Desarrolla tu propio UBUNTU partiendo de cero • Cómo Instalar Oracle JDK Versión 8

ODROID Magazine

Año Uno
Num #4
Apr 2014

HAZLO TU MISMO!!!

Campo a través con un ODROID PC

TUTORIALES PASO A PASO SOBRE COMO CREAR:

UN MINECRAFT SERVER

UNA RESISTENTE TABLET

COMPILA Y PERSONALIZA ANDROID

Qué defendemos

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

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

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

Simple, moderno y único.
De modo que tienes a tu alcance lo mejor
para alcanzar todo lo que puedas imaginar.

HARDKERNEL

EDITORIAL

Linaro 12.11, que llega al final de su vida este mes, fue la última versión de Ubuntu en ofrecer el entorno de escritorio Unity 2D, muy popular entre principiantes y expertos por sus amigables iconos, opciones únicas para personalizar el escritorio y una sencilla interfaz. Sin embargo, su predecesor y pariente cercano, Linaro 12.04, sigue todavía sano y salvo, y también ofrece Ubuntu 2D. 12.04 es la versión LTS (Soporte a Largo Plazo) más reciente y contará con soporte 3 años más, hasta abril de 2017. Si estás buscando una versión de Ubuntu que sea estable, Linaro 12.04 es tu mejor apuesta.

Sin embargo, la versión 12.04 de Ubuntu no está disponible como imagen pre-compilada desde Hardkernel. ¿Por qué? Porque, como se suele decir, ¡Puedes hacerlo tú mismo! La familia de ordenadores ODROID está principalmente dirigida a desarrolladores, a los que les gusta desarrollar todo desde cero, por dos razones:

1) Por lo general cobrán por hora, y 2) se pasan los días escribiendo grandes y complejos script de desarrollo que necesita horas para completarse, de modo que pueden ir a hacerte unos sándwiches y tomar un café mientras esperas a que la compilación finalice!

Este mes, Mauro nos muestra cómo crear una imagen personalizada de Ubuntu desde cero, de forma que puedes sorprender a tus amigos en tu próxima fiesta y demostrar que eres un autentico hacker Linux, digno de adoración y admiración.

También estamos muy orgullosos de presentar una nueva tendencia en el mundo del automóvil: un ordenador totalmente funcional instalado en el salpicadero de un coche. Conocidos como Car PCs (Ordenadores para coche), varias compañías de informática a gran escala han contratado con los fabricantes de automóviles más importantes para incluir este hardware como un extra en algunos modelos de alta gama.

Pero, ¿Quién dice que los Car PC tiene que ser caros? Nuestro artículo especial "Cómo crear un Car PC para mi Camioneta usando ODROID", ofrece una guía para crear tu propio ordenador de a bordo, como una alternativa económica a un iPad instalado en el salpicadero de tu coche. Necesita menos de 5W de potencia, el Car PC ODROID y su batería pueden cargarse directamente desde el sistema eléctrico, o usando un pequeño panel solar situado en el techo. Realmente la informática móvil ya es una realidad y la línea de micro-ordenadores ODROID ha demostrado una vez más adelantarse a su tiempo.

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

HARDKERNEL

ODROID Magazine

Robert Hall, Chief Editor

Soy un programador que vive y trabaja en Silicon Valley, CA, EE.UU. Diseño sitios web como Vevo, Hi5, Dolby Laboratories y Hyundai. Mis lenguajes principales son jQuery, angular JS y HTML5/CSS3. También desarrollo sistemas operativos pre-compilados, Kernels a medida y aplicaciones optimizadas para ODROID basadas en las versiones oficiales Hardkernel, por los cuales he ganado varios premios. Poseo una gran cantidad de ODROIDs, que uso para diversos fines: centro multimedia, servidor web, desarrollo de aplicaciones y plataforma de juegos, entre otros.

Bo Lechnowsky, Editor

Soy el presidente de Respectech, Inc., Consultoría tecnológica en Ukiah, CA, EE.UU. que fundé en 2001. Con mi experiencia en electrónica y programación dirijo a un equipo de expertos, además de desarrollar soluciones personalizadas a empresas, desde pequeños negocios a compañías internacionales. Los ODROIDs son una de las herramientas de las que dispongo para hacer frente a estos proyectos. Mis lenguajes favoritos son Rebol y Red, ambos se ejecutan en los sistemas ARM como el ODROID-U2. Tengo extensa experiencia con la mayoria de sistemas operativos.

Bruno Doiche, Art Editor

Yendo un poco más loco que de costumbre, estudio de nuevo como una persona daltónica ve, y pierde a un colega daltónico como cuando trabajó en EGM Brasil en sus viejos tiempos como editor de revista de videojuegos.

Manuel Adamuz, Traductor

Un fanático de las nuevas tecnologías, la informatica y como no, del mundo ODROID.

Novedades de esta Edición:

En este número observarás algunos cambios. El primero es que estamos usando colores en la parte superior de cada artículo para indicar su nivel de dificultad. Esto amplía nuestra gama de colores y además, estamos intercalando pequeños artículos sobre temas diversos como consejos linux y juegos Android.

También hemos cambiando el formato de los artículo técnicos, mediante el uso de dos columnas cuando lo necesitamos.

¿Por qué? Cuando necesitamos escribir largas cadenas de código, éstas terminan cortándose. ¡Esto es un fastidio!

Esto significa que, en algunos casos, modificaremos el estilo de tres columnas. Mi profesor de diseño le gustaría recordarme en este momento lo importante que es mantener una maquetación homogénea. Sin embargo, los artículos serán más comprensibles a la hora de entender y seguir el código, que al fin al cabo es lo más importante.

Por otro lado y a partir de ahora, deja-remos algo de espacio al final de estoss artículos técnicos, con el fin de añadir posibles revisiones futuras. Una vez que se publica un artículo, realizamos ajustes basándonos en los comentarios y sugerencias de los lectores, y si éstos están muy apretados resulta muy difícil añadir cualquier modificación.

Por último, hemos añadido un índice, que se puede ver en la siguiente página. Genial, ¿eh? Ahora no tendrás que detenerte con mis pequeñas bromas, ¡No es que no quiera bromear de ven en cuando!

INDICE

DESARROLLAR ANDROID EN ODROID-U3

PARTIENDO DE CERO, HAZTE CON EL CONTROL TOTAL DE TU SISTEMA ANDROID

por Nanik Tolaram y Fabien Robert

En este tutorial mostraré cómo desarrollar el sistema operativo Android para ODROID-U3 desde la fuente, incluyendo el kernel. El sistema de desarrollo de Android es sólido, pero también algo complicado si no lo has usado antes. Hay pasos que deben hacerse correctamente para conseguir un sistema viable y funcional. Al finalizar este artículo, espero que adquieras los conocimientos necesarios y comprendas cómo funciona todo.

Entorno y Hardware de Desarrollo

No voy a entrar en detalle sobre como configurar un servidor de desarrollo para Android puesto que la página de Android de Google tiene mucha información sobre este tema `http://source.android.com/source/initializing.html`. Si tienes problemas para instalar JDK 6, siga los pasos de este enlace: `http://askubuntu.com/questions/67909/how-do-i-install-oracle-jdk-6`.

Desarrollar el código fuente de Android es una tarea compleja y requiere de una potente máquina. Para que os hagáis una idea, mi equipo tiene:

32GB de RAM
Procesador Intel i5
2 Unidades SSD de 256GB

Los sistemas de desarrollo Android realizan muchas escrituras y lecturas, y esto a su vez requiere constantemente operaciones E/S. Incluso con una unidad SSD, tienes que esperar unos 25-35 minutos para que la compilación se complete. Esto te puede hacer perder mucho tiempo si trabajas con Android diariamente. Asegúrate de tener tanto espacio de disco libre como te sea posible, como mínimo 100 GB. Existe un truco para acelerar el proceso de compilación usando ccache, que explicare más adelante.

Si tu hardware no es tan potente como un i5 o i7 y está usando un disco duro normal ¡Asegúrate de tener un café listo!

Descargar la Fuente

El código fuente de Android 4.1.2 (JellyBean) usado en este artículo se puede descargar desde el sitio web droid.com en `http://dn.odroid.com/4412/Android/4.1.2_Jan-15-2014/BSP/`. Hay un par de archivos que debes descargar desde este enlace, com muestra la imagen de abajo.

Descarga los archivos android.tgz y kernel.tgz, y extraerlos a un directorio en tu unidad local. Ponga los archi-

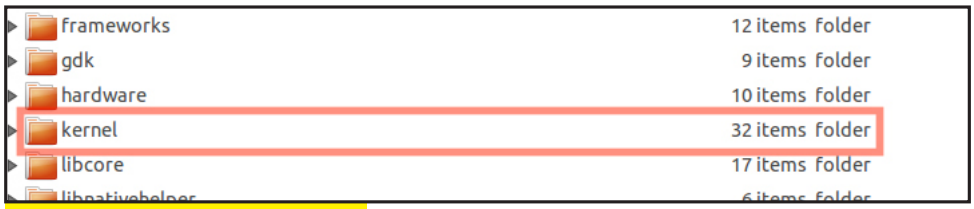
Código fuente del Kernel y Android

dn.odroid.com/4412/Android/4.1.2_Jan-15-2014/BSP/

Index of /4412/Android/4.1.2_Jan-15-2014/BSP

Name	Last modified	Size	Description
Parent Directory		-	
android.tgz	15-Jan-2014 17:30	2.4G	
android.tgz.md5sum	15-Jan-2014 17:30	46	
kernel.tgz	15-Jan-2014 17:30	104M	
kernel.tgz.md5sum	15-Jan-2014 17:30	45	
uboot.tgz	15-Jan-2014 11:46	22M	
uboot.tgz.md5sum	15-Jan-2014 11:46	44	

Apache/2.2.22 (Ubuntu) Server at dn.odroid.com Port 80



Directorio Kernel dentro de Android

vos del kernel en el directorio /kernel bajo el directorio principal de Android, como muestra la Figura de arriba

La principal razón de colocar el Kernel en del directorio Android es facilitar la creación del script de desarrollo, ya que el sistema de compilación gira en torno a los archivos de este directorio.

He creado un conjunto de parches para este artículo en **https://github.com/nanikjava/odroid-u-patch**. Este parche le permite compilar Android y el kernel al mismo tiempo. Ejecuta el comando:

```
git apply --stat ./odroid-u-patch/fix-build-odroid-u3.patch
```

y verá el resultado que se muestra en la parte inferior de esta pagina.

Hay 3 nuevo archivos y 2 modificaciones para este parche. Asegúrate de colocarlos dentro de tu directorio de Android y aplicar el parche ejecutando el siguiente comando:

```
git apply ./odroid-u-patch/fix-build-odroid-u3.patch
```

Verá los siguientes mensajes que se pueden ignorar sin problema:

```
./odroid-u-patch/fix-build-odroid-u3.patch:171: trailing whitespace.
./odroid-u-patch/fix-build-odroid-u3.patch:173: trailing whitespace.
    ccache =
warning: 2 lines add whitespace errors.
```

Resultado del parche

```
build/core/tasks/kernel.mk                 |  199 ++++++++++++++++++++++++++++++
buildOdroid.sh                             |    5 +
build_android.sh                           |    5 +
device/hardkernel/boards/vendorsetup.sh    |    1
device/hardkernel/odroidu/BoardConfig.mk   |    3
5 files changed, 212 insertions(+), 1 deletion(-)
```

Un fichero muy importante para el proceso de compilación es Makefile, debe ser copiado al directorio kernel/drivers/media/video/samsung/tvout.

Modificación del script y ccache

He mencionado el uso de ccache para acelerar el proceso de compilación, vamos a ponerlo en marcha. En primer lugar, debe saber que ccache requiere algo de espacio libre en disco. En este caso, vamos a configurarlo para que solo use 10 GB, que será más que suficiente.

Cree un directorio en tu unidad, a continuación, activa la variable de entorno y ejecutar los siguientes comandos para iniciar ccache:

```
export CCACHE_DIR=\
<your_ccache_directory>

<your__Android_directory>/
prebuilts/misc/linux-x86/
ccache/ccache -M 10G
```

Puedes comprobar si ccache se ha iniciado correctamente analizando el directorio caché, como se muestra en la parte superior derecha de esta página.

El último paso es modificar el script buildOdroid.sh y cambiar el directorio ccache para que apunte a tu directorio local:

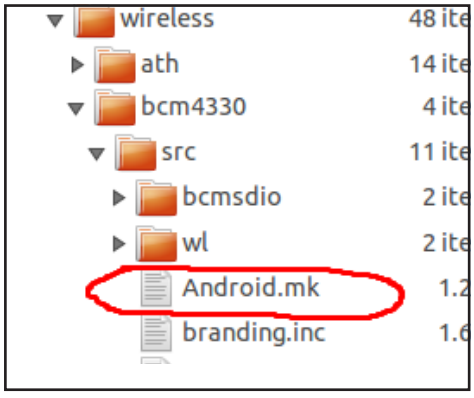
```
drwxrwxr-x 18 nanik nanik 4096 Jan 27 22:39 ./
drwxrwxr-x 28 nanik nanik 4096 Jan 27 22:36 ../
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 0/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 1/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 2/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 3/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 4/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 5/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 6/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 7/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 8/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 9/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 a/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 b/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 c/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 d/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 e/
drwxrwxr-x  2 nanik nanik 4096 Jan 27 22:39 f/
```

El directorio caché y sus subdirectorios, enumerados en hexadecimal desde 0 a F

```
source build/envsetup.sh
lunch odroidu-eng
export USE_CCACHE=1
export CCACHE_DIR=\
<your_ccache_directory>
/usr/bin/time -f "\n%E
elapsed,\n%U user,
\n%S system,\n%M memory,\n%x
status" make -j8
```

Modificación del Kernel

Hay un archivo que puede eliminarse del directorio Kernel/, que tiene que ver con el desarrollo de Broadcom 4330 y que no es necesario para ODROID-U. Elimine el fichero **Android.mk** dentro del directorio **kernel/drivers/net/wireless/bcm4330/src/** como se muestra a continuación.



¡Preparados... listos... Ya...!

Con los pasos anteriores, has terminado la pre-compilación. Dirigite al directorio fuente de Android y siga estos pasos para iniciar la compilación:

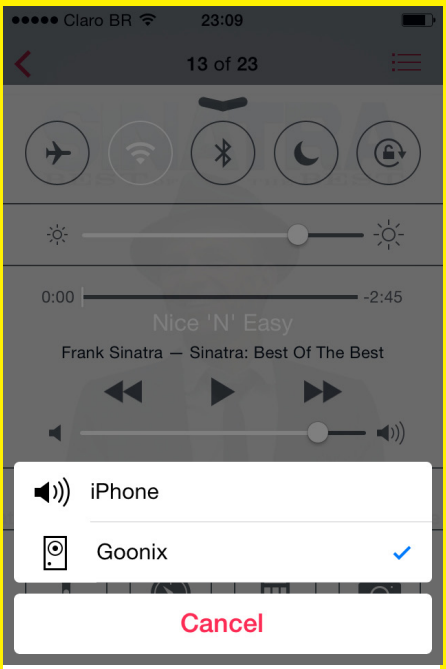
Ejecuta **build/envsetup.sh**. y verás el siguiente resultado.

SHAIRPORT
CONVIERTE TU ODROID EN UNA ESTACION DE SONIDO CON ITUNES

por Bruno Doiche

Por casualidad no dispondrás de un viejo equipo con buenos altavoces que no estás utilizando. Conecta el cable 2-RCA desde tu ODROID a los puertos AUX del equipo y haz lo siguiente en tu terminal Linux:

```
sudo apt-get install\
avahi-utils libmodule-build-perl\
libio-socket-inet6-perl libao-dev\
libssl-dev libcrypt-openssl-rsa-perl\
libwww-perl pkg-config
git clone https://github.com/
njh/perl-net-sdp.git perl-net-sdp
cd perl-net-sdp/
perl Build.PL
sudo ./Build
sudo ./Build test
sudo ./Build install
cd ..
git clone https://github.com/
hendrikw82/shairport.git
cd shairport
make
./shairport.pl -a name
```



Goonix, es un nombre de host bonito para una máquina ... ¡Pero espera! ¿No es el mismo usado enel servidor de medios PS3 del último número? ¡Qué gran versatilidad!

```
including device/asus/grouper/vendorsetup.sh
including device/generic/armv7-a-neon/vendorsetup.sh
including device/generic/armv7-a/vendorsetup.sh
including device/hardkernel/boards/vendorsetup.sh
including device/moto/wingray/vendorsetup.sh
including device/ti/panda/vendorsetup.sh
including sdk/bash_completion/adb.bash
```

Resultado de consola de envsetup.sh

Ejecute luch y veras una lista como la que se muestra a continuación. Seleccione la opción 7.

```
You're building on Linux

Lunch menu... pick a combo:
     1. full-eng
     2. full_x86-eng
     3. vbox_x86-eng
     4. full_grouper-userdebug
     5. mini_armv7a_neon-userdebug
     6. mini_armv7a-userdebug
     7. odroidu-eng
     8. full_wingray-userdebug
     9. full_panda-userdebug

Which would you like? [full-eng]
```

No necesitas pensar si vas a tener suerte con la opción 7, de hecho es la opción correcta en este caso.

Avisa a ccache del directorio que vas a usar para almacenar el archivo compilado, y también de la variable de entorno para informar del proceso de compilación que deseamos para usar ccache.

```
export USE_CCACHE=1
export CCACHE_DIR=<your_cca-
che_directory>
```

El último paso es ejecutar el proceso de compilación escribiendo make-j4

Aunque he creado el script buildOdroid.sh para poder compilar Android, es bueno que vayas paso por paso para ver como funciona todo el proceso.

