

# ODROID

## Magazine

Año Uno  
Num #8  
Ago 2014



# PRESENTANDO EL ODROID-W

EL DIMINUTO Y PORTÁTIL ORDENADOR DE HARDKERNEL  
QUE FUNCIONA CON PILAS Y ES COMPATIBLE CON PI

**SO DESTACADO:**

**POCKET ROCKET Y COUCH POTATO**

**LA MEJOR IMAGEN PARA JUGAR EN ANDROID CON REPRODUCCIÓN 1080P**



**CREAR TU PROPIO  
RELOJ INTELIGENTE  
CON ODROID**

- LA SIGUIENTE GENERACION DE ODROID-U3+
- COMO COMPILAR PASO A PASO EL KERNEL DE LINUX
- INSTALAR UN SERVIDOR WEB CON NGINX Y LIGHTTPD

# 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



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://www.pollin.de/shop/suchergebnis.html?S\\_TEXT=odroid&log=internal](http://www.pollin.de/shop/suchergebnis.html?S_TEXT=odroid&log=internal)





**H**ardkernel ha tenido un mes muy ajetreado, ha lanzado varios productos nuevos que una vez más han demostrado ser líderes innovadores en microinformática ARM. No sólo Justin y su equipo han desarrollado un nuevo modelo para la serie U y XU, sino que también han creado un producto compatible con Raspberry Pi diminuto y portátil que cuenta con varias características adicionales.

¡El ODRROID-VU es la pantalla táctil USB que has estado esperando! Un modelo de 9 pulgadas con una entrada multi-táctil de 10 puntos que estará disponible a mediados de agosto, y que no sólo funciona con la familia de ordenadores ODRROID, sino que también es compatible con cualquier máquina con Windows, Ubuntu y Android que soporte monitores USB.

El ODRROID-W es una versión diminuta de Raspberry Pi que está hecha para aplicaciones que funcionen con baterías, como elementos portátiles y microcontroladores. Ofrece muchas características nuevas no incluidas en el Raspberry Pi original, como los puertos USB desmontables, al mismo tiempo que sigue siendo muy económica.

Y finalmente, el tan deseado ODRROID-XU3 ya está disponible, ¡Es el ODRROID más rápido jamás creado! Incorpora todo lo que ofrecía la primera generación de XU-E con algunas características adicionales:

- Samsung Exynos5422 Cortex™-A15 2.0Ghz quad core y Cortex™-A7 quad core CPUs
- Mali-T628 MP6 GPU con soporte OpenGL ES 3.0/2.0/1.1 y OpenCL 1.1 Full profile
- Sensores de energía integrados como el XU-E

Daremos más información sobre el ODRROID-XU3 en la edición de septiembre, pero ya puedes conseguir tu XU3 en la tienda Hardkernel en <http://bit.ly/lprRcPL>

En la sección de noticias Linux, el usuario @dspd del foro nos ofrece un informe sobre el nuevo kernel 3.17, que ahora parece tener soporte para ODRROID. Sigue el debate en <http://bit.ly/lS50v7z>

Samsung ha estado trabando en la velocidad de carga (de subida) en ODRROID con Linux. Los primeros parches van incluidos en Linux-3.17, aunque se necesitan cargar más parches para hacer que esto pueda usarse como un sistema de escritorio como tal. Se necesitarán algunas versiones más antes de que todos los parches sean pulidos e incluidos en el sistema.

También están trabajando en la velocidad de carga con ODRROID en uboot, y han encontrado una manera de detectar automáticamente X2 frente al U2/U3 (mediante la comprobación del LED en el X2). Así que ahora uboot seleccionará automáticamente el dispositivo correcto al cargar. Eso significa que una única "imagen SO" creada adecuadamente funcionará para X2, U2 y U3.

Este número tiene un montón de información para los entusiastas del hardware, incluyendo un estudio en profundidad de los diferentes tipos de módulos eMMC que hace Hardkernel y un análisis práctico del poderoso ODRROID-SHOW. También contamos con una guía para compilar tu propio kernel directamente desde el repositorio GitHub de Hardkernel, un tutorial para instalar el servidor web ligero Nginx en un ODRROID, Nanik comienza su fascinante serie sobre el desarrollo de aplicaciones Android, y ¡mucho más!

ODRROID Magazine, que se publica mensualmente en <http://magazine.odroid.com/>, es la fuente de todas las cosas ODRROIDianas. • Hard Kernel, Ltd. • 704 Anyang K-Center, Gwanyang, Dongan, Anyang, Gyeonggi, South Korea, 431-815 • fabricantes de la familia ODRROID de placas de desarrollo quad-core y la primera arquitectura ARM "big.LITTLE" del mundo basada en una única placa.

Únete a la comunidad ODRROID con miembros en más de 135 países en <http://forum.odroid.com/> y explora las nuevas tecnologías que te ofrece Hardkernel en <http://www.hardkernel.com/>.



**HARDKERNEL**

# ODROID

Magazine



**Robert Hall,**  
**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 personalizados 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 e imágenes ODROID 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 de y mis cuatro hijos estupendos.



**Manuel Adamuz,**  
**Editor Español**

Tengo 31 años y vivo en Sevilla, España, pero nací en Granada. Recientemente he sido padre, y mi hijo tiene ahora 5 meses. ¡Es una experiencia increíble! Hace algunos años trabajé como técnico informático y programador, pero mi actual trabajo está relacionado con la gestión de calidad y tecnología de la información: ISO 9001, ISO 27001, ISO 20000. Soy un apasionado de la informática, especialmente de los microordenadores como ODROID, Raspberry Pi, etc. Mi otra gran afición es la bicicleta de montaña y de vez en cuando participo en competiciones semiprofesionales.



**Nicole Scott,**  
**Editor Artístico**

Nicole es una experta en Producción Transmedia y Estrategia Digital especializada en la optimización online y estrategias de marketing, administración de medios sociales y coordinación de equipo, así como la producción multimedia impresa, TV, cine y web. Nicole es experta en diseño gráfico y web, gestión de redes sociales y publicidad, edición de vídeo y maquetación DVD. Dispone de un ODROID U3 que usa para aprender Linux. Ella vive en el área de la Bahía de California, y disfruta haciendo senderismo, acampada y tocando música. Visite su web en <http://www.nicolecscott.com>



**Bruno Doiche,**  
**Editor Artístico**

Está disfrutando de su tiempo libre y de sus vacaciones durante el mes de Agosto





**DESARROLLO ANDROID - 6**



**COMO MONTAR TU TARJETA SD INTERNA - 9**



**PRESENTANDO EL ODROID-W - 10**



**BUSCAR CON GOOGLE BBS / SOLUCIONAR EL OVERCAN DE ANDROID - 14**



**TODO SOBRE LOS MODULOS EMMC DE HARDKERNEL - 15**



**COMPILACION DEL KERNEL DE LINUX - 17**



**REPRODUCTOR DE YOUTUBE ALTERNATIVO - 21**



**INTERESANTES COMANDOS LINUX - PARTE 1 - 22**



**ODROID U3 VS ODROID U3+ - 23**



**INSTALAR UN SERVIDOR WEB - 25**



**INTERESANTES COMANDOS LINUX - PARTE 2 - 25**



**ODROID-VU: UNA ECONOMIA PANTALLA TACTIL HDMI USB DE 9" - 30**



**PEPPERFLASH: PLUGIN CHROME PARA LUBUNTU 14.04 - 32**



**JUEGOS ANDROID: MUPEN64PLUS - 33**



**IO SHIELD DESMITIFICADO - 34**



**PROFUNDIZANDO (EN) EL ODROID-SHOW - 36**



**SO DESTACADO: POCKET ROCKET Y COUCH POTATO - 41**



**CONOCIENDO A UN ODROIDIAN: BO LECHNOWSKY - 47**

# DESARROLLO ANDROID: USANDO EL KERNEL LINUX

## UNA GUIA PARA LOS DRIVERS ESPECIFICOS DE ANDROID

por Nanik Tolaram

Linux es el corazón de Android y depende en gran medida de éste para su funcionamiento, sin él Android no podría ejecutarse. Sin embargo, Android no usa un “simple” kernel de Linux, lo que significa que no podemos descargar un kernel de [www.kernel.org](http://www.kernel.org) y utilizarlo directamente para ejecutar Android. Hay una serie de drivers, códigos y opciones de configuración que deben añadirse para que Android funcione correctamente. Este artículo te ayudará con esta configuración específica, además de analizar los drivers que son necesarios para el kernel de Linux del cual depende el entorno de trabajo de Android. La mayor parte del código que hace falta se encuentra en la carpeta `kernel/drivers/staging/android` tal y como muestra la Figura 1.

También hay varios drivers que se encuentran fuera del directorio base `/staging/android` y que ya están en el Kernel de Linux. Este artículo no pretende ser una exhaustiva lista de todos los drivers necesarios, pero es un buen comienzo para entender cuales son los diferentes drivers que precisa Android.

A continuación se presentan las opciones de configuración que deben ser incluidas como parte de la `zImage` del kernel. La siguiente configuración esta tomada del fichero `odroidu_android_defconfig` dentro de la carpeta `kernel/arch/arm/configs`, como se muestra en el siguiente cuadro.



```
CONFIG_ANDROID=y
CONFIG_ANDROID_BINDER_IPC=y
CONFIG_ANDROID_LOGGER=y
CONFIG_ANDROID_RAM_CONSOLE=y
CONFIG_ANDROID_RAM_CONSOLE_ENABLED_VERBOSE=y
CONFIG_ANDROID_RAM_CONSOLE_ERROR_CORRECTION=y
CONFIG_ANDROID_RAM_CONSOLE_ERROR_CORRECTION_DATA_SIZE=128
CONFIG_ANDROID_RAM_CONSOLE_ERROR_CORRECTION_ECC_SIZE=16
CONFIG_ANDROID_RAM_CONSOLE_ERROR_CORRECTION_SYMBOL_SIZE=8
CONFIG_ANDROID_RAM_CONSOLE_ERROR_CORRECTION_POLYNOMIAL=0x11d
CONFIG_ANDROID_TIMED_OUTPUT=y
CONFIG_ANDROID_TIMED_GPIO=y
CONFIG_ANDROID_LOW_MEMORY_KILLER=y
CONFIG_ASHMEM=y
```

Si nos fijamos en el interior del Makefile, que se encuentra en el directorio `drivers/staging/android`, podrás ver los drivers del Kernel que son específicos para Android. A continuación vamos a analizar detenidamente los diferentes drivers de este directorio, empezando con el driver `binder`.

binder.c	101.9 kB	C source code
binder.h	9.1 kB	C header
Kconfig	2.4 kB	plain text document
logger.c	15.1 kB	C source code
logger.h	1.7 kB	C header
lowmemorykiller.c	5.7 kB	C source code
lowmemorykiller.c.1	5.6 kB	C source code
Makefile	301 bytes	Makefile
ram_console.c	12.4 kB	C source code
timed_gpio.c	4.4 kB	C source code
timed_gpio.h	815 bytes	C header
timed_output.c	3.1 kB	C source code
timed_output.h	1.1 kB	C header

Figura 1 – Lista de Drivers en el interior del directorio `/staging/android`

### binder.c

Binder es la columna vertebral de las aplicaciones en Android. Controla la comunicación entre los diferentes procesos que se están ejecutando y está escrito en lenguaje nativo (C / C++) o Java.

Sin este driver, las aplicaciones de Android no funcionan en absoluto, ya que se montan en el sistema de ficheros virtual principal `/dev/binder`. Android utiliza bastante binder y puesto que las aplicaciones utilizan este entorno de trabajo, cualquier aplicación depende indirectamente del driver binder.

```
obj-$(CONFIG_ANDROID_BINDER_IPC) += binder.o
obj-$(CONFIG_ANDROID_LOGGER) += logger.o
obj-$(CONFIG_ANDROID_RAM_CONSOLE) += ram_console.o
obj-$(CONFIG_ANDROID_TIMED_OUTPUT) += timed_output.o
obj-$(CONFIG_ANDROID_TIMED_GPIO) += timed_gpio.o
obj-$(CONFIG_ANDROID_LOW_MEMORY_KILLER) += lowmemorykiller.o
```

#### Ejemplo de configuración del odroidu\_android\_defconfig

## ashmem.c

Ubicación: /kernel/mm/

Ashmem significa Android Shared Memory y como su nombre indica, es un driver que facilita el uso compartido de memoria entre procesos, nd proporciona un mecanismo para que Linux recupere memoria cuando ésta es demandada. Ashmem se encarga de que el directorio /dev/ashmem esté disponible para que las aplicaciones accedan a su función.

El driver funciona utilizando un descriptor de archivo que puede ser compartido por procesos usando el driver binder ya comentado. La característica singular de ashmem es que cuando el proceso muere, la memoria se va con él, de modo que nunca nos encontraremos con memoria huérfana. También es lo bastante listo como para permitir a las aplicaciones determinar qué área de memoria puede ser recuperada o no, lo que se conoce como “desanclar” y “anclar”.

Con el fin de ahorrar espacio en memoria, las páginas “desancladas” se reclaman periódicamente con el algoritmo LRU. Ashmem tiene su propio reductor que se activa si la memoria disponible es baja. Un supuesto del uso normal del ashmem se puede ver en la gráfica de la Figura 3, en la esquina inferior derecha.

## wakelock.c

Ubicación: /kernel/power

Este es el polémico driver que inicio el acalorado debate entre el equipo del kernel de Android y los desarrolladores del kernel de Linux en los días en los que Google acababa de cometer su código para el kernel estándar, y era considerado por los puristas como un “hack”. ofrece a las aplicaciones del espacio de usuario

la posibilidad de solicitar determinados servicios que siempre son usados por el hardware de modo que siguen funcionando cuando el dispositivo entra en modo resoso con el fin de ahorrar energía.

El driver wakelock está basado en el driver de gestión de energía de Linux. Linux proporciona funciones de gestión de energía como suspender y reanudar, permitiendo que el dispositivo sea “suspendido” para que se ejecute en modo de bajo consumo o “reanudado” para que se vuelva a activar. Este driver es un mecanismo de comunicación que permite a las aplicaciones de espacio de usuario y tareas del sistema determinar si el dispositivo puede entrar en modo suspensión. El driver no suspenderá el dispositivo si hay wakelocks activos en el sistema.

Imagina que estás ejecutando un reproductor de música en tu teléfono móvil y el sistema decide que es hora de suspender, tu reproductor de música dejaría de funcionar. O imagina que el sistema está a punto de suspensarse y te entra una llamada, pero no puedes cogerla porque el sistema está en proceso de suspensión. Wakelock logra evitar este tipo de situaciones, lo que te permite usar el dispositivo sin tener que preocuparte cuando entrará en modo suspensión. El driver wakelock introduce un nuevo sistema de ficheros virtual llamado /proc/wakelocks.

Wakelock y los drivers de suspen-

sión trabajan conjuntamente como una unidad para gestionar los recursos de energía en un dispositivo Android. La captura de pantalla de la siguiente pagina muestra el contenido del sistema de archivos virtual de wakelock dentro del Nexus 7

## ram\_console.c

El driver de consola RAM se puede considerar como un driver auxiliar

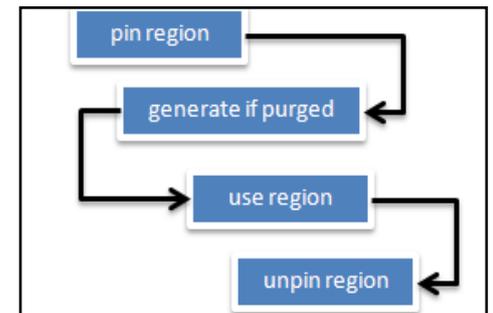
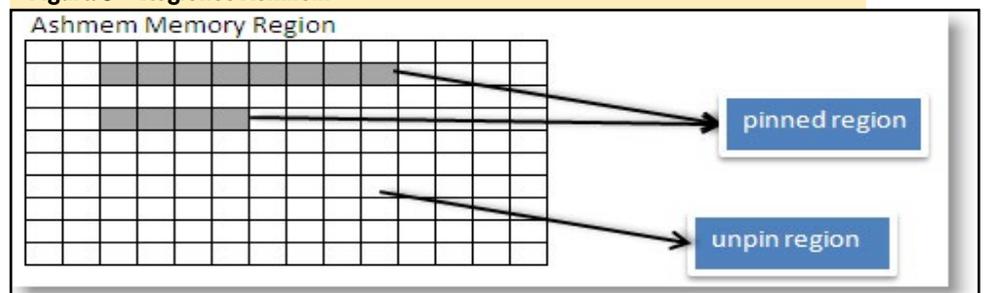


Diagrama de flujo de Ashmem

de registro Linux. Su función es la de guardar el contenido del registro de Linux en una ubicación diferente de la memoria antes de que se reinicie el dispositivo. El contenido del registro de Linux (al que normalmente se accede utilizando dmesg) se puede ver después de reiniciar en el sistema de archivos virtual /proc/last\_kmsg. Si el dispositivo se apaga completamente y no se le suministra energía alguna, el último registro de Linux no estará almacenado la próxima vez que inicies el dispositivo. Esto se debe a que el dispositivo está completamente apagado y no hay energía para conservar el contenido de la memoria. Siempre y cuando se le suministre energía al dispositivo, la próxima vez que lo reinicies podrás extraer el último registro de Linux a

Figura 3 – Regiones Ashmem



```

root@android:/sys/module/wakelock/parameters # cat /proc/wakelocks
name count expire_count wake_count active since total time sleep time max time last_change
power-supply" 35 0 0 0 690183515 665284482 22190375 1989782564706
main" 7 0 0 0 179313595648 0 95068307404 1983210289001
PowerManagerService" 62 0 0 0 45461460309 18975918653 25024409255 1983120599001
wlan_wake" 1115 0 0 0 3374948560 323453592 1088035000 1983040480001
alarm" 39 0 0 0 1158069467 1111682461 44861324 1973444387002
KeyEvents" 124 0 0 0 559400187 98981467 335318000 1973261028002
event2-358" 16 0 0 0 3676100 2735098 785188 1973260746002
event0-358" 35 0 0 0 9258000 0 6291000 1813541323002
alarm_rtc" 1 0 0 0 16929000 0 16929000 21404972002
event1-358" 0 0 0 0 0 0 0 0
bluesleep" 0 0 0 0 0 0 0 0
elan_touch" 0 0 0 0 0 0 0 0
power-supply" 2 0 0 0 83000 0 48000 2491732002
low_battery_detection" 0 0 0 0 0 0 0 0
power-supply" 1 0 0 0 125000 0 125000 1994648003
charger_configuration" 1 0 0 0 27480000 0 27480000 1990818003
wake_lock_dockin" 0 0 0 0 0 0 0 0
suspend_backoff" 0 0 0 0 0 0 0 0
unknown_wakeups" 0 0 0 0 0 0 0 0
deleted_wake_locks" 0 0 0 0 0 0 0 0
usb_config_wake_lock" 1 0 0 0 1987773182880 1987773182880 1815341403234 1987773182880 7215835002
cable_state_changed" 1 1 0 0 4996373567 0 4996373567 2491682002
mmc0_detect" 1 1 0 0 597118566 0 597118566 3430937003
mmc1_detect" 2 1 0 0 903140566 0 559122567 5028933002
wlan_ctrl_wake" 1 1 0 0 3327369567 0 3327369567 15840686002

```

Los drivers wakelocks están considerados como un hack de Google por los puristas

través del sistema de ficheros virtual `last_kmsg`. Este driver es muy útil para solucionar problemas como el kernel panic (error interno del sistema), de memoria o cualquier otro tipo de problemas que causa un kernel panic.

## earlysuspend.c

Ubicación: `/kernel/power`

Puedes ver este driver en acción cuando dejas tu dispositivo inactivo durante unos minutos y la pantalla se apaga por sí sola. Es responsable de asegurar que todos los componentes dentro de tu dispositivo quedan "suspendidos" para ahorrar energía. Este driver y el wakelock van de la mano. Depende del driver de gestión de energía de Linux y puede ser activado enviando información del estado al sistema de archivos `/sys/power/state`.

Si tienes acceso root a tu dispositivo, puede activar el modo suspensión ejecutando el siguiente comando:

```
echo mem > /sys/power/state
```

## logger.c

Este es el driver de registro central usado por el sistema Android, que también está a disposición de las aplicaciones de usuario a través de la API logger integrada. Internamente, el

driver divide el registro en diferentes categorías: main, events, radio y system. El siguiente sistema de archivos virtual esta accesible desde el driver:

```

/dev/log/system
/dev/log/radio
/dev/log/events
/dev/log/main

```

Android ofrece un visor de registro log denominado Logcat que nos permite ver el registro en un formato más legible. La captura de pantalla muestra un ejemplo del contenido de `/dev/log/radio` en su forma original, que no es tan fácil de leer.

## lowmemorykiller.c

Este driver se encarga de eliminar procesos de la memoria cuando la cantidad de memoria libre alcanza un determinado nivel. Cada vez que se inicia una aplicación, se le asigna un valor que indica el tipo de aplicación. Este "marcador" indicará al driver si este proceso puede ser eliminando.

A modo de ejemplo, si miramos la aplicación del teléfono que se utiliza para hacer y recibir llamadas. La aplicación es "marcada" con un valor que indica al driver que solo podrá ser cancelada si se llega al último nivel de memoria, lo que significa que sólo cuando la memoria caiga al nivel más bajo dis-

ponible la aplicación será cancelada. La siguiente tabla muestra los diferentes valores que el sistema utilizará para asignar el proceso lanzado:

```

HIDDEN_APP_MAX_ADJ
HIDDEN_APP_MIN_ADJ
SERVICE_B_ADJ
PREVIOUS_APP_ADJ
HOME_APP_ADJ
SERVICE_ADJ
BACKUP_APP_ADJ
HEAVY_WEIGHT_APP_ADJ
PERCEPTIBLE_APP_ADJ
VISIBLE_APP_ADJ
FOREGROUND_APP_ADJ
PERSISTENT_PROC_ADJ
SYSTEM_ADJ
MIN_HIDDEN_APPS

```

Internamente, el driver sólo tiene capacidad para un máximo de 6 valores distintos, de modo que el sistema hace una conversión interna para ajustarse a los espacios disponibles. Los distintos valores se pueden encontrar en el sistema de archivos virtual `/sys/module/lowmemorykiller/parameters/adj`

Las ranuras de memoria libres disponibles tienen una relación 1:1 con el valor adj y se almacenan en el sistema de archivos virtual `/sys/module/lowmemorykiller/parameters/minfree`

El valor libre mínimo (minfree) esta en páginas de 4k. Por ejemplo, si se dice que la memoria disponible es de 8192, significa  $(8192 * 4k) = 32768$  convertido a 32 KB  $(32768/1024)$ .

La forma en que el nivel minfree se asigna con el valor de adj se muestra en la siguiente tabla:

### Nivel Minfree asignado al adj value

adj	minfree
0	8192
1	10240
2	12288
4	14336
9	16384
15	20480



# PRESENTANDO EL ODROID-W

## UN MINUSCULO Y PORTATIL ORDENADOR COMPATIBLE CON RASPBERRY PI

por Justin Lee

**E**l ODROID-W es un módulo informático muy pequeño que puedes llevar encima y es compatible con todo el software disponible para Raspberry Pi (RPI).

La letra **W** esta relacionada con el:

Desarrollo de dispositivos portátiles

Desarrollo del Internet de las cosas (IoT) de amplia aplicación

Desarrollo de Prototipos electrónicos viables que puedas hacer tú mismo.

