

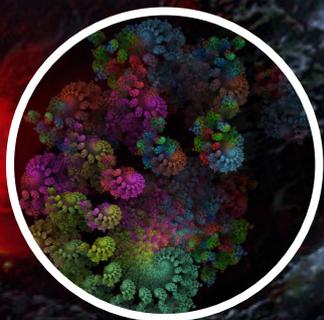
ODROID

Año Uno
Num #6
Jun 2014

Magazine

DREAM MACHINE Y WHISPER

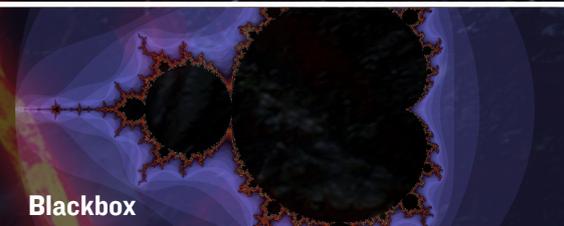
UN VISTAZO AL INTERIOR DE
UBUNTU 13.04 Y 13.10
IMAGENES QUE PERMITEN
A TU ODROID ARRANCAR Y
EJECUTARSE MÁS RAPIDO



Herramientas matemáticas Ubuntu



Lubuntu Whisper 13.10 Saucy Salamander



Blackbox



LXDE



XBMC

- ADEMAS:**
- EMPAQUETA TU PROPIO SOFTWARE COMPILADO
 - PROGRAMA TU ODROID-SHOW CON REBOL
 - ACTIVA LA SALIDA DE AUDIO MULTI-CANAL CON XBMC



LA FAMILIA ODROID
ODROID-X2
ODROID-XU
ODROID-U3

Qué defendemos..

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

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

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

Simple, moderno y único.

De modo que tienes a tu alcance lo mejor para alcanzar todo lo que puedas imaginar.



HARDKERNEL



Ahora estamos enviando los dispositivos ODROID U3 a los países de la UE! Venga y visite 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





Desde que iniciamos la publicación **ODROID Magazine**, hemos recibido muy buenos artículos de nuestros colaboradores internacionales en los que se recogen sus innovadores proyectos de hardware, como la gigante **Tablet Android**, el **CAR PC** con **ODROID** y el **Robot Todoterreno**. La comunidad internacional de desarrolladores ha demostrado que no hay prácticamente límite alguno para **ODROID**.

Puesto que muchos **ODROIDians** participan en proyectos de software, tenemos la intención de ofrecer algunas de las muchas y excelentes imágenes con **Sistemas Operativos** en las que la comunidad **ODROID** y el equipo de desarrollo de **Hardkernel** han estado trabajando. La sección de la revista destinada a tal fin recibe el nombre de **“SO Destacado”**.

Iniciamos esta sección el mes pasado con una descripción del software que incluye la imagen **Fully Loaded** de **Ubuntu 12.11**, y este mes hemos destacando **Dream Machine**, que es ideal para trabajos de oficina, programación, reproducción multimedia y para cualquier otra cosa que se pueda hacer con un **PC** de escritorio tradicional.

A 1 de junio, la cantidad de miembros de los foros **ODROID** ascendía a 5.000 (<http://forum.odroid.com>), mostrando un asombroso crecimiento del 500% en el último año y medio. Con tanto interés en la plataforma **ODROID**, la versión de **U3** se agotó el mes pasado, debido posiblemente a su reducido tamaño y precio muy asequible. Para responder a la creciente demanda, **Hardkernel** pronto comenzará a distribuir su nueva generación del hardware de la serie **U** denominada **ODROID-U3+**, que presenta varias y nuevas características. <http://bit.ly/1p8zi6L>

Para los aficionados a **Android**, **Hardkernel** ha publicado un avance de la versión **Beta** de **Android KitKat 4.4**, que se puedes probar y descargar desde <http://bit.ly/1hsFdRB> Todavía es un proyecto en desarrollo, así que si detectas cualquier error no olvides comentarlo en los foros **ODROID**.

En respuesta a las numerosas solicitudes relacionadas con el sonido **HDMI Passthrough DTS**, se ha creado un nuevo componente de hardware llamado **USB-S/PDIF**, disponible en la tienda **Hardkernel**. El **USB-S/PDIF** envía la codificación digital **AC3** y **DTS** a través del cable **HDMI** a tu receptor de audio, proporcionando un sonido envolvente **5.1** nítido a tu salón usando tu **ODROID** con **Linux** y **XBMC 13**. Su coste es muy asequible, asciende a **16\$**. <http://bit.ly/1kIt8aC>

Hardkernel no sólo ha estado actualizando sus imágenes de **Ubuntu** a la versión **14.04**, sino que también ha mejorado la experiencia **XBMC** con su tan esperada versión oficial **1080p** de la imagen **Media Center** basada en el popular **XBMC**. La nueva versión de **XBMC** para **Linux**, disponible para todas las plataformas, aprovecha las recientes mejoras en la decodificación de vídeo con el kernel **3.8** permitiendo la reproducción de vídeos a **720p** y **1080p** sin problemas para todas las codificaciones que soporten **GPU Mali** y hardware **VPU**. Ten en cuenta que se trata de un desarrollo experimental, pero nos han informado que funciona bastante bien. Es una gran distribución para aquellos que previamente utilizaban **OpenELEC** o **Xbox** como reproductor/decodificador. Echa un vistazo a su desarrollo en <http://bit.ly/11fkYpN>

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



HARDKERNEL

ODROID

Magazine



**Robert Hall,
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 las mías ya que tengo demasiadas. Eso ayudaría a que tuviese más tiempo para estar con mi maravillosa esposa de 23 años y mis cuatro hijos estupendos.



**Bruno Doiche,
Editor Artístico**

Consiguió sus habilidades informáticas después de lograr que una fibra óptica volviera a la vida, lograr que su Macintosh volviese de la muerte, lograr que una PS3 volviese de la muerte, lograr que el T400 de su novia volviese de la muerte (una transferencia de datos dd al viejo estilo), y liando con las entrañas de su permanente centro de datos de trabajo.



**Manuel
Adamuz,
Traductor**

ODROID en el salón, en el dormitorio, en el despacho, en la oficina... Es posible que tenga algo de obsesión!

Novedades de esta Edición:

Volver a nuestro horario habitual no ha sido tarea fácil. Después de un par de contratiempos sin importancia, hemos trabajado duro para lograr un mayor ritmo de trabajo, pero sin perder, como siempre, ¡Nuestra mejor sonrisa! ¿Cuáles han sido los resultados? Bueno ... no ha sido posible incluir todos los consejos y trucos que queríamos, pero hemos mejorado la gama de colores y el tamaño de letra. ¡La lectura se hace más llevadera y presenta mejor aspecto!

La portada también ha supuesto todo un reto, debatimos y decidimos que el artículo de portada incluiría las distribuciones Linux de Rob. He tenido una cierta obsesión por reunir todos los ODROIDS en producción, pero lo creas o no, no ten-

go la tirada completa (U3, X2, XU). Hay un montón de ODROIDians que nunca los han visto juntos. ¡Ya se me ha quitado la obsesión! Tengo un amigo que me ha enviado una foto de sus ODROIDS para que la incluya en la portada. Así que te doy las gracias, Patola y espero que aparezcas en la entrevista relacionada con la impresión 3D usando ODROIDS!

¿Qué más hay? Ahora, Rob trabaja directamente sobre el diseño de los artículos. Hemos conseguido mucha más agilidad en el proceso de revisión y ha sido como siempre, trabajando duro para ofrecer a nuestros fieles lectores datos concretos, ¡Una revista que nos apasiona publicar! ¡No es que no hagamos de vez en cuando algún curioso y alocado número!



PROGRAMANDO TU ODROID-SHOW 6

RECOMPILAR LOS DRIVERS MALI 7

EMPAQUETA TU SOFTWARE COMPILADO PARA SU INSTALACIÓN 8

DESCRIBIENDO LA FUNCIÓN MATEMÁTICA ATAN2 17

CONSOLA TERMINAL FRAMEBUFFER 20

HERRAMIENTAS MATEMATICAS PARA ODROID 20

ARCHIVOS IMAGEN DE ANDROID 22

REDIMENSIONAR LAS PARTICIONES DE ANDROID 26

GUIA GRAFICA PARA REDIMENSIONAR UNA SD O EMMC CON ANDROID 28

COMO ALIMENTAR A TU GATO POR INTERNET 30

FABRICA UNA CAJA LEGO A MEDIDA PARA TU U3 33

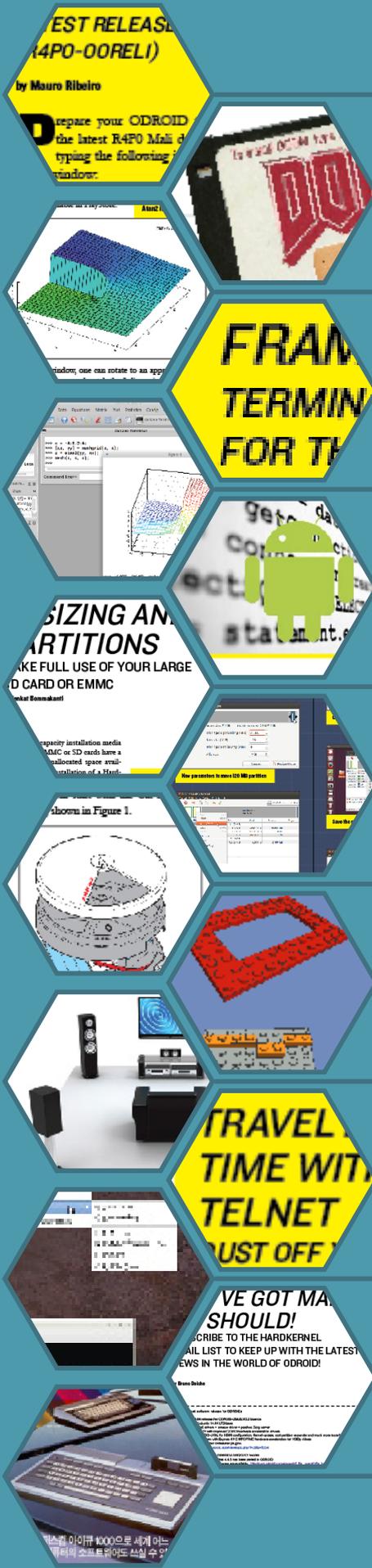
COMO ACTIVAR LA SALIDA DE AUDIO MULTI-CANAL CON XBMC 34

VIAJA A ATRAS EN EL TIEMPO CON TELNET 35

SISTEMA OPERATIVO DESTACADO: DREAM MACHINE Y WHISPER 36

TIENES EMAIL... O ¿DEFERIAS! 39

CONOCIENDO A UN ODROIDIAN 40



PROGRAMANDO TU ODROID-SHOW

USA EL LENGUAJE DE PROGRAMACION REBOL PARA MEJORAR LA INTERFAZ DE HARDWARE

por Bo Lechnowsky

El módulo ODROID-Show es increíblemente útil para muchas cosas. Por ejemplo, es un gran dispositivo para mostrar información que se pueda ver de un vistazo. Sin embargo, lograr que la información se muestre puede resultar una tarea algo engorrosa.

Para simplificar las cosas, he usado las increíbles capacidades de Rebol para crear un sencillo dialecto que permita la visualización de diversos tipos de información con ODROID-Show. Puesto que Rebol es multiplataforma, los scripts se pueden ejecutar en cualquier sistema operativo.

También puedes conectar varios módulos ODROID-Show a un único ODROID o a cualquier otro ordenador con puertos USB. Este dialecto te permitirá hacer esto con facilidad.

La función para iniciar el dialecto se llama oshow (La palabra show ya es usada en Rebol para actualizar GUI).

He resaltado en negrita el código que se envía al módulo ODROID-Show con el dialecto oshow. Como bien se sabe, la programación bash no es fácil de leer, especialmente cuando se entremezcla con los códigos de control. Este ejemplo de Rebol es aproximadamente un 35% más pequeño, permite cualquier rotación de colores, cuantas palabras se quieran de longitud variable y se puede ejecutar en cualquier plataforma con un intérprete de Rebol 3 (en casi todas excepto Windows RT y IOS a las que llegará pronto).



Usando el dialecto oshow, compara el programa escrito en Rebol con el ejemplo de programación bash facilitado en el Tutorial # 1: Guía de usuario para la salida de texto con ODROID-Show.

```
do %oshow.r3

  call {port_open}
  serialport: %/dev/ttyUSB0
  display-text: ["ODROID" 25x100 "SHOW" 55x150]
  oshow serialport [reset rotate 0]

  forever [
    foreach color [red green yellow blue magenta cyan white] [
      foreach [word position] display-text [
        oshow serialport compose [cursor to (position) font (color) 5]
        foreach letter word [
          oshow serialport compose [(to-string letter)]
          wait .02
        ]
      ]
    ]
  ]
]
```

Esta es la definición completa del dialecto.

font opcionalmente seguido de:

black	
red	opcionalmente seguido de:
green	default (establece el color por
yellow	defecto como color de primer plano)
blue	
magenta	opcionalmente seguido de:
cyan	1 (fija el tamaño a: 6x8 pixels *
white	el número especificado aquí)

Ejemplos:

```
oshow [font cyan]
oshow [font white default]
oshow [font 2] (fija el tamaño a 12x16 pixels)
oshow [font 1 cyan "Hello " 2 red "ODROID!"]
```

background seguido de uno de los colores fijados en primer plano.

opcionalmente seguido de:

default (establece el color por defecto como color de fondo)

Ejemplos:

```
oshow [background black]
oshow [background yellow default]
```

cursor seguido de:

down
up
position? (informa de la posición actual del cursor)
save (guarda la posición actual del cursor)
restore (restaura el cursor a la posición previamente fijada)
to (seguido de):
3x5 (mueve el cursor a la columna 3, fila 5)
home (mueve el cursor a la columna 1, fila 1)
bottom (mueve el cursor a la fila 1 de la misma columna)

reset (resetea la pantalla LCD)

clear (borra la pantalla)

off (apaga la pantalla LCD)

on (enciende la pantalla LCD)

key (teclado simulado) seguido de:

2 (opcional: el número de teclas - por defecto 1)
up
down
left
right

rotate (gira la pantalla 90 grados) seguido de:

left (en sentido contrario a las agujas del reloj, 90 grados)
right (en el sentido de las agujas del reloj, 90 grados)

draw (aún no funciona) seguido de:

10x20 (opcional: esquina superior izquierda de la imagen - por defecto 0x0)
50x50 (opcional: el tamaño de la imagen - por defecto 320x240)
<image data>

RECOMPILAR DRIVERS MALI ACTUALIZANDO A LA ULTIMA VERSION (R4P0-OORELI)

por Mauro Ribeiro

Prepara tu ODROID para ejecutar los últimos drivers R4P0 Mali escribiendo en una ventana de terminal:

```
sudo apt-get install xserver-xorg-video-armsoc xserver-xorg-dev libudev-dev
```

El 5 de mayo de 2014, Hardkernel dejó de dar soporte a los drivers r3p2 de Mali y ha pasado a utilizar exclusivamente la versión R4P0. Existen dos cambios principales:

1. En lugar de usar DDX Mali, se instala el driver Armsoc.
2. UMP ya no existe y Malí utiliza ahora dma-buf.

Hemos facilitado un DDX Armsoc actualizado con el fin de adaptar tu actual sistema operativo Ubuntu o Debian a los nuevos drivers Mali. El código fuente de la serie r4p0 se encuentra en <http://bit.ly/1oySeYE>. Descarga la fuente en una carpeta temporal, escribe **make install** en la línea de comandos tras de navegar hasta la carpeta temporal.

Blobs para EGL/GLESv1/GLESv2 están disponibles para su descarga gratuita en <http://bit.ly/1hiwJw9> que incluye un ejemplo del archivo xorg.conf opcional.

Ten en cuenta que si tu defconfig todavía tiene UMP, probablemente dejará de funcionar, así que asegúrate de eliminarlo primero.

Si tienes alguna pregunta sobre los nuevos controladores, no dudes en recurrir a los foros de la comunidad ODROID <http://forum.odroid.com/>

¿TIENES UN MANDO WII? CREA UN ASOMBROSO RATON GIROSCOPICO

por Rob Roy

Tienes un mando Wii al que últimamente no has dado mucho uso. De hecho es un periférico de entrada de hardware muy elegante, con un diseño ergonómico, un escáner de infrarrojos y sensores giroscópicos. Ubuntu tiene varios paquetes disponibles desde sus repositorios que leen automáticamente los sensores del mando Wii y los traduce como movimientos de ratón así como sus botones. Para instalar el software, escriba:

```
$ sudo apt-get install libc-wiidi lswm wmgui wminput
```

Una vez completada la instalación, conecte el mando Wii pulsando 1 y 2 al mismo tiempo, activando el modo de conexión por Bluetooth. Una vez que las luces del mando empiecen a parpadear, escriba:

```
$ hcitool scan
o
$ lswm
```

Cualquiera de los dos comandos mostrará la dirección MAC para todos los mandos Wii cercanos que se encuentren actualmente en modo de conexión por Bluetooth. Cargue el driver del mando escribiendo lo siguiente, usando la dirección MAC del paso anterior. En este ejemplo, la dirección MAC es 01:FA:2C:9D:BB:05:

```
$ sudo wminput \
01:FA:2C:9D:BB:05
```

El mundo Wii será considerado por Ubuntu como un ratón giroscópico, adecuado para situaciones en las que un ratón normal no sería tan apropiado, como el hecho de ofrecer una presentación en grupo.