Tras completar los pasos anteriores, podrás ver la ejecución del proceso de compilación, que llevará algún tiempo. En mi máquina, con el caché habilitado, tardo unos 18 minutos. En la parte inferior se muestra lo que aparece cuando la compilación finalice con exito.

Ficheros de imagen

Una vez finalizado el proceso, los binarios resultantes y archivos de imagen (. Img) se localizan en el directorio **`out/target/product/odroidu/`** .

```
-rw-rw-r--  1 nanik nanik    167626 Jan 27 21:36 ramdisk.img
-rw-rw-r--  1 nanik nanik   1955584 Jan 27 21:36 ramdisk-recovery.cpio
-rw-rw-r--  1 nanik nanik   1388672 Jan 27 21:36 ramdisk-recovery.img
-rw-rw-r--  1 nanik nanik    167690 Jan 27 21:36 ramdisk-uboot.img
-rw-rw-r--  1 nanik nanik       518 Jan 27 21:31 README.txt
drwxrwxr-x  3 nanik nanik      4096 Jan 27 21:36 recovery/
-rw-r--r--  1 nanik nanik   4526080 Jan 27 21:36 recovery.img
drwxrwxr-x  8 nanik nanik      4096 Jan 27 21:36 root/
drwxrwxr-x  6 nanik nanik      4096 Jan 27 21:36 symbols/
drwxrwxr-x 14 nanik nanik      4096 Jan 27 21:35 system/
-rw-r--r--  1 nanik nanik 234864788 Jan 27 21:46 system.img
drwxrwxr-x  3 nanik nanik      4096 Jan 27 21:33 test/
-rw-r--r--  1 nanik nanik  36463328 Jan 27 21:37 userdata.img
drwxrwxr-x  2 nanik nanik      4096 Jan 27 21:36 utilities/
```

Existen varios archivos .img que necesitarás para tu ODROID-U:

boot.img • es la imagen principal del gestor de arranque que contiene u-boot, kernel y ramdisk

recovery.img • Este archivo se usa para crear la partición de recuperación

system.img • Es la imagen principal que contiene el sistema Android completo y aplicaciones

Limpieza

Si surge algún problema durante la compilación, asegúrate de "limpiar" los directorios con el comando **`make clobber`** antes de reiniciar el proceso. Esto eliminará todos los objetos binarios/compilados que se generan durante la fase de compilación/unión.

```
make_ext4fs -s -l 1073741824 -a system out/target/product/odroidu/obj/PACKAGING/systemimage_intermediates/s
+ make_ext4fs -s -l 1073741824 -a system out/target/product/odroidu/obj/PACKAGING/systemimage_intermediates
Creating filesystem with parameters:
    Size: 1073741824
    Block size: 4096
    Blocks per group: 32768
    Inodes per group: 8192
    Inode size: 256
    Journal blocks: 4096
    Label:
    Blocks: 262144
    Block groups: 8
    Reserved block group size: 63
Created filesystem with 1277/65536 inodes and 61173/262144 blocks
+ '[' 0 -ne 0 ']'
Install system fs image: out/target/product/odroidu/system.img
out/target/product/odroidu/system.img+ maxsize=1096220928 blocksize=4224 total=234864788 reserve=11075328

32:26.38 elapsed,
5206.56 user,
346.32 system,
3510740 memory
0 status
```

Android ha sido creado correctamente ¡Buen trabajo!

BACKUP PORTABLE DE UNA IMAGEN

CREA UN ARCHIVO DE RECUPERACIÓN PARA TU SISTEMA OPERATIVO FAVORITO

por Rob Roy, Editor Jefe

Realizar copias de seguridad te mantendrá a salvo de tu mala suerte, de las mascotas, y especialmente de tu propio orgullo.

Una vez que tengas configurado tu ODROID a tu gusto, es importante conocer como restaurar tu sistema de forma rápida y sencilla. Si te gusta experimentar con Linux o Android, deseas instalar tu sistema operativo en varios ODROIDs o quieres guardar una copia de seguridad en caso de fallo, puedes hacerlo con una imagen de tu tarjeta SD o módulo eMMC. Se trata de una copia exacta con bootloader, kernel, y el sistema principal de ficheros y archivos de usuario.

Para empezar, apague ODROID y retire la eMMC o tarjeta SD a la que deseas realizar una backup. Usa otro equipo con Linux y un adaptador tarjeta SD a USB conectado al puerto USB. Si utilizas un módulo eMMC, conecte el adaptador de tarjeta SD que acompaña su ODROID antes de insertarlo en la ranura de tarjeta SD. Dependiendo del sistema operativo instalado en el ordenador, el procedimiento para realizar la backup en un archivo de imagen puede ser un tanto diferente.

Windows

¡Puff!.. Windows

Hardkernel publica una versión mejorada de Win32 DiskImager que rellena automáticamente el disco con ceros antes de escribir la imagen. Disponible para su descarga gratuita en **http://bit.ly/1lYQ7MF** y es muy fácil de usar. Simplemente selecciona la unidad USB en el menú desplegable, elige el archivo de imagen con el botón de la carpeta, y pulse "Read".

Dependiendo del tamaño de la SD o eMMC, el proceso de backup puede tardar de 15 a 60 minutos. El fichero .img resultante tendá el tamaño exacto del disco que se ha copiado, así que asegúrate en primer lugar de tener suficiente espacio en disco. La copia de seguridad de la imagen debe hacerse sobre una partición NTFS, puesto que DiskImager no es capaz de escribir un archivo de más de 4 GB en una unidad FAT32.

Cuando la imagen se haya completado, podemos comprimir el archivo usando la utilidad xz, que cuenta con un índice de compresión muy alto. Si no tienes instalado xzip, descargue y descomprima la versión para Windows desde **http://tukaani.org/xz/**, Luego, cópiela en el mismo directorio del archivo de backup. Escriba el siguiente comando en el intérprete de comandos de Windows, después de navegar hasta el directorio correcto:

```
xz -z mybackup.img
```

Este paso también te llevará algo de tiempo. Tras finalizar la compresión, un archivo llamado mybackup.img.xz reemplazará el archivo original de .img. Esto puede reducir el archivo hasta un 80%, dependiendo de la cantidad de datos almacenados en el sistema operativo. Haga backup de tus backups en diferentes discos, con el fin de asegurarte de no perder tus valiosos datos.

Cuando desees recuperar la imagen de backup y escribirla en una SD o eMMC, usa de nuevo el comando xz para descomprimir el archivo:

```
xz -dk mybackup.img.xz
```

Esto volverá a crear el archivo .img original invirtiendo el algoritmo de compresión. La opción **-k** conserva el archivo original img.xz, asi que puede usarse más tarde para hacer otra recuperación.

Por último, vuelve a Win32 DiskImager y selecciona el destinatario en la lista

RENOMBRA
TUS ARCHIVOS DE MAYÚSCULA A MINÚSCULA EN UNA LÍNEA DE COMANDOS

por Bruno Doiche

Siempre que necesitas organizar los archivos en tus directorios, tienen un montón de deben ser renombrado para que cumplan con tu estructura ordenada de ficheros. Seguramente, cuando son pocos utilizas el comando mv y lo resuelves. Pero ¿Qué hacer si son más de un centenal?
Usa la siguiente sintaxis en el terminal:

```
for i in *; do mv $i $(echo $i | tr [:upper:] [:lower:]); done
```

¡A qué es facil!

PROTEJERSE
DE LOS ACCIDENTES DE SUPERUSUARIO

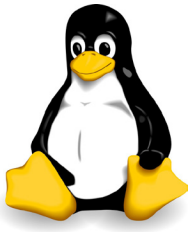
Siempre que editas los archivos del sistema usando tu editor de texto, ¿Puede entrar en modo superusuario usando sudo o su? Elimina el peligroso de exponerte a un borrado accidental de archivos. Salte y reinicia para crear un script que mantendrá tu entorno seguro. Lo llamaremos autosudo.sh

```
#!/bin/bash
   FILE=$1
# Check Write Permission
   if [ -w $FILE ]
then
    /usr/bin/vim $FILE
else
# Sudo If We Dont Have Write Permissions
    sudo /usr/bin/vim $FILE
fi
```

Dale permisos de ejecución con chmod + x, copialo en / bin y luego edítalo como:
`autosudo.sh yourfile_to_edit`

desplegable para escribir la imagen, selecciona el archivo .img con el explorador de archivos, y pulse "Write". El destinatario (SD o eMMC) debe tener igual o mayor tamaño que la imagen original. Tras finalizar el proceso, el disco resultante tendrá una copia exacta del sistema operativo original. Inserte la nueva SD en tu ODROID, enciéndalo y ¡Disfruta!

Linux



Yeah Linux baby!

Con Linux, las copias de seguridad de imagen se realizan en su totalidad desde la línea de comandos. Si xz todavía no está disponible en tu sistema, escriba **`sudo apt-get install xz-utils`** para instalarlo. A continuación, monta la SD o eMMC haciendo doble clic en el icono del escritorio del adaptador USB. Escriba **`df -h`** en la ventana de Terminal y anote el nombre del dispositivo que tendrá el formato **`/dev/sdX`**.

Navega hasta el directorio donde está el archivo de imagen. A continuación, escriba el siguiente comando, sustituyendo el nombre del adaptador USB indicado en el paso anterior con **`/dev/sdX`**:

```
sudo dd if=/dev/sdX bs=1M
of=./mybackup.img
```

Al igual que en Windows, cuando la lectura se haya completado, se puede utilizar xz para comprimir y descomprimir el archivo de imagen:

Compremiendo una imagen usando xz

```
xz -z mybackup.img
```

Decomprimir una imagen cerrada usando xz

```
xz -dk mybackup.img.xz
```

Cuando se escribe la imagen descomprimida a una nueva tarjeta, se usa el mismo comando dd de la operación de lectura con la entrada (if) y la salida (of) invertidas:

```
sudo dd of=/dev/sdX bs=1M
if=./mybackup.img
```

Mac OSX

No solemos hablar de Macs, pero la maquetación de la revista se hace es OSX

El procedimiento para crear una imagen usando OSX es similar a Linux, con tres pequeñas diferencias. En primer lugar, en lugar de utilizar apt-get para instalar xz, descarga el paquete xz-utils desde la misma web mencionada con Windows, asegurándote de seleccionar la versión para OSX (**`http://tukaani.org/xz/`**). Las otras diferencias son que el parámetro (bs) del tamaño del bloque para el comando dd esta en minúscula, y el nombre del adaptador USB se encuentra en formato **`/dev/diskX`**:

Lectura desde el disco original a un fichero de imagen usando OSX

```
sudo dd if=/dev/diskX bs=1m
of=./mybackup.img
```

Escritura desde un fichero de imagen a un nuevo disco usando OSX

```
sudo dd of=/dev/diskX bs=1m
if=./mybackup.img
```

Es una buena idea hacer una copia de seguridad de tu sistema antes de instalar una actualización importante, completar una configuración o instalar un gran conjunto de paquetes de software. Si el disco original se daña, una copia de seguridad de la imagen restaurara el sistema rapidamente, sin necesidad de invertir tiempo para reinstalar y reconfigurar todo el sistema.

Si tus datos son my importantes, también es una buena idea mantener una backup, así como algunas copias en otro lugar por seguridad. ¡Nunca tendrás demasiadas backups!

COMO DESARROLLAR TU PROPIO UBUNTU DESDE CERO

USANDO ROOTFS DE LINARO ¡COMPILA LINUX COMO UN PROFESIONAL!

por Mauro Ribeiro

Pronto, las tiendas de Ubuntu serán un lugar frecuentado por futuros ODROIDian

Un gran ventaja de los sistemas operativos de código abierto como Linux es que tienen la posibilidad de descargar su código fuente y compilarlo por uno mismo. Puede añadir parches, retocar el código y comprobar errores, sin necesidad de esperar a una versión oficial o actualización. La plataforma ODROID puede ejecutar muchos sistemas operativos, aunque algunos no están disponibles como imágenes ARM pre-compiladas. Tomarse la molestia de aprender cómo desarrollar un sistema desde cero le permite descargar sistemas operativos reciente y probarlos. En este ejemplo, usaremos la versión de Linaro de Ubuntu para demostrar lo fácil que es hacerse con el control de un sistema operativo en el nivel más básico.

Notas Generales

• Esta guía fue probada en un equipo con Ubuntu 13.10 de 64 bits con las librerias ia32 instaladas.

• Libera al menos 10 GB de espacio en disco en tu equipo.

•Dedica algo de tiempo libre.

•Si algo sale mal, empieza de nuevo.

• El usuario y contraseña por defecto es "linaro".

• TTY1 se conecta como root.

Cocfigurar el entorno

```
cd ~
mkdir ubuntu-guide
cd ubuntu-guide
export GUIDE=`pwd`
export SDCARD=/dev/sdX
```

Asegúrete de reemplazar la X por la letra correcta de tu tarjeta SD.

Descargar todos los componentes necesarios

• Gestor de arranque pre-compilado

Este artículo no cubre el desarrollo del gestor de arranque, puesto que no cambia nada usando el gestor de arranque pre-compilado disponible en la web de Hardkernel.

```
wget odroid.in/guides/
ubuntu-lfs/boot.tar.gz
```

• Código fuente del Kernel

```
git clone --depth 0 https://
github.com/hardkernel/linux.
git -b odroid-3.8.y odroid-
3.8.y
```

• Herramientas para Crossbuild

En esta guía, estoy usando GCC 4.7.2 de Archlinux ARM como Kit de herramientas. Este conjunto de herramientas ofrece una gran estabilidad.

```
wget odroid.in/guides/ubun-
tu-lfs/arm-unknown-linux-
gnueabi.tar.xz
```

• Rootfs de Linaro

Uso rootfs de Linaro porque viene empaquetado como archivo tgz y es perfecto para esta guía. Cuando escribí estas líneas, la versión 13.12 de Linaro era la que estaba disponible aunque cualquier otra debe funcionar.

```
wget http://releases.li-
naro.org/13.12/ubuntu/
arndale/linaro-saucy-serv-
er-20131216-586.tar.gz
```

5. Herramientas U-Boot

Las Herramientas U-Boot vienen con una utilidad denominada mkimage que necesitamos para crear boot.scr.

```
sudo apt-get install u-boot-
tools
```

Desarrollar y montar la Imagen

I. Vaciar la tarjeta.

Siempre suelo hacer esto, puesto que es mejor partir de una base limpia.

```
sudo dd if=/dev/zero
of=$SDCARD bs=1M
```

2. Instalar Gestores de Descarga

```
tar zxvf boot.tar.xz
cd boot
chmod +x sd_fusing.sh
sudo ./sd_fusing.sh $SDCARD
cd ..
```

3. Crear Particiones

Utilizamos dos particiones, una para el kernel + initrd (si fuese utilizado) y otra para rootfs. La partición kernel + initrd es del tipo FAT32 y rootfs es una partición ext4 sin registro de datos.

También es importante que la primera partición comience en el sector 3072 en adelante, ya que hasta éste el espacio es usado por gestor de arranque.

```
sudo fdisk $SDCARD
n
p
1
3072
+64M
n
p
2
134114
<just press enter here>
t
1
c
w
```

Puede parecer criptogafía, pero es muy simple:

n = nuevo
p = partición
1 es el número de la partición que estamos creando
3072 es el punto de partida de la partición
+64M es el tamaño de esta partición. Esta es la partición FAT32, asi que no tiene que ser muy grande.
n crea una nueva partición
134114 es el inicio de la partición 2, que se encuentra justamente después de la partición 1
No indicamos el tamaño a fdisk, así puedes usar el resto de la SD
t = tipo
1 es el número de la partición
c tipo para la partición FAT32
w = escribir

Ejecuta partprobe para ver las nuevas particiones reconocidas por el kernel:

```
sudo partprobe
```

4. Formatea y monta las particiones

Formatea las particiones y cambiar el UUID por el mismo UUID que utilizamos en el Ubuntu Oficial, de esta forma más adelante podras utilizar el script para actualizar el Kernel:

```
mkfs.vfat -n boot $SDCARD"1"
mkfs.ext4 -L rootfs \
$SDCARD"2"
```

Ahora que las particiones están formateadas, vamos a cambiar el UUID de la partición ext4.

```
tune2fs $SDCARD"2" -U
e139ce78-9841-40fe-8823-
96a304a09859
```

Y desactiva el registro de datos para evitar un excesivo desgaste de la tarjeta:

```
tune2fs -O ^has_journal
$SDCARD"2"
```

Luego, monta las particiones:

```
mkdir rootfs
mkdir boot
sudo mount $SDCARD"1" boot
sudo mount $SDCARD"2" rootfs
```

5. Instalar rootfs en nuestra tarjeta SD

Descomprimir rootfs y copiarlo a la tarjeta es muy simple:

```
sudo tar -zxf linaro-saucy-
server-20131216-586.tar.gz
sudo mv binary/* rootfs
```

6. Compilar el kernel

Esta también es una guía para compilar de forma cruzada el kernel

Primero, descomprime herramientas:

```
tar -Jxf arm-unknown-linux-
gnueabi.tar.xz
```

Ya tenemos el código fuente del kernel que descargamos con anterioridad.

```
cd odroid-3.8.y
export ARCH=arm
export CROSS_COMPILE=../arm-
unknown-linux-gnueabi/bin/
arm-unknown-linux-gnueabi-
make odroidu2_defconfig
```

¡Espera! ¿Ves la última línea **`make odroidu2_defconfig`**? Esta línea es para el U2 y U3, Si estas usando el X2, simplemente reemplazarla por **`make odroidx2_defconfig.`**

Compilar el Kernel llevará un tiempo, dependiendo de tu máquina.

```
make -j8
```

Uso **-j8** porque mi equipo es un quad-core con *hyperthreading*, (8 hilos de ejecución). Debes cambiar el número para que coincida con el procesador de tu ordenador.

