Instalar Linux en un unidad externa • Recuperar eMMC • Desconexión GPIO

ODROID Magazine

Año Dos
Num. #16
Abr 2015

Linux KALI

Distribución avanzada de pruebas de penetración en ODROID

Monta tu propio Ambilight

- Automatiza tu casa con OpenHAB
- Triple arranque con ODROID-C1
- Dev Android: Low Memory Killer Daemon
- Juega al original Warcraft sobre Linux

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

HARDKERNEL

Pollin ELECTRONIC

Ahora estamos enviando los dispositivos ODROID U3 a los países de la UE! Ven y visita nuestra tienda online!

Dirección: Max-Pollin-Straße 1
85104 Pförring Alemania

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

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

Kali Linux es una de nuestras distribuciones favoritas, ya que incluye una amplia variedad de software para realizar pruebas de penetración de seguridad. En Seguridad ofensiva, los desarrolladores de Kali son fans de ODROID y han lanzado una versión para cada modelo hasta ahora. Kali sobre ODROID puede ser la base de un negocio lucrativo, ofreciendo algo que toda empresa necesita: tranquilidad. Aprende cómo instalar Linux Kali en ODROID-C1, y podrás ofrecer pruebas de seguridad muy asequibles con un coste mínimo en la puesta en marcha.

Además presentamos la increíble guía técnica de Venkat que muestra cómo hacer que nuestros hogares sean más inteligentes usando OpenHAB, así como un tutorial de Marian sobre Ambilight, que hace más divertido ver vídeos y escuchar música. Tobías nos trae un especial sobre cómo ejecutar la serie original de Warcraft en un hardware moderno, Nanik analiza el denominado Low Memory Killer Daemon de Android, aprenderemos cómo crear un triple arranque en un ODROID-C1 con Android, Linux y OpenELEC, ¡y mucho más!

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

HARDKERNEL'S EXCLUSIVE NORTH AMERICAN DISTRIBUTOR

ameriDroid
High-Performance Embedded Computers

All Hardkernel products in stock at AmeriDroid.com

USB GPS MODULE
$26.95

ODROID-C1
$36.95

ODROID-VU
$119.95

C1 3.2 INCH TOUCHSCREEN DISPLAY SHIELD
$26.95

ODROID Magazine

Rob Roy, Editor Jefe

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

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-U3. En cuanto a aficiones, si necesitas alguna, yo estaría encantado de ofrecerte alguna de la mías ya que tengo demasiadas. Eso ayudaría a que tuviese más tiempo para estar con mi maravillosa esposa y mis cuatro hijos estupendos.

Bruno Doiche, Editor Artístico Senior

Desempolvó todos sus libros de Dungeons and Dragons para poder inspirarse a la hora de escribir artículo Kali Linux y la cubierta. Después de eso, vio una maratón de Star Trek: The Next Generation para el artículo OpenHAB. Y al fin, llego a convertirse en "THE ONLY ONE!"

Nicole Scott, Editor Artistico

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

James LeFevour, Editor Artístico

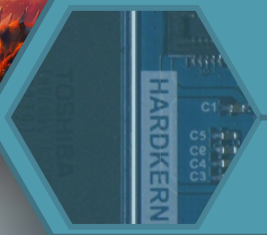
Soy un especialista en medios digitales que disfruta trabajando como freelance en marketing de redes sociales y administración de sitios web. Cuanto más aprendo sobre las posibilidades de ODROID más me ilusiona probar cosas nuevas con él. Me traslade a San Diego desde el Medio Oeste de los EE.UU. Todavía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

Manuel Adamuz, Editor Español

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

INCIDE

KALI LINUX - 6



RECUPERAR EMMC - 10

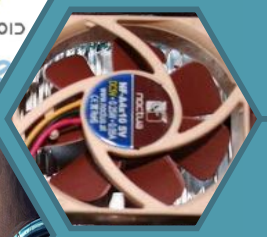
JUEGOS ANDROID: ONLY ONE - 10

MONTA TU PROPIO AMBILIGHT - 11

INSTALAR LINUX EN UNA UNIDAD USB EXTERNA - 14

OPENHAB - 16

TRIPLE ARRANQUE - 26



VENTILADOR XU3 - 27

JUEGOS LINUX: EL ORIGINAL WARCRAFT - 28

CONCURSO DE VIDEO - 31

DESARROLLO ANDROID: LMKD - 32

JUEGOS ANDROID: IRON FORCE Y BEACH BUGGY RACING - 33

CONOCIENDO A UN ODROIDIAN - 34

COMPILAR KALI LINUX

USA TU ODROID PARA CONOCER LA MAS AVANZADA DISTRIBUCION DE AUDITORIA Y SEGURIDAD INFORMATICA JAMAS CREADA

por @tmpuser y @yhfudev

Según sus desarrolladores, Kali Linux es la más avanzada y versátil distribución de test de penetración que existe. Cuenta con un completo y sorprendente kit de herramientas reunidas en una distribución de seguridad todo-en-uno, diseñada para simplificar la experiencia con los test de penetración. La familia de ordenadores ODROID es una de las favoritas para la seguridad ofensiva, los desarrolladores de Kali Linux han liberado sus distribuciones para cada uno de los modelo ODROID existentes hasta la fecha, incluyendo el U3 y XU3.

Varios han sido los usuarios de los foros ODROID los que han presentado sus propios métodos para llevar Kali al ODROID-C1 creando guías paso a paso para su instalación en una mini distribución de Debian, ArchLinux o Ubuntu. En primer lugar presento un método manual para compilar Kali, y en segundo lugar describo un simple script para instalarlo automáticamente.

Método manual

Para empezar, arranca una copia de Ubuntu o Debian y escribe los siguientes comandos en una ventana de Terminar o sesión SSH. Ten en cuenta que esto no es un script shell, así que todo debe introducirse manualmente.

```
$ sudo su -
# sudo apt-get install qemu-user-static binfmt-support lzop libncurses5-
dev dosfstools

# export architecture="armhf"
# export distro="kali"
# export path_rootfs="/mnt"
# export INSTALL_MIRROR=http.kali.org
# export INSTALL_SECURITY=security.kali.org
# PACKAGES_ARMHF="ssh,vim-tiny,nano,locales,dialog,xz-utils"
# PACKAGES_BASE="binutils console-common initramfs-tools sudo parted
e2fsprogs usbutils uboot-mkimage u-boot-tools ntfs-3g lsb-release module-
init-tools fbset ntpdate"
# PACKAGES_DESKTOP="kali-defaults gnome-core kali-root-login desktop-
base"
# PACKAGES_TOOLS="curl make bc lsof less git nmap ethtool unrar-free ca-
bextract unace unrar zip unzip p7zip p7zip-full p7zip-rar uudeview mpack
arj ncompress apt-file python-software-properties"
# PACKAGES_SERVICES="ntp"
# PACKAGES_EXTRAS="iceweasel iceweasel-l10n-en-en myspell-en wpasuppli-
cant gparted gnome-tweak-tool xserver-xorg-video-fbdev xserver-xorg-vid-
eo-mali mali-fbdev aml-libs file-roller totem totem-plugins totem-mozilla
rhythmbox deb-multimedia-keyring gedit"
# export PACKAGES="${PACKAGES_BASE} ${PACKAGES_DESKTOP} ${PACKAGES_TOOLS}
${PACKAGES_SERVICES} ${PACKAGES_EXTRAS}"
```

```
# mkdir -p ..path_rootfs/rootfs
# export pathlocal="..path_rootfs/rootfs"
# cd $pathlocal
# dd if=/dev/zero of=${pathlocal}/rootfs.img bs=1M count=4096
# mkfs.ext4 -F -O ^has_journal -E stride=2,stripe-width=1024 -b 4096 -L
rootfs -U e139ce78-9841-40fe-8823-96a304a09859 ${pathlocal}/rootfs.img
# mount -o loop ${pathlocal}/rootfs.img ${path_rootfs}

# debootstrap --verbose --foreign --arch ${architecture}
--include=${PACKAGES_ARMHF} ${distro} ${path_rootfs} "http://${INSTALL_
MIRROR}/${distro}"
# cp /usr/bin/qemu-arm-static ${path_rootfs}/usr/bin/
# DEBIAN_FRONTEND=noninteractive DEBCONF_NONINTERACTIVE_SEEN=true LC_
ALL=C LANGUAGE=C LANG=C chroot ${path_rootfs} /debootstrap/debootstrap
--second-stage -cpu cortex-a9
# DEBIAN_FRONTEND=noninteractive DEBCONF_NONINTERACTIVE_SEEN=true LC_
ALL=C LANGUAGE=C LANG=C chroot ${path_rootfs} dpkg --configure -a
# mount -t sysfs none ${path_rootfs}/sys && mount -t proc proc ${path_
rootfs}/proc && mount -B /dev ${path_rootfs}/dev && mount -B /dev/pts
${path_rootfs}/dev/pts && sync

# cat << EOF > ${path_rootfs}/etc/apt/sources.list
deb http://${ INSTALL_MIRROR }/${distro} ${distro} main contrib non-free
deb http://${ INSTALL_SECURITY }/${distro}-security ${distro}/updates
main contrib non-free
deb http://${ INSTALL_MIRROR }/${distro} ${distro}-proposed-updates main
non-free contrib
deb http://www.deb-multimedia.org wheezy main non-free
deb-src http://www.deb-multimedia.org wheezy main non-free
deb http://deb.odroid.in/c1/ trusty main
deb http://deb.odroid.in/ trusty main
EOF
# sed -i "s/^#\ en_EN/en_EN/g" ${path_rootfs}/etc/locale.gen
# echo "linux" > ${path_rootfs}/etc/hostname
# echo -e "nameserver\t8.8.8.8"> ${path_rootfs}/etc/resolv.conf
# echo -e '127.0.0.1\tlocalhost' > ${path_rootfs}/etc/hosts && echo -e
'127.0.1.1\tlinux' >> ${path_rootfs}/etc/hosts
# cat << EOF > ${path_rootfs}/etc/network/interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
EOF


# echo "KERNEL==\"mali\",SUBSYSTEM==\"misc\",MODE=\"0777\",GROUP=\"vid
eo\"" > ${path_rootfs}/etc/udev/rules.d/10-odroid_mali.rules && echo "KER
NEL==\"ump\",SUBSYSTEM==\"ump\",MODE=\"0777\",GROUP=\"video\"" >> ${path_
rootfs}/etc/udev/rules.d/10-odroid_mali.rules
# echo "KERNEL==\"ttySAC0\", SYMLINK+=\"ttyACM99\"" > ${path_rootfs}/etc/
udev/rules.d/10-odroid-shield.rules
# echo "KERNEL==\"fb1\", SYMLINK+=\"fb6\"" > ${path_rootfs}/etc/udev/
rules.d/50-odroid-hdmi.rules
# echo "KERNEL==\"CEC\", MODE=\"0777\"" > ${path_rootfs}/etc/udev/
rules.d/60-odroid-cec.rules
# echo "KERNEL==\"amstream*\",SUBSYSTEM==\"amstream-
dev\",MODE=\"0666\",GROUP=\"video\"" > ${path_rootfs}/etc/udev/
rules.d/10-odroid_am.rules && echo "KERNEL==\"amvideo*\",SUBSYSTEM==
\"video\",MODE=\"0666\",GROUP=\"video\"" >> ${path_rootfs}/etc/udev/
```

```
rules.d/10-odroid_am.rules

# cat << EOF > ${path_rootfs}/etc/fstab
LABEL=rootfs / ext4 errors=remount-ro,noatime,nodiratime 0 1
LABEL=BOOT /boot vfat defaults,ro,owner,flush,umask=000 0 0
tmpfs /tmp tmpfs nodev,nosuid,mode=1777 0 0
EOF

# cp -r ..path..modules_firmware_odroid/firmware/* ${path_rootfs}/lib/firm-
ware/
# cp -r ..path..modules_firmware_odroid/modules ${path_rootfs}/lib/
# sed -i "/exit 0/i\echo 0 > /sys/devices/platform/mesonfb/graphics/fb1/
blank" ${path_rootfs}/etc/rc.local
# echo "S0:23:respawn:/sbin/getty -L ttyS0 115200 vt102" >> ${path_root-
fs}/etc/inittab

# export MALLOC_CHECK_=0
# LC_ALL=C LANGUAGE=C LANG=C chroot $path_rootfs
# export PATH=/bin:/sbin:/usr/bin:/usr/sbin:$PATH
# apt-key adv --keyserver keyserver.ubuntu.com --recv-keys AB19BAC9
# dpkg-divert --add --local --divert /usr/sbin/invoke-rc.d.chroot --re-
name /usr/sbin/invoke-rc.d && cp /bin/true /usr/sbin/invoke-rc.d
# cat << EOF > /usr/sbin/policy-rc.d
#!/bin/sh
exit 101
EOF

# chmod +x /usr/sbin/policy-rc.d
# apt-get update && apt-get -y upgrade
# sed -i -e 's/KERNEL\!=\"eth\*|/KERNEL\!=\"/' /lib/udev/rules.d/75-per-
sistent-net-generator.rules
# cat << EOF > /tmp/debconf.set
console-common console-data/keymap/policy select Select keymap from full
list
console-common console-data/keymap/full select en-latin1-nodeadkeys
EOF
# debconf-set-selections /tmp/debconf.set && rm -f /tmp/debconf.set
# locale-gen en_EN en_EN.UTF-8 en_EN ISO-8859-1 && update-locale
LANG="en_EN.UTF-8" LANGUAGE="en_EN" LC_ALL="en_EN.UTF-8"
# dpkg-reconfigure tzdata && dpkg-reconfigure locales
# echo "root:root" | chpasswd && useradd -m -s /bin/bash -G adm,sudo,plug
dev,audio,video,cdrom,floppy,dip odroid && echo "odroid:odroid" | chpasswd

# apt-get --yes --force-yes install $PACKAGES

# mv /etc/X11/xorg.conf /etc/X11/xorg.conf_save
# cat << EOF > /etc/X11/xorg.conf
Section "Screen"
Identifier "Default Screen"
Monitor "Configured Monitor"
Device "Configured Video Device"
```

```
DefaultDepth 16
EndSection
EOF
# cat > /etc/asound.conf << _EOF_
pcm.!default {
type plug
slave {
pcm "hw:0,1"
}
}
ctl.!default {
type hw
card 0
}
_EOF_

# cat /etc/pulse/default.pa | sed s/"#load-module module-alsa-
sink"/"load-module module-alsa-sink"/g > /tmp/default.pa
# mv /tmp/default.pa /etc/pulse/default.pa
# cat /etc/pulse/default.pa | sed s/"#load-module module-alsa-source
device=hw:1,0"/"load-module module-alsa-source device=hw:0,1"/g > /tmp/
default.pa
# mv /tmp/default.pa /etc/pulse/default.pa
# update-rc.d ssh enable && rm -f /usr/sbin/policy-rc.d && rm -f /usr/
sbin/invoke-rc.d && dpkg-divert --remove --rename /usr/sbin/invoke-rc.d

# lsof |grep mnt
# /etc/init.d/dbus stop
# /etc/init.d/ssh stop

# rm -rf /root/.bash_history && exit && rm -f $path_rootfs/usr/bin/qemu*
# umount /mnt/dev/pts && umount /mnt/dev && umount /mnt/proc && umount /
mnt/sys && umount $path_rootfs && sync
```

