los siguientes comandos:

```
$ sudo apt-get install fsarchiver
$ sudo fsarchiver -o -v -A -j 4 savefs
$backupDir/partition_2.fsa /dev/mmcblk0p2
```

El delimitador “-o” significa sobrescribir el archivo de destino si existe, “-v” muestra resultados detallados del proceso, “-A” te permite hacer una copia de seguridad de una partición montada y “-j 4” te permite usar 4 núcleos para la compresión.

Para restaurar una copia de seguridad fsa, puedes ejecutar el siguiente comando:

```
$ sudo fsarchiver restfs
$backupDir/partition_2.fsa
id=0,dest=/dev/mmcblk0p2
```

Ten en cuenta que, puesto que FSArchiver admite múltiples particiones dentro de un archivo, necesitas especificar el ID de la partición a restaurar. En nuestro ejemplo, almacenamos solo una partición en el archivo, de modo que siempre especificaremos id=0 cuando restauremos.

SPI Flash

Las placas más modernas, como el Odroid N1, pueden presentar un chip SPI NAND Flash de baja capacidad diseñado para almacenar el gestor de arranque y el kernel, para que pueda arrancar desde la red o desde un disco SATA, sin la necesidad de una tarjeta eMMC o SD. Incluso si el diseño de este chip no se ha decidido por completo cuando escribí este documento, aún podemos hacer una copia de seguridad y restaurarla como un dispositivo de bloque con dd. Puede obtener una lista (y una

descripción) de los dispositivos MTD de tu Odroid ejecutando:

```
$ sudo cat /proc/mtd
 dev: size erasesize name
 mtd0: 01000000 00001000 "spi32766.0"
```

Para hacer una copia de seguridad, simplemente puedes usar:

```
$ sudo dd if=/dev/mtd0
of=$backupDir/flash_mtd0.bin bs=4096
```

Para escribir en un dispositivo flash, para restaurarlo, debes borrar el bloque en el que vas a escribir. Afortunadamente, puesto que vamos a escribir todo el dispositivo, podemos borrarlo totalmente antes de escribir. Para esto necesitamos mtd-utils que proporciona flash_erase:

```
$ sudo apt-get install mtd-utils
$ sudo flash_erase -q /dev/mtd0 0 0
$ sudo dd if=$backupDir/flash_mtd0.bin
of=/dev/mtd0 bs=4096
```

Herramienta ODROID backup

Ahora que sabes cómo hacer las cosas manualmente, tal vez te estés preguntando por qué no son más simples las tareas de copia de seguridad y restauración, simplemente seleccionando y haciendo clic. Estoy de acuerdo en que nadie tiene tiempo para recordar todos los argumentos de línea de comando de varios comandos, así que he montado una GUI rudimentaria que te puede ayudar con el proceso de respaldo y restauración.

La herramienta en cuestión se llama descriptivamente “odroid-backup”. Está escrita en Perl y usa zenity y dialog para montar una GUI rudimentaria, porque soy demasiado viejo para aprender Python. Para instalar la herramienta, puedes descargarla de mi repositorio GitHub:

```
$ sudo wget -O /usr/local/bin/odroid-backup.pl
https://raw.githubusercontent.com/mad-
ady/odroid-backup/master/odroid-backup.pl
$ sudo chmod a+x /usr/local/bin/odroid-
backup.pl
```

El script depende de un puñado de módulos Perl no estándar, así como de algunas utilidades de Linux. Cuando la ejecutes por primera vez mostrará una lista de las dependencias que faltan y las formas de solucionarlo. Para instalar todas las dependencias al mismo tiempo, ejecuta lo siguiente:

```
$ sudo apt-get install libui-dialog-perl
zenity dialog libnumber-bytes-human-perl
libjson-perl sfdisk fsarchiver udev util-linux
coreutils partclone parted mtd-utils
```

El script está diseñado para ejecutarse en sistemas Linux, como un PC al que conectas una tarjeta SD o un módulo eMMC a través de un adaptador USB, o directamente en el ODROID (lo siento por los fanáticos de Windows). Además, el script creará ventanas gráficas si detecta que estás ejecutando una sesión X11, o recurrirá a ncurses (pantalla) si estás conectado a través de ssh o terminal. Puedes forzarlo manualmente con la opción -text.

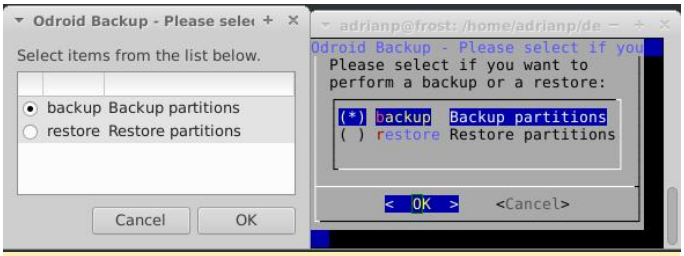


Figura 5 – Zenity vs representación de pantalla

Para realizar una copia de seguridad, inicia la herramienta en un terminal y selecciona “Backup partitions”, luego selecciona OK (1):

```
$ sudo odroid-backup.pl
```

Se te mostrará una lista de unidades desmontables en tu sistema. Puedes iniciar el programa con el delimitador -a para mostrar todas las unidades, que es el caso cuando se ejecuta directamente en el ODROID, puesto que eMMC y SD aparecen como no extraíbles. Selecciona una y haz clic en OK (2). A continuación, se mostrará una lista con todas las particiones en esa unidad. Selecciona las que deseas respaldar (3). Después, deberás seleccionar un directorio donde guardar las copias de seguridad. Lo mejor es tener un directorio vacio (4). Presiona OK, y la copia de seguridad comenzará con una barra de progreso básica que le hará compañía (5). Cuando

finalice la copia de seguridad, aparecerá una ventana con los resultados de la copia de seguridad y los posibles errores (6). Los archivos de copia de seguridad tienen la misma designación utilizada en este artículo. Para hacer una copia de seguridad de un NAND Flash necesitas volver a ejecutar la herramienta y seleccionarla desde los discos disponibles. Puede guardar el archivo resultante en el mismo directorio que las copias de seguridad de la partición.

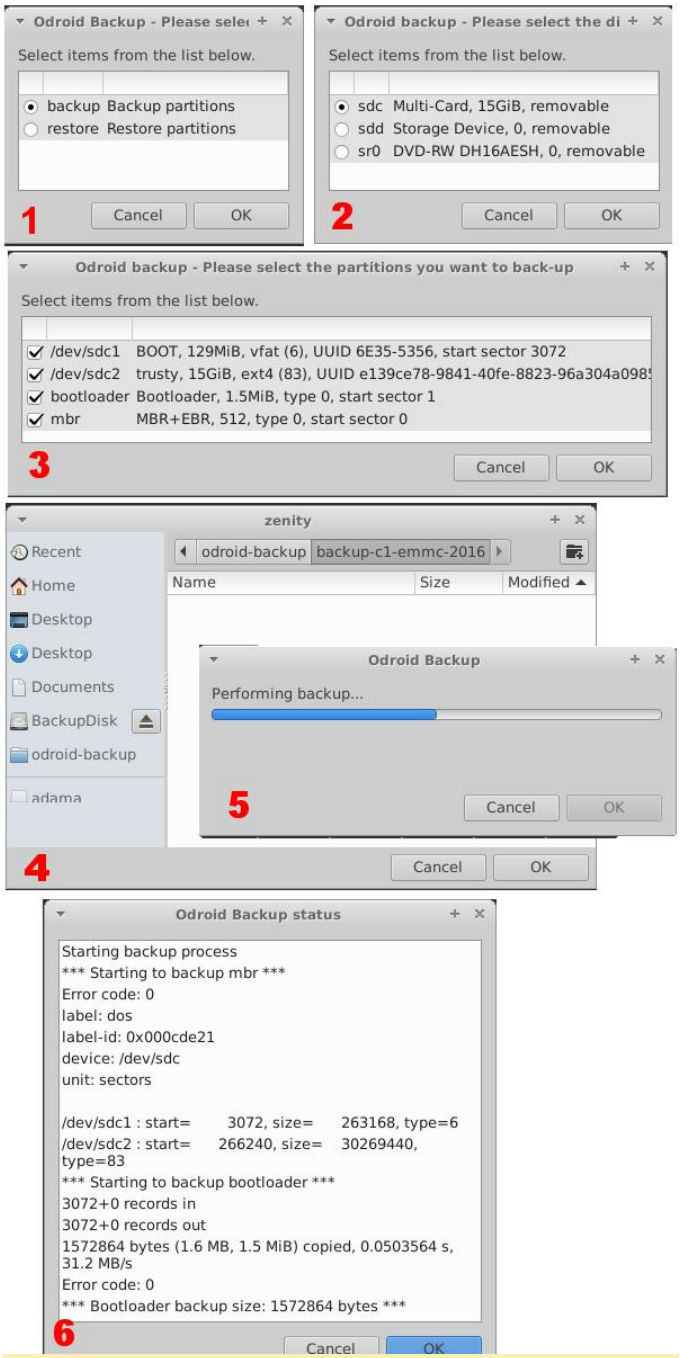


Figura 6 – Pasos de la copia de seguridad

Para realizar una restauración, inicia la herramienta en un terminal, selecciona “Restore partitions”, luego selecciona OK (1):

```
$ sudo odroid-backup.pl
```

Deberás seleccionar el directorio que contiene tus valiosas copias de seguridad y seleccionar OK (2). En la ventana resultante, selecciona qué particiones deseas restaurar desde la copia de seguridad y selecciona OK (3). Ten en cuenta que las particiones se restauran en el mismo orden en el que estaban en el disco original, lo que significa que la partición 1 será la primera partición, y así sucesivamente. En la última ventana, se te preguntará en qué unidad deseas restaurar los datos (4). Disfruta viendo avanzar la barra de progreso (5), y al finalizar aparecerá una ventana de estado con los resultados de la restauración (6). Tambien se guarda un archivo log en /var/log/odroid-backup.log.

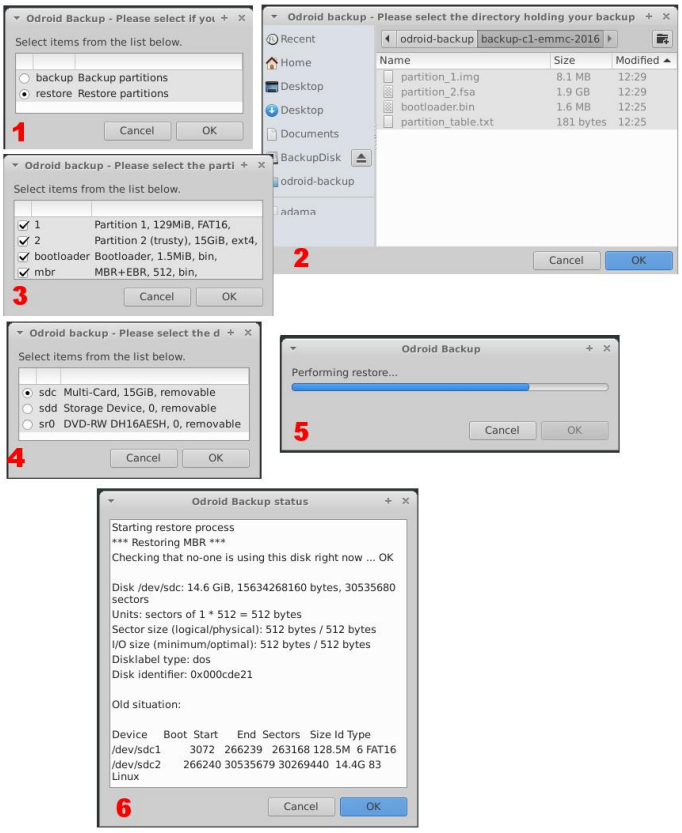


Figura 7 – Pasos de la restauración

Limitaciones conocidas

Si realizas una copia de seguridad de un eMMC para XU3/4, los sectores ocultos (/ dev/mmcblk0boot0, /dev/mmcblk0boot1) no se copian ni se restauran. Estos bloques contienen partes del cargador UBoot. Al restaurar una copia de seguridad en una tarjeta SD o en un nuevo eMMC, la placa puede arrancar con

una versión anterior de UBoot (almacenada antes de la primera partición). Como el resultado que el entorno de UBoot puede estar incompleto (por ejemplo, no hay $ {board_name} establecido), y el arranque puede ser diferente de lo normal (puede que falte la red). Una vez que arranques, se recomienda que reinstales uboot con este comando en la nueva tarjeta:

```
$ sudo apt-get install --reinstall uboot
```

Como cabría de esperar, ningún software está libre de errores, pero con algo de suerte este script de seis pasos tendrá sus aplicaciones. Este script tiene algunas deficiencias, como que las ventanas de zenity no siempre muestran el texto de la instrucción, así que agregué la barra de título. Tampoco hay validación de las copias de seguridad o restauraciones. Deberás revisar el registro para verificar que la operación de copia de seguridad o restauración se haya completado correctamente. Otra limitación es que las particiones FAT deben desmontarse manualmente antes de hacer la copia de seguridad, aunque particiones Ext2/3/4 puede respaldarse estando activadas. Finalmente, la utilidad sfdisk en Ubuntu 14.04 no es compatible con JSON, de modo que no funcionará, aunque puedo agregar soporte si fuera necesario. El programa fue probado respaldando y restaurando varias imágenes oficiales de Hardkernel Linux y Android, así como imágenes de triple arranque, y hasta ahora todo parece funcionar. Las ideas para mejorar y los parches son bienvenidos, como siempre, en el hilo de soporte en http://bit.ly/2bEyFzl.

Android Auto: Lleva tu ODROID de Viaje

May 1, 2018 By Chris Kim Android, ODROID-C1+, ODROID-C2

Android Auto es una aplicación de Google que permite que un ODROID-C2/C1+ funcione como un ordenador de a bordo con soporte para navegación, audio y manos libres. Las instrucciones en video están disponibles en youtube "[ODROID/Android Auto]chip car head unit". La aplicación se ejecuta dentro de Linux utilizando la aplicación OpenAuto.

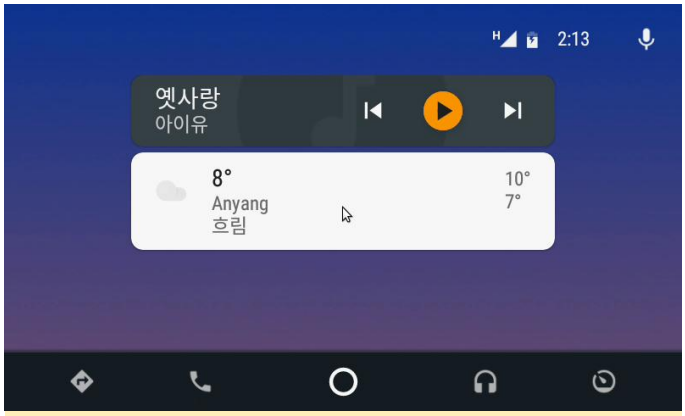


Figura 1

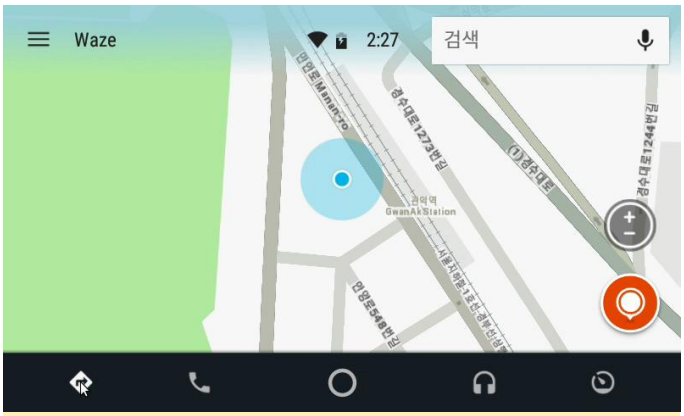


Figura 2

Materiales

- ODROID-C2 / ODROID-C1+
- ODROID-VU7. también puedes usar el ODROID-VU5, aunque puede ser demasiado pequeño para verlo mientras conduces
- Kit I2S 2Watt Stereo Boom Bonnet. Elegimos este por motivos de movilidad. Además, tiene suficiente volumen de sonido, aunque también puede usar otros altavoces.

- ODROID-USB-CAM 720P / Microfono USB. Conectamos un micrófono para el manos libres.
- Botón de encendido. Puede usar tu propio estilo para que coincida con el interior de tu coche
- SmartPower2 15V/4A
- Adaptador de corriente para el mechero
- Cable conector DC de 5,5 mm tipo L
- Cable conector DC de 2,5 mm tipo L

Figura 3 – Preparamos dos botones para usarlos como interruptor de apagado. Para este proyecto, utilizamos el botón más grande

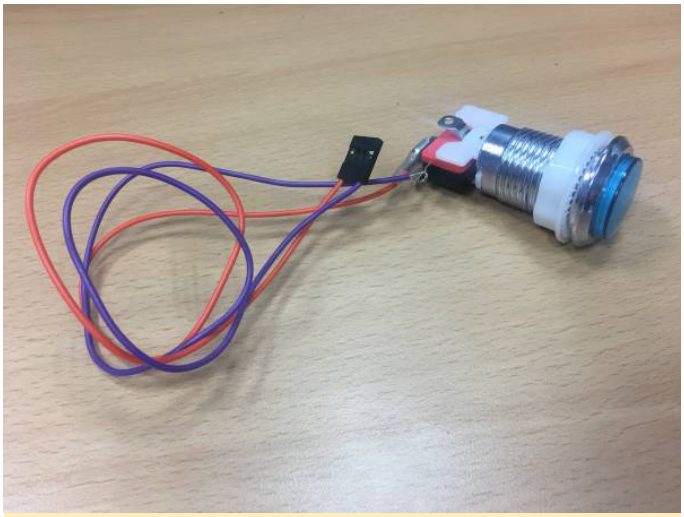


Figura 4 – Preparamos dos botones para usarlos como interruptor de apagado. Para este proyecto, utilizamos el botón más grande

Figura 5 – Usamos un adaptador de corriente para el mechero como fuente de alimentación

Figura 6 – Para conectar el adaptador de alimentación a SmartPower2, cambiamos el conector por un cable tipo L de 5,5 mm

Software

Este proyecto está basado en ubuntu64-16.04.3-mate, Android Auto versión 2.2. Funciona muy bien en ODROID-C2 y ODROID-C1+.

Instalar dependencias

Antes de instalar Audio Auto, debe instalar los paquetes de dependencia

```
$ sudo apt-get update && sudo apt-get upgrade
$ sudo apt-get install -y git-core curl dh-autoreconf libboost-all-dev libusb-1.0.0-dev libssl-dev cmake libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediawidgets5 qtmultimedia5-dev libqt5bluetooth5 libqt5bluetooth5-bin qtconnectivity5-dev pulseaudio gstreamer1.0-plugins-bad gst123 librtaudio-dev
```

La siguiente script se conectará automáticamente en la cuenta de Android después de cada inicio:

```
$ sudo vi /usr/share/lightdm/lightdm.conf.d/60-lightdm-gtk-greeter.conf
 [Seat:*]
 greeter-session=lightdm-gtk-greeter
 autologin-user=odroid
```

Instalación

Para usar Android Auto, necesitarás instalar OpenAuto, que requiere que se instale tanto aasdk como protocol-buffers. Antes de actualizar el compilador, verifica la versión del compilador GCC. La versión de GCC debe ser 6 o superior:

```
$ gcc -v
 Using built-in specs.
 COLLECT_GCC=gcc
 COLLECT_LTO_WRAPPER=/usr/lib/gcc/aarch64-linux-gnu/5/lto-wrapper
 Target: aarch64-linux-gnu
 Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.9' --with-bugurl=file:///usr/share/doc/gcc-5/README.Bugs --enable-languages=c,ada,c++,java,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-5 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-libquadmath --enable-plugin --with-system-zlib --disable-browser-plugin --enable-java-awt=gtk --enable-gtk-cairo --with-java-home=/usr/lib/jvm/java-1.5.0-gcj-5-arm64/jre --enable-java-home --with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-5-arm64 --with-jvm-jar-dir=/usr/lib/jvm-exports/java-1.5.0-gcj-5-arm64 --with-arch-directory=aarch64 --with-ecj-jar=/usr/share/java/eclipse-ecj.jar --enable-multiarch --enable-fix-cortex-a53-843419 --disable-werror --enable-checking=release --build=aarch64-linux-gnu --host=aarch64-linux-gnu --target=aarch64-linux-gnu
 Thread model: posix
 gcc version 5.4.0 20160609 (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.9)
```

Añade el repositorio a través de los comandos add-apt-repository, luego instala gcc-6:

```
$ sudo apt update
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test -y
$ sudo apt update
$ sudo apt install gcc-snapshot -y
$ sudo apt update
$ sudo apt install gcc-6 g++-6 -y
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-6 60 --slave /usr/bin/g++ g++ /usr/bin/g++-6
$ sudo update-alternatives --config gcc
```

Tras la instalación, deberías ver que el GCC actualizado está disponible:

```
$ gcc -v
 Using built-in specs.
 COLLECT_GCC=gcc
 COLLECT_LTO_WRAPPER=/usr/lib/gcc/arm-linux-gnueabihf/6/lto-wrapper
 Target: arm-linux-gnueabihf
 Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 6.3.0-18ub
 untu2~16.04' --with-bugurl=file:///usr/share/doc/gcc-6/README.Bugs --enable-languages=c,ada,c++,java,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-6
 --program-prefix=arm-linux-gnueabihf- --enable-shared --enable-linker-build-id
 --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --
```

```
libdir=/usr/lib --enable-nls --with-sysroot=/
--enable-clocale=gnu --enable-libstdcxx-debug
--enable-libstdcxx-time=yes --with-default-
libstdcxx-abi=new --enable-gnu-unique-object -
-disable-libitm --disable-libquadmath --
enable-plugin --with-system-zlib --disable-
browser-plugin --enable-java-awt=gtk --enable-
gtk-cairo --with-java-home=/usr/lib/jvm/java-
1.5.0-gcj-6-armhf/jre --enable-java-home --
with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-
6-armhf --with-jvm-jar-dir=/usr/lib/jvm-
exports/java-1.5.0-gcj-6-armhf --with-arch-
directory=arm --with-ecj-
jar=/usr/share/java/eclipse-ecj.jar --with-
target-system-zlib --enable-objc-gc=auto --
enable-multiarch --enable-multilib --disable-
sjlj-exceptions --with-arch=armv7-a --with-
fpu=vfpv3-d16 --with-float=hard --with-
mode=thumb --disable-werror --enable-multilib
--enable-checking=release --build=arm-linux-
gnueabihf --host=arm-linux-gnueabihf --
target=arm-linux-gnueabihf
 Thread model: posix
 gcc version 6.3.0 20170519 (Ubuntu/Linaro
6.3.0-18ubuntu2~16.04)
```

A continuación, descarga el código fuente del compilador protobuf:

```
$ wget
https://github.com/google/protobuf/archive/v3.
0.0.zip
$ unzip v3.0.0.zip
$ cd protobuf-3.0.0
```

En el archivo autogen.sh, cambia los paquetes Google Mock por los paquetes Google Test:

```
$ vi autogen.sh
 .
 .
 . (:32)
 if test ! -e gmock; then
 curl $curlopts -L -O
https://github.com/google/googletest/archive/r
elease-1.7.0.zip
 unzip -q release-1.7.0.zip
 rm release-1.7.0.zip
 mkdir -p gmock/gtest
 mv googletest-release-1.7.0 gmock/gtest
 fi
 .
 .
```

Si estás utilizando un ODROID-C2 para tu compilación, puede añadir la opción “-j4” al comando “make”, pero en el ODROID-C1+ debes evitar esta opción por contar con menor memoria del sistema.

```
$ ./autogen.sh
$ ./configure --prefix=/usr/lib/arm-linux-
gnueabihf/
$ make [-j4]
$ sudo make install
$ sudo ldconfig
$ export PATH=/usr/lib/arm-linux-
gnueabihf/bin/:$PATH
$ cd
$ git clone -b master
https://github.com/f1xpl/aasdk.git
$ mkdir aasdk_build
$ cd aasdk_build
$ cmake -DCMAKE_BUILD_TYPE=Release ../aasdk
$ make [-j4]
```

Finalmente, compila e instala Open Auto:

```
$ cd
$ git clone -b master
https://github.com/f1xpl/openauto.git
$ mkdir openauto_build
$ cd openauto_build
$ cmake -DCMAKE_BUILD_TYPE=Release -
DRPI3_BUILD=FALSE -
DAASDK_INCLUDE_DIRS="/home/odroid/aasdk/includ
e" -
DAASDK_LIBRARIES="/home/odroid/aasdk/lib/libaa
sdk.so" -
DAASDK_PROTO_INCLUDE_DIRS="/home/odroid/aasdk_
build" -
DAASDK_PROTO_LIBRARIES="/home/odroid/aasdk/lib
/libaasdk_proto.so" ../openauto
$ make [-j4]
$ echo "./openauto/bin/autoapp &" >> .bashrc
```

Añadir cuentas al grupo

Para solventar el problema del permiso de la cuenta, configura el grupo de usuarios como se muestra a continuación:

```
$ sudo usermod -a -G root odroid
$ sudo usermod -a -G tty odroid
$ sudo usermod -a -G voice odroid
```

```
$ sudo usermod -a -G input odroid
$ sudo usermod -a -G audio odroid
$ sudo usermod -a -G pulse odroid
$ sudo usermod -a -G pulse-access odroid
```

Deberías ver la pantalla de Android auto listo al arrancar.