Este es el código para el dialecto oshow, que se guarda como "oshow.r3", o se puede descargar desde la URL del encabezado que aparece a continuación.

```
REBOL [
  Title: {ODROID Show Dialect}
  Date: 20-May-2014
  Author: {Bohdan Lechnowsky}
  URL: http://respectech.com/odroid/odroid-show.r3
  Purpose: {
    Provides a dialect to allow easy usage of the ODROID Show hardware component
  }
]

oshow: func [
  {Allow easy utilization of the ODROID-Show display module}
  serport {Name of the serial port to send to (i.e. "serial://tty-USB0/57600" or "%/dev/ttyUSB0")}
  specs [block!] "Dialect block of layout"
  /local outstring
][
  outstring: copy {}

  out: func [str /local pstr buffs] [
    ;Break string into lines and output lines one at a time
    foreach lstr pstr: parse/all str "^/" [
      ;The ODROID-Show hardware is limited to 256 characters per line, but this
      ; can be overcome by breaking the line into 256-character chunks and
      ; outputting them one after the other.
      buffs: copy []

      ;Replace tabs with four spaces before breaking apart into chunks.
      while [not tail? detab/size lstr 4] [
        append buffs copy/part lstr 256
        lstr: skip lstr 256
      ]
      foreach buff buffs [
        write serport join buff "^J^M"

        ;In my tests, I find that wait values less than 80ms often lead
        ; to output corruption. Longer waits may be required with some
        ; output.
        wait .08
      ]
      append outstring lstr
    ]
  ]
]
```

```

]
]

font-rules: [
  color-rules (out join "^[[3" col)
  | 'default (out "^[[39m")
  | set font-sz integer! (out rejoin ["^[[["
font-sz "s"])
  | set textout string! (out textout)
]
back-rules: [
  color-rules (out join "^[[4" col)
  | 'default (out "^[[49m")
]
color-rules: [
  'black (col: "0m")
  | 'red (col: "1m")
  | 'green (col: "2m")
  | 'yellow (col: "3m")
  | 'blue (col: "4m")
  | 'magenta (col: "5m")
  | 'cyan (col: "6m")
  | 'white (col: "7m")
]
cursor-rules: [
  'down (out "^[[D")
  | 'up (out "^[[M")
  | 'position? (out "^[[6n")
  | 'save (out "^[[s")
  | 'restore (out "^[[u")
  | 'to any [
    set cpos pair! (out rejoin ["^[[[" cpos/x
";" cpos/y "H"])
    | 'home (out "^[[H")
    | 'bottom (out "^[[E")
  ]
]

```

; 'presses uses a fun trick. If keypresses have been defined by the input from the user,

; then use that number. If not, use the default (1 in this case). This is shorter than

; the old-school method that would look like this:

```

;
; unless value? 'keypresses [keypresses: 1]
; keypresses
;
; (52 bytes, not including the line break)
; ("unless" is a synonym in Rebol of "if not",
which is not any shorter than "unless")

```

```

;
; Another possible method could be like this:
;
; all [not value? 'keypresses keypresses: 1]
; keypresses
;
; (53 bytes, not including the line break)
;
; (The "do pick" method below is only 41
bytes)

presses: [do pick [keypresses 1] value? 'key-
presses]

key-rules: [
  'up (out rejoin ["^[[[" do presses "A"])
  | 'down (out rejoin ["^[[[" do presses "B"])
  | 'left (out rejoin ["^[[[" do presses "C"])
  | 'right (out rejoin ["^[[[" do presses "D"])
  | set keypresses integer!
]
rotate-rules: [
  'left (loop 3 [out "^[[[r")
  | 'right (out "^[[[r")
]
show-rules: [any [
  'font any font-rules
  | 'backdrop any back-rules
  | 'cursor any cursor-rules
  | 'reset (out "^[[c")
  | 'clear (out "^[[[2J")
  | 'off (out "^[[[0q")
  | 'on (out "^[[[1q")
  | 'key any key-rules
  | 'rotate any rotate-rules
  | 'draw
  | set textout string! (out textout)
]]
parse specs show-rules
outstring
]

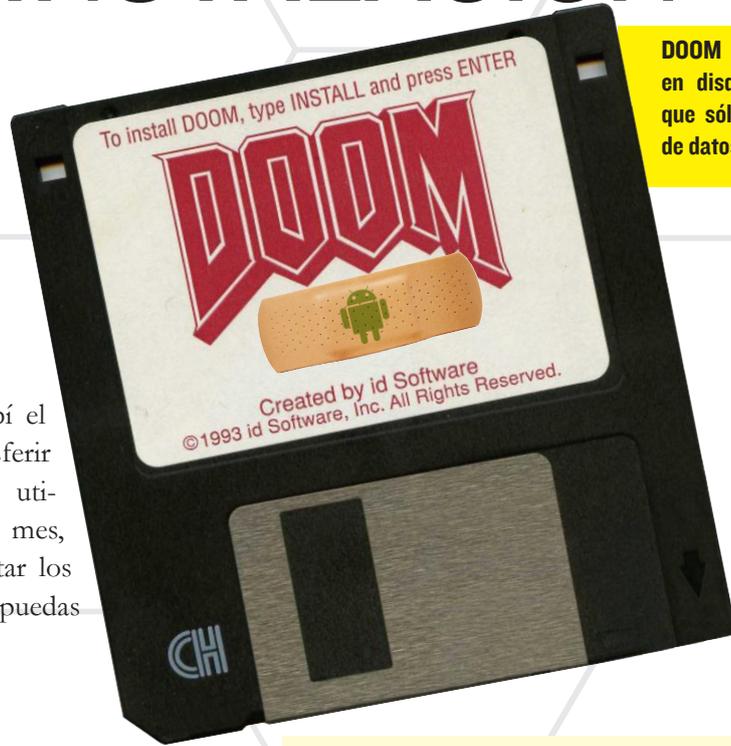
```

Este simple programa Rebol hace que los comandos de ODDROID-show sean más accesible al resumir la sintaxis nativa de estos.

Consulte el sitio web de Rebol para más información en <http://www.rebol.com>, o descarga el código fuente desde <http://www.rebolsource.net>.

EMPAQUETA TU SOFTWARE COMPILADO PARA SU INSTALACION

COMPILANDO DOOM PARTE 2



DOOM venía originalmente en disquetes muy frágiles que sólo contenían 1.44MB de datos del juego.

En el artículo anterior, te describí el proceso para compilar y transferir software y juegos a ODRROID, utilizando DOOM como ejemplo. Este mes, te presento una técnica para empaquetar los ficheros binarios compilados para que puedas compartirlos y distribuirlos.

Checkinstall

Checkinstall es una forma muy útil y simple de crear archivos de instalación (.deb). Te permite empaquetar todos tus programas en un único archivo y añadir las librerías necesarias como una dependencia, de este modo puedes instalarlos en otros sistemas que aún no dispongan de las librerías exigidas para su ejecución.

Si ejecutas checkinstall en el proyecto sldoom, verás algo como esto:

```

root@odroid-wheezy:/home/odroid/sources/sldoom-1.10$
checkinstall

checkinstall 1.6.2, Copyright 2009 Felipe Eduardo
Sanchez Diaz Duran
  This software is released under the GNU GPL.

The package documentation directory ./doc-pak does
not exist.
Should I create a default set of package docs? [y]:

Preparing package documentation...OK

Please write a description for the package.
End your description with an empty line or EOF.
>> SDL Doom

*****
**** Debian package creation selected ****
*****

This package will be built according to these values:
    
```

```

0 - Maintainer: [ root@odroid-wheezy ]
1 - Summary: [ SDL Doom ]
2 - Name: [ sldoom ]
3 - Version: [ 1.10 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ armhf ]
8 - Source location: [ sldoom-1.10 ]
9 - Alternate source location: [ ]
10 - Requires: [ ]
11 - Provides: [ sldoom ]
12 - Conflicts: [ ]
13 - Replaces: [ ]

Enter a number to change any of them or press ENTER
to continue:
    
```

Checkinstall toma el nombre de la carpeta en la se que encuentra actualmente el paquete (sldoom) y automáticamente coge -1.10, que se añadió al nombre de la carpeta como versión. Si los nombres de las carpetas no coinciden, puedes modificarlos desde aquí así como fijar la versión correcta. Asegúrate de cambiar el nombre en los valores 2 - Name y 11 - Provives.

Puedes configurar tu instalación y añadir los paquetes necesarios como libsdl1.2debian que es la librería libsdl exigida para este desarrollo bajo el valor 10 - Requires. Tras pulsar Intro, se debe generar un pequeño archivo .deb



DOOM, ¿Donde el jugador más rápido con la mejor arma gana! A menos que, por supuesto, modifiques el código fuente para hacerte 100% invencible, o seas capaz de disparar a través de las paredes.

```
Enter a number to change any of them or press ENTER
to continue:
```

```
Installing with make install...
```

```
===== Installation results
=====
make[1]: Entering directory `~/home/odroid/sources/
sdlboom-1.10'
/bin/sh ./mkinstalldirs /usr/local/bin
/usr/bin/install -c doom /usr/local/bin/doom
make[1]: Nothing to be done for `install-data-am'.
make[1]: Leaving directory `~/home/odroid/sources/sdl-
boom-1.10'
```

```
===== Installation successful
=====
```

```
Copying documentation directory...
```

```
./
./README.asm
./Changelog
./TODO
./FILES
./README.book
./README.SDL
./README.b
./README.gl
```

```
Copying files to the temporary directory...OK
```

```
Stripping ELF binaries and libraries...OK
Compressing man pages...OK
Building file list...OK
Building Debian package...OK
Installing Debian package...OK
Erasing temporary files...OK
Writing backup package...OK
OK
```

```
Deleting temp dir...OK
```

```
*****
*****
```

```
Done. The new package has been installed and saved to
/home/odroid/sources/sdlboom-1.10/sdlboom_1.10-1_arm-
hf.deb
```

```
You can remove it from your system anytime using:
dpkg -r sdlboom
```

```
*****
*****
```

Aunque el modo interactivo es un método simple para crear un archivo .deb, otra forma es asignar los parámetros correctos al iniciar checkinstall, permitiendo algunas opciones adicionales::

```
$ checkinstall --backup=no --install=no
--requires="libsdl1.2debian" --pkgname="sdlboom-
odroid" --pkgversion=1.10
```

Si inicias checkinstall sin la opción "--backup = no", se creará un archivo tar con todos los ficheros que se copiaron durante el proceso de instalación. Si arrancas checkinstall sin la opción "--install = no", tu software será instalado inmediatamente después de crearse el paquete. Una vez completado el proceso, te encontrarás con un pequeño archivo .deb en tu carpeta.

```
$ ls -l *.deb
-rw-r--r-- 1 root root 174490 Apr 5 15:16 sdl-
boom_1.10-1_armhf.deb
```

Después, el archivo .deb se puede copiar e instalar en cualquier otro ODROID y si has fijado los requisitos de dependencia correctos, se debe iniciar sobre la marcha. Obtener los requisitos correctos te llevará algún tiempo y experiencia, en función del proyecto. El paquete libsdl1.2-dev en realidad, sólo incluye los archivos cabecera para SDL (archivos que terminan en .h y ubicado en algún lugar dentro de /usr/include/). La librería SDL realmente se llama libsdl1.2debian.

Si revisas el paquete libsdl1.2-dev, verás libsdl1.2debian (= 1.2.15-5) en la lista de dependencias.

```
apt-cache show libsdl1.2-dev
Package: libsdl1.2-dev
Source: libsdl1.2
Version: 1.2.15-5
Installed-Size: 2358
Maintainer: Debian SDL packages maintainers <pkg-sdl-
maintainers@lists.alioth.debian.org>
Architecture: armhf
Replaces: libsdl-dev
Provides: libsdl-dev
Depends: libsdl1.2debian (= 1.2.15-5), libasound2-dev,
libcaca-dev, libdirectfb-dev (>= 0.9.22), libglu1-me-
sa-dev | libglu-dev, libpulse-dev, libx11-dev, libts-
dev, libxext-dev
Conflicts: libsdl-dev
Description-en: Simple DirectMedia Layer development
files
SDL is a library that allows programs portable low
level access to a video
framebuffer, audio output, mouse, and keyboard.
```

This package contains the files needed to compile and link programs which use SDL.

```
Homepage: http://www.libsdl.org/
Description-md5: 9a82f59c5790721baad7ffc5f181d3d6
Tag: devel::library, role::devel-lib, uitoolkit::sdl
Section: libdevel
Priority: optional
Filename: pool/main/libs/libsdl1.2/libsdl1.2-
dev_1.2.15-5_armhf.deb
Size: 861890
MD5sum: 4295708cab85d1eb546b449350dd2da6
SHA1: 76fa9923c9765d7b92e373df6fe12949f2092db5
SHA256: 71def7638b06e6711f6fa8d96724aa7eb238a7b10f9fb-
14192b5a5c1018d1322
```

Libsdl1.2debian es la librería exacta que contiene los ficheros binarios necesarios para ejecutar libsdl. Como ya se ha dicho, el archivo `-dev` sólo incluye las cabeceras, y es inútil si sólo quieres EJECUTAR el programa. Sin embargo, si no estás seguro de las librerías que debes añadir en `checkinstall`, puede agregar igualmente las librerías `-dev`. Este método no es tan limpio pero funciona, ya que el programa siempre contará con la librería que necesita.

Algunas notas especiales sobre checkinstall

Puesto que `checkinstall` copia los archivos a una ubicación, lo más probable es que tengan que ejecutarlo como `root`, ya que un usuario normal no tiene permisos para copiar archivos en determinadas carpetas y por tanto el desarrollo fallará.

A veces, `checkinstall` falla a pesar de tener privilegios de `root`, y hay veces que parece ser un problema la propia creación de carpetas cuando usamos `checkinstall`. Si un programa que has instalado requiere la creación de una carpeta, `checkinstall` puede fallar al no poder crear la carpeta para copiar los archivos. Para solucionar esto, ejecuta primero `make install` y luego ejecuta `checkinstall` de nuevo.

`Checkinstall` es muy útil, ya que desecha todos los archivos binarios innecesarios, como comentarios y símbolos de depuración que ya no son necesarios tras completar el desarrollo del programa, lo cual puede reducir en gran medida el tamaño de archivos.



Algunos de los peores monstruos de DOOM venian directamente de mis peores pesadillas. Un demonio araña, un cerebro robot de 4 patas con una ametralladora.

comentarios innecesarios sobre las librerías presentes en el directorio `/lib`, el tamaño se reduce de 307 a 18 MB y todavía tendrían la misma funcionalidad que cuando ocupaban 307 MB. Como resultado, el tamaño de la instalación del kernel se reduce de 354 MB a 65 MB con `checkinstall`. Dependiendo de los paquetes de software que crees, `checkinstall` puede ayudarte a mantener tu software elegante y limpio (y pequeño).

`Checkinstall` no siempre funciona. Algunos proyectos no permiten la función `make install` y `checkinstall` no funcionará para esas aplicaciones. Pero no te preocupes, ya que cuentas con cierta experiencia en la compilación, es probable que encuentres la forma de crear tus propias instalaciones.

Desarrollar un proyecto mas amplio

Para DOOM, compilar y empaquetar es muy sencillo, y sólo tuvimos que descargar un archivo en lugar de diez, cientos o miles de archivos, y sólo había una dependencia.

Como ejemplo de proyecto más amplio, he incluido indicaciones para compilar `OpenXCom`. Para acceder a la página de inicio de `OpenXcom`, haga clic en el siguiente enlace: <http://bit.ly/1rWH0ml>

Compilando OpenXCom

`OpenXcom` necesita las siguientes librerías:

SDL (libsdl1.2)
SDL_mixer (libsdl-mixer1.2)
SDL_image (libsdl-image1.2)
SDL_gfx (libsdl-gfx1.2), version 2.0.22 o superior
yaml-cpp, (libyaml-cpp), version 0.5 o superior

Genial, todo lo que necesitamos está justotamente aquí, sobre todo por la excelente documentación de `OpenXcom`. Ahora resulta muy sencillo instalar lo que necesitamos.

```
apt-get install libsdl1.2-dev libsdl-mixer1.2-dev
libsdl-image1.2-dev libsdl-gfx1.2-dev
```

Tamaño del binario doom antes de la depuración:

```
$ ls -lh doom
-rwxr-xr-x 1 root root 423K Apr  5 11:33 doom
```

Tamaño de la librería doom en el archivo .deb:

```
$ ls -lh doom
-rwxr-xr-x 1 root root 368K Apr  5 15:16 doom
```

El tamaño de DOOM no debería ser muy grande, pero si los son otros programas. Por ejemplo, los archivos del kernel facilitado por `Hardkernel` ocupan un total de 354MB sin comprimir.

```
$ du -h --max-depth=1
45M  ./usr
307M ./lib
2.5M ./boot
354M .
```

Y los mismos archivos del Kernel tras la depuración:

```
$ du -h --max-depth=1
45M  ./usr
18M  ./lib
2.5M ./boot
65M  .
```

Como podemos ver, los directorios `/usr` y `/boot` no han cambiado, ya que sólo tienen archivos cabecera o archivos que no pueden ser suprimidos, pero si quitamos los símbolos y

No tuve en cuenta yaml-cpp ya que si compilas en Ubuntu, la instalación de libyaml-cpp-dev te dará la versión 0.5 o una superior, pero para Debian Wheezy sólo conseguirás la versión 0.3 de libyaml-cpp que no funciona.

Para salvar esto, he compilado libyaml-cpp que puedes instalar escribiendo en el terminal:

```
$ wget http://oph.mdrjr.net/meveric/repository/libyaml-cpp0.5_0.5.1-1_armhf.deb
$ dpkg -i libyaml-cpp0.5_0.5.1-1_armhf.deb
```

Ahora que tenemos todos los requisitos, podemos comenzar a descargar las fuentes. En la parte superior de la Wiki OpenXCom, hay un enlace a un repositorio, que nos lleva a una página que contiene un montón de carpetas y archivos. Necesitaríamos un monton de tiempo si intentamos descargarlos todos con wget, pero por suerte hay una alternativa más rápida.

Desde el repositorio git, podemos descargar todos los archivos con un solo comando. En primer lugar, hazte con la dirección URL del sitio del proyecto:

```
$ git clone https://github.com/SupSuper/OpenXcom.git
Cloning into 'OpenXcom'...
remote: Reusing existing pack: 40755, done.
remote: Counting objects: 124, done.
remote: Compressing objects: 100% (118/118), done.
remote: Total 40879 (delta 67), reused 0 (delta 0)
Receiving objects: 100% (40879/40879), 14.05 MiB | 849 KiB/s, done.
Resolving deltas: 100% (33534/33534), done.
```

¡Ha sido rápido, eh!, nos acabamos de descargar cerca de 700 archivos en pocos segundos. Para otros proyectos que usen subversión en lugar de git, la línea de comandos es similar: **svn checkout <url>**. Lista de los nuevos archivos:

```
$ ls -l
total 152
-rwxr-xr-x 1 root root 82 Apr 5 17:12 autogen.sh
drwxr-xr-x 3 root root 4096 Apr 5 17:12 bin
-rw-r--r-- 1 root root 32796 Apr 5 17:12 CHANGELOG.txt
drwxr-xr-x 3 root root 4096 Apr 5 17:12 cmake
-rw-r--r-- 1 root root 5681 Apr 5 17:12 CMakeLists.txt
-rw-r--r-- 1 root root 3385 Apr 5 17:12 configure.ac
-rw-r--r-- 1 root root 35819 Apr 5 17:12 COPYING
drwxr-xr-x 2 root root 4096 Apr 5 17:12 docs
drwxr-xr-x 6 root root 4096 Apr 5 17:12 install
drwxr-xr-x 2 root root 4096 Apr 5 17:12 m4
-rw-r--r-- 1 root root 21142 Apr 5 17:12 Makefile.am
drwxr-xr-x 2 root root 4096 Apr 5 17:12 obj
-rw-r--r-- 1 root root 4446 Apr 5 17:12 README.txt
drwxr-xr-x 5 root root 4096 Apr 5 17:12 res
drwxr-xr-x 12 root root 4096 Apr 5 17:12 src
```

Esta vez, no hay ni configure ni Makefile, sólo un Makefile.am. Entonces, ¿Qué podemos hacer?

Hay dos archivos en esta carpeta que te ayudarán: un ejecutable llamado autogen.sh y otro llamado CMakeList.txt.