Las dimensiones del ODROID-W son muy pequeña 60 x 36 x 7 mm (2.4 x 1.4 x 0.3"). Además incluye una serie de nuevas características y mejoras con respecto al Pi original:

- Un cargador de pilas recargables de Li-polímero y circuito indicador de energía para aplicaciones de robótica portátiles
- Un Reloj de Tiempo Real (RTC) para mantener la hora exacta sin una conexión a Internet gracias a una pila de botón.
- Un ADC de precisión de 12-bit para medir las señales de tensión dinámicas a través de dos entradas de único extremo.
- Sistemas DC/DC que mejoran la eficiencia energética

- Intensificadores DC / DC para los raíles de 5 voltios (USB host y HDMI) desde una batería de Li-Polímero
- El Puerto USB host se puede colocar en la parte superior o inferior según se desee.
- Puertos GPIO 100mil/2.54mm (hasta 32 puertos) para el desarrollo de prototipos

El ODROID-W = RPI + RTC + ADC + UPS + indicador de energía, redujido al máximo.

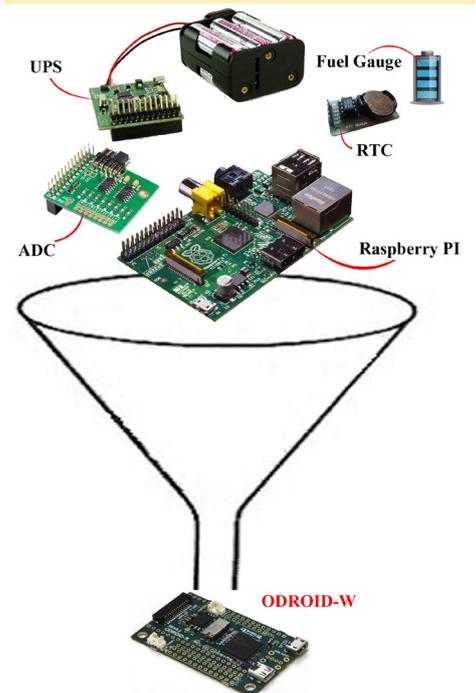
### Hitoria de Desarrollo

A principios de 2014, teníamos un importante proyecto con nuestros socios para ayudarles con el desarrollo de prototipos de algunos dispositivos portátiles y del Internet de las cosas (IoT).

En un principio, contemplamos la posibilidad de usar ODROID-U3 como plataforma base. Aunque ODROID-U3 es entre 8 y 12 veces más rápido que Raspberry Pi, el consumo de energía del U3 no es adecuado para dispositivos portátiles como relojes o collares. Incluso consideramos utilizar el propio Raspberry Pi por su bajos requisitos de energía y buen soporte para BSP Linux, pero la



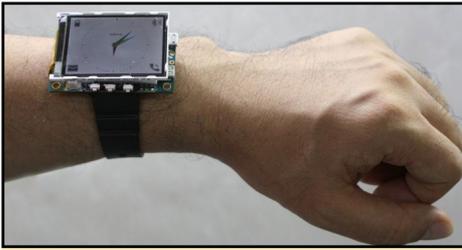
### Una primera edición del ODROID-W apodada ODROID-Pi



### ODROID-W añade características extra a un clon de Raspberry Pi totalmente compatible

PCB del RPi es enorme (más grande aún que la del ODROID-U3)

Con el fin de crear el accesorio portátil más pequeño posible, decidimos crear nuestra propia (diminuta) versión del Raspberry Pi, que permite utilizar muchos de los periféricos disponibles para Pi, como el módulo cámara para Pi y el puerto GPIO de 26-pin. El puerto HDMI y la ranura para SD han sido



**Este prototipo de reloj inteligente U3 requiere demasiada energía.**



**El reloj inteligente Raspberry Pi era demasiado grande para ser viable**

cambiados por micro-conectores, y se ha eliminado el puerto MIPI DSI debido al limitado espacio de la PCB. Además de su tamaño reducido, el ODROID-W mantiene total compatibilidad con todos los periféricos y software existente para Raspberry Pi.

La primera muestra de la PCB ODROID-W fue diseñada el 14 de abril de 2014 y contaba con unos cables puente. La segunda tirada fue diseñada el 19 de mayo de 2014, que corregía algunos de los diseños electrónicos y añadía una ranura para módulos eMMC para su testeo, aún cuando el eMMC no es mucho más rápido que la tarjeta SD debido a la baja velocidad del host eMMC en el SoC. Las pruebas de velocidad del eMMC dieron como resultado una mejoría de tan solo un 10%. La tercera muestra (rev 0.3) fue probada a fondo, y

**No seguimos las instrucciones al pie de la letra y conseguimos esto.**



**Comparando la diferencia de tamaño entre el Rpi y el ODROID-W**

estaba lista para su fabricación en serie.

El precio por unidad para el ODROID-W es de 30\$ y esta disponible en la tienda de Hardkernel <http://www.hardkernel.com>

**Procesador:** Broadcom BCM2835  
ARM11 700Mhz

**Memoria:** Samsung 4Gbit (512MB)  
LPDDR2 SDRAM

**PMIC:** Ricoh RC5T619 incluye  
DCDCs, LDOs, ADCs, RTC, Cargador de pilas y indicador de energía.

**DCDC:** TI TPS61259, intensificador  
DCDC 5V para USB host y HDMI

**Salida de Video:** HDMI Tipo-D  
(Micro-HDMI)

**USB:** USB 2.0 host de alta velocidad

**Conectores GPIO:** Cabezal con pines  
13x2 compatible con Rpi en la parte superior, así como cabezal con pines 20+6 agrupado en 2 vías para conexiones GPIO/RTC/USB adicionales

**GPIOs:** Un total de 32 GPIOs and 2  
ADCs estan disponibles

**Conector Camera:** 15pin MIPI-CS12

**Ranura SD:** Micro-SD (T-Flash)

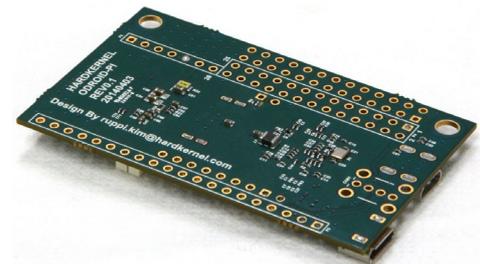
**Energía:** Enchufe Micro-USB para  
la entrada de 5V. Conector de batería  
Li-Polímero (Molex 53398-0271)

**Energía RTC:** Conector de batería de  
respaldo (Molex 53398-0271)

**Dimensiones de PCB en total:**  
60 x 36 mm



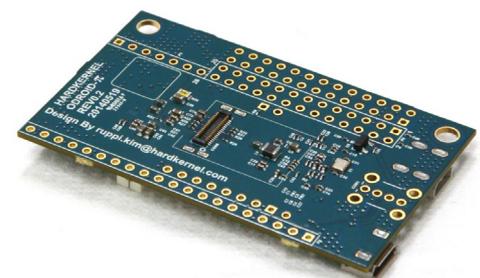
**Vista de la parte superior de la rev. 1 de la placa ODROID-W**



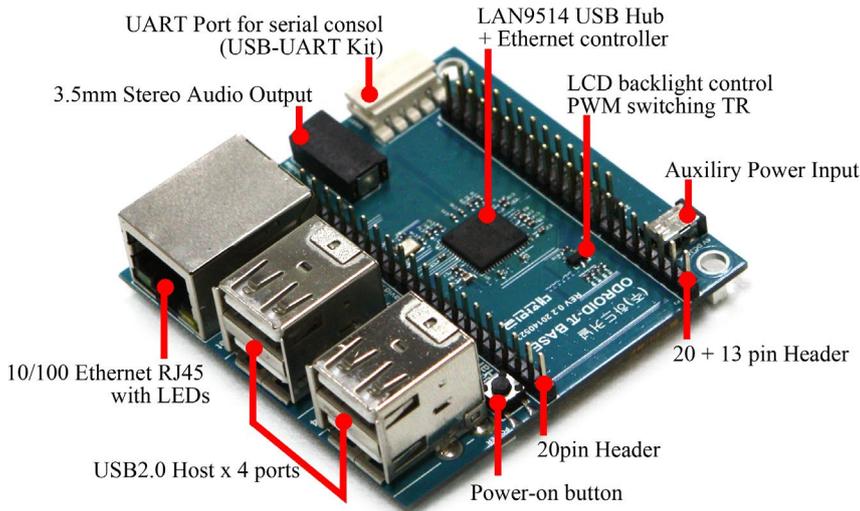
**Vista de la parte trasera de la rev. 1 de la placa ODROID-W**



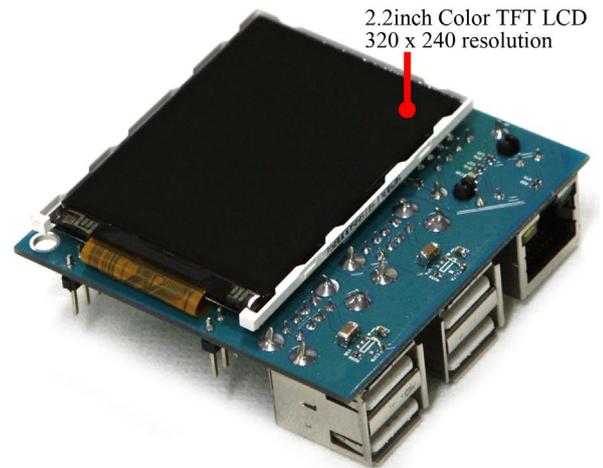
**Vista de la parte superior de la rev. 2 de la placa ODROID-W**



**Vista de la parte trasera de la rev. 2 de la placa ODROID-W**



W Docking Board with LCD  
Size : 60 x 60 x 21mm(Approx)

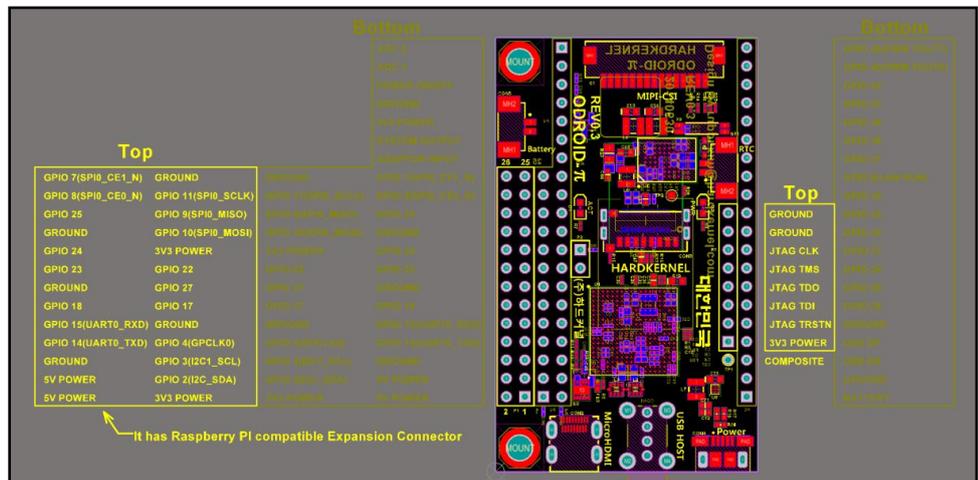


## Primer plano del ODR0ID-W con la placa de expansión y la LCD

La placa de expansión, que se fija al ODR0ID-W en la parte superior o inferior, dispone de 4 puertos USB y un puerto Ethernet. También puedes optar por una versión que incluye una LCD TFT 320x240.

Por favor, vea nuestro video en Youtube de la placa de expansión (<http://bit.ly/1rII5Li>) que incluye una demostración en vivo de las capacidades de la placa. El precio base para la placa es de 20\$ y para la versión con LCD TFT es de 30\$.

El esquema de la PCB del ODR0ID-W lo tienes a la derecha.

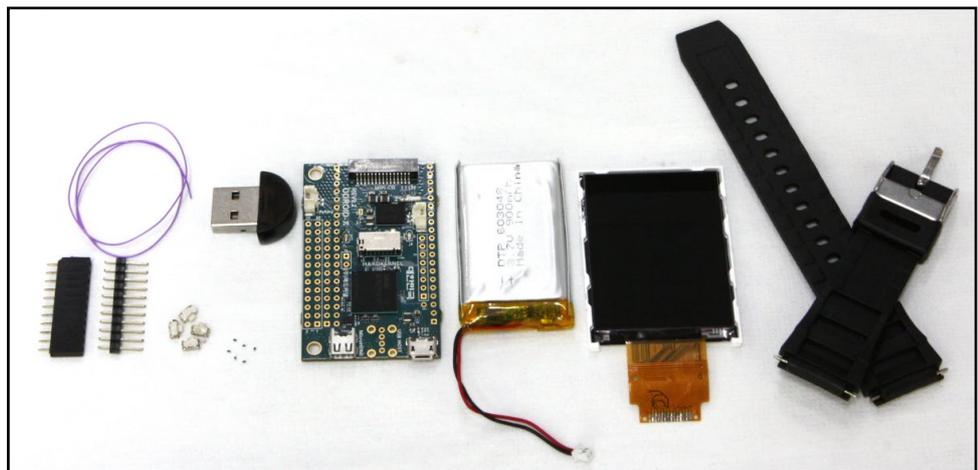


El diagrama esquemático de la PCB de ODR0ID-W

## Ejemplo de reloj inteligente

Este reloj inteligente que puedes hacer por ti mismo, tiene algunas funciones limitadas pero muy curiosas. Está hecho para conectarse a tu teléfono inteligente Android vía Bluetooth.

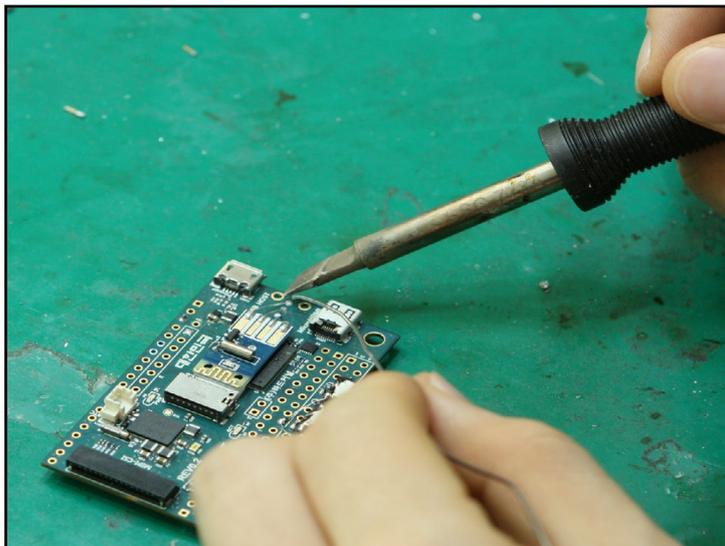
Cada vez que recibas un mensaje en tu teléfono, se envía una notificación al reloj, como se puede ver en este video <http://bit.ly/1sOkTOC>



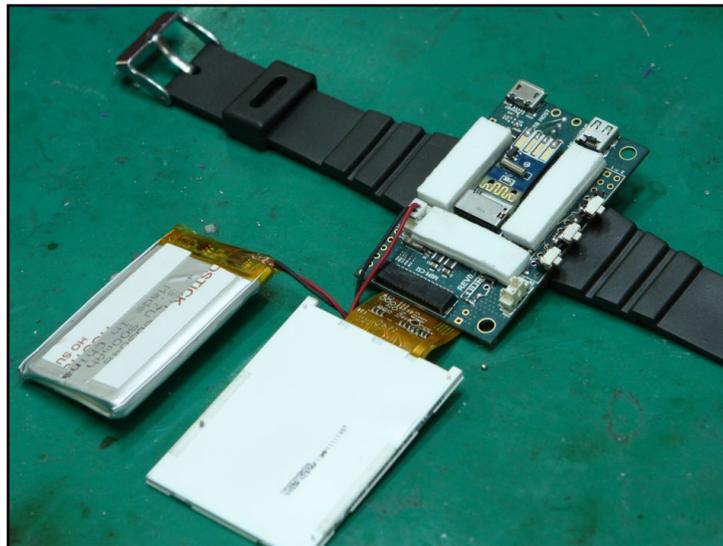
Materiales de hardware para montar el reloj inteligente

¡La siguiente página muestra cómo montar tu propio reloj inteligente usando un ODR0ID-W !

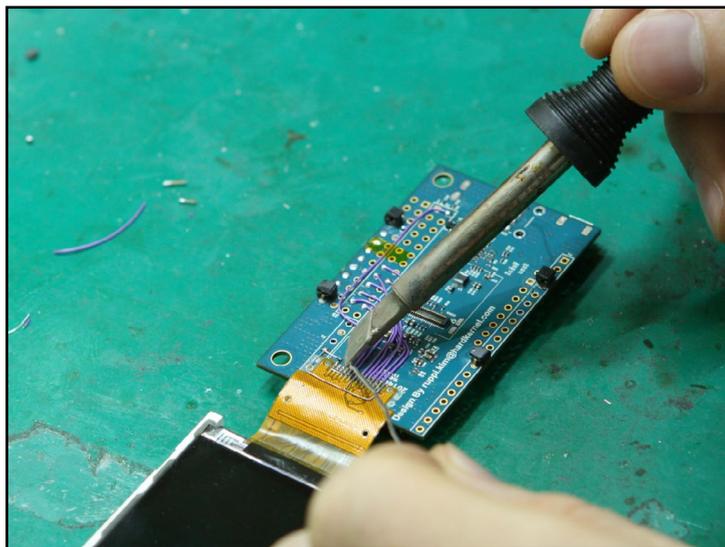
## Montaje del Reloj Inteligente



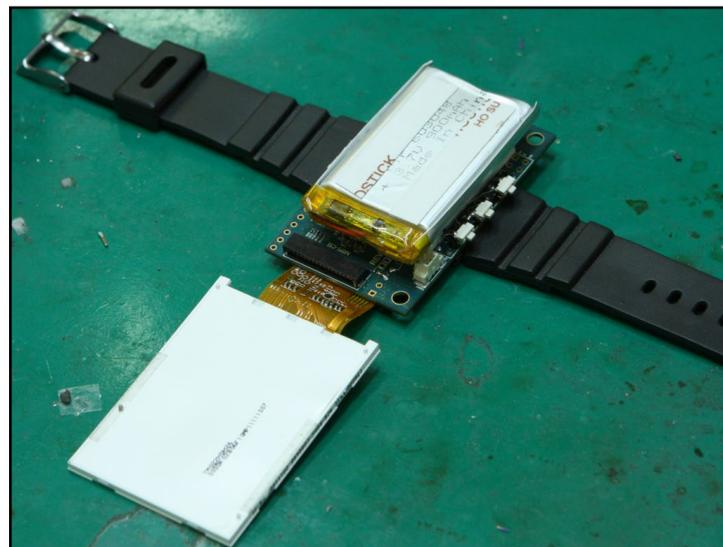
Soldar un módulo Bluetooth USB desmontado y unos cuantos pines



Colocar cinta de doble cara sobre la pantalla LCD y en la PCB ODROID-W



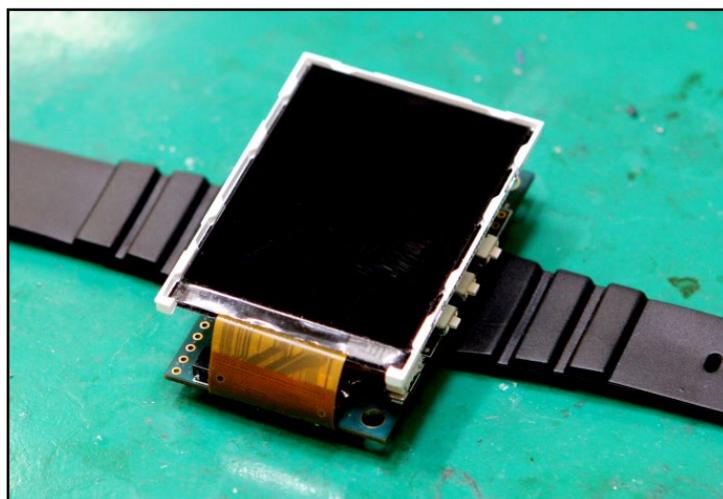
Conectar el módulo LCD TFT con varios cables.



Fijar la batería y la pantalla LCD en la PCB ODROID-W con cinta de doble cara como se muestra en la imagen.



Montar las correas de reloj. Tener en cuenta que los soportes de las bisagras deben soldarse primero



Montaje final del Reloj Inteligente ODROID con una pantalla fijada.

## BUSCAR CON GOOGLE BBS COMO SERIA GOOGLE SI FUERA SIDO INVENTADO EN LA DECADA DE LOS 80

por Rob Roy

Para una experiencia de Google al estilo retro, visite el Google BBS en <http://bit.ly/1iXk8yF>. ¡No te olvides coger tus botas de agua y el walkman!



## SOLUCIONAR EL OVESCAN EN ANDROID UNA SIMPLE APP PARA CAMBIAR LA RESOLUCION DEL ESCRITORIO

por Rob Roy

Lugalabs ofrece una aplicación para Android en Google Play que permite cambiar la resolución del escritorio con el fin de solucionar las cuestiones relacionadas con Overscan en monitores HDMI. Para más detalles, por favor visita su página de Google Play Store en <http://bit.ly/XYfZ51>.

Resolution	
Width	Height
2560	1600
Density	
320	
Overscan	
Top	Right
0	0

### Software

- SO Debian (Raspbian) con el Kernel 3.10 de Linux,
- Gestor de ventanas LXDE x11 (Si quieres lo puedes eliminar)
- Librería de entorno de trabajo Qt 4.8.4
- Aplicación Qt para mostrar el reloj y las notificaciones de SMS/Correo/llamadas
- Software de aplicación Android (en Java) para enviar la notificación al Reloj inteligente desde un Smartphone

El código fuente completo será liberado a través de Github en <http://www.github.com/hardkernel>

### Accesorios Oficiales

- Placa de expansión con LCD TFT
- Placa de expansión sin LCD TFT
- Pack de Conectores (cabezal 26-pin, ranura para cabezal 26-pin, ranura para cabezal 20-pin, ranura para cabezal 7-pin y USB host en vertical)
- Batería de reserva RTC (CR2032 con conector Molex 51021-0200 cableado)
- Batería de polímero litio (3,7 V/720mAh con conector Molex 51021-0200 cableado)
- Tarjeta Micro-SD de 8GB (Clase UHS-1) con Raspbian y drivers RTC/PMIC
- Módulo WiFi USB basado en RTL8188CUS



- Webcam USB a 720p
- Cable Micro-HDM
- Cable Micro-USB
- Fuente de alimentación 5V/2A para la placa de expansión, que sólo es necesaria cuando los dispositivos USB necesitan 600 mA o más.
- ODROID-VU (Monitor HDMI TFT 1280x800 de 9 pulgadas con pantalla táctil)

### Resumen

Esta versión diminuta del Raspberry Pi puede ejecutar un completo entorno de escritorio Raspbian por sí misma, o se puede utilizar junto con el hardware Pi para crear soluciones informáticas potentes y portátiles. Por favor envía tus preguntas y dudas a través de los foros ODROID <http://forum.odroid.com>

I can has smarticles?



Si deseas que publiquemos tu Proyecto basado ODROID, envíanos un e-mail a [odroidmagazine\(at\)gmail.com](mailto:odroidmagazine(at)gmail.com).