7. Instala el kernel y los módulos que hemos desarrollado

Primero, instala la imagen del kernel.

```
sudo cp arch/arm/boot/zImage
../boot
```

A continuación, instala los módulos:

```
sudo make ARCH=arm INSTALL_
MOD_PATH=../rootfs modules_
install
cd ..
```

Una vez instalados los módulos, el kernel está listo.

8. Crea un script de arranque para el primer inicio

```
cd boot
cat << __EOF__ | sudo tee
boot.txt
setenvinitrd_high"0xffffffff"
setenv fdt_high "0xffffffff"
setenv bootcmd "fatload mmc
0:1 0x40008000 zImage; bootm
0x40008000"
setenv boot-
args "console=tty1
console=ttySAC1,115200n8
root=/dev/mmcblk0p2 rootwait
rw mem=2047M"
boot
__EOF__
```

```
sudo mkimage -A arm -T
script -C none -n boot -d ./
boot.txt boot.scr
cd ..
```

Esto crea el archivo boot.txt y la línea **sudo mkimage** crea boot.scr.

9. Desmonta y haga limpieza

```
sudo umount boot
sudo umount rootfs
sync
```

Primer Arranque y Configuraciones

Ahora, estamos listos para hacer nuestro primer arranque. Quite la tarjeta del ordenador y conectarla a su placa.

1. Configura la tarjeta de red

```
cd /etc/network/interfaces.d
cat << __EOF__ >> eth0
auto eth0
iface eth0 inet dhcp
__EOF__
reboot
```

2. Configura FSTAB

```
mount -t devtmpfs devtmpfs /
dev
cat << __EOF__ >> /etc/fstab
UUID=e139ce78-9841-40fe-
8823-96a304a09859 / ext4
errors=remount-ro,noatime 0
1
/dev/mmcblk0p1 /media/boot
vfat defaults,rw,owner,flush
,umask=000 0 0
tmpfs /tmp tmpfs
nodev,nosuid,mode=1777 0 0
__EOF__
```

```
mkdir -p /media/boot
mount /media/boot
```

3. Ejecuta el script de actualización

```
apt-get install u-boot-tools
wget builder.mdrjr.net/
tools/kernel-update.sh
chmod +x kernel-update.sh
./kernel-update.sh
```

Ejecutar este paso es importante para crear un ulnitrd y añadir todos los archivos boot.scr para los diferentes monitores y resoluciones.

Todo lo que aparece a continuación tiene que ver con el uso de Linux y puede localizarse en Google o en los Foros Linux. Está dirigido a aquellos que desean un entorno gráfico.

```
Install xubuntu-desktop
```

Antes de iniciar la descarga, asegúrate de tener al menos 450 MB de espacio disponible en disco.

```
sudo apt-get install xubun-
tu-desktop
```

1. Instala los drivers de Mali

```
cd ~
mkdir mali
cd mali
```

2. Descarga las dependencias de Mali

```
wget http://builder.mdrjr.
net/tools/mali.txz
wget http://malideveloper.
arm.com/downloads/drivers/
DX910/r3p2-01rel4/DX910-SW-
99003-r3p2-01rel4.tgz
apt-get build-dep xserver-
xorg-video-armsoc
apt-get install mesa-utils
mesa-utils-extra libgles2-
mesa-dev libgles2-mesa
libgles1-mesa-dev libgles1-
mesa libegl1-mesa libegl1-
mesa-dev
```

3. Instala cabeceras y Blobs

```
tar zxf DX910-SW-99003-r3p2-
01rel4.tgz
tar Jxf mali.txz
mv /usr/lib/arm-linux-gnue-
abihf/mesa-egl ~
cp -aR blobs/* /usr/lib
cp -aR include/* /usr/in-
clude
ldconfig
```

4. Compila e instala el Driver XII

```
cd DX910-SW-99003-r3p2-01-
rel4/x11/xf86-video-mali-0-
.0.1
./autogen.sh
cd src
rm -rf compat-api.h
wget http://cgit.freedesk-
top.org/~cooperyuan/compat-
api/plain/compat-api.h
cd ..
make -j4
make install
mv /usr/local/lib/xorg/mod-
ules/drivers/mali* /usr/lib/
xorg/modules/drivers
```

5. Configura Xorg.conf para usar Mali

```
cat << __EOF__ >> /etc/X11/
xorg.conf
Section "Device"
 Identifier "Mali-Fbdev"
 Driver "mali"
 Option "fbdev"            "/
dev/fb1"
 Option "DRI2"
"true"
 Option "DRI2_PAGE_FLIP"
"true"
 Option "DRI2_WAIT_VSYNC"
"true"
 Option "UMP_CACHED"
"true"
 Option "UMP_LOCK"
"false"
EndSection

Section "Screen"
 Identifier "Mali-Screen"
 Device "Mali-Fbdev"
 DefaultDepth 24
EndSection

Section "DRI"
 Mode 0666
EndSection
__EOF__
```

6. Crear una regla udev para cambiar los permisos de malí y que un usuario normal pueda usarlo

```
cat << __EOF__ >> /etc/udev/
rules.d/10-mali.rules
KERNEL=="mali",SUBSYSTEM=="m
isc",MODE="0777"
KERNEL=="ump",SUBSYSTEM=="um
p",MODE="0777"
__EOF__
```

¡Felicidades, Lo has conseguido!

COMO INSTALAR LA VERSIÓN 8 DEL KIT DE DESARROLLO DE JAVA (JDK) DE ORACLE

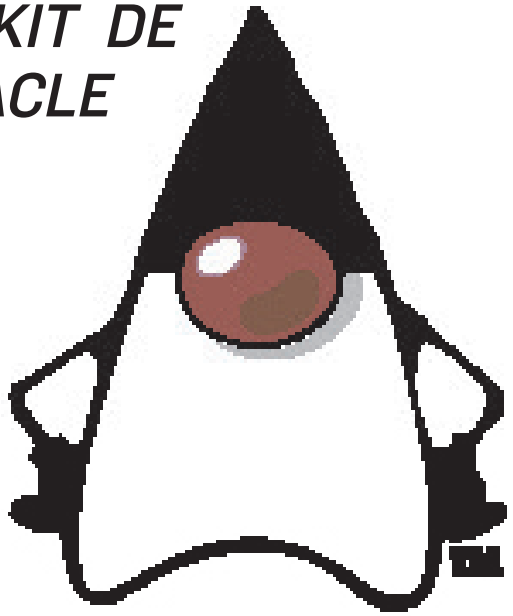
AHORRA TIEMPO CON LA ARQUITECTURA "DE CODIGO UNICO Y MULTIPLATAFORMA" DE JAVA

por Robert Raehm, Editado por Venkat Bommakanti

Java es uno de los lenguajes de programación más populares, tanto para aplicaciones como para desarrollo web. Tiene la ventaja de ser multiplataforma, lo que significa que un código escrito en Java se puede ejecutar en cualquier máquina virtual de Java independientemente del procesador, el ordenador, el sistema operativo o el hardware. Oracle publica un kit de desarrollo libre, que también está disponible para ARMHF, lo que significa que la familia ODROID puede ejecutar la amplia librería de software Java. La última versión disponible de abril 2014 es la JDK8, que puede ser instalada junto a las anteriores versiones de Java proporcionando una completa plataforma para el desarrollo, con mejoras significativas en la velocidad con respecto a versiones anteriores.

Requisitos

- **Un ODROID de las series X, U o XU**
- **Una eMMC o Micro SD Clase 10 de al menos 8GB**
- **Ubuntu, Debian o imagen similar (13.04 o superior), disponible en los foros ODROID (http://forum.odroid.com/)**



Aunque este ejemplar detalla diversos proyectos "Hazlo tú mismo", también podría llamarse la Edición de la "Mascota Graciosa"

Descargar Archivos

Para empezar, haz una copia de seguridad de tus archivos personales. En el escritorio de Ubuntu, crea una carpeta para almacenar los paquetes que descargaremos.

Los ejemplos de este artículo usan la versión de Oracle JDK del 13 de marzo de 2014. Puede descargar la versión más reciente desde **https://jdk8.java.net/download.html** haciendo clic en el enlace del paquete correspondiente a Linux ARM. En el momento de escribir esto, la última versión disponible era **jdk-8-fcs-b132-linux-arm-vfp-hflt-03_mar_2014.tar.gz**.

Después de aceptar los Términos y Condiciones y descargar el archivo, es una práctica habitual verificar el paquete para asegurar que se copio correctamente. Esto se hace mediante la utilidad md5sum:

```
$ md5sum jdk-8-fcs-b132-linux-arm-vfp-hflt-03_mar_2014.tar.gz
```

El resultado debe ser comparado con el contenido del archivo de verificación ubicado en el mismo directorio del paquete descargado. Para este ejemplo, el archivo md5sum se encuentra en **http://www.java.net/download/jdk8/archive/b132/binaries/jdk-8-fcs-b132-linux-arm-vfp-hflt-03_mar_2014.md5**.

Mi archivo descargado tenía el esquema de verificación c17b5194214b8ea9ad8e6fc302fe078. Si el archivo que has descargado tiene un esquema de diferente al que se encuentra en el servidor, descartarlo, reiniciar la descarga y realiza la verificación de nuevo.

Extraer el Archivo

En la ventana de terminal, cambia de directorio (cd) a la carpeta de descarga designada y descomprime el archivo:

```
$ tar -zxvf jdk-8-fcs-b132-linux-arm-vfp-hflt-03_mar_2014.tar.gz
```

Se creará un nuevo subdirectorio llamado jdk1.8.0 en el directorio de descarga.

Montar la instalación de Java

En los sistemas Linux, Java se instala normalmente en el directorio de sistema **/usr/lib/jvm** cuando se utiliza un instalador automático. Sin embargo y ya que estamos instalando manualmente el paquete, los archivos descomprimidos tendrán que ser movidos a la carpeta correcta desde la ventana de Terminal.

```
$ sudo mv jdk1.8.0 \
/usr/lib/jvm
```

Actualizar la variable de entorno PATH

Tu instalación de Linux puede tener una versión del kit de desarrollo de Java, y la ubicación de esta instalación estará probablemente especificada en la variable de entorno PATH. La variable PATH especifica determinados directorios para la búsqueda cuando un comando se inserta en la ventana de terminal, de modo que los paquetes pueden ser activados desde cualquier directorio.

Después de instalar JDK 1.8.0 siguiendo los pasos anteriores, tenemos que asegurarnos que la versión 1.8 se usa como máquina virtual a partir de ahora. Para ello, actualiza la variable de entorno PATH para incluir la nueva versión:

```
$ export PATH=/usr/lib/jvm/jdk1.8.0/bin:$PATH
```

Al final del comando $ PATH añade la variable de entorno PATH actual a una nueva. Puesto que la cadena $ PATH busca la primera coincidencia de un programa, una vez que ésta es localizada, el sistema ignora el resto de la cadena $PATH. De esta manera se evita cualquier instalación de Java anterior, que también pueden ser incluida.

Completar la instalación

Por lo general, cuando se instalan programas en Linux usando utilidades de instalación, se crean determinados accesos directos. Necesitamos actualizar manualmente estos accesos usando los siguientes 4 comandos:

```
sudo update-alternatives
--install /usr/bin/javac\
javac /usr/lib/jvm/jdk1.8.0/
bin/javac 1

sudo update-alternatives
--install /usr/bin/java\
java /usr/lib/jvm/jdk1.8.0/
bin/java 1

sudo update-alternatives
--config javac

sudo update-alternatives
--config java
```

Verificar la instalación

Como paso final, tenemos que asegurarnos de que JDK8 se ha instalado correctamente, y que se están utilizando los componentes adecuados. Para ello, ejecute java usando el parámetro de versión para mostrar la versión actual:

```
$ java -version
```

El resultado debe ser similar a este, indica que JDK8 es el valor por defecto:

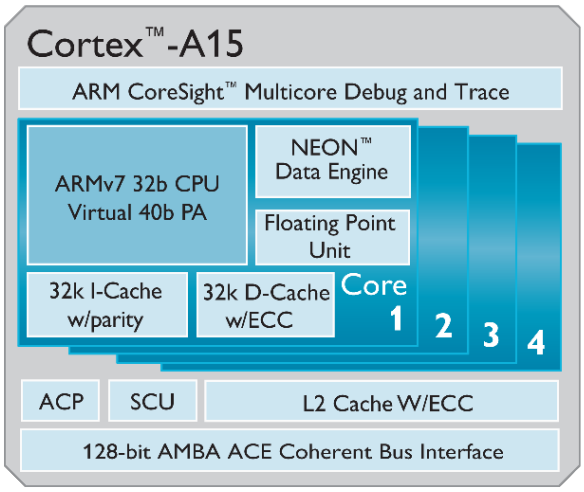
```
java version "1.8.0"
Java(TM) SE Runtime Environment (build 1.8.0-b132)
Java HotSpot(TM) 32-Bit Server VM (build 25.0-b70, mixed mode)
```

Para más información o resolver dudas, visita el hilo oficial del foro, en **http://forum.odroid.com/viewtopic.php?f=52&t=204**.

USAR ODROIDS EN LA INFORMATICA DE ALTO RENDIMIENTO

QUE DIFERENCIA A UN KERNEL COMPILADO

por Kurt Keville, MIT



A15 está diseñado con técnicas avanzadas de reducción de energía, y alimenta nuestro producto extrena XU, con el fin de sacarle el mejor partido!

Hemos comparado diferentes kernels y sus respectivos rendimientos sobre XU (ver **http://tinyurl.com/XUBench1** y **http://tinyurl.com/XUBench2**). Fue interesante ver las diferencias que se obtuvieron entre los kernels de las imágenes Whisper y Particle de Rob Roy (3.4.67 fue la última actualización que hicimos antes de realizar la evaluación) y el kernel 3.13 de la distribución en pruebas de Linaro 14.02

Para quienes trabajan con arquitecturas Calxeda Highbank o Midway, contempladas en las URL *openbenchmarking* indicadas anteriormente y en **http://tinyurl.com/ApacheOnARM** no les sorprenderá el rendimiento del Quad-Core ARM Cortex-A9 al envíar páginas vía httpd. De hecho, la mayoría de los ISP tradicionales no necesitan realizar demasiados cálculos para ofrecer una gran cantidad de páginas web, de modo que un procesador de clase A9 con extensión NEON inferior a la del Cortex-A15 debería funcionar sin problemas.

Es interesante ver cómo el XU supera al quad-core A9 usando como punto de referencia Apache, y también sorprende que el kernel 3.13 sea mucho mejor que el kernel 3.4 sobre el Exynos 5410 usando el mismo banco de pruebas. El XU gana la "carrera", probablemente porque los núcleos A15 se usaros al completo y los núcleos A7 al minimo, obteniéndose un consumo ideal frente a los índices de rendimiento en esta evalución.

Los futuros clúster ARM como Spinnaker tendrán cientos de miles de núcleos, de modo que cada pequeña mejora de eficiencia energética será muy importante. Muchas de las mejoras de rendimiento también conllevan importantes reducciones de consumo. Por ejemplo, si eliminas parte de los medios locales como tarjetas SD o discos SATA, puedes utilizar diversos trucos relacionados con tftp o arranque PXE y unidades Ram para agilizar las operaciones y así reducir los dispositivos a encender. Netbooting y NFSroot son las mejores técnicas de reducción de consumo eléctrico.

La eficiencia energética en Datacenter con ODROID puede agilizarse mediante simples kernel y realizando cambios en el espacio de usuario. No representan una gran diferencia por separado, pero si en conjunto. Estos son algunos ejemplos rentables:

La operación no ejecutada es la más eficiente en consumo energético.

En el código de aplicación puedes aprovechar al máximo las posibilidades de tu chip. Usando "fused multiply-add" conseguirás 2 operaciones por los mismos ciclos de reloj que ejecutando esas operaciones por separado.

Implementar modificaciones de HPC a nivel de gestión / usuario.

Nos referimos a seguir una secuencia en las tareas a ejecutar. Si usas algo como PowerNap o PowerWake, puedes ahorrar bastante energia durante la vida útil del equipo. Esta funcionalidad fue descrita en mi artículo del número 2 (febrero de 2014) de ODROID Magazine.

Agrupar y maximizar las cosas que se prestan a la consolidación y distribución con el fin de aprovechar arquitecturas híbridas.

Pon tus directorios de datos en recursos compartidos NFS, así no necesitas registrar diariamente el sistemas de archivos o controlar tus directorios (de sólo lectura) en los nodos cliente. Esto ahorra tiempo y energía.

Busca el modo de utilizar eficazmente los ciclos de cálculo inactivos

Usamos una herramienta para calcular la comunicación ideal frente al solapamiento de cálculo para coger la cantidad de datos adecuada para una

operación, de tal forma que nunca se llegase a una situación de escasez de datos o carencia de CPU. Si no se puede evitar un entorno de carencia de datos podemos ir a un estado de baja energía ACPI hasta llegar a la potencia anterior mientras esperamos a que la transferencia se complete.

Compilar código a nivel local para maximizar el uso de recursos.