Como último paso, sustituye /dev/sdX por el nombre del dispositivo de la tarjeta SD o módulo eMMC en la que has instalado la imagen. La partición de arranque ya debe estar instalada con <offset> reflejando el tamaño de la partición de inicio:

```
# dd if=rootfs.img of=/dev/sdX bs=1M skip=<offset>
```

Metodo de un único comando

Existe un script que permite crear la imagen Kali por completo usando un único comando, se puede descargar desde `http://bit.ly/1HdZgyn`.

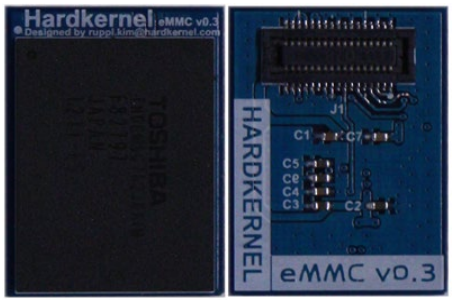
```
$ git clone https://github.com/yhfudev/arch-kali-odroidc1.git
$ cd arch-kali-odroidc1
$ sudo ./runme.sh
```

El script descarga e instala automáticamente los requisitos previos y los árboles fuente/ herramienta. Es compatible con múltiples distribuciones Linux, como Debian y ArchLinux. Para cualquier pregunta, comentario o sugerencia, visita los post originales en `http://bit.ly/1CEYLe8` y `http://bit.ly/18Oxfig`.

RECUPERAR UNA EMMC

APROVECHA Y UTILIZA EL BLOQUE FLASH PARA RECUPERAR TUS DATOS

por Bo Lechnowsky



Recientemente he perdido cerca de seis meses de trabajo con un importante script por una serie de desafortunados acontecimientos relacionados con mi backup y el delimitador -f de la línea de comandos, debería haber sido más precavido (¡oops!).

¿Sabías que siempre que guardas un archivo, en la mayoría de los casos, la nueva copia de ese archivo se encuentra en un bloque diferente del módulo eMMC o microSD? Esta cuestión me salvó de un montón de quebraderos de cabeza (literalmente). Para recuperar un archivo de texto perdido, siempre y cuando el medio en el que se encuentre siga intacto y tenga algo de espacio libre, el siguiente procedimiento en Ubuntu te puede ayudar a recuperar ese archivo de nuevo. Inserta una unidad flash USB externa para andar con cuidado y no sobrescribir cualquier resto de tu archivo.

Escriba el siguiente comando en una ventana de terminal:

```
$ df -h
```

Toma nota de la partición donde está guardado el archivo, debería ser similar a `/dev/mmcblk0p2`. Si has conectado una unidad flash, toma nota también de donde está montada, algo así como `/dev/sda`.

Intenta recordar algo de texto que sea único en el archivo que estabas editando. Además, trata de estimar el número de líneas del archivo. Es mejor que tires por lo alto, por ejemplo, si crees que el archivo tenía unas 500 líneas, usa mejor 1000.

Ejecuta el siguiente comando en el terminal:

```
$ sudo grep -a -A1000 -B1000 'bit of unique text here' /dev/mmcblk0p2 |
strings > recovered_file
```

Sustituye los dos números "1000" por tu estimación sobre el tamaño del archivo, el fragmento "bit of unique text here" por el texto único que recuerdes del archivo, "`/dev/mmcblk0p2`" por el nombre del dispositivo en el que está almacenado tu archivo, y "`recovered_file`"por la ruta donde quieres guardar el archivo, como puede ser la ruta a tu unidad flash USB.

Ten en cuenta que esto también recupera los bloques que están alrededor del texto que buscas, por lo que verás un montón de galimatías, pero también deberías ver fragmentos de tu archivo, o en algunos casos, el archivo completo, limitado por caracteres aleatorios. También ten en cuenta que, en ocasiones, los fragmentos aparecerán en orden inverso (ej. línea 3 -> línea 2 -> línea 1). Cuando utilice este método, me alegré un motón al ver mi archivo y no me importaba el orden en el que aparecían las líneas, con tal de poder recuperar mi trabajo.

Por último, los fragmentos más recientes suelen aparecer al final del archivo, así que empecé a reconstruirlo desde el final hacia atrás. Mi archivo recuperado tenía unas 239 líneas, ¡Pero es que había mucho trabajo en esas 239 líneas!

ONLY ONE

UN HOMENAJE AL GENERO DE ACCION COMO SIEMPRE HAS QUERIDO

por Bruno Doiche

Esto es todo amigos! Only One es el típico juego que te pone de los nervios: eres un héroe de 8 bits atrapado en lo alto una gigantesca torre donde te apoderas de una espada. Subes y gritas "Me convertiré en el Only One!". No he logrado encontrar a una sola persona que no se haya enganchado a esta aventura. Encontrarás alguna que otra sorpresa que, al igual que en los mejores juegos, no te ayudará pero resaltara tus mejores cualidades, haciéndote pensar rápido y astutamente. ¿Quieres empujar a tus enemigos al juicio final en vez de conseguir su fuerza? ¡Haz lo que quieras! ¿Quieres liquidar a otros y robar sus escudos? ¿No? ¡Muy bien! Simplemente asegúrese de luchar contra todos los enemigos de la mejor forma en los 90 niveles de pura diversión. ¡Ah, la banda sonora es muy famosa!

`https://play.google.com/store/apps/details?id=com.rebelbinary.onlyone&hl=en`

¡No consiguen rodearte, a menos que estés guardando fuerzas suficientes como para hacer que estos tipos corran por sus vidas!

MONTA TU PROPIO AMBILIGHT

AÑADE UNA DIMENSION EXTRA A TUS VIDEOS

por Marian Mihailescu

Ambilight es capaz de producir asombrosos efectos de iluminación, y puede sincronizarse con la música o el vídeo que estes reproduciendo en tu sorprendente centro multimedia ODROID

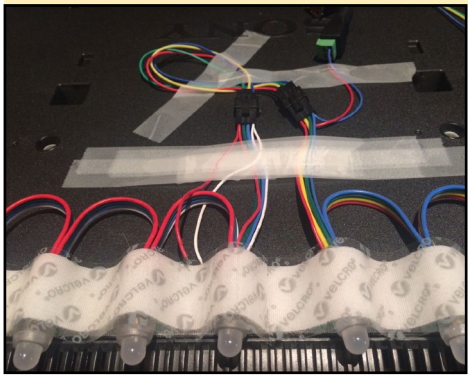
Ambilight es un sistema de iluminación ambiental para la televisión desarrollado por Philips, con el cual se crean efectos de luz alrededor de la TV en concordancia con el contenido del vídeo. Puedes lograr un efecto similar usando una tira de LEDs RGB y el software que muestrea la imagen de la pantalla, de modo que cada LED va cambiando de color en consecuencia. En este artículo, vamos a ver cómo utilizar ODROID para lograrlo.

Este artículo está ligeramente basado en el Proyecto Adalight (http://bit.ly/1EnG6zZ), que utiliza un Arduino para controlar los LEDs. En lugar de Arduino, utilizaremos un ODROID que ejecute XBMC como interfaz para reproducir el contenido del vídeo. El resto de componentes son:

- **Una tira de LED (al igual que el Adalight, puedes utilizar LEDs WS2801 de 12mm)**
- **Adaptador de corriente de 5V/2A - suficiente para 50 LEDs; para 100 LEDs necesitarás un adaptador de corriente de 5V/4A**
- **Un conector de alimentación hembra de 2.1mm**

Estos componentes son suficientes para ODROID-C1, pero si usas un modelo ODROID basado en un SoC Samsung Exynos como el ODROID-XU, también necesitarán un conmutador de nivel, como el Freetronics Logic Level Converter (`http://bit.ly/1wWkMEb`). Los LEDs se pueden comprar en Adafruit o en eBay (`http://bit.ly/1GYRjgr`). Puede utilizar una tira o combinar varias tiras. La figura 1 muestra cómo conectar los extremos de dos tiras de 25 LEDs.

Figura 1 - Conexión de dos tiras de LED, que se pueden comprar online en Adafruit o en Ebay.



Instalar el hardware es muy sencillo: conecta la entrada de la primera tira de LED al ODROID utilizando 3 cables:

- **Ground (GND: pin 6 en Odroid-CI, pin 2 en Odroid-XU),**
- **SPI clock (SCLK: pin 23 en Odroid-CI, pin 10 en Odroid-XU)**
- **SPI data (MOSI: pin 19 en Odroid-CI, pin 7 en Odroid-XU)**

En el caso del SoC Exynos, estos 3 cables están conectados a GND, Low I/O 1 y Low I/O 2 del conmutador de nivel lógico, con dos cables adicionales:

5V (pin 1 en ODROID-XU), conectado a High V IN
1.8V (pin 29 en ODROID-XU) conectado a Low V IN

Desde el conmutador de nivel lógico, conecta la tira de LED con tres cables: GND, High V I/O 1 (reloj) y High V I/O 2 (datos). Para alimentar la tira de LED conecta los cables rojo y azul al extremo de salida de la tira. Inserta los cables en la clavija de alimentación y conecta la alimentación insertando el adaptador en la clavija. En la Figura 2 se muestra el diagrama de conexiones.

El siguiente paso que puede ser algo más difícil, es el montaje de los LED sobre la TV. Si dispones de las herramientas adecuadas, podrías hacer una estructura de metal, madera o en 3D con los huecos para cada LED y montarla sobre el televisor usando los agujeros VESA. En mi caso utilice una cinta velcro de doble cara: por un lado pegue la cinta al TV situando los LEDs por encima de la cinta, por el otro coloque los LEDs en la posición correcta, como se muestra en la Figura 3.