# TODO SOBRE LOS MODULOS EMMC DE HARDKERNEL

## LA VENTAJA DE ODROID

por Justin Lee

La CPU Exynos-4412 incluye un controlador host compatible con eMMC 4.41 con una velocidad máxima limitada a 100 MB/s, una velocidad de reloj real de 48Mhz con DDR. El ODROID-X, X2, U2, U3, Q y Q2 están basados en la CPU Exynos-4412.

La CPU Exynos-5410, utilizada por el ODROID XU, XU + E y XU-Lite, incluye un controlador host compatible con eMMC 5.0. Sin embargo, el diseño del hardware es incapaz de soportar el modo HS400 de eMMC 5.0, debido a que no había ninguna especificación pública del eMMC 5.0 cuando diseñamos el ODROID-XU. Como resultado, toda la serie XU incorpora la especificación eMMC 4.5 (HS200) que alcanza una velocidad máxima de 160 MB/s, puesto que la velocidad de reloj real es de 160MHz con SDR.

La CPU Exynos-5422 también tiene un controlador host compatible con eMMC 5.0. El ODROID-XU3 se basa en Exynos-5422 y soporta el modo eMMC HS400. El reloj real es de

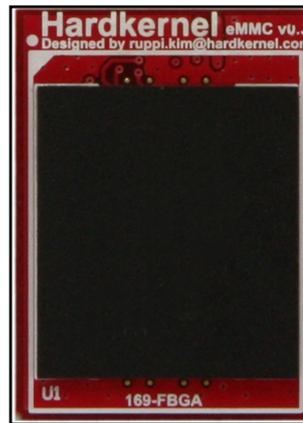
166Mhz con un ancho de banda de 330 MB/s por DDR.

Hemos estado vendiendo 2 tipos de módulos eMMC distintos: El PCB Verde o Azul tiene los chips eMMC 4.5.

La PCB rojo incorpora los chips eMMC 5.0, que es ligeramente más pequeño pero tiene el mismo tamaño de matriz BGA, para que podamos seguir usando la misma PCB.

La mayoría de la producción de eMMC se está trasladando al eMMC 5.0, por lo que ya resulta difícil adquirir chips eMMC 4.5 a excepción de los de menor capacidad como 4 y 8 GB. Recientemente, hemos empezado a usar los chips de 16GB y 32GB eMMC de Sandisk mientras que seguimos comprando eMMC de 8GB y 64GB de Toshiba.

Cuando instalas una nueva imagen SO en el módulo eMMC, necesita un lector de tarjetas USB y una placa convertidora eMMC a MicroSD (placa lectora). Por favor, consulta las siguientes imágenes para entender como usarla.



Los módulos eMMC desmontables de Hardkernel, que normalmente están soldados a la placa, son únicos para los ODROIDS

La placa lectora puede no ser detectada por algunos lectores de tarjetas USB. De modo que revisa la lista de compatibilidad en <http://bit.ly/1nPBE4i>.

La tabla de la siguiente página contiene la lista completa de los módulos eMMC que estamos vendiendo actualmente. La etiqueta redonda indica el sistema operativo y la capacidad.

Las PCBs eMMC 4.5 azules (o verdes) estarán pronto obsoletas excepto los modelos de 8 GB.

Incluso si tiene una eMMC diferente o errónea, puede utilizarla en tu ODROID si instalas el gestor arranque y la imagen SO adecuados

### Referencias

- Esquemas placa eMMC <http://bit.ly/1p4TX6N>
- Esquemas placa lectora <http://bit.ly/1p9j0Z3>

El lector eMMC se fija al módulo eMMC con un pequeño clip.



Vista superior placa lectora eMMC y el módulo eMMC insertado en el lector SD

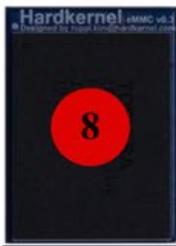
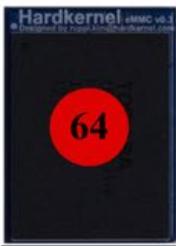


La siguiente página es una guía informativa todos los módulos eMMC de hardkernel

Vista inferior placa lectora eMMC y el módulo eMMC insertados en el lector SD



## TABLA DE REFERENCIAS DEL MODULO EMMC

Android for U2/U3 Green						
	8GB	16GB	16GB	32GB	64GB	64GB
Ubuntu For U2/U3 Red						
	8GB	16GB	16GB	32GB	64GB	64GB
Android For X2 Yellow						
	8GB	16GB	16GB	32GB	64GB	64GB
Android For XU Blue						
	8GB	16GB	16GB	32GB	64GB	64GB
Ubuntu for XU3 Light Blue						
			16GB	32GB		64GB
Android For XU3 White						
			16GB	32GB		64GB

# COMPILACION DEL KERNEL DE LINUX

CONSIGUE EL CONTROL TOTAL DE TU SISTEMA OPERATIVO COMO UN JEDI LINUX

por Venkat Bommakanti



**A** medida que progresas en el uso de Linux, es inevitable querer modificar tu configuración para adaptarla a los cambios de tu hardware, como la integración de un dispositivo externo. Estas tareas suelen requerir modificaciones en el kernel de Linux o la instalación de un nuevo driver. Este artículo ofrece información sobre cómo iniciarse en el desarrollo de un kernel "virgen" usando únicamente ODROID. Este artículo no cubre la compilación cruzada, que está relacionada con el desarrollo de un kernel ODROID usando una máquina secundaria, como un x86.

## Requisitos

1. Cualquier placa ODROID, con un adaptador de corriente apropiado.
2. Una tarjeta MicroSD 8 GB o módulo eMMC con la última imagen de Lubuntu disponible desde el sitio web de Hardkernel.
3. Una red para tener acceso a Internet y los foros ODROID.
4. Acceso SSH opcional al U3 con utilidades como PuTTY (Windows 7 +) o Terminal (Mac, Linux) para realizar los pasos desde un ordenador remoto.

## Conceptos básicos

Para empezar, es una buena idea tomar nota de la versión del Kernel inicial incluida en la imagen instalada con el siguiente comando:

```
$ uname -a
```

```
Linux u3-2 3.8.13.26 #1 SMP PRE-EMPT Wed Jul 9 22:14:37 UTC 2014 armv7l armv7l armv7l GNU/Linux
```

Esto indica que la imagen usada está basada en el kernel 3.8.y.z, desarrollado el Mié 09 de julio 2014 22:14:37 UTC.

Puedes obtener información adicional de la imagen usando los siguientes comandos y observando el resultado:

```
$ cat /proc/version
```

```
Linux version 3.8.13.26 (root@THEserver) (gcc version 4.7.2 (crosstool-NG 1.17.0) ) #1 SMP PREEMPT Wed Jul 9 22:14:37 UTC 2014
```

```
$ cat /etc/lsb-release
```

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04 LTS"
```

```
$ lsb_release -a
```

```
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 14.04 LTS
Release: 14.04
Codename: trusty
```

```
$ cat /etc/os-release
```

```
NAME="Ubuntu"
VERSION="14.04, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
```

Tras instalar un kernel recientemente desarrollado, la información de arriba es muy útil a la hora de verificar que el kernel realmente ha sido actualizado.

## Preparar el sistema

Antes de desarrollar un kernel, necesitas configurar el entorno de software apropiado. En primer lugar, inicia una sesión de terminal y cambia a usuario root, con el comando:

```
$ sudo -s
```

Todas las actividades relacionadas con el desarrollo deben ser ejecutadas por el usuario root, que se puede activar con el comando su después de introducir la contraseña de root.

Instala el grupo inicial de componentes necesarios, con el siguiente comando:

```
# apt-get install build-essential
libqt4-dev perl python git
```

Pulsa Y para aceptar la opción propuesta y permitir que se complete el comando.

Esta es una versión más completa del comando, que resalta todos los componentes necesarios para desarrollar un Kernel. Algunos de los componentes ya deben estar instalados en tu imagen, que no se verán afectados por el comando anterior ya que sólo se instalarán los componentes que faltan.

El resultado de la instalación puede sugerir componentes adicionales. Aunque no son obligatorios, los puedes instalar usando una lista de componentes como esta:

```
# apt-get install nas libqt4-
declarative-folderlistmodel
libqt4-declarative-gestures
libqt4-declarative-particles
libqt4-declarative-shaders
```

Para las imágenes de escritorio y suponiendo que el motor del sistema de ventanas por defecto está basado en X11/Xorg, debes instalar libxcb y sus paquetes adicionales. También utilizados para el desarrollo QT4. Se instala con el comando:

```
# sudo apt-get install "^libx-
cb.*" libx11-xcb-dev libglul-me-
sa-dev libxrender-dev
```

Pulsa Y para aceptar la opción propuesta. El soporte para QT4 puede no estar por defecto en imágenes de Hardkernel.

## Revisar el sistema

Es esencial asegurarse de que el sistema está listo para la compilación. En este caso, los siguientes comandos y resultados son muy útiles para verificar que todas las herramientas están disponibles y actualizadas:

```
# gcc --version
gcc-4.8.real (Ubuntu/Linaro
4.8.2-19ubuntu1) 4.8.2
Copyright (C) 2013 Free Software
Foundation, Inc.
This is free software; see the
source for copying conditions.
There is NO
warranty; not even for MERCHANT-
ABILITY or FITNESS FOR A PARTICU-
LAR PURPOSE.

# pkg-config --modversion QtCore
4.8.6

# perl --version

This is perl 5, version 18, sub-
version 2 (v5.18.2) built for
arm-linux-gnueabi-hf-thread-multi-
64int
(with 41 registered patches, see
perl -V for more detail)
Copyright 1987-2013, Larry Wall
...

# python --version
Python 2.7.6
```

```
# git --version
git version 1.9.1
```

## Coger el código fuente

Crea un directorio temporal dentro de tu directorio principal y navega hasta él usando los comandos:

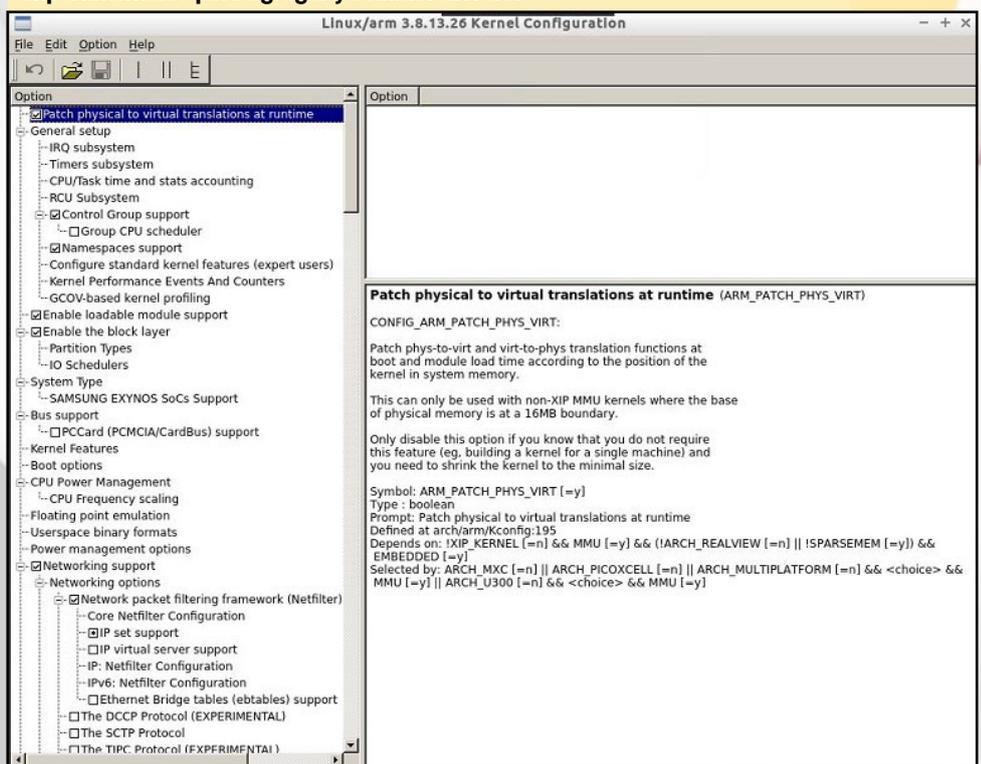
```
# mkdir hk-src
# cd hk-src
```

Tu copia del código fuente descargado se almacenará aquí. Hardkernel utiliza GitHub para el control de fuentes de su software de código abierto para U3.

Utiliza el siguiente comando para echar un vistazo al repositorio 3.8, que está recomendado para el X2 y U3:

```
# git clone --depth 1 https://
github.com/hardkernel/linux.git
-b odroid-3.8.y odroid-3.8.y
Cloning into 'odroid-3.8.y'...
remote: Counting objects: 44530,
done.
```

## Las opciones de configuración del kernel están disponibles para los usuarios experimentados para agregar y cambiar drivers.



```
remote: Compressing objects: 100%
(42780/42780), done.
remote: Total 44530 (delta 3428),
reused 16016 (delta 1191)
Receiving objects: 100%
(44530/44530), 122.03 MiB |
161.00 KiB/s, done.
Resolving deltas: 100%
(3428/3428), done.
Checking connectivity... done.
Checking out files: 100%
(42143/42143), done.
```

En este caso, vamos a coger sólo una copia del último kernel 3.8.y, en lugar de todo el repositorio. Ten en cuenta que si estás utilizando un ODROID-XU, debes copiar el 3.4.y:

```
# git clone --depth 1 https://
github.com/hardkernel/linux.git
-b odroid-3.4.y odroid-3.4.y
```

## Seleccionar un defconfig

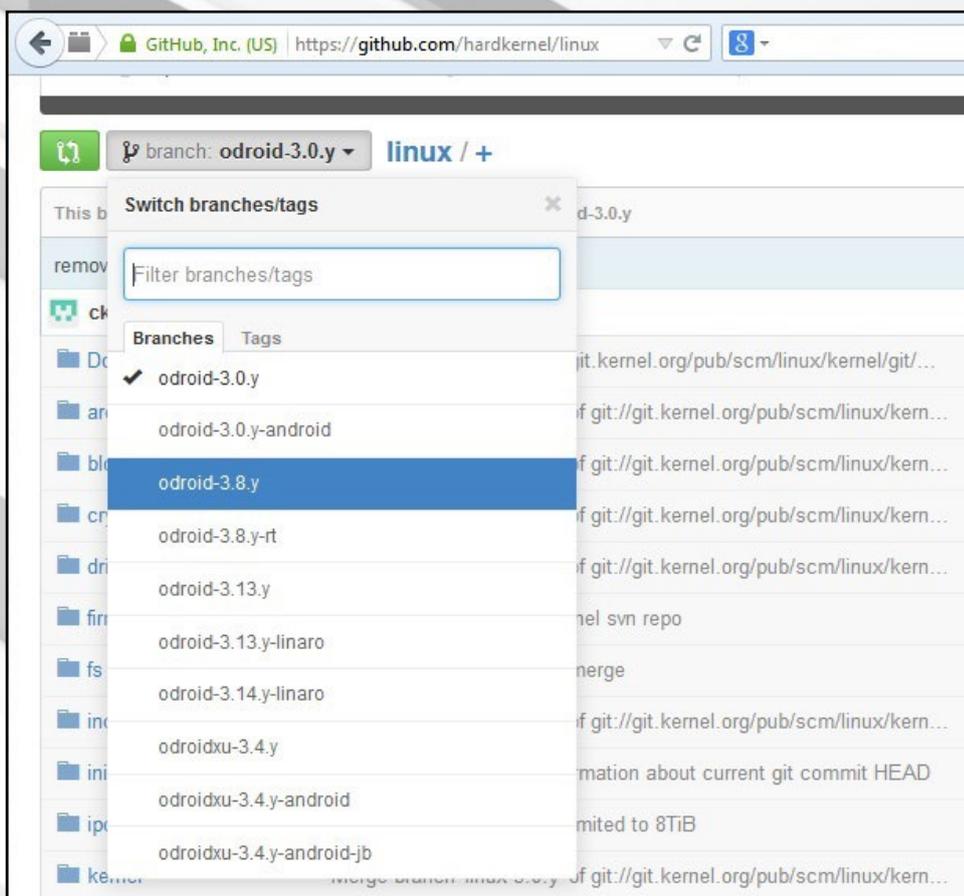
Cambia al directorio raíz del código fuente con el comando:

```
# cd odroid-3.8.y/
```

Desde aquí son lanzadas todas las actividades relacionadas con el desarrollo. Revisa la lista de archivos defconfig compatibles usando el comando:

```
# ls -lsa arch/arm/configs/
odroid*defconfig
100 -rw-r--r-- 1 root root 101207
Jul 14 13:11 odroidu_defconfig
96 -rw-r--r-- 1 root root 96792
Jul 14 13:11 odroidx2_defconfig
96 -rw-r--r-- 1 root root 96797
Jul 14 13:11 odroidx_defconfig
```

En nuestro caso, seleccionaremos la versión odroidu\_defconfig, que contiene las opciones de configuración del hardware específicas para el U3. Ahora, realiza el primer paso del desarrollo ejecu-



Hardkernel maintains open source-code repositories that are downloadable for free.

tando el siguiente comando e ignorando cualquier advertencia:

```
root@u3-2:~/odroid-3.8.y# make
odroidu_defconfig
HOSTCC scripts/basic/fixdep
...
#
# configuration written to .config
#
```

## Configurar módulos adicionales

En este ejemplo, puesto que se ha añadido soporte para QT4 (xconfig es una interfaz de usuario QT4), el módulo xconfig necesita ser desarrollado usando el siguiente comando con el fin de iniciar un editor para la selección de opciones:

```
# make xconfig
CHECK qt
MOC scripts/kconfig/qconf.moc
HOSTCXX scripts/kconfig/qconf.o
HOSTLD scripts/kconfig/qconf
```

Los usuarios novatos deben revisar y examinar la lista de opciones, pero no modificar ninguna. Los usuarios experimentados pueden desear ajustar las selecciones en el menú de configuración del kernel basándose en su amplio conocimiento del kernel.

## Usar recursos básicos

Para entender las implicaciones del proceso de desarrollo en el uso de recursos, es útil disponer de una visión del uso de los recursos antes y durante del proceso de desarrollo, como se muestra en la captura de pantalla del comando top.

## Desarrollar Kernel y los módulos elegidos

Para compilar el kernel lo más rápido posible, utilizando los 4 procesadores disponibles en el U3. Escriba el siguiente comando en la ventana de terminal:

```
# top
top - 13:35:15 up 2:33, 4 users, load average: 0.09, 0.13, 0.14
Tasks: 142 total, 1 running, 141 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.9 us, 0.7 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem: 2071392 total, 1519108 used, 552284 free, 78556 buffers
KiB Swap: 0 total, 0 used, 0 free. 1206844 cached Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM   TIME+  COMMAND
  929 root        20   0  88388 49212 34172 S  10.9  2.4   3:45.61 Xorg
 1066 root        20   0  38852 26304 14756 S   3.0  1.3   1:34.00 xllvnc
 4801 odroid     20   0   2572  1184   764 R   0.7  0.1   0:00.07 top
 1276 root        20   0     0     0     0 S   0.3  0.0   0:00.23 kworker/0:2
 1614 odroid     20   0 202540 15084 9492 S   0.3  0.7   0:17.28 lxterminal
 4437 root        20   0     0     0     0 S   0.3  0.0   0:01.77 kworker/u:1
    1 root        20   0   3532  2160  1100 S   0.0  0.1   0:04.60 init
...
```

**Uso de recursos antes de que comience el desarrollo**

```
# make -j5 zImage modules
```

Se recomienda llevar a cabo el proceso de desarrollo en un ODROID que utilice refrigeración por ventilador. Puedes experimentar con los valores -j más altos sólo si eres capaz de mantener la temperatura del ODROID por debajo de los límites térmicos. Con el comando anterior, el proceso de desarrollo puede tardar entre 15 y 30 minutos, dependiendo de la actividad del sistema preexistente. Presta atención a las advertencias, en especial a las relacionadas con tus cambios, si hay alguno.

Es importante tener presente el uso tan intenso que se hace de la CPU (90%) durante el proceso de desarrollo, como se ve en la imagen de abajo. Como era de esperar, el ventilador gira sin parar!

**Instalar kernel y módulos**

Los módulos y el núcleo recién desarrollados se pueden instalar usando los siguientes comandos:

```
# top
top - 13:38:36 up 2:37, 4 users, load average: 5.59, 2.45, 1.00
Tasks: 164 total, 6 running, 158 sleeping, 0 stopped, 0 zombie
%Cpu0 : 89.1 us, 8.9 sy, 0.0 ni, 0.3 id, 0.0 wa, 0.0 hi, 1.7 si, 0.0 st
%Cpu1 : 94.7 us, 5.3 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu2 : 93.4 us, 6.6 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu3 : 91.4 us, 8.6 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 2071392 total, 1612744 used, 458648 free, 83180 buffers
KiB Swap: 0 total, 0 used, 0 free. 1227700 cached Mem
```

**Uso de recursos (top con la opción P) durante el proceso de desarrollo**

```
# cp arch/arm/boot/zImage /media/
boot/zImage
# make modules_install
```

**Actualizar initramfs**

Antes de arrancar usando el nuevo kernel, se deben ejecutar los siguientes pasos para crear el sistema de archivos RAM inicial (initramfs). En primer lugar, copia la configuración actual a /boot, así update-initramfs podrá ejecutarse correctamente con el comando:

```
# cp .config /boot/config-`cat include/config/kernel.release`
# ls -lsa /boot/conf*
100 -rwxrwxrwx 1 root root
101420 Jun 13 01:02 /boot/config-3.8.13.23
100 -rwxrwxrwx 1 root root
101207 Jul 14 14:15 /boot/config-3.8.13.26
```

Vuelve a crear initramfs con el comando:

```
# update-initramfs -c -k `cat \
include/config/kernel.release`
update-initramfs: Generating /
boot/initrd.img-3.8.13.26
```

Convierte el initrd actual en una versión compatible con u-boot, añadiendo una cabecera u-boot de 64byte:

```
# mkimage -A arm -O linux -T ram-
disk -C none -a 0 -e 0 -n uInitrd
-d /boot/initrd.img-`cat include/
config/kernel.release` /boot/uIni-
trd-`cat include/config/kernel.
release`
Image Name: uInitrd
Created: Mon Jul 14 14:17:02
2014
Image Type: ARM Linux RAMDisk
Image (uncompressed)
Data Size: 2196336 Bytes =
2144.86 kB = 2.09 MB
Load Address: 00000000
Entry Point: 00000000
```

Guarda el uInitrd actual usando el comando:

```
# cp /media/boot/uInitrd /boot/
uInitrd-`uname -r`
# ls -lsa /boot/u*
2144 -rwxrwxrwx 1 root root
2195134 Jul 13 22:43 /boot/uIni-
trd
2144 -rwxrwxrwx 1 root root
2195134 Jul 14 14:17 /boot/uIni-
trd-3.8.13.26
```

Activa el nuevo uInitrd con el comando:

```
# cp /boot/uInitrd-`cat include/
config/kernel.release` /media/
boot/uInitrd
# ls -lsa /boot/u*
2144 -rwxrwxrwx 1 root root
2195134 Jul 14 14:18 /boot/uIni-
trd
2144 -rwxrwxrwx 1 root root
2195134 Jul 14 14:17 /boot/uIni-
trd-3.8.13.26
```

## Reiniciar

Si no ha habido errores, reinicia el sistema utilizando los siguientes comandos para que el nuevo kernel pueda tener efecto:

```
# sync && reboot
```

Tras completarse el reinicio, puedes verificar que el nuevo núcleo está instalado con el comando `uname` y comparar la fecha de registro y la versión del kernel de antes y después del desarrollo:

```
# uname -a
Linux u3-2 3.8.13.26 #1 SMP PRE-
EMPT Mon Jul 14 14:02:33 PDT 2014
armv7l armv7l armv7l GNU/Linux
```

Si tiene alguna pregunta, por favor visita al sección del foro en <http://bit.ly/1ucMAIN>



# REPRODUCTOR DE YOUTUBE ALTERNATIVO USA TAMPERMONKEY PARA VER VIDEOS

por Jeremy "Cartridge" Kenney



¿Youtube te coge por defecto el reproductor HTML5, y los complementos de Chrome Web Store que has probado no te han funcionado con la reproducción Flash? ¿Estás cansado de no poder activar la resolución deseada en Youtube o de ese diseño blanco soso de Youtube? Uno o dos trucos sobre Youtube te ayudarán a abordar estas cuestiones, e incluso te darán la opción de activar la pantalla completa o la versión con ventanas para una ejecución más rápida.

Un truco muy sencillo y muy bueno es utilizar Tampermonkey y un pequeño script. Tampermonkey es como Greasemonkey (Firefox) pero para Chrome.

Para aquellos que no conocen Greasemonkey, es una extensión que inyecta scripts en la web que estás usando y otras funciones interesantes utilizando scripts.

Puedes ir a la tienda Web de Chrome y hacerte con la Extensión en <http://bit.ly/1iL5YRd>. Puede utilizar scripts en cualquier sitio web que no tenga protección contra scripts.

Existe un sinfín de posibilidades para los desarrolladores y una sencilla interfaz para los principiantes que permite una rápida y fácil gestión de los scripts.

Si está usando una versión anterior de Chromium para utilizarla en compilaciones de Debian antiguas, instala Legacy Tampermonkey en ese caso. Para ello, debe descargar el CRX desde su sitio web

## Tampermonkey está disponible desde Chrome Web

Download
Features

Tampermonkey is a free browser extension and the most popular userscript manager for Blink-based Browsers like Chrome and Opera Next. Even though Google Chrome does have native support for userscripts, Tampermonkey can give you much more convenience in managing your userscripts. It provides features like easy script installation, automatic update checks, a overview what scripts are running at a tab and there is a good chance that scripts that are incompatible to Google Chrome run with Tampermonkey.

Tampermonkey is available for Google Chrome, Opera, Chromium and a lot of their derivatives like CoolNovo and Rockmelt and it is installed in just a minute, so give it a try!

+1 | 163
f Like
Share | 4k

Download/Installation



Tampermonkey Stable

Chrome >= 22



Tampermonkey Beta

Chrome >= 22



Tampermonkey Legacy

Chrome >= 17  
Chrome < 27



Tampermonkey Android

Android >= 2.2

Download Mozilla Firefox

mozilla-firefox.onfreedownload.com

Official 2014 Version for MacOS. Hurry Up Right Now: Free Download!

>

# INTERESANTES COMANDOS LINUX CURIOSOS PROGRAMAS PARA TU SIGUIENTE DESCANSO

por Rob Roy

**T**e gustaría ver algunos de los lados divertidos de Linux... Prueba estos comandos: **¡Todos a bordo!**

```
$ sudo apt-get install sl
$ sl
```

## Predecir el futuro

```
$ sudo apt-get install \
fortune-mod
$ fortune
```

## Mugir

```
$ sudo apt-get install cowsay
$ cowsay 'ODROIDS are cool!'
```

## Vaca adivina

```
$ sudo apt-get install cowsay
&& sudo apt-get install \
fortune-mod
$ fortune | cowsay
```

## Adivinación

### vaca XII

```
$ sudo apt-get install xcowsay
&& sudo apt-get install \
fortune-mod
$ fortune | xcowsay
```

¡La vaca XII siempre tiene la razón!



en <http://tampermonkey.net> Para instalar, haga clic derecho y pincha en Guardar como para guardarlo en una carpeta de tu elección. A continuación, abre la configuración de Chromium y haga clic en la pestaña Extensiones. Arrastra y suelta el archivo CRX que has descargado sobre la página de Extensiones. Por último, prueba las nuevas opciones en <http://www.youtube.com> y visualiza un video.

## Centro Youtube

Este script te da acceso a opciones

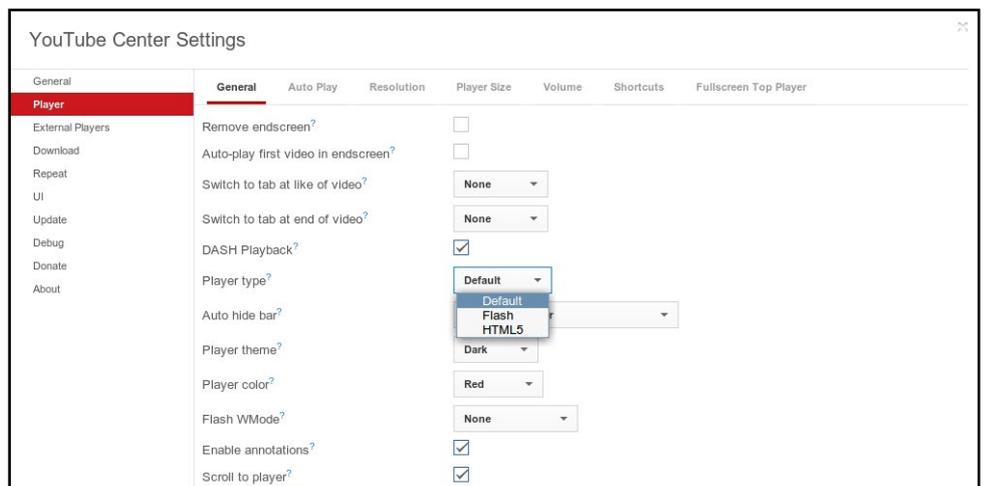
instalar la script que has elegido.

Haga clic en instalar, a continuación, abre el Youtube y una ventana te mostrará cuales son los nuevos ajustes.

Haz clic en la pestaña player y activa Flash Playback en Player Type. Si tienes una conexión rápida, desactiva Dash Playback para una carga más rápida y solucionar diversos problemas que aparecen con una señal wifi débil.

Ahora tiene acceso a más opciones de las que puedes imaginar, y puedes personalizar Youtube a tu gusto, incluso cambiar el tema y color del reproductor

El script se puede desactivar haciendo



Cambiando los ajustes del reproductor en el Centro Youtube

que nunca has visto antes en Youtube. Puedes hacer uso de la Auto Resolución para ajustar el vídeo a la resolución deseada, fijar la anchura y la altura de la ventana de reproducción, activar o desactivar el uso de Dash Playback (bloque de memoria intermedia en lugar de cargar el video por completo para su uso en 3G).

También puede agregar un botón de descarga en cada video para seleccionar el formato deseado, añadir un botón de repetición si te gusta la canción que estás escuchando y muchas más funciones.

Una de las opciones más útiles es la Alternativa de HTML5, Flash Playback. Ve al Github del Centro Youtube en <http://bit.ly/MANHYG>, desplázate hacia abajo para descargar y haz clic en Dropbox. Tu Tampermonkey debe abrirse al instante y te pedirá si deseas

hacer clic en el botón Tampermonkey en la barra de extensión al lado de la barra de direcciones. A continuación, hacer clic en Dashboard y desmarca las extensiones que desees.

¡Diviértete disfrutando con YouTube!

Tenemos funcionando Youtube, y ahora lo único que hace es subir videos de sus habilidades como jugador experto.



# ODROID-U3 VS ODROID-U3+

## LA SIGUIENTE GENERACION EN LA SERIE ODROID-U

por Justin Lee

**H**emos estado enviando el ODROID-U3 + (Rev 0.5) desde finales de junio de 2014, con 4 principales mejoras sobre el U3 inicial:

- El Puerto Micro-USB es capaz de trabajar en modo USB host.
- El Circuito de protección de energía del USB host ha sido cambiado del AP2411 al NCP380.
- El Circuito AP2331 HDMI actual se ha colocado en el lugar correcto.
- Se ha añadido el puerto SPI HW.

### Modo USB host con puerto Micro-USB

Al añadir el IC RT9715 de control de energía, el puerto micro-USB puede trabajar tanto en modo USB host como en modo dispositivo USB, al que llamamos Dual-Role-Device (DRD). Sólo tiene que conectar un cable OTG-a-Host y tendrás otro USB host disponible para periféricos, para que pueda disfrutar de 4 puertos USB sin utilizar un Hub USB externo. ¡Es un puerto de doble uso que



Cable OTG a Host que permitir usar del puerto Micro-USB como un 4º puerto USB host

funciona en ambos sentidos!

### Nuevo protector de energía USB host IC NCP380

El AP2411 fue usado en la PCB Rev 0.2 como un protector de energía para el bus USB. Sin embargo, algunos usuarios informaron que podría resultar dañado por una descarga eléctrica sobre el puerto USB.

Para solventar este problema, hemos eliminado el chip y añadimos uno más resistente, NCP380 que incluye varias funciones fiables como el bloqueo por baja tensión, arranque seguro integrado, protección térmica, desconexión segura, protección ante voltaje inverso, conformidad ESD con IEC61000-4-2 (Nivel 4), 8.0 kV (Contacto) y 15 kV (Aire).

### IC HDMI de bloqueo inverso

Algunos monitores o televisores tienen fugas de corriente en el puerto

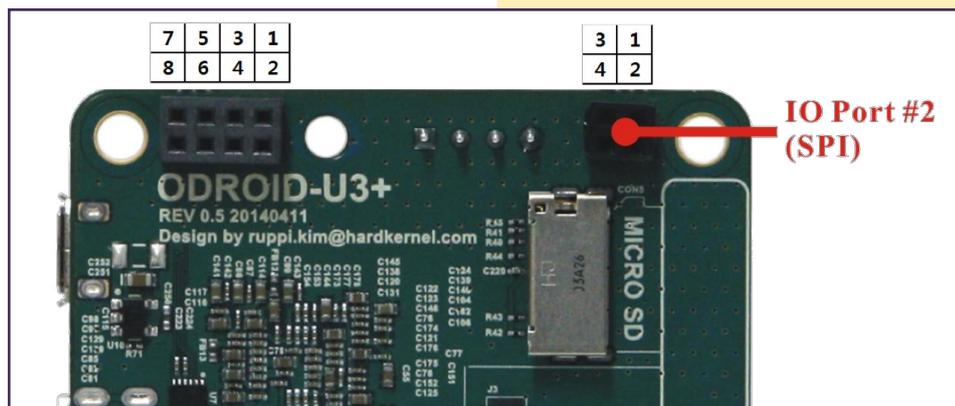
La siguiente generación de la serie U mejora con el popular U3

HDMI. Poner el PMIC en un estado indefinido evita la función de autoencendido. Movimos el IC de protección inversa desde lateral del conector HDMI al lateral de la CPU/PMIC, solucionando el problemático autoencendido. Ahora podemos conectar la placa de forma automática simplemente insertando la clavija DC de energía aunque el HDMI esté conectado.

### Puerto SPI HW

El bus Serial Peripheral Interface (SPI) es un enlace de datos en serie síncrono que funciona en modo Bidireccional. El bus SPI está incluido en el nuevo conector J5 de 4 pines (Puerto E/S #2) en la PCB. Ten en cuenta que los puertos SPI sólo soportan el modo maestro y una interfaz de 1.8V como los otros puertos E/S en el procesador Exynos. Puede usar 4 pines en modo GPIO, así como el modo SPI. Tras el arranque los pines están por defecto en modo GPIO.

Vista lateral inferior de la distribution de Pin



Pin Number	SPI Function	CPU GPIO pin	Node Number
1	SCLK	GPB[4]	20
2	nSS	GPB[5]	21
3	MOSI	GPB[7]	23
4	MISO	GPB[6]	22

## Descripciones de pin

Este script de ejemplo ajusta todos los pines en modo salida y conmuta 5 veces a una frecuencia de 1 Hz. En el momento de lanzar la aplicación, se descarga los drivers de SPI para asegurarse del modo GPIO y libera los GPIOs para el siguiente uso al final.

```
modprobe spi-s3c64xx
modprobe odroid-ioboard
dmesg | grep ioboard
modprobe -r odroid-ioboard
modprobe -r spi-s3c64xx

count=0
stop=5

echo 20 > /sys/class/gpio/export
echo 21 > /sys/class/gpio/export
echo 22 > /sys/class/gpio/export
echo 23 > /sys/class/gpio/export

echo out > /sys/class/gpio/
gpio20/direction
echo out > /sys/class/gpio/
gpio21/direction
echo out > /sys/class/gpio/
gpio22/direction
echo out > /sys/class/gpio/
gpio23/direction

while :
do
count=$((count+1))

echo 1 > /sys/class/gpio/gpio20/
value;
echo 1 > /sys/class/gpio/gpio21/
value;
echo 1 > /sys/class/gpio/gpio22/
value;
```

## Distribuciones del pin GPIO en el U3+

```
echo 1 > /sys/class/gpio/gpio23/
value;
echo "GPB[4],GPB[5],GPB[6],GPB[7]
set output HIGH"
sleep 1;

echo 0 > /sys/class/gpio/gpio20/
value;
echo 0 > /sys/class/gpio/gpio21/
value;
echo 0 > /sys/class/gpio/gpio22/
value;
echo 0 > /sys/class/gpio/gpio23/
value;
echo "GPB[4],GPB[5],GPB[6],GPB[7]
set output LOW"
sleep 1;

if [ $count -eq $stop ]; then
echo "GPIO TEST STOP"
echo 20 > /sys/class/gpio/unex-
port
echo 21 > /sys/class/gpio/unex-
port
echo 22 > /sys/class/gpio/unex-
port
echo 23 > /sys/class/gpio/unex-
port
exit 0
fi
done
```

Vamos a acceder al modo SPI. Si la versión de tu kernel es anterior al 17 julio de 2014, primero debe actualizar el kernel.

## Módulo driver SPI

Para uso genérico del SPI, primero carga el módulo para activar el host SPI.

```
$ sudo modprobe spi-s3c64xx
```

A continuación, carga el módulo para activar el SPI genérico, y tendrás un nodo SPI estándar.

```
$ sudo modprobe spidev
$ /dev/spidev1.0
```

El procedimiento para flash serie también empieza cargando el módulo para activar el SPI host.

```
$ sudo modprobe spi-s3c64xx
```

A continuación, carga el driver flash serie (misc) y tendrás un nodo flash serie.

```
$ sudo modprobe odroid-ioboard
$ /dev/ioboard-spi-misc
```

Ten en cuenta que spidev.ko y odroidioboard.ko no se puede cargar al mismo tiempo.

Driver de interfaz SPI host:  
<http://bit.ly/1qz8gHs>

Driver generico SPI  
<http://bit.ly/1rILc5X>

Emplo driver con Flash SPI Serie  
<http://bit.ly/1sOpccA>

Hemos probado una memoria Flash SPI serie SST25WF020A con el nuevo IO Shield, y la velocidad máxima fue de 40Mhz del reloj SPI. Para obtener más información sobre el driver SPIDEV, por favor visite <http://bit.ly/WAgUYC>

El ODROID-U3 Rev 0.5 tiene algunas mejoras de hardware, por eso se le llamó ODROID-U3+. Puede localizar los últimos esquemas del U3 Rev 0.5 en: <http://bit.ly/1nd5GDx> Además, el nuevo IO Shield Rev 0.3 del U3 ha sido mejorado para incluir una nueva memoria flash SPI que se puede conectar a un bus SPI (<http://bit.ly/1nSPTFj>).

# INSTALAR UN SERVIDOR WEB

USA LIGHTTPD Y NGINX PARA PUBLICAR TUS SITIOS WEB

por @hamiko y Venkat Bommakanti

Uno de los usos más comunes de una máquina Linux es la creación un servidor web para alojar sitios web, que son un componente esencial para cualquier estrategia comercial o proyecto personal. Si deseas crear un servidor de producción o uno personal para la intranet de tu casa, este artículo te ayudará los pasos necesarios para poner en marcha dos de los servidores web más populares y ligeros: nginx y lighttpd (alias lighty).

Aunque el servidor Apache también es una opción muy común para el alojamiento web en Linux, éste está tratado extensamente en la Web de PHP <http://bit.ly/1wXDju1> con instrucciones específicas para ODROID en <http://bit.ly/1rIb9Te>

Las instrucciones también incluyen soporte para un servidor web seguro (HTTPS basado en openssl) y aplicaciones Web para scripts (PHP5-FPM). FPM que significa FastCGI Process Manager, es una alternativa muy buena a la clásica implantación de FastCGI. Si te preocupa la configuración y los problemas derivados de la seguridad en el alojamiento de sitios web, te recomiendo que te informes muy bien y pongas en práctica los procedimientos adecuados antes de habilitar el acceso a Internet. También te recomiendo actualizar periódicamente el sistema operativo.

## Requisitos

1. Una placa ODROID con el adaptador de corriente adecuado.

2. Una tarjeta MicroSD de 8+ GB o módulo eMMC con la imagen más reciente de Lubuntu para U3 disponible en el sitio web Hardkernel.  
3. Una red donde el dispositivo tenga acceso a Internet y los foros ODROID.  
4. Acceso SSH opcional para U3 a través de utilidades como PuTTY (MS Windows 7+) o Terminal (Mac, Linux) para realizar los pasos desde un ordenador remoto.

## Instalación del Servidor

Aunque es posible instalar ambos servidores web en la U3 al mismo tiempo, se supone que con los siguientes pasos vas a instalar sólo un servidor web. Para un uso simultáneo, debemos solucionar en primer lugar los conflictos de configuración, como el uso del puerto web 80 por ambos servidores.



INTERESANTES  
COMANDOS  
LINUX  
CURIOSOS  
PROGRAMAS PARA TU  
SIGUIENTE DESCANSO

por Rob Roy

Quieres ver más lados divertidos de Linux... Prueba estos comandos:

## Nombre extraño

```
$ sudo apt-get install toilet
$ toilet -f mono12 -F metal \
  ODROID Magazine
```



¡El comando toilet afortunadamente no hace lo que piensas!

## Introducir matriz

```
$ sudo apt-get install cmatrix
$ cmatrix
```

## Cazar al ratón

```
$ sudo apt-get install oneko
$ oneko
```

## ¿Quieres jugar a un juego?

```
$ sudo apt-get install espeak
$ espeak \
  'Do you want to play a game?'
```

## Fuego cálido y acogedor

```
$ sudo apt-get install libaa-bin
$ aafire
```

## 1 - Lighttpd y Nginx

**Configurar los derechos de root, ejecutando el comando** `sudo -s`

## 2 - Nginx

**Instalar nginx usando el comando:**

```
# sudo apt-get install nginx
```

**Iniciar nginx usando el comando:**

```
# sudo service nginx stop
# sudo service nginx start
```

## 2 - Lighttpd

**Instalar lighttpd usando el comando:**

```
# sudo apt-get install lighttpd
```

**Iniciar lighttpd usando el comando:**

```
# sudo service lighttpd stop
# sudo service lighttpd start
```

## 3 - Nginx y Lighttpd

**Iniciar un navegador web como firefox e introducir la dirección ip del dispositivo, usando la URL `http://<dirección ip del u3>/`**



## 4 - Nginx

**Instalar las claves/certificados ssl autogenerados, utilizando los siguientes (de única línea) comandos (el comando use te ayudará a obtener información sobre parámetros concretos):**

```
# mkdir /etc/nginx/ssl
# sudo openssl req -x509 -nodes -days 365
-newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key
-out /etc/nginx/ssl/nginx.crt
```

**Primero se deben crear los directorios de destino adecuados, si no están ya presentes.**

**Cuando se te solicite información sobre certificados, puedes introducir información como:**

```
Country Name (2 letter code) [AU]:US
State or Province Name [Some-State]:CA
Locality Name (eg, city):San Jose
Organization Name:YourCompany
```

## 4 - Lighttpd

**Comprobar si SSL ha sido activado en la aplicación lighttpd, usando el comando:**

```
# lighttpd -v
lighttpd/1.4.33 (ssl) - a light and fast webserver
Build-Date: Jan 28 2014 17:19:37
```

**SSL debe estar activado. Ahora, instala las claves/certificados ssl autogenerados, con los siguientes (de única línea) comandos (el comando use te ayudará a obtener información sobre parámetros concretos):**

```
# mkdir /etc/lighttpd/ssl
# sudo openssl req -x509 -nodes -days 365
-newkey rsa:2048 -keyout /etc/lighttpd/ssl/lighttpd.
key -out /etc/lighttpd/ssl/lighttpd.crt
```

**Primero se deben crear los directorios de destino adecuados, si no están ya presentes.**

## 4 - Nginx

```
Organizational Unit Name:Engineering
Common Name:YourName
Email Address:you@yourcompany.com
```

### Comprobar el certificado generado, con el comando:

```
# openssl s_client -connect localhost:443 -re-
connect
```

**Si la información del certificado introducida anteriormente es enviada, entonces podemos suponer que el certificado se ha generado correctamente.**

### Añadir la configuración de SSL a la configuración de nginx, utilizando los siguientes pasos:

```
# cd /etc/nginx/sites-available
# cp default default-orig
# medit default
```

### El bloque del servidor debe coincidir con este ejemplo:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    # ssl support
    listen 443 ssl;

    root /usr/share/nginx/html;
    index index.html index.htm;

    # Make site accessible from http://localhost/
    server_name localhost;
    ssl_certificate /etc/nginx/ssl/nginx.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a
404.
        try_files $uri $uri/ =404;
        # Uncomment to enable naxsi on this location
        # include /etc/nginx/naxsi.rules
    }
}
```

### Reiniciar nginx usando los comandos:

```
# sudo service nginx stop
# sudo service nginx start
```

## 4 - Lighttpd

**Introducir la información del certificado como está indicado para nginx y comprobar certificado instalado.**

**Además, generar el archivo .pem correspondiente usando los comandos:**

```
# cd /etc/lighttpd/ssl
# cat lighttpd.key lighttpd.crt | sudo tee
lighttpd.pem
```

**Este archivo .pem se utiliza más adelante en el archivo de configuración de lighttpd.**

**Añadir la configuración SSL a la configuración lighttpd, usando los siguientes pasos:**

```
# cd /etc/lighttpd/
# cp lighttpd.conf lighttpd.conf-orig
# medit lighttpd.conf
```

### Añadir el siguiente bloque:

```
$SERVER["socket"] == ":443" {
    ssl.engine = "enable"
    ssl.pemfile = "/etc/lighttpd/ssl/lighttpd.pem"
    server.document-root = "/var/www"
}
```

**Compruebe si la configuración se ha actualizado correctamente, utilizando el comando (de única línea):**

```
# sudo lighttpd -t -f /etc/lighttpd/lighttpd.
conf
Syntax OK
```

### Reinicia lighttpd usando los comandos:

```
# sudo service lighttpd stop
# sudo service lighttpd start
```

**Reinicia firefox e introducir la dirección ip del dispositivo, usando la URL HTTPS https://<dirección ip u3>.**

**En el primer acceso, varios cuadros de diálogo solicitarán una respuesta. Selecciona las siguientes opciones:**

```
I understand the Risks
Add Exception
```

#### 4 - Nginx

Reiniciar firefox e introducir la dirección ip del dispositivo, usando la URL HTTPS `https://<dirección ip u3>`

En el primer acceso, varios cuadros de diálogo solicitarán una respuesta. Selecciona las siguientes opciones:

```
I understand the Risks
Add Exception
Confirm Security Option
```

La pantalla de bienvenida debería aparecer como se muestra en la imagen de la derecha.