El paquete GCC 4.8 en XU ofrece el mejor y más pequeño sistema binario.

Use el método más eficiente numéricamente

Una vez más, tiene que ver con el código de aplicación. A menudo se pueden representar los números de coma flotante en varios niveles de precisión.

Prestar a los grandes problemas la atención que merecen, pero también resolver todos los problemas menores.

Hay unos cuentos cambios menores que recomendamos realizar. Existe una importante flexibilidad para decidir que recursos puedes poner en práctica dando prioridad a las aplicaciones de producción, y qué puedes desactivar en el kernel con poco o ningún efecto negativo.

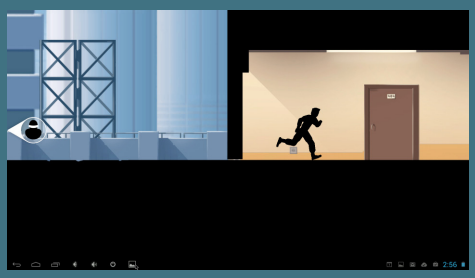
Conclusión

Para obtener el mejor rendimiento de tu ODROID, activa exactamente lo que deseas y apagar todo lo demás. Asegúrate de que estás justamente ejecutando la única aplicación que deseas (en nuestro caso, una prueba de rendimiento). No puedes usar el 100% de tu procesador en tu aplicación de producción si está ocupado respondiendo interrupciones, así que cancela (o no inicies) procesos innecesarios.

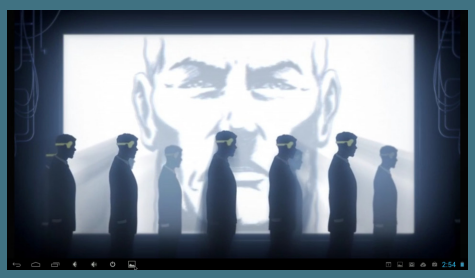
Puedes bajar al modo monousuario (init 2) para estar seguro de que no estás perdiendo recursos por aplicaciones no deseadas, incluyendo cualquier cosa que no seas apagar el Kernel, como el USB y el video. Hay algunos consejos y trucos adicionales, como la tecnología tickless descrita en `http://tinyurl.com/XULessWatts`. ¡Disfruta del viaje!

VECTOR
REPLETO DE ACCION PARKOUR

por Ronaldo Andrade



Vector es un emocionante juego al estilo árcade en el que un corredor excepcional resulta difícil mantener bajo control. El juego comienza con la visión de un mundo totalitario donde la libertad y autonomía no es más que un sueño lejano. Pero el corazón de un corredor es fuerte y pronto conseguirá escapar. Corre, salta, deslízate y trepa usando técnicas extraordinarias basadas en el deporte del ninja urbano Parkour mientras eres perseguido por el "Hermano Mayor", cuyo único propósito es capturarte y traerte de vuelta.



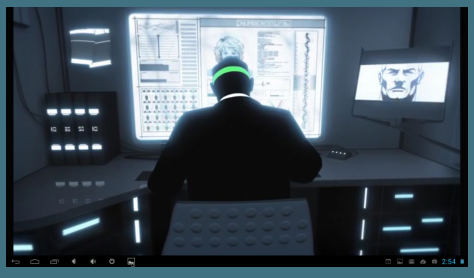
Inspirado en la práctica y los principios de Parkour, los controles intuitivos de Vector se adaptan a jugadores de todos los niveles. Los diseños completojs de niveles desafían a los jugadores más exigentes con rompecabezas cronometrados como los "movimientos de los traceur" en las azoteas

Es un impresionante juego de la empresa de desarrollo rusa Nekki. Si alguna vez observas el Parkour y lo encuentras interesante, te encantará este juego. La acción dinámica y los controles simples hacen de éste un juego muy divertido. Pero que no te engañes, el juego presenta grandes desafíos. Hay tres escenarios diferentes a los que se puedes jugar en la versión completa, con las

construcciones mas imponentes y desafiantes que puedas imaginar.

El objetivo principal es escapar de los guardias que te persiguen. Para conseguir las tres estrellas, tendrás que recoger los cubos completos y realizar todos los trucos. También hay algo de dinero extra en los niveles, pero no es necesario para obtener las 3 estrellas.

En ODROID, puede utilizar el teclado y el ratón para controlar el jugador, un joystick también vale.



CONSEJOS GENERALES:

Recoge todos los cubos y realizar todos los trucos. Esto te dará tres estrellas al finalizar el nivel.

Anticípate, comprueba el escenario para ver qué truco necesitas para rebasar el obstáculo con el menor esfuerzo, consumiendo menos tiempo.

En la tienda, puedes usar el dinero que recibes para comprar artilugios que pueden ser útiles en algunas situaciones.

COMO CONFIGURAR UN SERVIDOR MINECRAFT

¡ENREDATE!

por @qkpham

Editado por Venkat Bommakanti

A casi todo el mundo le gusta jugar a los juegos, ¡Especialmente a Minecraft! Lo han disfrutado más de 14 millones de personas en todo el mundo, por su jugabilidad adictiva y mapas personalizables. Aunque la versión oficial de Mojang Software es de código cerrado, hay varias versiones java de código abierto de Minecraft Server para ODROID. Programar un mundo virtual usando un paquete gratuito de servidor Minecraft como Spigot, Bukkit o BungeeCord es una buena forma de aprender Java mientras te diviertes.

Este artículo describe cómo instalar un servidor básico de Minecraft en tu ODROID, de modo que puedes jugar online con tus amigos en un mundo creado por ti mismo. Usar ODROID como un entorno de pruebas es también una buena manera de probar mapas, actualizaciones y modificaciones antes de subirlas a un servidor público.

Requisitos

1. Un ODROID de la serie X, U o XU
2. Una eMMC o MicroSD clase 10 de 8+ GB
3. Ubuntu, Debian o imagen similar (13.04 o superior), disponible en los foros ODROID (**http://forum.odroid.com/**)
4. Java versión 1.8 (OpenJDK8 o Oracle JDK8)
5. Conexión de red de área local (LAN), incluyendo un router con redireccionamiento de puertos.

Instalar Java

Si la versión de Java 1.8 no está instalada en tu sistema, consulte el artículo de este número de ODROID Magazine donde se detalla como Instalar Oracle JDK8". Mojang publica una versión en Java del software Minecraft compatible con otros sistemas operativos como Linux ARM.

Instalar Minecraft

En primer lugar, descargue el software más reciente Minecraft Server desde el sitio oficial en **https://minecraft.net/download**, asegurate de elegir la versión basada en Java.

Cree un directorio minecraft en tu directorio principal para almacenar minecraft_server.jar. Una vez que el archivo este descargado, escriba los siguientes comandos para iniciar el servidor:

```
$ cd ~/minecraft
$ java -Xms1536M -Xmx1536M
-jar minecraft_server.jar
nogui
```

El servidor de Minecraft debería estar funcionando. El ultimo paso es obtener la dirección IP del servidor para que nuestros jugadores puedan conectarse a éste a través de sus clientes Minecraft.

Cuando escuchas a las enredaderas hacer el ruido Sssss ... , sólo puedes hacer una cosa:

CORRER!

Obtener la dirección IP interna

Localiza la dirección IP (local) interna de tu servidor escribiendo **ifconfig** en la ventana de Terminal y busca la etiqueta **inet addr.** En mi ODROID, la dirección IP aparece como 192.168.1.10. Asegúrate que esta dirección facilitada por el servidor DHCP local o router no es modificada con el fin de evitar realizar cambios de configuración frecuentes.

Configurar la Reedirección de puertos

Minecraft utiliza el puerto TCP 25565, el cual debe ser redireccionado a la dirección IP del servidor por tu router. Consulte el manual del usuario para obtener ayuda de como configurar el router para redirigir el puerto 25565 a la dirección IP obtenida en el paso anterior.

Obtener la dirección IP externa

La dirección IP pública que identifica tu LAN en el mundo exterior se puede conocer visitando **http://www.whatismyip.com**. La dirección tendrá el formato **aaa.bbb.ccc.ddd**, lo que significa que la dirección URL completa para conectar al servidor Minecraft en tu LAN sería **http://aaa.bbb.ccc.ddd:25565**. Es muy importante añadir el puerto TCP al final de la URL

Si tu IP externa es dinámica (es modificada periódicamente por tu ISP) puedes utilizar servicios como No-IP. Puedes crear una cuenta en su sitio web, descargar e instalar el cliente de actualización de DNS dinámico (DUC) **http://www.noip.com/download**. Localiza las instrucciones para configurar el DNS dinámico en **http://bit.ly/1ggmo2n**. En este caso, la dirección completa del servidor Minecraft sería **http://youracctusername.no-ip.com:25565**.

Para asegurarte que todo funciona correctamente, puedes probar que tu servidor este visible en **http://www.canyouseeme.org**. También puede comprobar su estado en **http://dinnerbone.com/minecraft/tools/status/**.

El rendimiento del sistema es aceptable en una red inalámbrica, pero una conexión por cable disminuirá la latencia y aumentar la capacidad de respuesta del juego.

Unirse al juego

Inicie el cliente de Minecraft en una máquina con Windows o OSX introduciendo la dirección IP pública del paso anterior (**http://aaa.bbb.ccc.ddd:25565**) En el momento de crear este tutorial, el software cliente de Minecraft lamentablemente todavía no estaba disponible para la plataforma ODROID. Hay una edición Minecraft Pocket disponible para Android, pero no es compatible con la versión completa de Minecraft Server.

Una adecuada conexión del servidor Minecraft ODROID atraerá al usuario a nuestro mundo virtual como muestra la imagen de arriba.

Configuración Adicional del servidor

Las opciones de servidor de Minecraft se configuran editando el archivo **server.properties** ubicado en **/home/yourusername/minecraft/server.properties**:

```
#Minecraft server properties
#Mon Dec 24 09:23:18 EST
2012
#
generator-settings=
level-name=world
enable-query=false
allow-flight=false
server-port=25565
level-type=DEFAULT
enable-rcon=false
level-seed=
server-ip=
max-build-height=256
spawn-npcs=true
white-list=false
spawn-animals=true
hardcore=false
texture-pack=
online-mode=true
pvp=true
difficulty=1
gamemode=0
max-players=20
spawn-monsters=true
generate-structures=true
view-distance=10
motd=A Minecraft Server
```

Las tres configuraciones útiles para cambiar mapas y mejorar el rendimiento son:

level-name

Si deseas agregar otro mapa o mundo a tu servidor, descomprime el archivo del mapa dentro de la carpeta de Minecraft y luego cambia la configuración level-name al nombre de esta carpeta. Por ejemplo, si la carpeta extraída es **odroid**, cambie el valor del level-name a **odroid** en lugar del valor predeterminado **world.**

view-distance

Reducela a 7 para mejorar la capacidad de respuesta del servidor

max-players

Funciona mejor cuando se fija entre el 2 y 5.

DESCARGAR VIDEOS DE YOUTUBE PARA VERLOS SIN CONEXION

por Bruno Doiche

Aunque vivimos en un mundo conectado, en ocasiones, tenemos que ir a lugares donde no existe ningún tipo de conexión. Pues bien, no es mala idea disponer de un kit de supervivencia con tu contenido favorito de youtube con youtube-dl!

Para instalarlo, sólo tienes que escribir lo siguiente en el terminal:

```
sudo pip install --upgrade youtube_dl
```

Ahora se puede descargar cualquier vídeo que desees desde youtube:

```
youtube_dl <youtubevideo_url>
```

¿Qué dices? que solo quieres la musica de los videos y el audio desde los podcasts y ¿quieres ahorrar espacio?

Ok, vamos a crear un sctrpt para solucuonar esto:

```
echo "ffmpeg -i $1 -acodec libmp3lame -ac 2 -ab 128 -vn -y $2" > mp3zator.sh
```

Conviertelo en ejecutable con:

```
chmod + X mp3zator.sh
```

y ejecutalo así::

mp3zator <tu_video_.mp4> <tu_audio.mp3>

Muy bien!!! Consigues todo lo que necesitas y has perdidio el miedo a no tener tus películas, videos y música favoritos al mismo tiempo que desarrollar código en algún lugar muy muy lejano.

Minecraft depende principalmente de operaciones en coma flotante. A diferencia de las CPUs basadas en arquitectura x86, los SOCs ARM no están optimizados para operaciones de coma flotante, de modo que la configuración del servidor debe ajustarse hasta compensar la carga de trabajo.

Si deseas mejorar aun más el rendimiento, existen varias versiones de código abierto de Minecraft Server, que disminuyen significativamente la carga de trabajo del servidor, proporcionando una mejor experiencia y permitiendo que puedan unirse mas jugadores.

Craftbukkit

Crea una carpeta para Craftbukkit escribiendo **mkdir ~/craftbukkit** en una ventana de terminal. Luego, visite **https://dl.bukkit.org/downloads/craftbukkit/** para descargar la última versión de Craftbukkit al directorio recién creado. Tras completar la descarga, ejecuta el servidor para crear tu mundo.

```
java -Xms1536M -Xmx1536M
-jar craftbukkit.jar
cd ~/craftbukkit/plugins
wget http://dev.bukkit.org/
media/files/674/323/NoLagg.
jar
wget http://dev.bukkit.org/
media/files/665/783/PTweaks.
jar
wget http://dev.bukkit.org/
media/files/586/974/NoSpawnC-
hunks.jar
```

Spigot

Una alternativa a Craftbukkit es Spigot, ofrece más opciones de configuración y está optimizado para un mejor rendimiento y velocidad. Siguiendo el mismo procedimiento que en el caso anterior, descarga del paquete de Spigot desde **http://www.spigotmc.org/**.

```
mkdir ~/spigot
cd spigot
wget http://ci.md-5.net/job/
Spigot/lastSuccessfulBuild/
artifact/Spigot/target/spig-
ot.jar
java -Xms1536M -Xmx1536M
-jar spigot.jar
```

Spigot es muy estable, y puesto que está basado en Craftbukkit, los pluging Bukkit NoLagg, PTweaks y NoSpawnChunks también funcionan con Spigot.

MineOS

MineOS es un panel de administración Web que ofrece una fácil gestión de servidores Minecraft. Puede gestionar Vainilla, Bukkit, Tekkit y Canary por defecto, pero se puede instalar cualquier otro sistema de servidor y configurarlo para que descargue automáticamente una nueva versión cuando esté disponible.

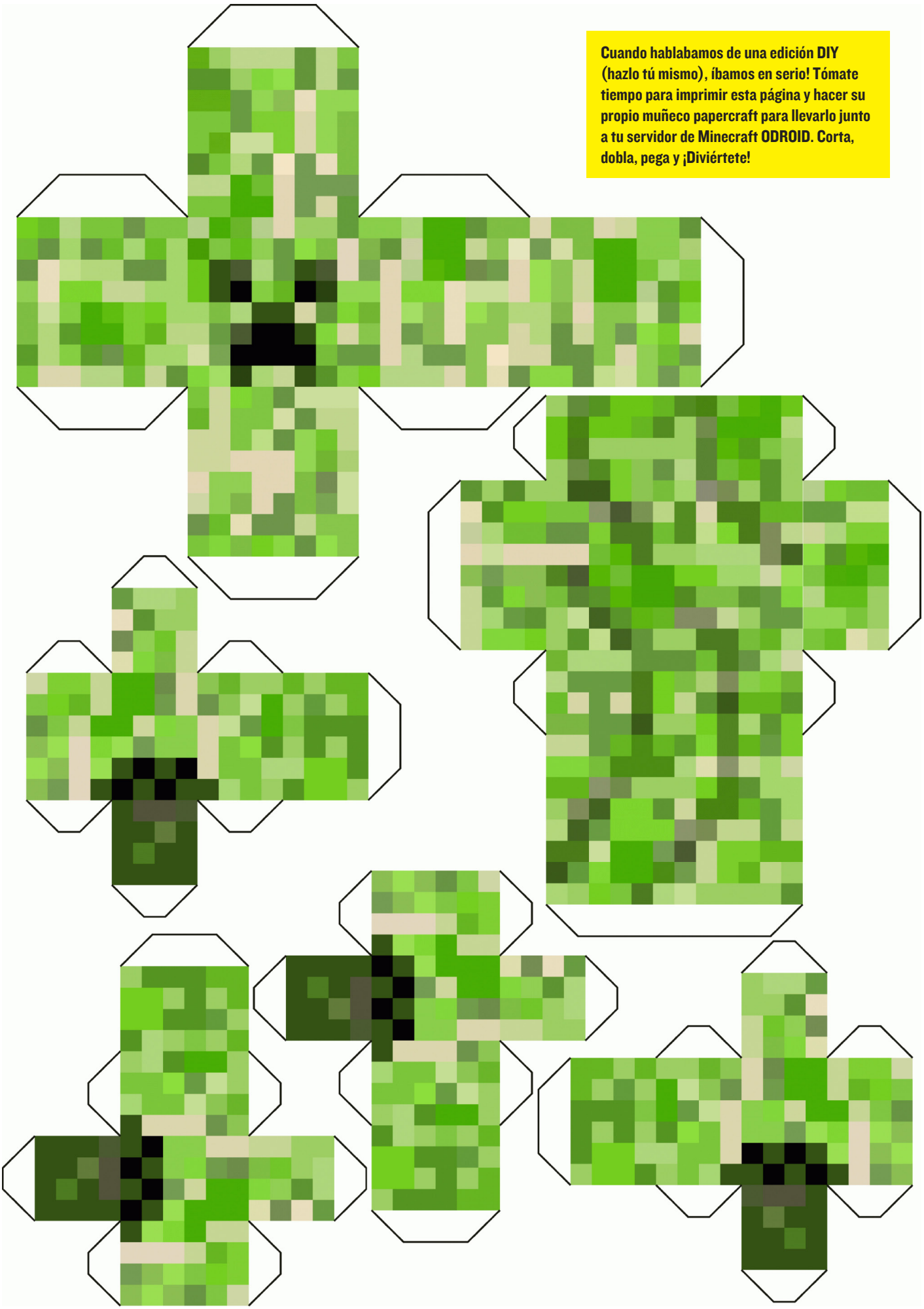
Copiar tu servidor a un servicio de alojamiento externo

Usar una versión de código abierto de Minecraft le permite cambiar cualquier aspecto del servidor, depurar errores e instalar complementos. Desde que Minecraft para ODROID está escrito en Java, resulta fácil para principiantes y expertos mejorar el software y adaptarlo a sus propias necesidades.