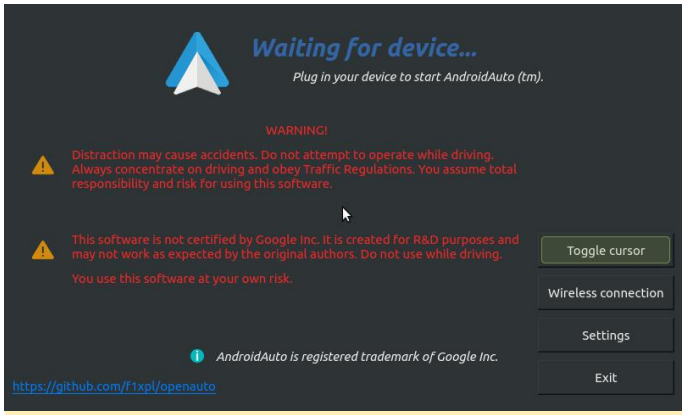


Figura 7

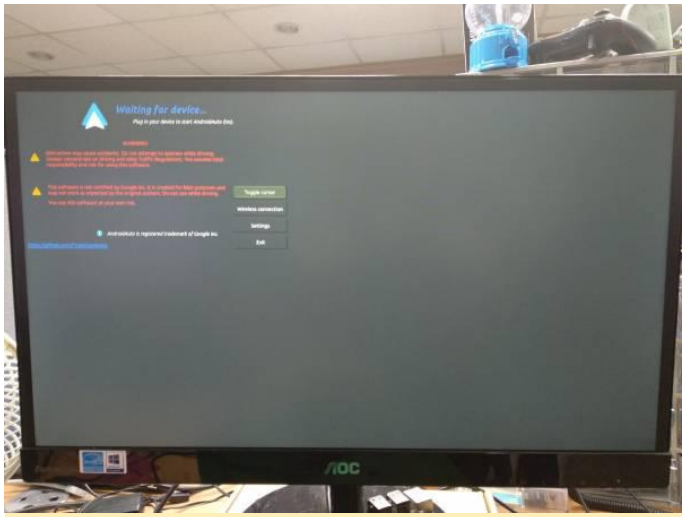


Figura 8

Conectar y configurar los componentes

Para usar la pantalla ODROID-VU7, edita el archivo boot.ini tal y como se muestra a continuación. Deberías editar las opciones de resolución y vout_mode. El ODROID-VU7 tiene una resolución de 800×480 60hz y el modo DVI.

```
$ sudo vi /media/boot/boot.ini

# setenv m "576p" # 720x576
setenv m "800x480p60hz" # 800x480
# setenv m "800x600p60hz" # 800x600
# setenv m "1024x600p60hz" # 1024x600
# setenv m "1024x768p60hz" # 1024x768
# setenv m "1360x768p60hz" # 1360x768
# setenv m "1440x900p60hz" # 1440x900
# setenv m "1600x900p60hz" # 1600x900
# setenv m "1680x1050p60hz" # 1680x1050
# setenv m "720p" # 720p 1280x720
# setenv m "800p" # 1280x800
# setenv m "sxga" # 1280x1024
# setenv m "1080i50hz" # 1080I@50Hz
# setenv m "1080p24hz" # 1080P@24Hz
# setenv m "1080p50hz" # 1080P@50Hz
# setenv m "1080p" # 1080P@60Hz
# setenv m "1920x1200" # 1920x1200

# HDMI DVI Mode Configuration
# setenv vout_mode "hdmi"
setenv vout_mode "dvi"
# setenv vout_mode "vga"
```

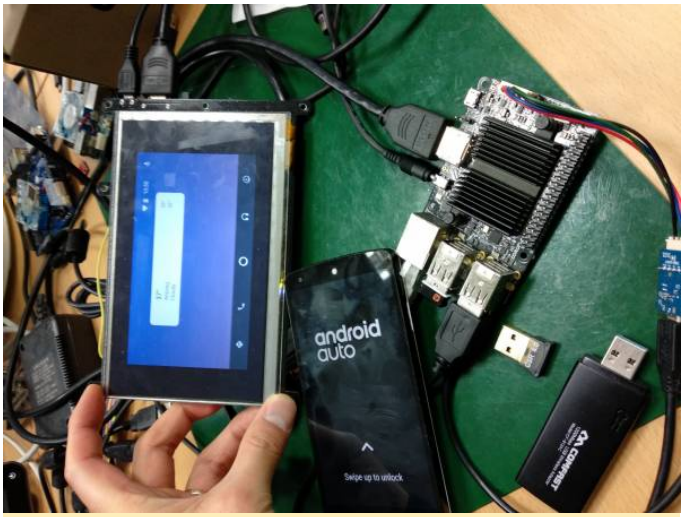


Figura 9

Montar SmartPower2

Este es un dispositivo opcional, y puedes usar cualquier otra fuente de alimentación 5V/3A para esto. SmartPower2 tiene una función de ejecución automática y puedes comunicarte con él a través de WiFi.

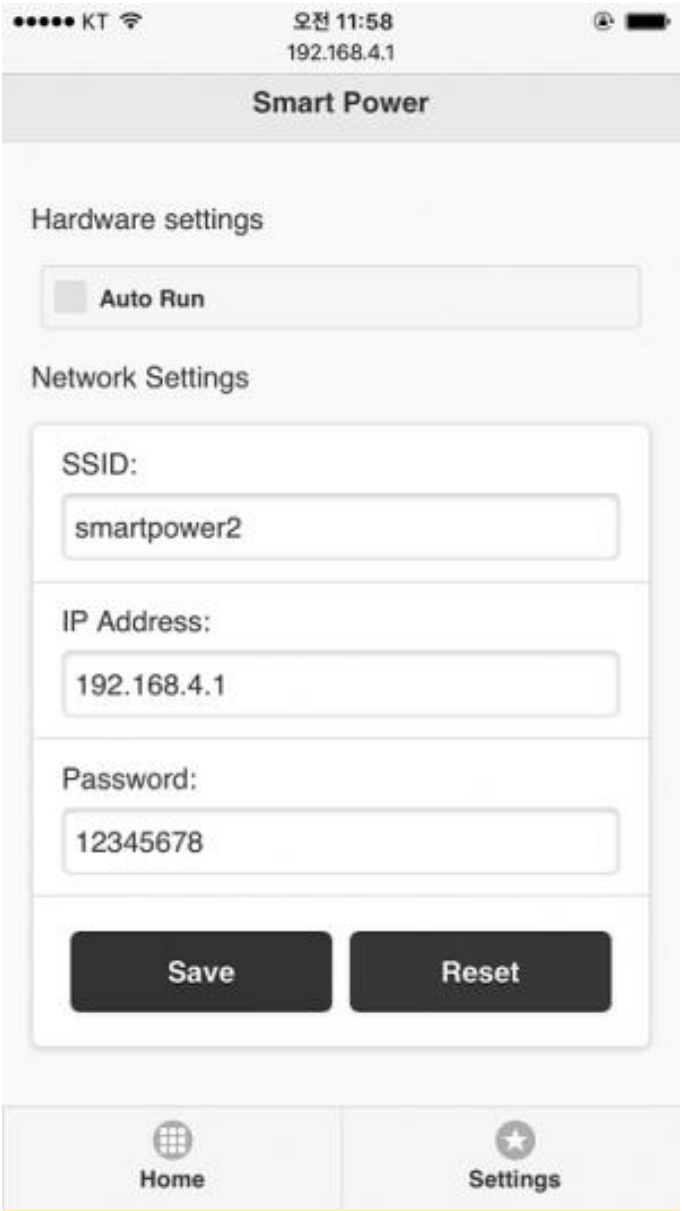


Figura 10 – Encendido/apagado automático de la potencia de salida cuando enciendes el SmartPower2

Comprueba la opción Auto Run y conecte SmartPower2 utilizando el adaptador de alimentación del mechero como entrada y el ODROID-C1+ como salida.

Figura 11

Figura 12

Montar Stereo Boom Bonnet

Si tuvieras que cargar el driver cada vez que arranques ODROID-C1+/C2, simplemente registra el driver en /etc/modules (más detalles):

```
odroid@odroid64:~$ su
 Password: /* root password is "odroid" */
 root@odroid64:/home/odroid# echo "snd-soc-pcm5102" >> /etc/modules
 root@odroid64:/home/odroid# echo "snd-soc-odroid-dac" >> /etc/modules
 root@odroid64:/home/odroid# exit
 exit
 odroid@odroid64:~$
```

Selecciona ” output to ODROID-DAC Analog stereo ” a través de System » Preferences » Hardware » Sound » Output.

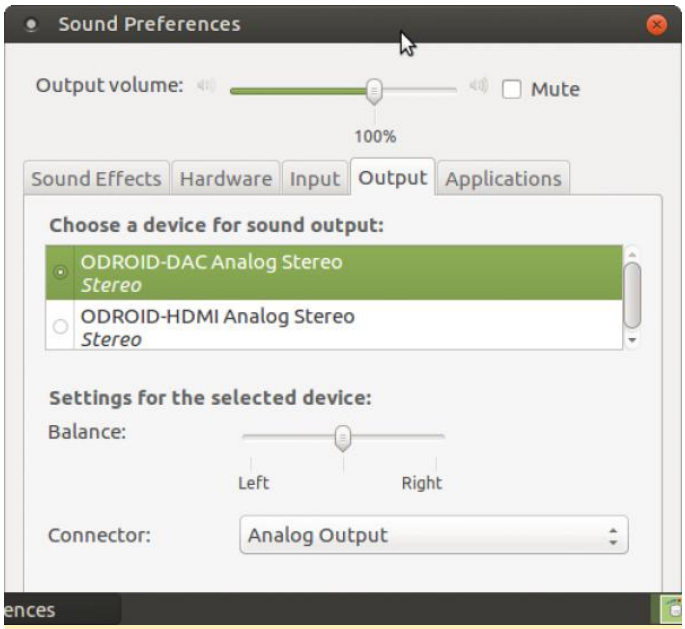


Figura 13 – Seleccionando la salida: ODROID-DAC Analog Stereo

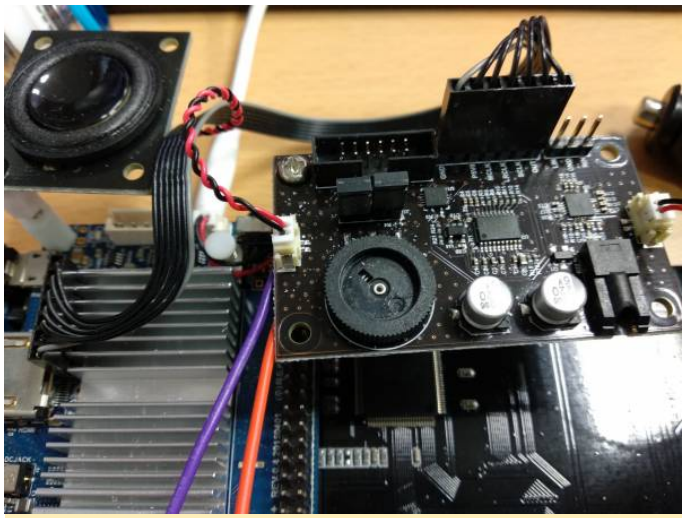


Figura 14 – ¡Asegúrate de verificar el conector!

Configurar el botón de encendido

Con el teclado numérico de la pantalla TFT LCD, se puede apagar el sistema Android Auto cuando paramos el sistema del coche, pero queríamos incluir un botón de encendido independiente para mayor comodidad. Para que esto funcione, hay que cambiar "KEY_UP" por "KEY_POWER" en el código fuente del servicio tftlcd_key:

```
{ PORT_KEY1, HIGH, KEY_UP, KEY_RELEASE },
```

A continuación, actualiza rc.local para cargar automáticamente el servicio tftlcd_key en el arranque:

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

# By default this script does nothing.
 sudo /home/odroid/tftlcd_key &

if [ -f /aafirstboot ]; then /aafirstboot
start ; fi
```

Ahora conecta el botón de encendido a los conectores de expansión GPIO J2. Usamos el Pin 6 y el Pin 12.

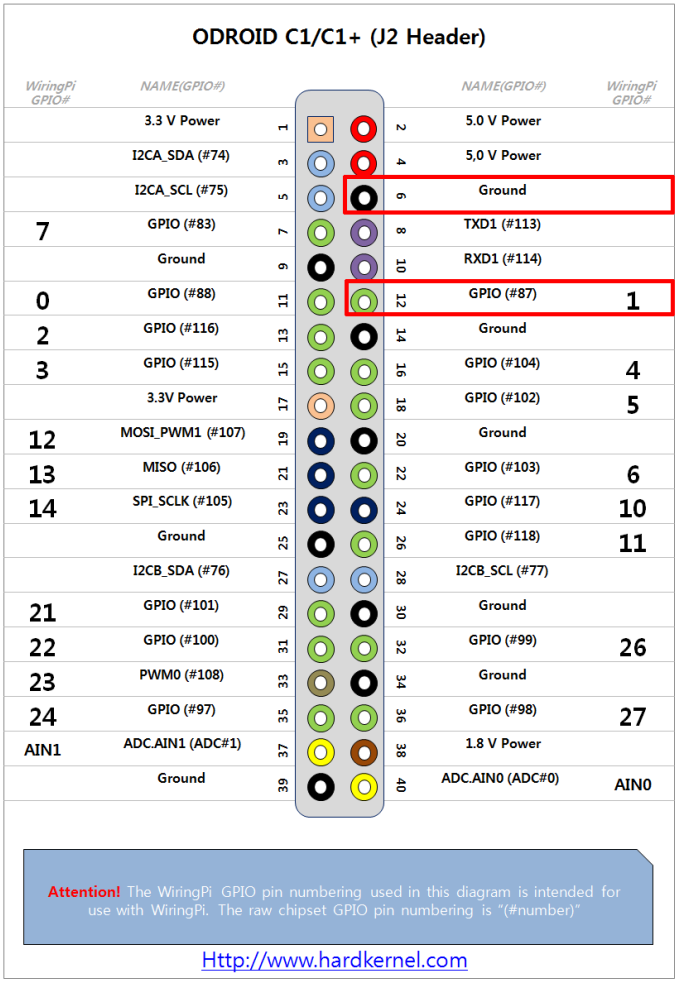


Figura 15

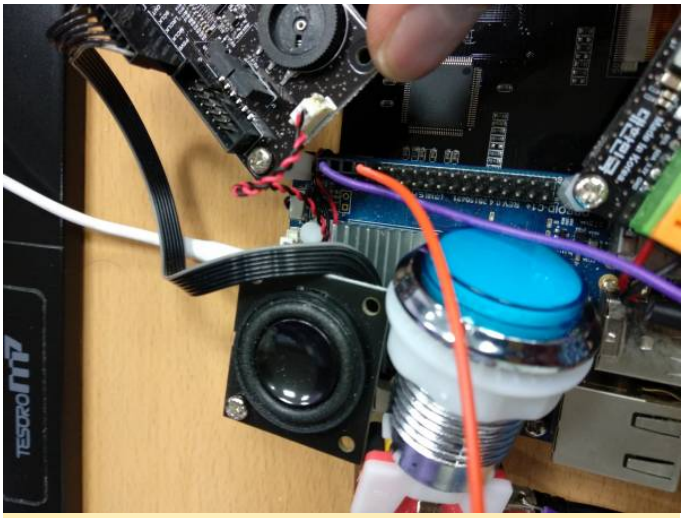


Figura 16

Para asignar la acción de apagado, dirígete a System » Preferences » Hardware » Power Management » General

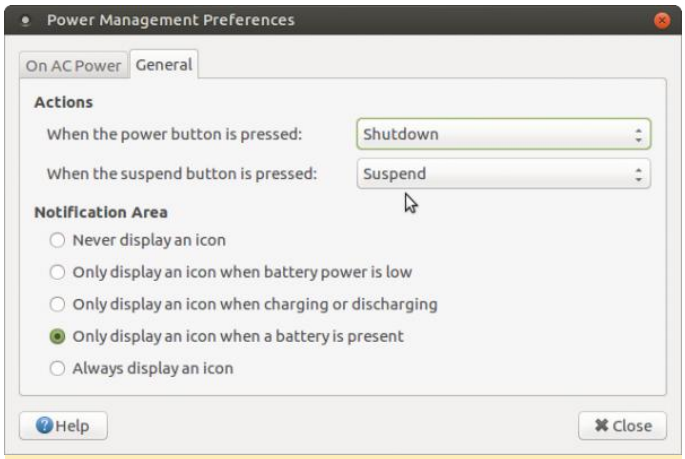


Figura 17 – Cuando se presiona el botón de encendido: Se Apaga

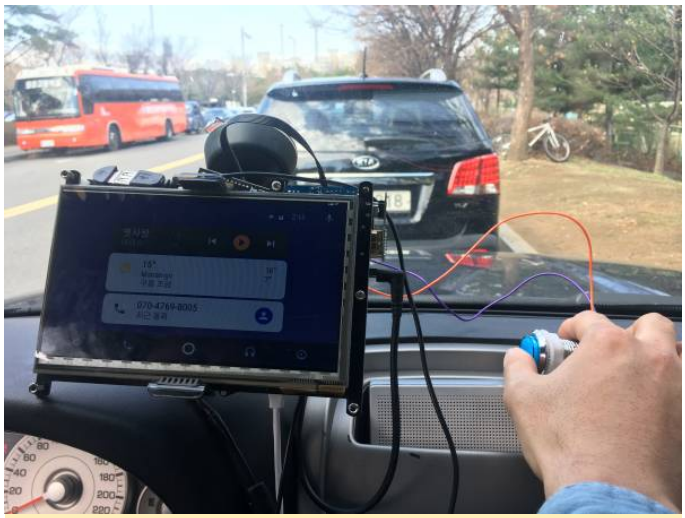


Figura 18

Montar los micrófonos USB

Para usar Google Assistant, también necesitarás un micrófono USB. Configura la entrada del dispositivo USB usando System » Preferences » Hardware » Sound » Input.

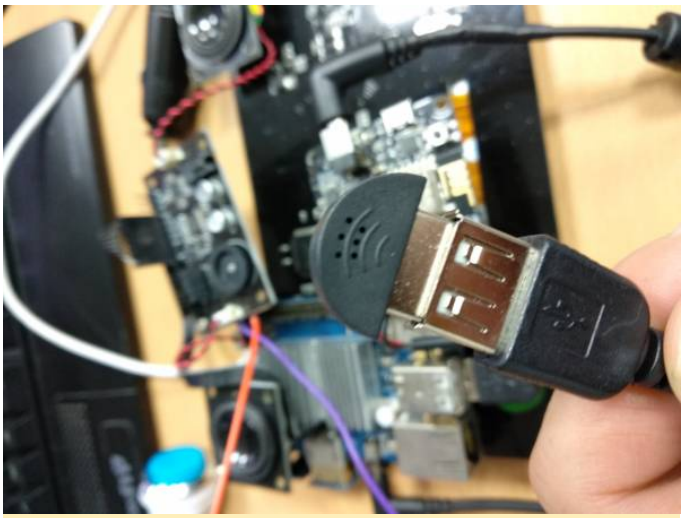


Figura 19

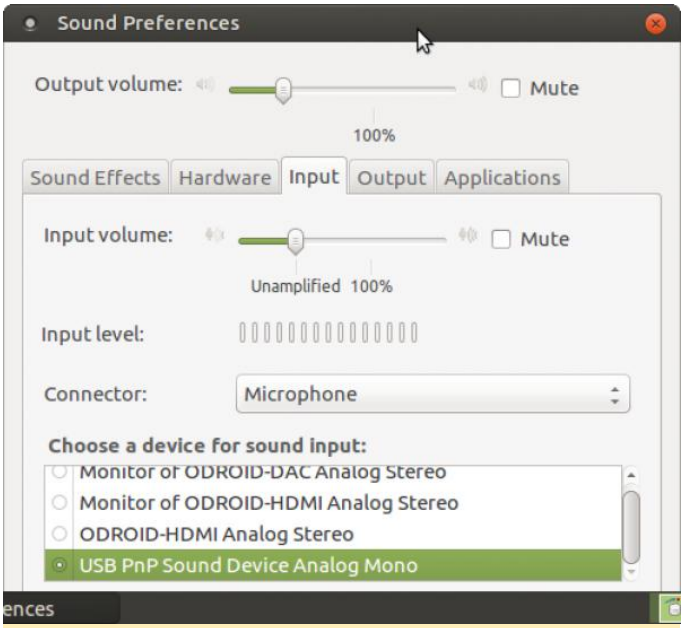


Figura 20 – Seleccionando la entrada: USB PnP Sound Device Analog Mono

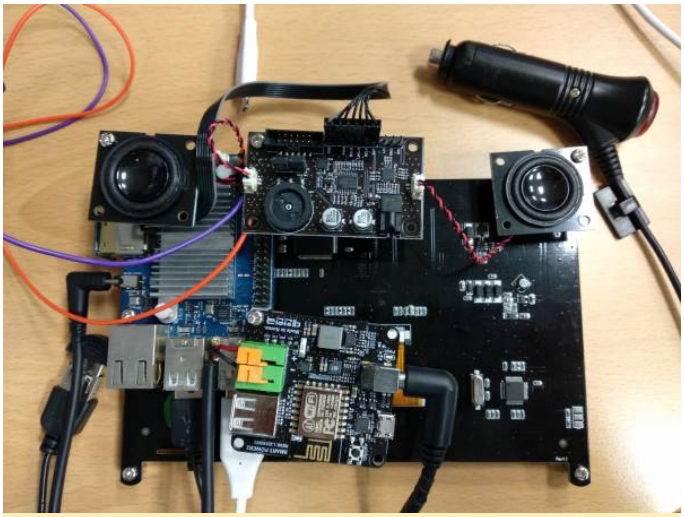


Figura 21

Instalar Android Auto en un coche

Ahora instalamos el dispositivo Android Auto en mi coche personal, el cual funciona muy bien. Después de arrancar, Android Auto está listo automáticamente para conectar tu dispositivo Android. Tras conectar tu dispositivo Android a Android Auto, Android Auto detectará el dispositivo para que puedas usarlo.

Figura 22

Figura 23

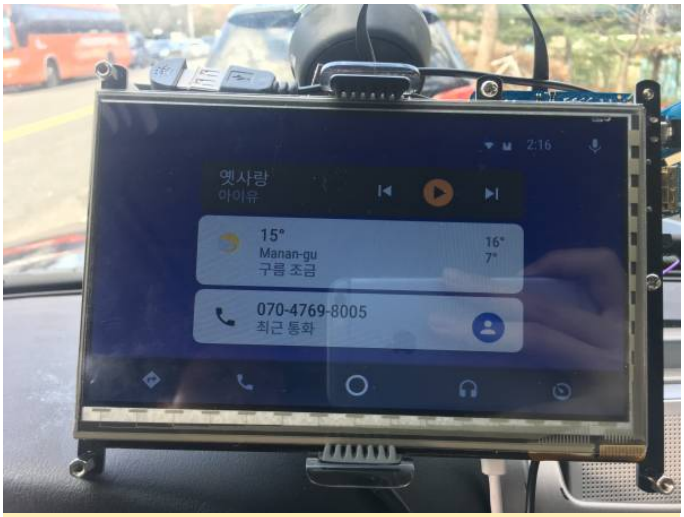


Figura 24

Centro Multimedia Kodi ODROID-C2: Monta tu Propio Sistema de Entretenimiento con una Carcasa LED Personalizada

May 1, 2018 By Gary Morgan ODROID-C2, Mecaniqueo

Soy una persona a la que le gusta ver películas, y también me gustar mecaniquear y montar cosas, así que he unido las dos para conseguir de un modo fácil reproducir todo lo que incluye mi colección de películas/música. Tenía que ser fácil de usar, fiable y que también se viese bien. Mi colección de películas/música está almacenada en un disco NAS, lo cual me permite guardar los originales en un sito aparte. De modo que, pensé que iba a ser algo muy simple puesto que todo lo que tenía que hacer era reproducir Blu-Ray ISO, DVD VideoTS / VOB, CD MP3 y archivos FLAC.

Desde hace años, lo he intentado con muchos clientes multimedia en diversas plataformas, pero a menudo me encontraba con un audio deficiente y una reproducción de video poco fiable. No parecía que estuviese proporcionado con el coste ya que he ido gastado bastante en PCs centro multimedia dedicados a lo largo de los años. Incluso he comprado soluciones ya montadas. Sin embargo, siempre terminaba volviendo a Kodi, anteriormente conocido como XBMC.

Luego vino la Raspberry Pi, repleta de promesas, con precios muy atractivos y con una gran cantidad de programación dedicada de la comunidad Kodi. Por dinero, es increíble lo que se puede hacer. Sin embargo, para un uso cotidiano me resultaba algo lenta, incluso con mis estándares de video tan básicos, en muchos casos se apreciaban saltos de imagen y el sonido se entrecortaba.

Instalación de Kodi

Cualquiera que haya usado Kodi sabrá que existen una gran cantidad de versiones y variantes que soportan una gran variedad de hardware. Después de soportar tantos problemas para mantener activo el sistema con Windows, sabía que tenía que dejarlo a

un lado y buscar una compilación llamada LibreELEC que me proporcionase un sistema operativo Linux básico que dedicará todo el potencial del dispositivo al objetivo principal de ejecutar Kodi. Tras investigar un poco y buscar plataformas compatibles con LibreELEC, me encontré con el **ODROID-C2** de Hardkernel. Similar a la RPi en tamaño y en coste. ¡Con el que tuve una plataforma Kodi lista y funcional en cuestión de minutos! Tanto es así que la placa tal cual se encuentra encima del decodificador de mi salón desde hace mucho tiempo. Prácticamente me he olvidado de ella, simplemente me acomodo y me pongo a ver películas. La instalación es simple:

- descargar el **USB SD creator** desde el sitio LibreELEC,
- descargar la imagen apropiada para tu plataforma, y
- Escribirla, en mi caso, en el **módulo eMMC** (8GB esta bien) usando el **Lector de modulos eMMC USB**

Una nota para el lector: no estoy seguro si esto ha sido solucionado, pero descubrí que la Rev0.2 20130402 del lector resulta algo complicada de utilizar y parece funcionar solo con ciertas ranuras o adaptadores SD. Creo que se debe a las tolerancias mecánicas de la PCB.

Buscando un sitio para el ODROID-C2

Justo cuando me llego mi reluciente televisor OLED 4K, decidí hacer balance de todo mi hardware pensando que ya iba siendo hora de encontrarle al ODROID-C2 una ubicación adecuada. La mayoría de las unidades/carcasas que envuelven la PCB, terminan con muchos cables que salen de la carcasa desde prácticamente todos los ángulos, yo quería que fuera más bien como un decodificador. Llevé a cabo una búsqueda rápida y descubrí un bonito chasis de aluminio de DoukAudio disponible en eBay.

Figura 1 – Chasis DoukAudio

Sin embargo, todavía tenía que lograr ubicar todas las conexiones que necesitaba en la parte posterior del chasis, incluidos la alimentación, LAN, USB y HDMI. Una vez más, una búsqueda en eBay dio como resultado un pequeño adaptador HDMI viable.

Figura 2 – Adaptador HDMI

PSU

Desde mi implicación en la RPi, sabía que una buena PSU también es algo fundamental, así que monté mi propia fuente interna de 5V 2A cableada directamente al J8 de la PCB C2 y seguí el consejo de eliminar el puente J1 deshabilitando la entrada USB OTG.

Por supuesto, puedes comprar un módulo montado listo para utilizar, si no se siente seguro con la red eléctrica o simplemente alimentarlo como normalmente se hace utilizando el puerto micro USB o la toma DC.

Disipador de calor

Teniendo en cuenta que tenía una bonita carcasa de aluminio, me pareció sensato disipar el calor del C2 a través de la carcasa en lugar de hacer agujeros y/o usar aire forzado. Retiré el disipador térmico original y utilicé un bloque de aluminio para reconducir el calor del procesador hacia la carcasa.

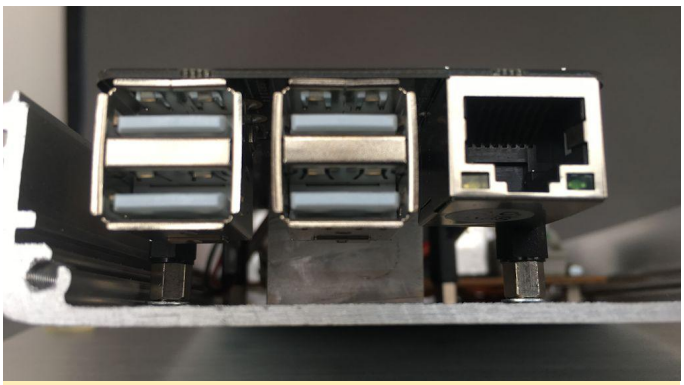


Figura 3 – Disipador de calor

Overclocking

Con un procesador tan genial, tenía algo de espacio para el overclocking. Se hace imprescindible disponer de un buen suministro de energía. Sugiero hacer overclocking paso a paso. En primer lugar, sigue los pasos de overclocking del artículo del foro: (**https://forum.odroid.com/viewtopic.php?t=30078&p=214852**). En mi caso simplemente cargue la tarjeta eMMC en un PC y edité el archivo boot.ini para comprobar qué funcionaba. Una frecuencia de CPU de 1.752GHz me proporciono buenos resultados. Esta fue la línea que añadí al archivo boot.ini:

```
setenv max_freq "1752"
```

Si hay otras líneas que empiecen por setenv max_freq, simplemente coméntelas colocando delante el signo "#", por ejemplo: #setenv max_freq "1536". Vuelve a colocar la tarjeta en el C2, reinicia y prueba. A continuación, haremos overclock del chip gráfico hasta los 792 Mhz (**https://goo.gl/TgGqVx**). Para ello, necesitas conectarte a tu placa mientras se inicia. Primero debe conocer la dirección IP, puedes localizarla en Kodi dentro de Configuración/Información del sistema/Red. A continuación, utilice un cliente Telnet como PuTTy (**https://www.chiark.greenend.org.uk/~sgtatham/putty**), selecciona la casilla SSH e inicia sesión como root utilizando la contraseña libreelec. En la línea de comandos, ejecuta lo siguiente, luego reinicia:

```
$ echo "echo 5 > /sys/class/mpgpu/cur_freq"
>> /storage/.config/autostart.sh
```

El último overclock que puedes aplicar es a la RAM (http://odroid.com/dokuwiki/doku.php?id=en:c2_adjust_ddrclk). Nuevamente, conectarte por SSH a la placa e introduce lo siguiente:

```
$ wget
https://dn.odroid.com/S905/BootLoader/ODROID-
C2/c2_update_ddrclk.sh
$ chmod +x ./c2_update_ddrclk.sh
$ ./c2_update_ddrclk.sh 1104
$ reboot
```

Ten en cuenta que el overclocking es un proceso de prueba y error, no todos los sistemas funcionarán de la misma forma debido a factores como la tolerancia de los componentes, aunque no está de más probarlo. Puedes hacer overclock con la memoria RAM, los gráficos y el procesador y aun así mantener un sistema Kodi muy estable, aunque siempre es mejor que simplemente usar y navegar.

Control remoto

Una desventaja de montar el C2 de esta forma era que el control remoto IR no funcionaría con la tapa puesta y éste era necesario para ENCENDER/APAGAR la placa. Hay algunas opciones para conectar un receptor IR a través de los pines GPIO, pero me resultó más sencillo quitar el receptor IR y extenderlo, de modo que lo ubique en uno de los agujeros LED de 3 mm existentes en la parte frontal del chasis. Agrandé ese agujero hasta los 5 mm. La navegación y control se hacía súper sensible al usar el adaptador Bluetooth USB (https://goo.gl/zYPdkc), que fue ubicado en el panel posterior de la carcasa.

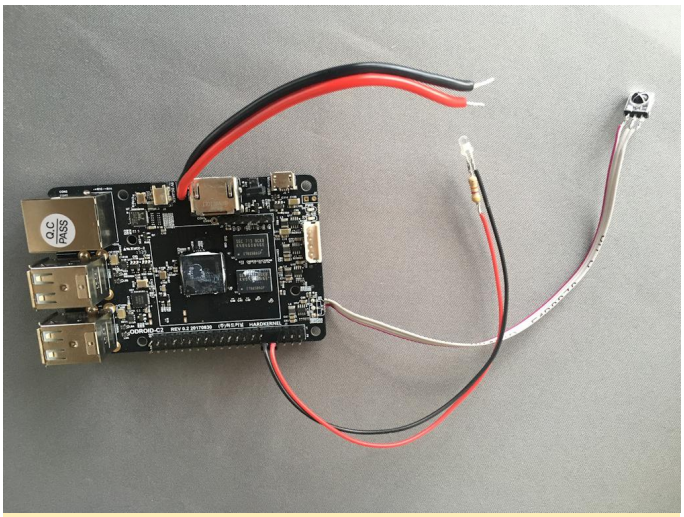


Figura 4 – Cableado del ODROID-C2

LED de encendido del panel frontal

Una cosa que me quedaba por hacer: colocar un LED en el frontal para saber si el sistema estaba o no encendido. Actualmente el C2 ya cuenta con algunos LED (rojo y azul) montados en la PCB. El LED rojo está conectado al riel de 5V y siempre permanece encendido mientras que la placa reciba energía incluso si el sistema operativo se encuentre suspendido. El LED Blue parpadea continuamente en consonancia con la actividad y como puede llegar a distraer bastante, se ha decidido desactivar en las compilaciones más recientes de LibreELEC.

Necesitaba otra forma y lo más obvio era recurrir a los pines GPIO (**https://goo.gl/GzKcVY**) pins. En la actualidad, no soy programador y ya me había hecho la idea de que iba a ser difícil de implementar y tendría que pedir ayuda en los foros (**https://forum.odroid.com/viewtopic.php?f=144&t=30505**). Mientras tanto, investigué un poco y resultó que el acceso a los pines es algo relativamente fácil (https://wiki.odroid.com/odroid-c2/application_note/gpio/enhancement_40pins) gracias a Hardkernel.

El LED es de 3mm, que ajusta muy bien en uno de los orificios pre-taladrados del panel frontal. Al ser un LED blanco de alta eficiencia, no necesita mucha energía para iluminarse. Además, no quería que la luz iluminara toda la habitación. De modo que, elegí una resistencia limitadora de corriente de alto valor de 47k ohm cableada en serie con el LED. Existe un límite de 3.3V @ 3mA en el pin. Podría haber recurrido a la ley de Ohm para calcular el valor, pero simplemente fui haciendo pruebas hasta lograr la intensidad de la luz deseada.

Luego lo conecté entre 0V y el pin GPIO 249 (Pasadores 7 y 9) del cabezal J2. Puedes usar los pines que quieras. Yo utilicé estos porque estaban uno al lado del otro y podía usar un conector pre-cableado que tenía.

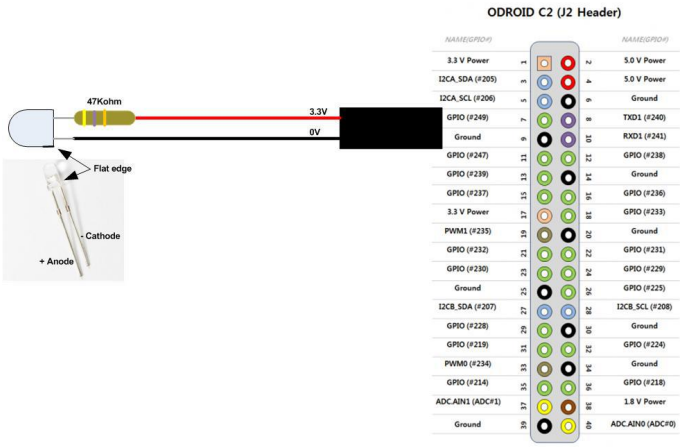


Figure 5 – ODROID-C2 y LED

Con algunas simples inclusiones en los scripts de inicio y cierre, ahora tengo mi LED. Edité los archivos mediante SSH, primero edité el archivo autostart.sh:

```
$ nano /storage/.config/autostart.sh
```

Luego agregué las siguientes líneas:

```
$ echo 249 > /sys/class/gpio/export
$ echo out >
/sys/class/gpio/gpio249/direction
$ echo 1 > /sys/class/gpio/gpio249/value
```

Para el procedimiento de cierre, edité el archivo shutdown.sh:

```
$ nano /storage/.config/shutdown.sh
```

Luego añadí las siguientes líneas:

```
$ echo 0 > /sys/class/gpio/gpio249/value
```

No estoy seguro de si esta es la mejor forma, pero a mí me funciona. Junto con un control remoto Logitech Elite para gestionar mi sistema AV, ahora dispongo de un centro multimedia ideal para la familia.

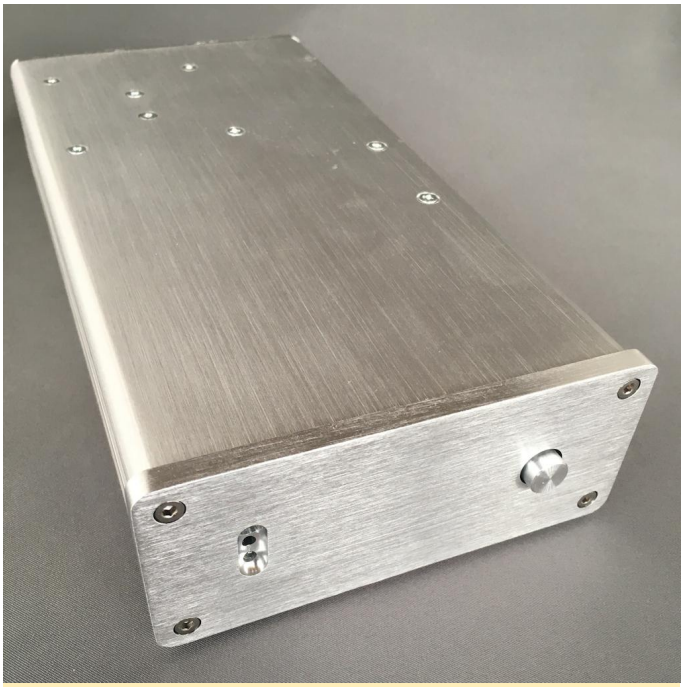


Figura 6 – Frontal del chasis

Figura 7 – Parte trasera del Chasis

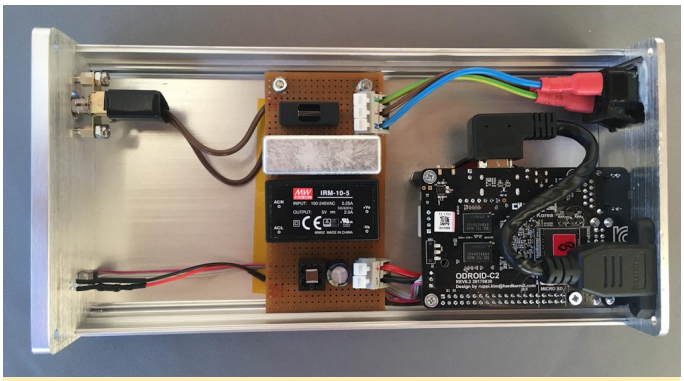


Figura 8 – Componentes internos del chasis

Para comentarios, preguntas y sugerencias, visita el post original del foro en https://forum.odroid.com/viewtopic.php?f=144&t=30505.

NAS Doméstico y Reproductor Multimedia: Montando el Perfecto Sistema de Entretenimiento

May 1, 2018 By Will Santana ODROID-XU4, Tutoriales

Si tienes edad suficiente, puedes recordar e incluso identificarte. Imagínate esto: principios de la década de 2000; DivX y más tarde, su rival XviD en términos de software, y Pentium 4 y Athlons en términos de hardware han hecho finalmente que la compresión de video se convierta en algo; no más cintas voluminosas y mohosas de VHS; Napster en sus mejores días, cambio la historia del P2P para siempre. Aquí en Sudamérica, el acceso telefónico a Internet tenia los días contados y el ADSL empezaba a implantarse. Las unidades de CD-RW eran relativamente accesibles al igual que sus soportes, especialmente con los nuevos y recientes DVD. Youtube ni siquiera había nacido. Júntalo todo y tendrá el escenario perfecto en tu casa:

Figura 1

Una torre de CD repleta de soportes físicos que contienen software, archivos mp3 y películas. Mi torre

era exactamente como la de la foto. Podías apilar hasta doscientos CD. Cuando me hice con ella, cambié los estuches de CD estándar por estuches más delgados, doblando la capacidad.

Avancemos ahora unos cuantos años hasta principios de la década de 2010. Youtube ya existía, pero no era tan grande como lo es hoy día. Netflix acaba de llegar a Sudamérica, con solo algunos títulos desconocidos en su cartera. Un ADSL más rápido, de 2 a 5 mbps, ya estaba ampliamente disponible en la mayoría de las grandes ciudades de por aquí. Descargar no solo películas, sino series enteras, se convirtió en un gran problema. Aunque las unidades de DVD y los DVDs se habían vuelto más baratas, todavía no eras más barato que el precio por Mb de las unidades de disco duro (HDD) antes de las inundaciones en Indonesia (**https://goo.gl/rP6kyE**). En lugar de administrar una gran cantidad de medios, algunos de los cuales empezaban a ir mal, ¿por qué no tenerlo todo en discos duros, disponibles a solo un clic? ¿Qué hay de todas las cubiertas de CD y DVD? ¿Guardaría todas las fotos de mi familia en una unidad de disco duro mecánica propensa a dar fallos?

Cuando empecé a plantearme cómo organizar todos los medios, algunas investigaciones me llevaron a pensar que XBMC (renombrado como Kodi años más tarde) sería la solución perfecta. XBMC es capaz de gestionar una biblioteca multimedia, descargar portadas y letras, reproducir casi cualquier códec de video con subtítulos, mostrar mis fotos familiares y mucho más. Todo lo que necesitaba era llevar a cabo un extenso trabajo para estandarizar los nombres de las carpetas y de los archivos.

¿Y los datos en sí? Los discos fallan y presentan bloques defectuosos con el tiempo. RAID está bien, pero presenta problemas cuando trabajamos con una gran cantidad de archivos casi estáticos: ¡la eficiencia! RAID 1 duplica el coste de los discos, el espacio, el ruido y la energía. RAID 5 parece estar bien, pero ¿qué pasa si falla un disco y no puedo conseguir otro del mismo tamaño inmediatamente? ¿Qué pasa si dos discos fallan? Hay RAID 6 pero pocos controladores lo admiten. ¿Qué pasa si el controlador falla? ¡Diferentes fabricantes tienen diferentes implementaciones, haciendo que la recuperación de todos los datos se convierta en una auténtica pesadilla!

Ahora le toca el turno a RAID Snapshot. Comparado con el RAID habitual, no funciona a nivel del disco. Más bien, funciona a nivel de datos, sobre cualquier sistema de archivos. Siempre que dispongas de un disco con una cantidad de espacio igual o superior que los datos del disco más grande de la matriz, todo irá bien. Incluso puedes usar discos de diferentes tamaños. El problema lo encontramos en la falta de velocidad y disponibilidad de los datos. Algunos controladores RAID 5 disponen de reconstitución de datos en tiempo real siempre que solo falle uno de los discos. La paridad RAID se calcula en tiempo real a medida que los datos van cambiando. Está bien, pero mantiene todos los discos activos la mayor parte del tiempo, incluso si simplemente se está modificando un pequeño archivo. El RAID Snapshot, como su nombre indica, funciona calculando la paridad con instantáneas. Si falla un disco, te quedarás con la última instantánea. Lo bueno es que cualquier archivo que no esté en el disco que ha fallado estará disponible para usarse, sin necesidad de hacer reconstrucción.

También tenía la intención de fusionar todos los datos y mostrarlos como un gran disco, casi como JBOD, pero con seguridad. Usar múltiples discos está bien, pero hace que la gestión de los soportes resulte demasiado compleja. Con esto en mente, y con planes de usar también el servidor como un PC para juegos (que limitaba las opciones del sistema operativo a Windows) decidí recurrir a FlexRAID (**http://www.flexraid.com/**). Por aquel entonces aún se encontraba en su fase beta, pero mostraba un gran potencial. Para mi sorpresa, como traductor brasileño, recibí una licencia completa cuando se hizo de pagó.

Con todas las necesidades de software cubiertas, llegó el momento de mancharse las manos y montar el hardware. Como se suponía que iba a ser un PC multimedia y para juegos, con una o dos unidades de disco duro de 1.5TB serían suficientes para almacenar todo mi contenido multimedia en ese momento. Me decante por esta carcasa Sentey para que fuera el

rack de mi sala de estar. Explicaré porque tiene tanto polvo más adelante.

Figura 2

La carcasa podría albergar hasta tres discos duros de 3,5″, una unidad de DVD-RW, una placa base micro ATX, una completa fuente de alimentación ATX y una GPU de la misma altura. Funcionó estupendamente hasta que añadí el tercer disco. No importa cuántos ventiladores agregase o a cuanta velocidad girasen (produciendo mucho ruido), al final se sobrecalentaba el sistema.

Se hizo bastante obvio que necesitaría más ventiladores y una carcasa adicional para incluir más discos a medida que la biblioteca crecía rápidamente como resultado de la comodidad de disponer de un servidor 24 horas y una conexión a Internet más rápida.

Simplemente se dio el caso de que tenía esta vieja carcasa de escritorio Compaq por los alrededores:

Figura 3

Con unas “cuantas” modificaciones no precisamente bonitas en la carcasa y en la fuente de alimentación, se convirtió en esto:

Figura 4

Agregué siete discos duros y dos controladores HDD adicionales, ya que la placa base solo podía manejar dos discos. En ese momento, la GPU, el ventilador adicional de la parte trasera y el de 120 mm de la parte superior de la carcasa aún no habían sido colocados. Esta era la placa base original, que

reemplacé unos meses después por una que funcionaría mucho mejor con el procesador. Los cuatro ventiladores de 80 mm en la parte delantera, colocados para enfriar los discos duros, fueron retirados para ser limpiados. El conector de alimentación está ubicado en el frontal.

Especificaciones:

- Placa base: Abit VA-10 (cambiada después)
- Procesador – AMD Athlon XP 2000+ (cambiado por un Atlhon X2 después)
- RAM – 2 Gb DDR 333 (en realidad, DDR 400 con bloqueo de reloj) (se agregaron dos más)
- Controlador RAID Sil 3112
- Controlador RAID Sil 3114

HDDs: (los pequeños se cambiaron por unos más grandes después)

- 1x40Gb IDE (Sistema)
- 3×1.5Tb SATA (Multimedia)
- 1×1.5Tb SATA (Datos personales)
- 1x500Gb SATA (Descargas y Memoria virtual)
- 1×1.5Tb SATA (Pariedad)

Software:

- Windows 7 32 bits
- Flexraid
- XBMC

Esta configuración funcionó decentemente durante semanas hasta que mi entonces novia (ahora mi esposa) empezó a quejarse del ruido. He de admitir que era RUIDOSO, especialmente con los ventiladores adicionales que acompañaban a la tarjeta gráfica. Con una boda a las puertas y sin planes de abandonar a mi novia, decidí ampliar la reforma del baño que planeamos para la sala de estar y los dormitorios. ¿Recuerdas todo el polvo que se veía en la primera foto? Esta es la razón:

Figura 5

Figura 6

Figura 7

Figura 8

Sí, nosotros usamos aquí en Brasil la albañilería, no placas de yeso. Y sí, reformar un apartamento al mismo tiempo que se vive en él prácticamente se puede comparar al infierno sin exagerar. Puedes ver la torre de CDs en las dos primeras fotos, junto a la mesa. La carcasa del ordenador está en la parte derecha de la primera foto. Todas esas cosas amarillas son conductos para ocultar los cables que conectarían los televisores del dormitorio y de la sala de estar al servidor, ahora apilados en un armario del pasillo. Por supuesto, el calor sería un problema en un espacio reducido. Sin embargo, fueron añadidos más ventiladores al propio armario, bombeando el aire caliente hacia un compartimento superior que estaba ventilado. Tras mucho polvo y varios días de duro trabajo, los resultados se pueden ver en las Figuras 9 y 10.

Figura 9

Figura 10

El servidor aún podía escucharse, incluso con la puerta cerrada, pero nada que ver con el ruido que teníamos antes. En cuanto a la sala de estar, en la figura 11 tienes el resultado.

Figura 11

Si continúas leyendo después de todo esto, probablemente te estés preguntando: "¿Qué papel tiene el ODROID en todo esto?". Después de unos años centrados en ampliar la familia, decidimos mudarnos a un apartamento más grande. Dado que casi todos los apartamentos en los que hemos vivido tienen unos 70 años de antigüedad, era necesario nuevamente hacer reformas, y otra vez tendríamos que convivir con la reforma. Como teníamos una pequeña habitación extra en la parte trasera del nuevo piso, decidí poner el viejo servidor allí, pasando los conductos por las paredes y todo lo demás. Tras varios meses de reforma, llegó el momento de encender el viejo servidor, y después de discutir el potencial de la Raspberry Pi con algunos colegas del trabajo, me vino a la mente una gran idea. Existen pequeñas e increíblemente potentes placas con muchos núcleos de procesamiento y memoria RAM. Mucha gente las utiliza como consolas de juegos retro e incluso mini PC. Las grandes preguntas fueron: "¿Ya está aquí la tecnología?" Y "De ser así, ¿qué placa usar?"

Si tenemos éxito, el nuevo servidor sería mucho más pequeño, más silencioso y con una mayor eficiencia energética que el anterior. Han transcurrido al menos cinco años desde la creación de mi servidor inicial. Unas cuantas semanas de investigación y la respuesta a esa pregunta fue clara: la tecnología había llegado y el **ODROID-XU4** era la opción más obvia. ¿Por qué? Sus ocho núcleos eran mucho más que los dos que tenía en el servidor anterior. Los 2 GB de menos de RAM no sería un problema, ya que no ejecutaríamos Windows, en su lugar usaríamos un liviano Linux. La conexión ethernet Gigabit sería perfecta para conectar todos los dispositivos en el nuevo apartamento. El ODROID-XU4 también cuenta con una GPU muy decente con capacidades para la decodificación por hardware, lo cual era importante, ya que tenía pensado usarlo no solo como servidor sino también como reproductor de video, al igual que el antiguo servidor PC. Por último, pero no por ello menos importante, era la disponibilidad del USB 3.0. Como el XU4 no tiene interfaces SATA (aunque tuviera uno o dos, no serían suficientes), el USB 3.0 era la alternativa perfecta. Sus velocidades teóricas de 625 MB/s son mucho más rápidas que cualquier disco rígido mecánico. De hecho, cuando probé los HDD que estaba usando, el más rápido solo llego a los 120MB/s estando conectados directamente a un PC vía SATA o usando un adaptador SATA-USB a través de USB 3.0.

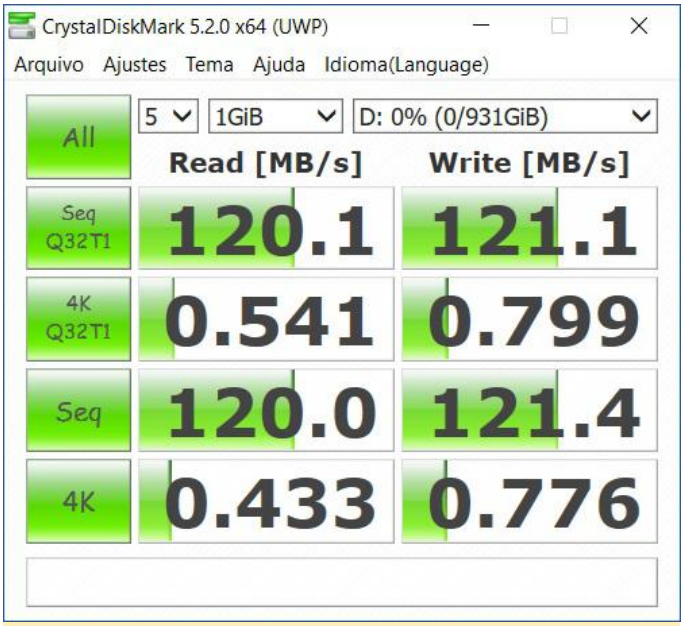


Figura 12

Una cuestión importante era que ya no podría usar las viejos HDD de 3.5". Consumen mucha energía, son voluminosos y necesitan 12V para funcionar. Podría haber usado un PC o una fuente de alimentación ITX para alimentarlos, pero una vez más esta solución habría sido voluminosa e ineficiente. Así que las unidades de disco duro de 2.5" entraron en el juego, ya que son pequeñas, resistentes, eficientes con el consumo de energía, silenciosas y pueden funcionar con solo 5V desde un puerto USB si tienes suficiente flujo eléctrico para ellas.

Pedí el ODROID-XU4 a Hardkernel y en unos días lo tenía en mis manos. ¡Qué maravilla! Incluso cuenta

con un ventilador inteligente para reducir el ruido. Unos días después, recibí algunas unidades de disco duro y un hub USB 3.0 autoalimentado para empezar con las pruebas. Antes de poner a prueba la tecnología, en la siguiente imagen puedes comparar la carcasa del ODROID que contenía el XU4 y el adaptador de corriente, en comparación con el servidor anterior. No se puede ver, pero dentro de la carcasa, junto con el XU4 y su adaptador de alimentación, había siete discos duros con sus adaptadores SATA-USB, un concentrador USB y su adaptador de alimentación, y muchas tarjetas SD con imágenes del sistema operativo. Tener todas estas cosas en una carcasa del mismo tamaño que la PSU anterior no estaba nada mal.

Figura 13

En la versión final del servidor antiguo, la PSU ya no cabía dentro de la carcasa junto con la unidad de DVD (el cable rojo de la izquierda era utilizado para ello).

Durante las primeras pruebas, el hub USB alimentado creó un cuello de botella. Un puerto USB 3.0 solo puede llegar hasta 900 mA, lo cual estaba muy lejos de la cantidad necesaria para ejecutar más de uno o dos discos, dependiendo el modelo utilizado. Los discos que funcionaban aleatoriamente durante el arranque y el uso general lo dejaba muy claro. El primer hub USB con alimentación que probé no proporcionaba la suficiente potencia para solo dos discos. Incluso intenté ejecutar uno directamente desde el puerto USB 3.0 del XU4 y otro desde el USB 2.0 para dividir la carga, pero los que estaban conectado al hub continuaban deteniéndose.

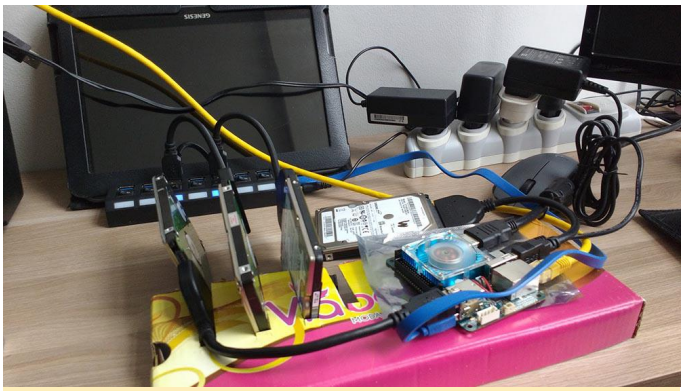


Figura 14

La solución pasaba por buscar un hub USB con una buena alimentación. Tras investigar sobre ello y leer algunas críticas, decidí optar por un hub USB 3.0 Xcellon de 10 puertos. Está hecho de aluminio por lo que disipa el calor. El 5A de su adaptador de corriente demostraron ser suficientes para ejecutar al menos los cinco discos que ahora uso. Nunca lo he llegado a probar con los siete que tengo.

Figura 15

Testeado el hardware, era hora de centrarse en el software. Como Windows ya no era una opción, la pregunta se centraba en qué distribución de Linux usar. Como he dicho antes, la PC anterior hacia de NAS, reproductor multimedia, máquina de juego y gestor de descargas con Sonarr y Torrent. No aceptaba perder ninguno de estos servicios.

En primer lugar, me centré en la disponibilidad y en la seguridad de los datos. No almacenaría mis datos en un servidor poco fiable. La alternativa más sencilla y segura que pude encontrar fue OpenMedia Vault (**https://www.openmediavault.org/**). Es de código abierto y tiene todo lo que necesitaba, en una interfaz

web fácil de usar con multitud de complementos. Como está basado en Debian, la elección del sistema operativo ya estaba hecha. Descargué una imagen de Debian de los foros de Hardkernel y comencé a instalarlo todo. Me llevo algunas semanas configurarlo todo. El complemento Greyhole de OMV se encargaría de fusionar todos los discos en uno. El complemento SnapRAID se ocuparía de la paridad de datos. OMV gestionaría el resto y me dejaría configurar los dos. Pero justo entonces, se me ocurrió algo. Después de tantos años, ya no dispondia de una conexión de acceso telefónico de 56 kbps, sino una por cable de 60 mbps. Proteger algo más que las fotos de mi familia y mis documentos personales ya era irrelevante. Ahora puedo ver la mayoría de las películas y series via streaming o simplemente descargarlas en minutos si no es posible hacer streaming. No necesito tanto espacio en el disco. Tal vez ni siquiera paridad. Como Greyhole ofrece una opción para replicar los datos en tantos discos duros como se quiera, y como mis datos confidenciales no son tan grandes, disponer de dos y tres copias sería suficiente.

Pero dado que una copia local no es una copia de seguridad, he configurado estos datos para que también se sincronicen con la nube (copia de seguridad real), por si las moscas. De modo que, tras unas semanas abandone SnapRAID porque ya no me era necesario. Todavía puede ajustarse a tus necesidades, si tiene una gran cantidad de datos sensibles que cambian muy poco. Ejecuté este sistema durante aproximadamente dos meses sin problemas, de modo que la elección fue la correcta. Era hora de volver a centrar la atención en la reproducción multimedia y en los juegos. Indagando en cómo instalar Kodi en Debian me topé con el increíble trabajo de Meveric: ODROID GameStation Turbo (**https://forum.odroid.com/viewtopic.php?f=98&t=7322**).

Está basado en Debian, aunque tiene muchas optimizaciones para la reproducción de video y los juegos retro. Unos días y lo tuve todo fusionado. Después de dos o tres meses funcionando bien, empecé a experimentar algunas caídas del sistema en ocasiones. Resultó ser que la tarjeta SD estaba dando sus últimos coletazos. Como no pude recuperarla, la solución pasaba por reconstruir todo desde cero. Pensé que, si reconstruía el software, ¿por qué no el hardware? Durante meses, un gran manojo de cables se escondería detrás de la puerta de mi sala de estar.

Con un poco de creatividad y una herramienta Dremel, convertí un mini organizador de escritorio de dos cajones en la última versión de mi servidor. Aquí tienes fotos del proceso:

Figura 16 – Probando los espaciadores

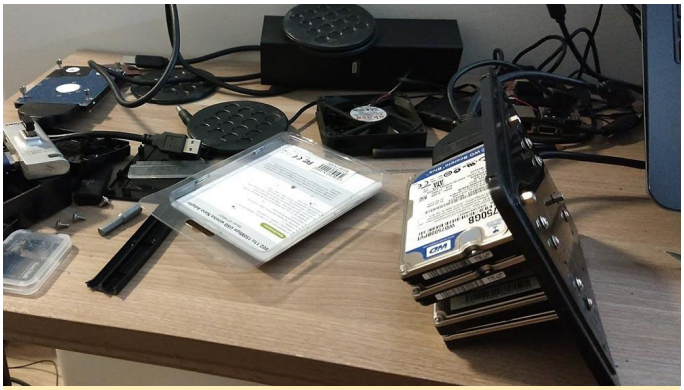


Figura 17 – Taladrando los agujeros, fijando los espaciadores y las unidades de disco duro a la “carcasa”

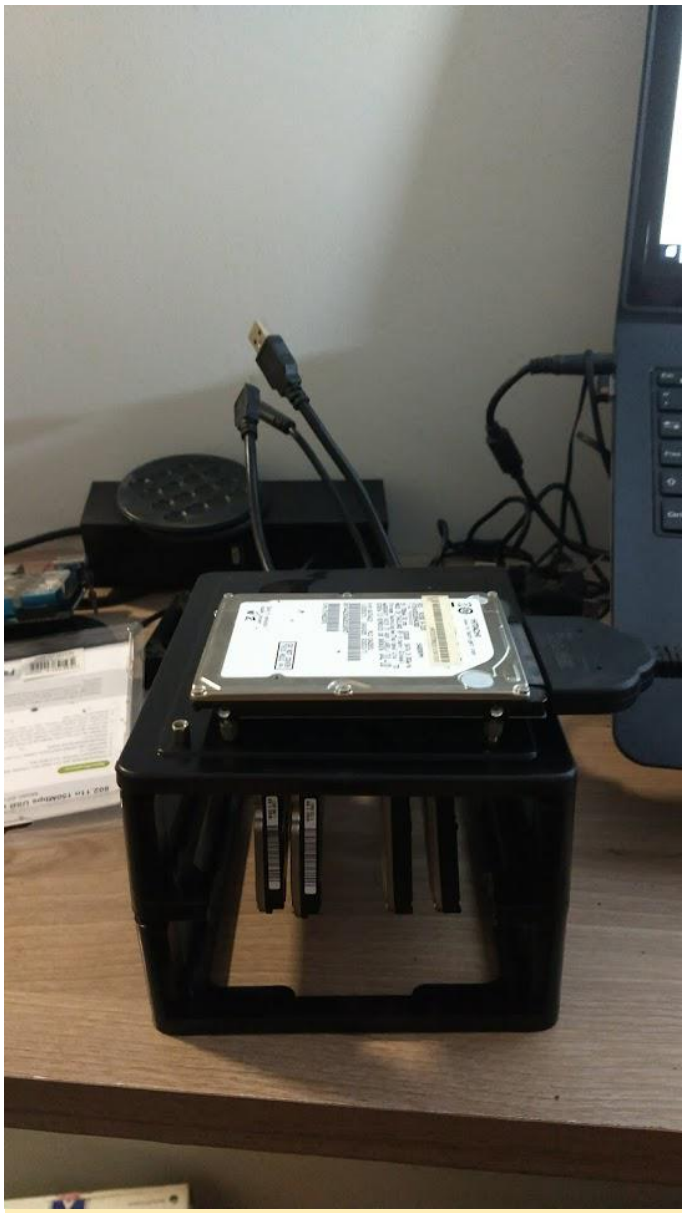


Figura 18 – Disco duro con los archivos del SO/Temp en la parte superior para facilitar el acceso

Figura 19 – No olvidemos la estrella del show. Coloqué el ODROID-XU4 (y un ventilador también)

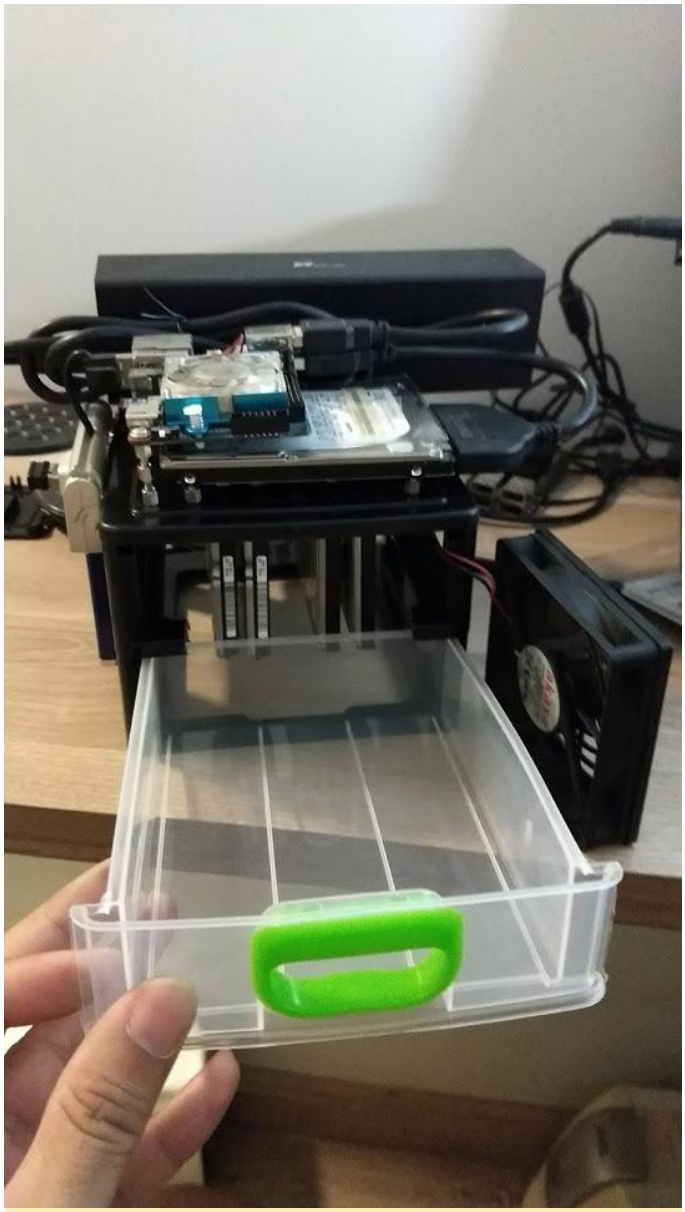


Figura 20 – El Hub USB, algunos cables y un recuerdo de lo que fue la “carcasa” en primera instacia

Figura 21 – Todo junto

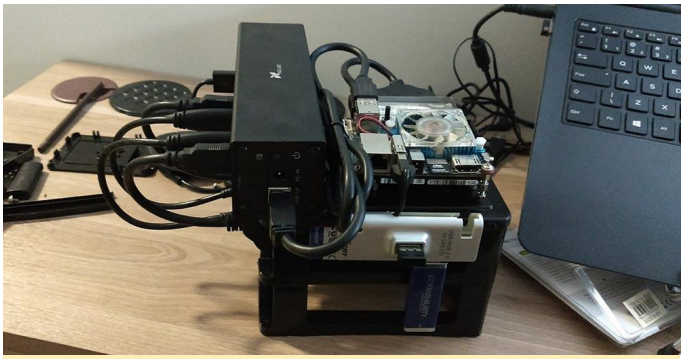


Figura 22 – A la izquierda hay un Hub USB 2.0 con el receptor de mi teclado inalámbrico y una vieja unidad flash de 4GB que es necesaria para mantener algunos datos de SnapRAID, que permanecerá allí por si decidiera volver a utilizarlo

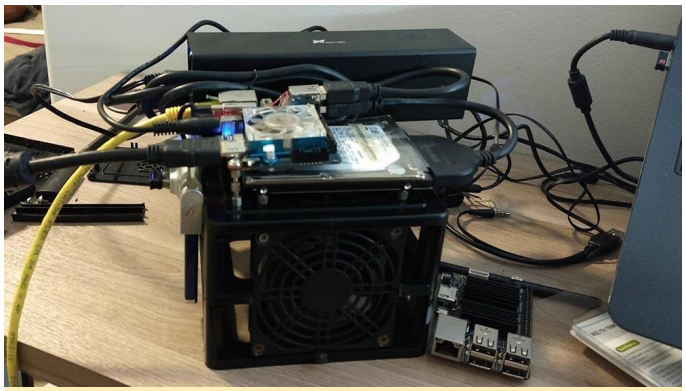


Figura 23 – Todo encendido. A la derecha del servidor, el ODROID-C2 que utilizo como reproductor multimedia con el televisor de mi dormitorio

En este último montaje, añadi uno de los ventiladores del viejo servidor para que ayudase a enfriar los discos duros y quizás extender su vida útil. La unidad de disco duro en la parte superior contiene los archivos del SO/descargas/temporales. Pude haberlo montado detrás del ventilador para mejorar su refrigeración, pero opté por hacerlo más accesible para posibles mantenimientod. Es más fiable que la tarjeta SD anterior, pero no es inmune ante posibles fallos. Además, puedo conectar un disco duro adicional al servidor a través del Hub USB, si fuera necesario. Todo el sistema es tan silencioso hasta el punto de haberlo ubicado en el estante del TV en la sala de estar. Sin fijarme en los LED, no puede saber si está encendido o apagado.

Figura 24

Esto no quiere decir que todo fuera perfecto. Hace unas semanas, el sistema se colgó un par de veces. La causa fue fácil de detectar. De nuevo, sobrecalentamiento. El servidor estaba en el estante superior del mueble, en el mismo lugar donde ahora está el estéreo. Es difícil de ver, pero también hay un APC nobreak 400VA en la parte trasera. El cableado sale por un agujero de la parte trasera del estante inferior. El calor del servidor y de las fuentes de alimentación, junto con el nobreak y la falta de ventilación hicieron que el ODROID-XU4 trabajase a unos 79 °C (174 ° F) todo el tiempo, incluso sin carga de trabajo. Es probable que las unidades de disco duro también lo hicieran, especialmente la que contiene el sistema operativo. Con una carga media de trabajo, la CPU y/o las unidades de disco duro (me olvidé de controlar las temperaturas) sobrepasaron su límite, deteniendo todo el sistema. Cambiando el estéreo por el servidor resolvió el problema. El calor ahora puede fluir a través del orificio del cableado. Incluso después de seis horas de una maratón de series, el servidor continuó funcionando bien durante semanas sin ningún tipo de cuelgue. Las temperaturas cayeron alrededor de 4 ° C (alrededor de 7 °F). Podrían bajar aun más, abriendo la puerta del estante siempre que sea necesario. Si el sistema no llego a colgarse durante un verano brasileño con temperaturas de hasta 43 ° C (109 ° F), dudo que vuelva a colgarse debido al sobrecalentamiento. En el futuro, podría crear una carcasa en 3D para el servidor. Pero la actual cumple mis necesidades.

La saga no ha terminado, y probablemente nunca lo haga. Me encantaría actualizar a un XU5 con 4 GB de RAM, compatibilidad H.265 y 4K, y un procesador aún más rápido. Sería la plataforma de transcodificación/reproducción perfecta. Ejecutar OMV combinado con Emby o Plex sería la mejor fuente para la transcodificación y el streaming. Imagine un C3 con WiFi AC 5GHz integrado: ¡sería la bomba, haciendo que cualquier televisor “inteligente” parezca una basura, en términos de potencia y flexibilidad! En este momento, es más que suficiente para ejecutar juegos retro, alamcenar casi todos mis medios multimedia (H.265 está ganando fuerza con 4K), poner a disposición mis archivos en todo el apartamento, administrar y descargar mis películas y series gracias a algo tan pequeño, silencioso y eficiente en términos de consumo de energía. Cualquier medio que no pueda reproducir en el XU4 lo puede hacer el ODROID-C2 de mi dormitorio (otra obra de arte de Hardkernel).

Así que esto es todo. Espero que este articulo les ayude a otros como yo. Si tiene preguntas, puede

encontrarme en los foros de ODROID (**https://forum.odroid.com**) bajo will_santana.

Administrador de Volúmenes Lógicos de Linux (LVM2)

May 1, 2018 By Cristian Sandu Linux

La gestión de volúmenes Lógicos de Linux (LVM) es un software diseñado para añadir una capa entre los discos reales y la vista que hace el sistema operativo de los mismos para facilitar su administración, reemplazo y ampliación. Suele usarse en los centros de datos que usan hardware de actualización de discos, además de replicar datos para evitar pérdidas. Existen, por supuesto, alternativas: el hardware RAID es mejor en cuanto a rendimiento, pero más restrictivo: por ejemplo, no puedes reemplazar limpiamente un disco en un RAID0; luego está el mdadm, o software RAID, que es una implementación de software (nivel del sistema operativo) de RAID, pero tiene deficiencias similares. LVM es más flexible, permite una configuración que RAID no puede hacer. Dicho esto, como se trata de una mera solución de software (compuesta por módulos kernel y daemons de espacio de usuario), implica una cierta pérdida de rendimiento, bajará algo la velocidad con respecto al uso de discos de forma nativa.

El wiki de Debian explica bastante bien los conceptos básicos, de modo que no le voy a hacer la competencia, mira esto: **https://wiki.debian.org/LVM**. Si quieres un tutorial más detallado sobre LVM, recurre al excelente manual de RedHat en **https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Cluster_Logical_Volume_Manager/ch_Introduction-CLVM.html**.

Los conceptos básicos que usaré en este artículo son:

- PV – volumen físico
- VG – grupo de volumen
- LV – volumen lógico

Todos los comandos LVM usan estas iniciales para referirse a los conceptos anteriores.

Instalación

Si estás ejecutando alguna de las imágenes oficiales de Ubuntu del repositorio de Hardkernel, esto es todo

lo que necesitas hacer:

```
$ sudo apt install lvm2
```

Esto instalará los paquetes del kernel, el daemon del espacio de usuario y todo lo demás que necesitas para trabajar con LVM.

Cloudshell2

Cloudshell2 ofrece RAID por hardware con 2 discos, pero la capacidad de actualizar tu disco está algo limitada a menos que tenga una forma de clonar la matriz cada vez que quieras actualizar. La Wiki ODROID explica cómo configurar el controlador JMicron en **https://wiki.odroid.com/accessory/add-on_boards/xu4_cloudshell2/raid_setting**. Si quieres usar LVM, necesitarás utilizar la configuración JBOD. También puedes ejecutar LVM sobre una configuración RAID hw como RAID0 o RAID1 pero creo que el contexto de tan solo 2 discos es lo que elimina cualquier ventaja que el LVM pueda aportar. Después de conectar tus discos, querrás particionarlos. Los documentos LVM recomiendan no utilizar discos raw como los PVs (volúmenes físicos) y usar particiones en su lugar, incluso si se trata de una partición que se extiende a lo largo de todo el disco. En mi configuración hice esto, utilicé 2 HDDs de 3TB que contienen una partición que cubre todo el disco.

Una forma rápida de particionar tu disco es usar los siguientes comandos:

```
$ sudo fdisk /dev/sda
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-621, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-621, default 621):
w
```

Para obtener más información sobre el particionado de discos, consulta **https://www.tldp.org/HOWTO/Partition/fdisk_partitioning.html**.

Una de las mejores cosas que tiene el LVM es que puedes planificar y simular tu configuración con dispositivos en un circuito cerrado antes de llegar a implementarla realmente. La siguiente sección detalla la configuración con 2 discos de tamaño idéntico utilizados para 2 volúmenes seccionados (por rendimiento, no por seguridad). Como paso adicional, también vamos a simular la actualización de uno de los discos con un disco nuevo, dos veces su tamaño, esto también lo hice, simulé y luego implementé. En esta parte final del escenario, tu volumen ya no se seccionará porque los discos ya no tienen el mismo tamaño.

Configurar dispositivos en circuito cerrado

```
$ dd if=/dev/zero of=/tmp/hdd1.img bs=1G count=1
$ dd if=/dev/zero of=/tmp/hdd2.img bs=1G count=1
# twice the space
$ dd if=/dev/zero of=/tmp/hdd3.img bs=1G count=2

$ losetup -f

$ losetup /dev/loop0 /tmp/hdd1.img
$ losetup /dev/loop1 /tmp/hdd2.img
$ losetup /dev/loop2 /tmp/hdd3.img
```

Así que hemos creado 3 discos: dos discos de 1GB y un disco de 2GB, puedes usar las dimensiones que desees, son las proporciones las que importan y no los tamaños reales. Crea los volúmenes físicos (PV). $ pvcreate /dev/loop0 $ pvcreate /dev/loop1 $ pvcreate /dev/loop2

Lo que esto ha hecho es decirle a LVM que planeamos usar estos discos como soporte físico para nuestros futuros volúmenes lógicos. Lo que tenemos que recordar es que a cada PV se le asigna un ID único que se escribe en el disco para que, incluso si mueves los discos de tu sistema, LVM los encuentre en función de sus ID. Esto nos será muy útil cuando vayamos a actualizar nuestros discos de la carcasa Cloudshell2 y uno de los discos nuevos se conectará a través de USB 3.0 y luego se intercambiará con uno de los discos de la carcasa.

Tendremos que poner nuestros PVs en un Grupo de volumen antes de usarlos para crear volúmenes lógicos, este es un paso obligatorio, también ten en cuenta que no es posible crear volúmenes lógicos utilizando PVs desde diferentes VGs.

```
$ vgcreate vgtest /dev/loop0 /dev/loop1
```

Ahora que hemos creado un VG usando nuestros 2 discos simulados de 1GB, podemos verificar el estado de nuestra configuración en cualquier momento usando estos comandos

```
$ pvs
$ vgs

$ pvdisplay
$ lvdisplay
```

Crear volúmenes lógicos (LV)

Ahora crearemos nuestros volúmenes que se convertirán en las unidades que el sistema operativo considera utilizables. En este contexto, estoy creando 2 volúmenes seccionados, uno de 1 GB y otro que simplemente se llenara con el espacio restante.

```
$ lvcreate -i2 -I4 -L1G -nlvdata1 vgtest
$ lvcreate -i2 -I4 -l100%FREE -nlvdata2 vgtest
```

Los parámetros son:

- -i2 : tira este volumen entre 2 PVs
- -I4 : amplía el tamaño (el equivalente de un bloque en el lenguaje LVM) será de 4 MB
- -n : qué nombre el volumen

El último parámetro es el grupo de volumen para operar. El tamaño se especifica con la opción -L o -l, -L requiere tamaños específicos, mientras que -l permite porcentajes y otros especificadores. Al final, tendremos 2 volúmenes que están distribuidos uniformemente en nuestros 2 PV en franjas, similar a un RAID0 (en realidad, 2 RAID0s, uno para cada volumen lógico o LV). Llegados a este punto, también tendremos que formatear nuestros nuevos volúmenes lógicos con el sistema de archivos que queremos utilizar, ejecuta los siguientes comandos:

```
$ mkfs.ext4 /dev/mapper/vgtest-lvdata1
$ mkfs.ext4 /dev/mapper/vgtest-lvdata2
```

Al igual que en cualquier partición normal, excepto en la ruta de los dispositivos, estos dispositivos lógicos están expuestos por LVM. Para mayor seguridad, monta estos discos y escribe algunos archivos de prueba, como si montaras un disco normal. Esto te permitirá probar la integridad al final.

Una vez que lo domines, puedes implementar el escenario anterior con discos reales en lugar de usar un circuito cerrado. Simplemente reemplaza /dev/loop0 y /dev/loop1 por /dev/sda y /dev/sdb y ajusta los tamaños de tus LVs.

Actualizar tus discos

Ahora, aquí es donde LVM realmente brilla por su ausencia; a diferencia del hardware RAID, que puede ser bastante rígido en cuanto a la capacidad de actualización (a menos que se use una configuración duplicada), LVM permite todo tipo de configuraciones de discos. Esta parte está basada en el genial tutorial de **https://www.funtoo.org/LVM_Fun**.

El escenario que vamos a implementar es el siguiente: reemplazaremos uno de los discos de la configuración por uno que tenga el doble de capacidad que el original, por ejemplo, si tenemos discos de 2x2GB, reemplazaremos uno de ellos por un disco de 4GB. Para identificar los discos, usa este comando:

```
$ sudo smartctl -i /dev/sda1
$ sudo smartctl -i /dev/sdb1
```

Asegúrate de ejecutar esto en las particiones y no en los discos. Debido al controlador JMicron que hay delante de nuestros discos, no recibirán ninguna información de los discos. El comando anterior te devolverá el código del producto, como ST2000DM para un Seagate Barracuda de 2TB. Esto to ayudará a decidir qué disco físico deseas reemplazar.

El procedimiento completo es:

- Conectar el nuevo disco de repuesto en el segundo puerto USB3.0 utilizando una carcasa externa (el cloudshell solo admite 2 unidades SATA y ambos puertos ya están ocupados de momento)

- Crear un PV (volumen físico) en el nuevo disco
- Agregar nuevo PV al VG existente (grupo de volumen)
- Desmontar todos los volúmenes de VG y/o congelar la asignación en el PV para migrar
- Pvmove uno de los 2 PV existentes en este nuevo PV
- Dejarlo trabajar toda la noche ya que se va a copiar un disco de 2tb sector por sector
- Reducir VG eliminando el PV anterior (el que se movió en el paso anterior)
- Apagar y cambiar el disco viejo por uno nuevo
- Arrancar y comprobar que los LVs (volúmenes lógicos) están asignados correctamente a los PVs

Ten en cuenta que no todas las carcasas USB3.0 externas serán compatibles con el controlador UAS. Yo utilicé una de la marca ORICO.

Al igual que ocurre con todas las cosas de LVM, puedes (y deberías) simular la actualización antes de ejecutarla.

Conectar el nuevo PV

En primer lugar, conectemos el tercer dispositivo del bucle que hemos creado anteriormente (el de 2GB) a nuestro VG existente:

```
$ vgextend vgtest /dev/loop2
```

Migrar el viejo PV al nuevo PV

En este paso, migramos el disco viejo al disco nuevo:

```
$ pvmove --atomic -v -A y /dev/loop1
/dev/loop2
```

Aquí tenemos 2 parámetros importantes:

- – –atomic: el movimiento se realizará de forma transaccional, si falla en algún punto, simplemente se revierte, no se pierde la información.
- -A y: realiza automáticamente una copia de seguridad de la configuración de LVM para restaurar en caso de que algo salga mal. La herramienta que necesitarás usar despues es vgcfgrestore.

Si interrumpes el proceso o experimentas una pérdida de energía, puede reiniciar el proceso ejecutando el siguiente comando:

```
$ pvmove
```

Aunque, te sugiero abortar y empezar de nuevo:

```
$ pvmove --abort
```

Debido a que nuestro pvmove era atomic, esta interrupción lo restaurará todo a su estado original (si no usáramos -atomic entonces algunas PE – extensiones físicas se moverían y otras seguirían estando asignadas al volumen anterior y tendríamos que moverlas manualmente). En el mundo real, este paso lleva bastante tiempo, y normalmente lo dejo trabajar durante toda la noche (estamos moviendo 2 TB de datos).

Redimensionar el nuevo PV

Ahora necesitamos cambiar el tamaño del PV recientemente movido para incluir el espacio libre adicional en el disco, esto simplemente se hace con el siguiente comando:

```
$ pvresize /dev/loop2
```

Redimensionar el volumen lógico

Ahora podemos aprovechar ese nuevo espacio en disco y ampliar uno de nuestros LV para incluirlo. Debido a que nuestra configuración utiliza franjas, y debido a que este nuevo espacio libre está solo en un PV, no podremos hacer que el nuevo espacio tenga franjas, de modo que necesitaremos usar este comando:

```
$ lvextend -i1 -l +100%FREE
/dev/vgtest/lvdata2
```

El parámetro -i1 le dice a LVM que estamos reduciendo a solo 1 franja. Esto dará como resultado un rendimiento de impacto general, ya que los datos escritos en esta parte del volumen estarán en un solo disco. Al ejecutar el comando “lvdisplay -m”, podemos revisar nuestra configuración resultante:

```
$ lvdisplay -m /dev/vgtest/lvdata2
  --- Logical volume ---
  LV Path                /dev/vgtest/lvdata2
  LV Name                lvdata2
  VG Name                vgtest
  LV UUID                vDefWQ-1ugy-1Sp5-T1JL-
8RQo-BWqJ-Sldyr2
  LV Write Access        read/write
```

```
  LV Creation host, time odroid, 2018-03-06
17:43:44 +0000
  LV Status              available
  # open                 0
  LV Size                1.99 GiB
  Current LE             510
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:2

  --- Segments ---
  Logical extents 0 to 253:
  Type                striped
  Stripes             2
  Stripe size         4.00 KiB
  Stripe 0:
    Physical volume   /dev/loop0
    Physical extents  128 to 254
  Stripe 1:
    Physical volume   /dev/loop2
    Physical extents  128 to 254

  Logical extents 254 to 509:
  Type                linear
  Physical volume     /dev/loop2
  Physical extents    255 to 510
```

Como puede ver, el segundo LV contiene un segmento lineal al final, ese es el nuevo espacio que acabamos de agregar y que no se puede franjear. En teoría, si también reemplazas el segundo disco, puedes volver a franjearlo, pero aún no he encontrado una forma segura de hacerlo. Si lo hago, escribiré otro artículo sobre ello.

Reciclar el disco de repuesto

Ahora es el momento de eliminar el disco que migramos desde nuestra configuración:

```
$ vgreduce vgtest /dev/loop1
```

Esto solo lo elimina del grupo de volúmenes. Si quieres también puedes usar pvremove para borrar la etiqueta LVM. También vamos a simular la eliminación física del disco:

```
$ losetup -d /dev/loop1
```

Ahora, vamos a simular la parte en la que apagamos el sistema y colocamos el nuevo disco directamente en Cloudshell2 (recuerda que lo teníamos en una carcasa externa), reemplazando efectivamente el anterior. En este paso, el disco 3 se desconectará y luego volverá como un nuevo disco:

```
$ losetup -d /dev/loop2
$ losetup /dev/loop1 /tmp/hdd3.img
```

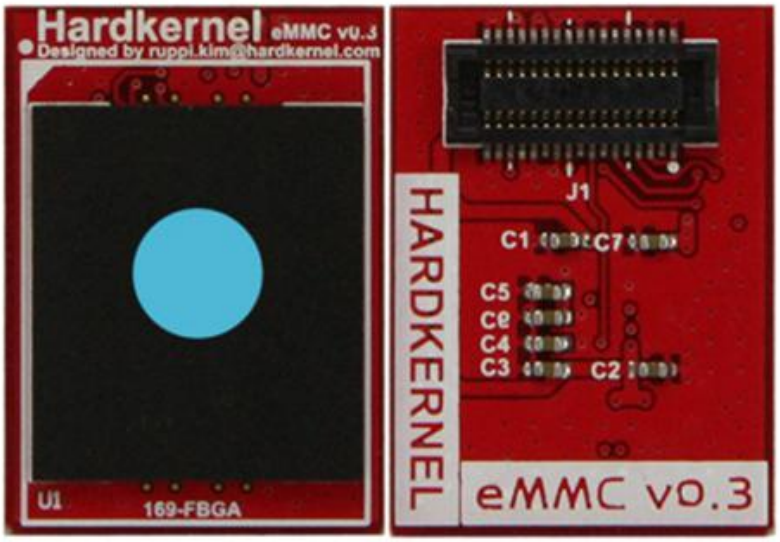
Si ejecutas pvs, vgs y lvs, deberían indicar que tus volúmenes están intactos:

```
  PV         VG       Fmt  Attr PSize   PFree
  /dev/loop0 vgtest   lvm2 a--  1020.00m  0
  /dev/loop1 vgtest   lvm2 a--    2.00g  0
```

Finalmente, monta los volúmenes y verifica que los archivos de prueba todavía estén allí. Para comentarios, preguntas y sugerencias, visita el artículo original en **https://www.cristiansandu.ro/2018/03/06/lvm-fun-swap-out-disk-in-lvm2-stripe/**.

Lector eMMC USB 3.0

May 1, 2018 By Rob Roy Android, Linux, Mecaniqueo, Tutoriales



La plataforma ODROID de Hardkernel tiene una clara ventaja sobre el resto de ordenadores de placa reducida (SBCs) similares, permite que el módulo eMMC sea retirado y vuelva a grabarse con un adaptador USB externo. Todos los módulos eMMC de Hardkernel se entregan con un adaptador de tarjeta SD que permite al usuario grabar un sistema operativo o inspeccionar el contenido de la unidad en otros ordenadores utilizando utilidades como Etcher o dd. Sin embargo, el adaptador de tarjeta SD requería que se utilizara otro adaptador para acceder a la unidad a través del USB, y muchos adaptadores SD a USB no eran compatibles con el adaptador de Hardkernel.

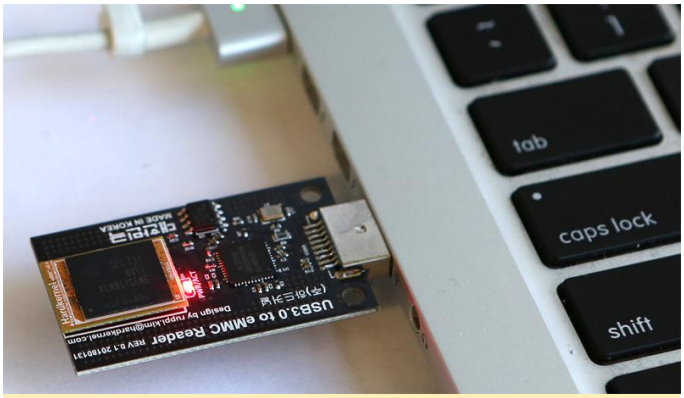


Figura 1 – El nuevo adaptador eMMC USB de Hardkernel es una forma muy cómoda de leer y escribir los contenidos del módulo eMMC desde un ordenador sin la necesidad de comprar un adaptador adicional.

Un nuevo adaptador eMMC USB todo en uno ya está disponible en la tienda Hardkernel en **http://www.hardkernel.com/main/products/prdt_info.php?g_code=G152105300286** por 9.90$, supone una gran mejora sobre el adaptador de tarjeta SD original, puesto que ya no se necesita un adaptador USB de terceros para convertir de SD a USB. Tampoco

tiene los típicos problemas de compatibilidad del adaptador de tarjeta SD original, y se puede usar con cualquier sistema operativo y cualquier plataforma que suporte USB 3.0.

El adaptador original de la tarjeta SD también resultaba algo confuso para algunos usuarios nuevos, que presuponían que el módulo eMMC debía estar conectado al adaptador, y luego lo insertaban en la ranura de la tarjeta SD del ODROID, lo cual disminuía significativamente el rendimiento del módulo EMMC. Todos los ODROID tienen una ranura para eMMC directamente en la PCB, la cual optimiza el rendimiento del módulo eMMC y lo hace mucho más seguro durante su manipulación.

Para usar el lector eMMC USB 3.0, simplemente alinee el módulo eMMC con el cuadro blanco que tiene la etiqueta “eMMC” en la PCB del adaptador, y presiona suavemente hasta que el adaptador encaje en su lugar. Luego, inserte el adaptador USB en la ranura USB 3.0 del ordenador y acceda a él como cualquier otra unidad USB. Se encenderá una luz roja cuando se esté aplicando alimentación USB.

Cuando termines de usar el módulo eMMC en el ordenador, expúlsalo usando la función del sistema operativo para este fin, luego retira el adaptador del puerto USB y desconecta el módulo eMMC del adaptador tirando de él suavemente, retirándolo de la PCB. Finalmente, el módulo eMMC se puede conectar al ODROID apagando primero el ODROID y desconectándolo de la fuente de alimentación, después alinea el módulo eMMC con el recuadro blanco que tiene la etiqueta “eMMC” en el ODROID, y presiona suavemente hasta que el adaptador encaje en su lugar. Conecta la fuente de alimentación al ODROID y éste debería arrancar desde el adaptador eMMC. Algunos modelos de ODROID cuentan con un interruptor de tarjeta SD/eMMC que necesitará estar en la posición “eMMC” para arrancar desde el módulo eMMC.

Para obtener más información sobre cómo grabar sistemas operativos en el módulo eMMC, visita la Wiki de ODROID en **https://wiki.odroid.com/troubleshooting/odroid_flashing_tools**. Ten en cuenta que este lector no funciona con los módulos eMMC Black.

Especificaciones:

- Interfaz USB 3.0
- Amplia Interfaz de datos nativa eMMC de 8 bits, en lugar de la lenta SD de 4 bits
- Funciona en modo HS200
- Compatible con Windows / Mac / Linux
- Funciona con los módulos eMMC Orange, Red y Blue de ODROID
- Usa el software Etcher o Win32DiskImager en tu PC
- No permite acceder a los bloques de inicio ocultos de la eMMC
- Potencia nominal: 5V/500mA (incluido el módulo eMMC)
- Dimensiones: 60x26x4.5 mm

Pruebas de velocidad

Comparamos la velocidad de grabación del sistema operativo entre nuestro Modulo eMMC USB 3.0 y un lector de tarjetas genérico. El Modulo eMMC USB 3.0 es 3-4 veces más rápido que un lector de tarjetas USB 3.0 normal.

- Intel(R) Core(TM) i7-7700 CPU @3.60GHz 3.60GHz / RAM 16GB / 64bit Windows 10
- Etcher version 1.3.1
- eMMC Yellow 64GB
- ubuntu-16.04.3-4.14-minimal-odroid-xu4-20171213.img(1.63GB)

Módulo eMMC USB3.0: Grabación 28.18s, Validación 20.31s, Total 48,49s

Lector de tarjetas Transcend USB3.0: Grabación 93.64s, Validación 81.23s, Total 174.86s

Económica Solución UPS: Asegúrate de que tu ODROID-HC2 Siempre Esté Funcionado al 100%

May 1, 2018 By Neal Kim Mecaniqueo, ODROID-HC2

Muchos sistemas NAS cuentan con una fuente de alimentación ininterrumpida (UPS) para proteger sus valiosos datos de la corrupción accidental provocada por la pérdida del suministro eléctrico principal.

Este artículo te ayudará a montar un sistema UPS para el **ODROID-HC2** utilizando algunos componentes estándar. Está basado en un mini UPS DC muy económico, que creo que es una muy buena alternativa a los costosos UPS actuales. Este mini UPS DC proporciona una salida de 12V. Ésta se puede reducir a 5V usando dos resistencias como divisores de voltaje.

Los siguientes pasos te permitirán crear tu propio UPS.

Listado de Componentes

- Mini UPS DC de 7800 mMH (**https://goo.gl/HjjxHo**), o puede usar cualquier otro UPS DC de 12V si tiene capacidad para 2A o superior
- ODROID-HC2 (**https://goo.gl/1oKiVr**)
- Placa IO USB (**https://goo.gl/GNsp7T**)
- Resistencia axial 10K
- Potenciómetro 10K
- Unos cuantos cables

Desmontar el mini UPS DC y cableado

El mini UPS DC tiene dos partes: por un lado, la PCB y por otro la batería.

Figura 1

Figura 2

El conector de la batería de la PCB proporciona una salida de 12V. El cable rojo soldado está a +12 V y el cable negro está la puesta a tierra (GND).

Cambiar el voltaje de referencia del ADC

La placa IO USB se puede configurar para 3,3V o 5V por defecto, en función de la posición de R1 (se requiere soldadura). La selección pasara a ser el voltaje de referencia ADC.

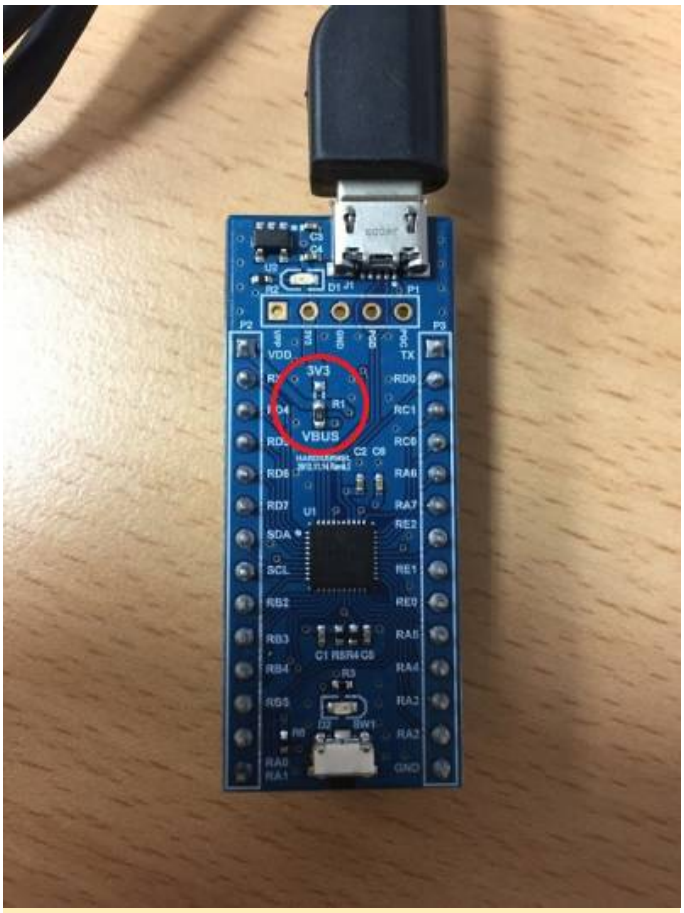


Figura 3

He decidido usar un voltaje de referencia 5V ADC. He soldado la resistencia R1 al VBUS 5V desde 3V3 en la PCB.

Definir el valor de las resistencias

Usando dos resistencias de 10 KOhm (R1) y 7.143 KOhm (R2), podemos dividir la salida de 12V a 5V y 7V usando la fórmula: 12V x (R2 / (R1 + R2)) = 5V Por ejemplo, si R1 es de 10,000 Ohm, R2 es de aproximadamente 7,143 Ohm.

Sin embargo, puesto que no hay resistencia de 7,143 Ohm, utilicé un potenciómetro de 10 K. Con mi elección de R1 de 9.98 KOhm (5V) y R2 de 7.44 KOhm (7V) observé que el mini UPS DC ofrecía algo menos

de 12V cuando estaba completamente cargado. Aumenté el valor de R2 un poco más para que el valor ADC alcanzase los 10 bits de 1024.

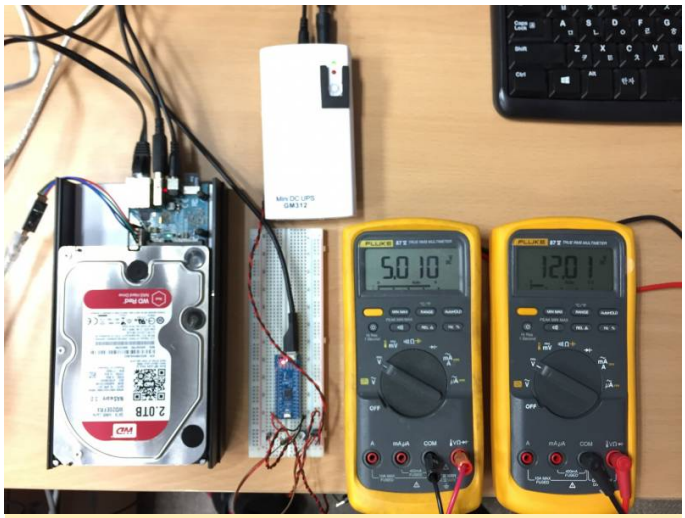


Figura 4

Los diagramas del circuito los tienes a continuación:

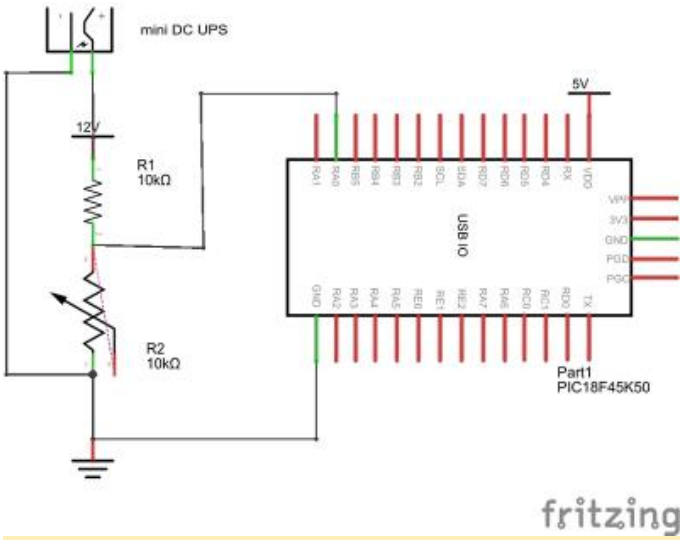


Figura 5

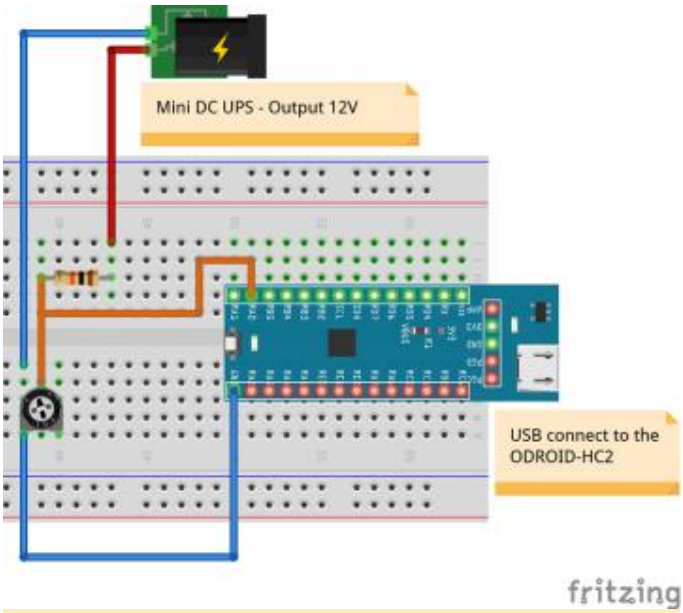


Figura 6

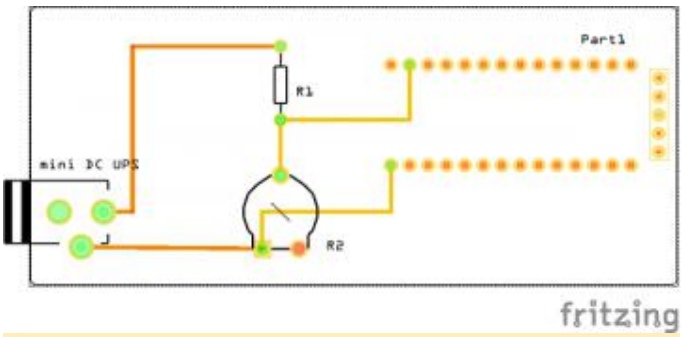


Figura 7

Definir el valor máximo y mínimo de ADC

Compila el software usando los siguientes comandos:

```
$ sudo apt-get install libusb-1.0-0-dev
 $ git clone
https://github.com/hardkernel/Odroid-USBIO
 $ cd Odroid-USBIO/usbio/linux
 $ make
 $ sudo ./usbio
```

A continuación, el orden de las opciones es: a. Toggle LED b. AN0/POT Value c. Button status d. Get Register e. Set Register f. Get Register Bit g. Set Register Bit q. Exit

Usa los siguientes valores:

msb = 512, lsb = 212 potentiometerValue = 724

He ajustado el valor máximo de ADC en 1023 (10 bits en total) a la hora de manipular R2 cuando se carga por completo el mini UPS DC. Tenemos que conocer el valor mínimo de ADC para ver el nivel de batería restante. He descubierto que este valor mínimo de

ADC es de 724 al proporcionar una carga similar a la aplicación de tensión hacia la alimentación suministrada por el sistema usando solo el mini UPS DC.

La siguiente script me ayuda a obtener el valor mínimo de ADC.

```
# cat -n batCheck.sh
 #!/bin/bash

for i in {1..100000}
 do
 ./usbio << endl >> ./adcValue.log
 b
 q
 endl
 echo "`date +%Y/%m/%d-%H:%M:%S` : ${i}"
 sleep 2
 done
 # nohup ./batCheck.sh &
```

Nivel de batería restante

Ahora que hemos calculado el valor máximo y mínimo de ADC, podemos calcular el nivel de batería restante. Puede consultar el script shell en el artículo de la Wiki.

```
Remaining battery level(%) =
  (ADC value - minimum ADC value) x 100
  ---------------------------------------
  (1023 - minimum ADC value)
```

Como he comentado anteriormente, he descubierto que el valor mínimo de ADC es de 724 experimentado un poco, por lo que establecí el valor mínimo de ADsC en 800 para dejar un amplio margen en este script. Si el nivel de batería restante del mini DC UPS es inferior al 10% tal y como he configurado en ${minRemainBat} en la secuencia de comandos, se activará el procedimiento de apagado.

Referencias

Para obtener más información, consulta el artículo de la wiki en **https://goo.gl/zMwFbf**.

Control del ventilador: Adapta el ODROID-XU4 a tu Entorno Perfecto

May 1, 2018 By Justin Lee ODROID-XU4, Tutoriales

El ODROID-XU4 soporta 3 niveles de refrigeración para el ajustar el control térmico: 0, 1, 2. El nivel 0 es el nivel más bajo para el control térmico y hace girar al ventilado a la velocidad más baja. El nivel 2 es el nivel más alto y hace girar el ventilador a la máxima velocidad, tal y como se muestra en la siguiente tabla:

Trip point	-	0	1	2
Temperature	0	60°C	70°C	80°C
Fan speed	0	120	180	240

Esta tabla muestra los valores por defecto de cómo se comporta el ventilador. Cuando la temperatura alcance los 60 °C, el valor de giro objetivo cambiará al nivel 1 y el ventilador comenzará a funcionar a un valor de 120 PWM (0 ~ 255). La misma condición se cumple cuando el valor de giro objetivo pasa al nivel 3, el ventilador funciona a 240 PWM cuando la temperatura alcanza los 80 ° C. Puedes ajustar los valores objetivos y la velocidad de tu ventilador para que sean los que quieras. Incluso puede configurar la velocidad del ventilador para que ésta sea constante.

Modificar los valores de giro

Puede verificar los valores de giro actuales del siguiente modo:

```
$ cat
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/trip_point_{0,1,2}_temp
# results
60000
70000
80000
60000
70000
80000
60000
```

```
70000
80000
60000
70000
80000
```

Sí, son otros valores de giro llamados 3, 4, 5. Pero puedes ignorarlos ya que no los usamos. Lo mismo con thermal_zone4. Como podemos ver, cada valor objetivo en cada zona térmica tiene el mismo valor 60000, 70000, 80000. Eso significa que en cada valor objetivo se activa a 60 ° C, 70 ° C, 80 ° C. Cada valor de giro es editable escribiendo un valor personalizado para cada archivo de valor objetivo. Por ejemplo, si desea configurar el valor objetivo 1 para que se active a los 30 ° C, puedes escribir un valor para éste.

```
$ echo 30000 | sudo tee
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/trip_point_0_temp
$ cat
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/trip_point_0_temp
# results
30000
30000
30000
30000
```

A continuación, el ventilador empezará a girar a 30 °C. Si deseas hacerlo automáticamente, tendrás que escribir algo de código en el archivo /etc/rc.local. Copia y pega el siguiente código:

```
# Target temperature: 30°C, 50°C, 70°C
TRIP_POINT_0=30000
TRIP_POINT_1=50000
TRIP_POINT_2=70000

echo $TRIP_POINT_0 >
/sys/devices/virtual/thermal/thermal_zone0/tri
p_point_0_temp
echo $TRIP_POINT_0 >
/sys/devices/virtual/thermal/thermal_zone1/tri
p_point_0_temp
echo $TRIP_POINT_0 >
/sys/devices/virtual/thermal/thermal_zone2/tri
p_point_0_temp
echo $TRIP_POINT_0 >
/sys/devices/virtual/thermal/thermal_zone3/tri
p_point_0_temp

echo $TRIP_POINT_1 >
/sys/devices/virtual/thermal/thermal_zone0/tri
p_point_1_temp
echo $TRIP_POINT_1 >
/sys/devices/virtual/thermal/thermal_zone1/tri
p_point_1_temp
echo $TRIP_POINT_1 >
/sys/devices/virtual/thermal/thermal_zone2/tri
p_point_1_temp
echo $TRIP_POINT_1 >
/sys/devices/virtual/thermal/thermal_zone3/tri
p_point_1_temp

echo $TRIP_POINT_2 >
/sys/devices/virtual/thermal/thermal_zone0/tri
p_point_2_temp
echo $TRIP_POINT_2 >
/sys/devices/virtual/thermal/thermal_zone1/tri
p_point_2_temp
echo $TRIP_POINT_2 >
/sys/devices/virtual/thermal/thermal_zone2/tri
p_point_2_temp
echo $TRIP_POINT_2 >
/sys/devices/virtual/thermal/thermal_zone3/tri
p_point_2_temp
```

Reinicia y comprueba que los cambios se hayan aplicado.

Modificar la velocidad del ventilador

Puedes comprobar la escala de la velocidad actual del ventilador con el siguiente comando:

```
$ cat /sys/devices/platform/pwm-
fan/hwmon/hwmon0/fan_speed
# results
0 120 180 240
```

Puedes ajustar estos valores escribiendo un valor fijo en el archivo. Si quieres que tu ventilador sea más agresivo, puede usar el siguiente comando:

```
$ echo "0 204 220 240" | sudo tee
/sys/devices/platform/pwm-
fan/hwmon/hwmon0/fan_speed
# results
0 204 220 240
```

Esto hace que un ventilador gire al 80% (204 == 80 * 255 * 0.01) cuando la temperatura alcance el valor objetivo 0. Cuando la velocidad del ventilador se

ajuste nuevamente, aparecerá un mensaje del kernel, puedes averiguarlo usando el comando dmesg:

```
$ dmesg
# results
...
[ 1998.019631] hwmon hwmon0: fan_speeds :
set_fan_speed [0 204 220 240]
```

Si deseas hacerlo automáticamente, copia y pega las siguientes líneas en el archivo /etc/rc.local:

```
# Target fan speed (PWM): 0, 204, 220, 240
echo "0 204 220 240" >
/sys/devices/platform/pwm-
fan/hwmon/hwmon0/fan_speed
```

Reinicie y comprueba si los cambios se han aplicado.

Emular la temperatura

No tienes por qué poner al límite tu ODROID para probar la nueva configuración. La configuración del ventilador se puede verificar con estos archivos.

```
$ ls -l
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/emul_temp
# results
--w------- 1 root root 4096 Apr 11 01:55
/sys/devices/virtual/thermal/thermal_zone0/emu
l_temp
--w------- 1 root root 4096 Apr 11 02:05
/sys/devices/virtual/thermal/thermal_zone1/emu
l_temp
--w------- 1 root root 4096 Apr 11 02:05
/sys/devices/virtual/thermal/thermal_zone2/emu
l_temp
--w------- 1 root root 4096 Apr 11 02:05
/sys/devices/virtual/thermal/thermal_zone3/emu
l_temp
```

Estos archivos modificables nos permiten simular cualquier valor de temperatura para fingir la temperatura real de la placa y finalmente hacer que el ventilador responda según la configuración. Si quiere configurarla a 85 °C, simplemente escríbelo.