Autogen

Algunos proyectos tienen un archivo con el nombre autogen, automake o autoconf en lugar de un archivo configure, en su directorio fuente.

```
$ ./autogen.sh
aclocal: installing `m4/pkg.m4' from `/usr/share/aclocal/pkg.m4'
autoreconf: Entering directory `.'
autoreconf: configure.ac: not using Gettext
autoreconf: running: aclocal --force -I m4
autoreconf: configure.ac: tracing
autoreconf: configure.ac: not using Libtool
autoreconf: running: /usr/bin/autoconf --force
autoreconf: configure.ac: not using Autoheader
autoreconf: running: automake --add-missing --force-missing
configure.ac:106: installing `./config.guess'
configure.ac:106: installing `./config.sub'
configure.ac:17: installing `./install-sh'
configure.ac:17: installing `./missing'
Makefile.am: installing `./depcomp'
autoreconf: Leaving directory `.'
```

El resultado es muy corto y no parece que haya finalizado. Comprobemos si se ha hecho algo.

```
$ ls -l
total 1204
-rw-r--r-- 1 root root 36830 Apr 5 17:23 aclocal.m4
-rwxr-xr-x 1 root root 82 Apr 5 17:12 autogen.sh
drwxr-xr-x 2 root root 4096 Apr 5 17:23 autom4te.cache
drwxr-xr-x 3 root root 4096 Apr 5 17:12 bin
-rw-r--r-- 1 root root 32796 Apr 5 17:12 CHANGELOG.txt
drwxr-xr-x 3 root root 4096 Apr 5 17:12 cmake
-rw-r--r-- 1 root root 5681 Apr 5 17:12 CMakeLists.txt
lrwxrwxrwx 1 root root 37 Apr 5 17:23 config.guess -> /usr/share/automake-1.11/config.guess
lrwxrwxrwx 1 root root 35 Apr 5 17:23 config.sub -> /usr/share/automake-1.11/config.sub
-rwxr-xr-x 1 root root 211749 Apr 5 17:23 configure
-rw-r--r-- 1 root root 3385 Apr 5 17:12 configure.ac
-rw-r--r-- 1 root root 35819 Apr 5 17:12 COPYING
lrwxrwxrwx 1 root root 32 Apr 5 17:23 depcomp -> /usr/share/automake-1.11/depcomp
drwxr-xr-x 2 root root 4096 Apr 5 17:23 docs
drwxr-xr-x 6 root root 4096 Apr 5 17:12 install
lrwxrwxrwx 1 root root 35 Apr 5 17:23 install-sh -> /usr/share/automake-1.11/install-sh
drwxr-xr-x 2 root root 4096 Apr 5 17:23 m4
-rw-r--r-- 1 root root 21142 Apr 5 17:12 Makefile.am
-rw-r--r-- 1 root root 822542 Apr 5 17:23 Makefile.in
lrwxrwxrwx 1 root root 32 Apr 5 17:23 missing -> /usr/share/automake-1.11/missing
drwxr-xr-x 2 root root 4096 Apr 5 17:12 obj
-rw-r--r-- 1 root root 4446 Apr 5 17:12 README.txt
drwxr-xr-x 5 root root 4096 Apr 5 17:12 res
drwxr-xr-x 12 root root 4096 Apr 5 17:19 src
```

Tras este paso, ahora existe un archivo configure, que ofrece el mismo punto de partida para la compilación como en el caso del primer ejemplo.

CMakeLists.txt

Cmake también es una herramienta muy amigable y si un proyecto tiene un CMakeLists.txt en su directorio, entonces es que soporta cmake. Podríamos continuar y escribir **cmake**, pero ¡Te aconsejo que no lo hagas!

Aunque cmake funciona y genera el código que necesitamos, es un tanto desorganizado. Además, no es posible diferenciar el código fuente del código que procede de cmake. Existe una forma mejor de hacerlo.

En su lugar, crea una nueva carpeta, cambia a ésta e inicia cmake desde aquí.

```
$ mkdir build
$ cd build
$ cmake ..
```

Cmake se inicia con una ruta de acceso al archivo CMakeLists.txt. En el ejemplo de DOOM, era “cmake.”. Ya que estábamos en el mismo directorio, usamos el “.” para decir a cmake que usara el directorio actual. Ahora que estamos en una subcarpeta llamada build, le decimos a cmake que CMakeLists.txt está una carpeta escribiendo “cmake.”. Con todo configurado correctamente, debemos ver algo similar a esto:

```
$ cmake ..
-- The C compiler identification is GNU 4.7.2
-- The CXX compiler identification is GNU 4.7.2
-- Check for working C compiler: /usr/bin/gcc-4.7
-- Check for working C compiler: /usr/bin/gcc-4.7 --
works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/g++-4.7
-- Check for working CXX compiler: /usr/bin/g++-4.7
-- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Looking for include file pthread.h
-- Looking for include file pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found PkgConfig: /usr/bin/pkg-config (found version
"0.26")
-- Looking for XOpenDisplay in /usr/lib/arm-linux-
gnueabi/libX11.so;/usr/lib/arm-linux-gnueabi/libX11.so
-- Looking for XOpenDisplay in /usr/lib/arm-linux-
gnueabi/libX11.so;/usr/lib/arm-linux-gnueabi/libX11.so - found
-- Looking for gethostbyname
-- Looking for gethostbyname - found
-- Looking for connect
-- Looking for connect - found
-- Looking for remove
-- Looking for remove - found
-- Looking for shmctl
-- Looking for shmctl - found
-- Looking for IceConnectionNumber in ICE
-- Looking for IceConnectionNumber in ICE - found
-- Found X11: /usr/lib/arm-linux-gnueabi/libX11.so
```

```
-- Found OpenGL: /usr/local/lib/libGL.so
found SDL 1.2.15 (/usr/lib/arm-linux-gnueabi/include/SDL)
found SDL_mixer 1.2.12 (/usr/lib/arm-linux-gnueabi-
include/SDL)
found SDL_gfx 2.0.23 (/usr/lib/arm-linux-gnueabi/hf:/usr/include/SDL)
found SDL_image 1.2.12 (/usr/lib/arm-linux-gnueabi-
hf:/usr/include/SDL)
found yamll-cpp(/usr/local/lib:/usr/local/include/
yamll-cpp;/usr/local/include/yamll-cpp/..)
-- Found Git: /usr/bin/git (found version "1.7.10.4")
git found: /usr/bin/git
version:0.9.g8bcafeaf
No doxygen command found. Disable API documentation
generation
-- Configuring done
-- Generating done
-- Build files have been written to: /home/odroid/
sources/OpenXcom/build
```

```
$ ls -l
total 68
drwxr-xr-x 2 root root 4096 Apr 5 17:37 bin
-rw-r--r-- 1 root root 29527 Apr 5 17:37 CMakeCache.
txt
drwxr-xr-x 4 root root 4096 Apr 5 17:37 CMakeFiles
-rw-r--r-- 1 root root 1906 Apr 5 17:37 cmake_in-
stall.cmake
-rw-r--r-- 1 root root 3526 Apr 5 17:37 CPackConfig.
cmake
-rw-r--r-- 1 root root 3942 Apr 5 17:37 CPackSource-
Config.cmake
drwxr-xr-x 3 root root 4096 Apr 5 17:37 docs
-rw-r--r-- 1 root root 6481 Apr 5 17:37 Makefile
drwxr-xr-x 3 root root 4096 Apr 5 17:37 src
```

Puesto que aquí obtenemos un Makefile, ejecutamos **make** y ¡Observamos cómo pasan los colores!

```
$ make
Scanning dependencies of target openxcom
[ 0%] Building CXX object src/CMakeFiles/openxcom.
dir/main.cpp.o
[ 0%] Building CXX object src/CMakeFiles/openxcom.
dir/lodepng.cpp.o
[ 1%] Building CXX object src/CMakeFiles/openxcom.
dir/Basescape/BaseView.cpp.o
[ 1%] Building CXX object src/CMakeFiles/openxcom.
dir/Basescape/CraftSoldiersState.cpp.o
[... ]
[ 99%] Building CXX object src/CMakeFiles/openxcom.
dir/Ufopaedia/ArticleStateTextImage.cpp.o
[ 99%] Building CXX object src/CMakeFiles/openxcom.
dir/Ufopaedia/ArticleStateArmor.cpp.o
[100%] Building CXX object src/CMakeFiles/openxcom.
dir/Ufopaedia/ArticleStateVehicle.cpp.o
Linking CXX executable ../bin/openxcom
[100%] Built target openxcom
```

Todo tiene buen aspecto, y contamos con un desarrollo de OpenXCom en la carpeta bin. Ahora, ejecuta de nuevo checkinstall. Ya conocemos que librerías debemos añadir puesto que cogimos la lista del sitio de OpenXCom.

```
$ checkinstall --backup=no --install=no
--requires="libdl1.2debian, libsdl-image1.2, lib-
sdl-mixer1.2, libsdl-gfx1.2-4, libyamll-cpp0.5"
--pkgname="openxcom" --pkgversion="0.9.g8bcafeaf"
```

Este es uno de los momentos en los que checkinstall fa-

llará por el problema con la creación de directorios, así que ejecuta primero **make install** y luego vuelve a ejecutar **checkinstall**.

```
$ checkinstall --backup=no --install=no
--requires="libdl1.2debian, libsdl-image1.2, lib-
sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5"
--pkgname="openxcom" --pkgversion="0.9.g8bcafeaf"
```

```
checkinstall 1.6.2, Copyright 2009 Felipe Eduardo San-
chez Diaz Duran
This software is released under the GNU GPL.
```

```
The package documentation directory ./doc-pak does not
exist.
Should I create a default set of package docs? [y]: y
```

```
Please write a description for the package.
End your description with an empty line or EOF.
>> OpenXCom
>>
```

```
*****
**** Debian package creation selected ****
*****
```

This package will be built according to these values:

```
0 - Maintainer: [ root@odroid-wheezy ]
1 - Summary: [ OpenXCom ]
2 - Name: [ openxcom ]
3 - Version: [ 0.9.g8bcafeaf ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ armhf ]
8 - Source location: [ build ]
9 - Alternate source location: [ ]
10 - Requires: [ libdl1.2debian, libsdl-image1.2, lib-
sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5 ]
11 - Provides: [ openxcom ]
12 - Conflicts: [ ]
13 - Replaces: [ ]
```

Enter a number to change any of them or press ENTER to continue:

Installing with make install...

```
===== Installation results
=====
[100%] Built target openxcom
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/bin/openxcom
-- Removed runtime path from "/usr/local/bin/openxcom"
-- Installing: /usr/local/share/openxcom/data
-- Installing: /usr/local/share/openxcom/data/Re-
sources
-- Installing: /usr/local/share/openxcom/data/Re-
sources/UI
-- Installing: /usr/local/share/openxcom/data/Re-
sources/UI/reserve.png
-- Installing: /usr/local/share/openxcom/data/Re-
sources/Weapons
-- Installing: /usr/local/share/openxcom/data/Re-
sources/Weapons/Terror.png
-- Installing: /usr/local/share/openxcom/data/Re-
sources/Pathfinding
-- Installing: /usr/local/share/openxcom/data/Re-
sources/Pathfinding/Pathfinding.png
-- Installing: /usr/local/share/openxcom/data/Re-
sources/BulletSprites
-- Installing: /usr/local/share/openxcom/data/Re-
sources/BulletSprites/BulletSprites.png
```

```
-- Installing: /usr/local/share/openxcom/data/Re-
sources/BulletSprites/TFTD-LAND.png
-- Installing: /usr/local/share/openxcom/data/Re-
sources/BulletSprites/TFTD-UNDERWATER.png
-- Installing: /usr/local/share/openxcom/data/Shaders
-- Installing: /usr/local/share/openxcom/data/Shad-
ers/Phosphor-simple.OpenGL.shader
-- Installing: /usr/local/share/openxcom/data/Shad-
ers/Openxcom.OpenGL.shader
CMake Error at src/cmake_install.cmake:54 (FILE):
file INSTALL cannot find
"/home/odroid/sources/test/OpenXcom/bin/data/Shaders/
heavybloom.OpenGL.shader".
Call Stack (most recent call first):
cmake_install.cmake:38 (INCLUDE)
```

```
make: *** [install] Error 1

**** Installation failed. Aborting package creation.

Cleaning up...OK

Bye.
```

```
$ make install
[100%] Built target openxcom
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/bin/openxcom
-- Removed runtime path from "/usr/local/bin/openxcom"
-- Installing: /usr/local/share/openxcom/data
-- Installing: /usr/local/share/openxcom/data/Re-
sources
...
-- Installing: /usr/local/share/openxcom/data/Ruleset
-- Installing: /usr/local/share/openxcom/data/Rule-
set/Xcom1Ruleset.rul
```

```
$ checkinstall --backup=no --install=no
--requires="libdl1.2debian, libsdl-image1.2, lib-
sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5"
--pkgname="openxcom" --pkgversion="0.9.g8bcafeaf"
```

```
checkinstall 1.6.2, Copyright 2009 Felipe Eduardo San-
chez Diaz Duran
This software is released under the GNU GPL.
```

```
*****
**** Debian package creation selected ****
*****
```

This package will be built according to these values:

```
0 - Maintainer: [ root@odroid-wheezy ]
1 - Summary: [ OpenXCom ]
2 - Name: [ openxcom ]
3 - Version: [ 0.9.g8bcafeaf ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ armhf ]
8 - Source location: [ build ]
9 - Alternate source location: [ ]
10 - Requires: [ libdl1.2debian, libsdl-image1.2, lib-
sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5 ]
11 - Provides: [ openxcom ]
12 - Conflicts: [ ]
13 - Replaces: [ ]
```

Enter a number to change any of them or press ENTER to continue:

```
Installing with make install...

===== Installation results
=====
[100%] Built target openxcom
Install the project...
-- Install configuration: ""
-- Up-to-date: /usr/local/bin/openxcom
-- Installing: /usr/local/share/openxcom/data
-- Installing: /usr/local/share/openxcom/data/Resources
...
-- Installing: /usr/local/share/openxcom/data/Ruleset
-- Up-to-date: /usr/local/share/openxcom/data/Ruleset/Xcom1Ruleset.rul

===== Installation successful
=====

Copying documentation directory...
./
./docs/
./docs/Makefile
./docs/CMakeFiles/
./docs/CMakeFiles/CMakeDirectoryInformation.cmake
./docs/CMakeFiles/progress.marks
./docs/cmake_install.cmake

Some of the files created by the installation are inside the home directory: /home

You probably don't want them to be included in the package.
Do you want me to list them? [n]: n
Should I exclude them from the package? (Saying yes is a good idea) [n]: y

Copying files to the temporary directory...OK
Stripping ELF binaries and libraries...OK
Compressing man pages...OK
Building file list...OK
Building Debian package...OK
NOTE: The package will not be installed
Erasing temporary files...OK
Deleting temp dir...OK

*****
*****
Done. The new package has been saved to

/home/odroid/sources/test/OpenXcom/build/openxcom_0.9.g8bcafea-1_armhf.deb
You can install it in your system anytime using:

dpkg -i openxcom_0.9.g8bcafea-1_armhf.deb
*****
*****
```

Finalizado este paso, el paquete de instalación .deb de OpenXcom está listo para su uso. Esta información debería ser suficiente para empezar a compilar tus proyectos.

Notas sobre cmake

Hay algo que me gusta de cmake. Después de que éste haya creado el árbol de desarrollo de un proyecto, encuentras una carpeta con el nombre del archivo binario del proyecto. Por ej, en el proyecto OpenXcom, la carpeta se localiza en src/CMakeFiles/openxcom.dir/. En ésta, encontraras un archivo llamado link.txt que muestra cómo se ha creado el archivo binario final, así como las librerías que están vinculadas a éste.



El Cyber Demon tiene una descripción aterradora en el manual original de DOOM: “un rascacielos que lanza misiles con patas de cabra.”

```
$ cat src/CMakeFiles/openxcom.dir/link.txt
[...]
-o ../bin/openxcom -rdynamic -lSDL_image -lSDL_mixer
-lSDL_gfx -Wl,-Bstatic -lSDLmain -Wl,-Bdynamic -lSDL
-lpthread /usr/local/lib/libyaml-cpp.so /usr/local/lib/libGL.so -Wl,-rpath,/usr/local/lib:
```

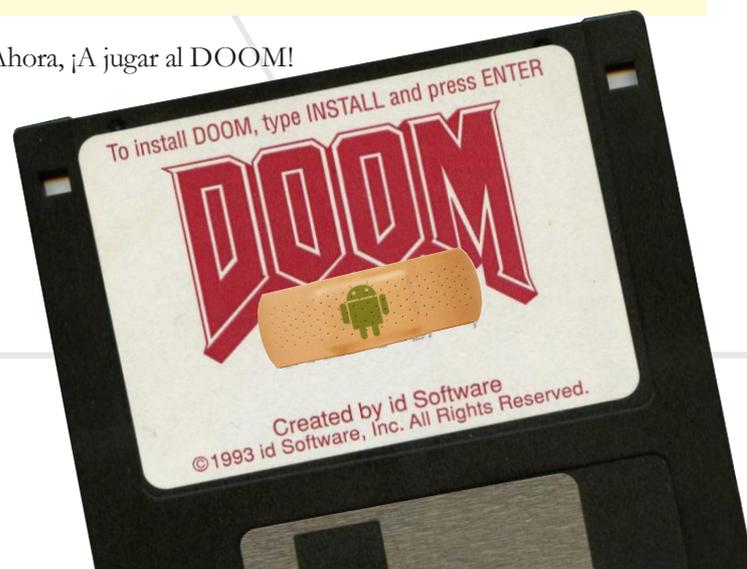
Si compruebas la última línea, verás las librerías vinculadas al archivo binario, por ejemplo, -lSDL_image -lSDL_mixer -lSDL_gfx -lSDL. Como ya se ha mencionado en un artículo anterior, los archivos que empiezan por “-l” son librerías. Estas son exactamente las librerías que usamos con el comando checkinstall, y que están enumeradas en la web OpenXCom. Por lo tanto, en caso de no sabes qué librerías necesitas, también se puede localizar aquí.

Comprobando dependencias

Con el comando ldd, se puede comprobar fácilmente las dependencias de un archivo binario que ya ha sido compilado.

```
$ ldd /usr/local/bin/doom
libm.so.6 => /lib/arm-linux-gnueabi/libm.so.6 (0xb6e80000)
libSDL-1.2.so.0 => /usr/lib/arm-linux-gnueabi/libSDL-1.2.so.0 (0xb6e03000)
libc.so.6 => /lib/arm-linux-gnueabi/libc.so.6 (0xb6d1e000)
/lib/ld-linux-armhf.so.3 (0xb6f08000)
[...]
```

Ahora, ¡A jugar al DOOM!