Una vez que tengas tu mundo a punto, puede migrar tu creación Minecraft a un servidor de alto rendimiento para que pueda soportar más jugadores. Simplemente sube todos los archivos del directorio minecraft, spigot o craftbukkit de ODROID utilizando el panel de administración del servicio de alojamiento web.

Disfruta de tu nuevo ODROID Minecraft Server, y ¡Recuerda mantenerte lejos de la lava! Para más información y resolver dudas, puedes visitar el hilo orginal del foro en **http://forum.odroid.com/viewtopic.php?f=52&t=84**.

Cuando hablabamos de una edición DIY (hazlo tú mismo), íbamos en serio! Tómate tiempo para imprimir esta página y hacer su propio muñeco papercraft para llevarlo junto a tu servidor de Minecraft ODROID. Corta, dobla, pega y ¡Diviértete!



CONOCER REBOL

ESCRIBIR PROGRAMAS ÚTILES CON UN CÓDIGO ASOMBROSAMENTE PEQUEÑO Y FÁCIL DE ENTENDER

por Nick Antonaccio y Bohdan Lechnowsky

En la primera entrega de "Conocer Rebol" discutimos la motivación que hay detrás de Rebol y aprendimos lo fácil que es crear un programa basado en GUI en Rebol con Android. Ampliamos estos ejemplos en la edición del mes pasado. Este mes, indagaremos aún más en lo que podemos hacer con Rebol 3 en ODROID y otras plataformas.

En esta entrega, mostraremos las direcciones web de donde obtener la versión más actualizada de Rebol para diferentes plataformas. Los binarios no ARM están enumerados para que puedas probar tus programas Rebol 3 en tu portátil y en ordenadores de escritorio, (tenga en cuenta que no todos las versiones Rebol 3 tienen componente gráfico disponible, todavía).

También es un placer anunciar que los actuales desarrollos de Rebol 3 para Linux ARM hard-float se están compilando y probando en ordenadores ODROID.

Y recuerda, que puede ejecutar igualmente cualquier aplicación creada en Rebol 3 para ODROID en tu teléfono o tablet con Android.

Instalación

Android:

Abra su navegador Web y vaya a `http://development.saphirion.com/experimental/builds/android/`

Descargue r3-droid.apk (ocupa menos de 2 MB).

Cuando haya terminado, haga doble clic sobre el icono de descarga (normalmente un reloj) y conceda los permisos necesarios para su instalación

Vaya a la lista de aplicaciones y haga clic en el icono de R3/Droid

Ubuntu:

Abra el navegador web y descarga la versión ARM desde `http://atronixengineering.com/downloads.html`.

Ejecuta los siguientes comandos desde el terminal en el directorio donde descargaste r3:

```
sudo mv r3-armv7hf-view-linux r3
sudo chmod +x r3
sudo ./r3
```

Windows (x86), Linux (x86), OSX (x86):

`http://atronixengineering.com/downloads.html` o `http://rebolsource.net/ *`

Windows (x64), Linux (x64):

`http://atronixengineering.com/downloads.html`

OSX (PPC), Haiku (x86), Linux ARM (soft-float), Linux (IA64), OSX (x64):

`http://rebolsource.net/ *`

(* Estos desarrollos no tienen todavia componentes gráficos)

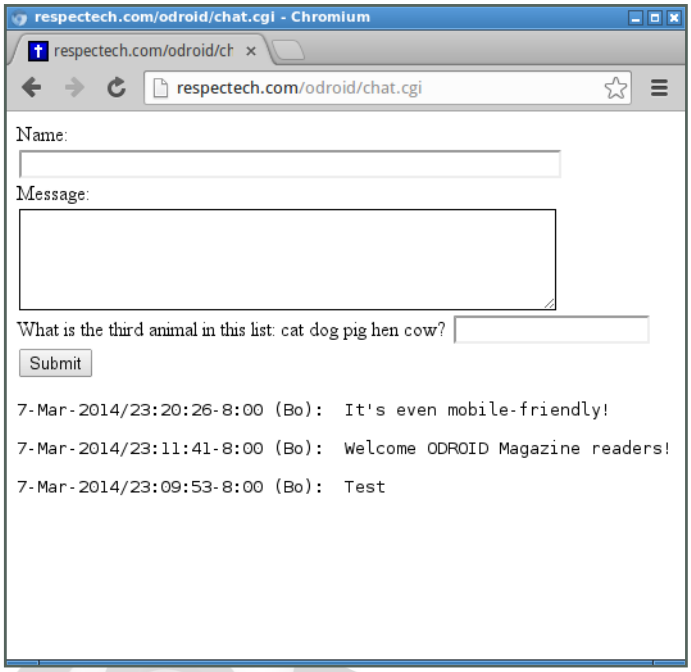
Escribir más programas en Rebol

El objetivo de estos ejemplos no es *enseñar* programación en Rebol, sino más bien mostrar lo que podemos hacer con muy poco. Si deseas más recursos para aprender, ve al final de este artículo.

Esta es una pequeña app web de chat ejecutada en `http://respectech.com/odroid/chat.cgi`, completa y con un sistema de verificación sencillo que hace más difícil el envío de spam. El sistema de verificación utiliza una característica de Rebol donde datos y código son intercambiables. Esto permite hacer cosas simples como un sistema de verificación:

```
#!./rebol3 -cs
REBOL [title: "Group Chat"]

;The following line is required as the first
line in cgi output
print {content-type: text/html^/}

;Define where the chat messages are stored
url: %./chat.txt
```

```
;Initialize the username
username: copy ""

;Read the POST string to see if there is data
to be processed
if attempt [
    submitted: parse (to string! read system/
ports/input) "&="
][
    ;Only process the following lines if POST
data was submitted

    ;In POST data, spaces are replaced by "+",
so change them back to
    ; spaces
    foreach item submitted [replace/all item
"+" " "]

    ;If there was some data to process and the
verification question was
    ; correctly answered, add the message to
the end of the chat file
    if all [
        submitted/2 <> none

        ;The "load" statement takes the ordi-
nal value picked at random
        ; (e.g. The word "first") and converts
it to a Rebol word.
        ; The "do" statement tells Rebol to
evaluate what follows it,
        ; in the case of this example, the
command "first", which picks
        ; the first item out of a series.
        submitted/6 = do load submitted/5
parse "cat dog pig hen cow" ""
    ][
        write/append url mold rejoin [
            now " (" submitted/2 "):  "
submitted/4 "^/^/"
        ]
        username: submitted/2
    ]
]

;Convert the chat file into plain text, includ-
ing any new message that was
; just added above.  Display it in reverse or-
der so the newest messages
; stay on top, right after the input section.
notes: head reverse load dehex copy read/
string url

;Generate the pivotal part of the verification
question
random/seed now/time/precise
ordinal: to-string pick [first second third
fourth fifth] random 5

;Output the HTML page
print rejoin [
    {<FORM METHOD="POST">
        Name:<br>
        <input type=text size="65"
name="username" value="} username {"><br>
        Message:<br>
        <textarea name=messwage rows=5
cols=50></textarea><br>
        What is the } ordinal { animal in this
list: cat dog pig hen cow?
        <br><input type=text name="} ordinal
{"><br>
        <input type="submit" name="submit"
value="Submit">
    </FORM>}
    "<pre>" notes "</pre>"
]
```



Nota importante: Para hacer más eficaz la ejecución de los ejemplos a partir de ahora, vamos a descargar la definición del dialecto gráfico r3-gui.r3 en la memoria local de tu dispositivo en lugar de descargarlo cada vez que lo necesitemos. Podemos hacerlo desde el mismo Rebol, simplemente escriba lo siguiente:

```
write %r3-gui.r3 read/string http://www.atron-
ixengineering.com/r3/r3-gui.r3
```

Si se produce algún error al ejecutar cualquiera de los scripts de ejemplo en tu dispositivo, intente esto:

```
write %r3-gui.r3 read/string http://develop-
ment.saphirion.com/resources/r3-gui.r3
```

Rebol 3 es de código abierto y existen diversos grupos trabajando en mejoras. Esto da lugar a tener diferentes versiones para diferentes dispositivos en situaciones ligeramente distintas en un momento dado. Con el paso del tiempo se unificarán y estos problemas desaparecerán.

Con el comando anterior se acelerará la ejecución ya que el dialecto r3-gui no necesita ser descargado cada vez que lo necesitemos. Sin embargo, en la mayoría de las tablets y los teléfonos Android no root no se permite acceso de superusuario, por lo que no es posible escribir en el directorio raíz. Esto no debería ser un problema en tu ODROID con Android. En este caso, bien seguimos utilizando load-gui o escribimos r3-gui.r3 para otra ubicación, como la SD con este comando:

```
write %/sdcard/r3-gui.r3 read/string
http://.../r3-gui.r3
```

(Reemplaza "…" por una de las rutas URL de los ejemplos anteriores.)

He modificado los ejemplos en la página web para probar r3-gui.r3 en el directorio actual y en la raíz de la SD. si no aparece en cualquiera de las ubicaciones, utiliza load-gui. Hice esto para reemplazar load-gui en los siguientes ejemplos con este código:

```
foreach cmd [[do %r3-gui.r3][do %/sdcard/r3-
gui.r3][load-gui]][
    if attempt [do probe cmd][break]
]
```

Básicamente, hay tres formas diferentes de cargar el dialecto r3-gui, empieza a probarlas hasta que alguna te funcione correctamente.

Para ejecutar los ejemplos de la página web en lugar de escribirlos, tan sólo tienes que insertar:

```
do http://respectech.com/odroid/learnrebol/
file.r
```

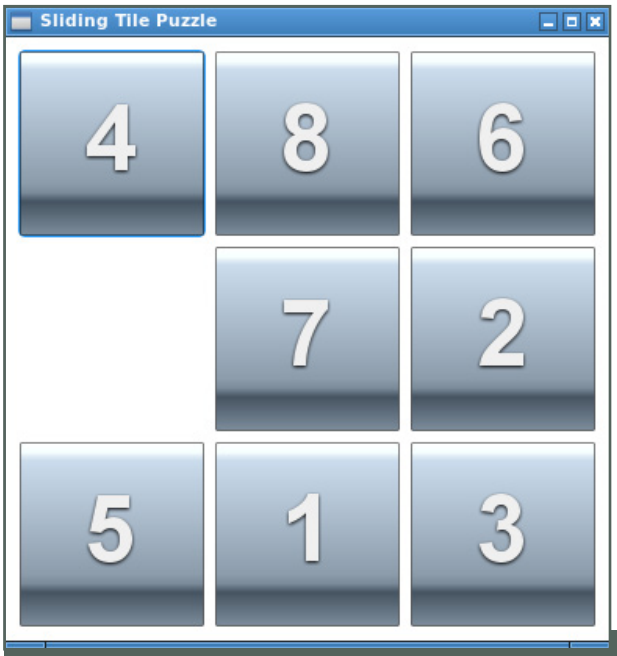
Reemplaza file.r por el nombre del archivo en la cabezera de Rebol (deja el signo "%").

Este es un pequeño juego grafico de fichas deslizante. No se necesita ninguna herramienta compleja de desarrollo GUI para crear este código. Es muy sencillo y lo suficientemente legible para que lo único que necesites sea un editor de texto y los servicios de ayuda de Rebol. El código solo contiene 5 líneas. ¿Alguna vez has visto un código tan simple para crear un juego de Android (o incluso un escritorio)? No se necesitan IDE, SDK o desarrollar scripts, sólo tienes que descargar el pequeño intérprete R3 a tu dispositivo Android o PC, hacer clic en el archivo de texto plano y se ejecutará en cualquier plataforma, con gráficos, soporte táctil y sin cambios en el código:

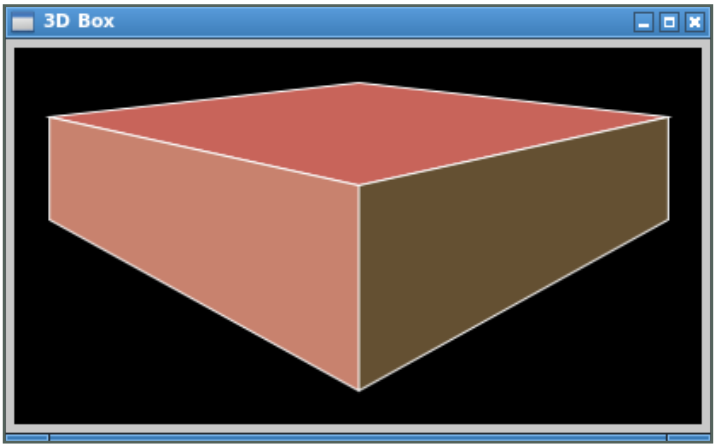
```
REBOL [title: "Sliding Tile Puzzle" file:
%sliding-tile-game.r]
load-gui
sz: 120x120
fontize [
    p: button [font: [size: 60]]
]
stylize [
    p: button [
        facets: [text-style: 'p init-size: sz
max-size: sz]
        actors: [
            on-action: [
                t: reduce [face/gob/offset x/
gob/offset]
                face/gob/offset: t/2  x/gob/
offset: t/1
            ]
        ]
    ]
]
view/options [
    hgroup [
        p "8"   p "7"   p "6"   return
        p "5"   p "4"   p "3"   return
        p "2"   p "1"   x: box sz white
    ]
] [bg-color: white]
```

En el tema de juegos, hay que señalar que R3 permite trazar gráficos y crear animaciones con gran facilidad. Aquí tienes un ejemplo rápido:

```
REBOL [title: "3D Box" file: %3d-box.r]
load-gui
bck: make image! 400x220
view/no-wait [image bck]
draw bck to-draw [
    fill-pen 200.100.90
    polygon 20x40 200x20 380x40 200x80
```

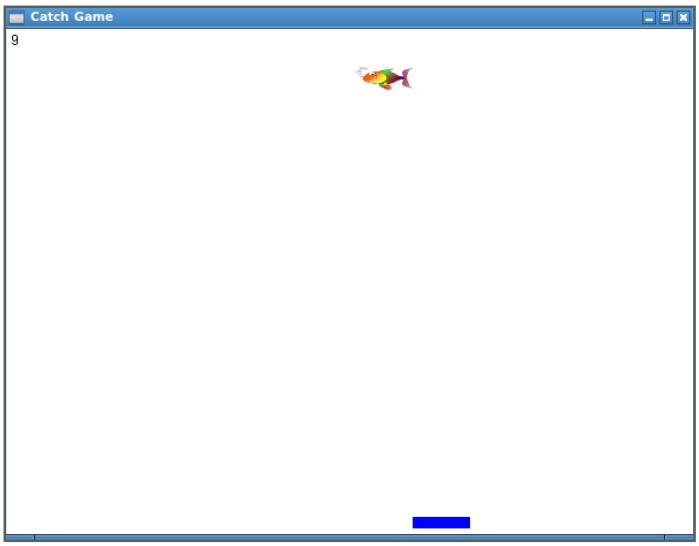



```
    fill-pen 200.130.110
    polygon 20x40 200x80 200x200 20x100
    fill-pen 100.80.50
    polygon 200x80 380x40 380x100 200x200
] copy []
do-events
```



Este un completo juego arcade con animación de imágenes, detección de colisiones, controles de teclado, guardar puntuaciones y mucho más. Intenta atrapar los peces que caen. Ten cuidado, ¡Se hace más rápido a medida que avanzas!