El icono de exclamación del círculo indica que SSL se ha configurado correctamente. Tenga en cuenta que el signo de exclamación indica que un certificado auto-generado esta en uso.

#### 4 - Lighttpd

La pantalla de bienvenida debería aparecer como se muestra en la imagen de abajo.

El icono de exclamación del círculo indica que SSL se ha configurado correctamente. Tenga en cuenta que la exclamación indica que un certificado auto-generado esta en uso.



#### 5 - Nginx y Lighttpd

Instalar PHP5-FPM y otros modulos usando el comando:

```
# apt-get install php5-cgi autoconf automake autotools-dev curl libapr1 libtool curl libcurl4-openssl-
dev php-pear php-xml-parser php5 php5-cli php5-common php5-curl php5-dev php5-gd php5-sqlite php5-fpm php5-
mysql
```

#### 6 - Nginx

Configurar nginx para soporte php, utliando las siguientes opciones de configuración (en base al socket):

```
# cd /etc/nginx/sites-available/
# medit default

...
index index.html index.htm index.php;

location ~ .php$ {
fastcgi_split_path_info ^(.+\.php)(/.+)$;

fastcgi_pass unix:/var/run/php5-fpm.sock;

fastcgi_index index.php; include fastcgi_
params;
}
```

#### 6 - Lighttpd

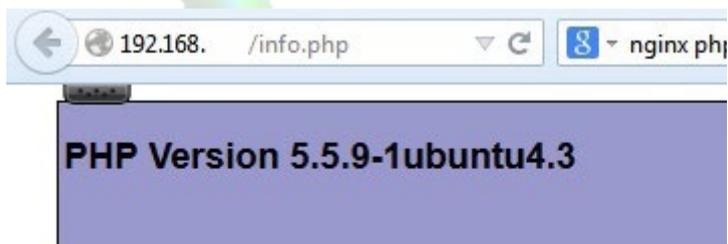
Actualizar lighttpd para soporte php, usando las siguientes opciones de configuración (en base al socket):

```
# cd /etc/lighttpd/conf-
available/
# cp 15-fastcgi-php.conf 15-fastcgi-php.conf-orig
# medit 15-fastcgi-php.conf
```

y la siguiente configuración (en base al socket):

```
fastcgi.server += ( ".php" =>
(
"socket" => "/var/run/php5-fpm.sock",
"broken-scriptfilename" => "enable"
))
)
```

Activar la configuración fastcgi adicional, usando los comandos:



System	Linux u3-2 3.8.13.26 #1 SMP PREEMPT
Build Date	Jul 7 2014 17:14:15
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File	/etc/php5/fpm

Una vez que hayas llegado a esta pantalla, ¡Te felicito por haber logrado instalar un servidor web de nivel avanzado!

## 6 - Lighttpd

```
# lighttpd-enable-mod fastcgi
# lighttpd-enable-mod fastcgi-php
```

**Estos comandos crean ficheros/enlaces:**

```
# ls -lsa /etc/lighttpd/conf-enabled
total 0
lrwxrwxrwx 1 root root 33 Jul 13 16:14 10-fastcgi.conf -> ../conf-available/10-fastcgi.conf
lrwxrwxrwx 1 root root 37 Jul 13 16:15 15-fastcgi-php.conf -> ../conf-available/15-fastcgi-php.conf
```

## 7 - Nginx y Lighttpd

**Actualizar la configuración fpm:**

```
# cd /etc/php5/fpm/pool.d/
# cp www.conf www.conf-orig
# medit www.conf
```

**Añadir la siguiente configuración (en base al socket):**

```
listen = /var/run/php5-fpm.sock
```

**Actualizar el archivo de configuración php:**

```
# cd /etc/php5/fpm/
# medit php.ini
```

**Activar la siguiente línea de config:**

```
cgi.fix_pathinfo=1
```

## 8 - Nginx

**Crear un script de prueba .php, usando los comandos:**

```
# grep -w '^[^#]*root' /etc/nginx/sites-available/default root /usr/share/nginx/www;
# cd /usr/share/nginx/www
# echo '<?php phpinfo(); ?>' > info.php
```

**Reiniciar nginx usando los comandos:**

```
# service php5-fpm stop &&
sudo service nginx stop
# sudo service nginx start && service php5-fpm start
```

**Reiniciar Firefox y navegar hasta el archivo .php usando la URL <http://<direccion ip u3>/info.php>**

**Verificar que la página PHP Info es correcta.**

## 8 - Lighttpd

**Crear un script de prueba .php, usando los comandos:**

```
# grep -w '^[^#]*root' /etc/nginx/sites-available/default root /usr/share/nginx/www;
# cd /usr/share/nginx/www
# echo '<?php phpinfo(); ?>' > info.php
```

**Reiniciar lighttpd usando los comandos:**

```
# service php5-fpm stop &&
sudo service nginx stop
# sudo service nginx start && service php5-fpm start
```

**Reiniciar Firefox y navegar hasta el archivo .php usando la URL <http://<direccion ip u3>/info.php>**

**Verificar que la página PHP Info es correcta.**

## Recursos Adicionales

Puedes leer más sobre nginx en <http://wiki.nginx.org>, y aprender sobre Lighttpd en <http://lighttpd.net>.

# ODROID-VU ASEQUIBLE PANTALLA TÁCTIL HDMI USB DE 9"

UNA PANTALLA  
MUTITÁCTIL PORTÁTIL  
PARA ANDROID, LINUX  
Y WINDOWS.

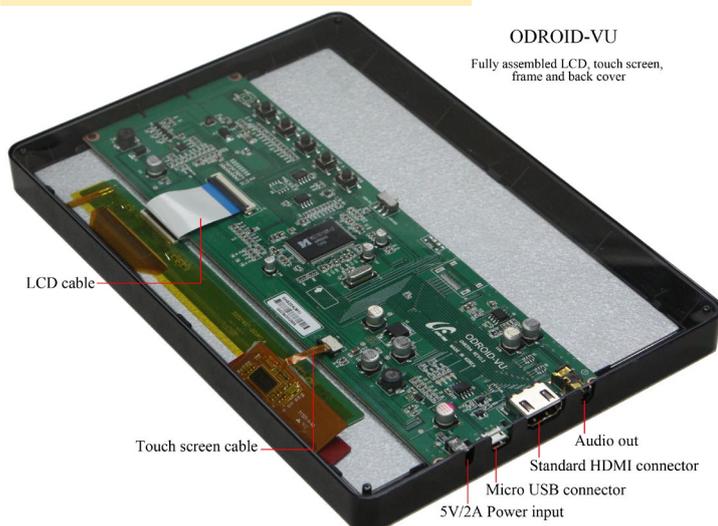
por Justin Lee

**P**ensarás que con el enorme crecimiento de las Tablet PC, debía resultar fácil:

1. Buscar una pequeña pantalla LCD de alta definición (HD)
2. Conectarla a una placa con conexión HDMI/LCD
3. Añadir una pantalla multi-táctil capacitiva sobre la LCD
4. Conectarla a tu ODROID u otro dispositivo HDMI

Lamentablemente, no es tan sencillo como parece o sino todos lo hubiéramos hecho ya. Los ordenadores ODROID

## La parte trasera de los componentes del ODROID-VU



**El ODROID VU – Una pantalla multitáctil 1280x800 de 9 pulgadas**

soportan resoluciones de 1920x1080 y puedes comprar un monitor a tamaño real con esa resolución, pero eso no es lo que queremos. Estamos buscando algo más pequeño y portátil, es ahí donde la cosa se vuelve más complicada.

Hay muy pocas pantallas HDMI pequeñas y portátiles, disponible en el mercado. La mayoría de las que hay tienen 800x480 o 800x600 píxeles y están destinadas en su mayor parte a fotógrafos y productores de video profesionales. Con un precio demasiado e-

levado, la mayoría no tienen opción de pantalla táctil. Así que decidimos desarrollar una pantalla HDMI universal de 9 pulgadas con una entrada multi-táctil capacitiva de 10 puntos para los usuarios ODROID. El nombre de pantalla es ODROID-VU, que se pronuncia como "view". Esperamos que este producto pueda dar una "cierta imagen" a tu equipo.

## Especificaciones

**Pantalla:** LCD TFT de 9 pulgadas con 1280 x 800 píxeles

**Touch screen:** Multi-táctil capacitiva 10 puntos de entrada (HID USB)

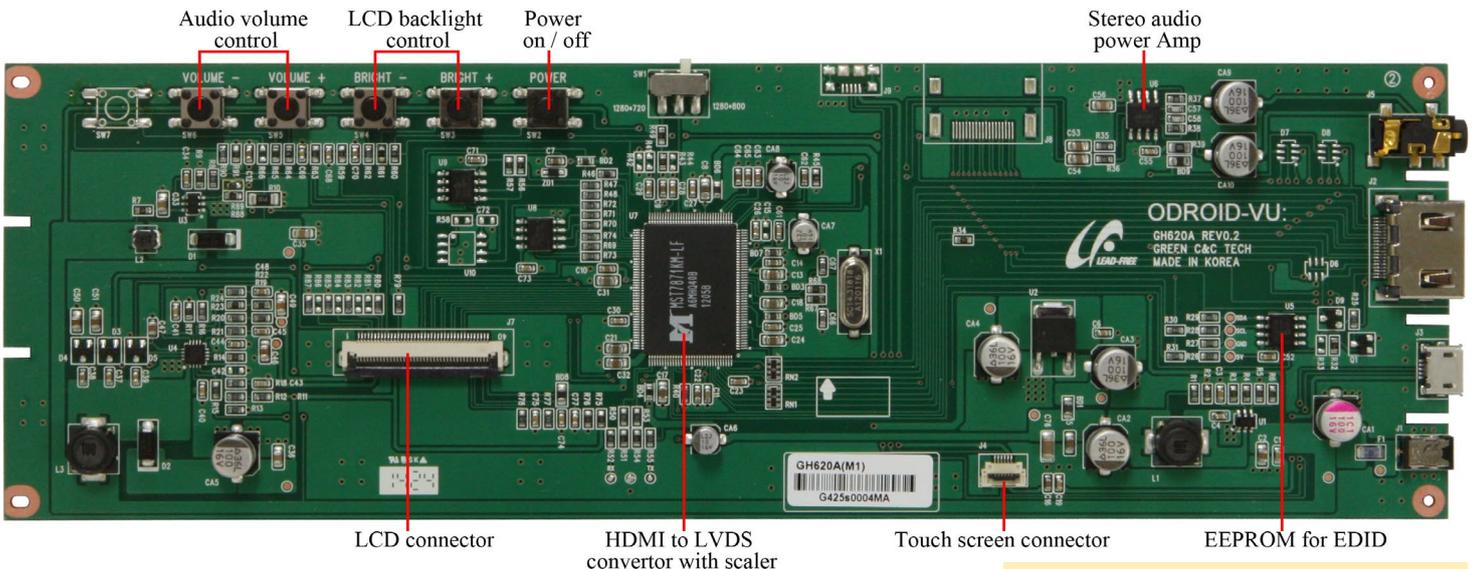
**Dimensiones:** 224 x 153 x 11 mm con el marco de plástico.

**Energía:** DC 5V/1A

**Puertos:** Energía, Micro-USB, HDMI tipo-A, clavija de audio (estéreo de 3,5 mm)

**5 Teclas:** Control de volumen, Control de brillo, Encendido/Apagado

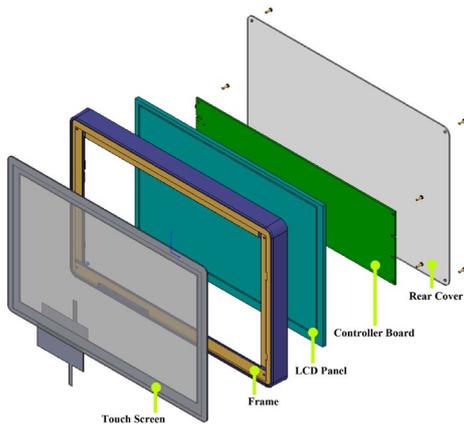
**El ODROID-VU está disponible en la tienda Hardkernel en [hardkernel.com](http://hardkernel.com)**



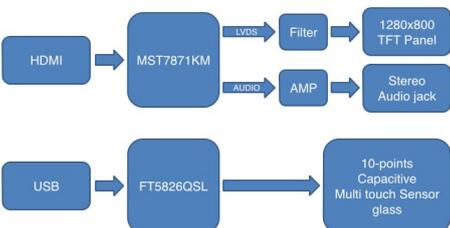
## Qué hay dentro

El IC convertidor HDMI a LVDS MST-7871KM tiene un receptor HDMI y un transmisor LVDS con salida de audio analógico. El IC también tiene un marco interno para mover la imagen hacia arriba o abajo para ajustarla a la resolución 1280x800. El control de volumen y de brillo también son gestionados por el IC con la interfaz de usuario.

El controlador FT5826QSL de panel táctil de único chip soporta un panel táctil capacitivo de hasta 10 puntos al



Diagramas de bloques del montaje de ODROID-VU



mismo tiempo. También incluye el protocolo estándar USB HID Multi-táctil, compatible con Linux y Windows.

El VID:PID USB es 2808:81c9 que es información muy importante si modificas el driver HID del kernel.

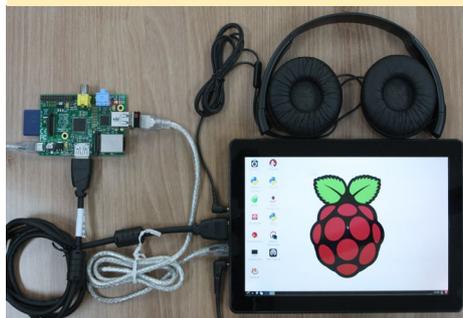
## Ejemplos de uso

Puedes conectar el ODROID-VU a placas ODROID, así como a un PC, televisión y otras plataformas que dispongan de salida HDMI.

Algunas placas ODROID no pueden generar una señal adecuada HDMI para la resolución nativa 1280x800 de ODROID-VU. De forma que 1280x720 (720p) es escalada hasta alcanzar 1280x800 y adaptarse a la pantalla completa. La resolución 1920x1080 también es compatible a través de la función scale-down.

Puedes comprar ODROIDVU con un cable micro-USB y una fuente de alimentación, en la tienda Hardkernel (<http://bit.ly/UmZEod>).

### Raspberry Pi y Debian con ODROID-VU



### Primer planod del ODROID-VU

### XBMC, Chromium y Terminal en U3 con Ubuntu sobre el ODROID-VU



### ODROID-U3 ejecutando Android 4.4 con un ODROID-VU como pantalla principal

### Portatil ejecutando Windows 8.1 con un segundo monitor utilizando ODROID-VU



# PEPPERFLASH PLUGIN DE CROME PARA LUBUNTU 14.04

UNA SENCILLA FORMA DE  
VISUALIZAR ADOBLE FLASH  
EN TU ODROID CON LINUX

por @Miltos

editado por Venkat Bommakanti



**E**l Adobe Flash Player para el navegador web Chrome en Ubuntu fue cancelado por Adobe hace varios años, sin contar con otra alternativa para reproducir vídeos Flash en Linux hasta hace muy poco. Google lanzó recientemente un plugin reproductor compatible con Flash llamado PepperFlash, que incluye una versión ARM para Ubuntu. En este artículo se describen los pasos para descargar este plugin de Google y añadirlo al navegador web de Chrome, ejecutandolo en Lubuntu 14.04 LTS y en cualquier placa ODROID, incluyendo las series X, U y XU.

## Requisitos

1. Una placa ODROID, con un adaptador de alimentación.
2. Una tarjeta MicroSD 8 + GB o módulo eMMC con la última imagen Lubuntu disponible desde el sitio web Hardkernel.
3. Una red donde el dispositivo tenga acceso a Internet y a los foros ODROID.
4. Acceso SSH opcional para ODROID mediante utilidades como PuTTY (MS Windows 7 +) o Terminal (Mac, Linux) para ejecutar los pasos desde un ordenador remoto.

5. PepperFlash Ver. 12.0.0.77 para Lubuntu 14.04
6. Administrador de archivos PC-Man (pcmanfm) para Lubuntu 14.04

## Descargar Pepper Flash

Entrar en ODROID utilizando el nombre de usuario por defecto de "odroid", y descarga la versión pre-compilada del PepperFlash compatible con ODROID desde <http://bit.ly/1yYEDQf> que se guardará en el directorio /home/odroid/Downloads

Inicia una sesión de terminal y ve al directorio de descargas. A continuación, inicia pcmanfm con privilegios de root escribiendo el siguiente comando en la ventana de terminal:

```
$ sudo pcmanfm
```

Haga clic derecho en el icono del archivo comprimido descargado, que debe ser visible en el extremo derecho del panel, y selecciona la opción de menú "Extract Here"

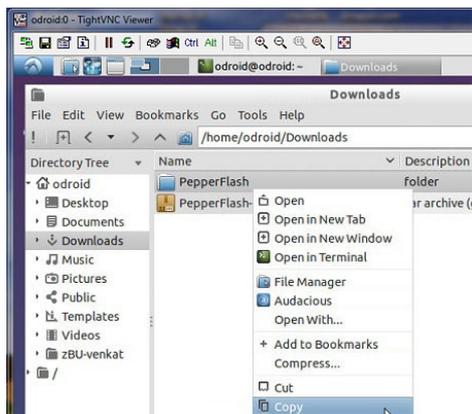
El plugin será extraído al directorio home/odroid/Downloads/PepperFlash, como se muestra en la captura de pantalla de abajo.

Para algunos usuarios es más fácil utilizar la ventana de terminal de línea de comandos casi en su totalidad para llevar a cabo los siguientes pasos. Sin embargo, el uso de pcmanfm simplifica el proceso para aquellos que quieran ahorrarse un poco de tiempo a la hora de teclear.

En pcmanfm, copiar el directorio PepperFlash y pegarlo en el directorio /

### Contenido del directorio después de extraer PepperFlash

Downloads				
File Edit View Bookmarks Go Tools Help				
! /home/odroid/Downloads				
Places	Name	Description	Size	Modified
Home Folder	PepperFlash	folder		03/21/14 18:40
Desktop	PepperFlash-12.0.0.77-armv7h.tar.gz	Tar archive (gzip-compressed)	8.1 MiB	06/19/14 12:20
Applications				



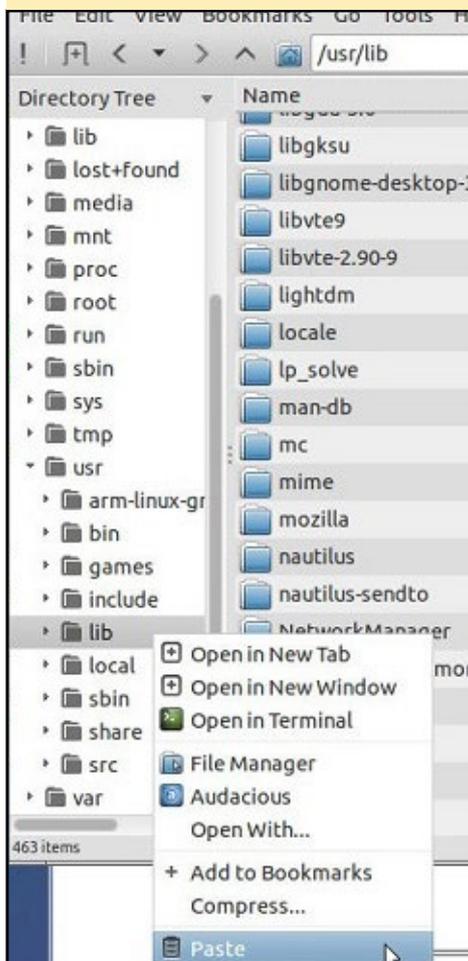
### Copiar el directorio PepperFlash

usr/lib.

Pega el directorio copiado en el directorio del sistema library en /usr/lib/ dando como resultado /usr/lib/PepperFlash. Si utilizas la línea de comandos, copia el directorio PepperFlash al directorio del sistema library escribiendo el siguiente comando:

```
$ sudo cp PepperFlash /usr/lib/
```

### Pegar el directorio PepperFlash



## Configurar plugin Pepper Flash

Tras cerrar pmanfm, inicia el editor medit (instalado por defecto en Lubuntu) y abre el archivo de configuración por defecto de PepperFlash con el comando:

```
$ sudo medit \
/etc/chromium-browser/default
```

Modifica el parámetro de configuración CHROMIUM\_FLAGS como una única línea:

```
CHROMIUM_FLAGS=" --ppapi-flash-
path=/usr/lib/PepperFlash/ \
libpepflashplayer.so --ppapi-flash-
version=11.7.700.225"
```

Guarda el archivo de configuración y cierra medit y Chrome.

## Validar la instalación

Verifica que el plugin PepperFlash se ha instalado correctamente iniciando de nuevo el navegador web Chrome y dirigiéndote a la URL chrome://plugins. Selecciona el plugin PepperFlash y haga clic en la opción de Detalles en la parte derecha de la página web.

Revisa los detalles asegurándote que la información del plugin corresponde con lo que aparece en pantalla. Por último, activa la opción "Always allowed" marcando la casilla de selección y cierra todas las ventanas de Chrome.

Para asegurarte de que el plugin se ha instalado correctamente, reinicia ODROID y luego inicia de nuevo Chrome. Ve a cualquier página con contenido Flash, como el sitio web de muestras de Adobe Flash en <http://adobe.ly/UmTzrV>

Para más información y plantear tus cuestiones, por favor visita las fuentes de información originales en <http://bit.ly/1lstuyP>

## MUPEN64PLUS CONVIERTE TU ODROID EN UNA CONSOLA DE JUEGOS RETO NINTENDO 64

por Rob Roy



**M**upen64Plus es un emulador de Nintendo 64 para la plataforma Android, ¡y funciona muy bien en ODROID! Para instalar Mupen64plus, visite la página de Mupen64Plus en Play Store <http://bit.ly/1mE82Ye>, conecte sus joysticks, y ¡Sumérgete en la gloria de la que gozo Nintendo 64!

### De arriba a abajo: F-Zero-X, Mario World 64 y Mario Tennis 64



# IO SHIELD DESMITIFICADO

## COMO CREAR UN INTERMEDIARIO ENTER EL HARDWARE Y EL HUMANO

por Bohdan Lechnowsky

**R**ecientemente he sido contratado para trabajar en un proyecto que necesitaba un U3 para controlar potencialmente válvulas, accionadores y otros dispositivos y leer las entradas. Fue emocionante, porque sabía que tendría la oportunidad de trabajar con el IO Shield para U3. Algo que siempre he querido hacer desde que estaba disponible, pero no he tenido la oportunidad de hacerlo debido a mi apretada agenda de trabajo.

Mi primer paso cuando trabajo con un nuevo hardware es ver si puedo hacerlo funcionar usando los ejemplos proporcionados. El segundo generalmente es enviar preguntas a forum.odroid.com relacionadas con problemas de configuración que destecto, hasta que logro hacerlo funcionar. El tercer y cuarto paso, en cambio, son con los que particularmente disfruto.

La dificultad del tercer paso pasa por reducir la funcionalidad del nuevo hardware a la forma más simple, elegante, flexible y eficiente que me sea posible al mismo tiempo que desecho cualquier idea preconcebida de cómo está actualmente diseñado el hardware para su acceso y control. Esto no implica ningún tipo de codificación, sino que más bien se trata del proceso de diseño de un dialecto que actúe como intermediario entre el usuario y el hardware, hasta que alcanzo todos los objetivos fijados.