DESCRIBIENDO LA FUNCION MATEMATICA ATAN2

UNA UTIL HERRAMIENTA PARA PROGRAMAR APLICACIONES QUE REQUIERAN TRIGONOMETRIA EN TIEMPO REAL

por Jussi Opas

Muchos de nosotros hemos aplicado funciones trigonométricas como $\sin(x)$, $\cos(x)$ o $\tan(x)$ y sus funciones inversas \arcsin , \arccos , \arctan para hacer cálculos geométricos. Esto es lo que nos han enseñado en las matemáticas de la escuela. Cuando las utilizamos nos encontramos con problemas como que, el intervalo de valor del \arcsin es $[-\pi/2, \pi/2]$, y que existe la posibilidad de dividir por cero cuando se usa la $\tan(y/x)$. Para resolver estos problemas, los programas deben dividirse en sentencias "if" y los problemas de divisiones por cero también deben ser tratados. Hay un excelente método para resolver todos estos problemas dentro de los programas informáticos mediante la aplicación del método $\text{atan2}(y,x)$.

La función atan2 fue originalmente introducida en el lenguaje de programación FORTRAN para facilitar los cálculos trigonométricos [WIK14]. También se encuentra en otros sitios, como en C, Java y (sorpresa, sorpresa) en .NET y Excel, también. La ventaja de atan2 es que ofrece valores continuos en el intervalo de $(-\pi, \pi]$. Cuando se utiliza la función atan , el planteamiento es el siguiente:

```
atan2(y, x) = atan(y/x), if x > 0
atan2(y, x) = atan(y/x) + pi, if x < 0, y > 0
atan2(y, x) = atan(y/x) - pi, if x < 0, y < 0
atan2(y, x) = 0, if x = 0, y > 0
atan2(y, x) = pi/2, if x = 0, y < 0
atan2(y, x) = undefined, if x = 0, y = 0
```

Cuando miramos la fórmula, la conclusión es que se trata de una función bastante compleja, con muchas derivaciones para implementar la funcionalidad de atan2 . Lo mismo ocurre si tratamos de usar \arcsin o \arccos . Un problema de esta definición es que no hay un valor definido cuando tanto x como y son cero. Otra cuestión es que el orden de los parámetros es y, x en

Reloj

lugar de x, y . Aunque, en orden inverso en Excel.



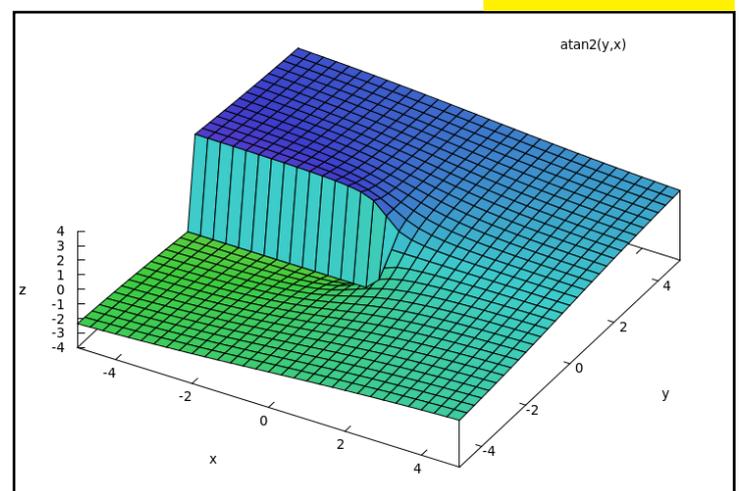
En un reloj interactivo, podemos calcular, a partir de la ubicación del cursor, el ángulo del puntero seleccionado para determinar cómo se puede ajustar el tiempo seleccionado. Por ejemplo, se puede calcular el ángulo desde el norte al cursor de la siguiente forma:

```
double angleAsRadians = Math.atan2(point.y - centerY,
point.x - centerX);
// atan2 returns values in the range of [-PI, PI]
// so, we can get values into the ranges [0, 2PI] by
adding one PI
angleAsRadians += PI;
```

Es tan asombroso que no se necesitan expresiones "if". Sin embargo, esto no es tan preciso como un juego de disparos, donde el ángulo y la distancia al objetivo son calculados continuamente. La Trigonometría se utiliza bastante en aplicaciones de trazado de rayos, tales como la cobertura de radio y modelado en 3D, donde atan2 puede aplicarse correctamente. De modo que, atan2 es un monstruo en su expresividad, con seguridad podemos considerarla como la mejor función trigonométrica dentro del kit de desarrollo de un programador.

El funcionamiento de atan2 se puede mostrar fácilmente con algún software de matemáticas como Octave o Maxima. Ambos disponibles en el Centro de Software de Ubuntu. Android tiene sus propias versiones en PlayStore.

¡Atan2 es hermoso!



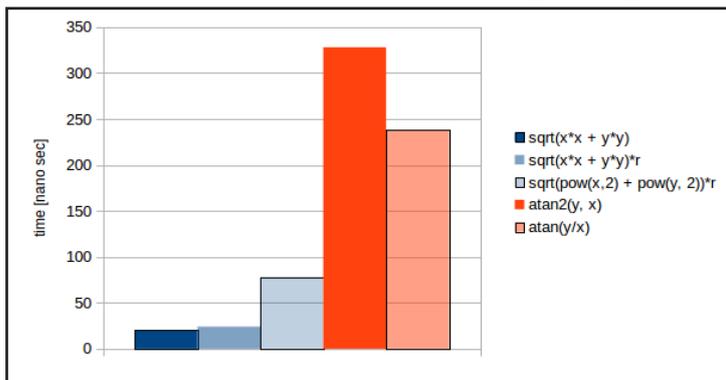
En la ventana gráfica, se puede rotar el ratón hasta el ángulo de visión apropiado y ver una vista en 3 dimensiones.

El desarrollo de software está motivado por casos de uso e historias de usuarios que se hacen realidad. Los proyectos

están, a menudo muy ocupados poniendo en práctica todas sus funciones y las pruebas de rendimiento sólo se realizan al final del proyecto. En ese momento, puede ser incluso demasiado tarde para reaccionar a un mal funcionamiento, resulta difícil mejorar en la última fase de proyecto. Por lo tanto, sería bueno conocer cuál va a ser la implicación de su funcionamiento en el momento de escribir el programa.

Si una aplicación utiliza fórmulas matemáticas como cálculos angulares y de distancia, sería bueno conocer el coste de activar estos métodos. En Java, por ejemplo, llamados Math.sqrt, Math.atan o Math.atan2 son muy importantes para entenderlo.

Hemos realizado una prueba ejecutando diferentes métodos de cálculo angular y de distancia 10 millones de veces con Java. El entorno de trabajo de pruebas añade el método overhead a cada función usando un ODROID XU a 1,6 GHz ejecutando Ubuntu con Java. El funcionamiento de los métodos matemáticos probados es similar en otras plataformas. Los resultados se muestran en el siguiente gráfico.



Calculando distancia y angulos

El cálculo de la distancia es rápido, realizado con sqrt (x * x + y * y) * r tarda sólo 24 nano segundos de media. La multiplicación con resolución r es casi libre, con sqrt(x*x, y*y) es sólo 4 nanosegundos más rápido. Entonces, se ve claramente que pow(x, 2) es esencialmente más costosa que x * x. Observamos que los métodos de cálculo de ángulos son lentos, especialmente en funciones trigonométricas con Java, que es una ventaja y un inconveniente al mismo tiempo. En este caso, atan2 parece ser más lenta que atan.

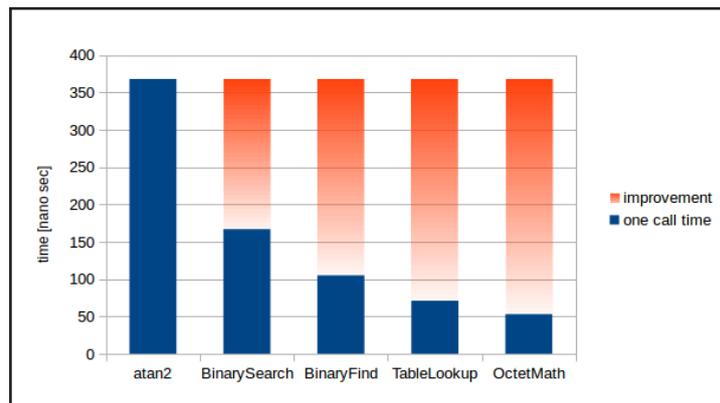
Si queremos usar el método atan2, debemos buscar algún modo de realizar el cálculo más rápido. Podríamos implementar más rápido pero con menos precisión el método para calcular atan2. Podríamos utilizar aproximación parabólica, series de Taylor o series de Chebyshev o similares [COR09], o una implementación de software del famoso método CORDIC. Por otra parte, se podría aplicar algún método de tabla de búsqueda. En primer lugar, la utilización de los anteriores valores calculados podrían ser los más fácil de implementar mediante la creación de los siguientes métodos de búsqueda para el acceso rápido de valores ATAN2:

- Búsqueda binaria con el método Arrays.binarySearch de Java
- Búsqueda binaria es un método de auto implementación para encontrar el valor objeto de la matriz.
- Búsqueda en la tabla con un gran tabla n*n
- Octeto matemático, otro método de tabla de búsqueda donde cada octeto simétrico se representa como matriz, [RIV06].

La siguiente figura muestra el tiempo de ejecución de atan2 frente a los métodos de búsqueda usados. En la ejecución representativa el cálculo necesito 368 nanosegundos. Los métodos de búsqueda son comparados con ese valor. La ejecución de un método de acceso se muestra con barras azules y la mejora lograda se muestra con barras naranja degradadas

Es una sorpresa que una búsqueda binaria auto escrita sea más rápida que la búsqueda binaria basada en un método estándar de Java. Debemos desconfiar de esto y sería necesario verificar el resultado antes de usar un método escrito con código de producción.

Los dos métodos de búsqueda basados en matriz, tabla de búsqueda y octeto de matemáticas son los más rápidos. Usando octeto matemáticas, un valor atan2 se puede completar en aproximadamente 50 nanosegundos. Ese es el método que hemos utilizado en nuestra aplicación de ejemplo.



Metodos de busqueda Atan2

Este es el código fuente para aquellos que están interesados en la implementación real de Java:

```

/**
 * Octet based atan2 lookup method.
 * Created by xyz on 12/13/13.
 */
public class OctetMath {
    private static final double ROUND = 0.5;
    private static final double PI = Math.PI;
    // To be accurate enough, at least 2000 values are
    // needed, we use exponent of two in here.
    // The memory consumption is 2048 * 8 * 4 = 65536
    // bytes ~ 65.5 kBytes.
    // Using of bigger size will increase accuracy.
    private static final int SIZE = 2048;
    // last index per side
    private static final float LAST = SIZE - 1;
    private static final float NEGATIVE_LAST = -LAST;
    // Octets are illustrated below.
    //      ^
    //      |
    //      8 | 1st octet
    //      7 | 2nd octet

```

```

//      -----> x
//      6 | 3
//      5 | 4
//      |
// Array representations of strictly computed atan2
values.
// Using double would not make the result more ac-
curate.
private static final float[] OCTET1 = new float[SIZE];
private static final float[] OCTET2 = new float[SIZE];
private static final float[] OCTET3 = new float[SIZE];
private static final float[] OCTET4 = new float[SIZE];
private static final float[] OCTET5 = new float[SIZE];
private static final float[] OCTET6 = new float[SIZE];
private static final float[] OCTET7 = new float[SIZE];
private static final float[] OCTET8 = new float[SIZE];

static {
    initializeOctets(SIZE);
}

/**
 * Get atan2 as table lookup.
 * The complexity of the method is 14, but we do
not want this method to be split.
 * @param y
 * @param x
 * @return angle as radians, the value is in the
interval [-PI, PI]
 */
@SuppressWarnings("all")
public static float atan2(final float y, final
float x) {
    if (0 <= x) {
        if (0 <= y) {
            if (x < y) {
                return OCTET1[(int) (LAST * x / y
+ ROUND)];
            } else {
                return OCTET2[(int) (LAST * y / x
+ ROUND)];
            }
        } else {
            if (-y < x) {
                return OCTET3[(int) (NEGATIVE_LAST
* y / x + ROUND)];
            } else {
                return OCTET4[(int) (NEGATIVE_LAST
* x / y + ROUND)];
            }
        }
    } else {
        if (y < 0) {
            // both x and y are negative
            if (x > y) {
                return OCTET5[(int) (LAST * x / y
+ ROUND)];
            } else {
                return OCTET6[(int) (LAST * y / x
+ ROUND)];
            }
        } else {
            if (y < -x) {
                return OCTET7[(int) (NEGATIVE_LAST
* y / x + ROUND)];
            } else {
                return OCTET8[(int) (NEGATIVE_LAST
* x / y + ROUND)];
            }
        }
    }
}

private static void initializeOctets(final double
size) {
    final double y = size;
    for (double i = 0; i < size; i++) {

```

```

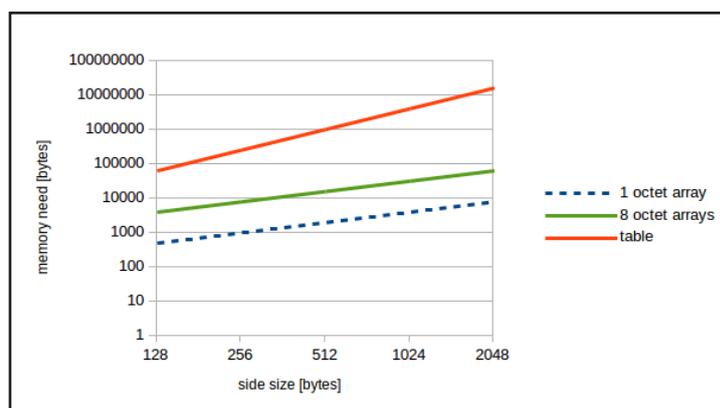
final double x = i;
final double theta = Math.atan2(y, x);
final int index = (int) i;
OCTET1[index] = (float) theta;
OCTET2[index] = (float) (PI / 2.0 - the-
ta);
OCTET3[index] = (float) (-PI / 2.0 + the-
ta);
OCTET4[index] = (float) -theta;
OCTET5[index] = (float) (-PI + theta);
OCTET6[index] = (float) (-PI / 2.0 - the-
ta);
OCTET7[index] = (float) (PI / 2.0 + the-
ta);
OCTET8[index] = (float) (PI - theta);
    }
}

private OctetMath() {
    //
}
}

```

Siempre debemos verificar las implementaciones hechas por nosotros mismos, puesto que la fiabilidad del método es muy importante. La fiabilidad e integridad alcanzada se puede probar frente al método Java Math.atan2. Es posible mejorar la fiabilidad usando matrices más grandes para representar los valores atan2 calculados anteriores. La desventaja de las matrices más grandes es, por supuesto, que se necesita asignar más memoria. El consumo de memoria de los tres esquemas de memoria se muestra en escala logarítmica en la siguiente figura.

El esquema de tabla es el que consume más memoria. Lo peor de todo es que su funcionamiento es exponencial. Como los 8 octetos de un círculo de unidad son simétricos, el mismo



Tamaño de memoria y secuencia

método puede ser implementado almacenando sólo los datos de uno de los octetos, en cuyo caso sólo es necesario 1/8 de la cantidad de memoria. Entonces, el tiempo de ejecución es mayor, ya que las ramas de más control de flujo deben ser atravesadas durante una ejecución.

Resumen

Con ODDROID, se puede hacer estudios matemáticos con herramientas específicas, mediante la implementación de tu

propia aplicación y estudiando su funcionamiento. Recomendamos la aplicación de atan2 en programas informáticos, especialmente en programas que utilizan una gran cantidad de cálculos de ángulos y de distancias tales como juegos y trazado de rayos. Los métodos trigonométricos de Java son lentos. Por lo tanto se debe contar con métodos más rápidos. Hemos implementado un método de búsqueda en tabla basado en octeto con el fin de acceder a los valores ATAN2 de forma más rápida. Nuestra implementación de OctetMath.atan2 es 7 veces más rápida que el Math.atan2 estándar. Asigna 66 kB de memoria estática, que es asumible en términos de recursos.

Referencias

[WIK14] atan2. 8 pages, 2014. <http://en.wikipedia.org/wiki/Atan2>

[COR09] Coranac / Vijn Jasper. Another fast fixed-point sine approximation. 16.7.2009. www.coranac.com/2009/07/sines

[RIV06] Riven X. 13.8x faster atan2. 2006. <http://www.java-gaming.org/topics/13-8x-faster-atan2-updated/14647/view.html>

FRAMEBUFFER CONSOLA DE TERMINAL PARA ESOS MOMENTOS SIN GUI

by Rob Roy

Cuando ejecutamos una distribución Ubuntu usando el Kernel 3.8.13 o superior en un ODROID-X/X2/U2/U3, hay una consola de terminal framebuffer residente que puede activarse pulsando Control+Alt+F [1-6]. Para volver a la interfaz gráfica X11, pulse Control+Alt+F7. Esto le dará acceso a la línea de comandos, incluso cuando la interfaz X11 deje de funcionar.

INSTALANDO HERRAMIENTAS MATEMATICAS DESDE EL CENTRO DE SOFTWARE UBUNTU

CREA BONITOS GRÁFICOS EN 3D PARA TU OFICINA E IMPRESIONA A TUS COLEGAS

por Jussi Opas

El Centro de Software de Ubuntu contiene muchas aplicaciones gratuitas y de bajo coste, incluyendo programas científicos y matemáticos. No todos ellos están disponibles para los ordenadores basados en procesadores ARM, pero muchos de ellos se pueden ejecutar en ODROID y el Centro de Software te asegura que puedes cargar una versión compilada correctamente. En este artículo se describe cómo cargar una aplicación matemática desde la tienda de Ubuntu.

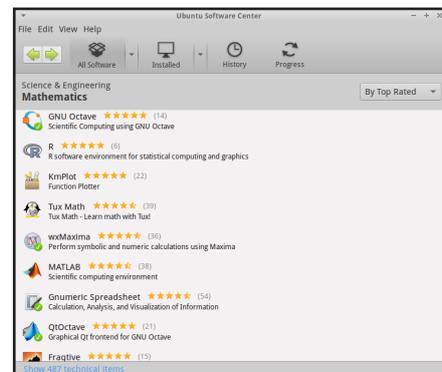
En la esquina superior derecha del Centro de Software, hay un cuadro de búsqueda que se puede usar si sabemos lo que estamos buscando. A la izquierda hay un listado de categorías, como se muestra en la imagen.

Para encontrar una herramienta matemática, presione el botón 'Science & Engineering'. A continuación, aparecerá una lista de las alternativas representadas por iconos.