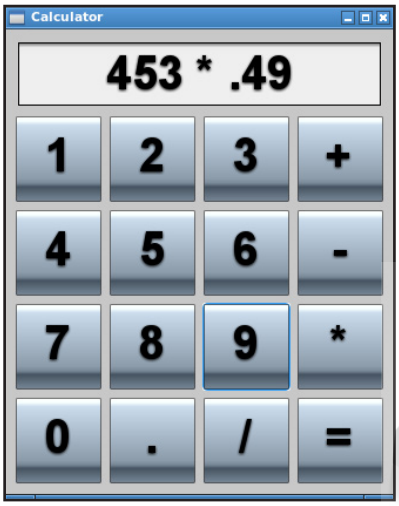
```
REBOL [title: "Catch Game" file: %catch-game.r]
load-gui
fish: load http://learnrebol.com/r3book/fish2.
png
s: 0  p: 3  random/seed now/time
stylize [
    paddle: box [facets: [max-size: 50x10]]
    img: image [facets: [max-size: 50x20 min-
size: 50x20]]
]
view/no-wait/options [
    t: text"ARROW KEYS" y: img 50x20 (fish) pad
z: paddle blue
    return
    arrow left 120x120 arrow right 120x120
] [
    shortcut-keys: [
        left  [z/gob/offset/1: z/gob/offset/1 -
50 draw-face z]
        right [z/gob/offset/1: z/gob/offset/1 +
50 draw-face z]
    ]
    min-hint: 600x440  bg-color: white
]
forever [
    wait .02
   y/gob/offset/2: y/gob/offset/2 + p draw-face
y show-now y
    if inside? y/gob/offset (z/gob/offset -
49x0) (z/gob/offset + 49x10)[
        y/gob/offset: random 550x-20 s: s + 1
set-face t form s  p: p + .3
    ]
    if y/gob/offset/2 > 425 [alert join "Score:
" s unview unview break]
]
```



Esta es una versión R3 de un programa que se encuentra en casi todos los textos de formación GUI, una calculadora básica. No hay más archivos, plantillas de diseño, scripts de arranque o herramientas que sean necesarios para ejecutar esta aplicación en cualquier plataforma. Es un programa completo y totalmente portátil. Como puedes imaginar cuenta con muy poco código, se necesitan pocos conocimientos para entender cómo funcionan ejemplos como este. Compara este código con C++

(**http://afsalashyana.blogspot.com/2012/06/gui-simple-calculator-visual-c-source.html**), Visual Basic (**http://archive.msdn.microsoft.com/spektrum1calculator**), o incluso con el ejemplo RFO mas simple (**http://rfobasic.com/#section-12.2**). Este último ejemplo fue escrito para probar la herramienta de desarrollo de Android, sencilla y productiva disponible a tu alcance y cada uno de esos ejemplos sólo se ejecuta en un único sistema operativo. Aquí ienes un pequeño ejemplo de HTML5 (**http://thecodeplayer.com/walkthrough/javascript-css3-calculator**). Requiere varias páginas de HTML, CSS y Javascript. Todos estos ejemplos sólo rozan la superficie de la complejidad de otros entornos de desarrollo:

```
REBOL [title: "Calculator" file: %calc.r]
load-gui
sz: 100x100
fontize [btn: button [font: [size: 60 color:
black]]]
stylize [
    btn: button [
        facets: [text-style: 'btn init-size:
sz max-size: sz]
        actors: [on-action:[set-face f join
get-face f get-face face]]
    ]
    field: field [
        facets: [text-style: 'btn init-size:
415x60 max-size: 415x60]
    ]
]
view [
    hgroup [
        f: field return
        btn "1"  btn "2"  btn "3"  btn " + "
return
        btn "4"  btn "5"  btn "6"  btn " - "
return
        btn "7"  btn "8"  btn "9"  btn " * "
return
        btn "0"  btn "."  btn " / "   btn "="
on-action [
            attempt [set-face f form do get-
face f]
        ]
    ]
]
```



Recursos

Chat Online y Soporte:

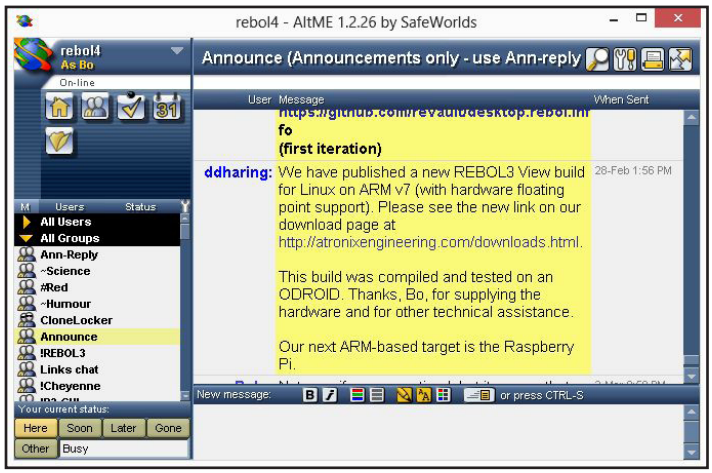
StackOverflow.com:

Actualmente hay más de 1100 preguntas (y respuestas) relacionadas con Rebol en StackOverflow.com (**http://stackoverflow.com/search?q=rebol**).

Se necesitan 20 puntos para chatear en StackOverflow.com (**http://chat.stackoverflow.com/rooms/291/rebol-and-red**). Si no tiene 20 puntos (o una cuenta en absoluto), ve igualmente y busca [Rebol y Red] en las salas de chat. Por lo general es una de los más activas. Te ayudaremos a conseguir los 20 puntos que necesitas para chatear.

AltME:

Para unirse al mundo AltME de Rebol, envía un correo electrónico al usuario bo del dominio respectech.com pidiendo ser invitado. Somos una comunidad cerrada para evitar el spam. No seas tímido, suelen decir que somos la comunidad de desarrollo de software más amigable del planeta.



Facebook:

https://www.facebook.com/groups/rebol/

SER ESCUCHADO CON ÜBERCASTER

UN TRANSMISOR DE AUDIO EN TIEMPO REAL

por K.J Yoo of Echos Design (www.echosdesign.com)

El Ubercaster, desarrollado en la plataforma ODROID, es un modo del siglo 21 de escuchar música en vivo a través de su smartphone sin tener que dar empujones a la gente!!!

Corría el año 2010. En las calles de la Altstadt en Marburg, Alemania solia tocar el violín como músico callejero. Algunos encontraban mi música molesta, en cambio a otros le gustaba. Como estudiante de ingeniería pensé en una mejor forma de ofrecer mi música para que, sólo aquellos que estubiesen interesados pudieran escucharla sin problema. Tras comprobar que los sistemas de emisora FM son caros, enormes y simplemente poco prácticos. Decidi tomar las riendas por mi cuenta. La idea era simple: transmitir audio al dispositivo favorito de la gente: el smartphone.

Objetivo del Diseño

Deseaba que cualquier persona pudiera acceder con facilidad a cualquier archivo de audio con Übercaster. No importaba si procedía de un instrumento, TV, iPod o un micrófono. El Übercaster transmitiría el sonido a nivel local y los oyentes utilizarían su smartphone para conectarse al Übercaster como un punto de acceso wifi para "sintonizarse". También deseaba que Übercaster fuese un dispositivo elegante e intuitivo cumpliendo los 10 principios del buen diseño de Dieter Ram.

Desarrollo

He estado desarrollando el Übercaster con ODROID X2/U2/U3 desde agosto de 2013, y se compone de un dispositivo y de aplicaciones móviles clientes.

En esencia, el dispositivo Übercaster es un ODROID U3 que ejecuta hostap. (si no estas familiarizado con Hostap, revisa el artículo de Mauro Ribeiro de la edición de febrero "Usar un ODROID-XU como un router WiFi.") El dispositivo ejecuta Ubuntu 13.06 con un kernel ODROID-3.8.y personalizado. La aplicación Übercaster captura audio con ALSA, codifica éste con OPUS (`http://www.opus-codec.org/`) dividiéndolo en paquetes OPUS para su difusión por UDP. Este proceso emplea un promedio de 8ms y requiere en torno a un 6-9% de la CPU. Admito que ODROID U3 puede ser excesivo, pero no fue capaz de encontrar otra placa que permitirá un codec de audio de tan alta calidad.

Así que ODROID funciona perfectamente, y doy ¡Mis felicitaciones a Hardkernel!

A través de Hostap, dispositivos wifi, como smartphones, tablets y ordenadores pueden conectarse al dispositivo Übercaster, que ejecuta isc-dhcp-server para gestionar todos los clientes. Cuando se establece una conexión, la aplicación cliente móvil Übercaster se puede utilizar para escuchar lo que el dispositivo Übercaster está transmitiendo. La aplicación escucha a la dirección IP del dispositivo emisor, recibe los paquetes, lo decodifica y reproduce el sonido.

A primera vista, parece una aplicación básica de streaming, como VLC o Icecast. Sin embargo, Übercaster se ejecuta en tiempo real. El tiempo real es relativo y subjetivo, dependiendo de las aplicaciones. Para el sistema Übercaster el objetivo es que el tiempo total de descarga del audio sea inferior a 25 ms. Esta latencia de

Un prototipo impreso en 3D del dispositivo Ubercaster , que no debe confundirse con una pastilla de jabón con salida de auriculares .

audio es el retraso entre el momento en el que el audio entra en el dispositivo Übercaster y éste es reproducido en un iPhone 5S. (iOS tiene una latencia inferior a los dispositivos Android). 25ms de latencia de audio no es un número arbitrario, sino que representa la supuesta latencia máxima antes de que una persona sea capaz de percibir el retraso. En la actualidad, la latencia de audio es < 50 ms en los dispositivos iOS y en los dispositivos Android varía significativamente de un dispositivo a otro. En el Google Nexus 7 (2013), la latencia es de 80 ms. He probado la Übercaster con múltiples participantes y aunque la latencia total es actualmente el doble de mi objetivo y ésta varía entre dispositivos iOS y Android, el 95 % de los oyentes no fueron capaces de percibir retrasos cuando veían la televisión o una película .

Entonces, ¿cuántos clientes puede soportar Übercaster? He probado hasta 25 clientes. Sin embargo, en teoría es posible tener muchos más. Una vez establecida la relación cliente-servidor, Übercaster es básicamente un sistema de sentido único. Übercaster transmite los paquetes UDP y los clientes simplemente sintonizar una dirección IP. Eso es todo. Sin embargo, tiene una desventaja: UDP no siempre es fiable. UDP envía paquetes más rápido y más eficiente que TCP, ya que no utiliza confirmación. Esta es la razón por la Übercaster transmite usando pequeñas estructuras de paquete, para protegerse contra el alto índice de pérdida de paquetes, lo que ofrece una reproducción más fluida.

Video Demostrativo

Por favor, vea la siguiente demostración del Übercaster.

Übercaster Zwei:

`vimeo.com/85006122`

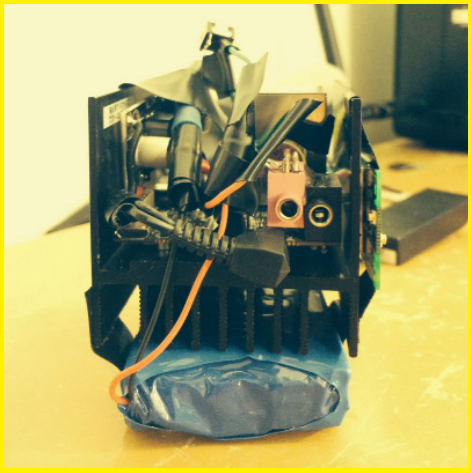
Übercaster Drei:

`vimeo.com/88467399`

Afrontar los Problemas

1. Para minimizar la interferencias estoy usando 802.11n en banda de 5GHz. La banda de 2,4 GHz no funciona en zonas moderadamente llenas. Usando 5GHz el alcance es más corto y requiere un poco más de potencia, pero es muy estable. Así, en el CES 2014 en Las Vegas, no tuve ningún problema al ofrece una demostración en medio del Pabellón

El Ubercaster se ha convertido en una máquina elegante y sexi desde su primer prototipo de cinta y goma de mascar.



Usando un hardware mínimo, el Ubercaster ofrece un sonido de alta fidelidad y sólo consume 8W de potencia.

Sur repleto de gente. (Será muy interesante trabajar con un módulo 802.11ac muy pronto!)

2. Con el fin de reducir la latencia, uso OPUS, SPSC Circular Queue y un protocolo adaptado basado en UDP. Probé RTMP, RTSP y HTTP pero no me llegaron a funcionar. Al principio, queria usar VLC u otro cliente RTSP para transmitir contenido a los dispositivos cliente pero la latencia era muy alta. Es por eso que decidí optar por aplicaciones nativas, que son muy ligeras. Actualmente estoy desarrollando una API que ayudará a los desarrolladores de aplicaciones móviles integrar la función de transmisión Übercaster. Una sugerencia relacionada con Android: es importante que coincida la tasa de muestreo y el tamaño de búfer de latencia mínima. Echa un vistazo a esta interesante charla de Google I/O 2013 sobre Audio de alta calidad en Android: `https://www.youtube.com/watch?v=d3kfEeMZ65c`.

La Aplicación

Übercaster comenzó con una simple pregunta: ¿cómo pueden las personas tener total libertad y perfecto control de lo que escuchan en un local de sus alrededores? o, ¿Cómo pueden las personas tener libertad para transmitir con facilidad sonido a los miembros del un local de sus alrededores?

Resulta que lugares como gimnasios, restaurantes, zonas turísticas, salas de conciertos, bares y aeropuer-

tos han estado pensando en formas innovadoras de transmitir el sonido. Ha habido intentos de utilizar FM o infrarrojos, sin embargo no resultaron ser prácticas además de caras y complicadas de usar.

Desde el principio, el móvil era la idea central del producto. Actualmente el 65% de los usuarios de teléfonos móviles en los EE.UU. utilizan un smartphone. Es ampliamente usado y está creciendo a un ritmo asombroso. Así que en esencia todo el mundo ya tiene dispositivos receptores de Übercaster

Imagínate que vas a un bar concurrido y puedes escuchar cualquier televisor, o sintonizar las noticias de última hora mientras esperas el vuelo a Frankfurt, o escuchar con perfecta claridad al músico callejero tocar la guitarra a 50 metros de distancia, o de experimentar un viaje a Roma a través de tu smartphone.

Übercaster no sólo ofrece una mejor y mayor calidad de audio que los productos actuales, sino que también ofrece una experiencia increíble y trasparente tanto para los que transmiten como para los que escuchan. Übercaster simplifica, reduce y mejora la difusión de audio local en un único dispositivo.

La Visión

El sonido permite probar si realmente funciona la distribución de información pública a nivel local. Pienso bastante en el futuro y está claro que las bandas de frecuencia se están llenando, la gente quiere más ancho de banda y la información más rápidamente. Creo que en los espacios públicos hay demasiadas redundancias de datos/bits. Si una gran cantidad de personas en una área pública están interesados en saber más sobre algo como el *Equipo del Real Madrid*, resulta redundante que sus dispositivos acceder a la información desde el mismo servidor a miles de kilómetros de distancia, en Texas o California. Los Televisores en un área pública son, en esencia una forma de difusión local que ven las personas en un radio de 50 metros. Sin embargo no estoy satisfecho con la forma en la que funcionan actualmente. Así que mi objetivo es la distribución local de contenidos. Digamos que alguien ve en una televisión en el aeropuerto en la que la CNN retransmite una noticia de última hora. Se debe ser capaz de acceder al sonido en tiempo real al mismo tiempo que se transmite el vídeo de alta definición en su teléfono a una distancia local, y de igual forma al contenido web adicional relacionado con las noticias que se están continuamente agregando al dispositivo Übercaster para su disfusión. Es muy eficiente, las personas obtienen información mas rápidamente y sin problemas.

Si estás interesado en saber más sobre el Übercaster o tienen interés en esta tecnología, por favor envíeme un e-amil a **KJ@EchosDesign.com**.

COMUNICACION I²C ODROID U3

CIRCUITOS INTEGRADOS PARA EL RESTO

por John Taylor

Después de encargar mi ODROID-U3 especificamente para la comunicación I2C con varios dispositivos esclavos, no he podido encontrar una guía completa que detalle el proceso de cómo configurarlo todo. Con el fin de compartir con otros lo que he aprendido, he creado mi propia guía para configurar un sistema I2C en ODROID.

Este artículo pretende introducirte en la comunicación I2C usando ODROID-U3 como patrón. Nos comunicaremos con una matriz de LED de Adafruit. Inicialmente pensé en escribir este tutorial usando un microprocesador MSP430 de Texas Instruments, que he configurado con éxito. Comprobe sin embargo, que los materiales y programación necesarios para ese proyecto estaban fuera del alcance de este artículo.

Reunir el Equipamiento

- **ODROID-U3**
- **Matri I2C LED** http://www.adafruit.com/products/1049
- **Conversor de Nivel** http://www.adafruit.com/products/757?gclid=CINsJL057wCFURk7AodZkAArg

Configurar ODROID-U3

Tenemos que instalar las herramientas I2C para que podamos probar el bus I2C. Esto se hace fácilmente ejecutando el siguiente comando en el terminal, requiere unos minutos para completarse:

```
sudo apt-get install i2c-tools
```

Ahora que tenemos el paquete de herramientas i2c, necesitamos cargar el módulo i2c-dev para poder usarlo. Para ello usa el comando modprobe. Cada vez que ODROID se reinicie tendrás

que volver a cargar el módulo. Para evitar esto, añadiremos i2c-dev a la lista de módulos que se cargan durante el arranque. Abre el archivo /etc/modules con tu editor de texto favorito, como nano y añade i2c-dev a la lista.

```
nano /etc/modules
```

```
 GNU nano 2.2.6                File: /etc/modules

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
i2c-dev

                              [ Read 5 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

El fichero /etc/modules es editado usando Nano.

Una vez guardado el archivo, reinicia ODROID y asegúrate de escribir el siguiente comando. El resultado que aparecera será similar a la imagen de abajo.