```
$ echo 85000 | sudo tee
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/emul_temp
# results
85000
```

Verifica si los cambios han tenido efecto:

```
$ cat
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/temp
# results
85000
85000
85000
85000
```

Si quieres volver a la normalidad, escribe 0:

```
$ echo 0 | sudo tee
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/emul_temp
# results
0
$ cat
/sys/devices/virtual/thermal/thermal_zone{0,1,
2,3}/temp
# results
30000
30000
30000
29000
```

Esto es muy útil para que compruebes la nueva velocidad del ventilador y la configuración del valor objetivo que acaba de fijar.

Modo totalmente manual de controlar la velocidad del ventilador

Este es el método más sistemático para ajustar la velocidad del ventilador de forma manual:

```
# Set fan to manual mode
$ echo 0 | sudo tee /sys/devices/platform/pwm-
fan/hwmon/hwmon0/automatic

# Set speed to 100%
$ echo 255 | sudo tee
/sys/devices/platform/pwm-
fan/hwmon/hwmon0/pwm1
```

El ventilador ignora los archivos con las escalas (valores objetivos y velocidad del ventilador) y se ejecuta constantemente a la misma velocidad, también puedes hacerlo automáticamente. Edita el archivo /etc/rc.local y reinicia para comprobar si se han aplicado los cambios. El siguiente ejemplo hace

que el ventilador siempre se ejecute a máxima velocidad:

```
# Fix fan speed
echo 0 | sudo tee /sys/devices/platform/pwm-fan/hwmon/hwmon0/automatic
echo 255 | sudo tee /sys/devices/platform/pwm-fan/hwmon/hwmon0/pwm1
```

Además, puedes escribir una aplicación usando el ventilador.

Ejemplos de scripts para controlar el ventilador

Hay algunos ejemplos de script muy buenos que deberías consultar. **https://forum.odroid.com/viewtopic.php?f=77&t=30743**

https://forum.odroid.com/viewtopic.php?f=146&t=30745

Referencias

https://forum.odroid.com/viewtopic.php?f=52&t=16308
https://forum.odroid.com/viewtopic.php?f=99&t=30675

El texto original lo puedes encontrar en la página wiki de ODROID y así poder controlar manualmente el ventilador del ODROID-XU4, **https://wiki.odroid.com/odroid-xu4/application_note/manually_control_the_fan#fully_manual_way_to_control_the_fan_speed**.

Cliente Minecraft en ODROID

May 1, 2018 By Sebastien Chevalier Juegos, Linux

Entorno de Minecraft

¡Ahora se puede jugar al Minecraft en el ODROID! La instalación es bastante sencilla, gracias a las habilidades de empaquetado de Tobias también conocido como @meveric. Tras instalar su repositorio, escribe el siguiente comando:

```
$ sudo apt-get install minecraft-odroid
```

Minecraft se instalará con un par de dependencias y estará listo para ejecutarse. Viene empaquetado con el lanzador por defecto, para que puedas reproducir la demo, o puedes iniciar sesión con tu cuenta para jugar.

Rendimiento

Una cosa que debe saber es que actualmente no es realmente compatible con los mipmaps, de modo que el rendimiento será muy bajo a menos que configures los mipmaps en “ninguno”. Una vez que el juego se haya iniciado, ve al menú de Opciones, selecciona Video, luego elige Mipmap Levels: OFF.

Figura 1 – Configuración del video

Después de esto, el resto de configuraciones son bastante estándar y tienen el efecto deseado. Recomiendo reducir la "Render Distance" (5 están bien, aunque es posible que desees bajarlo más para obtener más FPS), elije "Graphics: Fast" (para que las hojas del árbol no sean transparentes), y ajusta el Smooth Lightning en OFF para velocidad máxima o Minimum para algunas sombras suaves (pero es más lento). Además, el Max Framerate debe estar alrededor de tus FPS actual (30 fps es bien para que el juego vaya fluido). Con estos parámetros, logré conseguir entre 12 y 15 FPS en Full HD. Puedes usar "F3" durante el juego para que se muestren algunas estadísticas, incluido los FPS, pero ten en cuenta que la pantalla F3 utiliza algunos FPS, unos 3 o 4 como mínimo.

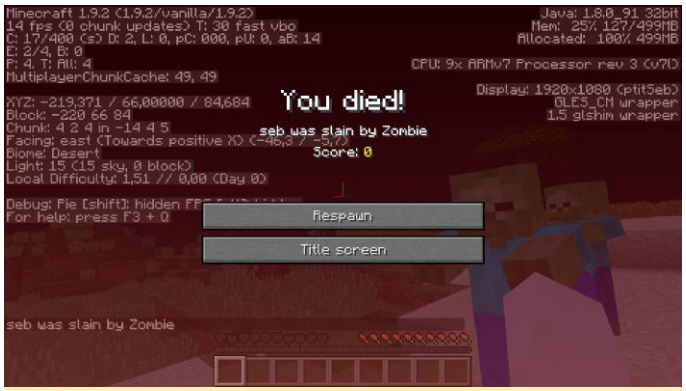


Figura 2 – Pantalla de la muerte

Mejorar el rendimiento

Si quieres que tu Minecraft funcione más rápido, puedes utilizar OptiFine, que es un mod que te permite modificar muchos ajustes de Minecraft, así como el método de renderizado para logara que el juego vaya más fluido. Primero debe iniciar Minecraft y lanzar un juego, para que la versión actual de Minecraft se registre y descargue. Luego, dirígete al sitio web de OptiFine en http://bit.ly/1jOG2Di y descarga la versión para tu versión de Minecraft (cuando escribí este artículo, era la 1.9.4). Recibirá un archivo .jar que se puede iniciar y que se instalará automáticamente. Para iniciarlo, simplemente haga doble clic o, usando un terminal, escribe el siguiente comando:

```
$ java -jar OptiFine_1.9.4_HD_U_B4.jar
```

A continuación, verás un menú que te pregunta qué deseas hacer. OptiFine primero detecta automáticamente la carpeta local de Minecraft y después de un rato, debe instalarse sin problemas.

Figuras 3 – Instalación del mod Optifine

Vuelve a iniciar Minecraft, y notará que el perfil ahora se llama OptiFine. Una vez en el juego, notarás que los trozos se cargan más rápido. Hay muchos más parámetros para probar en la pantalla Opciones, tal y como se muestra en la Figura 5.

Figura 4 – Configuración de video Optifine

Sin tocar nada, OptiFine puede darte unos cuantos FPS más (En mi caso, obtuve 4 FPS más en mi ODROID), pero depende en gran medida de la configuración real. Estimo que puede llegar a mejorar el rendimiento entre un 25% y un 50%.

Cómo funciona

La primera pregunta que podrías hacerse es por qué no ha estado disponible antes Minecraft para ODROID. Después de todo, es un juego de Java, de modo que debería ejecutarse tal cual. Sin embargo, los programas de Java no dependen de la CPU, y tampoco dependen del sistema, ya que se trata de una Máquina Virtual. Por lo tanto, un programa Java que se ejecuta en un sistema x86/Windows también puede ejecutarse en un sistema x86/Linux o ARM/Linux. En algunas ocasiones, Java no es suficiente para hacer un programa, y necesitas interactuar con alguna librería nativa para hacer cosas más avanzadas. Se trata de JNI (Java Native Interface), un mecanismo que permite que un programa Java llame directamente a una librería nativa. Por ejemplo, necesitas esto para usar un sonido OpenAL o gráficos OpenGL, y eso es lo que hace Minecraft: utiliza una librería Java llamada "lwjgl" (Light Weight Java GL) para acceder a OpenGL para el renderizado. Para usar esta librería, Minecraft la descarga directamente desde su servidor, junto con el resto de librerías y recursos necesarios, cuando lanzas un juego por primera vez, y comprueba que tienes todo correctamente en su lugar cada vez que lo ejecutas. El problema es que Minecraft no es compatible con ARM. Ni siquiera conoce esta arquitectura. Entonces, cuando descarga su versión de lwjgl, se obtiene una versión para una CPU x86, que simplemente no funciona porque no es la correcta. Para solucionar esto, se ha creado un lanzador especial que intercepta todas las llamadas a Java, analiza los comandos y reemplaza el enlace por la versión x86 con el que está instalada en el sistema. Es un poco bruto, pero funciona permitiendo que Minecraft se ejecute. Aunque se inicia, no llega muy lejos, ya que necesita OpenGL, y ODROID solo proporciona GLES. De modo que, debemos usar glshim para traducir todas las llamadas de OpenGL a GLES. Glshim solo proporciona OpenGL 1.5 hasta ahora, así que Minecraft nos advierte que actualicemos nuestros drivers, indicándonos una advertencia de que ya no es compatible con OpenGL 1.x, y que necesitarmos OpenGL 2.0. Por cierto, glshim contiene algunos hacks especiales que fueron creados específicamente para Minecraft. La primera versión de Minecraft sobre máquina ARM fue en OpenPandora, hace 2 años. Y al principio, se veía tal como se muestra en la Figura 5.

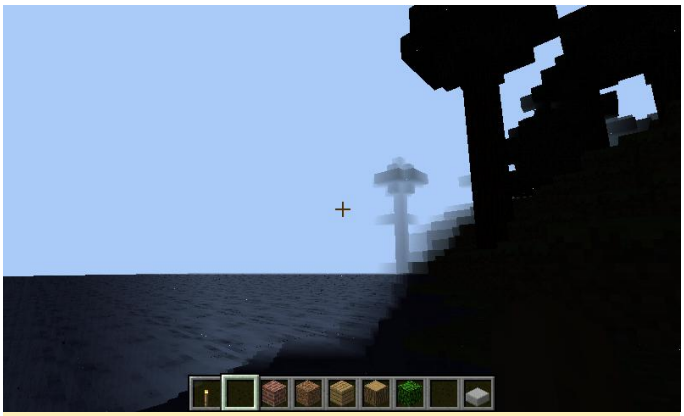


Figura 5 – Versión inicial de Minecraft en OpenPandora

Como puede ver, ¡No era muy colorido! Tras algunas depuraciones, finalmente codifiqué un hack en glshim para compensar la forma en que Minecraft monta su iluminación. Utiliza multitextura, donde la primera textura es del color del bloque, y la segunda textura es el mapa de luz, el cual es un método muy común de iluminación. Sin embargo, lo que Minecraft hace es renderizar las texturas de forma que cada bloque se tiene en cuenta para la iluminación uniforme, de modo que no tienes bloques a medio iluminar. De modo que, cuando se emite el comando de trazado para un bloque/cubo, todas las coordenadas de vértice son proporcionadas a openGL, junto con las coordenadas de texturas para la primera textura, con solo una coordenada de textura para el mapa de luz. Y este supuesto, que es técnicamente correcto de acuerdo con las especificaciones de OpenGL, no es gestionado por Glshim. Se corrigió comprobando si solo había una coordenada para una textura. En este caso, las coordenadas están duplicadas para todos los vértices, lo que facilita su manejo en glshim. Si tiene curiosidad por los detalles técnicos, busca la función "glshim_glEnd" en el archivo "gl.c" (**http://bit.ly/24WP30W**).

¿Y ahora qué?

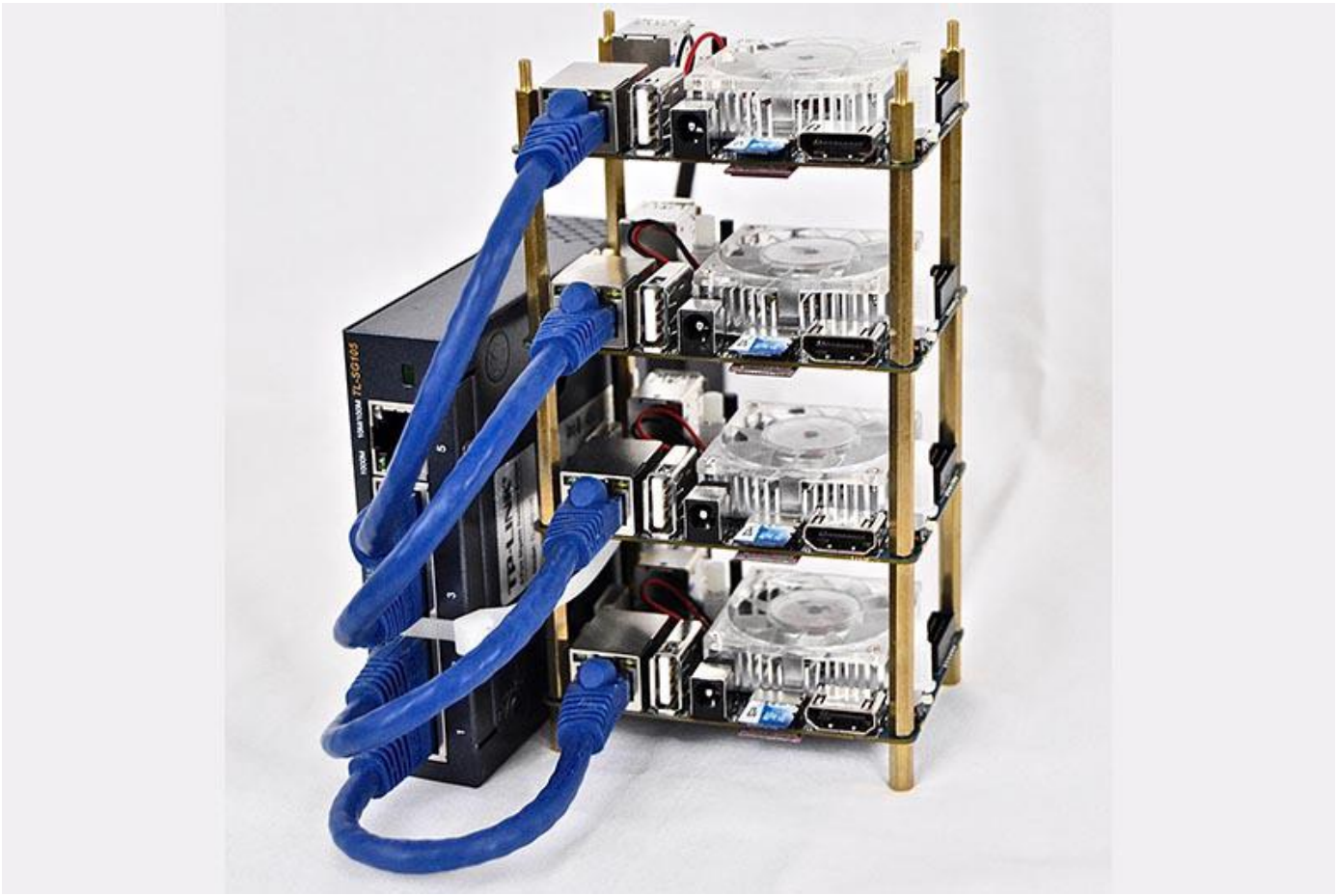
Después de crear el lanzador personalizado y modificar Glshim, Minecraft funciona bastante bien. Aun así, las cosas siempre se pueden mejorar. Todavía hay tres áreas principales en la que se puede trabajar en la aplicación glshim:

- Mejorar la gestión de la configuración de Mipmap
- Aumentar la velocidad usando el modo Batch de glshim
- Tener un glshim funcionando en GLES2

La configuración de mipmap es un poco confusa, y tengo que entender lo que realmente hacen los niveles de Mipmap, lo cual no es fácil con un software de código cerrado. El modo Batch puede ser bastante efectivo a veces, como con Xash3D o Emilia Pinball, por ejemplo, pero en otras ocasiones es totalmente ineficaz. Incluso puede llegar a romper el motor de renderizado, como es el caso de Minecraft. Se necesita mucho trabajo para estabilizar esta característica. Tener un Glshim usando GLES2 y proponer una versión OpenGL 2.x son objetivos a largo plazo para glshim, aunque será necesario tarde o temprano, ya que cada vez más el software está dejando de tener soporte para la tubería fija, que es una función OpenGL 1.x. a favor de usar sombreado en su lugar.

Clúster ODROID-XU4

May 1, 2018 By Michael Kamprath ODROID-XU4, Tutoriales



En los últimos años, los temas de big data y ciencia de los datos se han convertido en una corriente dominante en innumerables industrias. Las empresas de alta tecnología de Silicon Valley ya no son las únicas proveedoras de temas como Hadoop, regresión logística y aprendizaje de máquinas. Estar familiarizado con las tecnologías del big data se está convirtiendo en un requisito cada vez más necesario en los trabajos relacionados con las tecnologías en cual parte del mundo. Desafortunadamente, conseguir experiencia real y práctica con las tecnologías de big data generalmente significa tener acceso a un costoso clúster de ordenadores que te permitan ejecutar tus consultas. Sin embargo, la reciente revolución de los ordenadores de placa reducida ha hecho que la verdadera computación distribuida sea accesible para un uso personal y a nivel de formacion para realizar tareas como estas y otras.

He trabajado en el ámbito del Big Data durante ocho años. Aunque he tenido acceso a un clúster para procesar petabytes de datos durante un tiempo, nunca tuve la oportunidad de diseñar y crear un clúster propio. Decidí crear un pequeño clúster principalmente para familiarizarme con las operaciones y la configuración básica del software de big data y un clúster subyacente. Mi objetivo teniendo en cuenta el coste era montar un clúster de cuatro nodos por menos de 600$. También quería crear un clúster lo suficientemente potente como para poder procesar datos con tamaños de 10 gigabytes.

Los elementos clave a considerar a la hora de seleccionar la tecnología del clúster son el almacenamiento de datos, la E/S, el rendimiento de la red, los núcleos de la CPU y la memoria RAM disponible. Afortunadamente, Hardkernel ha fabricado un ordenador de placa reducida que destaca en estas especificaciones: el **ODROID-XU4**. Con un procesador Samsung Exynos 5422 8-core de

2GHz, ethernet Gigabit integrado, múltiples puertos USB 3.0, 2 GB de RAM y disponibilidad de almacenamiento de datos de alto rendimiento con unidades eMMC y tarjetas microSD UHS-1, el XU4 es un ordenador de placa reducida formidable con un coste relativamente bajo.

Con el hardware de los nodos ya elegido, nuestra primera tarea es diseñar la topología del clúster o, dicho de otro modo, cómo se conectarán los nodos entre sí. Hay varias cosas que influyen en esto, especialmente el tipo de computación distribuida que tengas pensado hacer. Los modelos de computación distribuida se pueden categorizar básicamente como grandes CPU o big data. Para este proyecto, nos centrando en el big data, específicamente en el análisis de datos. El modelo de big data más común actualmente en uso para el análisis de datos es mapreduce, que está implementado por Apache Hadoop y Apache Spark, que son tecnologías de almacenamiento de datos muy populares utilizadas por muchas de las grandes compañías de tecnología.

En la mayoría de los clústeres MapReduce a escala comercial, la topología de clúster general tiene varios de nodos perimetrales en los que el usuario inicia sesión para usar el clúster, uno o más nodos principales que son usados por el clúster para coordinar la actividad informática y el almacenamiento de datos, y varios nodos esclavos que se utilizan para tareas de cálculo o almacenamiento de datos, o ambos (consulte la Figura 1). Piensa en ello como si dividieran un gran proyecto entre varias personas para mejorar el rendimiento de todos: un director envía la solicitud del proyecto (el nodo central) con varios gerentes coordinando qué hacer (los nodos principales) y los empleados que realizan esas tareas y combinan su trabajo (los nodos esclavos) en una solución final para el director.

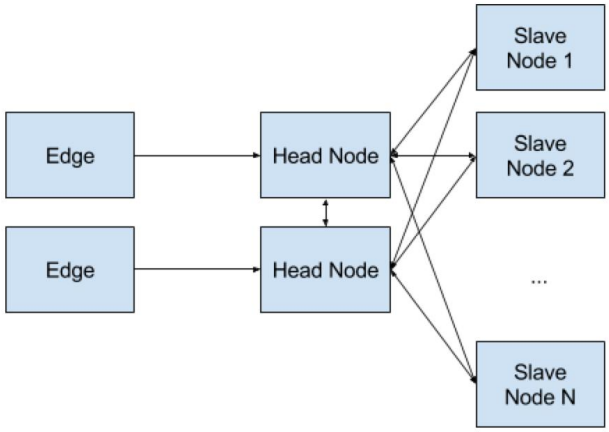


Figura 1: Típica topología de clúster de MapReduce

Para nuestro clúster XU4, vamos a combinar el concepto de nodo perimetral y de nodo principal en un nodo maestro, y luego vinculamos los esclavos al nodo maestro. Esto significa que el nodo maestro será el nodo en el que los usuarios inicien sesión para usar el clúster y el nodo que coordina a los esclavos. Esto también implica que la comunicación de nodo a nodo del clúster ocurrirá dentro de una red privada, mientras que el nodo maestro necesitará tener conexión a una red externa. El diseño de red del XU4 para un clúster de cuatro nodos debería parecerse al que se muestra en la Figura 2.

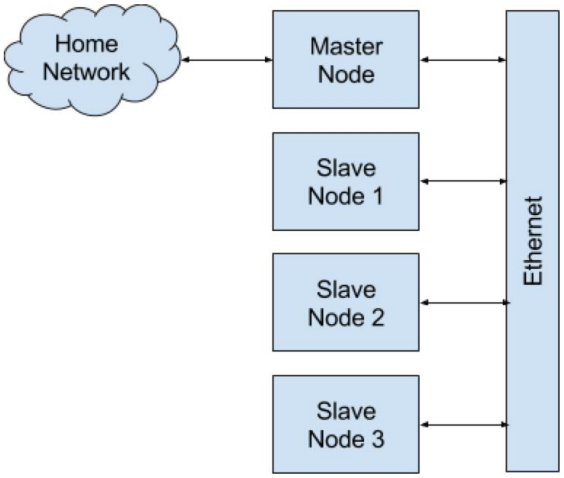


Figura 2 – Topología del clúster ODROID-XU4

Esta topología requiere que el nodo maestro pueda conectarse a dos redes independientes. Sin embargo, el XU4 solo tiene un puerto ethernet. Debemos añadir una segunda conexión de red al nodo maestro con un dongle ethernet USB 3.0

El XU4 ofrece dos opciones de almacenamiento: una unidad eMMC y una tarjeta microSD. Ambos tienen

pros y contras. La unidad eMMC es extremadamente rápida, mientras que el coste de la tarjeta microSD por gigabyte es más asequible, pero es más lenta que la unidad eMMC. La buena noticia es que el rendimiento de lectura y escritura de una tarjeta microSD UHS-1 está prácticamente a la par de los discos duros mecánicos, que normalmente se utilizan en grandes clusters comerciales. Esto significa que la tarjeta microSD representa una buena opción para el almacenamiento masivo de datos. Sin embargo, la velocidad de la unidad eMMC es muy atractiva para usarla como unidad de inicio desde la cual se ejecutara el software. Dado que, cada nodo de nuestro clúster tendrá una unidad eMMC para el arranque y una tarjeta microSD para el almacenamiento masivo de datos. Recomiendo utilizar al menos una unidad eMMC de 16 GB para el nodo maestro, ya que será donde, como usuario, trabajarás, mientras que te puedes ahorrar dinero obteniendo unidades eMMC de 8 GB más baratas para los nodos esclavos. Para el almacenamiento de datos, busca algunas tarjetas microSD rápidas de 64 GB o más para cada uno de los nodos.

El conjunto final de materiales necesarios para el proyecto incluye un pequeño switch Ethernet para la red interna del clúster, varios cables Ethernet de 6 pulgadas y separadores de PCB para apilar los XU4. También cogí un UART serial para el XU4 en caso de que tuviera que conectarme a un dispositivo directamente para solucionar algún problema, aunque nunca lo necesité. Un elemento que no llegue a comprar y que sería bueno tener a posteriori es una fuente de alimentación que pueda proporcionar 5V de potencia a 4 amperios simultáneamente a todos los nodos, en lugar de tener un conjunto desordenado e ineficiente de adaptadores de pared conectados a una regleta de enchufes. Esta será una futura mejora del proyecto.

Una vez que todos los materiales hayan sido recopilados y se monte el clúster, nuestra primera tarea es configurar el sistema operativo y la red en todos los nodos. Yo opté por la distribución Ubuntu 15.10 actual de ODROID para XU4. Grabé este sistema operativo en cada uno de los módulos eMMC, y luego uno por uno fui arrancando cada dispositivo sin la tarjeta microSD adicional (que se utilizará para el almacenamiento posterior) mientras los fui conectado directamente a mi red doméstica. Esto me permitió conectarme por SSH directamente a los dispositivos después del primer arranque. Una vez que el dispositivo arrancase, localice la dirección IP que es asignada a cada XU4 por el servidor DHCP de mi casa e inicié sesión. La cuenta de usuario por defecto es “odroid” y la contraseña también “odroid”. Después de conectarme, instalé ODROID Utility para configurar mejor el sistema operativo. Esto se puede hacer descargando directamente la utilidad desde Github:

```
$ sudo -s
 $ wget -O /usr/local/bin/odroid-utility.sh
https://raw.githubusercontent.com/
mdrjr/odroid-utility/master/odroid-utility.sh
 $ chmod +x /usr/local/bin/odroid-utility.sh
 $ odroid-utility.sh
```

Las tres tareas que realiza ODROID Utility son nombrar el nodo, deshabilitar Xorg y maximizar el tamaño de la partición de la unidad eMMC. Llamé maestro al nodo maestro, y a los otros tres: esclavo1, esclavo2 y esclavo3.

El nodo maestro debe configurarse adicionalmente para usar el dongle ethernet USB 3.0 ya que se conectará a una red externa. Para configurar el nodo maestro y así poder tener conexión a Internet desde la red conectada al dongle USB, deberás crear un archivo llamado “eth1” en el directorio /etc/network/interfaces.d/ con el siguiente contenido (suponiendo que esa red tiene un servidor DHCP):

```
auto eth1
 iface eth1 inet dhcp
```

Del mismo modo, para que el ethernet integrado sea usado para la red interna del clúster, necesitas crear un archivo llamado eth0 en la misma carpeta indicando una dirección IP estática:

```
auto eth0
 iface eth0 inet static
 address 10.10.10.1
 netmask 255.255.255.0
 network 10.10.10.0
 broadcast 10.10.10.255
```

Es necesario configurar un servidor DHCP en el nodo maestro para proporcionar una dirección IP a los nodos esclavos de la red interna, y el nodo maestro deberá proporcionar servicios NAT entre las redes externa e interna. Además, en todos los nodos es necesario editar el archivo /etc/hosts para permitir el direccionamiento nemotécnico de los nodos por su nombre sin necesidad de un servicio DNS. Puedes encontrar instrucciones detalladas para realizar estas tareas en mi blog en http://bit.ly/2aJdAmi.

Una vez que los nodos estén configurados según el diseño de red deseado, los nodos se pueden cerrar y desconectar de la red doméstica. El Ethernet integrado de los nodos debe estar conectada al switch Ethernet de la red interna, y la red doméstica debe conectarse al dongle Ethernet USB 3 del nodo principal.

Antes de reiniciar cada nodo, formatea las tarjetas microSD con un sistema de archivos ext4 y conecta una a cada nodo. Arranca todos los dispositivos. Deberías conectarte por SSH al nodo maestro, y desde allí conectarte por SSH a cada esclavo. Tu ultima tarea de configuración es editar el archivo /etc/fstab en cada dispositivo de forma que la tarjeta microSD esté montada en un punto de montaje /data. Para hacer esto, necesita encontrar el UUID del volumen de la tarjeta microSD con el comando blkid, luego añade una línea al archivo /etc/fstab similar a que aparece a continuación:

```
UUID=c1f7210a-293a-423e-9bde-1eba3bcc9c34
/data ext4 defaults 0 0
```

Reemplazar el UUID de tu tarjeta microSD por el que aparece arriba, que también se detalla en mi blog. Una vez completados estos pasos, tendrá un clúster completamente configurado que está listo para albergar un software de big data como Hadoop. Instalar Hadoop es un proceso bastante complicado, y lo cubriré en otro artículo. Por ahora, hemos montado con éxito un clúster XU4 que puede utilizarse para cualquier tipo de procesamiento de datos complejo. Puedes encontrar más información sobre este clúster ODROID-XU4 en http://bit.ly/2aJdAmi.

Conceptos Básicos de BASH: Introducción a BASH

May 1, 2018 By Erik Koennecke ODROID-HC1, Tutoriales, ODROID-HC2

Esta guía es una introducción amigable para los principiantes del shell BASH (**https://www.gnu.org/software/bash/**), terminal y conceptos generales de Linux, como es la organización de archivos. Lo más probable es que, si aún no te estás acercando a la edad de jubilación, los ordenadores que has ido utilizado siempre habrán tenido una interfaz gráfica de usuario (GUI). Cuando inicias tu SBC ODROID, eres recibido con un agradable escritorio Ubuntu MATE no muy diferente al de Windows 10 u OS X. Por el contrario, un shell o un intérprete de línea de comandos como BASH parece más bien una reliquia de hace 50 años, entonces ¿por qué deberías abandonar la comodidad de contar con una GUI?

Hay varias respuestas para esto:

- Eres un poco vago, como yo. Es mejor pensar unos minutos en el problema y automatizar una tarea tediosa y repetitiva en lugar de hacerla tú mismo.
- Su SBC no tiene salida de video, como el HC1 o el **HC2**, o tienes la placa en otro lugar a la que te conectas solo a través de la red. Las GUI por la red consume ancho de banda y resulta un poco molesto trabajar; el retraso de todo lo que haces debido a la latencia añadida terminará desesperándote con el tiempo. Las soluciones de línea de comandos generalmente son más receptivas y más fáciles de usar en conexiones remotas.
- Deseas entender mejor el funcionamiento del sistema y tener un mayor control sobre él. Esto también se logra mejor con la línea de comando en el shell. Se trata de una capa inferior a la GUI. Tu pequeño ODROID es como una caja negra, tienes un control exhaustivo y puedes hacer más cosas de las que puede hacer solo con la GUI.

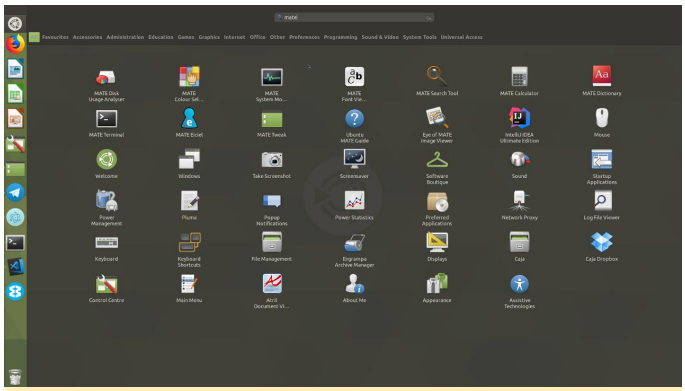


Figura 1 – El escritorio Ubuntu Mate es una moderna interfaz gráfica de usuario (GUI)

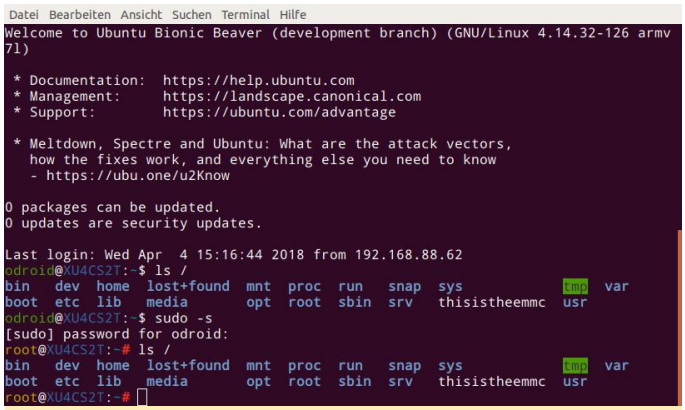


Figura 2 – La línea de comandos BASH parece más bien una reliquia

Comencemos ahora por terminal con el shell BASH por defecto y veamos cómo puede ayudarnos con todo esto. En Ubuntu MATE, simplemente presiona CTRL-TAB-T para abrir el terminal o shell.

Ventana de terminal

Al usar el método abreviado, abrimos un terminal de caracteres de 80×24 (usando el perfil por defecto, que puede ser cambiado, copiado y editado) con la línea de comando en la parte superior. El prompt que aparece se compone de tu nombre de usuario (generalmente ODROID), el nombre de la máquina, dos puntos, la ruta o el directorio de trabajo, y un signo $ que identifica que eres un usuario normal. Con el usuario root se muestra una # en su lugar, tal y como puedes ver en la Figura 2. Más adelante, personalizaremos esto a nuestro gusto.

Puesto que empiezas en tu directorio de inicio, éste se muestra con el signo ~ para abreviarlo. El comando ls te muestra el contenido de tu directorio de trabajo, similar a abrir el Explorador de archivos.

¿Qué es un Shell?

Este terminal ejecuta el shell BASH. Un shell es un intérprete de línea de comandos que se ejecuta en una ventana de texto, el terminal. El estándar para Linux es BASH. BASH también puede leer y ejecutar comandos desde un script, llamado script shell.

Hasta ahora, no es muy espectacular. Sin embargo, si usas ls -l o ll para abreviar, recibes más información de la que te proporciona el Explorador de archivos sin tener que ir al menú de Preferencias para cambiar la configuración. Pero espera, aún hay más. Si deseas ver un buen árbol de lo que tienes en tu directorio de inicio, intente ejecutar el comando tree. Si no lo tienes instalado, instálalo ejecutando: $ sudo apt update && apt install tree Dependiendo de la cantidad de archivos que tengas, pueden aparecer más o menos resultados. Limita el resultado solo a los directorios con tree -d, y si tiene muchos niveles que no te interesan en este momento, puede limitar a, por ejemplo, 2 niveles usando el árbol -d -L 2. Cuando usa un comando, puede obtener un resumen abreviado con:

```
$ <command></command> --help
```

or usar:

```
<command></command> $ man
```

para una página del manual completa de ello.

Sistema de archivos

Con el comando ls /, obtienes el contenido del directorio raíz. Todo lo demás a lo que puedes acceder está vinculado a una rama de su árbol. Para una vista general, usa el siguiente comando:

```
$ tree -d -L 1 /
```

Los directorios que ves siguen un estándar, el Estándar de Jerarquía del Sistema de Archivos (FHS). Los que nos interesan son principalmente:

- /bin – binarios de comandos esenciales que deben estar disponibles en modo de usuario único para todos los usuarios, por ejemplo, cat, ls, cp.
- /boot – archivos del cargador de arranque como el disco RAM initrd, el kernel y los blobs del árbol del dispositivo ARM para la placa. Estos también se encuentran en /media/boot, el lugar donde se monta

la partición FAT32 desde la cual arranca la SBC ODROID.

- /dev – los archivos de dispositivo para todo lo que se conecta al ODROID. /dev/mmcblk0 es la eMMC, /dev/mmcblk0p1 es la primera partición en la eMMC, la partición de arranque FAT32 para el procesador ARM. /dev/mmcblk0p2 es la partición del sistema que es tu partición raíz si arranca desde la eMMC. En caso de utilizar la tarjeta SD, es mmcblk1 en lugar de mmcblk0.
- /etc – Este es el lugar de todos los archivos de configuración del sistema.
- /home – Los directorios de inicio para los usuarios. Con la configuración estándar, tienes /home/ODROID para el usuario de ODROID, el método abreviado para el directorio de inicio de cada usuario es ~.
- /lib – librerías para los programas en /bin y /sbin.
- /media – Punto de montaje para el almacenamiento extraíble como memorias USB. La partición de arranque para el procesador ARM también se monta aquí.
- /mnt – Sistemas de archivos montados temporalmente.
- /opt – Software opcional. Cosas como ffmpeg, Google Chrome, Skype, TeamViewer. Si desea ver lo que hay aquí, tree -d /opt o simplemente ls opt te proporciona una vista general.
- /proc – Sistema de archivos virtual que proporciona información del proceso y del kernel, habitado por el sistema.
- /root – Directorio de inicio para el usuario root, el superusuario.
- /run – Información sobre el sistema desde el último arranque.
- /sbin – Binarios esenciales del sistema como mount, iw, fdisk, mkfs.
- /srv – Datos servidos por este sistema. Si tiene uno de los nuevos HC1 o HC2 o generalmente utiliza tu ODROID como servidor de archivos, el lugar recomendado para montar tu disco duro sería
- /srv/samba o /srv/ftp o /srv/nfs.
- /sys – Información sobre dispositivos, drivers y algunos servicios del kernel. Desde aquí se controlan bastante los SBCs. Para controlar el ventilador en un XU4, deberías usar:

```
$ echo 0 > /sys/devices/platform/pwm-fan/hwmon/hwmon0/automatic
```

para apagarlo y

```
 $ echo 255 > /sys/devices/platform/pwm-fan/hwmon/hwmon0/automatic
```

para encenderlo de nuevo

- /tmp – archivos temporales, a menudo no se conservan al reiniciar el sistema.
- /usr – Contiene la mayoría de las utilidades y aplicaciones del (multi-)usuario. Tiene su propia jerarquía con /usr/bin, /usr/lib, /usr/local, /usr/sbin y así sucesivamente.
- /var – Archivos de variables como registros logs y archivos spool.

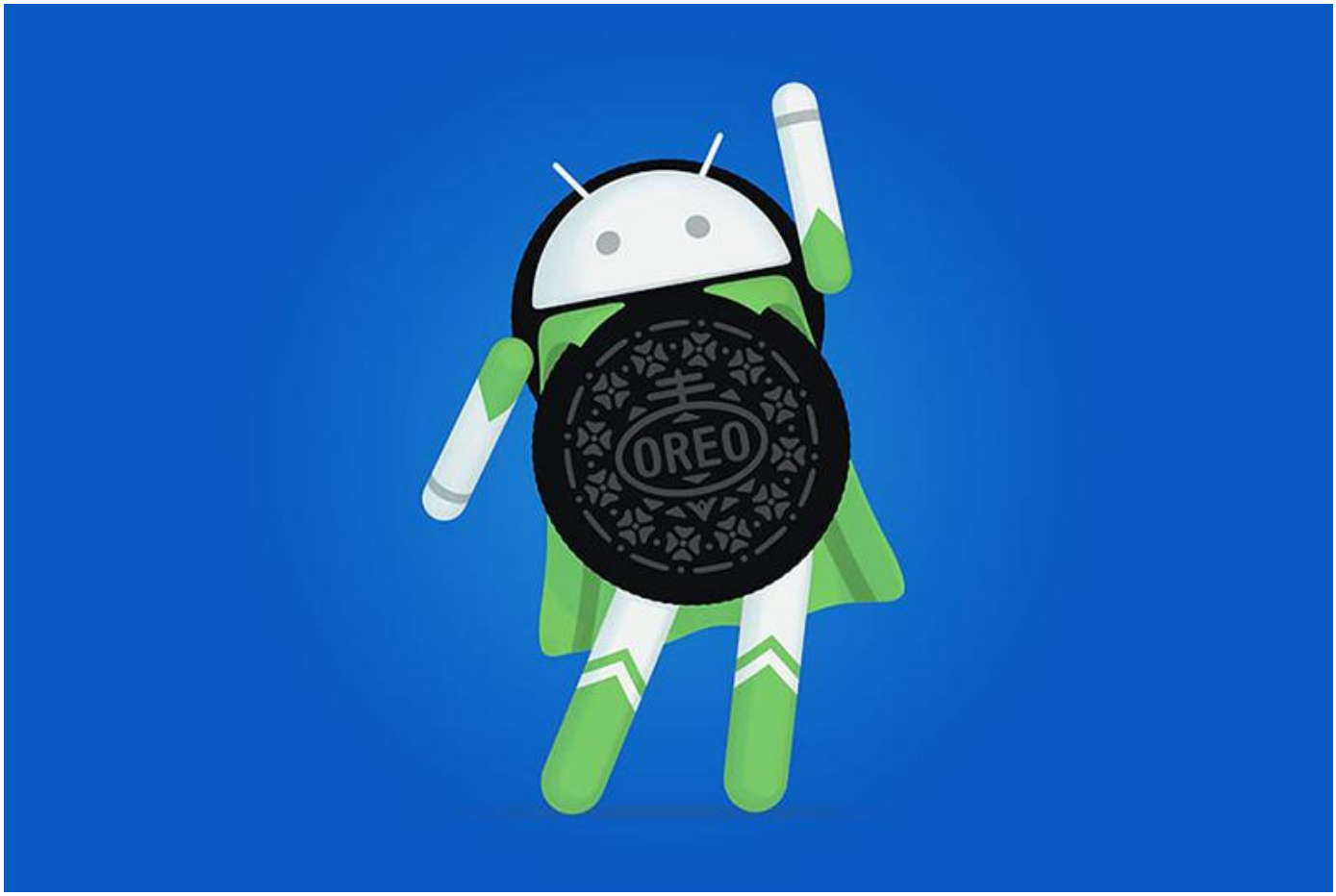
Comandos útiles

Ahora que conoces el diseño de tu sistema, ¿Qué otros comandos son útiles en terminal además de ls y tree? En las siguientes entregas, trataremos:

- Los comandos más básicos, uso, aplicación para los SBCs ODROID
- Qué sucede durante el arranque y el inicio de sesión con respecto a BASH
- Personalizar el prompt BASH
- Breve introducción al scripting, incluyendo variables, pruebas, bucles
- Simples líneas que son muy útiles en la línea de comandos

Android Oreo: Hazte con la Última Versión de Android para tu ODROID-XU4

May 1, 2018 By Justin Lee Android, ODROID-XU4



El usuario de los foros ODROID voodik ha estado exportando Android 8.1 (basado en LineageOS 15.1) para ODROID-XU4 desde el pasado mes de octubre. Recientemente ha lanzado la primera versión alfa con fines de depuración por parte de la comunidad.

Qué funciona

- Driver GPU acelerado por hardware para el renderización 3D
- Driver VPG acelerado por hardware para reproducción de video
- Ethernet
- Receptor de GPS
- Tarjeta de sonido USB
- WiFi (incluido el modo AP)
- Modo fuente de Bluetooth
- Barra de navegación
- El resultado de la prueba de rendimiento Antutu es excelente

Problemas conocidos

- Modo Sink del Bluetooth
- Algunos problemas con Play Store: al descargar una aplicación, puede quedarse bloqueado con el mensaje de “Descarga pendiente”. Simplemente cancela Play Store en aplicaciones recientes y ábrelo de nuevo
- No todas las características específicas de ODROID han sido exportados, como usar la rueda del mouse para hacer zoom. Están actualmente en proceso.

Si lo deseas puedes unirte al grupo de depuración visitando el hilo de desarrollo en **https://forum.odroid.com/viewtopic.php?f=94&t=28622**. También puede contribuir con la fuente del kernel en Github en **https://goo.gl/JBrPiB**.

Buscadores, Mineros y 49 – Parte 3: Operación y Mantenimiento de Sistemas de Minería de Cripto-Monedas

May 1, 2018 By Edward Kisiel (@hominoid) ODROID-XU4, Tutoriales

En los dos últimos artículos sobre buscadores, Mineros y 49, me centré en la minería dual CPU/GPU con sgminer-arm-5.5.6-RC y analicé brevemente la evolución térmica del sistema y la puesta a punto de la GPU. En este tercer artículo, echaremos un vistazo a los problemas operacionales más graves de la minería d cripto-monedas y sus repercusiones en el mantenimiento y en el propio sistema, así como los resultados de una prueba de estabilidad de minería dual CPU/GPU que duró cuatro días y ocho horas. De algún modo, estas son las áreas más importantes que pueden marcar la diferencia entre un sistema con un funcionamiento estable y uno con una total inestabilidad; incluso hasta el punto de llegar a provocar un posible daño físico al sistema.

¿Por qué mi sistema se cuelga, se bloquea o no es estable durante el proceso de minado de cripto-monedas? Algunas personas que se enfrentan a este problema se lo preguntan muy a menudo. No hay una única respuesta que responda esta pregunta. El proceso de minado por CPU, y más aún la minería dual CPU/GPU, son actividades informáticas complejas y extremas para un sistema en chip, independientemente de la fabricación del SOC o SBC. Los posibles usos para los que están diseñados e implementados los sistemas SOC y SBC no incluye el tipo de informática extrema a la que están sometidos cada vez con más frecuencia estos sistemas hoy en día. La información que aparece a continuación es el resultado de la experiencia adquirida durante la puesta en funcionamiento de forma activa de un clúster de minería de treinta ODROID. El clúster se compone de veinticinco XU4 con refrigeración activa OEM, un XU4 con refrigeración activa hecha a medida y un sistema cuádruple **MC1**.

Si tenemos en cuenta los típicos usos que se hace de la informática general en la actualidad, casi ninguno se acerca a la asignación de recursos y al potencial que necesita la moderna minería de cripto-monedas y otras aplicaciones informáticas extremas en clúster. Sin embargo, son muchas las aplicaciones de uso extremo de los recursos que se ejecutan regularmente sin tener que gestionar o alcanzar la capacidad y los recursos físicos máximos del sistema. Cualquier alteración por pequeña que sea, en una amplia variedad de áreas puede, y causará, inestabilidad. Estas inestabilidades se pueden manifestar con cuelgues del sistema, bloqueos, errores y daños potenciales en el hardware. Cuando usamos este tipo de aplicaciones, debemos aplicar un enfoque sistemático que nos permita probar los muchos criterios posibles durante la implementación. Éste incluye la frecuencia de la CPU, la temperatura del sistema, la capacidad de refrigeración, el uso de energía, la temperatura ambiente, el uso de recursos del sistema y aplicaciones y el mantenimiento preventivo. La naturaleza dinámica de cualquier entorno, incluso uno que esté pensado para ser controlado, debe ser monitorizado y llevar a cabo los ajustes apropiados. Cualquier variación en un factor cambia y afectará potencialmente a los demás y al sistema en su conjunto.

Esta es una nueva frontera para los SBCs ARM, así que ten en cuenta que te encuentras al borde de la extrema utilización del sistema. Para hacer más hincapié en este punto, podemos recurrir a la analogía de coger tu coche e intentar conducir tan rápido como te sea posible, llevando el tacómetro al límite (línea roja) las 24 horas del día, los 7 días de la semana. Se puede hacer, pero ¿cuánto tiempo durará el coche y qué problemas causará este tipo de conducción? ¿Cómo de fiable y seguro será? Los coches no han sido diseñados para ese tipo de uso, tampoco lo fue el hardware que estamos usando para minar cripto-monedas. ¿Cómo nos enfrentamos a esto? Para empezar, debemos recurrir a la monitorización y realizar ajustes constantemente, aunque hay otra pregunta que debes hacerte: ¿Cuál es mi filosofía operativa? Hay dos líneas de pensamiento en las que se podrían encuadrar la mayoría de los mineros. Un grupo piensa que la inversión de capital de los equipos de minería se ha ido a pique y no tendrá valor residual al final de su ciclo de vida. Creen que la mejor estrategia es hacer que el equipo mine lo máximo posible con el único propósito de maximizar la rentabilidad, y tras un par de años, deshacerse del hardware con cero valor residual. El otro grupo piensa que hay, o debería haber, un valor residual después de un par de años y, como tal, ejecutan sus plataformas mineras de una forma mucho más conservadora. ¿Con qué grupo te identificas? La respuesta a esta pregunta determinará cómo vas a operar y qué es o no aceptable. Quizás haya gente tenga una opinión y un planteamiento diferente. Independientemente de tu estrategia, deberías tener en cuentas estas ocho cuestiones si quieres tener un sistema con un funcionamiento fiable las 24 horas, los 7 días de la semana:

- Frecuencia de la CPU
- Temperatura del sistema
- Capacidad de refrigeración
- Temperatura ambiente
- Consumo de energía
- Uso de recursos del sistemas y aplicaciones
- Mantenimiento preventivo
- Gestión activa

Frecuencia de la CPU

El uso para el que han sido diseñados los SOCs y SBCs no permite minar a la frecuencia máxima de reloj. Como norma general a la hora de configurar un sistema, se debe empezar aproximadamente al 60% de la frecuencia nominal. Esto nos proporciona un punto de partida cómodo para probar la configuración. Si tienes alguna duda sobre la idoneidad de una determinada frecuencia, opta por una estrategia conservadora hasta que estabilices la plataforma. Puedes aumentar fácilmente la frecuencia una vez que hayas probado otras áreas. Deberás ajustar constantemente la frecuencia como parte de la gestión activa de la configuración de tu minado; hablaremos de esto más adelante.

Temperatura del sistema

La realidad es que los 70 °C – 75 °C (158 °F – 167 °F) son las temperaturas máximas de los SOCs XU4/MC1/HC1/HC2 que se pueden mantener para las operaciones de minería las 24 horas los 7 días de la semana. Si superes este límite, es probable que experimentes problemas intermitentes. Puede que aguante un día o dos, pero al final aparecerán fallos en el sistema, bloqueos, errores y una mayor probabilidad de dañar el SOC permanentemente cuanta más alta sea y más tiempo se mantenga la temperatura. No todos los modelos SBCs funcionan de forma idéntica. Existen varias razones que explican esto, no solo las que aparecen en este artículo, lo que algunos denominan la "lotería del silicio". Si estás ejecutando un clúster medianamente grande, se recomienda dividir el clúster en grupos térmicos. Algo tan simple como un sistema de cuatro niveles: caliente, cálido, frío y helado te permitirá administrar el clúster de manera más efectiva y ajustar diferentes parámetros que seran los apropiados para un determinado grupo térmico. Esto se aplica particularmente a la posibilidad de controlar la temperatura del sistema mediante la manipulación de la velocidad del reloj para un determinado grupo.

Capacidad de Refrigeración

Lo primero que tenemos que tener en cuenta en relación a la refrigeración es asegurarnos de que tenemos una pasta térmica que cubre el 100% del SOC y que no hay huecos de aire. Los huecos de aire y las áreas descubiertas actúan como aislante y causarán retenciones de calor y flujos térmicos anormales. Aunque la mayoría de los fabricantes utilizan una pasta térmica aceptable en el rango de 2.5W/mK, considera la posibilidad de cambiarla por algo mejor. Existen muchas pastas térmicas con una transferencia térmica 2-3 veces mejor. Busca alguna que pueda funcionar en el rango de 5W/mK – 8W/mK. Esto nos ayudará a transferir más calor desde el SOC al disipador de calor. Ten cuidado con cualquier cosa que no esté claramente etiquetada o que use una métrica diferente.

Por lo general, la refrigeración pasiva no debe usarse para la minería. Añadir un ventilador a un sistema de refrigeración pasiva puede tener muchos problemas. La cantidad y calidad del flujo de aire depende de muchos factores y, a menos que dediques suficiente tiempo a realizar pruebas con un cambio tan significativo, opta por un sistema de enfriamiento activo OEM. Si va a probar algo diferente, debes tener en cuenta algunos factores como la proximidad del ventilador, el ángulo, la cobertura, la cantidad de flujo de aire y la presión estática. Las pruebas cuantitativas son las únicas que indicarán si realmente se ha logrado una mejoría. Tener un disipador de calor más grande no es necesariamente la mejor solución. El desarrollo de la carcasa XU4 Split Airflow que está descrita en los foros Odroid (**https://forum.odroid.com/viewtopic.php?f=97&t=26373**) y en los números de los meses de abril y junio 2017 de Odroid Magazine (**https://magazine.odroid.com/wp-content/uploads/ODROID-Magazine-201704.pdf** and **https://magazine.odroid.com/wp-content/uploads/ODROID-Magazine-201706.pdf**) puede servirte de ejemplo. A mucha gente le gusta el gran disipador de calor North Bridge utilizado en ese proyecto. Pero no es perfecto y tiene algunos matices que deben abordarse para mejorarlo significativamente. Vale la pena tomarse unos minutos para hablar de ellos a modo de guía para montar un sistema de refrigeración personalizado para la minería.

En el diseño del prototipo inicial, el ventilador y la carcasa no estaban encajonados, lo cual reducía la presión estática de aire. Como tal, proporcionaba un rendimiento muy similar al disipador de calor activo OEM de fábrica. Fue el hecho de introducirlo en una caja y utilizar un ventilador con un mayor volumen de aire lo que permitió aumentar considerablemente el rendimiento. Solo después de agregar una perca de cobre y un esparcidor que incidía en la tubería térmica, se observó una mejora significativa a medida que se completaba la prueba (https://forum.odroid.com/viewtopic.php?f=97&t=26373&start=104). Aun así, cuanto más duraba la prueba, menos eficaz era el disipador de calor en condiciones de mucho trabajo. A la larga termina saturándose a medida que aumenta el tiempo. Está bien para la informática general, pero

cuando se realiza un minado las 24 horas, la mejora va a ser muy poco significativa.

Si se utiliza un ventilador en la parte superior, como suele ser el caso, la presión estática del aire disminuye significativamente porque el disipador de calor no es plano y las aspas son más gruesas y están más juntas. Ambos funcionan en virtud de mejorar la refrigeración reduciendo la cantidad de presión para forzar el aire por debajo del disipador de calor, y desviar más aire hacia afuera en la parte superior del disipador de calor. Es fácil suponer que como es más grande, debería funcionar mucho mejor, sin darte cuenta de que esto puede no ser del todo cierto en el caso de la minería. Si simplemente retiras el ventilador del disipador original de serie y lo montas en la parte superior, el ventilador no está en encajonado, lo que reduce aún más la presión estática permitiendo que penetre menos aire en el disipador. La mayor parte irá por los lados. La lección que tienes que aprender de todo esto es que, si vas a hacer un sistema de refrigeración a medida, presta atención a todos los detalles y realiza pruebas a largo plazo. Un disipador de calor o un ventilador más grande no siempre es significativamente mejor para la minería, depende en gran medida de cómo se implemente y si se apliquen las mejoras adicionales.

Temperatura del Ambiente

Una de las cuestiones que se suele pasar por alto en los sistemas de minería es la temperatura ambiente, especialmente para sistemas no administrados. Los cambios de temperatura en entornos no monitorizados y no controlados pueden ser muy significativos. La temperatura ambiente media de la casa puede variar mucho en un período de 24 horas. Esto importa bastante cuando sobrepasamos los límites en operaciones de minería. Los mineros experimentados lo saben y constantemente revisan sus plataformas por este motivo. Deja que el sol incida en todo o parte del sistema de minería y el efecto será aún mayor. Incluso una determinada ubicación dentro de una habitación o edificio puede ser importante. Cuando se trabaja en el rango de 70 °C – 75 °C (158 °F – 167 °F), como debería ser en la mayoría de los casos, solo es necesario un cambio de 1-2 grados para que afecte al sistema de minado y potencialmente se salga fuera de un rango seguro de trabajo.

Gestión Activa y Mantenimiento

Muchas veces los nuevos “operarios de minería” configuran sus plataformas, las inician utilizando todos los recursos del sistema posibles y piensan que ya terminaron. Esta forma de operar tiene todas las papeletas para que se llegue a una inestabilidad grave en una operación de minería, ya sea ejecutando un único sistema o un gran clúster. Todos los factores de los que estamos hablando deben ser monitoreados constantemente y deben realizarse continuamente ajustes para tener un funcionamiento adecuado y estable. Con recursos mínimos del sistema, la temperatura de la CPU/GPU y la temperatura ambiente de la habitación deben ser monitorizadas permanentemente, la frecuencia de la CPU/GPU o la carga de trabajo cambian en consecuencia.

El mantenimiento preventivo es otra área importante que a menudo suele descuidarse. Como mínimo, debe realizarse con una planificación regular. Aun así, con los ventiladores ensuciados o poco lubricados, se debe mantener una vigilancia constante para disminuir las RPM, el ruido, el polvo y la suciedad. Los disipadores y ventiladores deben mantenerse limpios. Los ventiladores deben girar a sus RPM completas. La operación continua y la electricidad estática aumenta significativamente la aparición de polvo, suciedad y polen. Solo son necesarios algunos meses para que un sistema como este baje su rendimiento y tenga un comportamiento inestable en una habitación con la ventana abierta.

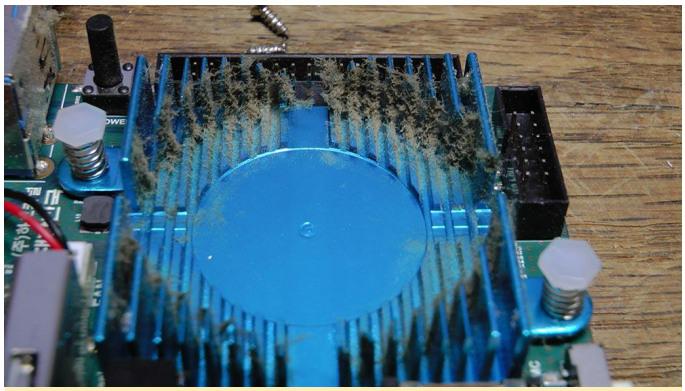


Figura 1 – El disipador de calor ODROID-XU4 debe tener un mantenimiento cuando trabaja en un entorno donde suele acumularse el polvo.

Los ventiladores deben mantenerse en buenas condiciones y, probablemente, también deban lubricarse cada pocos meses. El mejor momento para hacerlo es cuando se limpien los ventiladores y los disipadores. En el caso del disipador de calor activo OEM, se pueden retirar los cuatro tornillos y separar el ventilador de plástico del disipador de calor sin que afecte al disipador térmico y el pegamento térmico. Usa un cepillo de dientes seco para limpiar a fondo las aletas del disipador de calor y ambos lados de las aspas del ventilador. Se puede aplicar un poco de lubricante al eje del ventilador. Para los ventiladores ruidosos, esto se puede hacer entre los ciclos de mantenimiento, sosteniendo el SBC al revés con el ventilador girando mientras aplicas el lubricante con un spray, deteniendo temporalmente el ventilador permitiendo así que el lubricante gotee hacia la estructura del eje. Aunque no es apropiado en todos los casos, WD-40 funcionará bastante bien y no es conductor. Como es de esperar, es bueno contar con ventiladores adicionales para, en caso de ser necesario, reemplazarlos.

En informática general, el mantenimiento es algo que la gente puede descuidar un poco sin que llegue a tener impacto desastroso. En el caso de la minería dual CPU/GPU, el aumento de la demanda de algunos algoritmos de criptografía y de la minería en grupo, al mismo tiempo que los sistemas se ejecutan con todo su potencial, debes asumir el riesgo que implica la ausencia o dejadez de este mantenimiento. La fiabilidad del sistema puede verse seriamente comprometida cuando se permite la introducción de múltiples factores a través de una gestión deficiente, y que se van acumulando con un insuficiente mantenimiento. Ten en cuenta que no estamos hablando de un uso moderado del ordenador: estamos hablando de ejecutar los sistemas operativos con todo su potencial durante un periodo de tiempo indefinido. Recuerdas nuestra analogía con el coche; llevar el pedal al fondo con el tacómetro al límite. Completando el círculo, volvamos a nuestra pregunta original: ¿por qué mi sistema se cuelga, se bloquea o no es estable mientras se procesas las cripto-moneda? Aquí tienes donde buscar.

Prueba de estabilidad a largo plazo

Tras una prueba de estabilidad de cuatro días y ocho horas con un sistema dual CPU/GPU minando Monero utilizando el algoritmo cryptonight en un clúster ODROID-MC1, todo funciona como era de esperaba sin errores reportados en ningún registro log de sistema. Se usaron Sgminer-arm-5.5.6-RC1, XMRig y cpuminer-multi y su funcionamiento fue normal. El hashrate reportado era aproximadamente para cada las GPUs de 19 h/s, para las CPUs de 19 h/s según lo reportado por la aplicación. Todas las máquinas se ejecutaban a una frecuencia de 1.7Ghz, temperatura ambiente 71f (21.66c)

Versión de Linux