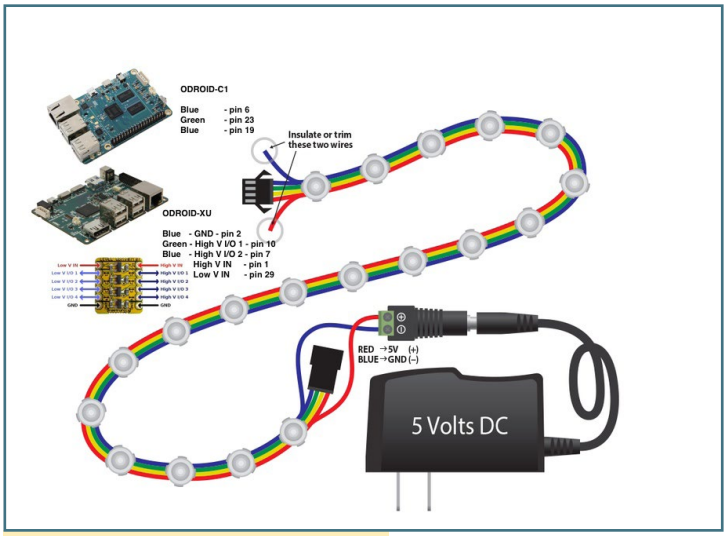


Figura 2 – Diagrama de conexiones

Para el software, utilice el demonio boblight, disponible en `http://bit.ly/1bdW7S9`, e instalé el correspondiente complemento para XBMC boblight. Después configuré, compilé e instalé boblight escribiendo los siguientes comandos:

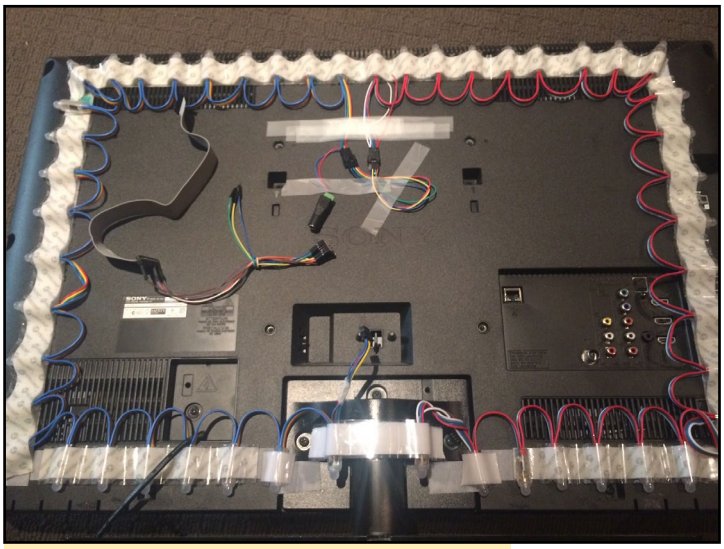


Figura 3 –LEDs instalados con cinta velcro

```
./configure –without-portaudio --without-x11 --with-
out-libusb
make; sudo make install
```

Boblight necesita un fichero de configuración, que se puede fácilmente generar con Boblight Config Maker, un programa en el que se especifica el número de LEDs que usas y su posición en el televisor. Para el ODROID-C1, la interfaz de salida es `/dev/spidev0.0`, mientras que para el ODROID-XU es `/dev/spidev1.0`:

```
# config file created with BobLight Config Maker v.1.0
# (C) Hans Luijten - http://www.tweaking4all.com
# Orientation naming as seen from the REAR of the TV
# Date: 16-8-14 01:10:39

[global]
interface 127.0.0.1
port      19333

[device]
name        ODroidLight
type        ws2801
channels    150
output      /dev/spidev1.0
interval    20000
rate        48000
```

Para que el demonio boblight se inicie en el arranque, añade el siguiente script (Ubuntu 14.04) en /etc/init/boblight.conf:

```
#!upstart
description “Boblight daemon”

env USER=root
```

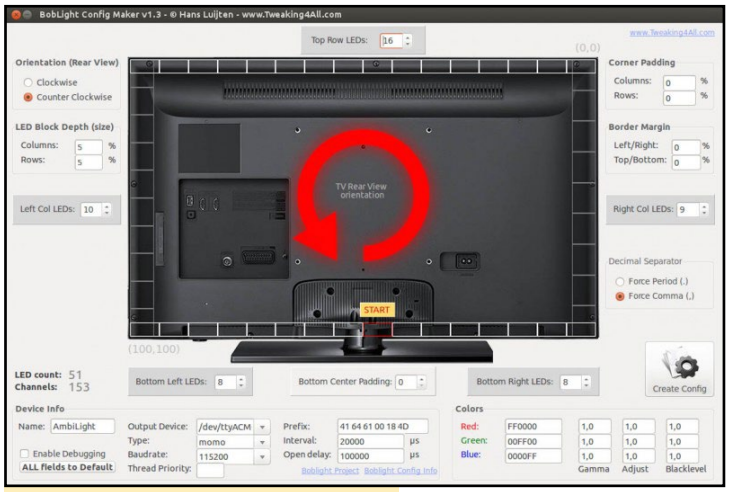


Figura 4 - Boblight Config Maker

```
env GROUP=root
env DAEMON="/usr/local/bin/boblightd"
env DAEMON_ARGS=""

start on runlevel [2345]
stop on runlevel [06]

respawn
respawn limit 15 5

expect daemon

exec start-stop-daemon --start --quiet -b --make-
pidfile --pidfile /run/boblightd.pid --chuid $USER -g
$GROUP --exec $DAEMON -- $DAEMON_ARGS

pre-stop exec start-stop-daemon --stop --signal TERM
--pidfile /run/boblightd.pid --quiet --exec $DAEMON
```

Para habilitar la interfaz SPI en ODROID-XU, es necesario realizar el siguiente ajuste del kernel:

```
Author: memeka <mihailescu2m@gmail.com>
Date:   Sun Aug 31 09:01:52 2014 +0930

    enable spi pins on spidev

diff --git a/arch/arm/mach-exynos/board-odroidxu-
ioboard.c b/arch/arm/mach-exynos/board-odroidxu-
ioboard.c
index 6157e07..f643003 100644
--- a/arch/arm/mach-exynos/board-odroidxu-ioboard.c
+++ b/arch/arm/mach-exynos/board-odroidxu-ioboard.c
@@ -84,24 +84,24 @@ static struct   platform_device
odroidxu_ioboard_adc = {
```

```
 static struct s3c64xx_spi_csinfo spi1_csi[] = {
        [0] = {
+               .fb_delay = 0x2,
                .line       = EXYNOS5410_GPA2(5),
                .set_level  = gpio_set_value,
        },
 };
 static struct spi_board_info spi1_board_info[] __
initdata = {
        {
-               .modalias                       =
"ioboard-spi",
-               .platform_data          = NULL,
-               .max_speed_hz           = 20 * 1000 *
1000,     // 20 Mhz
+               .modalias                       =
"spidev",
+               .max_speed_hz           = 40 * 1000 *
1000,     // 20 Mhz
                .bus_num                        = 1,
                .chip_select            = 0,
-               .mode                           =
SPI_MODE_0,
+               .mode                           =
SPI_MODE_3,
                .controller_data    = &spi1_csi[0],
        }
 };

 static struct platform_device odroidxu_ioboard_spi
= {
-       .name                   = "ioboard-spi",
+       .name                   = "spidev",
        .id                     = -1,
 };
```

Puede que se nos haya ido un poco la pinza a la hora de crear nuestro propio Ambilight

INSTALAR LINUX EN UNA UNIDAD EXTERNA USB

CONSIGUE MAS ESPACIO EN TU PARTICION ROOT

por @Guso

En la edición de marzo de 2014 de ODROID Magazine presenté un método para ejecutar un sistema operativo desde una unidad USB externa, compatible con los anteriores modelos ODROID, como el X, X2 y U2. Sin embargo, ahora hay una forma más sencilla de hacerlo, ya que Hardkernel ha actualizado su gestor de arranque que utiliza un script boot.ini legible en lugar de un archivo boot.scr compilado, que se puede utilizar con los nuevos modelso XU3, U3 y C1. Esta técnica te permite utilizar cualquier dispositivo externo para la partición principal de Linux independientemente del tamaño, permitiendo expandirla más allá del límite de 64 GB de la tarjeta microSD interna o módulo eMMC.

Puesto que los ODROIDs están diseñados para arrancar desde un almacenamiento interno, es necesario que la partición de arranque se encuentre en una tarjeta microSD o módulo eMMC. Sin embargo, la partición root se puede mover a una unidad USB, permitiendo al ODROID utilizar únicamente la unidad interna para el arranque inicial.

Para empezar, graba la imagen en la tarjeta microSD o módulo eMMC, luego grabarla también en la unidad USB externa utilizando Win32DiskImager o la utilidad dd de Linux. Inserta la tarjeta microSD o módulo eMMC en el ODROID, conecta la unidad USB externa y arranca el dispositivo.

Una vez que se haya completado el proceso de arranque, toma nota del nombre del dispositivo USB externo escribiendo el siguiente comando en una ventana de terminal:

```
$ sudo fdisk -l

[...]

Disk /dev/sda: 1000.2 GB, 1000204885504 bytes
255 heads, 63 sectors/track, 121601 cylinders, total
1953525167 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00074666
```

En este ejemplo, el dispositivo está identificado como /dev/sda. A continuación, determina el UUID de la unidad USB externa, haciendo coincidir el identificador del dispositivo con el resultado del comando blkid:

```
$ sudo blkid
/dev/mmcblk0p1: SEC_TYPE="msdos" LABEL="BOOT"
UUID="6E35-5356" TYPE="vfat"
/dev/mmcblk0p2: LABEL="trusty" UUID="e139ce78-9841-
40fe-8823-96a304a09859" TYPE="ext4"
/dev/sda1: LABEL="MyLabel" UUID="213C-1DF1"
TYPE="ext4"
```

En este caso, el UUID de /dev/sda1 es 213C-1DF1. Abre el fichero /media/boot/boot.ini almacenado en la tarjeta microSD o módulo eMMC y busca la siguiente línea:

```
setenv bootrootfs "console=tty1
console=ttySAC2,115200n8 root=UUID=e139ce78-9841-
40fe-8823-96a304a09859 rootwait ro"
```

Edita la variable "root" para que coincida con el identificador UUID anotado en el paso anterior y guarda el archivo:

```
setenv bootrootfs "console=tty1
console=ttySAC2,115200n8 root=UUID=213C-1DF1 rootwait
ro"
```

Finalmente, edita la entrada del archivo /etc/fstab para la partición "/" para que también coincida con la nueva unidad. Este archivo debe ser editado en la unidad externa, y no en la tarjeta microSD o módulo eMMC. Asegúrese de sustituir el identificador de dispositivo real por /dev/sda1:

```
$ mkdir ~/temp
$ sudo mount /dev/sda1 ~/temp
$ cd temp/etc
$ sudo nano fstab
```

La primera línea del contenido original del archivo fstab:

```
UUID=e139ce78-9841-40fe-8823-96a304a09859 / ext4
errors=remount-ro,noatime,nodiratime 0 1
```

Cambia la primera línea del archivo fstab para utilizar el UUID de la unidad USB externa:

```
UUID=213C-1DF1 / ext4 errors=remount-ro,noatime,nodiratime 0 1
```

Tras guardar el archivo, reinicia el ODROID y comprueba que la unidad USB externa está siendo usada como partición root:

```
$ blkid -o list
device        fs_type label     mount point   UUID
------------------------------------------------------------------------
/dev/mmcblk0p1
              vfat    BOOT      /media/boot   6E35-5356
/dev/sda1
              ext4    MyLabel   /             213C-1DF1
```

Para obtener más información o enviar comentarios, preguntas y sugerencias, por favor visita el post original en t http://bit.ly/18DTdo7.

DESCONEXION GPIO

AÑADIR UN SISTEMA DE APAGADO EXTERNO PARA DETENER TU DISPOSITIVO CON UN SIMPLE TOQUE

por Mauro Ribeiro y @joerg

Si dispones de un ODROID que estás utilizando a modo de quiosco, como un ordenador portátil, un ordenador para el coche o como una consola arcade, puede serte útil añadir un botón externo para apagarlo y así evitar daños en el soporte de almacenamiento, que suele ser difícil de reparar en sistemas integrados.

Versión Linux

Para activar la versión para Linux del script, utiliza un botón interruptor de dos pines que se puede adquirir en cualquier tienda de electrónica. Conecta el botón entre los pines GPIO 15 y 17. A continuación, crea un archivo BASH con el siguiente contenido y añade una referencia a al script dentro del archivo /etc/rc.local para que se ejecute durante el arranque:

```
#!/bin/bash

# PIN 15 -> GPIO 115
# CONNECTION is PIN17(3.3V) to PUSH-BUTTON to PIN15
GPIO_N=115

# Setup GPIO for INPUT!
echo $GPIO_N > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio$GPIO_N/direction

while true; do

    GPIO_VALUE=`cat /sys/class/gpio/gpio$GPIO_N/value`

    if [ $GPIO_VALUE -eq 1 ]; then

        # Do something here if the button is pressed
        # like reboot, poweroff...

    fi

    sleep 1

done
```

Tras añadir tus propias modificaciones indicadas en los comentarios del script, puedes ejecutar cualquier comando de Linux cuando pulses botón, como por ejemplo un script de cierre que te asegura que todos los programas residentes se cierran con seguridad antes de que apague el dispositivo.

Versión Android

La versión APK de Android, que se puede descargar desde `http://bit.ly/1xgUHAl` permite cerrar o reiniciar usando el Pin 35 (GPIO97) y el Pin 36 (GPIO98). La aplicación controla las dos entradas, que se activan al cambiar a 0V usando una resistencia de 2K. El Pin 35 provoca el cierre y Pin 36 inicia el reinicio. Ten en cuenta que la aplicación no tiene una interfaz de usuario, por lo que no verás un icono de escritorio.

Reinicia tras instalar la APK y concede permisos de root a la aplicación cuando te lo solicite. El servicio se iniciará automáticamente aproximadamente 30 segundos después de que se inicie Android. Para obtener más información o enviar comentarios, preguntas y sugerencias, por favor visite los post originales en `http://bit.ly/1FxyMWQ` y `http://bit.ly/1Ex9Wo8`.

OPENHAB

UNA GUIA SOBRE DOMOTICA DE CODIGO ABIERTO

por Venkat Bommakanti

En la edición de septiembre 2014 de ODROID Magazine, presentamos la plataforma de software Freedomotic, que utiliza la arquitectura del Internet de las Cosas (IoT) para desarrollar y gestionar espacios inteligentes. En este artículo, presentamos otra interesante plataforma de código abierto llamada open-HAB, un completo entrono de trabajo de domótica basado en java, compatible con OSGi y que se gestiona a través de reglas. Puesto que es una solución independiente de la tecnología y del proveedor utilizado, es una aplicación perfecta para su integración con la plataforma ODROID.