```
i2cdetect -l -y
```

Este comando le dice al ordenador que detecte y enumere (`-l`) todos los puertos I2C disponibles. Si no usas el delimitador `-y` el ordenador te preguntará si estás seguro que deseas realizar esta acción y te avisa de los posibles daños al experimentar con buses I2C.

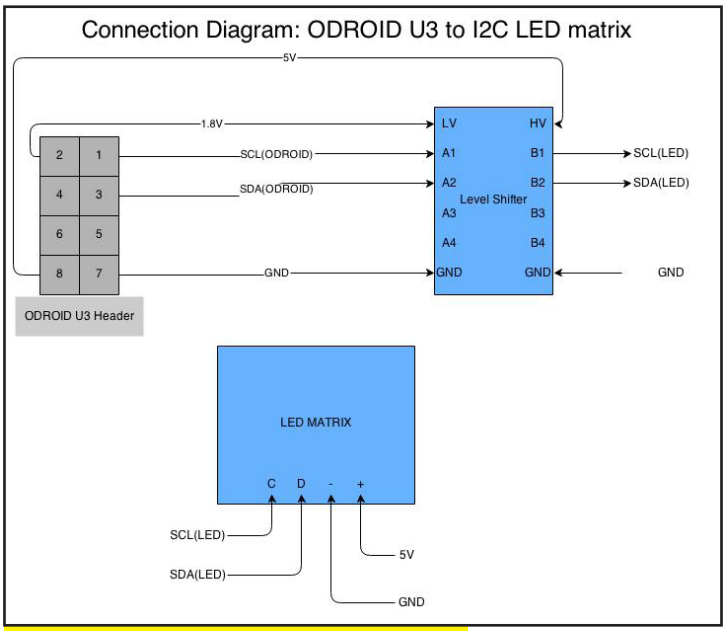
```
i2c-0   i2c             s3c2410-i2c                             I2C adapter
i2c-1   i2c             s3c2410-i2c                             I2C adapter
i2c-2   i2c             i2c-gpio2                               I2C adapter
i2c-3   i2c             s3c2410-i2c                             I2C adapter
i2c-4   i2c             i2c-gpio4                               I2C adapter
i2c-7   i2c             s3c2410-i2c                             I2C adapter
i2c-8   i2c             s3c2410-i2c                             I2C adapter
```

Lista de puertos I2C disponibles en ODROID-U3.

El bus asignado al conector de 8 pines es i2c-4. Detallaremos su uso después de conectarle un dispositivo esclavo.

Cableado

Ahora que tenemos nuestro ODROID configurado para la comunicación I2C, podemos conectar nuestro dispositivo esclavo. El dispositivo que vamos a usar es una matriz LED de Adafruit Industries. Puesto que ODROID-U3 es un dispositivo de 1.8V y nuestra matriz LED un dispositivo de 5V, usaremos un conversor de nivel I2C seguro, también de Adafruit Industries.



Un simple diagrama de la interacción entre el U3 y la Matriz LED I2C.

Comunicación

Ahora nos asegurarnos de que hemos conectado todo correctamente. Por suerte, podemos hacerlo con relativa facilidad usando las herramientas I2C. Después de conectarlo todo ejecuta el siguiente comando:

```
i2cdetect -y 4
```

Este comando le dice a ODROID que enumere todos los dispositivos I2C conectados al bus 4. Como muestra la siguiente figura nuestra matriz LED esta en la posición 70.

Si no ves un dispositivo en la posición 70, vuelva a comprobar el cableado.

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
```

Matriz de periféricos I2C, mostrando el dispositivo I2C del ODROID en la posición 70

Código C

Una vez que sabemos que todo está conectado correctamente, podemos escribir algo de código C simple para controlar la matriz LED. El código que se muestra a continuación activa la matriz de LED e ilumina de forma secuencial cada LED.

Después de compilar y ejecutar el código en ODROID, verá

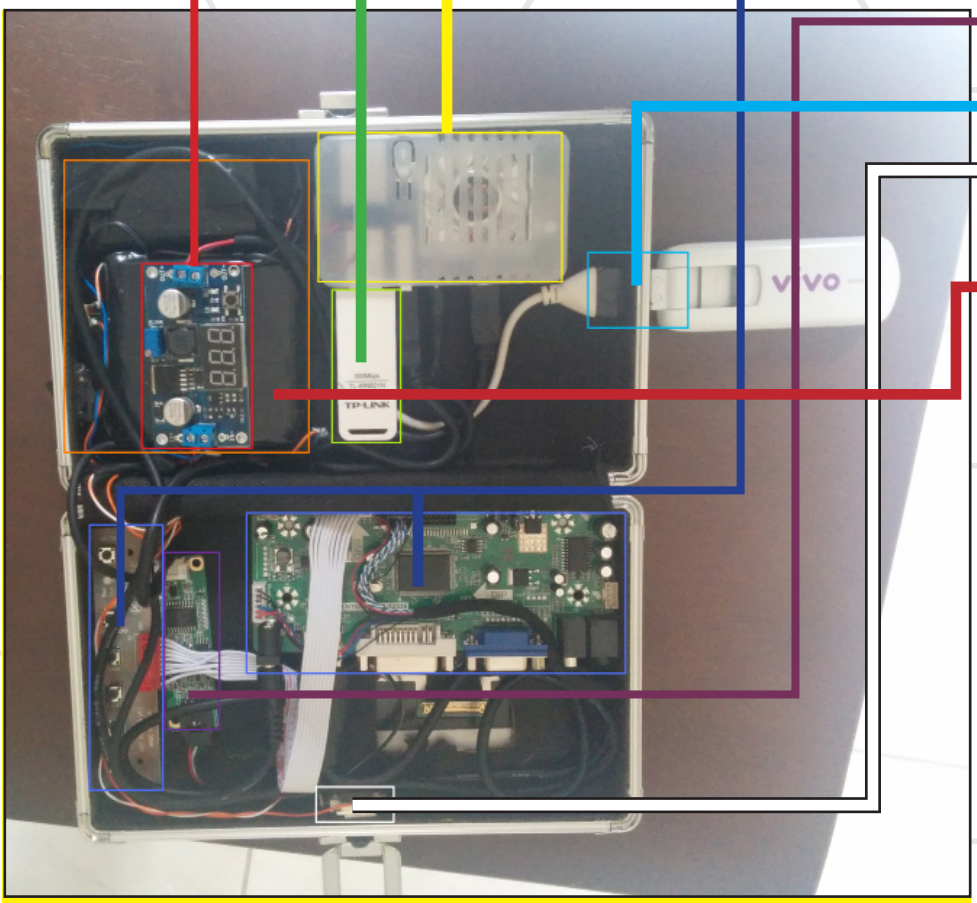
un resultado similar al del vídeo que se muestra en **http://bit.ly/1fMOyMt.** El código puede ser modificado con facilidad para mostrar otros patrones o formas en la matriz LED.

```
#include <stdlib.h>
#include <unistd.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <string.h>
#include <stdio.h>

int i = 0;
int j = 0;
int k = 0;

int main(void)
{
char recieveBuffer[32];  //The Buffer that will
hold onto recieved data
     char transferBuffer[32]; //The buffer that
holds data that we will send
     int address = 0x70;       //The address of
the LED matrix
     int tenBitAddress = 0;    //variable that
says we aren`t using 10-bit
    //addressing

     //Initialize the I2C channel
     int i2cHandle = open("/dev/i2c-4",O_
RDWR);

//Tell the I2C channel we aren`t using ten bit
addressing
     ioctl(i2cHandle,I2C_
TENBIT,tenBitAddress);

//Tell the I2C channel we have a slave at Ad-
dress 70
     ioctl(i2cHandle,I2C_SLAVE,address);

//make sure there is no data in our buffers
memset(recieveBuffer, 0 , sizeof(recieveBuffer)
);
memset(transferBuffer,0,sizeof(transferBuffer));

//start internal oscillator on the LED matrix
by sending 0x21 command
transferBuffer[0] = 0x21;
write(i2cHandle, transferBuffer, 1);

//enable display and turn blink off by sending
0x81
transferBuffer[0] = 0x81;
write(i2cHandle, transferBuffer,1);

//set brightness to max by sending 0xEF
transferBuffer[0] = 0xEF;
write(i2cHandle, transferBuffer,1);

//top level loop keeps track of which column
we are on
for(i = 0; i<16;i=i+2)
{
for(j = 0; j<9;j++)
          {
//we send two bytes in this case, so we load
the
//transfer buffer with 2 bytes
//and set the first Byte to transfer to the
column number
transferBuffer[0] = i;

//set the second Byte to transfer to the
lights to turn on
transferBuffer[1] = 0x01 <<  j;
write(i2cHandle, transferBuffer,2);

//wait a while
for(k = 0; k < 5000000;k++);
}

//make sure a column is completely off before
leaving it
transferBuffer[1] = 0x00;
write(i2cHandle, transferBuffer,2);
}
}
```

TABLET LINUX RESISTENTE Y PORTATIL

CON ROUTER LTE

por Mauro Ribeiro



Deseaba desarrollar una Tablet PC basada en ODROID que fuese duradera, resistente y montada con componentes fácil de conseguir. Debajo de la imagen de la derecha aparece una lista con los componentes que he usado.

Software

La parte de software de este proyecto fue lo que más me dio dolor de cabeza. Durante mis primeras pruebas con Ubuntu 13.10, no podía conectarme a la red LTE estando fijado el 3G, no podía entender por qué. Me hizo falta un tiempo para descubrir que la versión ModemManager en Ubuntu 13.10 está ligeramente desactualizada y no soportaba correctamente LTE. Actualizar ModemManager por sí solo era casi imposible, ya que está conectado a NetworkManager y tiene muchas dependencias, así que necesitaba algo nuevo. Mi mejor opción era usar Arch Linux ARM, que cuenta con las últimas versiones de los paquetes.

Una vez que conseguí ejecutar Arch Linux ARM, necesitaba una interfaz de usuario agradable para la pantalla táctil, ya que no dispongo de teclado o ratón conectado a éste. He probado algunas interfaces de usuario como KDE Plasma Active, pero luego escuché algunas noticias sobre Mate 1.8 y decidí darle una oportunidad. Dio buen resultado!

Sin embargo, todavía no podía usar mi módem para enviar y recibir SMS y necesitaba algo para monitorizar el estado de la conexión. Al principio, mi idea

La caja esta hecha de una combinación de aluminio y madera y las medidas son 21x13x6cm, la compré en una tienda local de artículos de oficina.

La pantalla es una pantalla táctil LCD de 9". Ahora, seguramente queréis saber de dónde la conseguí, ¡Tengo buenas noticias para vosotros! La pantalla es un prototipo de un kit que Hardkernel pondrá a la venta muy pronto: Un monitor HDMI de 9" con una pantalla táctil integrada

He decidido usar un **ODROID-U3** principalmente por mi interés en el consumo de energía. Deseaba que fuese lo más bajo posible puesto que la pantalla y el adaptador LTE también consumen energía. ¡El adaptador LTE usa casi 500ma por si sólo!

Para la **batería**, he usado 6 celdas Li-Ion conectadas entre sí por cable en dos grupos de 3 celdas, lo que arroja unos 11.1V (12.3V) y 5000mAh. He usado la batería de un portátil.

Un **Convertidor** de potencia. Estoy usando el kit LM2596 pre-fabricado. Este puede manejar 2A sin disipador de calor, así que es más que suficiente para nuestro proyecto.

El Adaptador LTE, proporcionado por mi compañía de telefonía movil. Es un adaptador Huawei E3276 CAT 4, puede alcanzar una velocidad máxima de 150Mbps.

El Adaptador WiFi esta basado en el chipset Realtek 8192CU que es muy común. El que he usado es el modelo TL-WN821N de TP-Link.

LED Balnco es de Aliexpress. Se trata de una matriz de LED 3x3 de 10W. La estoy usando a 0,5 W para iluminar la parte interna.

ROJO es el convertidor unido a la batería

AMARILLO es el ODROID-U3

VERDE es el Adaptador WiFi

2 marcas **AZULES** representan el panel HDMI->LVDS y el panel para la visualización de la pantalla.

PULPULA es el controlador USB de la pantalla táctil

CIAN es el controlador USB de la pantalla táctil

BLANCO es una LED de 10W a sólo 0.5W para iluminar el interior en caso de quedarnos a oscuras.

NARANJA es la batería

La batería de 6 celdas está cableada como se muestra en la imagen. Emite 11.1V y 5000mAh.

Cargar las baterías de Li-Ion no es complicado, pero si requiere algunos conocimientos.

Cada celda de Li-Ion debe cargarse con 0,4V mas su tensión valorada (3,7 V) y sólo puede alimentar a la mitad de su carga nominal como corriente de carga.

Así que para este caso, puesto que estoy usando 3 células en serie (3,7 x 3 = 11.1V + 3x 0,4 V = 1,2 V = 12.3V), mi tensión de carga es 12.3V. Como sé que la capacidad total es de 5000mAh (~800mAh por batería), no utilizaré un cargador que sea superior a 2,5A. Así que terminé usando una fuente de alimentación de 13V 2A para cargar las baterías.

He usado un Dremel para hacer los agujeros externos para el USB, el enchufe, interruptor y cables.

era poner todas esas funciones en una aplicación personalizada, pero terminé encontrando Modem Manager GUI (http://linuxonly.ru/cms/page.php?7) que hace todo esto. Sin embargo, todavía faltaba una función. Necesitaba controlar el módem para poder redirigirlo a una determinada red (2G/3G/4G).

Así que hice una aplicación personalizada con QT Creator. Necesitaba co- nocer que comandos requiere el módem para forzarlo a un determinado tipo de red. No encontre documentación sobre éste en línea, así que cogí el camino difícil. Conecte el módem a un ordenador con Windows, instalé un programa para analizar el puerto serie y use la aplicación facilitada con el módem para controlarlo. Una vez conocidos los comandos, finalice mi aplicación.

Por si sientes curiosidad, los comandos para mi Modem HUAWEI son

Para seleccionar la red automáticamente:

```
AT^SYSCFGEX="00",3fffffff,1,
2,5a,"","
```

Para modo 2G:

```
AT^SYSCFGEX="01",3fffffff,1,
2,5a,"","
```

Para modo 3G:

```
AT^SYSCFGEX="02",3fffffff,1,
2,5a,"","
```

Y para LTE:

```
AT^SYSCFGEX="03",3fffffff,1,
2,5a,"","
```

Estos comandos son enviados a un puerto serie de "control". Algunos módems (como el mío) tienen incluso comandos AT que permiten conocer su temperatura interna. Proporcionan una gran cantidad de información para analizar la calidad de tu conexión/señal.

Otro problema con el que tuve que lidiar fue NetworkManager. No puedes iniciar hostapd para crear una red wifi si NetworkManager está gestionado una interfaz. Incluso si estás desconectado de la red wifi, todavía es posible que NetworkManager no pueda gestionar esa interfaz.

Sólo tiene que añadir la siguiente línea

```
[keyfile]
unmanaged-devices=\
mac:xx:xx:xx:xx:xx:xx
```

Donde xx: xx: xx: xx: xx: xx es la dirección MAC del dispositivo que quieres administrar. Luego añadí otra función para cambiar entre AP encendido y apagado.

AP encendido:

Indica a NetworkManager que no gestione mi adaptador wifi.

Inicia hostapd para crear mi red wifi.

Crear una única regla iptables para configurar NAT (compartir la conexión de Internet).

Inicia DNSMASQ para conectar el servidor DNS y DHCP

AP apagado:

Desactiva DNSMASQ

Limpia las Reglas de firewall

Detiene hostapd

Indica a NetworkManager que gestione mi wifi de nuevo.

Tal vez te preguntes, ¿Por qué no dejarlo sin control todo el tiempo? Porque todavía quiero usar el wifi como cliente cuando estoy en casa, así puedo realizar actualizaciones y curiosear.

Otra función necesaria para la tablet era instalar un teclado virtual en Linux, que está disponible tanto en Ubuntu 13.10 como en Arch Linux mediante un paquete denominado onboard. Onboard es un teclado virtual sumamente configurable y personalizable con muchas características.

Por último, activar el botón derecho cuando se usa la pantalla táctil. Esto se hace incluyendo la siguiente configuración en el archivo / etc/X11/org.conf, o creando un nuevo archivo en el directorio / etc/X11/xorg.conf.d.

```
Section "InputClass"
    Option "EmulateThirdBut-
ton" "1"
    Option "EmulateThirdBut-
tonTimeout" "750"
    Option "EmulateThirdBut-
tonMoveThreshold" "30"
EndSection
```

El EmulateThirdButtonTimeout es la cantidad de tiempo en milisegundos que debes mantener presionada la pantalla táctil para ser identificado como un clic derecho. EmulateThirdButtonMoveThreshold es la cantidad de píxeles que tu dedo puede mover y todavia considerarse que está en la misma posición.

Con todo hecho, ahora dispones de una tablet linux con pantalla táctil que también funciona como un router LTE, lo que te permite conectarte a una red 4G desde cualquier lugar.

COMO CREAR UN CAR PC PARA MI CAMIONETA USANDO ODROID

NO IMPORTA LOS PRODUCTOS QUE HAYA EN EL MERCADO ¡SACA EL MAXIMO PARTIDO A TU DINERO!

por Jeremy Leesmann (Killer Turtle)

Los coches con sistemas de navegación de alta gama son cada vez más comunes, presentandose como una importante mejora en la tecnología de los vehículos. Sin embargo, muchas unidades PC para coches presentas dos grandes problemas: 1) las actualizaciones de mapas son muy caras, y 2) están limitadas por sus funciones y software. Hay disponibles algunas unidades con Android, pero ejecutan versiones obsoletas de Android y son muy lentas. De modo que, quise instalar un PC en mi camioneta sacando el máximo partido a mi dinero y elegí ODROID-U3 para ello.

Este sistema de navegación PC ODROID mantiene la camioneta de Jeremy dentro de la carretera y fuera del barro!