```
 Linux c5n0 4.14.5-92 #1 SMP PREEMPT Mon Dec 11 15:48:15 UTC 2017 armv7l armv7l armv7l GNU/Linux
```

Aplicationes Usadas

```
 c5n0 - GPU sgminer-5.5.6-ARM-RC1, CPU XMRig version 2.44
 c5n1 - GPU sgminer-5.5.6-ARM-RC1, CPU XMRig version 2.51
 c5n2 - GPU sgminer-5.5.6-ARM-RC1, CPU cpuminer-multi version 1.3.1
 c5n3 - GPU sgminer-5.5.6-ARM-RC1, CPU cpuminer-multi version 1.3.1
```

Configuraciones de las aplicaciones

```
 sgminer-5.5.6-ARM-RC1 GPU Configuration
 -I 6 -w 32 -d 0,1 --thread-concurrency 8192 --monero --pool-no-keepalive

XMRig version 2.44 & 2.51 CPU Configuration
```

```
 -t 7 --cpu-affinity 0xFE

cpuminer-multi CPU Configuration
 -t 7 --randomize --no-redirect --cpu-affinity
0xFE
```

Resumen de los resultados de sgminer-arm-5.5.6-RC1

c5n0

```
 [13:53:00] Shutdown signal received.
 [13:53:00]
 Summary of runtime statistics:

[13:53:00] Started at [2018-03-25 05:38:19]
[13:53:00] Pool:
stratum+tcp://pool.supportxmr.com:3333
[13:53:00] Runtime: 104 hrs : 14 mins : 40
secs
[13:53:00] Average hashrate: 0.0 Kilohash/s
[13:53:00] Solved blocks: 0
[13:53:00] Best share difficulty: 16.2M
[13:53:00] Share submissions: 1012
[13:53:00] Accepted shares: 995
[13:53:00] Rejected shares: 17
[13:53:00] Accepted difficulty shares: 5006256
[13:53:00] Rejected difficulty shares: 85000
[13:53:00] Reject ratio: 1.7%
[13:53:00] Hardware errors: 352
[13:53:00] Utility (accepted shares / min):
0.16/min
[13:53:00] Work Utility (diff1 shares solved /
min): 0.16/min

[13:53:00] Stale submissions discarded due to
new blocks: 0
[13:53:00] Unable to get work from server
occasions: 272
[13:53:00] Work items generated locally:
407984
[13:53:00] Submitting work remotely delay
occasions: 0
[13:53:00] New blocks detected on network:
3096

[13:53:00] Summary of per device statistics:

[13:53:00] GPU0 | (5s):9.359 (avg):9.341h/s |
A:2522369 R:25000 HW:170 WU:0.081/m
[13:53:00] GPU1 | (5s):9.361 (avg):9.329h/s |
A:2483886 R:60000 HW:182 WU:0.081/m
```

c5n1

```
[13:52:55] Shutdown signal received.
13:52:55]
Summary of runtime statistics:

[13:52:55] Started at [2018-03-25 05:38:28]
[13:52:55] Pool:
stratum+tcp://pool.supportxmr.com:3333
[13:52:55] Runtime: 104 hrs : 14 mins : 26
secs
[13:52:55] Average hashrate: 0.0 Kilohash/s
[13:52:55] Solved blocks: 1
[13:52:55] Best share difficulty: 1.23M
[13:52:55] Share submissions: 1027
[13:52:55] Accepted shares: 1008
[13:52:55] Rejected shares: 19
[13:52:55] Accepted difficulty shares: 5053564
[13:52:55] Rejected difficulty shares: 95000
[13:52:55] Reject ratio: 1.9%
[13:52:55] Hardware errors: 353
[13:52:55] Utility (accepted shares / min):
0.16/min
[13:52:55] Work Utility (diff1 shares solved /
min): 0.16/min

[13:52:55] Stale submissions discarded due to
new blocks: 0
[13:52:55] Unable to get work from server
occasions: 223
[13:52:55] Work items generated locally:
407460
[13:52:55] Submitting work remotely delay
occasions: 0
[13:52:55] New blocks detected on network:
3096

[13:52:55] Summary of per device statistics:

[13:52:55] GPU0 | (5s):9.331 (avg):9.351h/s |
A:2405910 R:50000 HW:176 WU:0.078/m
[13:52:55] GPU1 | (5s):9.324 (avg):9.340h/s |
A:2647653 R:45000 HW:177 WU:0.086/m
```

c5n2

```
[13:52:48] Shutdown signal received.
[13:52:48]
Summary of runtime statistics:

[13:52:48] Started at [2018-03-25 05:38:38]
[13:52:48] Pool:
stratum+tcp://pool.supportxmr.com:3333
```

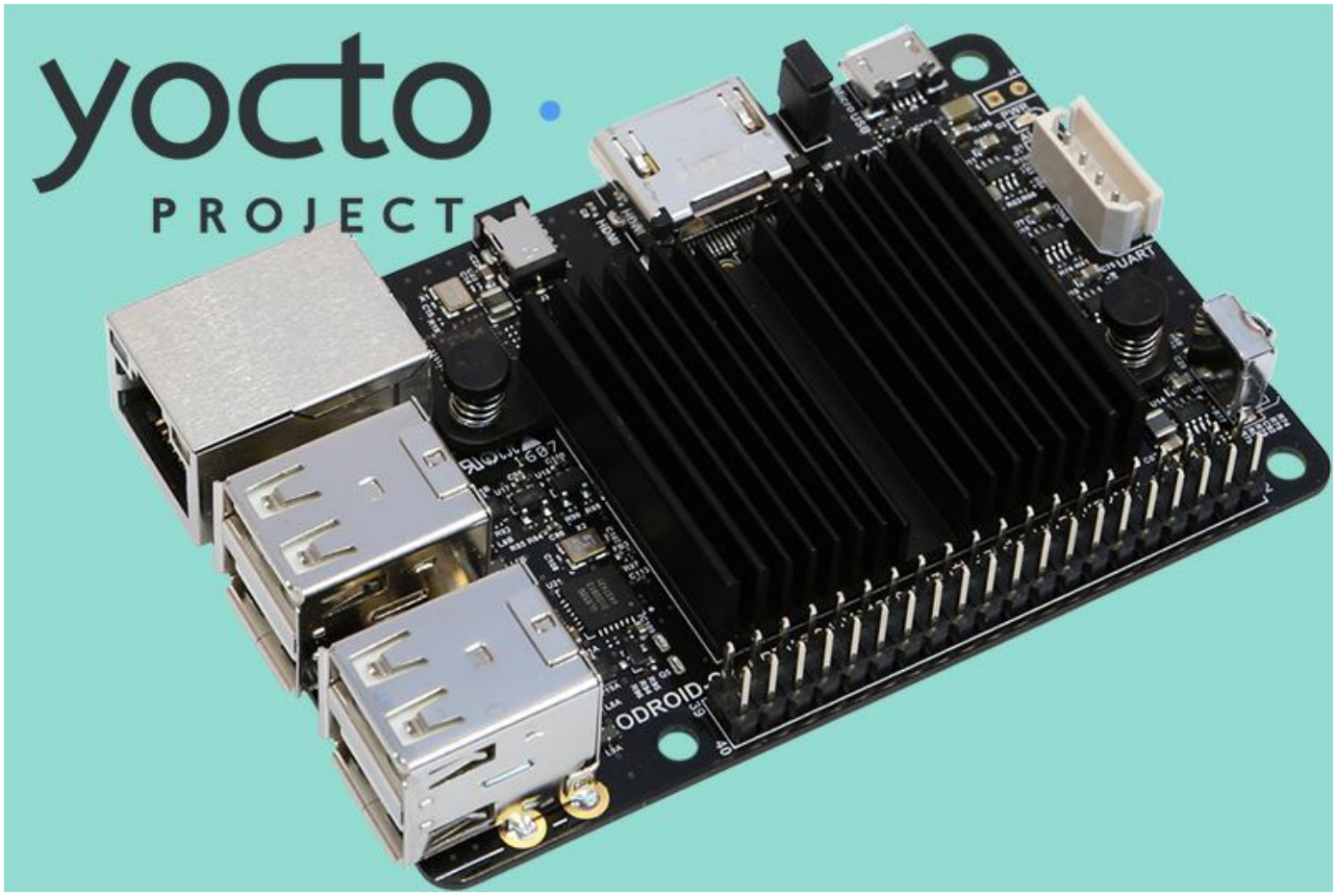
```
[13:52:48] Runtime: 104 hrs : 14 mins : 9 secs
[13:52:48] Average hashrate: 0.0 Kilohash/s
[13:52:48] Solved blocks: 1
[13:52:48] Best share difficulty: 50.1M
[13:52:48] Share submissions: 1034
[13:52:48] Accepted shares: 1009
[13:52:48] Rejected shares: 25
[13:52:48] Accepted difficulty shares: 5081646
[13:52:48] Rejected difficulty shares: 125000
[13:52:48] Reject ratio: 2.4%
[13:52:48] Hardware errors: 334
[13:52:48] Utility (accepted shares / min):
0.16/min
[13:52:48] Work Utility (diff1 shares solved /
min): 0.17/min

[13:52:48] Stale submissions discarded due to
new blocks: 0
[13:52:48] Unable to get work from server
occasions: 257
[13:52:48] Work items generated locally:
414051
[13:52:48] Submitting work remotely delay
occasions: 0
[13:52:48] New blocks detected on network:
3099

[13:52:48] Summary of per device statistics:

[13:52:48] GPU0 | (5s):9.226 (avg):9.186h/s |
A:2607526 R:45000 HW:172 WU:0.084/m
[13:52:48] GPU1 | (5s):9.225 (avg):9.188h/s |
A:2474119 R:80000 HW:162 WU:0.081/m
```

c5n3

```
[13:52:38] Shutdown signal received.
[13:52:38]
Summary of runtime statistics:

[13:52:38] Started at [2018-03-25 05:38:47]
[13:52:38] Pool:
stratum+tcp://pool.supportxmr.com:3333
[13:52:38] Runtime: 104 hrs : 13 mins : 51
secs
[13:52:38] Average hashrate: 0.0 Kilohash/s
[13:52:38] Solved blocks: 3
[13:52:38] Best share difficulty: 4.01M
[13:52:38] Share submissions: 1059
[13:52:38] Accepted shares: 1028
[13:52:38] Rejected shares: 31
[13:52:38] Accepted difficulty shares: 5165010
[13:52:38] Rejected difficulty shares: 155000
[13:52:38] Reject ratio: 2.9%
[13:52:38] Hardware errors: 350
[13:52:38] Utility (accepted shares / min):
0.16/min
[13:52:38] Work Utility (diff1 shares solved /
min): 0.17/min

[13:52:38] Stale submissions discarded due to
new blocks: 1
[13:52:38] Unable to get work from server
occasions: 251
[13:52:38] Work items generated locally:
405818
[13:52:38] Submitting work remotely delay
occasions: 1
[13:52:38] New blocks detected on network:
3096

[13:52:38] Summary of per device statistics:

[13:52:38] GPU0 | (5s):9.319 (avg):9.247h/s |
A:2365471 R:75000 HW:175 WU:0.078/m
[13:52:38] GPU1 | (5s):9.336 (avg):9.265h/s |
A:2799539 R:80000 HW:175 WU:0.092/m
```

El Proyecto Yocto: Instalado y Funcionando en el ODROID-C2

May 1, 2018 By Khem Raj, Himvis LLC Linux, ODROID-C2



El proyecto Yocto es un proyecto de código abierto que proporciona un conjunto de herramientas flexibles para crear distribuciones de Linux integradas personalizadas para dispositivos embebidos y IoT. Incluye soporte para las principales arquitecturas de CPU que predominan en la industria embebida. A través de la colaboración, se crean flujos de trabajo en toda la industria para los desarrolladores que permiten el intercambio de tecnologías y pilas de software. Los mismos flujos de trabajo, plantillas de infraestructura y configuraciones también proporcionan un lugar para alojar capas BSP. Las versiones del proyecto Yocto son publicadas cada seis meses, en abril y octubre.

Este artículo describe los componentes fundamentales y el proceso para compilar una imagen Linux ODROID-C2 personalizada. Estos mismos pasos se pueden usar en otras máquinas ODROID. Yocto es la herramienta estándar de la industria para la creación de sistemas embebidos Linux personalizados y complejos que utilizan las últimas tecnologías de código abierto como Qt5, QtWebEngine y Grafana.

Configuración del sistema host y requisitos previos

El proyecto Yocto requiere un sistema de compilación basado en Linux y es compatible con las principales distribuciones de escritorio y de servidor Linux, tienes una lista de distribuciones compatibles en **https://www.yoctoproject.org/docs/current/ref-manual/ref-manual.html#detailed-supported-distros**. El sistema de compilación Yocto compila la mayor parte de los paquetes dependientes del sistema host, lo cual proporciona una mayor coherencia entre las diferentes distribuciones Linux. Sin embargo, se espera que ciertos paquetes estén preinstalados en el sistema de compilación del sistema host. Para un sistema como Debian, se debe instalar los siguientes paquetes:

```
$ sudo apt-get install gawk wget git-core
diffstat unzip texinfo gcc-multilib build-
essential chrpath socat cpio python python3
python3-pip python3-pexpect xz-utils
debianutils iputils-ping
```

Tienes una lista completa de los requisitos del sistema de desarrollo host **https://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html#qs-native-linux-build-host**.

Obtención de las Fuentes

El proyecto Yocto usa el concepto de capas para crear el espacio de trabajo. La capa central proporciona todas las piezas comunes y las capas adicionales cambian la pila de software según sea necesario. Las siguientes instrucciones se basan en el maestro en sentido ascendente; sin embargo, también es posible usar una rama de la versión, por ejemplo, "sumo" o más nuevo.

```
$ git clone -b master
git://git.yoctoproject.org/poky.git yocto-
odroid
$ cd yocto-odroid
```

Descargar la capa BSP ODROID:

```
$ git clone -b master
git://github.com/akuster/meta-odroid
```

Inicia la configuración:

```
$ source yocto-odroid/oe-init-build-env
```

Ahora tenemos un espacio de trabajo común con una capa básica donde podemos compilar un emulador y las imágenes basadas en la placa de referencia, por ejemplo, qemuarm. A continuación, podemos agregar la capa BSP ODROID en el proyecto para que podamos realizar compilaciones para las placas ODROID:

```
$ bitbake-layers add-layer ../meta-odroid
```

A continuación, Selecciona el ODROID-C2 como nuestra máquina:

```
$ echo 'MACHINE = "odroid-c2"' >>
conf/local.conf
```

El espacio de trabajo ahora está listo para empezar una compilación.

Compilación

El sistema de compilación del proyecto Yocto proporciona algunas imágenes de referencia a modo de ejemplo para varios posibles usos. Nosotros vamos a compilar una imagen gráfica basada en X11 y Matchbox. Existen varias imágenes de referencia adicionales disponibles en **https://www.yoctoproject.org/docs/current/ref-manual/ref-manual.html#ref-images**.

```
$ bitbake core-image-sato
```

Esta compilación llevará un tiempo dependiendo de la potencia de la máquina de compilación y puede tardar entre 20 minutos y varias horas.

Grabando una tarjeta SD

Tras una compilación exitosa, los ítems de la compilación son almacenados en el directorio "tmp/deploy/images/odroid-c2". Se puede usar una herramienta como Etcher para crear una tarjeta SD de arranque. Esto también se puede hacer usando la línea de comandos shell, con dd. Sin embargo, se debe tener mucho cuidado ya que, si se eliges el dispositivo incorrecto, puedes sobrescribir un disco duro que pertenezca al sistema host de compilación.

```
$ cd tmp/deploy/images/odroid-c2
$ xzcat core-image-sato-odroid-c2.wic.xz |
sudo dd of=/dev/sdX bs=4M iflag=fullblock
oflag=direct conv=fsync status=progress
```

Asegúrate de que sdX apunta a la tarjeta SD montada, esto puede confirmarse con dmesg una vez insertada la tarjeta

```
% dmesg|tail
[  +0.000149] scsi host6: usb-storage 4-4:1.0
[  +0.000077] usbcore: registered new
interface driver usb-storage
[  +0.002803] usbcore: registered new
interface driver uas
[  +1.005024] scsi 6:0:0:0: Direct-Access
TS-RDF5  SD  Transcend    TS37 PQ: 0 ANSI: 6
[  +0.291506] sd 6:0:0:0: [sdb] 15523840 512-
byte logical blocks: (7.95 GB/7.40 GiB)
[  +0.000682] sd 6:0:0:0: [sdb] Write Protect
```

```
is off
[  +0.000003] sd 6:0:0:0: [sdb] Mode Sense: 23
00 00 00
[  +0.000688] sd 6:0:0:0: [sdb] Write cache:
disabled, read cache: enabled, doesn't support
DPO or FUA
```

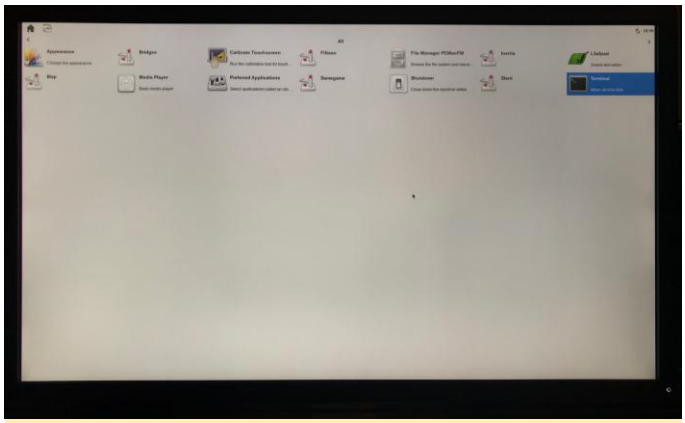


Figura 1 – UI Yocto Project Sato ejecutándose en ODROID-C2

Capa BSP ODROID

La capa BSP ODROID es compatible con varias máquinas ODROID, sobre todo con el kernel y el cargador de arranque (u-boot) basado en fuentes ascendentes. Hay un ejemplo de configuración de máquina para el ODROID-C2, odroid-c2-hardkernel, que utiliza un Kernel y u-boot compatibles con Hardkernel.

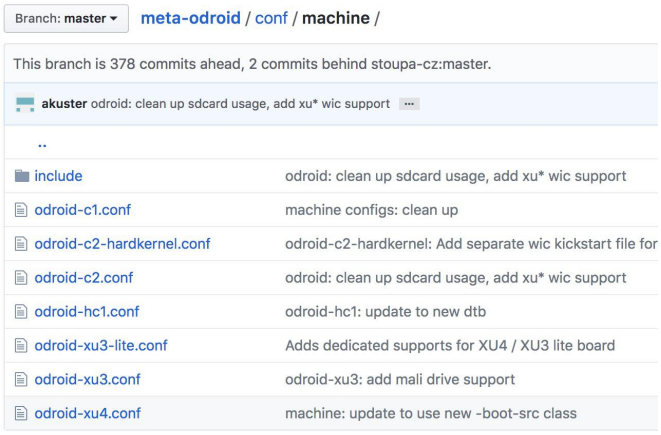


Figura 2 – Máquina ODROID compatible con Yocto Project

A día de hoy, los siguientes elementos BSP son compatibles:

- Linux estándar – 4.14(LTS) and 4.16
- Linux HardKernel – 3.14(EOL) y 3.16(LTS)
- U-Boot – 2018.01 así como u-boot-hardkernel 2015.10
- Mali 450 drivers pre-compilados r6p1 y Mali t62x drives pre-compilados r10p0_00rel0
- El Soporte para el módulo LCD de 3.5” de Hardkernel ha sido añadido además de odroid-lcd35

Liberando el Ecosistema del Proyecto Yocto

Hay muchas capas disponibles (consulta el Índice de capas en http://layers.openembedded.org/layerindex/branch/master/layers/) que se pueden añadir para crear imágenes más complejas. Por ejemplo, puede agregar la capa meta-qt5 para compilar un sistema basado en Qt5 utilizando la tecnología QtWebEngine para sistemas kioscos. La Figura 3 muestra un cuadro de mandos Grafana ejecutándose en un Navegador Kiosco creado usando QtWebEngine en ODROID-C2, todo compilado desde la fuente usando el Proyecto Yocto.

Panel de control Grafana en QtWebEngine ejecutándose en ODROID-C2 creado con el Proyecto Yocto

Conociendo a un ODROIDian: Matthew Kinderwater (WebClaw)

May 1, 2018 By Rob Roy Conociendo un ODROIDian

Por favor, háblanos un poco sobre ti.

Soy el Director de Servicios TI en una empresa llamada **iCube Development** que tiene su sede en Calgary, Alberta, Canadá. Entre mis funciones principales se encuentran la recuperación de datos, trabajar en un laboratorio realizando tareas como el reemplazo de cabezales, reparaciones eléctricas y recuperar datos de volúmenes RAID. Tengo 34 años y tengo experiencia como programador independiente de PHP y MySQL. He desarrollado un sistema gratuito de facturación online llamado iCDBILL y recientemente he impulsado el desarrollo de otras aplicaciones de código abierto como ICDBill y Billwerx. En 2005 empecé a trabajar en un laboratorio de recuperación de datos llamado iCube Development en Calgary. Disfruto con la acampada, montando en bicicleta y cazando. Mis vacaciones perfectas serían estar con mi familia lejos de la tecnología y el WiFi.

De joven, era socorrista, maestro instructor de natación, y trabajaba como un EMR (entrevistado médico de emergencias). Mi esposa trabaja para iCube Development como directora de finanzas. Actualmente está completando nuestro Programa Contable General Certificado. Tengo una hija que (en el momento de escribir esto) tiene tan sólo 13 semanas. Recientemente he aparecido en una publicación llamada City Life, donde formé parte del artículo **Top 40 Under 40 article**, y he recibido un premio de la **Aboriginal Multi-Media Society por mi trayectoria profesional**. Puedes echar un vistazo a mi contribución de código abierto llamada **Billwerx on Youtube**, y leer sobre uno de mis **casos de exito en la recupación de big data.**

¿Cómo empezaste con los ordenadores?

Siempre me ha interesado la tecnología y, siendo muy joven, mi padre llevaba a casa ordenadores que no funcionaban de su trabajo para que los desmontara.

A los 9 años, mi padre me trajo mi primer ordenador. ¡Se trataba de un 8086 que se ejecutaba a 4MHz con un disco duro de 20MB y MSDOS 3.3! En la escuela primaria, aprendí BASIC y continúe aprendiendo más sobre sistemas operativos, programación y hardware.

¿Qué te atrajo a la plataforma ODROID?

La arquitectura ARM está creciendo muy rápido. Los fabricantes como Intel y AMD han sido muy dominantes con su conjunto de instrucciones x86 y x64, y creo que ARM aporta una dosis muy buena de competitividad al mercado. A diferencia de muchos productos chinos que anuncian grandes especificaciones técnicas, HardKernel y la comunidad ODROID desarrollan activamente el kernel, lanzan parches y ofrecen soporte técnico gratuito en los foros. Si comparamos el bajo rendimiento del hardware Raspberry Pi con ODROID, los desarrolladores lo tienen fácil a la hora de optar por una u otra. En mi opinión, el ODROID-C2 es el dispositivo con capacidad 4K y almacenamiento eMMC más estable del mercado hoy en día.

¿Cómo usas tus ODROID?

Usamos muchos ODROID en casa y en la oficina. Para el hogar, utilizamos ODROID-C2 con LibreElec y ODROID-XU4 como almacenamiento conectado en red (NAS). En el trabajo, utilizamos ODROID-XU4s en casos de tipo CloudShell impresión 3D para diagnósticos de HDD, reparaciones simples a nivel de archivos y funciones automatizadas de duplicado de datos.

¿Cuál es tu ODROID favorito y por qué?

Eso es fácil: el ODROID-C2 es muy estable, tiene compatibilidad con GPIO de Raspberry Pi, soporta salida 4K y utiliza las rápidas unidades eMMC para el almacenamiento de datos.

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

Me gustaría que se añadiera WiFi y Bluetooth a la PCB con cables de antena U.FL. Esto haría que estuviese completamente a la par con los dispositivos Raspberry Pi.

¿Qué hobbies s e intereses tienes aparte de los ordenadores?

iCube Development financia proyectos de código abierto, por lo que he tenido la suerte de contar con fondos para Maker Projects. Durante los últimos 5 años, he estado muy metido en el proyecto Layer3D, que diseña impresoras 3D y componentes fiables en todo el mundo. Todos nuestros diseños y construcción de materiales se hacen públicos, incluidos los archivos fuente.

```
[livingroom:~ # cd /dev
[livingroom:/dev # ls i2c*
i2c-1
[livingroom:/dev # i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10: 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20: 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30: -- -- -- -- -- -- -- -- 38 39 3a 3b 3c 3d 3e 3f
40: 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50: -- UU -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70: 70 71 72 73 74 75 76 77
livingroom:/dev #
```

Figura 1 – La impresora 3D Layer3D Theta es uno de los proyectos de código abierto de Matthew

¿Qué consejo le darías alguien que quiere empezar a aprender sobre tecnología de la información y programación?

Los foros de la comunidad son más valiosos que los libros y la educación formal. Los libros y la escuela son buenos, pero la mayoría de los problemas informáticos que los técnicos necesitan resolver usan la formación reglada como base para saber cómo funciona el sistema operativo y los programas, pero no son una guía efectiva para resolver los problemas. Por ejemplo, si el motor de su automóvil se rompe, un mecánico recibe una guía del fabricante, como Ford o GM, para reemplazar el motor. La guía le dice exactamente qué hacer y cómo se debe hacer. Sin embargo, algunos mecánicos tienen la habilidad de montar su propio motor y corregir problemas que no están en un código de diagnóstico.

Resumiría diciendo que te cuestionaras por qué estás haciendo las cosas. Un buen técnico puede seguir instrucciones de Google, pero un excelente técnico sabe por qué está haciendo algo. Arriésgate y no tenga miedo de meter la pata. Es cómo mejor aprendemos y cómo nos volvemos mejores que la escuela o los libros.