Con el fin de estar lo más actualizado posible, este artículo proporciona los pasos necesarios para implementar un reciente desarrollo de la próxima versión de openHab (v1.7.0), usando paquetes precompilados disponibles al público en general. Sin embargo, no cubre:

- **El uso de los paquetes precompilados de la última versión estable (1.6.2)**
- **La compilación desde el código fuente (GitHub). Este método se intentó con las última versión 1.7.0, pero la compilación falló en el CI, el U3 y en los sistemas Linux x86, a pesar de seguir el procedimiento recomendado de openHAB, y debido a problemas a la hora de buscar los componentes específicos para el desarrollo del openHAB, como Maven y Eclipse, no compatibles con el hardware.**
- *** Instalación de la versión alpha 2.0.0 de nueva generación**

Requisitos

1. **Un ODROID-C1, aunque también puedes usar un ODROID de mayor potencia.**
2. **Accesorios para el C1 como son el cable HDMI, cable ethernet CAT 5E + o dongle wifi 3. Recomendados PSU, batería RTC y ODROID-VU o monitor HD.**
3. **Un módulo eMMC 5.0 de 16GB + con la última imagen de escritorio Lubuntu específica para C1 o una MicroSD de Clase 10 de 16 GB o más, con un adaptador de tarjeta SD compatible.**
4. **Software Wiringpi y Kit C-Tinkering que se utilizarán para simular los sensores y las luces en una vivienda.**
5. **Sistema de sonido Sonos WiFi para su integración con el sistema multimedia.**
6. **Una red en la que el dispositivo tenga acceso a Internet.**
7. **Oracle Java 8**
8. **Software openHAB (Release Candidate: 1.7.0 Build # 879).**
9. **Acceso vía intranet al C1 a través de utilidades como PuTTY, FileZilla, TightVNC Viewer o Terminal que se puedan ejecutar en otro equipo.**

Instalar Lubuntu

Copia la última imagen ODROID-C1 al módulo eMMC o tarjeta SD, insértala en el dispositivo y arranca el sistema con el monitor conectado. Ejecuta ODROID Utility y configura la resolución de pantalla para que se corresponda con el monitor, después reinicie. A continuación, ejecuta ODROID Utility d de nuevo y amplia la partición de instalación para usar toda la eMMC seleccionando la opción " Resize your root partition". Reinicie de nuevo, ejecuta ODROID Utility una vez más para configurar y actualizar el resto de cuestiones del sistema, como son el kernel y los drivers u-boot, continua con otro reinicio.

Despues, actualiza las aplicaciones de Ubuntu y reinicie el ODROID cuando hayan finalizado:

```
$ sudo apt-get autoremove && sudo apt-get update
$ sudo apt-get dist-upgrade && sudo apt-get upgrade
$ sudo apt-get install linux-image-c1
```

Por último, comprueba la versión del sistema desde un Terminal para asegurarte de que tiene la última versión:

```
$ uname -a
Linux odroid 3.10.70-74 #1 SMP PREEMPT Wed Mar 4
04:13:57 BRT 2015 armv7l armv7l armv7l GNU/Linux
```

Verificar hard float

Se puede comprobar de varias formas que está instalada la versión hard float de Linux. En primer lugar, se puede verificar la información del Executable and Linkable Format (ELF) de la imagen instalada::

```
$ readelf -a /usr/bin/readelf | grep arm
      [Requesting program interpreter: /lib/ld-linux-
armhf.so.3]
 0x00000001 (NEEDED) Shared library: [ld-linux-armhf.
so.3]
  000000: Version: 1  File: ld-linux-armhf.so.3  Cnt:
1
```

Toma nota de la descripción de la librería compartida hard float (armhf). Esto confirma la presencia de un sistema operativo (SO) basado en hard float. Otra posibilidad para verificar hard float es inspeccionar los paquetes instalados en el sistema. El valor de la arquitectura para cada uno de los paquetes aparecerá junto con el nombre usando el siguiente comando:

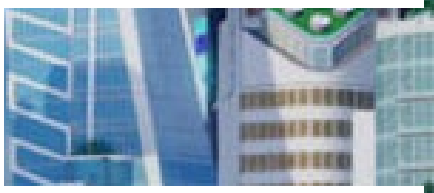
```
$ dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-
inst/trig-aWait/Trig-pend
```

```
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name               Version        Architecture   Description
+++-==================-==============-==============-==============================
ii  abiword            3.0.0-4ubuntu1 armhf          efficient, featureful word
ii  abiword-common     3.0.0-4ubuntu1 all            efficient, featureful word
…
```

Toma nota de la descripción de la arquitectura hard-float (armhf/all) de la lista. Finalmente, las librerías instaladas también se pueden utilizar para verificar la arquitectura hard float buscando la etiqueta "gnueabihf":

```
$ sudo ls -lsa /lib
…
12 drwxr-xr-x  4 root root 12288 Mar 16 12:23 arm-linux-gnueabihf
…
```

Verificar versión de Java



OpenHAB necesita la versión 8 de Java Orable, que debería estar instalada de forma predeterminada con el Lubuntu oficial, y que se puede comprobar mediante los siguientes comandos:

```
$ sudo update-alternatives --config java
There are 3 choices for the alternative java (providing /usr/bin/java).
Selection  Path           Priority  Status
------------------------------------------------------------------------------
* 0  /usr/lib/jvm/java-8-oracle/jre/bin/java   1067   auto mode
  1  /lib/jvm/jdk1.8.0/bin/java       180    manual mode
  2  /usr/lib/jvm/java-7-openjdk-armhf/jre/bin/java  1063   manual mode
  3  /usr/lib/jvm/java-8-oracle/jre/bin/java   1067   manual mode
Press enter to keep the current choice[*], or type selection number:
```

Pulsa la tecla Enter para salir, tras comprobar que la versión actual identificada por un asterisco es la 8 de Oracle. Otro método para verificar Java es utilizar el comando "which":

```
$ which java
/usr/bin/java
$ java -version
java version "1.8.0_33"
Java(TM) SE Runtime Environment (build 1.8.0_33-b05)
Java HotSpot(TM) Client VM (build 25.33-b05, mixed
mode)
```

Asegúrate de que la versión no está anulada comprobando la variable de entorno $ JAVA_HOME, que debería mostrar un resultado null:

```
$ echo $JAVA_HOME
```

Si aparece algún resultado, entonces es que la variable del entorno está configurada para otra versión de Java, y se debería desactivar antes de ejecutar openHAB. Reinicia tras actualizar

la variable y comprueba de nuevo que la versión 8 de Oracle es la que está en uso.

Descargar openHAB

Los tres (3) componentes principales del software openHAB son:

Runtime: Core, Addons y Demo (optional),
Designer
Client UI: Android, iOS & WebUI

En este artículo, hemos optado por instalar la versión 1.7.0 (Build # 879) del software openHAB, que era la más reciente cuando se escribió este artículo. Se puede obtener desde `http://bit.ly/1CJO36d`, como se muestra en la Figura 1.

Descarga los siguientes componentes (Linux 32 bits) en el directorio ~/Downloads:

```
distribution-1.7.0-SNAPSHOT-addons.zip
distribution-1.7.0-SNAPSHOT-demo.zip
distribution-1.7.0-SNAPSHOT-designer-linux.zip
distribution-1.7.0-SNAPSHOT-greent.zip
distribution-1.7.0-SNAPSHOT-runtime.zip
```

Si deseas utilizar los archivos de la versión estable 1.6.2, descarga los archivos desde http://bit.ly/1EBtbe2.

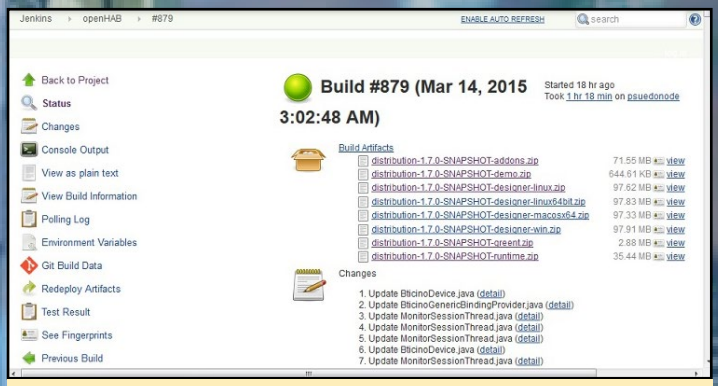


Figura 1: Descarga del software openHAB (Proyecto en curso)

Instalar openHAB

Tras descargar los archivos, crea un directorio temporal para almacenar todo el software relacionado. Esta carpeta temporal se puede mover luego a la raíz del sistema (/) para que pueda acceder cualquier usuario:

```
$ cd ~ && mkdir openhab && cd openhab
$ mv ~/Downloads/distribution-1.7.0-SNAPSHOT*.zip .
```

A continuación, descomprime el archivo zip y verifica los archivos, limpiando los directorios en cada paso:

```
$ unzip distribution-1.7.0-SNAPSHOT-runtime.zip
$ ls -lsa

…
4 drwxrwxr-x 2 odroid odroid  4096 Mar 14 03:03 addons
4 drwxrwxr-x 8 odroid odroid  4096 Mar 14 03:03 configurations
4 drwxrwxr-x 2 odroid odroid  4096 Mar 14 03:03 contexts
4 drwxrwxr-x 2 odroid odroid  4096 Mar 14 03:03 etc
12 -rw-rw-r-- 1 odroid odroid       11232 Mar 14 03:03 LICENSE.TXT
4 -rw-rw-r-- 1 odroid odroid   626 Mar 14 03:03 README.TXT
4 drwxrwxrwx 6 odroid odroid  4096 Mar 14 04:04 server
4 drwxrwxr-x 2 odroid odroid  4096 Mar 14 03:03 sounds
4 -rw-rw-r-- 1 odroid odroid   994 Mar 14 03:03 start.bat
4 -rw-rw-r-- 1 odroid odroid  1146 Mar 14 03:03 start_debug.bat
```

```
4 -rwxr-xr-x 1 odroid odroid  1132 Mar 14 03:03 start_
debug.sh
4 -rwxr-xr-x 1 odroid odroid   969 Mar 14 03:03 start.sh
4 drwxrwxr-x 4 odroid odroid  4096 Mar 14 03:03 webapps

$ rm distribution-1.7.0-SNAPSHOT-runtime.zip
```

Luego Descomprime los complementos descargados:

```
$ mv distribution-1.7.0-SNAPSHOT-addons.zip addons/
$ cd addons
$ unzip distribution-1.7.0-SNAPSHOT-addons.zip

$ rm distribution-1.7.0-SNAPSHOT-addons.zip
$ cd ..
```

Aunque hay muchos complementos, no todos serán necesarios para la puesta en funcionamiento. Los complementos también se pueden añadir o reinstalar en cualquier momento en esta carpeta, aunque el uso de recursos se incrementa con cada complemente que se añada al sistema. Por último, crea una configuración básica para openHAB:

```
$ cd ~/openhab/configurations
$ cp openhab_default.cfg openhab.cfg
$ mv openhab_default.cfg openhab_default-orig.cfg
```

Editaremos el fichero de configuración openhab.cfg más tarde, presente durante el arranque y en tiempo de ejecución, mientras que experimentamos con nuestra instalación.

Instalar la demo

La aplicación demo se puede instalar tras respaldar algunos archivos con nombres similares a los utilizados en la aplicación:

```
$ cd ~/openhab
$ mv README.TXT README-openhab.TXT
$ unzip distribution-1.7.0-SNAPSHOT-demo.zip
```

Se te preguntara si es seguro sobrescribir los archivos existentes con el mismo nombre, Selecciona Y (para sí) para autorizar que se sobreescriban.

Preparar el experimento

Para validar la instalación de openHAB de nuestro experimento, vamos a realizar las siguientes pruebas:

Utiliza el kit C-Tinkering para simular un sensor de movimiento en nuestro taller de electrónica de última generación. El sensor de luz juega el papel de sensor de movimiento. Cuando alguien se acerca al sensor de movimiento, se enciende un LED simulando una alarma o una luz. El sistema de alarma será muy básico, integrado con la interfaz web de openHAB en tiempo real. Para esta prueba, vamos a utilizar el complemento GPIO integrado.

Manejar un sistema de sonido Sonos WiFi, pausando y reanudando la música en un canal Pandora, y visualizar el nombre de la pista que se este reproduciendo. Para esta prueba, vamos a utilizar el complemento Sonos. En este experimento, vamos a abordar dos aspectos básicos de una típica instalación:

Integración de sistemas con sensores y detectores
Integración de sistemas fabricados de serie

El otro aspecto que queda por cubrir es la integración de los complementos desarrollados por el usuario. Este artículo debería contribuir en gran medida a que empieces con esta tarea.

Usaremos la aplicación demo para simular una casa con dos plantas (baja y sótano), donde el sótano representa el taller de electrónica con sensor de movimiento, y la planta baja una sala de estar donde se configurará el sistema Sonos. Para el sensor de movimiento, sigue los pasos del kit C-tinkering configurado en el artículo del Osciloscopio publicado en la edición de marzo 2015 de ODROID Magazine. Prueba la configuración que se describe asegurándote que el LED se enciende al pasar un dedo por el sensor de luz. Esto simulará la detección de movimiento, de modo que si alguien se mueve cerca del sensor, la alarma se activará, representada mediante el encendido del LED.