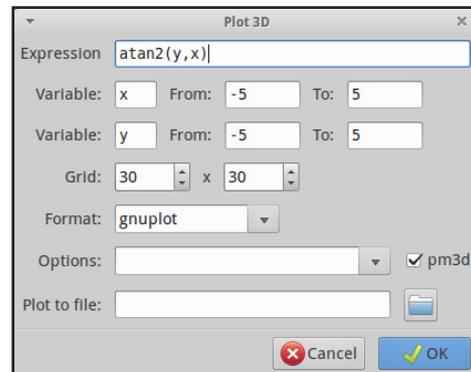


Ciencia y Ingeniería

Selecciona el icono "Mathematics", y la siguiente ventana mostrará una colección de herramientas relacionadas con las matemáticas. Es fácil de instalar una her-



Tienda de matemáticas de Ubuntu



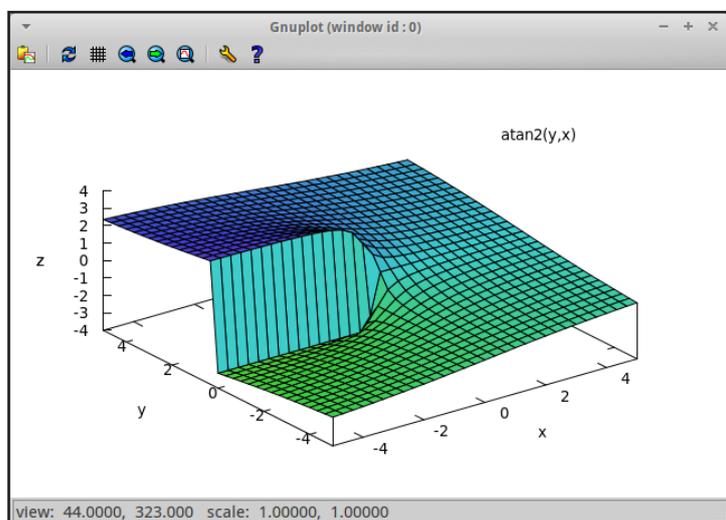
Dibujo en 3D

ramienta seleccionada. La desinstalación también es posible desde de esta interfaz.

Para empezar, te recomiendo cargar wxMaxima y/o QtOctave. Para demostrar que los programas funcionan, vamos a dibujar atan2 con Maxima y Octave.

Tras la instalación, wxMaxima aparecerá en el menú de Educación. Seleccione “Plot 3D...” en el menú Plot de wxMaxima para ver el cuadro de diálogo de dibujo.

A continuación, escriba su fórmula matemática favorita en el campo Expresión y rellena los valores de x e y.



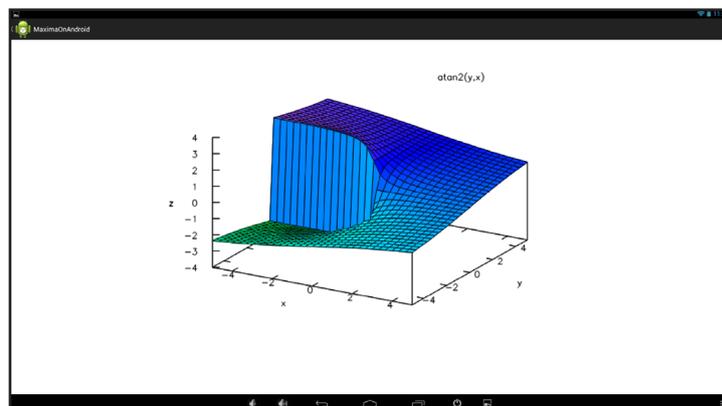
atan2 Maxima

En la ventana gráfica, rotando hacia un ángulo de visión adecuado verás la ilustración en 3D. Esto mismo se puede escribir también en la línea de comandos de la aplicación wxMaxima como sigue: `plot3d(atan2(y,x), [x,-5,5], [y,-5,5], [plot_format,gnuplot], [gnuplot_pm3d,true])`.

En QtOctave, también se puede definir una fórmula y a continuación, girar la imagen resultante con el ratón.

Las herramientas matemáticas también están disponibles para Android en Play Store. La versión de Maxima para An-

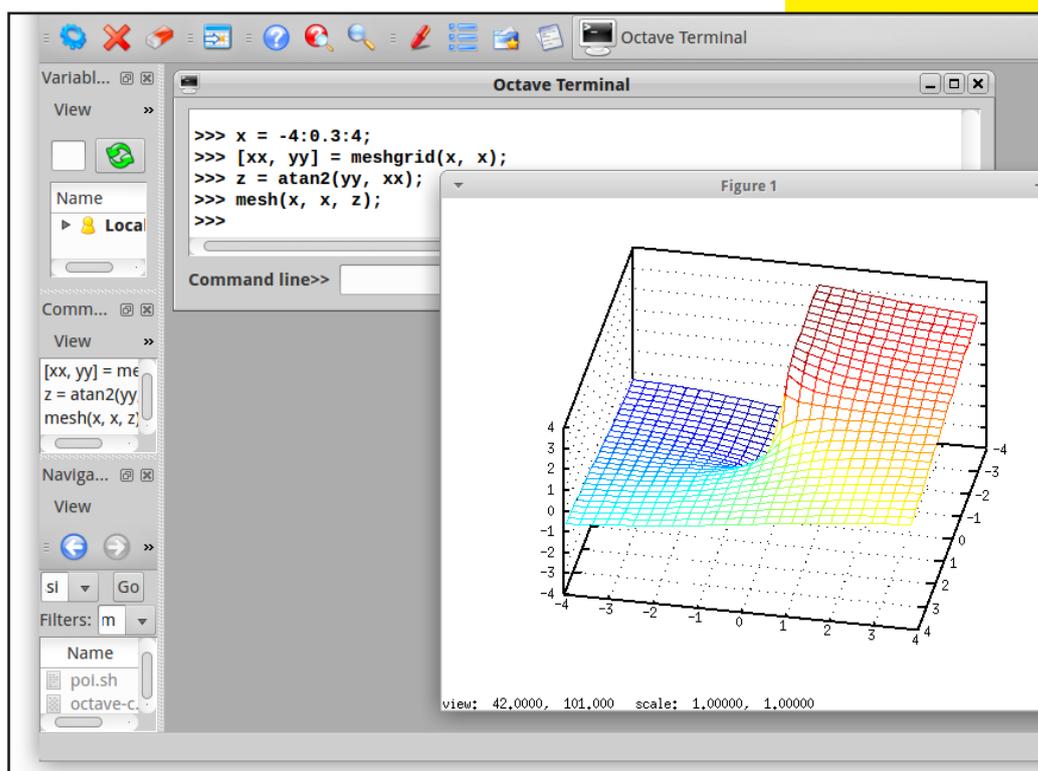
droid se muestra en la siguiente figura. Sin embargo, en Android no se puede girar la imagen, a diferencia de la versión para Linux de la aplicación Maxima.



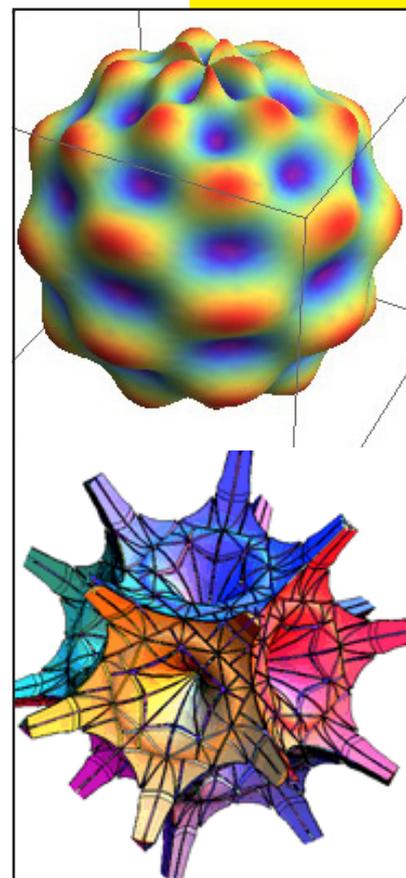
Atan2 ODROID

Puesto que Ubuntu es muy popular en la ejecución de aplicaciones científicas, hay muchos e interesantes programas de trazado disponibles para comprobar tus resultados experimentales, visualizar fórmulas matemáticas y crear gráficos complejos en 3D. También hay un grupo dedicado a la ingeniería y la ciencia en el sitio oficial de Ubuntu <http://bit.ly/1vV1VHX> donde se puede encontrar los programas de CAD, aplicaciones de electrónica, paquetes de visualización de datos y mucho más para que te diviertas con tu proyectos matemáticos con ODROID.

QTOctave Atan2



Matematica



ARCHIVOS IMAGEN DE ANDROID

UN VISTAZO A LOS ARCHIVOS COMPRIMIDOS QUE HACEN A ANDROID PORTATIL Y LIGERO

por Nanik Tolaram

El código fuente de Android es increíblemente enorme (más de 4GB), pero ¿Cómo se puede instalar esta extensa base de código en teléfonos, tabletas y en nuestras placas ODROID? La respuesta es simple, la fuente se compila en archivos binarios compactos y luego se comprime en un formato especial. En los dos últimos números, hemos analizado el código fuente de Android y cómo compilarlo para ODROID-U3. En el artículo de este mes, vamos a explorar los diferentes archivos de imagen generados por el sistema de desarrollo de Android. Todas las herramientas de software utilizadas en este artículo se pueden descargar desde <http://bit.ly/1hy3vdc>

Los dispositivos Android se ejecutan con recursos y espacio en disco muy limitados, lo que hace que sea fundamental que el sistema operativo y los archivos complementarios sean lo más pequeños posibles. Los archivos de imagen generados por el sistema de desarrollo son:

```
boot.img
cache.img
recovery.img
system.img
userdata.img
```

Hay situaciones en las que conocer bien los archivos de imagen tiene sus ventajas:

**Eliminar o añadir ficheros a la imagen.
Experimentar con diferentes configuraciones de forma rápida.
Actualizar las versiones de Android sin tener que compilar el código fuente por completo.**



boot.img

Este archivo de imagen contiene el archivo de inicialización (init) que se utiliza para arrancar Android. El fichero init.rc contiene información para alojar los diferentes servicios y configuraciones que se necesitan para iniciar Android. Hay 2 archivos dentro del directorio /sbin y uno en particular es muy importante llamado adbd, que es el demonio adb que se ejecuta como parte del proceso de arranque de Android. Adbd te permite conectarte a tu dispositivo mediante el comando adb.

Los siguientes pasos te muestran cómo extraer el boot.img:

1. Ejecuta la utilidad unmkbooting escribiendo

```
$ unmkbooting boot.img.
```

```
Kernel size 3133124
Kernel address 0x10008000
Ramdisk size 167690
Ramdisk address 0x11000000
Secondary size 0
Secondary address 0x10f00000
Kernel tags address 0x10000100
Flash page size 2048
Board name is ""
Command line ""
Extracting kernel.gz ...
Extracting initramfs.cpio.gz ...
All done.
```

```
-----
To recompile this image, use:
mkbooting --kernel kernel.gz --ramdisk initramfs.cpio.gz -o new_boot.img
-----
```

Un conjunto de archivos de los directorios se ubicarán en el directorio actual.

```
total 6460
drwxrwxr-x 2 nanik nanik 4096 May 10 22:09 .
drwxrwxr-x 14 nanik nanik 4096 May 10 21:53 ..
-rw-r--r-- 1 nanik nanik 3303424 May 10 22:09 boot.img
-rw-rw-r-- 1 nanik nanik 167690 May 10 22:09 initramfs.cpio.gz
Google Chrome nanik nanik 3133124 May 10 22:09 kernel.gz
```

Lista de archivos despues de extraer boot.img

2. Ejecuta el script extract_ulmage.sh escribiendo

```
$ extract_uImage.sh initramfs.cpio.gz.

Checking for uImage magic word...
1+0 records in
0+1 records out
4 bytes (4 B) copied, 3.9012e-05 s, 103 kB/s
uImage recognized.
Extracting data...
2619+1 records in
327+1 records out
167626 bytes (168 kB) copied, 0.00185786 s, 90.2 MB/s
Checking for ARM mach-type...
3+0 records in
0+1 records out
3 bytes (3 B) copied, 3.5863e-05 s, 83.7 kB/s
Checking for zImage...
1+0 records in
0+1 records out
4 bytes (4 B) copied, 2.6783e-05 s, 149 kB/s
>>> initramfs.cpio.gz extracted to Image
```

Los resultados con los archivos se muestran aquí:

```
total 6624
drwxrwxr-x 2 nanik nanik 4096 May 10 22:11 ./
drwxrwxr-x 14 nanik nanik 4096 May 10 21:53 ../
-rw-r--r-- 1 nanik nanik 3303424 May 10 22:09 boot.img
-rw-rw-r-- 1 nanik nanik 167626 May 10 22:11 Image.gz
-rw-rw-r-- 1 nanik nanik 167690 May 10 22:09 initramfs.cpio.gz
-rw-rw-r-- 1 nanik nanik 3133124 May 10 22:09 kernel.gz
```

Lista de archivos despues de extraer initramfs.cpio.gz

3. El archivo extraído es ahora un archivo gzip, así que tenemos que cambiarle el nombre.

```
$ mv Image Image.gz
```

4. Extrae el archivo recién renombrado usando gunzip.

```
$ gunzip Image.gz
```

Comprueba el tipo de archivo escribiendo

```
$ file Image
Image: ASCII cpio archive (SVR4 with no CRC).
```

5. Utiliza la herramienta de cpio para extraer el contenido

```
$ cpio -i < ./Image
563 blocks.
```

Ahora podrás ver el contenido completo de boot.img como se muestra a continuación. Archivos como boot.img se han eliminado de la captura de la pantalla para facilitar la visualización.

Lista completa de archivos de boot.img

system	0 items
(Empty)	
sys	0 items
(Empty)	
sbin	2 items
ueventd	105.2 kB
adbd	141.2 kB
proc	0 items
(Empty)	
dev	0 items
(Empty)	
data	0 items
(Empty)	
ueventd.rc	3.9 kB
ueventd.odroidu.rc	1.7 kB
ueventd.goldfish.rc	272 bytes
init.usb.rc	3.9 kB
init.trace.rc	1.6 kB
init.rc	17.2 kB
init.odroidu.usb.rc	2.9 kB
init.odroidu.rc	5.0 kB
init.goldfish.rc	2.3 kB
init	105.2 kB
default.prop	120 bytes

cache.img

Este archivo de imagen no contiene nada importante. Como su nombre indica, se utiliza como almacenamiento de caché. En un dispositivo Android, este archivo de imagen reside en su propia partición, o a veces en su propio directorio denominado /cache. El caché es utilizado por la aplicación OTA (Over The Air) de Android para almacenar archivos de imagen actualizados, o para actualizar tu sistema Android.

Los siguientes pasos muestran cómo extraer el cache.img:

1. Usa la herramienta simg2img para extraer el archivo de imagen.

```
$ simg2img cache.img cache.raw

computed crc32 of 0xc76ce614, expected 0x00000000
```

2. Utilizando el comando file cache.raw verás el tipo del archivo extraído.

```
$ file cache.raw
cache.raw: Linux rev 1.0 ext4 filesystem data,
UUID=57f8f4bc-abf4-655f-bf67-946fc0f9f25b (extents)
(large files)
```

3. Cree un directorio para montar el archivo extraído al directorio.

```
$ mkdir cc
$ sudo mount -t ext4 -o loop cache.raw cc/
```

4. Verás que no hay nada dentro del directorio /cc.

```
total 8
drwxr-xr-x 2 root root 4096 Jan 1 1970 .
drwxrwxr-x 3 nanik nanik 4096 May 10 22:26 ..
```

Lista completa de archivos cache.img

recovery.img

La imagen recovery contiene herramientas busybox y del kernel de Linux, y reside en su propia partición. En los dispositivos Android hay una secuencia de teclas en particular que tienes que pulsar para indicar al dispositivo que arranque desde la partición recovery. Hay otra forma de arrancar recovery usando el siguiente comando dentro de Android:

```
adb reboot recovery
```

Siga estos pasos para extraer el recovery.img:

I. Ejecuta la utilidad unmkbooting

```
$ unmkbooting recovery.img

Kernel size 3133124
Kernel address 0x10008000
Ramdisk size 1388735
Ramdisk address 0x11000000
Secondary size 0
Secondary address 0x10f00000
Kernel tags address 0x10000100
Flash page size 2048
Board name is ""
Command line ""
Extracting kernel.gz ...
Extracting initramfs.cpio.gz ...
All done.
-----
```

Para recompilar esta imagen, utiliza:

```
mkbooting --kernel kernel.gz --ramdisk initramfs.cpio.gz -o new_boot.img
-----
```

```
total 8848
drwxrwxr-x 2 nanik nanik 4096 May 10 22:34 ./
drwxrwxr-x 14 nanik nanik 4096 May 10 21:53 ../
-rw-rw-r-- 1 nanik nanik 1388735 May 10 22:34 initramfs.cpio.gz
-rw-rw-r-- 1 nanik nanik 3133124 May 10 22:34 kernel.gz
-rw-r--r-- 1 nanik nanik 4526080 May 10 22:34 recovery.img
```

Lista de archivos extraídos de recovery.img

Utiliza el comando **file** para comprobar del tipo de archivo `initramfs.cpio.gz`, verás lo siguiente

```
$ file
./initramfs.cpio.gz: u-boot legacy uImage, ramdisk, Linux/ARM, RAMDisk Image (Not compressed), 1388671 bytes, Wed Apr 2 10:35:12 2014, Load Address: 0x40800000, Entry Point: 0x40800000, Header CRC: 0x70CA98DA, Data CRC: 0xC14A4AFD
```

2. Ejecuta el script extract_ulmage.sh

```
$ extract_uImage.sh initramfs.cpio.gz

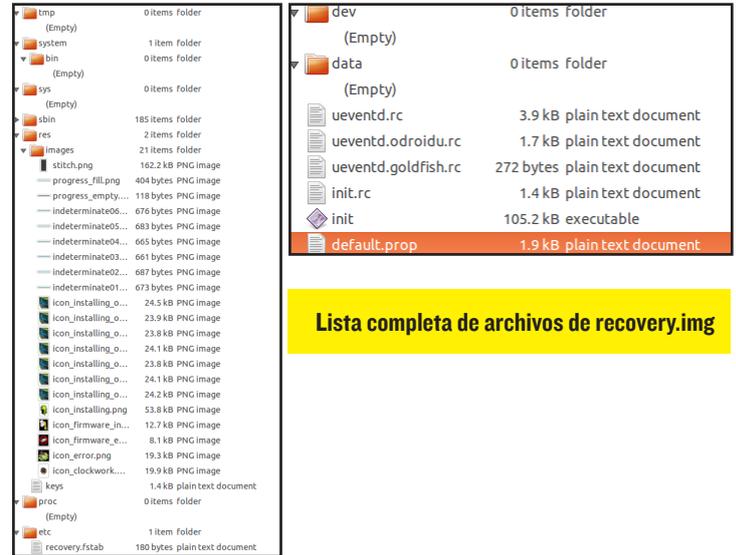
Checking for uImage magic word...
1+0 records in
0+1 records out
4 bytes (4 B) copied, 2.322e-05 s, 172 kB/s
uImage recognized.
Extracting data...
21697+1 records in
2712+1 records out
1388671 bytes (1.4 MB) copied, 0.0140089 s, 99.1 MB/s
Checking for ARM mach-type...
3+0 records in
0+1 records out
3 bytes (3 B) copied, 3.6532e-05 s, 82.1 kB/s
Checking for zImage...
1+0 records in
0+1 records out
4 bytes (4 B) copied, 3.2578e-05 s, 123 kB/s
>>> initramfs.cpio.gz extracted to Image
```

3. El archivo extraído es ahora un archivo gzip, así que tenemos que cambiarle el nombre:

```
$ mv Image Image.gz
```

4. Extrae el archivo recién renombrado usando gunzip:

```
$ gunzip Image.gz
```



Lista completa de archivos de recovery.img

5. Utiliza la herramienta cpio para extraer el contenido

```
$ cpio -i < ./Image
```

3820 blocks

```
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 bbconfig -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 blkid -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 bunzip2 -> busybox
lrwxrwxrwx 1 nanik nanik 8 May 10 22:37 busybox -> recovery
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 bzip2 -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 cal -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 bzcat -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 catv -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 chattr -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 chgrp -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 chmod -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 realpath -> busybox
lrwxrwxrwx 1 nanik nanik 8 May 10 22:37 reboot -> recovery
-rwxr-x-- 1 nanik nanik 840564 May 10 22:37 recovery
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 renice -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 reset -> busybox
lrwxrwxrwx 1 nanik nanik 7 May 10 22:37 resize -> busybox
```

Enlaces simbólicos Busybox

El directorio `/sbin` contiene un montón de herramientas, la mayoría de las cuales son enlaces simbólicos a `recovery` (vía `busybox`) como se muestra en la captura de pantalla.

system.img

Es el archivo de imagen más importante, ya que contiene el corazón de Android y es el más grande en términos de tamaño.