Mi objetivo era disponer de las siguientes funciones a través de un PC instalado en my Camioneta:

- **Navegación**
- **Radio**
- **Reproductor MP3**
- **Botones Fisicos para su control**

Estas son las piezas que utilicé para desarrollar mi sistema:

ODROID-U3 con eMMC de 16 GB, y batería Real-Time Clock (RTC)
Adaptador USB Wifi
Adaptador Bluetooth
Hub autoalimentado USB
USB GPS (BU-353-S4)
Arduino Uno Rev 3
Pen USB 16 GB
Pantalla táctil 7" Lilliput
12 Botones pulsadores

Lista de las aplicaciones que he instalado en ODROID

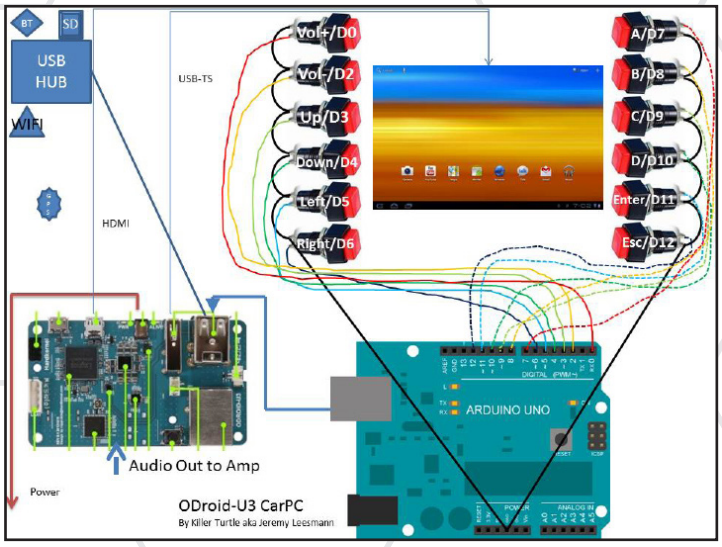
Waze (Navigación)
Pandora (Radio por Internet)
Google Play Music (MP3)
USBGPS (acceso al GPS a través del puerto USB)
Anycut (Acceso a la configuración de barra de tareas)
Nova Launcher (pantalla de inicio personalizable)
SwiftKey keyboard (o cualquier otro teclado)
GApps para Play Store
kernel adaptado a pantalla táctil (gracias a @ mdrjr por su desarrollo)
Y algunas aplicaciones de PC usadas para configurar Arduino y GPS
Arduino IDE
SiRF Demo (configurar GPS USB para la transferencia de datos)

Configurar Arduino como un teclado

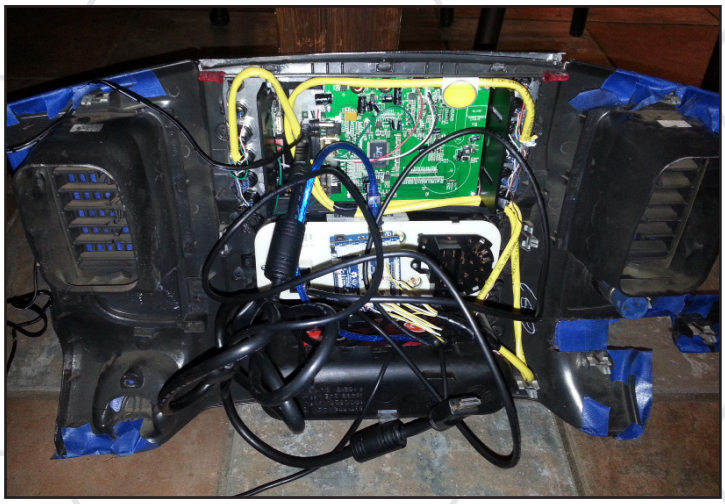
La primera tarea es conseguir que Arduino emulara un teclado USB HID. Empecé con el desarrollo de un código para Arduino, incluido al final de este artículo.

Una vez desarrollado y cargado el código, ve a http://hunt.net.nz/users/darran/ y siga las instrucciones para poner Arduino en modo DFU y prográmalo como un teclado USB HID. Como referencia, hay un mapa con todos los códigos de teclado en: http://www.usb.org/developers/devclass_docs/Hut1_11.pdf

Conecte todos los botones a un punto común, y después cada uno a su respectivo pin de la placa Arduino. Luego, todo lo que tenía que hacer es conectar Arduino a ODROID.



El Arduino Uno, cuando se combina con un ODROID-U3, facilita la conexión de botones de interfaz tan fácil como conectar los cabos sueltos



Configurar ODROID

En primer lugar, instale el kernel personalizado para la pantalla táctil. Si deseas instrucciones detalladas, dirígete al artículo sobre como convertir tu odroid en una gigante tablet de la edición de febrero de ODROID Magazine.

A continuación, instala GApps con el fin de obtener acceso a Play Store o puedes instalar Amazon Appstore. [Nota del Editor: En los foros ODROID se explica cómo instalar GApps en tu ODROID. La forma más sencilla es utilizar *Android epic Loot Software Collection*, disponible para su descarga gratuita desde los foros, que incluye el instalador de aplicaciones GAPPS para las versiones de Android 4.1.2 y 4.2.2]

Para la pantalla principal, instalé el Launcher Nova porque se ve muy bien, pero puedes usar cualquier otra aplicación similar para personalizar el escritorio. Para conseguir que los botones de Arduino funcionen como teclas de acceso directo para abrir aplicaciones, ve al Play Store e instala Anycut. Tras su instalación, agregue un acceso directo, haga clic en "activity" y elija el primer Quick Launch que se muestre (probablemente haya 3). Esto colocará un acceso directo en la pantalla principal para los ajustes de "Quick Launch". Abra la configuración y asignar los cuatro primeros a las aplicaciones deseadas. En mi caso son Waze, Pandora, reproducir música y el último para volver a la pantalla de inicio.

A continuación, instale el teclado virtual. Yo tengo SwiftKey, pero cualquier otro servirá. Una vez que el teclado funcione, vaya a Settings, Language e input, haga clic en Default y a apague el teclado de hardware. Esto permitirá que el teclado virtual funcione mientras que el teclado físico este conectado.

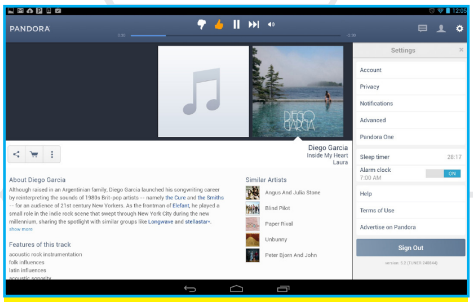
Ahora, conecta tu GPS. Si consigues uno como el que yo tengo (BU-353-S4) sigue estas instrucciones: http://bit.ly/1gzbAXr. Completa la instalación del software con cualquier otra aplicación que pueda serte de utilidad, tal como Skype o Google Hangouts.

Instalar ODROID en tu vehículo

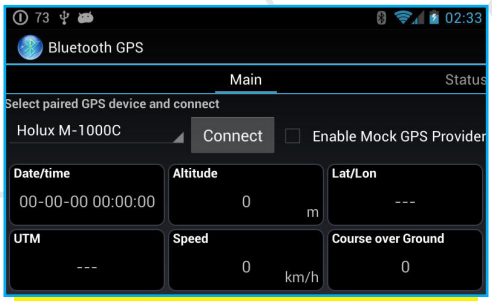
En mi camioneta, instale un enchufe de 12V conectado a una línea conmutada 12V para que funcionase la pantalla y el hub USB. Mi fuente de alimentación tiene un puerto USB de hasta 2,1 amperios que usé para la alimentar ODROID. También instale un amplificador de 400W de 4 canales, conectado a todas las conexiones estéreo y lance un cable RCA de auriculares desde la salida de audio de ODROID a la entrada del amplificador.

El ODROID se conecta a Internet mediante un punto de acceso Wifi en mi teléfono. Puede que tengas que montar el receptor GPS en el techo (o en cualquier otra área sin obstáculos) con un cable de extensión USB para asegurar una conexión estable. En mi caso, todo funcionó muy bien, y fui capaz de montar de una forma rápida y fiable un PC en mi camioneta usando un barato ODROID-U3.

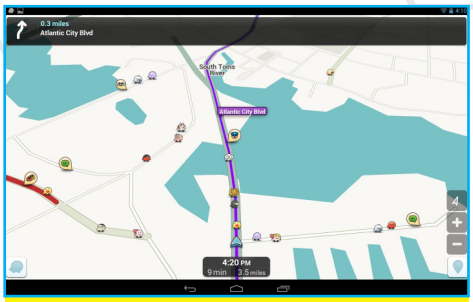
```
/* Arduino USB Keyboard HID for ODroid
* Made by Jeremy Leesmann aka Killer Turtle
*/

#define KEY_LEFT_CTRL 0x01
#define KEY_LEFT_SHIFT 0x02
#define KEY_RIGHT_CTRL 0x10
#define KEY_RIGHT_SHIFT 0x20
#define KEY_LEFT_GUI 0xE3

uint8_t buf[8] = { 0 }; /* Keyboard report
buffer */
#define PIN_VolP 7
#define PIN_VolM 8
#define PIN_Enter 5
#define PIN_Escape 6
#define PIN_Up 9
#define PIN_Down 10
#define PIN_Left 11
#define PIN_Right 12
#define PIN_A 0
```



Este Car PC está listo para una fiesta Pandora frenética en la playa!



La interfaz de GPS te asegura llegar puntualmente a tu aventura romántica por el campo!



Por qué abrirse paso entre el tráfico cuando puedes conducir sobre él con tus grandes ruedas!

```
#define PIN_B 2
#define PIN_C 3
#define PIN_D 4
#define PIN_Space 13

int state = 1;

void setup()
{
Serial.begin(9600);
pinMode(PIN_VolP, INPUT);
pinMode(PIN_VolM, INPUT);
pinMode(PIN_Enter, INPUT);
pinMode(PIN_Escape, INPUT);
pinMode(PIN_Up, INPUT);
pinMode(PIN_Down, INPUT);
pinMode(PIN_Left, INPUT);
pinMode(PIN_Right, INPUT);
pinMode(PIN_A, INPUT);
pinMode(PIN_B, INPUT);
pinMode(PIN_C, INPUT);
pinMode(PIN_D, INPUT);
pinMode(PIN_Space, INPUT);
// Enable internal pull-ups
digitalWrite(PIN_VolP, 1);
digitalWrite(PIN_VolM, 1);
digitalWrite(PIN_Enter, 1);
digitalWrite(PIN_Escape, 1);
digitalWrite(PIN_Up, 1);
digitalWrite(PIN_Down, 1);
digitalWrite(PIN_Left, 1);
digitalWrite(PIN_Right, 1);
digitalWrite(PIN_A, 1);
digitalWrite(PIN_B, 1);
digitalWrite(PIN_C, 1);
digitalWrite(PIN_D, 1);
digitalWrite(PIN_Space, 1);
delay(200);
}
void loop() {
```

```
state = digitalRead(PIN_VolP);
if (state != 1) {
buf[2] = 128; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_VolM);
if (state != 1) {
buf[2] = 129; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_Enter);
if (state != 1) {
buf[2] = 40; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_Escape);
if (state != 1) {
buf[2] = 41; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_Up);
if (state != 1) {
buf[2] = 82; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
```

Con ODROID PC para camionetas vas a cualquier parte con estilo, Incluso a tu colina favorita!!!

```
state = digitalRead(PIN_Down);
if (state != 1) {
buf[2] = 81; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_Left);
if (state != 1) {
buf[2] = 80; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_Right);
if (state != 1) {
buf[2] = 79; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_A);
if (state != 1) {
buf[0] = KEY_LEFT_GUI; // Windows Key
buf[2] = 4; // Letter a
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_B);
if (state != 1) {
buf[0] = KEY_LEFT_GUI; // Windows Key
buf[2] = 5; // Letter a
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_C);
if (state != 1) {
buf[0] = KEY_LEFT_GUI; // Windows Key
buf[2] = 6; // Letter a
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_D);
if (state != 1) {
buf[0] = KEY_LEFT_GUI; // Windows Key
buf[2] = 7; // Letter a
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
state = digitalRead(PIN_Space);
if (state != 1) {
buf[2] = 44; // Vol +
//buf[2] = 27; // Letter X
// buf[2] = 123; // Cut key: Less portable
Serial.write(buf, 8);// Ssend keypress
releaseKey();
}
}
void releaseKey()
{
buf[0] = 0;
buf[2] = 0;
Serial.write(buf, 8);// Release key
delay(500);
}
```

CONOCIENDO A UN ODROIDIAN

MARIAN MIHAILESCU: UNO DE NUESTROS PRINCIPALES COLABORADORES DEL FORO

Por favor, Háblanos un poco sobre ti.

Soy colaborador de investigación de las ciencias de la computación que actualmente vive en Adelaide, Australia. Trabajo en el Centro de Investigación de Telecomunicaciones de la Universidad de Adelaida. Originamente de Rumania, curse mis estudios de licenciatura en ciencias de la computación en Bucarest, y luego me doctore en la Universidad Nacional de Singapur.

Visitando Arapiles - Australia.

¿Cómo fueron tus inicios con los ordenadores?

Cuando era niño, vi un Spectrum ZX a un amigo de la familia y me engancharlon los juegos. Todavía me suelo aficionar a muchos de ellos, Saboteur, Dizzy, Elite o ChuckieEgg. Tuve la oportunidad de unirse a una clase especial de secundaria donde se enseñaba BASIC, y también me uní al (pequeño) club informática local. El instituto al que fui también tenía un grado de informática. Crecí en un país comunis-ta donde estaba prohibió importar tecnología, de modo que Spectrum era todo con lo que crecí. Vi mi primera PC cuando comencé el instituto.

Viajando por el Nepal a 5000M de altitud

¿Qué tipos de proyectos has realizado con tu ODROID?

Para empezar adquirí un ODROID U2 para un HTPC de bajo consumo, ya que mi Raspberry Pi era demasiado lenta. De todas las placas basadas en ARM A9 hasta el momento, ODROID era la única con cuatro núcleos con la que tenía posibilidades de conseguir que Open GLES funcionase en Linux. De hecho, actualmenete colaboro en esto y en que XBMC llege a funcionar totalmente. Gane un ODROID XU como premio mensual en los foros

Actualmente, uso el XU para algo más que HTPC, es además un servidor principal (Apache, MySQL, SSH), un sistema de descarga y un sistema de monitoreo para el hogar (detección de

Escalando a 4000M de altitud en las montañas de Malasia

Explorando playas exóticas Tailandia.

movimiento y registro de temperatura). Ahora estoy trabajando en agregar más funciones. Quiero usarlo para controlar el comedero del gato, controlar la unidad de aire acondicionado y la puerta del garaje. También está previsto integrar todas estas funciones con SiriProxy, para conseguir controlar todo ello desde mi teléfono.

Qué otras aficiones e intereses tienes?

Me encanta viajar, el montañismo, la escalada y el bulder. Vivir en Singapur me dio la oportunidad de visitar un montón de lugares al sureste de Asia, Tailandia es el lugar perfecto donde se puede disfrutar de la escalada justo a una hermosa playa. Adelaide donde vivo actualmente, también está cerca del Parque Nacional de Grampians y el Monte Arapiles, dos lugares emblemáticos para el bulder y la escalada tradicional. En cuanto al montañismo, actualmente estoy planeando mi segundo viaje a Nepal - una visita al campamento base del Everest.

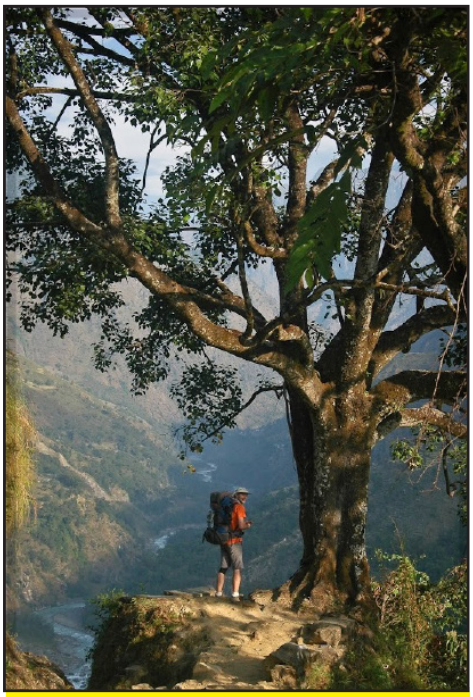
Escalando en Tailandia, disfrutando de la naturaleza y del subidón de adrenalina

¿Qué es lo que más te gusta de la comunidad ODROID?

Es la propia comunidad en sí misma la que me gusta. Si estás trabajando en un proyecto, puede pedir consejo y la gente se esfuerza en ayudarte y ofrece consejos útiles. Si tienes problemas, puedes plantearlos en los foros y si la solución no existe, se te facilita algúna alternativa.

¿Estás trabajando con cualquier otro proyecto de hardware o software a parte de ODROID?

Además de mi trabajo, mis proyectos con ODROID ocupan la mayor parte de mi tiempo. Así que la respuesta es no. Mi viejo Raspberry Pi fue completamente reemplazado por el XU. Descubro y escribo para otros, he contribuido con varios consejos prácticos disponibles en los foros, un par de artículos en ODROID Magazine y con la librería gráfica que he creado, que permite exportar datos de RRDs a Highcharts . La mayoría de las aportaciones son muy comunes y deberían funcionar en cualquier plataforma, no son específicas para la plataforma ODROID.



Desde las altas montañas de Nepal a sus valles escondidos, esperamos que Marian tuviera la oportunidad de usar su ODROID para capturar vídeo de sus espectaculares viajes.