En una ventana de terminal, ejecuta la aplicación de ejemplo que fue utilizada en el artículo del osciloscopio. En otra ventana de terminal, ejecuta la utilidad GPIO que fue compilada cuando se instaló la librería Wiringpi:

```
$ sudo ./gpio readall
```

El LED se conecta al GPIO # 101, que corresponde al pin # 21 de Wiringpi. Siempre que el sensor no esté bloqueado, verás el siguiente resultado (valor de estado: 0, LOW):

```
+------+-----+----------+------+- Model ODROID-C -+------+----------+-----+------+
| GPIO | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | GPIO |
+------+-----+----------+------+---+----++----+---+------+----------+-----+------+
…
|  101 |  21 | GPIO.101 |  OUT | 0 | 29 || 30 |   |      | 0v   |     |      |
…
```

Siempre que el sensor este bloqueado, verás este resultado (valor de estado: 1, HIGH):

```
+------+-----+----------+------+- Model ODROID-C -+------+----------+-----+------+
| GPIO | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | GPIO |
+------+-----+----------+------+---+----++----+---+------+----------+-----+------+
…
|  101 |  21 | GPIO.101 |  OUT | 1 | 29 || 30 |   |      | 0v   |     |      |
…
```

Para la segunda parte del experimento con Sonos, conecta Sonos a la red para que sea visible al C1, y pon en marcha el sistema de altavoces Sonos. Espera hasta que todos los LEDs

estén en intermitente en los dos dispositivos. En ese momento, podrás buscar las direcciones IP de estos dos dispositivos utilizando la interfaz de tu router o la interfaz de usuario de Windows 7 de Sonos (Figura 2), valores importantes marcados con un círculo rojo.

La dirección IP del altavoz Sonos en este ejemplo es 192.168.7.45, con número de serie 00: 0E: 58: FB: 99: A2. Escribiendo en el navegador http://192.168.7.45:1400/status/topology y sustituyendo la IP por la tuya, verá una información similar a la mostrada en la figura 3.

Toma nota de la información UID del altavoz, que es RINCON_000E58FB99A201400 en este ejemplo. Casualmente, se crea utilizando el número de serie del dispositivo. Esta información UID se utilizará más adelante, durante la etapa de configuración de openHAB.

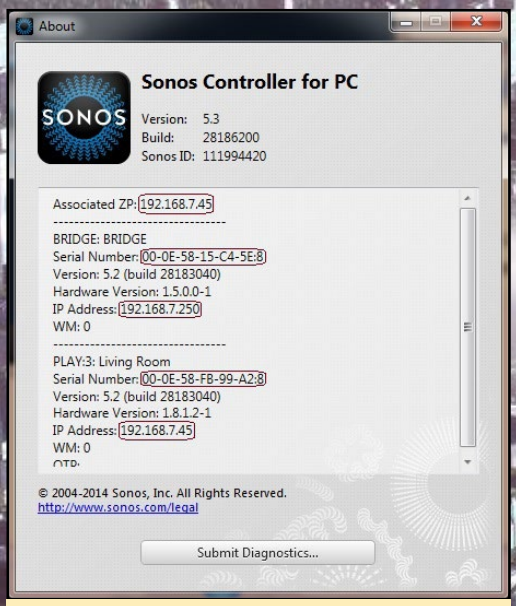
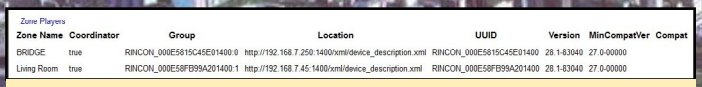


Figura 2: Dirección IP de Sonos



Zone Players

Zone Name	Coordinator	Group	Location	UUID	Version	MinCompatVer	Compat
BRIDGE	true	RINCON_000E5815C45E01400:0	http://192.168.7.250:1400/xml/device_description.xml	RINCON_000E5815C45E01400	28.1-83040	27.0-00000	
Living Room	true	RINCON_000E58FB99A201400:1	http://192.168.7.45:1400/xml/device_description.xml	RINCON_000E58FB99A201400	28.1-83040	27.0-00000	

Figura 3: Localizaciones zonas Sonos zone locations

Actualizar la configuration

Con el experimento en mente, ahora podemos modificar los archivos de configuración. En primer lugar, vamos a actualizar el archivo de configuración de openHAB:

```
$ cd ~/openhab/configurations
$ medit openhab.cfg
```

Actualiza la sección de Sonos Binding e incluye la información de Sonos Bridge y Sonos Speaker para que coincida con el siguiente código y guarda los cambios:

```
################################ Sonos Binding ########################################
#
#Add a line for each Sonos device you want to pre-define
#The format is <name>.udn=<RINCON UID>
#
sonos:office.udn=RINCON_000E5815C45E01400
sonos:living.udn=RINCON_000E58FB99A201400

#
#Interval, in milliseconds, to poll the Sonos devices for status variables
sonos:pollingPeriod=1000
```

Despues, actualiza la configuración de la vivienda demo:

```
$ cd cd ~/openhab/configurations/items/demo.items
$ medit demo.items
```

Cambia la entrada de luz del taller, que se comunica con GPIO a través del pin # 21:

```
Switch Light_C_Workshop "Workshop"   (gC, Lights)
{gpio="pin:21"}
```

Luego, cambia la entrada de la fecha/hora para reflejar tu zona horaria, que en este ejemplo corresponde a Los Angeles:

```
/* NTP binding demo item */
DateTime Date "Date [%1$tA, %1$td.%1$tm.%1$tY]" <calendar> { ntp="America/Los_Angeles:de_DE" }
```

Por último, añade las entradas relacionadas con los elementos de Sonos, utilizando una línea por entrada:

```
/* multimedia */
Switch Sonos_GF_Living "Sonos Play/Pause" (GF_Living, Sonos) {sonos="[ON:living:play],[OFF:living:
pause]"}

String Sonos_GF_Living_Track "Current Track [%s]" (GF_Living, Sonos) {sonos="[RINCON_000E58FB99A20
1400:currenttrack]", autoupdate="false"}
```

Para el dispositivo Sonos de la sala de estar simulada, hemos implementado un interruptor de reproducción/pausa, y al tiempo se visualiza la pista que se está reproduciendo.

Mover la carpeta openHAB

Para hacer openHAB accesible a cualquier cuenta de usuario, mueve todo el directorio a su destino final:

```
$ sudo mkdir /opt
$ sudo mv ~/openhab /opt
$ cd /
$ sudo chown -R root:root /opt
```

Ejecutar openHAB

Para iniciar openHAB, escribe los siguientes comandos en una ventana de terminal:

```
$ cd /opt/openhab
$ sudo ./start.sh
```

Estate atento a cualquier error en el resultado, para ver si corresponden a errores de configuración y tratarlos antes de pasar al siguiente paso. La configuración de Sonos puede que necesite ajustarse si el UID no coincide que debería indicarse en las siguientes entradas:

```
2015-03-18 12:16:30.480 [INFO ] [.service.AbstractActiveService] - Sonos Refresh Service has been
started
2015-03-18 12:16:30.889 [INFO ] [.b.sonos.internal.SonosBinding] - Querying the network for any
other Sonos device
2015-03-18 12:16:32.797 [INFO ] [.b.sonos.internal.SonosBinding] - Found a Sonos device (S3) with
UDN uuid:RINCON_000E58FB99A201400
```

Una vez que openHAB se esté ejecutando correctamente, utiliza el navegador del ordenador de pruebas y navega hasta la aplicación demo openHAB de ejemplo del ODROID:

```
http://<direccion-ip-de-c1>:8080/openhab.
app?sitemap=demo
```

El servidor web basado en Jetty utilizado por openHAB mostrará una pantalla de bienvenida (Figura 4). Observa que la página de inicio de la demo muestra los atributos de los nivel (primer planta, planta baja y sótano/taller) tal y como se especifica en la configuración. Asegúrate de que la fecha también se corresponde con la zona horaria configurada previamente.

Para comprobar que el experimento funciona correctamente, haz clic en el elemento "Cellar", deberían aparecer todos los dispositivos relacionados con éste, como la luz del taller, (figura 5). A continuación, coloca un dedo cerca del sensor de luz del kit C-tinkering, y comprueba que se enciende el LED. Esto quedará reflejado en la UI de openHAB como se muestra en las Figuras 5 y 6, simulando la entrdada de un intruso.

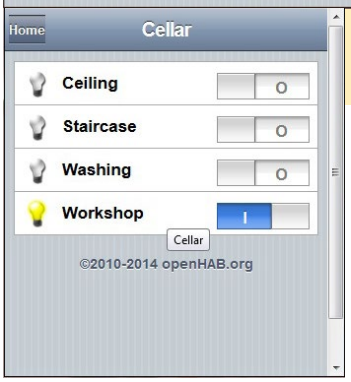
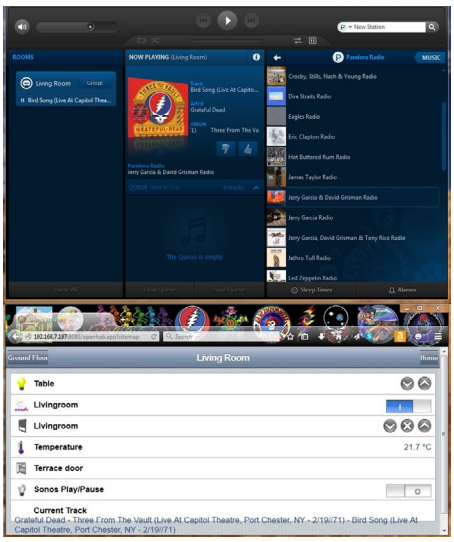


Figura 5: La luz del Taller (alarma) OFF
Figura 6: La luz del Taller (alarma) ON

En una situación real, puedes probar otras respuestas a estos eventos, como enviar un mensaje de texto o correo electrónico. La información importante es transmitida a través del pin # 21 (GPIO # 101), utilizando la infraestructura GPIO, como ya se ha señalado anteriormente.

Figura 4: Pantalla de bienvenida openHAB



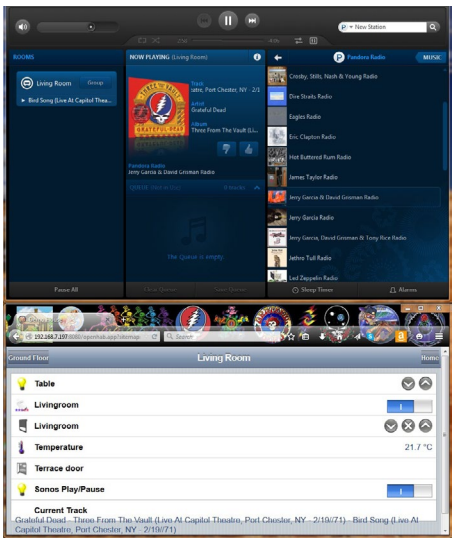


Figura 7: La UI del openHAB & Sonos refleja reproducción
Figura 8: La UI del openHAB & Sonos refleja una pausa

A continuación, inicia la aplicación Sonos de Windows 7, debería detectar el sistema de altavoces y el Sonos Bridge. Para probarlo, inicia el canal Pandora y reproduce algo de música. Luego, en la página de inicio de openHAB, haz clic en el elemento “ground floor”, debería aparecer una pantalla con los botones play/pause y la pista de audio (Figura 7).

Comprueba que la música que se está reproduciendo a través de Pandora (en Sonos) es la misma que la que se muestra en openHAB. Haga clic en el botón de play/pause para detener la música, debería dar lugar a un cambio en el estado (Figura 8)

Experimetnos adicionales

Te animo a descargar componentes adicionales de openHAB, como greent, una interfaz web más elegante y de diseño, para que experimentes con su instalación y uso. Estos archivos estan incluidos en las compilaciones Jenkins como distribution-1.7.0-SNAPSHOT-designer-linux.zip y distribution-1.7.0-SNAPSHOT-greent.zip.

Una vez que te sientas cómodo con la configuración, puede desarrollar tus propios complementos e implementarlos en tu instalación de openHAB. Como puede ver, sólo hemos rozado muy por encima las capacidades de openHAB. Hay gran cantidad de información disponible en la página web de openHAB http: //www.openhab.org, así como en otras web y foros, ¡así que diviértete integrando todo tipo de dispositivos y eventos!

Responsabilidades

Por favor, dedica algo de tiempo a investigar bien el tema de la domótica antes de implementar openHAB en tu propia casa, y toma todas las precauciones que estimes oportuno cuando la automatices. La información que incluye este artículo sólo tiene fines educativos y de diversión, el autor y los editores no se hacen responsables de ningún tipo de daño resultante de cualquier implementación de openHAB. Asegúrate de revisar tus pólizas de seguros en relación a la cobertura de accidentes, ya que en última instancia tu eres responsable de tu propia protección y seguridad.

Recursos adiconales

http://bit.ly/1G0rHOP
http://bit.ly/1BHniKe
http://bit.ly/1N3kIoh
http://bit.ly/1CJR7z2
http://bit.ly/1FQn7lS
http://bit.ly/19JuH6d
http://bit.ly/1IhiaCf
http://bit.ly/19dWx9n
http://bit.ly/1xRitO3

TRIPLE ARRANQUE EN ODROID-CI

PACK LINUX UBUNTU/DEBIAN, ANDROID Y OPENELEC EN UN UNICO VOLUME FISICO

por @loboris

He creado un script que permite iniciar Linux Ubuntu/Debian, Android y OpenELEC desde la misma tarjeta SD. Incluye un menú de arranque para seleccionar el sistema operativo deseado al arrancar ODROID.