I. Usa la herramienta simg2img para extraer el archivo de imagen

```
$ simg2img system.img system.raw
computed crc32 of 0x9a5d4d54, expected 0x00000000
```

2. Usando la herramienta file, verás el tipo del archivo extraído.

```
$ file system.raw
./system.raw: Linux rev 1.0 ext4 filesystem data,
UUID=57f8f4bc-abf4-655f-bf67-946fc0f9f25b (extents)
(large files)
```

3. Crea un directorio para montar el archivo extraído. Monta el archivo en el directorio

```
$ mkdir sys
$ sudo mount -t ext4 ./system.raw ./sys
```

4. Verás un montón de carpetas dentro del directorio /sys.

Si miras cada una de las carpetas, verás los archivos de aplicaciones, de configuración y ejecutables que utiliza Android.

▶ app	59 items folder
▶ bin	178 items folder
▶ etc	22 items folder
▶ fonts	39 items folder
▶ framework	24 items folder
▶ lib	193 items folder
▶ media	1 item folder
▶ pseudorec	3 items folder
▶ tts	1 item folder
▶ usr	5 items folder
▶ vendor	1 item folder
▶ xbin	259 items folder
▶ build.prop	1.8 kB plain text document

Lista completa de archivos de system.img

userdata.img

Este archivo de imagen reside en su propia partición y será usado como la partición /data. El archivo de imagen no contiene muchos archivos y a veces, en algunos dispositivos es simplemente un archivo de imagen vacío. Dado que se utiliza para los datos, no importa su contenido, Android no lo usa y su único propósito es el almacenamiento.

Los siguientes pasos muestran cómo extraer el userdata.img

1. Usa la herramienta simg2img para extraer el archivo de imagen

```
$ simg2img userdata.img userdata.raw
computed crc32 of 0x20aec0ac, expected 0x00000000
```

2. Usa la herramienta file, podrás ver el tipo del archivo extraído.

```
$ file ./userdata.raw
./userdata.raw: Linux rev 1.0 ext4 filesystem data,
```

```
UUID=57f8f4bc-abf4-655f-bf67-946fc0f9f25b (extents)
(large files)
```

3. Cree un directorio para montar el archivo extraído y monta el archivo en el directorio .

```
$ mkdir data
$ sudo mount -t ext4 ./userdata.raw ./data
```

4. Verás que sólo hay 2 carpetas dentro del directorio /data

(Estas 2 carpetas no se utilizan en Android)

Empaquetar

En la sección anterior desempaquetamos el archivo de imagen, de modo que después de realizar los cambios en los archivos de imagen, tendremos que empaquetarlos de nuevo y utilizarlos para copiarlos en tu sdcard o eMMC.

boot.img y recovery.img parte 2

Después de realizar los cambios dentro del directorio, tiene que empaquetarlo y convertirlo en un archivo .gz con el siguiente comando.

```
$ find . | cpio -o -H newc | gzip > ../newramdisk.cpio.gz
```

Tienes que ejecutar el comando anterior dentro del directorio extraído después de descomprimirlo con el comando cpio como se muestra en el paso anterior.

Se creará un archivo ../newramdisk.cpio.gz, y este archivo será empaquetado usando otro comando:

```
$ mkbooting --kernel kernel.gz --ramdisk newramdisk.cpio.gz -o new_boot.img
```

El kernel.gz procede del boot.img original, mientras que el newramdisk.cpio.gz contiene nuestros cambios. Al finalizar el comando mkbooting tendrá una nueva imagen de arranque llamada new_boot.img.

system.img part 2

Después de realizar cambios en los archivos dentro del directorio system.img extraído, puede empaquetarlo con el siguiente comando:

```
sudo make_ext4fs -s -l 512M -a system ./system_new.img ./system
```

El último parámetro ./system es el directorio que contiene los archivos de imagen extraídos.

REDIMENSIONAR LAS PARTICIONES DE ANDROID

USA POR COMPLETO EL ESPACIO DE TU TARJETA SD O EMMC

por Venkat Bommakanti



Los medios de instalación de alta capacidad como eMMC o tarjetas SD tienen un montón de espacio disponible no asignado tras la instalación inicial de una imagen de la comunidad o de Hardkernel. Las particiones se mantienen pequeñas durante el proceso de creación de la imagen para que el archivo resultante sea más portátil y se pueda, incluso instalar en dispositivos de almacenamiento más pequeños de 4 GB o 8 GB. Con un dispositivo de 16, 32 o 64 GB, las particiones pueden ampliarse con el fin de aprovechar el espacio no usado y ponerlo a disposición del sistema.

Este artículo detalla los pasos necesarios para redimensionar correctamente la partición FAT32 de tu imagen de Android, normalmente denominada /mnt/sdcard. Si las particiones no se redimensionan correctamente, la resolución se fijará a 720p debido a que el archivo boot.scr no será localizado por el gestor de arranque. El procedimiento básicamente reasigna la mayor parte del espacio sin asignar a la partición Android al mismo tiempo que permite una resolución de 1080p - ¡Es tan simple como esto!

Android recién copiada y sin arrancar. Para descargar la última versión de Android correspondiente a tu plataforma, consulta el siguiente enlace <http://bit.ly/lk801bP>

3. Un Ordenador Linux con GParted instalado (para gestionar las particiones de disco). Si sólo dispones de un sistema MS Windows, puedes utilizar una de estas dos opciones:

Un Live CD con GParted para sistemas x86 (<http://bit.ly/1hGJVvw>), o

Una máquina virtual con Ubuntu 14.04 alojada en el sistema X86 con Gparted instalado. (<http://bit.ly/1r2OP48> <http://bit.ly/1n1DA5d>)

GParted para Linux es más recomendable que herramientas como EaseUS Partition Master para Windows, ya que Gparted detecta las particiones EXT3/EXT4 de Android sin problemas, sin embargo las versiones de EaseUS Partition no puede leer particiones EXT3/EXT4. Por lo tanto, se aconseja a los usuarios de Windows que ejecuten GParted a través de una memoria USB o máquina virtual.

partición de Android ya existente.

La primera captura de pantalla incluida en este artículo muestra el resultado de seguir los principales pasos en una tarjeta de 32 GB. Hay bastantes sub-pasos, así que asegúrate de seguir cuidadosamente las indicaciones que aparecen a continuación. Las tarjetas de una capacidad superior a 64 GB aún no han sido probadas, pero deberían funcionar correctamente si el espacio adicional se agrupa en una segunda partición FAT32.

La razón de separar instrucciones cuando se utiliza una tarjeta de 64 GB, es que las particiones FAT32 están limitadas a 32 GB en Android, aunque las imágenes de Android más recientes permiten tamaños de particiones más grandes. La mayoría de los pasos son compartidos entre el procedimiento para tarjetas de 32GB (o menos) y el de 64GB. Hemos incluido una tabla unificada que te puede ayudar en ambos casos.

Además, hemos incluido una guía gráfica para redimensionar las particiones con Gparted en las páginas 28 y 29.

Si has completado los pasos correctamente, el archivo boot.scr debe reconocerse permitiendo una resolución de 1080p. Es aconsejable realizar esta operación en una nueva copia de Android para evitar cualquier pérdida de datos. ¡Haga siempre una backup primero!

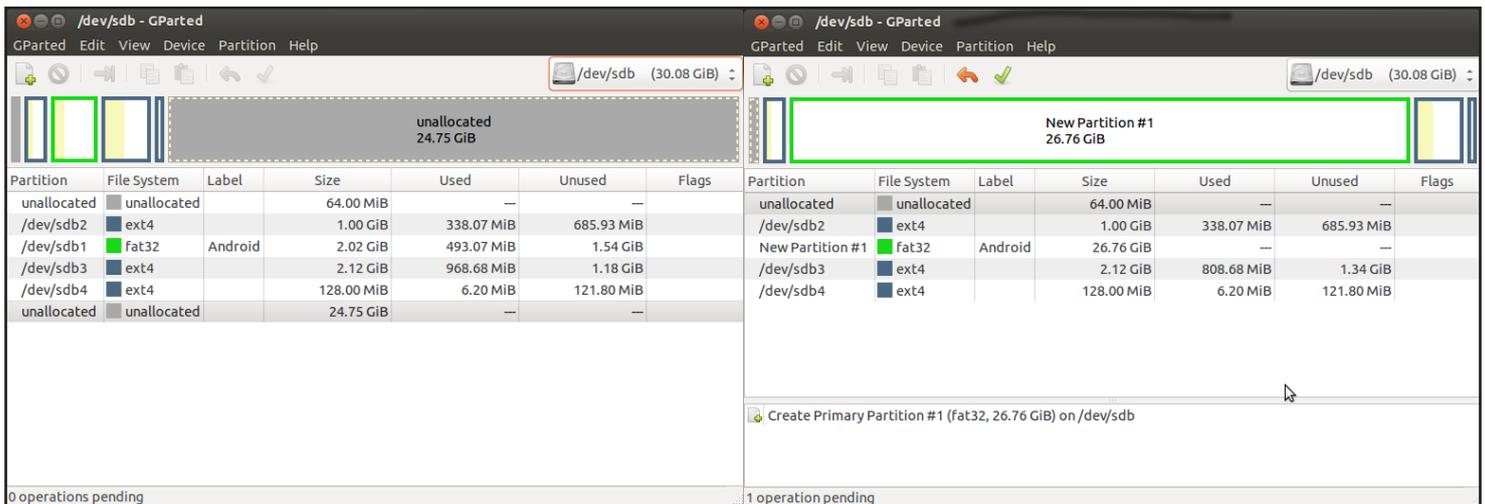
Requisitos

1. Una placa ODROID U3 o XU, con el adaptador de alimentación adecuado
2. Cualquier MicroSD o eMMC de 8+ GB, (junto con un lector de tarjeta SD) que contenga una imagen del an-

Introducción

Los dos primeros pasos implican mover dos particiones al final del dispositivo de almacenamiento (conservando sus respectivos tamaños), y

El tercer paso consiste en integrar la mayor parte del espacio sin asignar a la



Antes de redimensionar las particiones de Android

Despues de redimensionar las particiones de Android

Redimensionar una tarjeta SD o eMMC de menos de 64GB

Para eMMC o tarjetas SD de menos de 64 GB, siga los pasos 1-6, despues use los pasos 6a o 6b, y finalice con los pasos 8-10.

En primer lugar, sigue estos pasos para preparar tu SD o eMMC para cambiar su tamaño.

Tras hacer los pasos 1-5, pase al 6a o 6b dependiendo de la capacidad de tu SD o eMMC

1. Inserte la tarjeta o módulo en un PC con Linux usando un adaptador USB apropiado.
2. Crea un directorio temporal en la unidad local del PC, para almacenar los archivos de la partición FAT32 de la tarjeta. Realiza una backup de los archivos de la partición FAT32 de la tarjeta en el directorio de backup recién creado en el PC.
3. Usando GParted, realiza las siguientes operaciones en la partición FAT32 de la tarjeta:
 - Desmonta la partición FAT32 ,
 - Elimina la partición FAT32, y por último,
 - Selecciona [Apply] para guardar los cambios.
4. Al mismo tiempo en GParted, realiza estas operaciones:
 - Desmonta la última partición de la tarjeta,
 - Mueve esta última partición hasta el final de la tarjeta, y por último,
 - Selecciona [Apply] para guardar los cambios
5. A continuación, realiza estas operaciones finales:
 - Desmonta la segunda y última partición.

6a. (<64GB) Mueve esta segunda y última partición hacia el final, justo antes de la (última) partición previamente movida. Las dos últimas particiones deben tocarse entre sí, sin ningún espacio vacío (sin asignar) entre ellas.

- Selecciona [Apply] para guardar los cambios

7a. (<64GB) Cuando se haya completado, el esquema será:

- Una única partición intacta en el inicio de la tarjeta,
- Un montón de espacio en blanco y, a continuación,
- Dos particiones hasta llegar al final

Redimensionar una tarjeta SD o eMMC de 64GB

El proceso para redimensionar una tarjeta de 64 GB es algo diferente, las particiones FAT32 están limitadas a 32 GB.

6b. (64GB) Mueve esta segunda y última partición hasta más o menos la mitad del espacio no utilizado, de manera que: haya ~ 28GB de espacio libre a la izquierda de la partición, haya ~ 26GB de espacio libre a la derecha.

- Selecciona [Apply] para guardar los cambios.

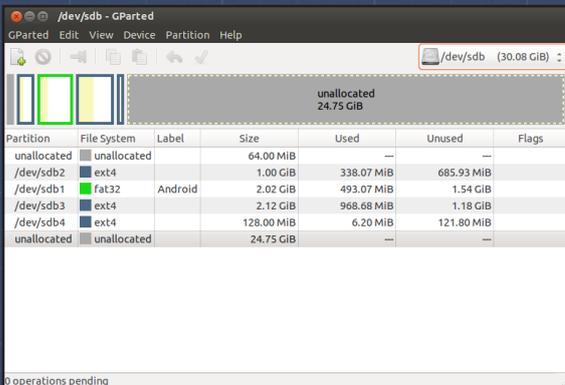
7b. (64GB) Edita la segunda y última partición:

- Cambiar su tamaño para utilizar ~26GB de espacio disponible hacia su derecha, y por último,
- Selecciona [Apply] para guardar los cambios.

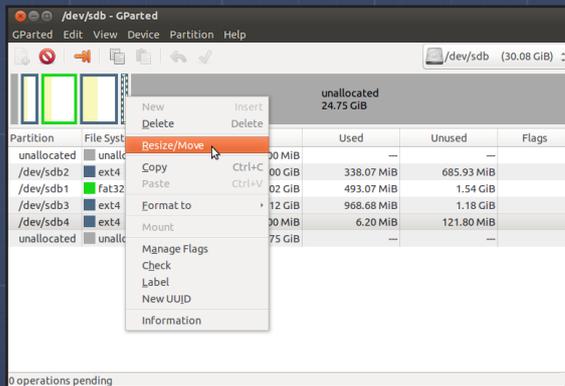
8. Formatea el espacio (disponible) no asignado como FAT32 y seleccione [Apply].
9. Copie los archivos de backup del directorio temporal (creado en el segundo paso) a la nueva partición FAT32.
10. Inserta la tarjeta en tu ODROID, arranquelo y disfruta de Android en su ODROID!

UNA RAPIDA GUIA GRAFICA PARA REDIMENSIONAR UNA SD O EMMC CON ANDROID

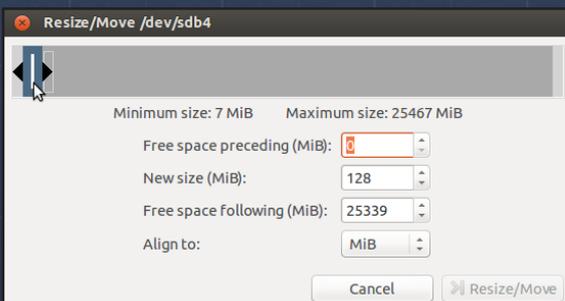
por Venkat Bomakanti



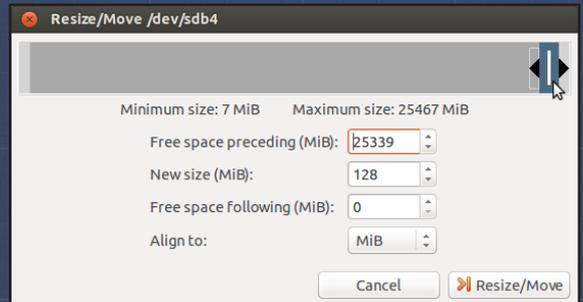
Esquema inicial de particiones



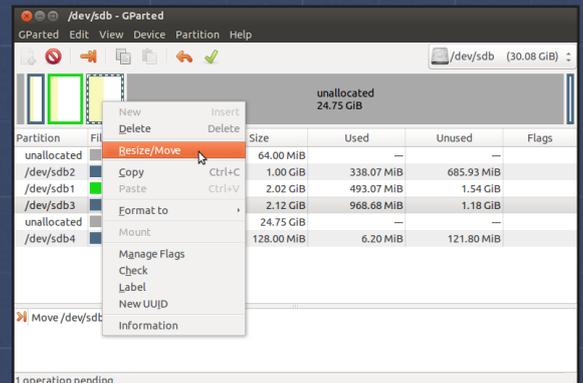
Redimensionar la partición de 128 MB seleccionada



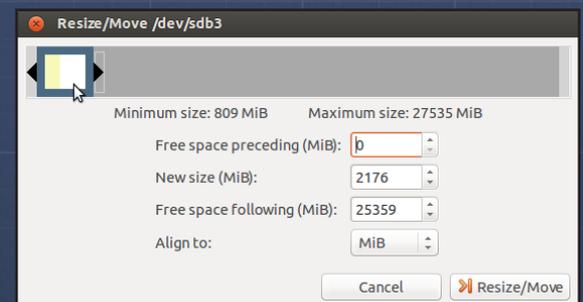
Parámetros iniciales de la partición de 128 MB