El cuarto paso implica convertir ese dialecto en código utilizando la potente

función de análisis de Rebol. Sorprendentemente, una vez que la tercer paso esté completado, el cuarto es bastante simple. En la mayoría de los casos durante este cuarto paso, veo necesario afinar el diseño elaborado en el tercero. Por supuesto, el cuarto paso también implica realizar pruebas y depurar el diseño basandome en los resultados de las pruebas.

### Pasos 1 y 2

La primera vez que intenté usar IO Shield en Ubuntu, tuve dificultades en localizar instrucciones claras para lograr que funcionase y se comunicara con Rebol. Esto me llevó a los foros donde finalmente pude encontrar las instrucciones básicas necesarias:

Actualiza el kernel de Ubuntu siguiendo las instrucciones del foro: <http://bit.ly/1o3vMJx>

Escriba lo siguiente en Terminal:

```
odroid@odroid:/sys/class/gpio$
sudo su
root@odroid:/sys/class/gpio# mod-
probe gpio-pca953x
root@odroid:/sys/class/gpio#
modprobe i2c-gpio-custom
bus0=4,200,199
root@odroid:/sys/class/gpio# echo
tca6416 0x20 > /sys/devices/plat-
form/i2c-gpio.4/i2c-4/new_device
root@odroid:/sys/class/gpio# echo
304 > export ;304 is equivalent
```



**I/O shield para U3 es una potente herramienta en el kit mágico del mago ODROID**

```
to P17 on the IO Shield
root@odroid:/sys/class/gpio# cd
gpio304
root@odroid:/sys/class/gpio/
gpio304# echo out > direction
root@odroid:/sys/class/gpio/
gpio304# echo 1 > value ;Turns on
the pin
root@odroid:/sys/class/gpio/
gpio304# echo 0 > value ;Turns
off the pin
```

Esto es lo que necesitaba para empezar a desarrollar mi dialecto IO-Shield, que decidí llamar gpio.

### Paso 3

Después de mucho pensar y diseñar, te muestro algunos ejemplos de lo que es capaz de hacer mi dialecto:

```
gpio [
init out P17 ;initialize pin
P17 as an output
on ;turn on P17
wait .1 ;wait .1 seconds
off ;turn off P17
deinit P17 ;deinitialize pin
P17
]
```

```

gpio [
    init in [P00 P01]    ;Set
pins P00 and P01 as inputs
    init out [P02 P03 P06];Set
pins P02, P03 and P06 as outputs
    init pwm 1 .1 P04    ;Set pin
P04 as a pulse-width-modulated
output with a
                ; period of 1
second and a pulse of .1 seconds
    [P02 P06] on        ;Turn
pins P02 and P06 on
    init .1 .01 P05     ;Set pin
P05 as a pwm output - the last
initialization
                ; (P04) was a
pwm, so it was remembered for
this init.
    init gpio in P07    ;To
change back to standard I/O,
specify gpio.
    wait .01            ;Wait .01
seconds
    P06 off             ;Turn pin P06
off
    P03 on              ;Turn pin
P03 on
    read speed P00      ;Read the
value of pin P00 into system/
gpio/val/speed
    read rpm P01        ;Read the
value of pin P01 into system/
gpio/val/rpm
    wait 5              ;Wait 5
seconds
    deinit [P04 P05]    ];Deini-
tialize pins P04 & P05 (this is
the way to turn off
; a PWM pin)
    wait 1              ;wait 1
second
    reset               ;Deinitial-
ize all standard I/O and PWM pins
]

```

## Paso 4

Para llegar a este nivel de abstracción, el dialecto tuvo que superar los siguientes obstáculos:

- Siempre que gpio es activado, determina si el subsistema IO Shield ya había sido iniciado. Si no es así, hay que iniciarlo.
- Convertir los sistemas de tiempo dispares en un sistema de tiempo estandarizado (PWM utiliza microsegundos, wait utiliza segundos). La mayoría de la gente utiliza los segundos por más comodidad, así que se decidió normalizar en segundos.
- Proporcionar un nivel de abstracción entre los números de puerto impresos sobre el IO Shield y los números de puerto / sys/class/gpio.
- Permitir gestionar PWM y I/O estándar dentro del mismo dialecto sin necesidad de entender el mecanismo básico.
- Reducir el exceso de palabras necesarias del dialecto permite puertos múltiples o individuales para que funcionen de forma simultánea, y las palabras innecesarias no sean requeridas. Aunque permite que el usuario las utilice para recordar los últimos puertos y tipos de inicialización, así que los siguientes comandos no necesitan volver a especificarse innecesariamente.
- Las Pruebas tuvieron que llevarse a cabo con diferentes métodos de lectura/escritura a/ desde el /sys/class/gpio y escribiendo en los directorios sys/class/soft\_pwm para ver su fiabilidad y rendimiento en diferentes condiciones.

Afortunadamente, incorporar este tipo de funcionalidad dentro de la sintaxis de Rebol es bastante simple una vez que te familiarizas con él.

Casualmente, el análisis sintáctico utiliza un dialecto que te permite definir fácilmente las reglas para tu dialecto. Por eso, si conoces una forma más fácil de escribir un dialecto, los desarrolladores

de código abierto de Rebol estarán encantados de escucharte.

## Pruebas

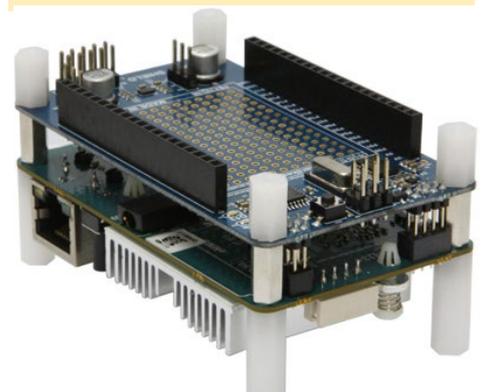
Durante las pruebas en el U3 he notado que utilizando PWM, a veces aparece una cierta inestabilidad en el sistema operativo. Pienso que esto podría tener algo que ver con las constantes interrupciones necesarias para hacer que el PWM funcione a una alta velocidad en un sistema de software. También me di cuenta que a veces incluso a velocidades bajas el PWM era inexacto.

También descubrí durante las pruebas que usando el comando write de Rebol a veces no funcionaba, pero otras veces iba muy bien. Todavía no he sido capaz de determinar las circunstancias que llevan a este comportamiento indeseable. Rebol también tiene un comando echo que funciona incluso cuando write no lo hace. El método más fiable (pero con mucha diferencia, el más lento) fue iniciar un comando echo bash desde dentro de Rebol. En mis pruebas, este método es 25 veces más lento que la escritura nativa y los comandos echo en Rebol.

## Resultados

Puedes descargar el dialecto para tu propio uso en <http://respectech.com/odroid/gpio.r3>, donde se pueden dejar comentarios. El script del dialecto gpio.r3 es editable, así que eres libre de hacer cambios y mejoras.

¡El I/O Shield está ahora bajo tu control!



# PROFUNDIZANDO (EN) EL ODROID-SHOW

## DESCUBRIENDO TODO EL POTENCIAL DEL HARDWARE ARDUINO

por Declan Malone

En números anteriores de ODROID Magazine se han tratado conceptos básicos sobre el funcionamiento del hardware ODROID-SHOW y cómo usarlo para visualizar diversos datos desde tu ODROID o un PC. Aunque tiene muchas y útiles aplicaciones, sólo se ha tratado superficialmente lo que verdaderamente se puede llegar a hacer con SHOW. La parte más importante del dispositivo es un chip ATmega328P, que es el mismo tipo de microcontrolador que se encuentra en muchos modelos de la popular plataforma Arduino. Esto significa que cualquier cosa que hagas utilizando Arduino, podrás fácilmente hacerla funcionar en ODROID-SHOW.

Con esta serie de artículos que inicio pretendo, por decirlo de algún modo explorar bajo la superficie, introduciéndote en el uso de ODROID-SHOW como plataforma de programación. En esta entrega, cubriré los componentes de software que vienen preinstalados en el ODROID-SHOW, cómo funcionan y cómo modificar y cargar tus cambios en el dispositivo. Utilizaré un ejemplo sencillo de un visor de Mandelbrot Set para demostrártelo.

En el segundo artículo, haré frente a ciertas opciones para conectar el SHOW a algunos componentes electrónicos simples a través de dos cabezales GPIO. Por último, en el tercer artículo me centraré en el desarrollo de un juego completo e independiente sobre el propio hardware.



El ODROID-SHOW puede mucho más que mostrar simple texto

### Software oficial

La última versión del software de ejemplo de ODROID-SHOW se puede descargar desde github:

```
$ mkdir ~/src; cd ~/src
$ git clone https://github.com/
hardkernel/ODROID-SHOW
$ cd ODROID-SHOW
```

Encontrarás varios directorios, “example/” con los programas que se ejecutan en ODROID o un PC, “show\_main/” con el programa principal para ejecutarse sobre el SHOW y “libraries/”, donde se almacena diversas librerías compatibles con Arduino. Dependiendo de cuando descargaste las fuentes de github, veras otros directorios como uno para el módulo de tiempo, que no voy a tratar en estos artículos.

Es posible que ya hayas visto los programas del directorio “example/”. Sino, ahora podría ser un buen momento para echar un vistazo, para que te familiarices con algunas de las posibilidades que ofrece ODROID-SHOW. También puedes ojear los artículos anteriores para obtener más información.

En este artículo voy analizar los contenidos de los directorios “show\_main/” y “libraries/”, que están explicados en el diagrama por bloques que se muestra en este artículo más adelante.

### Partes explicadas

Los Terminales (una especie de combinación de teclado y pantalla) existen desde hace más de un siglo, incluso antes de que se desarrollaran los primeros ordenadores. A medida que los ordenadores se volvían más comunes, muchos de los viejos dispositivos terminales se empezaron a utilizar como meros dispositivos de entrada y visualización de datos. Al principio, en lugar de ordenadores personales, era mucho más común tener un equipo grande y potente (como un macro ordenador) conectado a muchos terminales diferentes.

Debido a la amplia variedad de diferentes tipos de dispositivos terminales, así como el gran número de funciones que los más nuevos y sofisticados terminales ofrecían (como texto en color, un timbre sonoro, la posibilidad de colocar el cursor o limpiar la pantalla, etc), se hacía necesario algún tipo de estan-

darización. La lista de “códigos de escape ANSI” (como “ESC [2]” para limpiar la pantalla) fue el resultado.

Aunque esto puede sonar arcaico (al fin al cabo, actualmente todos disponemos de PCs, portátiles y tabletas, no terminales), todavía son muy comunes hoy en día. Si utilizas “xterm” o un equivalente más moderno (como “gnome-terminal”) puedes seguir utilizando los códigos de escape ANSI para todo lo que se solía hacer en los terminales dedicados.

Esto es exactamente lo que está implementando en el programa “show\_main” mediante programación Arduino, para comportarse como si fuese una pantalla de terminal.

Puede revisar la lista completa de los “códigos de escape” que el software implementa en <http://bit.ly/1nkYKU> Volveré a esto más adelante cuando añada un código ANSI para dibujar un simple píxel en la pantalla.

El otro cajón de la izquierda es para la conexión en serie al PC (o ODROID). Se utiliza para la recibir datos desde el PC y también para implementar información de depuración, que puede ser muy útil cuando escribimos programas de Arduino/SHOW.

En la parte derecha, tenemos dos librerías que también fueron desarrolladas por Adafruit. La de arriba implementa elementos gráficos de alto nivel, como la impresión de un carácter, trazado de puntos, líneas y círculos, y rellenar áreas rectangulares con colores sólidos. Esto puede verse como una parte independiente del resto.

Por debajo, tenemos la parte específica de la plataforma con el código de los gráficos. Se sabe cómo está conectado por cable el chip ATmega al módulo TFT, así como los protocolos necesarios para comunicarse con el hardware actual.

No vamos a profundizar demasiado en el funcionamiento de este lateral, aunque puedes analizar el código si tienes curiosidad.

## Programación básica en Arduino

Todos los componentes incluidos en el software oficial para el dispositivo SHOW están escritos en C++. Si ya conoces C++ (o incluso C), no deberías tener ningún problema para programar en esta plataforma. Hay algunas pequeñas diferencias con respecto a la forma en la que es usado C++ en Arduino, que debes tener presente. Por ejemplo:

- En lugar de utilizar los comandos de bajo nivel como “gcc” y “make”, toda la edición, compilación y carga de programas se hace en un IDE gráfico (Entorno de Desarrollo Integrado) llamado apropiadamente, Arduino.
- Los programas escritos para Arduino son a menudo conocidos como “sketches” y generalmente tienen la extensión “.ino”
- Los programas no tienen una función “main ()”, ya que ésta es proporcionada por el sistema Arduino
- Toda la inicialización se hace con la función “setup ()” del programa.
- Los Programas de Arduino también requieren una función “loop ()”. Como su nombre indica, esta función es activada en un bucle infinito directamente después de que devuelva la función “setup ()”.

Estas son todas las diferencias relativamente menores a la hora trabajar en un ordenador “real”, por lo que en realidad es bastante fácil adentrarse en el desarrollo de programas para la plataforma Arduino (y SHOW)

## Entorno Desarrollo

Para nuestra primera fase con la programación en SHOW, tendremos que

compilar y volver a instalar el software oficial. Vamos a empezar instalando el IDE Arduino. En los sistemas Debian/Ubuntu esto se hace escribiendo:

```
$ sudo apt-get install arduino
```

Esto instalará muchos otros programas además del IDE, incluyendo un compilador (“avr-gcc”) y algunas herramientas auxiliares (“avrdude” es la principal). Por lo general, no tienes que preocuparte por todas estas herramientas, puesto que el IDE Arduino la activará de forma transparente cuando las necesite. Más adelante, si deseas usar directamente estas herramientas (con tu editor favorito y Makefiles), podrás hacerlo. En nuestro caso nos quedaremos sólo con el IDE.

Hay otros cuatro pasos de configuración importante:

1. Decirle al IDE donde almacenarás tus “sketches” (no dónde has descargado las fuentes de github)
2. Decirle qué puerto serie usar para comunicarse con ODROID SHOW (esto puede cambiar si reinicias o mueves los dispositivos USB)
3. Decirle que versión de Arduino se está usando
4. Decirle dónde buscar tus librerías

Todos los pasos excepto el paso 3 están tratados en la wiki oficial de ODROID en <http://bit.ly/1p8uitU>, así que no los voy a repetir. Para el paso 3, selecciona “Tools >Board >Arduino Uno”. El ODROID SHOW es totalmente compatible con la configuración Arduino Uno.

Si estás ejecutando una versión anterior del programa de Arduino (mi distribución Debian Wheezy sólo permite la versión 1.0.1), en el paso 4 no aparecerá la opción “Add Library ...” dentro de las opciones de menú “Sketch >Im-

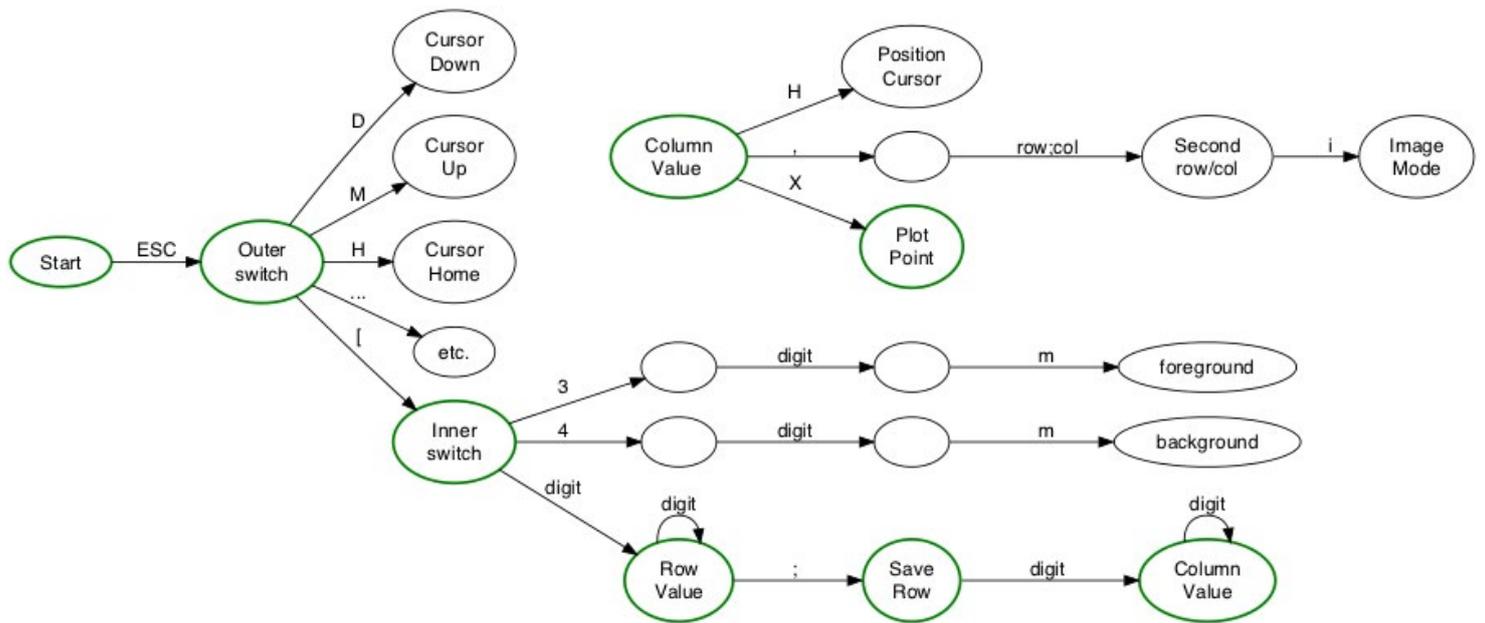


Diagrama de Finite State Machine para el analizador ANSI

port Library”. Si tu versión de Arduino es tan antigua como la mía, entonces tendrás que añadir las librerías “GFX” y “ILI9340” manualmente creando enlaces simbólicos desde el directorio de “librerías” de tu directorio de sketch (configurado en el paso 1) a los directorios de librerías individuales que figuran en el código que ha descargado desde Github. Por ejemplo, si lo has descargado en el directorio “~/src/ODROID-SHOW”, puede crear los enlaces con:

```
$ cd ~/sketchbook

$ mkdir libraries; cd libraries
$ for d in \
~/src/ODROID-SHOW/libraries/*;
do ln -s $d .; \

$ done
```

Finalizados estos pasos de configuración, podemos cargar “show\_main.ino” en el IDE. Después, hay que hacer una cosa más antes de reprogramar el dispositivo, y es añadir el conector de programación. Esto se utiliza para anular un grupo de dos pines identificados como P2 en la placa. Están ubicados entre el interruptor de reinicio (marcado como ‘Reset’) y el LED de alimentación (marcado como ‘Alive’). Una vez que hayas hecho esto, todo lo que tienes que hacer

es hacer clic en la señal verde “right arrow” en la fila de iconos justamente debajo de la barra de menú.

Si todo va bien, debería ver algún tipo de indicación en el panel de mensajes en la parte inferior, y el SHOW se reiniciará a continuación. Tras el reinicio, tendrás que eliminar el conector antes de ejecutar cualquier prueba, como las del directorio “examples /”. Los conectores tienden a ser pequeños y enrevesados, así que procura recordar dónde lo pones después de quitarlos. Alternativamente, puedes déjalo unido a uno de los pines de la cabecera P2. También te puedes encontrar con diferentes tipos de conectores que tienen una pestaña adicional en la parte superior. Ellos tienden a ser mucho más fácil de poner y quitar y son más difíciles que perder porque son más grandes. No necesitas tener uno de estos, pero te pueden ayudar.

Si las cosas no van bien, comprueba el panel de mensajes para hacerte una idea de lo que ha salido mal. Obtendrás diferentes mensajes si la compilación ha fallado (por ejemplo, debido a la falta de librerías o por errores de sintaxis) o surgió algún problema con la carga (olvidaste volver a fijar el conector). Es bastante fácil y seguramente consigas averiguar que salió mal basándote en estos mensajes.

## Añadir un comando “plot”

La primera vez que utilicé ODROID-SHOW (v1.1 del software), permitía muchos caracteres escape ANSI para dibujar texto en la pantalla, pero no incluía ningún otro elemento gráfico. Decidí cambiar eso añadiendo una nueva secuencia de escape que permitiera trazar un solo punto en la pantalla. Así es como pude añadir esta función.

Lo primero que hice fue comprobar si había un código de escape ANSI existente para trazar puntos. Resultó que no, así que elegí esto como sintaxis para el nuevo comando:

```
ESC [ row ; col X
```

Examinado el código de “Show\_main.ino” se puede ver que la función “parsechar ()” es activada cada vez que un nuevo carácter se lee desde el PC a través del puerto serie. El programa analizador utiliza el denominado “Finite State Machine” (o “FSM”). Esto se utiliza con bastante frecuencia en cualquier aplicación donde se tiene que desarrollar complejos comandos de un carácter.

Un analizador basado en FSM es simplemente una lista de “estados” (que busca los comandos parcialmente recon-

ocidos) y “transiciones” (que son seguidas para ver un nuevo carácter desde la entrada). Véase el diagrama de la página anterior, que muestra algunos de los FSM para el analizador ANSI.

Esto no es una descripción 100% completa, pero debería ayudar a explicar cómo funciona el analizador en general. He resaltado en verde la ruta que coge el analizador para reconocer mi nuevo comando de trazado de puntos. Como puede ver, el código es simplemente “insertado” en la estructura gráfica existente. Una vez que supe donde iba el nuevo código, el código para implementar el nuevo comando sería muy simple:

```
case 'X':

row = (row > right_edge0) ?
right_edge0 : row;

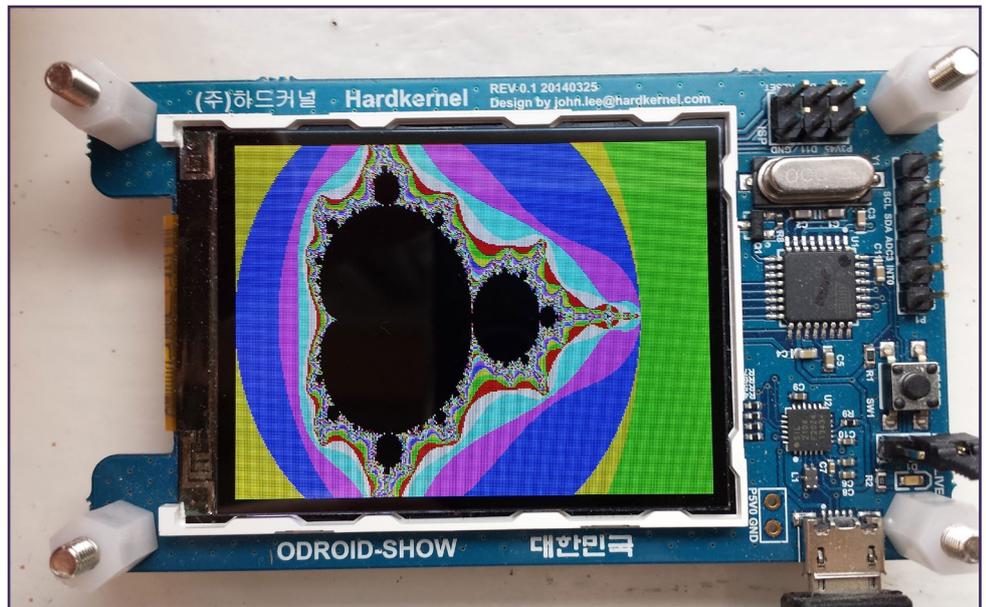
col = (tmpnum > bottom_edge0) ?
bottom_edge0 : tmpnum;

tft.drawPixel(row, col,
foregroundColor);

break;
```

Esto comprueba los valores de fila y columna facilitados. A continuación, activa el método apropiado desde el objeto GFX (“tft.drawPixel()”) utilizando el color de primer plano actual.