Para empezar, descarga los script compilados del triple arranque desde `http://bit.ly/19dGxEv`, que deben ejecutarse en un PC Linux. Para poder trabajar con imágenes de tarjetas SD, debes tener instalado kpartx en el equipo:

```
$ sudo apt-get install kpartx
```

También necesitarás 4 tarjetas SD: una con una copia de Android (creado para ejecutar la imagen autoinstable de Hardkernel), otra con Linux, como una de mis imágenes Linux de `http://bit.ly/1DLugEZ`, otra con OpenELEC, y una tarjeta más en blanco donde se almacenará la imagen final de arranque múltiple y que debe tener un tamaño de al menos 8 GB.

Extraer Android

Extrae Android insertando la tarjeta SD con Android en un lector USB y navega hasta directorio donde fue descargado el script. Edita el script "extract_android " con un editor de texto y actualiza el nombre del dispositivo de la tarjeta SD (/dev/sdX) para que coincida con el nombre de la tarjeta SD con Android. Además, cambiar el directorio de salida deseado y guardar el script. Para ejecutar el script, escribe lo siguiente en una ventana de terminal:

```
$ sudo chmod +x ./extract_android && ./extract_android
```

Esto copiará la partición de arranque de Android, así como las particiones del sistema, userdata, caché y de almacenamiento a tu directorio. Esta será la base para la imagen, y en los siguientes pasos extraeremos las otras particiones (Linux y OpenELEC) a la tarjeta SD.

Extraer Linux

A continuación, inserta la tarjeta SD con Linux en el lector USB. En el directorio del script, edita "extract_linux " y actualiza el nombre del dispositivo de la tarjeta SD (/dev/sdX), así como el directorio de salida al igual que hiciste con Android. Luego, ejecuta el script Linux, que extraerá las particiones de arranque, FAT y Linux (ext4):

```
$ sudo ./extract_linux
```

Extraer OpenELEC

El procedimiento para preparar OpenELEC es idéntico al de Android y Linux. Utiliza script "extract_openelec", luego ejecutalo para extraer las particiones de OpenELEC al directorio de salida:

```
$ sudo ./extract_openelec
```

Preparar la tarjeta SD

Inserta la tarjeta SD 8GB en balnco que usarás para el triple arranque, Edita el script "create_dual_boot_sd", y actualiza el dispositivo para la tarjeta SD (/dev/sdX), los directorios de las fuentes y los tamaño de las particiones deseadas. Ten en cuenta que la partición de Linux siempre se extenderá hasta el final de la tarjeta SD, utilizando el espacio restante tras asignar las particiones Android y OpenELEC. Guarda el script y ejecútalo:

```
$ sudo ./create_dualboot_sd
```

Una vez que el script se haya completado, la distribución de particiones en la tarjeta SD será la siguiente:

Almacenamiento interno para Android (mmcblk0p1)
System (mmcblk0p2)
Userdata (mmcblk0p3)

Patición ampliada
Cache (mmcblk0p5)
OpenElec (mmcblk0p6)
Linux (mmcblk0p7)

Copiar los archivos en la SD

Edita el script "copy_to_sdcard ", actualiza el dispositivo de la tarjeta SD (/dev/sdX) y los directorios de las fuentes, luego ejecuta el script:

```
$ sudo ./copy_to_sdcard
```

Una vez que se complete el script, la tarjeta SD ya estará lista para el triple arranque con Android, Linux y OpenELEC usando un ODROID-C1. Durante el arranque, te aparecerá un menú para seleccionar el sistema operativo deseado.

Notas

La primera partición de la tarjeta SD está formateada en FAT32, permitiendo que pueda ser leída por Windows. Es importante no eliminar ninguno de los archivos del directorio raíz de la primera partición, ya que son necesarios para el triple arranque funcione.

Si su tarjeta SD es lo suficiente grande (16 o 32 GB), los scripts pueden ser fácilmente adaptados para arrancar más de una instalación de Linux. Puedes lograr mayor personalización con dos script adicionales:

"Update_logo", permite actualizar la imagen del logotipo
" update_uboot", facilita la actualización de la partición de arranque de Android sin tener que borrar toda la SD

Imagenes precompiladas

Por comodidad, puede descargar imágenes precompiladas desde `http://bit.ly/1MY2zKv`, listas para ser copiadas en la tarjeta SD. El archivo multiboot.tar.xz contiene todos los scripts y directorios para Android, Ubuntu (minina) y OpenELEC 5.0. En primer lugar, extrae el paquete como root para mantener las propiedades y los permisos:

```
$ sudo tar -xpsf multiboot.tar.xz
```

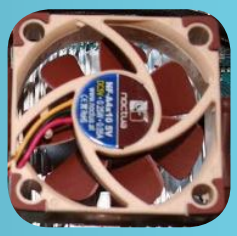
Navega hasta el directorio de arranque múltiple, ajusta tu dispositivo de tarjeta SD y tamaño de la partición editando el script como ya se ha descrito anteriormente, luego ejecútalo:

```
$ sudo ./create_dualboot_sd
$ sudo ./copy_to_sdcard
```

Puedes instalar paquetes de escritorio completos o mínialistas tras el arranque. Para preguntas, comentarios o sugerencias, por favor consulte el post original en `http://bit.ly/1EBmLvt`.

VENTILADOR XU3 MEJORADO

MANTENLO QUIETO



por Tomasz Nazar

El ventilador del ODROID-XU3 se activa cada dos por tres, incluso cuando la CPU está al ralentí, así que este script mejora el funcionamiento del ventilador original. Ha sido probado en Ubuntu 14.04.1 y archlinux 3.10.69-1-ARCH, y es muy probable que funcione en otras distribuciones con pequeños ajustes.

Seguridad

Me he esforzado en ajustar la velocidad del ventilador basándome en la temperatura máxima de los sensores. Cuando el script se cierra, el ventilador vuelve a la configuración original de fábrica. ¡Úsalo bajo tu propio riesgo!

Instación

El script está disponible para su descarga desde mi cuenta de GitHub `http://bit.ly/1EysWA8`:

```
$ cd /usr/bin
$ sudo apt-get install git
$ git clone https://github.com/nthx/odroid-xu3-
fan-control.git
$ cd odroid-xu3-fan-control
$ sudo chmod +x *
$ sudo ./odroid-xu3-fan-control.sh
```

Ejecutar como un servicio

Para que el script mejorado del ventilador se inicie automáticamente cuando arranque el sistema, edita el archivo odroid-fan-controller y añade la ruta completa del script odroid-xu3-fan-control.sh, luego, escribe los siguientes comandos en una ventana de terminal con el fin de añadirlo a la lista de servicios:

```
$ cd /etc/init.d/
$ sudo ln -s /usr/bin/odroid-xu3-fan-control/
odroid-fan-controller
$ sudo update-rc.d odroid-fan-controller defaults
```

También puedes utilizar el siguiente comando para iniciar y detener manualmente el controlador:

```
$ sudo /etc/init.d/odroid-fan-controller start
$ sudo /etc/init.d/odroid-fan-controller stop
```

Para mejorar aún más el sistema del ventilador, se pue-de hacer una simple modificación en el hardware que consiste en instalar un ventilador de sustitución.

JUEGOS LINUX

LA SERIE WARCRAFT ORIGINAL

por Tobias Schaaf

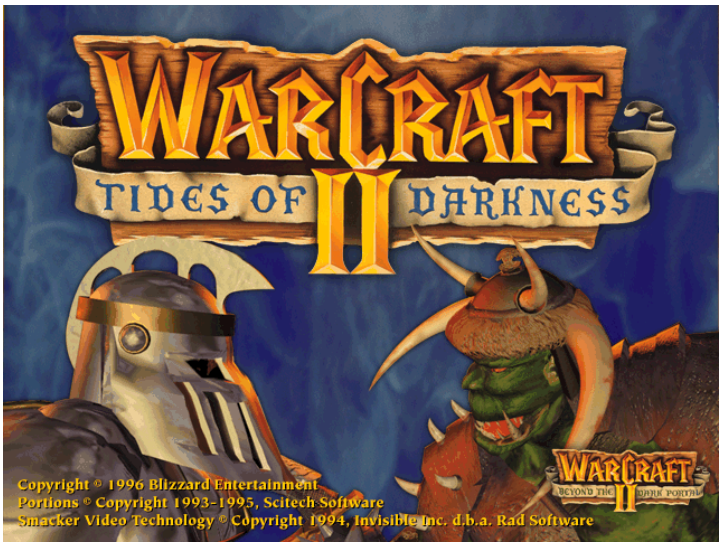
Hace poco me preguntaron si podría escribir un artículo sobre los juegos WarCraft sobre ODROID. Puesto que se trata una serie de juegos de estrategia en tiempo real (RTS) muy buena, quiero compartir mis resultados y experiencia, y lo que puedes esperar si lo ejecutas en ODROID.

Emulación DOS

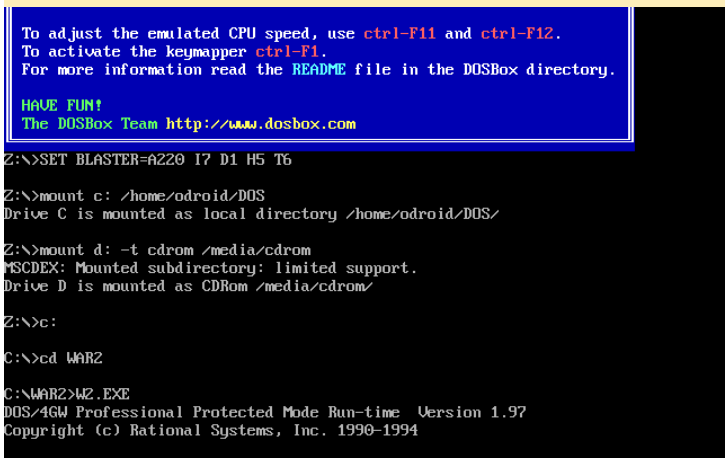
Warcraft - Orcs & Humans (1994), WarCraft II - Tides of Darkness (1995) y WarCraft II - Beyond the Dark Portal (1996) son juegos similares al Dune 2 y al Command and Conquer. Lideras un ejército con el que puedes construir diferentes tipos de edificios, modificarlos con diferentes objetos, y enviar tus soldados a luchar contra un enemigo.

En WarCraft, todo esto se desarrolla en un escenario de fantasía con Elfos, Orcos y Humanos, que combaten en batallas épicas y usan recursos como el oro, la madera y el petróleo para construir sus ejércitos. Es una serie muy buena que te puede proporcionar horas y horas de diversión, más aún si juegas partidas contra otros jugadores.

Dado que los juegos sólo están disponibles para DOS, opino que la forma más sencilla de ejecutarlos es a través de DOSBox, que proporciona emulación DOS y permite jugar a los juegos del modo en el que fueron originalmente pensados, de vuelta a "los viejos tiempos".



Figuras 1, 2, 3 - WarCraft 2 sobre ODROID U3 con DOSBox



Inicie DOSBox, cargué el juego y funcionó al instante, sin complicaciones. Simplemente monte el CD y empecé a jugar, sin tener que realizar ningún tipo de programación en absoluto. Así que parece que he terminado con el artículo, ¿no? No hay nada que hacer, sólo iniciarlo y listo.

Bueno, si hubiese sido así, probablemente ni siquiera hubiera escrito el artículo. Así que, ¿qué podemos hacer si todo funciona correctamente? Bueno, el juego es bastante antiguo

y por lo tanto está bastante limitado. Se ejecuta con una resolución de 640x480, por lo que no presenta un aspecto muy moderno. Si lo ejecutas en modo de pantalla completa aparecen problemas, ya que a partir de 640x480 no se visualiza correctamente. Así que pensé que al menos debería comprobar si podía mejorar algo la experiencia de juego.

Para ello, voy a hacer referencia a mi anterior artículo sobre DOSBox de la edición de noviembre 2014 de ODROID Magazine, donde expliqué que glshim se puede utilizar para mejorar la experiencia de DOSBox usando la interfaz de OpenGL. Para esto, primero cambiaremos las siguientes opciones en el archivo de configuración de DOSBox:

```
fullscreen=true
fullresolution=1920x1080
output=opengl
```

Después, me aseguré de instalar el paquete libglodroid de mi repositorio, e inicie DOSBox con el siguiente comando, que activa la compatibilidad con OpenGL:

```
LD_LIBRARY_PATH=/usr/local/lib/
dosbox
```

Con esto, podía ejecutar el juego a una resolución de 1920x1080 y fue capaz de escalarlo a pantalla completa, viéndose bastante bien. El juego seguía ejecutándose perfectamente y la imagen se ajustaba bien a un monitor HD.

Aparecieron algunos problemas con los efectos del menú principal, que lo hacía muy lento pero que no afectaban al juego en sí, por lo que no eran importantes. Si estás buscando ejecutar el juego tal y como era, esto es lo más cerca que puedes llegar y realmente su rendimiento es muy bueno sobre ODROID.

Aún no satisfeho

Aunque todo está resuelto y el juego funcionaba muy bien, no estaba satisfecho. Ahora que estamos en el 2015, el juego realmente se ha vuelto muy antiguo con el tiempo. Sólo se pueden seleccionar nueve unidades al mismo tiempo y no hay cola de construcción. Limitaciones como éstas hacen que el juego este muy desfasado, los juegos RTS "modernos" no tienen estas limitaciones.

Además, la resolución está simulada con glshim. De hecho, sería muy bueno ejecutar el juego en HD sin tener que escalar 640x480 a 1080p. De modo que cogí otro proyecto que sabía que se ejecutaría en ODROID denominado Stratagus. Se trata de un motor RTS que te permite ejecutar juegos de estrategia en casi cualquier tipo de hardware. El motor es muy capaz y muchos proyectos lo utilizan, por lo que en realidad es un mod para ejecutar WarCraft, WarCraft II e incluso StarCraft. Además, hay bastantes juegos que se desarrollaron directamente para usarse con el motor Stratagus. Entra en http://stratagus.com para conocer los diferentes juegos que utilizan Stratagus.