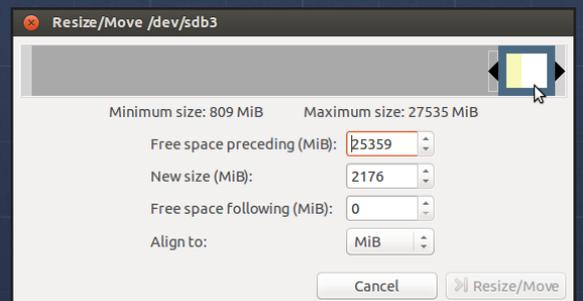
Nuevos parámetros para la partición de 128 MB



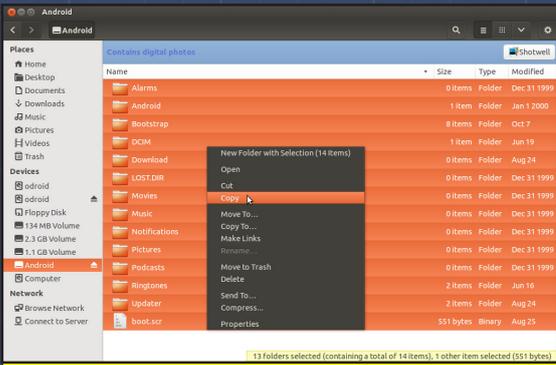
Redimensionar la partición de 2.12 GB seleccionada



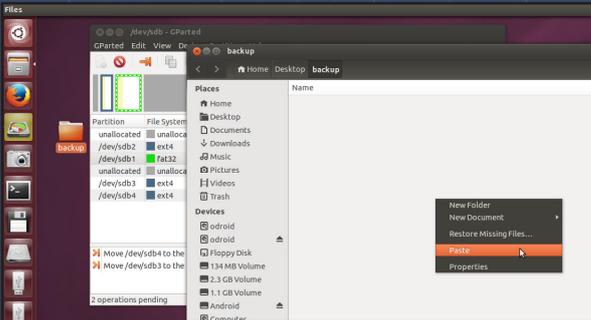
Parámetros iniciales de la partición de 2.12 GB



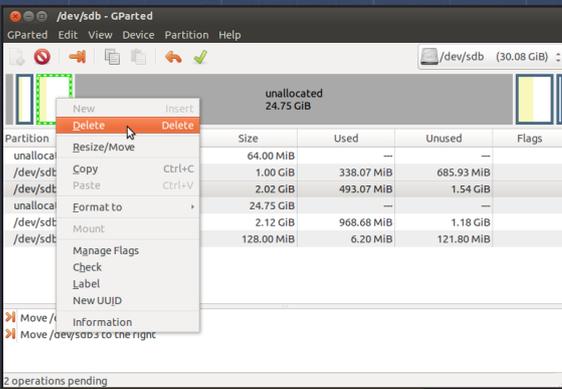
Nuevos parámetros para la partición de 2.12 GB



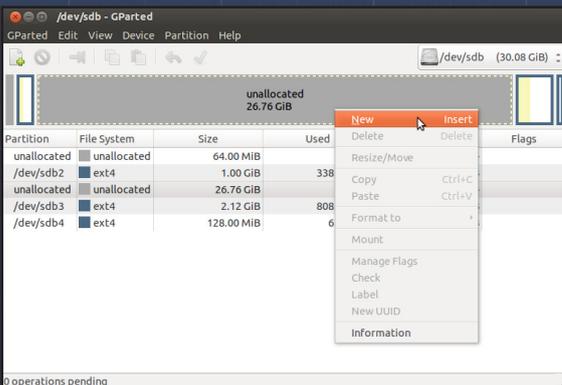
Copia la partición original de Android para hacer una backup



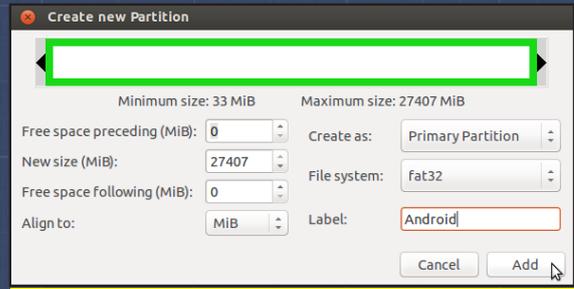
Guarda la copia de la partición de Android en la carpeta de backup



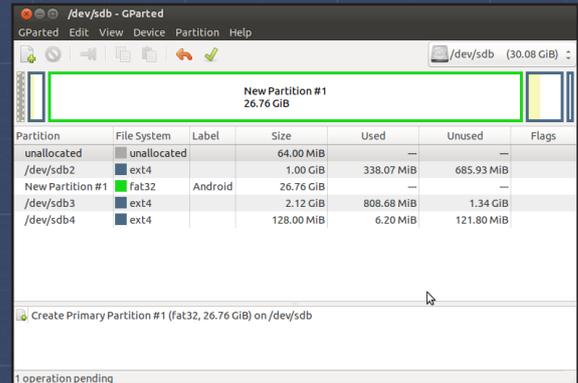
Elimina la partición original de Android



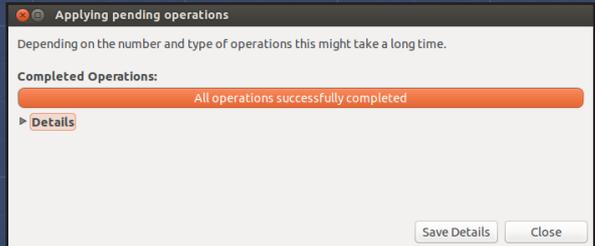
Manda crear una nueva partición de Android



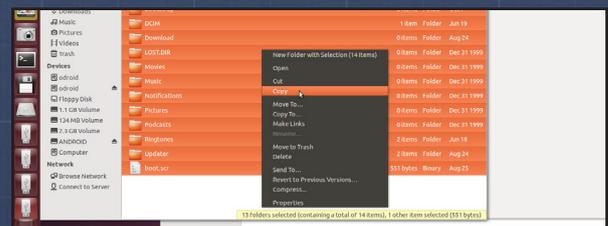
Introduce los parámetros para la nueva partición de Android



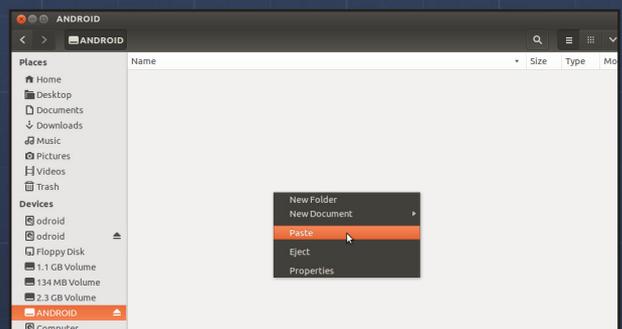
Guarda el nuevo esquema de particiones



Esquema de particiones completado



Copia el contenido de la carpeta backup (Partición original de Android)



Restaura los archivos de la backup de Android

COMO ALIMENTAR A TU GATO POR INTERNET

UNA GUIA PARA CONECTAR MOTORES DE PASO (MOTORES PASO A PASO) AL ODROID U3

por Marian Mihailescu

En el artículo anterior de automatización mostramos cómo conectar sensores al ODROID-XU: un sensor de temperatura y un detector de movimiento para monitorizar los movimientos del gato. En este artículo vamos a ir un paso más allá y aprenderemos a controlar un motor de paso utilizando ODROID. Y qué mejor aplicación para un motor que desarrollar un comedero automático para gatos controlado por internet.

Empezaremos con el comedero Cat Mate C50, que se muestra en la Figura 1.

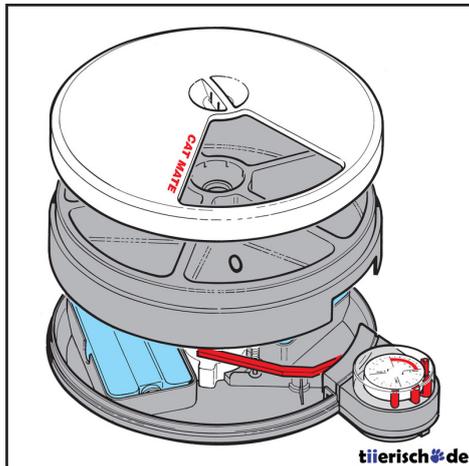


Figura 1. Cat Mate C50

Este comedero en particular tiene un mecanismo de reloj que funciona con pilas que se activa cada 24 horas, desplazando la palanca roja a la siguiente posición pasando al siguiente compartimento de comida. Es muy fácil de piratear - sólo tienes que eliminar el mecanismo del reloj que hace girar la palanca y reemplazarlo por un motor, como se muestra en las Figuras 2 y 3.

Para este proyecto, se va a utilizar el ODROID-U3, el cual presenta varias ventajas frente al ODROID-XU para este proyecto en concreto: es más económico, requiere menos energía, no tiene ventilador y cuenta con el accesorio IO-SHIELD opcional, que incorpora puertos GPIO adicionales. El comedero de gato se puede controlar mediante un motor de paso de 5V (modelo 28BYJ-48), que se puede adquirir en eBay junto con la placa controladora ULN2003 por menos de 3\$.

El motor de paso es ideal en este caso, ya que puede girar un número fijo de pasos, permitiendo un control preciso del comedero del gato. En particular, el mo-



Figura 2a - Mecanismo de reloj sin el engranaje de la palanca visible



Interrumpimos este artículo para anunciar el regreso de nuestro fiel gato!

tor 28BYJ-48 tiene 32 pasos por giro, con un engranaje de 1/64, lo que significa que hace 2,048 pasos en un ciclo completo (360 grados). La mayoría de los motores normalmente necesitan más corriente de la que las placas de desarrollo pueden soportar, y por lo general están conectados a una fuente de alimentación externa o a un pack de pilas. Sin embargo, este motor en particular puede obtener toda la energía que necesita desde nuestro ODROID.

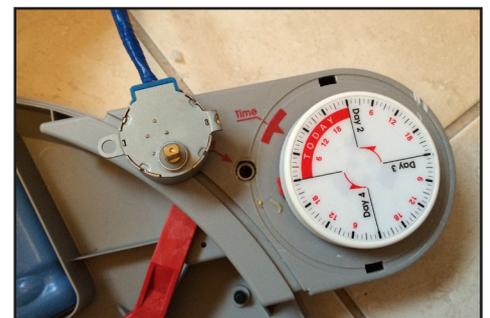


Figura 2b - Vista frontal del engranaje



Figura 3a - Motor de paso 28BYJ-48 de 5V con la placa controladora ULN2003

Figura 3b - El motor encaja perfectamente en el engranaje



La función de la placa controladora es permitir una pequeña salida de corriente desde ODROID para controlar la corriente que necesita el motor de paso. La placa está controlada por cuatro entradas que determinan cómo se magnetiza la bobina con el fin de rotar el núcleo magnético del motor. También existen la alimentación de 5V y la toma de tierra, lo que hace un total de seis conexiones al ODROID.

Puesto que el motor requiere 5V, necesitamos un conmutador para trabajar con salidas de 1.8V del ODROID. Al igual que en el artículo anterior de automatización, vamos a utilizar el Freetronics Logic Level Converter. Dado que la controladora del motor requiere cuatro pines GPIO para su control y ODROID U3 sólo tiene tres GPIOs (GPIO199, GPIO200 y GPIO204), es necesario el componente IO-SHIELD. No obstante, sin disponer de IO-SHIELD, es posible controlar la placa controladora con sólo tres GPIO usando un registro de desplazamiento.

El registro de desplazamiento consigue en que todas las salidas GPIO requeridas se coloquen en serie y luego se envíen al motor de paso en paralelo. Tendrás que conectar la entrada SERIAL IN al GPIO que envía la lógica de control del motor. El RELOJ se usa para hacer que el registro de desplazamiento cargue el valor lógico de SERIAL IN en una transición baja-alta, mientras que se trasladan los demás valores almacenados. LATCH activa la salida del registro de desplazamiento en una transición baja-alta. OUTPUT ENABLE se utiliza para desactivar las salidas - ya que que-



Aunque un hipermotor espacial sería mucho más útil, sucumbimos al deseo de combinar robótica y Linux para dar de comer de forma automática a nuestros felinos, en caso de que de repente "desapareciéramos".

remos que estén activadas todo el tiempo, podemos conectarlo a GND, y RESET se usa para borrar el registro de desplazamiento. Podemos conectar OUTPUT ENABLE a 5V (VCC) y configurar cada vez todas las salidas, para que RESET no sea necesario. Este modelo de registro de desplazamiento en particular tiene ocho salidas, así que cargaremos ocho valores cada vez, aunque sólo necesitamos cuatro.

La controladora del motor está conectado al motor en el comedero por un cable de 2 m, lo que permite que la placa sea colocada de forma segura lejos del gato. Para controlar el comedero, el script bash mostrado a continuación se ejecuta a través de una

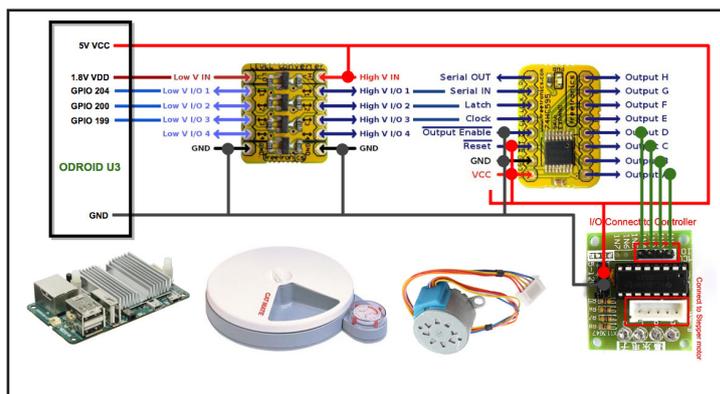


Figura 4 - Esquema del comedero controlado por internet con ODROID

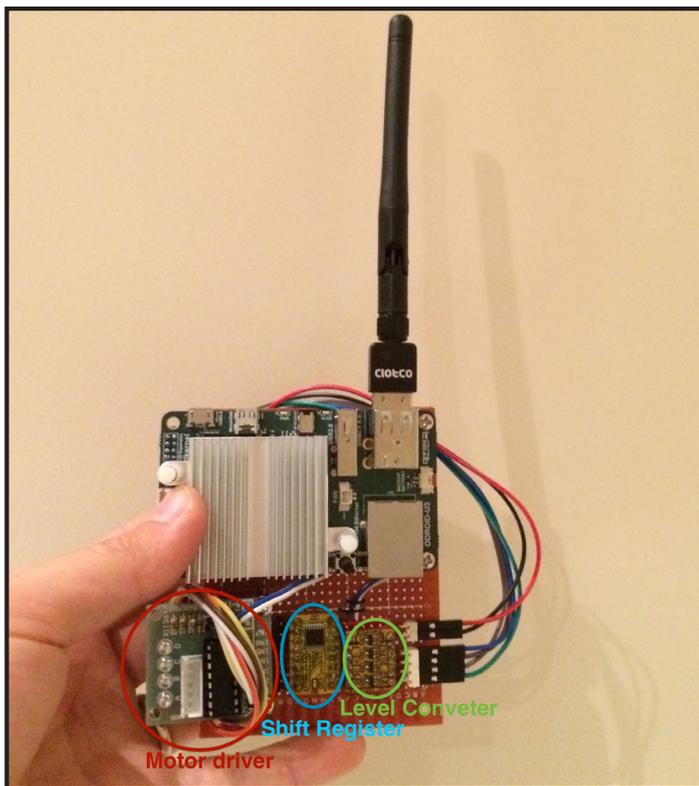


Figura 5a - Placa con todos los componentes conectados

página web. También se ha acoplado una webcam a ODROID. El servidor webcam Motion permite video en tiempo real en la página web junto con los controles del comedero del gato. Con una versión algo más compleja del script y el planificador cron job de Linux es posible activar el comedero no sólo cuando sea necesario, sino también de forma automática, en períodos de tiempo predefinidos. ¡Tu gato nunca ha estado tan contento!

```
#!/bin/bash

### enable ODROID U3 GPIO

echo 199 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio199/direction
echo 0 > /sys/class/gpio/gpio199/value

echo 200 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio200/direction
echo 0 > /sys/class/gpio/gpio200/value

echo 204 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio204/direction
echo 0 > /sys/class/gpio/gpio204/value

### setup motor sequence

declare -a SEQ0=(0 0 0 0 1 0 0 0)
declare -a SEQ1=(0 0 0 0 0 1 0 0)
declare -a SEQ2=(0 0 0 0 0 0 1 0)
declare -a SEQ3=(0 0 0 0 0 0 0 1)

### define level shifter GPIO

IN=/sys/class/gpio/gpio204/value
LATCH=/sys/class/gpio/gpio200/value
CLK=/sys/class/gpio/gpio199/value

### control motor

STEP_COUNTER=0
```