Puesto que estos cambios fueron in-



ODROID-SHOW mostrando un Mandelbrot set

corporados en una versión posterior del software oficial, puedes comprobar por ti mismo donde se ha añadido el código.

Si puedes entender cómo se organiza el analizador, deberías poder añadir más funciones siguiendo el mismo método que el anterior. Podrías, por ejemplo, agregar más secuencias de escape para dibujar círculos, líneas o recuadros, los cuales están disponibles desde la librería GFX.

Sería muy sencillo implementar un conjunto de formas al estilo LOGO, por ejemplo, con comandos para mover hacia adelante o hacia atrás, girando la “tortuga”, poniendo el lapicero hacia arriba o abajo, etc.

## Mandelbrot Set

Un primer paso común cuando analizamos un nuevo hardware gráfico es visualizar en éste el Mandelbrot set. En el improbable caso de que nunca hayas oído hablar de esto antes, la página de Wikipedia <http://bit.ly/1mQ75y3> es un buen lugar para empezar. Esta página también incluye pseudocódigo para generar el Set, así que no necesitaré tratarlo en este artículo.

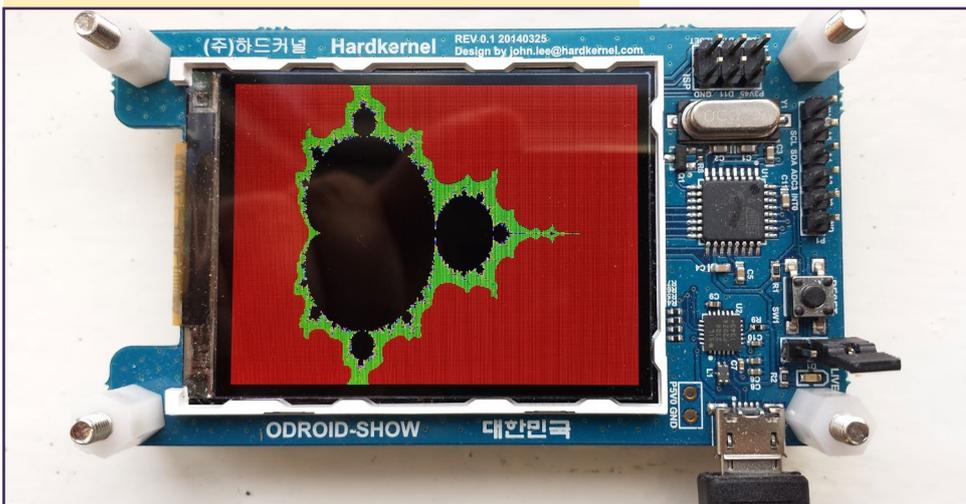
Puede descargar el código de un teneador que hice para el repositorio SHOW oficial de Hardkernel con los comandos:

```
$ cd ~/src

$ git clone \
https://github.com/declanmalone/
ODROID-SHOW.git SHOW-fork

$ cd SHOW-fork
```

ODROID-SHOW mostrando un Mandelbrot set



En realidad, hay dos versiones del código. La primera, en el directorio “examples/” denominada “mandelbrot.c”, está pensada para ejecutarse en el host (PC). Se incluye un “Makefile” para compilarlo automáticamente en una máquina ODROID u otro sistema Linux. Hay algunas cosas que tienes que tener en cuenta sobre este código:

1. Incluye algo de código basado en el programa oficial “port\_open.c”. Este código abre el puerto serie USB y fija la tasa de baudios y otros parámetros importantes. Esto significa que el código no necesita tener una ejecución “port\_open” independiente.
2. La función “plot\_point ()” envía la cadena de caracteres “ESC [ x ; y X” adecuada sobre el puerto serie. También envía una cadena “ESC 3 ? m” independiente para ajustar el color del punto.
3. Esa función también incluye la verificación de errores para asegurarse de que realmente la cadena completa ha sido escrita en el dispositivo. Si hay desbordamientos de buffer, entonces “write ()” fallará y fijará el error “EAGAIN”.  
El código trata bien esta cuestión haciendo avanzar el búfer de escritura y procesando el “write ()” de nuevo hasta que la cadena sea enviada por completo.
4. Hay un llamada a “usleep ()” tras cada escritura. Esto es necesario porque el ATmega es sólo un equipo de un solo sub-proceso y si el PC envía datos demasiado rápido, es posible que algunos se pierdan. Añadir un breve retraso, ajustable por el usuario después de cada escritura es una forma sencilla de trabajar con esto.

Aunque escribí esta versión del programa para utilizar el código de escape “X”, también sería posible enviar los datos mandando primero una secuencia de escape “ESC [x1; y1, x2; y2 i” y luego enviar los datos de color para cada píxel

como si fueras a enviar una imagen.

Por favor, consulta el programa de ejemplo “images.sh” de las fuentes oficiales para ver cómo se puede lograr esto.

## Generador nativo

También escribí una versión del programa Mandelbrot set como un sketch diseñado para funcionar de forma independiente sobre el ODROIDS-HOW, sin necesidad de un PC. El sketch se puede encontrar en el directorio “duino\_fractal /”. La compilación y la instalación se hace del mismo modo que se realizo para el software emulador de terminal oficial describió anteriormente.

El código nativo es muy similar a la versión PC, con algunas diferencias:

- Está escrito en C ++ al estilo Arduino, con una función “setup ()” y (un vacía) función “loop ()”
- Está hecho para usar directamente las librerías GFX y ILI9340
- Se devuelven mensajes de depuración por la conexión serie con llamadas “Serial.print ()”
- También utiliza la función de tiempo interna de Arduino “get\_millis ()” para medir el tiempo que necesita para presentar la pantalla.
- Escribe en la memoria de la pantalla TFT directamente en lugar de activar “drawPixel ()”. Esto significa que activa “setAddrWindow ()” una vez para cada

“panel”, a continuación, escribe cada dato de color del píxel con dos llamadas a “spiwrite ()”.

Se necesita el chip ATmega algo más de dos minutos para hacer la imagen completa con un valor máximo de repetición de 1000. Teniendo en cuenta que el chip no tiene ninguna capacidad de punto flotante y sólo funciona a 16 MHz, esto no es tan malo. El código nativo también tiene una ventaja sobre el código de PC, ya que no recibe ningún dato sobre la conexión serie, lo que no provoca retrasos tras escribir cada píxel.

Cuando escribí el código, tenía inicialmente la intención de usar un potenciómetro y dos botones pulsadores para permitir al usuario ampliar y reducir una área de la pantalla (de forma similar al vídeo en [http:// bit.ly/1yZ0UgB](http://bit.ly/1yZ0UgB)), pero nunca llegue a terminar esa parte. De modo que, sólo una parte del código soporta esto, como la definición de los pines de hardware y la posibilidad de pintar “paneles” individuales, para un mejor desplazamiento de la pantalla.

## Resumen

Hemos abarcado mucho terreno en este artículo. Espero que haya servido como una buena introducción a los conceptos básicos de programación de ODROID-SHOW y haya ayudado a explicar algunas de las cosas de la que es capaz de hacer la plataforma. Estarás de acuerdo que en realidad no es tan difícil escribir código. En el próximo artículo, conectaremos el SHOW a otros compo-



# SO DESTACADO: POCKET ROCKET Y COUCH POTATO

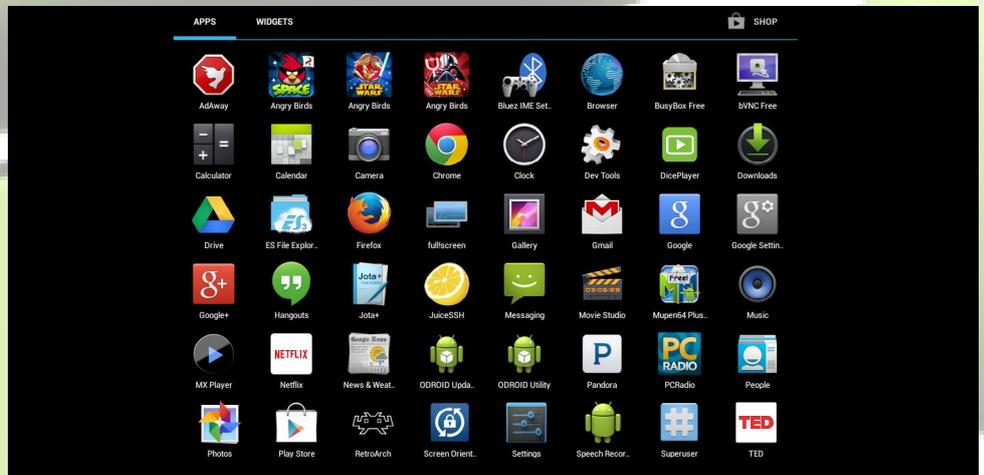
IMAGENES PRECOMPILADAS ANDROID 4.X  
PARA EL MEJOR DECODIFICADOR  
AHORA DISPONIBLE PARA KITKAT

por Rob Roy, Editor jefe

La familia de ordenadores ODROID fue inicialmente comercializada como un sistema Android abierto, ofreciendo a los programadores de Android una alternativa a la compra de un teléfono Android y hacerlo root. ODROIDs ejecuta Android extremadamente bien, al tiempo que ofrece muchas características integradas tales como la decodificación de vídeo por hardware, drivers para joystick, los cuales tendrían que ser instalados por separado en un sistema basado en Linux X11.

Cuando empecé a utilizar ODROID, configuré un decodificador usando un entorno de escritorio de Ubuntu llamado LXDE, el cual me permitía hacer streaming y descargar archivos multimedia, para luego reproducirlos con Xine o XBMC. Podía reproducir correctamente vídeos a 720p, pero con los videos a 1080p se perdían muchos fotogramas y la reproducción se entrecortada. Después de probar sin éxito varios ajustes para lograr una mayor fluidez en la reproducción, incluyendo overclocking y adaptar las opciones del kernel, decidí formatear mi tarjeta SD e instalar Android 4.1.2 Jelly Bean en mi U2.

La reproducción con vídeos a 1080p en Jelly Bean fue increíble, y era exactamente lo que estaba buscando. Utilizando tanto XBMC como MX player, ODROID podía reproducir sin problemas incluso los videos más densos de la



Lista de apps pre-instaladas en Pocket Rocket y Couch Potato

página web de XBMC (como Big Buck Bunny), utilizado en otros sistemas para detectar problemas. La decodificación de video en todos los desarrollos Android (X, U, y XU) es perfecta cuando usamos MX Player, pudiendo reproducir vídeos de cualquier resolución y codificación.

Por aquel entonces, también había comprado un Apple TV, que cuenta con una interfaz muy amigable, pero no permitía instalar nuevas aplicaciones, ya que la tienda de iTunes no está disponible para el Apple TV. Sin embargo, un sistema ODROID con Android te permite instalar y eliminar aplicaciones al igual que un teléfono Android, miles de aplicaciones y juegos disponibles en Google Play Store. Android ya tiene acceso a root, por lo que el sistema operativo puede ser manipulado, se pueden instalar aplicaciones personalizadas y se pueden

ejecutar funciones de administrador con el usuario root.

Antes de comprar mi primer ODROID, nunca antes había considerado usar Android como sistema operativo principal, pero después de probar la imagen Pocket Rocket, me di cuenta de todos sus ventajas. Android es un sistema operativo muy estable, incluso cuando una aplicación falla o se ralentiza, el sistema operativo facilita el cierre de la aplicación. Android es fácil de aprender, cuenta con iconos grandes y amigables, así como botones de acceso directo que son intencionadamente similares a los otros sistemas operativos.

Para navegar por Android de forma eficiente utilizo los atajos de teclado que incluye, que son similares a los de otros sistemas operativos como Windows, Ubuntu y OSX.

Pocket Rocket y Couch Potato para las series U, X y XU pueden descargarse desde <http://bit.ly/lzYsZFt>

## Atajos de Teclado

### Ctrl-C

Copiar

### Ctrl-V

Pegar

### Ctrl-A

Seleccionar todo

### Ctrl-W

Cerrar ventana

### Ctrl-Right Arrow

Mover a derecha una palabra

### Ctrl-Left Arrow

Mover a izquierda una palabra

### Ctrl-Shift-Right Arrow

Elegir palabra a la derecha

### Ctrl-Shift-Left Arrow

Elegir palabra a la izquierda

### Alt-Tab

Cambiar de programa

### Alt-Esc

Escritorio

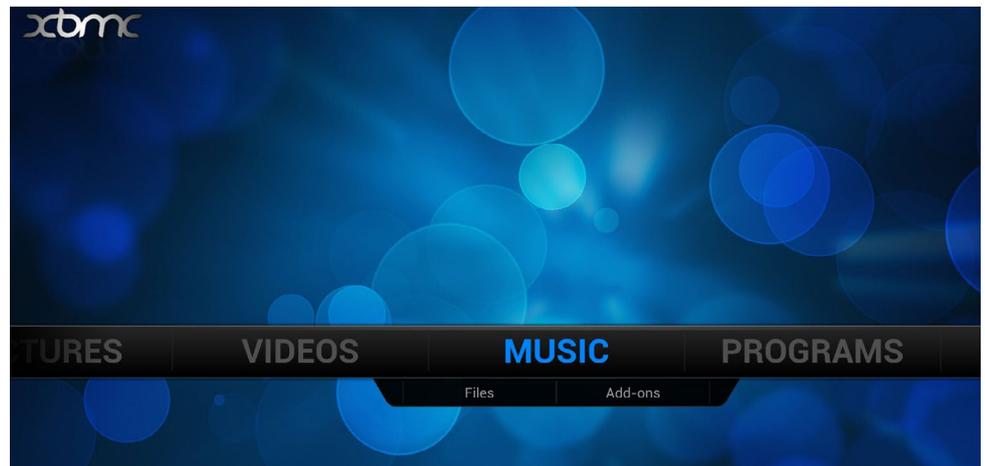
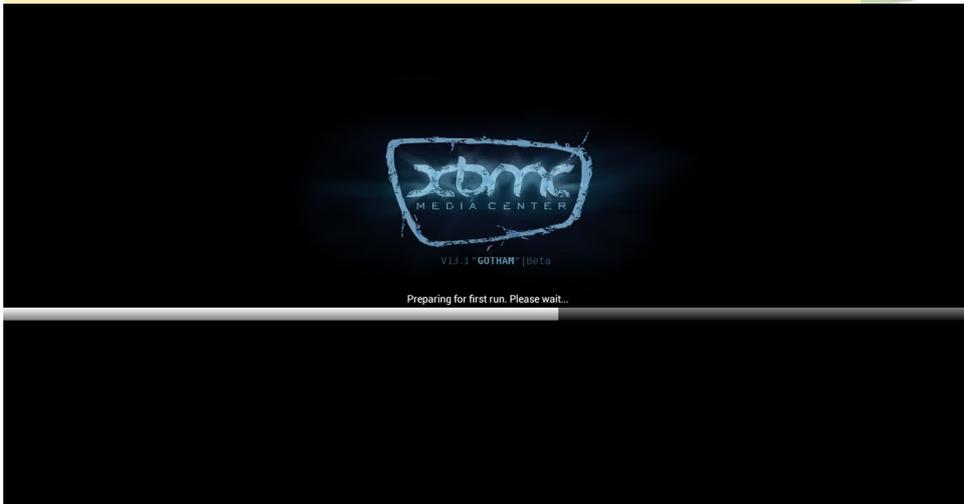
### Control-L

Ir a la barra de direcciones

## Experiencia Android

Android es especialmente adecuado para principiantes, o para cualquiera que tenga experiencia con teléfonos móviles. Incluso para un usuario experto que a veces sólo quiere realizar una simple tarea sin tener que ajustar la configuración, navegar por los menús o mover ventanas. Android es perfecto para esto y básicamente, permite ejecutar cualquier tarea

**XBMC Gotham 13 arranca por primera vez. Esta es la versión final de XBMC, ya que para la versión 14 pasará a llamarse Kodi.**



**XBMC es un programa que cubre todas tus necesidades como Centro Multimedia. Soporta vídeo, música, complementos, e incluso puede lanzar otras aplicaciones.**

que un PC de escritorio puede realizar, pero de un modo distinto.

A modo de ejemplo, la mayoría de los PCs de escritorio tienen un procesador de texto y guarda sus archivos en el disco duro local. En Android, el software utilizado para la edición de documentos se llama Google Drive, que almacena su información en la nube. Con Google Drive puedes coger cualquier dispositivo móvil, ordenador portátil, tablet, o PC de escritorio, iniciar sesión en Google Drive y tener acceso a todos sus documentos sin necesidad de tener que llevar una unidad USB.

Una notable diferencia con Android es que el botón derecho del ratón se utiliza en realidad como botón "Atrás". Para abrir los menús, normalmente lo hacemos con un clic derecho. En su lugar utiliza un "clic largo" presionando el

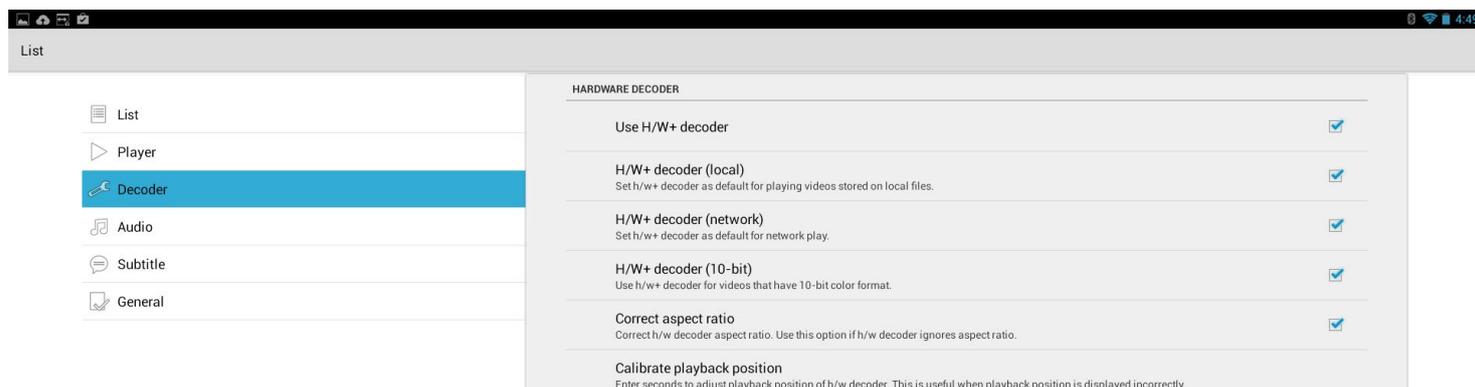
botón izquierdo del ratón durante varios segundos. Por ejemplo, en ES File Explorer para seleccionar varias carpetas, haz un "clic largo" en la primera carpeta, se abrirá una casilla de selección que permitirá seleccionar otras carpetas con un sólo clic.

## eMMC vs tarjeta SD

Android está diseñado para ejecutarse desde un módulo de eMMC, y una clara ventaja de los ordenadores ODROID es que este módulo puede ser retirado y reemplazado. Aunque una tarjeta SD técnicamente permite ejecutar Android, su lenta velocidad limita su funcionamiento debido al tráfico E/S. Dejé de publicar imágenes Rocket Pocket para tarjetas SD, debido a la cantidad de tiempo que dedicaba a hacer clic en los pop-up "No responde" de Android, mientras que personalizaba la imagen.

Sin embargo, descubrí que la versión Ice Cream Sandwich (ICS) de Android se ejecuta mejor que Jelly Bean cuando las comparé usando una tarjeta SD. En consecuencia, he creado una versión especial para tarjetas SD de Pocket Rocket llamada Couch Potato que cuenta con las mismas aplicaciones que Pocket Rocket, pero utiliza ICS para un mejor rendimiento.

Si no tienes un módulo eMMC, puedes instalar Couch Potato y disfrutar de las mismas características que Pocket Rocket. Es posible que, de forma pun-



tual Couch Potato te interrumpa con un pop-up “No responde”, como cuando intentas descargar múltiples aplicaciones desde Play Store, pero son menos frecuentes que cuando ejecutaba Jelly Bean desde una tarjeta SD. Ambas imágenes ofrecen exactamente las mismas aplicaciones y sólo se diferencian por la versión de Android 4.0 vs 4.1.

## Google Play Store

Al Google Play Store se accede presionando el icono de Apps creado con 6 pequeños cuadrados en la parte superior derecha del escritorio de Android. Luego, pulsa el botón Play Store en la esquina superior derecha del menú Aplicaciones. Tras entrar en la Play Store con una cuenta de Google o Gmail, aparecerá una amplia biblioteca de aplicaciones y juegos, muchos de los cuales ofrecen versiones gratuitas. Así es como también se actualizan las aplicaciones. Las notificaciones aparecerán periódicamente solicitando permiso para ser actualizadas.

Si ya tienes un teléfono con Android, puedes instalar rápidamente tus aplicaciones favoritas seleccionando la sección “Mis aplicaciones” de la esquina superior izquierda de la Play Store, la cual mantiene un registro de todas las aplicaciones que has usado en cualquier otro dispositivo Android.

## Chromium y Firefox

Sólo hay unos pocos navegadores disponibles para la plataforma Android, siendo Firefox y Chromium los más populares. Chromium tiene varias ca-

racterísticas no disponibles en Firefox, como deslizamiento táctil y atajos de teclado. Cualquier navegador puede sincronizarse con otros dispositivos para descargar automáticamente los marcadores favoritos que se crearon en la misma cuenta de Google. Los videos de Youtube se deben reproducir con la aplicación nativa de Youtube, para poder usar la decodificación de vídeo por hardware en ODROID.

## Gmail y Google Drive

Se puede añadir más de una cuenta de Google en la aplicación de configuración, tanto Gmail como Google Drive ofrece un desplegable de cuentas para cambiar fácilmente entre múltiples cuentas. Google Drive incluye aplicaciones de procesamiento de textos, hojas de cálculo, presentaciones, formularios y de dibujo que pretenden ser una alternativa liviana a Microsoft Office. La ventaja de usar estas aplicaciones basadas en la nube es que si hay un problema con el dispositivo de almacenamiento y el sistema operativo necesita ser restaurado, no se perderán los documentos, ya que se almacenan en la nube y no en un disco duro local.