Sin ser muy técnico, Strategus funciona con ODROID, aunque no fui capaz de activar OpenGLES, ya que el juego requiere SDL 1.2 que no es compatible con OpenGLES. En su lugar, requieren versiones especiales de SDL 1.2 para utilizar OpenGLES. Estas versiones son muy difíciles de encontrar, y aunque encontré una versión de SDL_GLES simplemente se trataba de una extensión que no llego a funcionar.

Se rumorea que hay una versión SDL 1.2 con OpenGLES integrado, pero no he sido capaz de encontrarla (todavía).

Sin embargo, Los ODROIDs generalmente son lo bastante rápidos como para ejecutar el juego simplemente con SDL sin aceleración por hardware, por lo que podía iniciar Strategus usando sólo SDL. Luego creé un par de paquetes para facilitar la instalación de Stratagus. Los paquetes para WarCraft I y II se llaman war1gus (Warcraft I) y Wargus (Warcraft II), los pueden encontrar en mi repositorio en http://bit.ly/1MKIQvQ. Puedes instalar los juegos con *apt-get install* añadiendo la lista de paquetes a tu distribución de Linux.

Instalación

Instalar Stratagus es muy fácil: simplemente tienes que escribir "apt-get install stratagus" y ya está. La instalación de Wargus (WarCraft 2) es más complicada. Primero, asegúrate de tener bien una unidad de CD/DVD externa conectada con el CD insertado (o una ISO con el juego montado), o bien simplemente copia la carpeta "data" que contiene los archivos .war a tu ODROID

A continuación, instala Wargus con el comando "apt-get install Wargus". Tras completarse la instalación, se abrirá una ventana de diálogo pidiéndote que confirmes las opciones de instalación. Debes responder a las preguntas del siguiente modo:

```
Extract Warcraft II data files
now? - Yes
```

Después, tienes que introducir la ruta de donde está montado el CD o de donde se encuentra la carpeta de datos.

```
Extract and convert MIDI sound
files? - No
```

Esto es muy importante porque la opción MIDI fue eliminada de la herramienta que se utiliza para extraer los datos, pero olvidaron retirar la pregunta del menú. Si seleccionas Yes, la instalación fallará.

```
Extract and convert videos?
```

Esta opción depende de ti. Los videos no están en muy buena resolución y ocuparán alrededor de 5,7 MB de espacio

```
Rip sound tracks from CD-ROM?
```

Esta opción sólo funciona si tienes el CD físico en la unidad de CD/DVD. Decirle No es probablemente lo correcto para la mayoría de la gente. Wargus también necesita las fuentes de sonido MIDI TimGM6mb.sf2, que fueron descargadas desde http://bit.ly/1MHNqga.

Figura 4 - WarCraft II a 1080p en el ODROID U3

Puesto que son de uso gratuito, las agregué al paquete para que se instalarán automáticamente. Tras extraer los datos del juego (y si resultan seleccionadas, las películas son convertidas) ya está listo para jugar. Puesto que son de uso gratuito, las agre-gué al paquete para que se instalarán automáticamente. Tras extraer los datos del juego (y si son seleccionadas, las películas serán convertidas) ya está listo para jugar.

Ventajas de Stratagus

Tal vez te estés preguntando qué es lo que se consigue usando Stratagus en lugar de la versión DOS. La figura 4 muestra como se ve Warcraft II con Stratagus, puedes apreciar algunos de los cambios como el soporte para 1080p en lugar de la resolución escalada de 640x480. Una pantalla más amplia muestra asimismo partes más grandes del mapa.

Las barras de progreso azules de los obreros y de las construcciones indican el progreso de la tarea actual o el orden de construcción.

Barra de Salud de las unidades y rango del edificio de verde sobre amarillo y naranja con rojo, para que pueda ver el estado de tus unidades y edificios sin tener que hacer clic en cada uno de ellos. También hay unas cuantas cosas que no puedes ver en la imagen, como que el número de unidades que se pueden seleccionar y mover ya no se limita a nueve. Si tienes un ejército de veinte unidades, puede moverlas todos al mismo tiempo. También puede crear grupos de unidades de seis o más, aunque sólo se mostrarán seis. Si haces clic diez veces para construir una unidad, se crearán diez unidades.

Desventajas de Stratagus

Aunque la mayoría de las cosas en el juego suponen una mejora, existen algunos puntos débiles que quiero mencionar. Puesto que el programa se ejecuta en SDL, trabaja en su mayoría sobre la CPU, por lo que puede ser muy agotador para el procesador y puede provocar ciertos retrasos con grandes mapas con muchas unidades y edificios. Descubrí que desactivando la opción "Fog of War" el rendimiento aumenta bastante, pero también hace que cambie la experiencia del juego, de modo que depende de ti si quieres o no desactivarlo.

Mientras que con dispositivos ODROID de las series Exynos 4 y 5 no deberías tener ningún problema con el juego, Strategus en ODROID-C1 puede ir un poco lento debido a su CPU, en comparación con los otros modelos ODROID. No obstante, debería funcionar a una velocidad aceptable.

El desarrollo sobre stratagus y Wargus no está claro, por ahora. Stratagus ha permanecido en la versión 2.7.2 durante un par de años y su desarrollo parece haber sido asumido por algunos desarrolladores de Ubuntu en Launchpad. Descubrí que los fallos de las versiones aparecen en el nivel 5 del juego Alliance, pero Orcs parece no tener el mismo problema. Probablemente se pueda solucionar, pero aun no lo he mirado.

Otro indicio de que Stratagus parece ser un proyecto no terminado es la existencia del proyecto "Stargus", StarCraft ejecutándose sobre stratagus. StarCraft es uno de mis favoritos al que me encanta jugar, pero no está terminado. Los humanos y los Zergs parecen estar creados en su mayoría, pero faltan los Protoss. Ni siquiera se puede jugar a la campaña, lo cual supone un inconveniente, ya que StarCraft tiene una historia muy buena. El motor stratagus y los juegos desarrollados para éste parecen muy prometedores, pero todavía no estan terminados, y parece que el desarrollo se ha detenido completamente en algunos aspectos del proyecto.

WarCraft I – WarIgus

Aunque existe una versión de WarCraft conocida como war1gus para Stratagus, que funciona de forma similar a Wargus, parece ejecutarse de manera extraña. Los edificios se presentan a una velocidad normal, pero las unidades caminan a cámara lenta. Creo que se puede logar una mejor experiencia con la variante DOSBox. También colgaré un paquete de war1gus en mi repositorio para aquel que quiera probarlo. Ten en cuenta que esta versión realmente puede utilizar las opciones de conversión MIDI y que puede llevar bastante tiempo en ejecutarse: unos 30 minutos en un U3 y probablemente más tiempo en un C1.

StarCraft – Stargus

StarCraft 1 es realmente bueno y también funciona con Stratagus. Requiere de la versión antigua de CD en lugar de la versión descargable desde Internet.

Se extraerán todos los archivos necesarios en pocos minutos. Stargus no ofrece campañas, aunque las misiones de la campaña son extraídas de todas formas. Los gráficos están plagados de errores: los Protoss usan gráficos de Zerg y de los humanos cuando construyen algo, además el juego se bloquea con bastante frecuencia. Sólo los humanos funcoina lo bastante bien como para poder jugar.

Conclusión

Stratagus es un buen motor que parece estar hecho para WarCraft II, puesto que se ejecuta muy bien. Tambien hay otros juegos basados en este motor que parecen funcionar muy bien. Si intentas ejecutar juegos que no estan hechos para trabajar con este motor, como WarCraft I o StarCraft, te encontraras con algunos problemas.

Sin embargo, jugar WarCraft II con Stratagus te reportará una gran experiencia y WarCraft I también se ejecuta lo suficientemente bien como para ser jugable. Si quieres jugar al juego original, debes utilizar la versión de DOS que se ejecuta sobre DOSBox con glshim. Si quieres una experiencia más moderna y tienen la posibilidad de jugar en modo multijugador, utiliza mejor Stratagus.

¡Warcraft es un juego muy adictivo, así que asegúrate de jugar con moderación y salir a la calle de vez en cuando!

CONCURSO DE VIDEO TUTORIALES CON CI

MUESTRA AL RESTO LO QUE SABES HACER

por Bo Lechnowsky

Te gustaría ganar de 25 a 100$ en crédito para la tienda ameriDroid. Todo lo que tienes que hacer es:

1. Piensa en una idea para un video tutorial que aún no se haya hecho, o en uno que podrías mejorar.
2. Haz un video que muestre el tutorial
3. Menciona que los productos del tutorial se pueden comprar en ameriDroid.com o en otros distribuidores de Hardkernel de todo el mundo.
4. Sube tu video a un sitio donde puedas compartirlo: YouTube o Vimeo
5. Enviar el enlace a tu vídeo usando nuestra página de contacto en http://ameridroid.com/contact

Veremos el vídeo y juzgamos basandonos en:

- La calidad del contenido
- La calidad del vídeo
- La calidad del audio

Asegúrate de hablar con claridad y que la música de fondo este muy por debajo del volumen de la voz. Ve al grano, hazlo lo más corto posible pero sin prisas. Mantén una actitud positiva y amigable. Si publicamos el vídeo (o un enlace a tu video), te daremos 25$ en crédito para la tienda. Al enviarnos tu video (o el enlace), autorizas a ameriDroid a que pueda editar y/o volver a publicar el video. Además, ¡El mejor video de la semana ganará 75$ adicionales en crédito!

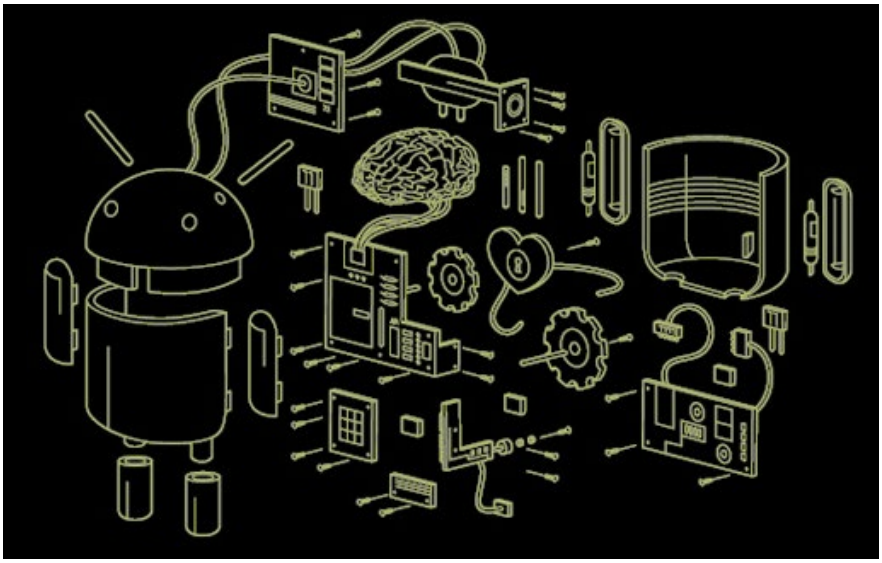
Ten en cuenta que AmeriDroid se reserva el derecho de retener los 75$ de bonificación adicionales si ninguno de los videos presentados durante la semana cumple con ciertos niveles de calidad. Una nueva semana comienza cada domingo a las 12:01 am hora del Pacífico y se prolongará hasta la media noche del sábado siguiente. AmeriDroid también se reserva el derecho de cambiar las reglas del concurso o suspender el concurso en cualquier momento. ¡Tenemos muchas ganas de ver cómo la comunidad crea excelentes videos tutoriales para ayudar a unos y a otros!

Para obtener más información y hacer preguntas, por favor visite el sitio web Ameridroid en http://bit.ly/1NKcIed.

DESARROLLO ANDROID

PRESENTANDO EL NUEVO LOW MEMORY KILLER DAEMON

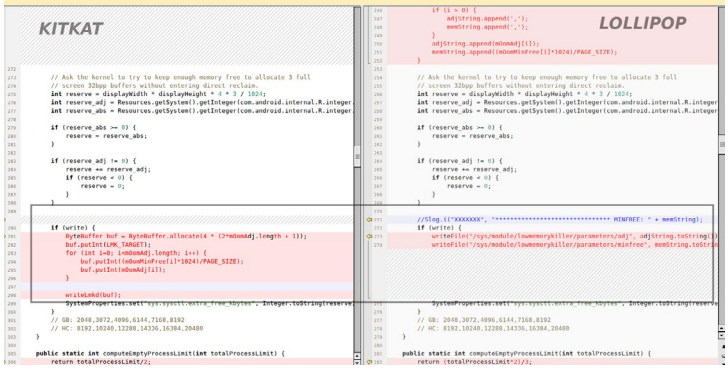
por Nanik Tolaram



Android ha madurado bastante desde su primera versión, y con las versiones que han ido apareciendo desde Ice Cream Sandwich, se han añadido un montón de nuevas características. En cada versión actualizada, las nuevas funcionalidades dependían de donde se centrase Google. En el caso de la última versión de Android (Lollipop), se han centrado en el diseño y rendimiento. Una de las nuevas herramientas que se ha introducido internamente en Lollipop ha sido LMKD, que significa Low Memory Killer Daemon. En versiones anteriores de Android, esta herramienta formaba parte del entorno de trabajo que se comunicaba con la capa del controlador del núcleo, pero ahora se separa. Vamos a echar un vistazo a lo que es LMKD y cómo se funciona internamente.

En Android, incluso cuando el usuario sale de la aplicación, el proceso de la misma aún permanece en el sistema para facilitar su reinicio. Sin embargo, el aumento del número de aplicaciones abiertas puede hacer que la memoria del sistema sea insuficiente, lo que hace necesario eliminar parte de esos procesos a fin de liberar espacio en memoria. Corresponde al mecanismo Low Memory Killer tomar estas decisiones. LMKD recibe notificaciones de demanda de memoria, y luego elimina las tareas apropiadas cuando los recursos de memoria bajan.

Figura I: Diferencias de código del KitKat frente Lollipop con lmk



LMKD

El corazón del Low Memory Killer reside en el archivo drivers/staging/android/low memory killer.c, que se carga como parte del kernel. Puedes consultar el artículo de Desarrollo de Android de la edición de agosto 2014 de ODROID Magazine para obtener más información sobre el funcionamiento interno de este controlador. El Low Memory Killer tiene por 2 partes: el demonio con el cual el entorno de trabajo se comunica con el driver mediante una comunicación empaquetada, y el driver principal del kernel que se encarga de reducir la memoria.

Como se muestra en la Figura 2, el entorno de trabajo se comunica con el low memory killer daemon que, a su vez, configura los parámetros correctos a través de los filesys que han sido expuestos por el driver del kernel.

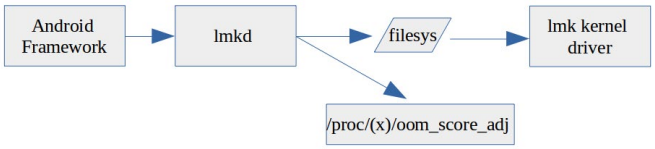


Figura 2: La comunicación entre las diferentes partes

Filesys

Los filesys que LMKD usa para comunicarse con el módulo del kernel low memory killer se muestra en la Figura 3..

```
#define INKERNEL_MINFREE_PATH "/sys/module/lowmemorykiller/parameters/minfree"
#define INKERNEL_ADJ_PATH "/sys/module/lowmemorykiller/parameters/adj"
```

Figura 3: Sistema de archivos para comunicarse con el kernel

Init.rc

Como todo en Android, la aplicación LMKD se inicia durante el proceso de inicialización, lo que significa que debe ser

declarado en el init.rc, o en uno de los archivos .rc disponibles. Puesto que esto constituye la parte principal de la inicialización del entorno de trabajo, en realidad se declara dentro del init.rc ubicado en system/core/rootdir, que se muestra en la Figura 4.

```
service lmkd /system/bin/lmkd
    class core
    critical
    socket lmkd seqpacket 0660 system system
```

Figura 4: Declaración LMKD en init.rc

Entorno de trabajo

El archivo principal que comunica la aplicación de Android y la app LMKD se encuentra dentro del entorno de trabajo. Sólo hay 1 archivo que se encarga de esta comunicación y reside en el interior del archivo ProcessList.java. Internamente, el entorno de trabajo determina lo que se debe comunicar a LMKD mediante códigos de comando, definidos como muestra la Figura 5.

```
// Low Memory Killer Daemon command codes.
// These must be kept in sync with the definitions in lmkd.c
//
// LMK_TARGET <minfree> <minkillprio> ... (up to 6 pairs)
// LMK_PROCPRIO <pid> <prio>
// LMK_PROCREMOVE <pid>
static final byte LMK_TARGET = 0;
static final byte LMK_PROCPRIO = 1;
static final byte LMK_PROCREMOVE = 2;
```

Figura 5 : Código comandos LMKD

LMK_PROPRIO es el comando que utiliza el entorno de trabajo con mayor frecuencia, que informa a LMK para que modifique y cree nuevas estadísticas fuera de memoria para una aplicación en particular cada vez que se esté procesando una solicitud (iniciar, parar o pausar).

Otro comando interesante es LMK_TARGET que se utiliza específicamente para configurar la información general fuera de memoria para el driver low memory killer, utiliza los filesys que se muestran en la Figura 2. Cuando el kernel empieza a llamar a la función low memory killer, utiliza el parámetro que se ha pasado mediante este comando.

JUEGOS ANDROID

por Jeremy Leesmann

Iron Force

Batallas de tanques muy explosivas

Alguna vez ha querido controlar un tanque en la batalla. Pues ahora puedes con Iron Force. Lucha contra otros en un intenso tiroteo, gana batallas y consigue los tanques más poderosos para convertirse en el Top Commander. ¡Así que avanza y destruye todo lo que veas!

https://play.google.com/store/apps/details?id=com.chillingo.ironforce.android.ajagplay

Beach Buggy Racing

Carreras de acción al estilo dibujos animados en la mejor ubicación de la tierra ... ¡la playa!

Beach Buggy Racing combina los estilos de carreras de Beach Buggy Blast y Mario kart en un divertido jugueteo en la playa. Si te gustan las carreras y las batallas, entonces te encantará este juego. Los controladores y gráficos funcionan muy bien tanto en el ODROID-U3 y C1, y pueden unirse hasta 4 jugadores en un sólo monitor.

https://play.google.com/store/apps/details?id=com.vectorunit.purple.googleplay

CONOCIENDO A UN ODROIDIAN

ANTHONY SALTER (@VIRIDIANGAMES) PADRE DEVOTO Y EXPERTO EN ENTRETENIMIENTO

Anthony nos muestra que no somos conscientes de lo pequeño que es el U3 hasta que no lo colocamos al lado de una tarjeta de crédito

editado por Rob Roy

Por favor, háblanos un poco sobre ti.

Mi nombre es Anthony Salter. Tengo 44 años, y soy desarrollador de juegos profesional en Austin, Texas.

Tu blog viridiangames.com incluye comentarios y experiencias con juegos modernos y retro, y ha estado activo más de 10 años. ¿Cómo fueron tus inicios con los juegos?

Por el año 1978, mi padre llevo a casa un sistema de video juegos Magnavox Odyssey 3000. Se conectaba al televisor, incluía dos joysticks por cable y se podía jugar a ocho versiones diferentes de Pong. Seleccionabas la versión que querías deslizando el interruptor de la parte frontal de la consola. Mi hermana y yo jugamos cada vez que teníamos oportunidad. Más tarde, cuando tenía 9 o 10 años, mi clase recibió un ordenador TRS-80 Model III. Una vez que fui consciente de que podía escribir programas para que los ordenadores pudiesen ejecutar juegos, ¡Simplemente fue dedicarme a ello!

¿Cuáles son algunos de tus juegos favoritos?

Woo, tengo una extensa lista. Mis juegos favoritos indudablemente son los que permiten flexibilidad y que muestren algo nuevo, sin importar cuántas veces juegues. Juegos como la serie Elder Scrolls, la serie Grand Theft Auto y Deus Ex. Aunque ahora mismo también estoy jugando al Battlefield 4, World of Warcraft y Dota 2. Básicamente, los únicos a lo que no suelo jugar son los de deportes, a excepción del NBA Jam.

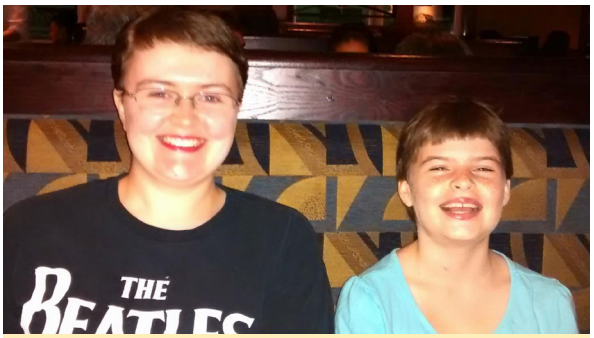
Mencionas en Reddit que ha estado configurando un ODROID para tu hijo para que se involucre más con los ordenadores. ¿Puede describir tu proyecto y sus objetivos?

Tengo una estupenda esposa y tres hijos, uno de ellos (mi hijo) es autista. Pensando en mi propio comportamiento, es muy posible que yo estuviese a ese mimo nivel cuando era más joven pero, por supuesto por aquel entonces, nadie sabía lo que suponía ese nivel. Al igual que yo, David se hico con los ordenadores muy rápido. Sin embargo, como es común en los niños autistas, se suele frustrar fácilmente y de muchas formas, la tecnología moderna está diseñada para frustrar.

Por ejemplo, nuestra PS3 ha dejado de funcionar recientemente. Así que cuando él quiso insertar nuestro Blu-ray The Lego Movie, tuve que decirle que no podía verlo. Cuando se vio frustrado me preguntó el por qué, sinceramente, no podía pensar en una buena razón. Claro que podría haberle dicho: "Porque no tenemos ningún otro dispositivo en casa que pueda leer el formato de datos propietario de la película que está codificada en", pero él no lo habría entendido y esto realmente no es un buena razón.

El objetivo de mi proyecto ODROID es eliminar, o al menos minimizar las frustraciones que se crean con las fisuras legales y las plataformas cuando estás intentando ver, escuchar o ejecutar algo. Y aunque David fue el impulsor del proyecto, no tengo ninguna duda que el resto de mi familia también apreciará el hecho de tener un acceso más sencillo a los archivos y juegos. En concreto, el dispositivo tiene que ser capaz de acceder a los archivos de vídeo y audio a través de un recurso compartido en red y reproducir cualquier formato. También necesita acceder a Netflix y YouTube, y poder ejecutar juegos nativos y emulados. Nada de esto es difícil, ya que los reproductores son muy comunes en los ordenadores de placa única. Pero, yo quería ir un paso más allá y asegurarme de que David podía hacerlo todo usando las herramientas con las que ya estaba familiarizado, como es el gamepad. Yo sabía que esto iba a ser un gran desafío y que requeriría una gran cantidad de programación, diversos script y posiblemente algo de trabajo de diseño del GUI.

Cuando empecé el proyecto, estaba seguro de que iba a usar una Raspberry Pi. El Pi 2 acababa de ser anunciada y parecía ser la plataforma perfecta, pero algunas personas comentaban que aunque la Pi 2 ha supuesto un gran avance, "el ODROID era aún mejor". Así que busqué a este misterioso ODROID, ¡Estoy muy contento de haberlo hecho! Cuando más investigaba sobre el proyecto, más me daba cuenta de las limitaciones



Las hermosas hijas de Anthony, Megan y Jewel

de la Pi 2 y la Pi. Llegue a pensar que el soporte nativo de Android me proporcionaría todas las herramientas de software que necesitaba para conseguir que el dispositivo funcionase, eliminándome la integración GUI y el controlador, que es exactamente lo que yo quería. En un principio encargue un C1, pero debido a un error en la dirección de envío tuve que cancelar el pedido. Después compre un U3, puesto que su CPU es más rápida

Mi única preocupación sobre si utilizar o no ODROID era el hecho de que la comunidad podría no ser tan robusta, y que no encontrase a nadie que me pudiera ayudar. No debería haberme preocupado. Inicialmente tuve dificultades para conseguir que la imagen "Pocket Rocket" de Rob Roy funcionase, ¡y fuel el propio Rob el que me ayudo! El hecho de que todas las distribuciones Android de Hardkernel tengan soporte nativo para los mandos Xbox 360 significa que hay un montón de cosas que ya funcionan "de serie". En la edición del mes pasado se publicó un artículo sobre cómo configurar el mando de la Xbox 360 para que funcionase con diversos emuladores. Realmente lo único que queda es la integración, ¡y en este punto las cosas funcionan muy bien aunque tengo ciertas dificultades para conseguir que mis hijos dejen de jugar con el ODROID y me dejar trabajar en el proyecto.

¿Qué mejoras de interactividad o hardware ves en el futuro de los juegos?

Necesitamos un sistema 3D sin gafas. Sistemas VR como el Oculus Rift y Vive son grandes noticias y tienen un buen factor "sorpresa", pero no me imagino a una familia sentada en el sofá con los auriculares en un futuro próximo. Por supuesto, no me pregunten cómo hacerlo - ¡No tengo ni idea!

¿Ha escrito alguno de tus propios programas o juegos?

Sí, de hecho soy un desarrollador de juegos profesional. Trabajo exportando a linux juegos como Borderlands 2 y Civilization: Beyond Earth. También escribo mis propios juegos de forma extraoficial. Puedes ver en lo que he estado trabajando en mi blog, aunque recientemente el trabajo me ha dejado poco tiempo para trabajar en mi propio material.

¿Está involucrado en otros proyectos aparte de los ordenadores?

Tengo algunas aficiones más, como la cocina y los RPG con papel y lápiz, pero para ser honesto, mi familia absorbe la mayor parte de mi tiempo.

¿Qué tipo de mejoras te gustaía ver en futuras placas ODROID?

Mejorar el hardware de gráficos, posiblemente con memoria RAM dedicada, sería fantástico, aunque soy consciente de que resultaría difícil mantener su bajo precio.

El blog de Anthony en http://viridiangames.com incluye artículos sobre programación y juegos desde hace muchos años.

El hijo de Anthony, David, para el cual está destinado su proyecto de centro multimedia ODROID.

ODROID Magazine está ahora en Reddit!

ODROID Talk Subreddit

http://www.reddit.com/r/odroid