## XBMC y MX Player

¿Qué decodificador estaría completo sin un buen reproductor multimedia? Tanto XBMC y MX Player ofrecen experiencias de visualización a 1080p de primera categoría, con característi-

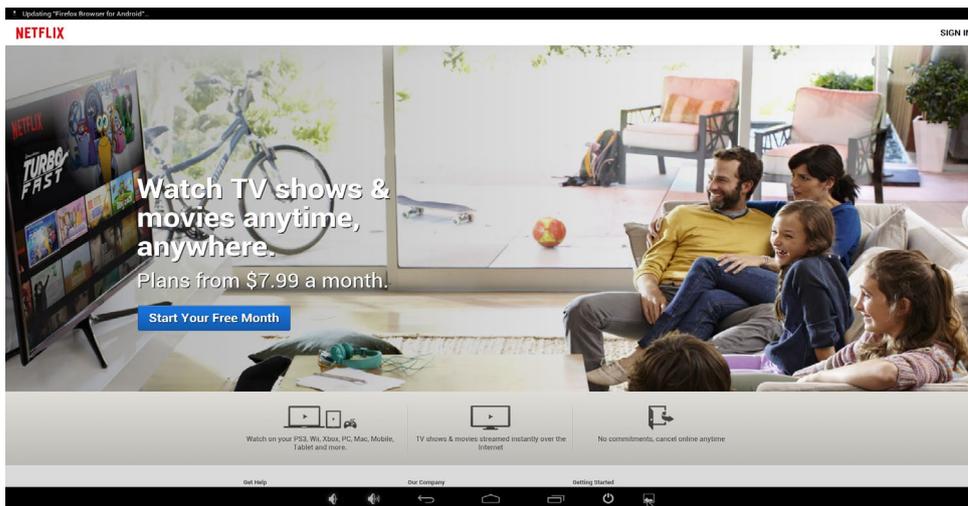
**MX Player es el mejor reproductor multimedia para Android sobre ODROID y permite la decodificación de video por hardware a 1080p**

cas avanzadas tales como subtítulos en varios idiomas, visualizaciones de audio, streaming en red y uso compartido de archivos Samba. XBMC también es compatible con los complementos que ofrecen canales de televisión y otros servicios de medios de comunicación.

MX Player es posiblemente el mejor reproductor de audio y vídeo disponibles para Android. Es capaz de reproducir casi cualquier tipo de archivo multimedia. También tiene una opción de pantalla completa que oculta los botones de Android de la parte inferior de la pantalla de forma automática, que permanecen visibles mientras preproduces vídeos en XBMC. Asegúrate de activar todas las opciones de decodificación HW+ en el menú Configuración.

## tTorrent

Los Torrents se asocian automáticamente a Chromium con la aplicación tTorrent, que es un programa de descarga en red peer-to-peer adecuado para archivos multimedia de gran tamaño. Los archivos completados son, por defecto, guardados en el directorio /mnt/sdcard/Downloads, con la opción de elegir una ubicación alternativa. Tenga en presente que tTorrent consume muchos recursos y una alta actividad del disco puede ralentizar la interfaz de Android. Lo mejor es mover cualquier torrent o actividad de descarga a un segundo ODROID que funcione con Linux, y acceder a los archivos completados usando Samba integrado en ES File Explorer.



¡Netflix es una parte esencial de cualquier decodificador!

## Netflix

Netflix es lo que necesitas como sistema de entretenimiento y ODROID lo puede ejecutar a 1080p en la enorme pantalla de tu salón. Este servicio basado en suscripciones no sólo ofrece un enorme catálogo de películas, hay varios shows originales que sólo se pueden ver por Netflix. La app de Netflix para Android es fácil de usar, con una interfaz similar a la versión de escritorio. Las películas se pueden buscar utilizando un teclado real en lugar de seleccionar letras con un mando remoto desde una lista alfabética como un juego de los 80.

## Youtube y TED Talks

La aplicación de Youtube funciona extraordinariamente bien y puede reproducir videos 1080p a pantalla completa

sin la reproducción glitching. La aplicación TED Talks ofrece horas de videos educativos y de estimulación sobre una gran variedad de temas, presentada por una organización sin ánimo de lucro. Para obtener más información sobre TED Talks, visite [www.ted.com](http://www.ted.com).

## Hangouts

ODROID, cuando se combina con una cámara externa, como la Logitech C920 compatible con video 1080p, es igual que un PC de escritorio. Se puede utilizar el micrófono interno de la cámara o un auricular externo con micrófono.

## Angry Birds

Uno de los juegos más populares para móviles de todos los tiempos es incluso más divertido en ODROID. Los desar-

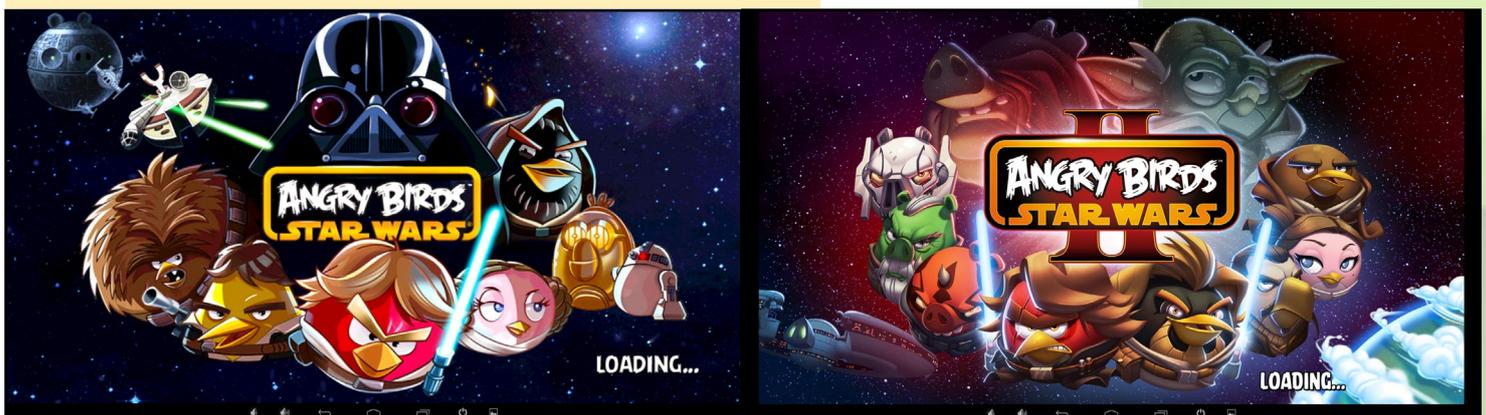
rolladores del juego lo adaptaron para poder usarse con un teclado y un ratón. Si mantienes presionado el botón izquierdo del ratón es lo mismo que arrastrar en una pantalla táctil. Las versiones gratuitas de Star Wars, Star Wars II y Space ya están instaladas,. Otras versiones incluyendo las más recientes, se pueden descargar desde Google Play Store.

## Mupen64 Plus y Retroarch

Pocket Rocket puede convertir un ODROID en una superconsola de juegos para la familia con estas aplicaciones de emulador. Tanto Mupen64 Plus y Retroarch soportan hasta 4 jugadores de forma simultáneos y son compatibles con decenas de miles de juegos. Play Store ofrece muchos emuladores gratuitos y he probado algunos, tanto para jugar como para ver su compatibilidad con distintos mandos antes de decidirme a incluir Mupen64 Plus y Retroarch.

Retroarch es un emulador multi-sistema compatible con muchas consola que van desde Atari 2600, SNE, Sega Genesis, PlayStation Portable y PlayStation 1. Es una aplicación de código abierto que combina muchas generaciones de juegos de videoconsolas retro y modernas en un único sistema. Permite saltar de una consola a otra sin tener que volver a configurar los controles. Ojea la edición de julio de ODROID Magazine para obtener instrucciones sobre cómo conectar el mando Xbox 360 inalámbrico a Retroarch, que es el mando más compatible que he encontrado para u-

¿A quién no le gusta Angry Birds? Pocket Rocket y Couch Potato vienen con 3 versiones de este juego altamente adictivo



sarse con Android.

Mupen64 Plus es un excelente emulador de Nintendo 64 y fue el que menos errores presentaba de todos los que probé. Tiene una excelente interfaz para configurar rápidamente los controles y viene con varios perfiles de mandos pre-cargados, incluyendo el mando Xbox 360 inalámbrico. Tanto Retroarch como Mupen64 Plus permiten el uso de teclados y de mandos, así como una interfaz de control alternativa para pantallas táctiles, que es ideal para usarse con la pantalla táctil externa ODROID-VU.

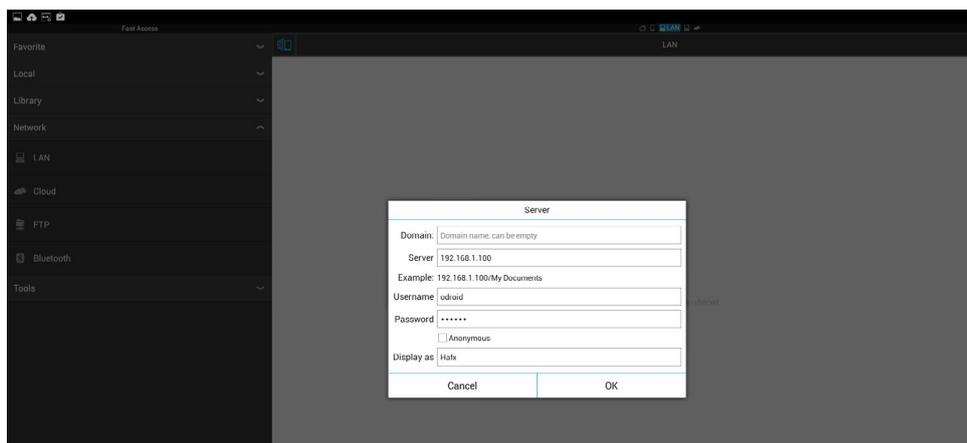
## Pandora y PC Radio

Pandora es una aplicación basada en suscripciones que selecciona música basándose en tus preferencias y gustos personales. Si quieres tener una mezcla de música continua para una fiesta o cena, monta una estación Pandora y pulsa el botón de encendido en la parte superior de U3 (o el botón de alimentación del teclado) para apagar la pantalla. PC Radio es una aplicación de radio por Internet que incluye las emisoras de todo el mundo.

## ES File Explorer y Ultra Explorer

Estos exploradores de archivos son más completos que algunos exploradores de escritorio, permiten compartir por Samba, extraer archivos zip, asociar archivos a aplicaciones y cuentan con múltiples bloques para copiar/pegar. Ultra Explorer está disponible como un proyecto de código abierto en el que los desarrolladores se animan a contribuir con sus propias modificaciones: <http://bit.ly/1AzAeVG>

Usaba la pestaña LAN del ES File Explorer para conectarme con mi servidor multimedia, de modo que cuando hago clic en algún archivo multimedia se inician automáticamente en MX Player, el cual hace streaming a través de la red. A los Discos duros locales conectados por



**ES File Explorer se puede utilizar para conectar dispositivos por Samba y hacer streaming en red.**

USB se accede mediante la pestaña Local del ES File Explorer, siendo enumerada cada unidad USB tras auto-montarse.

## bVNCFree y JuiceSSH

Uso Android como sistema operativo principal sobre todo por bVNCFree. Con esta aplicación VNC, puedo acceder a cualquiera de mis equipos Linux y utilizar el escritorio de Ubuntu remoto como si se tratara de una aplicación nativa para Android. Configurando un servidor VNC en el segundo ODROID usando Vino sólo necesito un único

monitor, teclado y ratón, que pueden ser compartidos entre muchos ordenadores. JuiceSSH ofrece las mismas funciones que la versión Linux de SSH, con una bonita interfaz gráfica para organizar la lista de ordenadores.

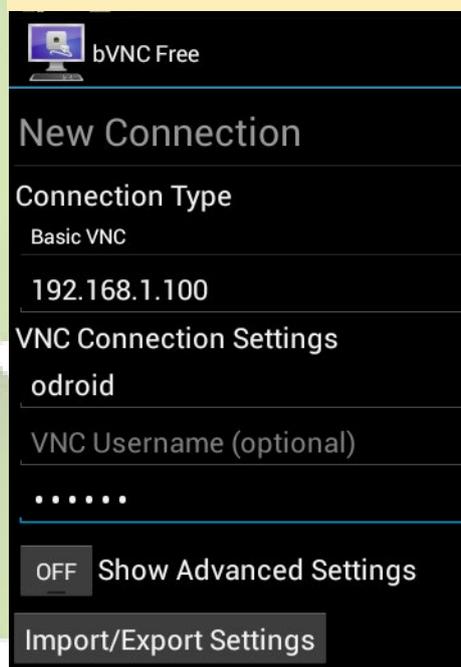
## Terminal Emulator y BusyBox Free

Terminal Emulator es una versión para Android de terminal BASH, y se puede utilizar junto con BusyBox para realizar todas las tareas administrativas de Linux que no se incluyen con Android, como el montaje, grep, ping y uname. Para obtener más información sobre los comandos de BusyBox, puedes consultar la página oficial de Wikipedia en <http://bit.ly/1zwVNEP>

## Jota+

Jota+ es un editor de texto estándar, con casi todas las características que normalmente incluye un editor de texto de escritorio como TextPad. Se puede utilizar para desarrollo web utilizando ES File Explorer para asociar los archivos html y php con Jota+. He creado un recurso compartido Samba en mi servidor web ODROID para editar de forma remota los archivos del sitio web. A menudo utilizo Jota+ para hacer cambios rápidos en los archivos a través de Samba sin tener que arrancar un IDE completo como Netbeans en Linux.

**bVNCFree puede conectarse a cualquier máquina con Windows, Linux, Android o OSX**



## Screen Orientation y full!screen

Estas dos aplicaciones permiten un mejor control del escritorio de Android. Screen Orientation bloquea la pantalla en modo horizontal o vertical, dependiendo de la orientación del monitor. Full!screen oculta temporalmente el menú inferior de Android hasta que se hace click en una esquina inferior.

## Bluez IME Settings, Joysticks y Touchpad

Los Joysticks Bluetooth, como Wii-mote, se pueden configurar usando Bluez IME Settings, para ser usados como teclado. Los mandos de la Xbox 360 se conectan directamente a Pocket Rocket. Se pueden utilizar para navegar por el escritorio, lanzar aplicaciones y utilizar el teclado virtual en pantalla.

## ODROID Updater y ODROID Utility

La versión oficial Hardkernel de KitKat Pocket Rocket está en fase beta desde agosto de 2014, pero ODROID Updater se puede utilizar para añadir las últimas mejoras y correcciones a medida que KitKat avanza hacia su versión estable. Siga los botones que aparecen en pantalla para instalar las actualizaciones del sistema, que normalmente requieren reiniciar. ODROID Utility está diseñado para ajustar las variables de arranque como la resolución de pantalla a 720p o 1080p, orientación de la pantalla y configuración del regulador de CPU. Las opciones de arranque adicionales, como el parpadeo del LED son ajustadas en el archivo /mnt/sdcard/boot.ini

## Instalar aplicaciones personalizadas

Aunque Pocket Rocket está cargado de funciones, todavía puede servir como

un entorno de prueba Android para tus propias aplicaciones, como estaba previsto originalmente, mediante el software Android Debug Bridge. Instala el cliente ADB en una máquina Linux seleccionándolo desde el Gestor de paquetes Synaptic. Conecte el puerto micro-USB del ODROID a la máquina con un cable USB, y establezca una conexión ADB escribiendo los siguientes comandos en una ventana de terminal:

```
$ su
# adb remount
# adb shell
```

Esto abre una sesión parecida a SSH en la máquina de Android, permitiendo el acceso a la línea de comandos en la máquina remota. Los archivos se pueden transferir a través de ADB a Pocket Rocket con el comando:

```
# adb push <local directory> \
<remote directory>
```

Para obtener más información sobre el uso de ADB y otras formas de conectarse a ODROID desde otra máquina, consulta <http://bit.ly/1xC42wk>



Escritorio Pocket Rocket con una animación de una vía láctea en remolino



En el menú de juegos es donde paso la mayor parte de mi tiempo



Relájate con Pandora o con la Radio por internet a nivel mundial usando PCRadio



El menú del sistema le permite acceder a archivos protegidos y a otros equipos



ES File Explorator y Ultra Explorator son dos exploradores de archivos muy completos



Chrome y Firefox son los navegadores disponibles más populares



El menú de video también contiene Google Hangouts y TED Talks

# CONOCIENDO A UN ODROIDIAN

## BO LECHNOWSKY: EXPERTO CREADOR E INVENTOR DE INSPIRACION

por Bohdan Lechnowsky  
editado por Rob Roy

*Por favor, háblanos un poco sobre ti.*

Nací en Omaha, Nebraska EE.UU. de padres inmigrantes ucranianos y alemanes. Me mudé a Ukiah, California EE.UU. después de haber estado casado durante 7 años y haber tenido un hijo y una hija. Ahora tengo dos hijas más que hacen un total de 4 niños, y estoy casado desde hace 23 años. En 1994, monté una tienda de ordenadores Amiga. En 2001, puse en marcha una consultora de tecnología que en la actualidad da empleo a cerca de una docena de personas, en su mayoría técnicos.

*¿Cómo fueron tus inicios con los ordenadores?*

Con anterioridad a los ordenadores, tenía un cierto interés en la electrónica. Recibí mi primer soldador a los 7 años. Mis padres vivían con una economía muy ajustada, así que no podían permitirse el lujo de comprar nuevos com-

ponentes electrónicos. De modo que, tuve que desmontar la electrónica que iba a ser desechada y salvar lo que pudiera. Pasé algún tiempo en la biblioteca pública leyendo libros sobre circuitos electrónicos para aprender más sobre este mundo. Cuando tenía 12 años, mi escuela compró unos cuantos ordenadores Apple II. Tenía que quedarme después de clase para poder usarlos lo que implicaba perder el autobús y darme un paseo de 2.5 millas hasta casa. Una vez más, tuve que aprender por mí mismo programación con los libros de la biblioteca. Tuve mi propia Commodore 64 a los 15 años. En la escuela y en la universidad, trabajé con todo tipo de ordenadores, en la mayoría de los casos programando.

Bo Lechnowsky, Uno de los editores de ODROID Magazine, tocando su instrumento de viento de madera al aire libre



*¿Cuál consideras que es tu ODROID favorito?*

Mi favorito ODROID es el U3 ya que es un equipo muy potente y capaz, de bajo precio y con un montón de accesorios de su mismo tamaño como el Show, placa E/S y la placa UPS, con los cuales he estado trabajando bastante últimamente.

*Eres un experto en lenguaje Rebol, y a menudo escribes artículos que muestran la sencillez del lenguaje. ¿Qué software has desarrollado con Rebol?*

**¡Aquí es donde todos los experimentos informáticos tienen lugar!**



Toda mi empresa funciona con software Rebol que desarrollé a comienzos del 2003, desde el inventario y registro del tiempo a la facturación, cuentas de la empresa y gráfica en tiempo real que muestra el progreso de nuestros objetivos. Además, he desarrollado cámaras de seguridad súper eficientes que capturan y procesan el vídeo a una resolución de 1920x1080 @ 30fps. He escrito el software de seguimiento que se puede utilizar para ver lo que está pasando, y revisar lo que ya ha sucedido. Para otro proyecto, traslade la base de datos de selección médica de un cliente desde un macro ordenador a una solución Rebol / SQL que está en constante uso desde 2001. Podría seguir, pero en serio, tendría que escribir un libro para poder enumerar todos mis proyectos escritos en Rebol.

*¿Qué aficiones e intereses tienes aparte de los ordenadores?*

Me gusta decir que si necesitas un hobby, yo estaría encantado de darte uno de los míos. Sería más fácil enumerar las aficiones que no tengo. Algunos de los hobby con lo que estado pasando últimamente más tiempo han sido criar una familia, tocar instrumentos con la banda de mi iglesia, desarrollar nuevos productos tecnológicos y comerciales, trabajar en la mecánica y carrocería de mi camioneta, trabajos de herrería tradicional, participar en actividades al aire libre (bicicleta de montaña, camping, barbacoas, World Human-Powered Speed Championships), viajar (incluyendo tres visitas a Ucrania), la lingüística y la edición.

*¿Estás trabajando en algún proyecto?*

Actualmente, He sido contratado para ofrecer una solución que ayude a mejorar la producción de alimentos mediante un proceso automatizado en tiempo real. En este proyecto, estamos utilizando un U3 con las placas Show y IO Shield junto con sensores GPS y láser, y Rebol gestiona toda la lógica. Otro proyecto en el que acabo de terminar fue la puesta en funcionamiento de un servidor de video vigilancia usando un U3 y un dispositivo de almacenamiento externo. Actualmente estoy tratando de sacar tiempo para mejorar mi camioneta Ford F150 4x4 de 1978 incorporándole un ODROID que me permita sustituir el cuadro de mandos por pantallas táctiles digitales y placas Show, arrancar y parar el motor con mi teléfono inteligente, registrar y controlar los niveles de combustible y los datos de rendimiento, incorporar navegación GPS y otros servicios automáticos. Por desgracia, mis proyectos del trabajo tienen prioridad sobre los personales, como mejorar mi camioneta. Si tengo la oportunidad de ponerme con ello, escribiré un artículo sobre el proyecto de mi camioneta para un futuro ejemplar de la revista.

*¿Qué tipo de novedades en hardware te gustaría ver en futuras placas de Hardkernel?*

No he podido mirar los requisitos técnicos de cualquiera de ellas, pero me gustaría añadir algunas sugerencias, me encantaría ver una futura placa de Hardkernel con la nuevo Soc de Sam-



**Hiking Smith Rock State Park en Oregon con la familia**

sung (parece que conseguiré mi deseo con la XU3). También sería increíble tener una ranura para memoria similar a los ordenadores portátiles con vista a su expansión, y ejecutar el ODROID con PoE (Alimentación a través de Ethernet). Por supuesto, disponer de muchos puertos USB son siempre una gran ventaja (sé que algunas placas ya tienen unos cuantos). ¿Quizás bluetooth integrado? Dicho esto, mi opinión es que Hardkernel hace un gran trabajo manteniendo un equilibrio entre potencia, servicios y precio.

**El laboratorio de informática de Bo es tan grande que es necesaria una segunda página para mostrarlo por completo!**





Parte de mi negocio, Respectech, con el suelo de metal flotante y la iluminación LED por debajo.

*¿Qué consejo le darías a alguien que quiera empezar con la programación?*

Uno de los grandes obstáculos al que tiene que hacer frente muchas personas es la configuración de un entorno de desarrollo. Con esto en mente, No puede hacer frente a la programación C con gcc (un compilador de C) que está en casi todas las distribuciones, y casi todas las plataformas utiliza C (incluyendo dispositivos como Arduino). Uno podría pensar que Bash también sería una buena opción, ya que también está incluida en casi todas las distribuciones. Sin em-

bargo, mi opinión es que bash se excede en la sintaxis y suele ser algo engorroso para desarrollar soluciones. Mi favorito, por supuesto, es Rebol. Funciona en casi todas las plataformas sin tener que realizar cambios en el código, se trata de único archivo de menos de 1 MB descargado. Las cosas simples son fáciles de hacer, mientras que las complejas son todavía posibles. No sólo eso, sino que permite desarrollar dialectos que permiten crear sistemas complejos con cierta facilidad, lo que reduce los esfuerzos en desar-

rollo y depuración. El lenguaje que uses debe basarse en los objetivos finales de tu proyecto. Para mí, Rebol tiene la flexibilidad y la capacidades para gestionar casi todos los proyectos que he lanzado con él, y me permite desarrollar soluciones por la decima parte de tiempo que tardaría con otros lenguajes. Y ahora que Red está madurando, va a ser una gran alternativa con muy poco aprendizaje para los desarrolladores que deseen familiarizarse con este lenguaje.

Puedes visitar el sitio web de Bo en <http://www.respectech.com>.

**Los "trabajos pesados" que hacemos para conseguir que mi Ford F150 4x4 de 1978 esté listo para instalar los ODROIDS y controlar los componentes electrónicos. Mi suegro me ayudó con este paso.**



**No puedo decir exactamente lo que hace este proyecto (todavía es alto secreto), pero te puedo decir que implica a un U3, un SHOW y un I/O Shield.**