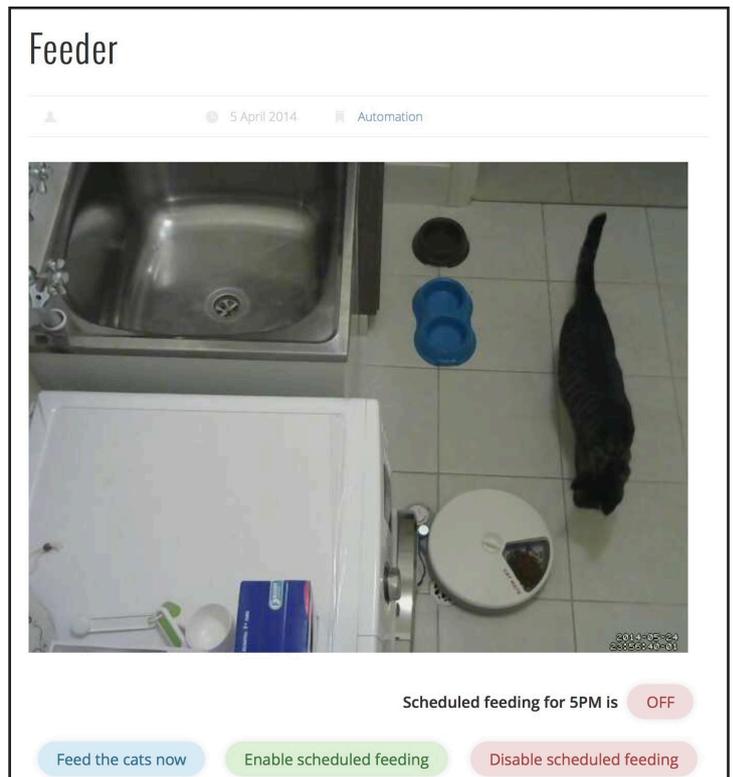


Figura 5b - Web para controlar el comedero a través de Internet

```
for STEP in {1..4096}
do
    echo 0 > $LATCH

    for PIN_COUNTER in {0..7}
    do
        PIN_VALUE="SEQ$STEP_COUNTER[$PIN_COUNTER]"

        echo ${!PIN_VALUE} > $IN
        echo 1 > $CLK
        echo 0 > $CLK
    done

    STEP_COUNTER=$((STEP_COUNTER + 1))

    if [ "$STEP_COUNTER" = "4" ]
    then
        STEP_COUNTER=0
    fi

    echo 1 > $LATCH
    echo 0 > $LATCH

    sleep 0.000000001

done

### after feeder activated, reset motor inputs

for RESET in 0 0 0 0 0 0 0 0
do
    echo $RESET > $IN
    echo 1 > $CLK
    echo 0 > $CLK
done

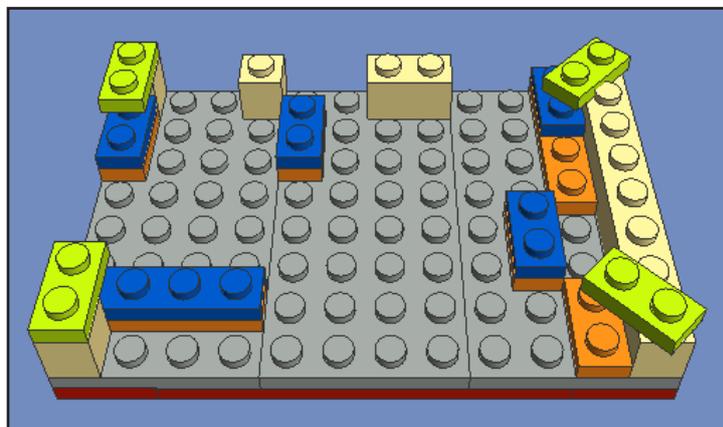
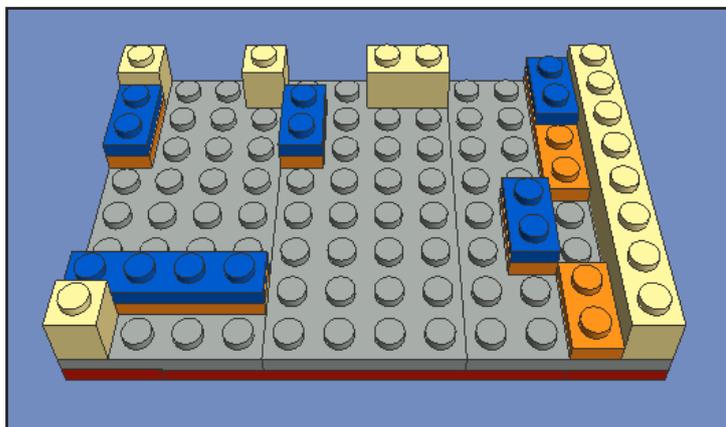
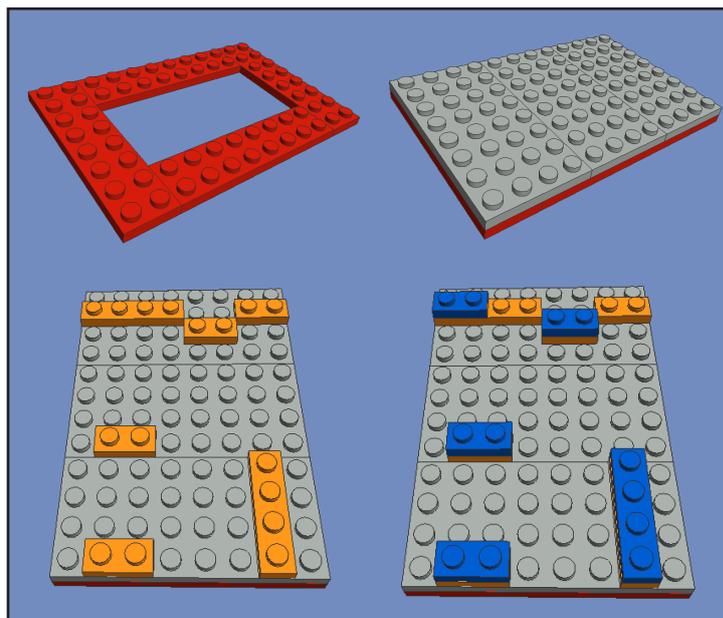
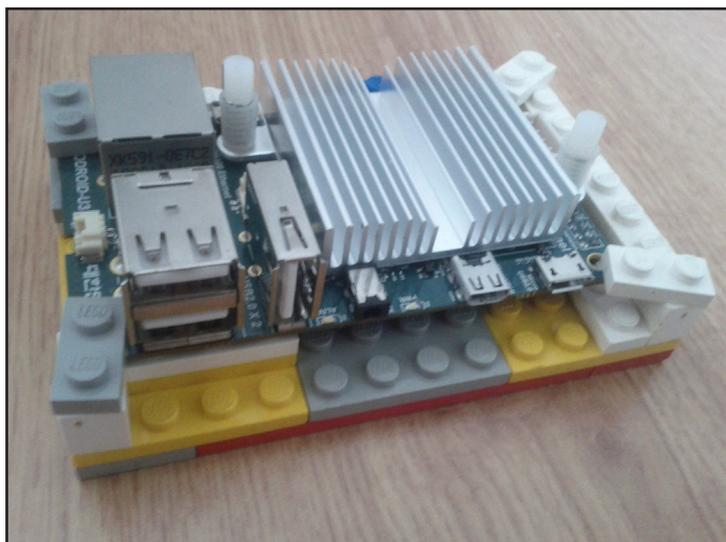
echo 1 > $LATCH
echo 0 > $LATCH

echo "[$(date)] : Feeder activated successfully... meow!"
```

FABRICA UNA CAJA LEGO A MEDIDA PARA TU U3

¿TODAVIA NO TIENEN TU PROPIA IMPRESORA 3D?
MONTA ESTA GRAN CAJA CON TU CUBO DE LEGO!

por @Zylophone



COMO ACTIVAR LA SALIDA DE AUDIO MULTI-CANAL CON XBMC

USANDO EL PERIFERICO USB-S/PDIF PARA REPRODUCIR SONIDO ENVOLVENTE DIGITAL 5.1

por Justin Lee, CEO de Hardkernel

A menudo recibimos peticiones en los foros sobre cómo activar el Sonido Passthrough de 5 canales con los ordenadores ODROID. Sin embargo, no hemos podido encontrar la forma de implementar el Passthrough a través del canal de audio de HDMI con el hardware nativo, aun cuando lo hemos intentado de muy diversas formas. Así que, hemos decidido desarrollar la funcionalidad Passthrough de otro modo, mediante una interfaz S/PDIF a USB.

Configurar el Hardware

Para permitir Passthrough por HDMI, hemos creado una placa USB-S/PDIF utilizando el chip CM108AH. La denominada USB-S/PDIF, con soporte para AUDIO LINK tiene un precio de sólo 16\$ e incluye un cable USB y carcasa de plástico.

El Cmedia CM108AH es una solución de audio USB con un único chip in-



¿Qué necesita esta configuración? Un ODROID conectado, ¡por supuesto!



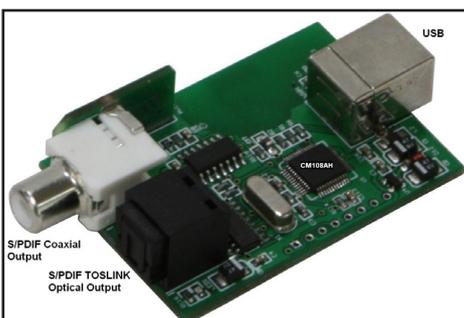
Diagrama de flujo del Audio

tegrado. Todos los TOSLINKS S/PDIF así como las salidas coaxiales, incluyendo PLL, reguladores y receptores USB pueden implementarse sobre una pequeña PCB.

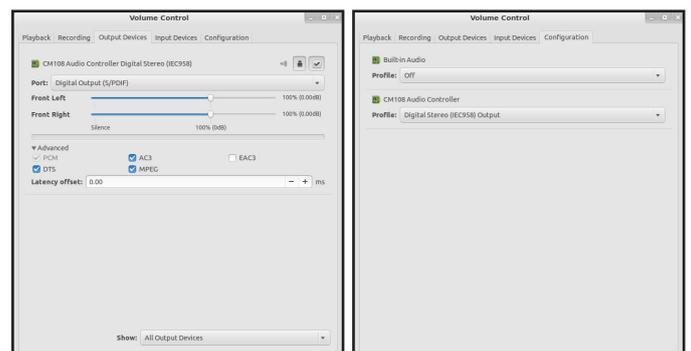
Configurar el Software

La función Passthrough en XBMC 12 Frodo no funciona con ODROID pero afortunadamente, XBMC 13 Gotham es capaz de activar la salida Passthrough HDMI. Para un mejor resultado utiliza la última imagen de Ubuntu o Debian que tiene el XBMC 13 preinstalado.

Hemos probado nuestro hardware con estas 2 imá-



ODROID Audio Link I



Opciones de configuración de PulseAudio



ODROID Audio Link 2

genes pre-compiladas, Se puede usar cualquiera de las dos para lograr la configuración necesaria para activar el sonido envolvente de 5 canales:

<http://forum.odroid.com/view-topic.php?f=29&t=4823>

<http://forum.odroid.com/view-topic.php?f=79&t=4626>

Para empezar la instalación, conecte el USB-S/PDIF y arranca tu placa. Para activar los Codecs de Passthrough, primero configure PulseAudio Volume Control de acuerdo a las capturas de pantalla. Selecciona Digital Output (S / PDIF), luego elige AC3 y DTS en el menú de opciones avanzadas.

Por último, arranca XBMC, para configurar la función Passthrough. Selecciona el dispositivo CM108 en el menú de salida de audio, y luego habilita las opciones: Enable passthrough, Dolby Digital AC3 y DTS.



Configurar XBMC

El receptor que usamos es este tutorial es de un nivel básico y barato, el cual es ideal para desarrollos y pruebas.

Para verificar que el hardware y el software están configurados correctamente, la reproducción de un vídeo con sonido AC3 debe mostrar que la codificación Dolby Digital está siendo usada



Nuestro sistema de audio que permite tanto Dolby Digital como DTS. 5.1 es impresionante!

por el amplificador, y al reproducir un video con sonido DTS debe indicar codificación DTS.

Hay tarjetas de sonido USB que utilizan salida S/PDIF que deberían funcionar como dispositivos Passthrough 5.1, siempre y cuando el controlador de Linux sea compatible. Tenga en cuenta que PulseAudio no permite actualmente Passthrough DTS-MA o TrueHD, el cual es una limitación de PulseAudio, no relacionada con el hardware USB-S/PDIF o la configuración de XBMC.

El USB-S/PDIF ya está disponible en la tienda Hardkernel por 16\$ <http://bit.ly/1kIt8aC>

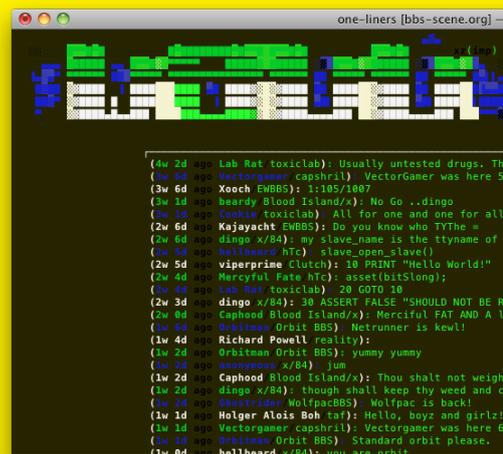
VIAJA A ATRAS EN EL TIEMPO CON TELNET DESEMPOLVA ESE VIEJO MODEM DE 1200 BAUDIOS

por Rob Roy

Antes de que se inventara la World Wide Web y el HTTP, el protocolo conocido como Telnet era el modo más conocido para comunicarse entre ordenadores. Mediante la creación de una página web (primitiva) que funcionaba a través de Telnet, los hackers desarrollaron una gran comunidad de nodos servidores basados en texto que ejecutaban un software llamado BBS (Bulletin Board System). Esta red incluía a muchas comunidades con intereses especiales que compartían aficiones, tales como juegos de rol, música y programación. Para viajar hacia atrás en la historia, hasta 1984, cuando la red BBS estaba en su edad de oro, escriba en una ventana de terminal:

```
telnet 1984.ws 23
telnet gcomm.com
```

Regístrate y disfruta de algunos de los gráficos ASCII más avanzados de los primeros días de la informática en red. Experiencias nostálgicas con BBS como Multi-User Dungeon (MUD), programas de chat y otros sistemas que puedes encontrar en <http://bit.ly/1qnkmSq>



SO DESTACADOS: DREAM MACHINE Y WHISPER

UBUNTU 13.04 Y 13.10 CON 8 ENTORNOS DE ESCRITORIO TAMBIEN DISPONIBLES EN TURBO

por Rob Roy, Editor jefe

Dream Machine es una imagen pre-compilada por la comunidad, que ofrece casi todos los entornos de escritorio de Ubuntu disponible para las versiones 13.04 y 13.10, incluyendo Lubuntu, LXDE, Kubuntu (KDE), Blackbox, Openbox, Matchbox, Unity and Xubuntu (XFCE). El entorno de escritorio se puede seleccionar con el icono de la esquina superior de la pantalla de inicio de sesión. Whisper viene con un único entorno orientado al rendimiento para optimizar la velocidad. Lightweight X11 Desktop Environment (LXDE), gráficamente es muy rápido y su interfaz es similar a la de Windows.

Puesto que Dream Machine incluye todo el software que contiene la imagen Fully Loaded Ubuntu 12.11, descrita en el número anterior (mayo 2014) dirígete a éste para conocer los paquetes incluidos. No obstante, Dream Machine también viene con XBMC, no incluido en Fully

Loaded puesto que XBMC sólo funciona con el Kernel 3.8. de ODROID. Hay varias mejoras en la reproducción a 1080p en XBMC desde que finalizó el desarrollo de Dream Machine 13.10. La última versión de XBMC estará incluida en la próxima versión Dream Machine 14.04.

KDE Plasma Workspace

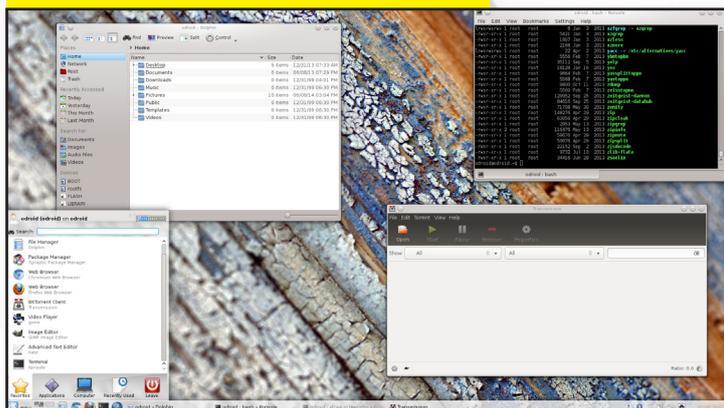
Aunque todos los entornos de escritorio en Dream Machine contienen básicamente el mismo software, hay muchas diferencias en su apariencia, así como características adicionales relacionadas con la interfaz de usuario. El entorno por defecto en la versión 13.10 es KDE Plasma Workspace, que cuenta con magníficos fondos y efectos de transición acelerados por hardware. KDE viene con un driver para OpenGL llamado `kwin_gles`, para que las animaciones sean suaves y no incidan en la CPU. Puesto que los escritorios de Ubuntu están (generalmente) escritos para OpenGL, no

son compatibles con la GPU Malí y están creados a nivel de software, excepto KDE que ofrece una capacidad de respuesta de interfaz de usuario muy rápida.

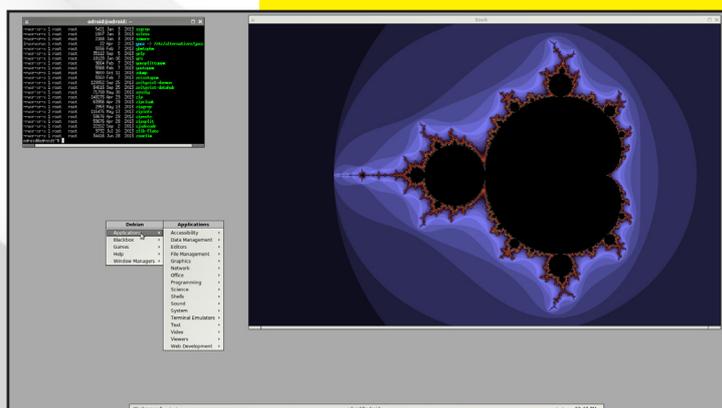
KDE Plasma Workspace resultará familiar a los usuarios de los sistemas operativos de Windows, ya que cuenta con un menú de inicio bien organizado y utiliza un diseño de escritorio similar, con un área de notificación en la esquina inferior derecha. Tiene su propio protector de pantalla y es el único entorno que permite incluir sonidos de notificación. También cuenta con una reciente área de Documentos y Aplicaciones así como muchos widgets útiles, tales como relojes y el tiempo. Es mi escritorio favorito y es una buena opción para el trabajo de oficina o navegar por internet. Para obtener más información sobre el escritorio KDE, o para contribuir con su proyecto de código abierto en desarrollo, visita <http://www.kde.org/>

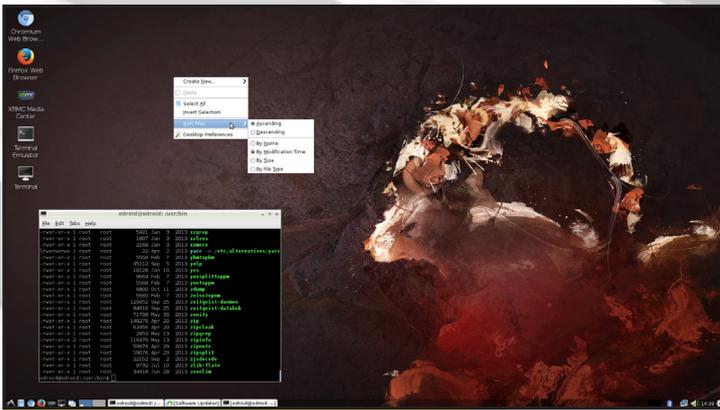
Dream Machine y Whisper con Kernel 3.8 para U2/U3/X/X2 puede descargarse desde <http://bit.ly/IrhHymu>
Vigila los foros ODROID en <http://forum.odroid.com/> para la versión 14.04. ¡Disponible muy pronto!

KDE plasma Workspace cuenta con aceleración GPU Open GLES2 con asombrosos efectos!



Blackbox sigue una estética minimalista.





LXDE (Lightweight X11 Desktop Environment) busca el equilibrio entre el rendimiento, la velocidad y la funcionalidad.

Blackbox

Es el entorno de escritorio más simple, sólo muestra una barra de tareas y un menú compacto de Aplicaciones, al que se accede haciendo clic (botón derecho del ratón) en el escritorio. Se presenta como una forma muy ligera para monitorear los procesos del sistema, o simplemente reducir los ciclos de CPU para aplicaciones que demanden mucha potencia. Un escenario típico para usar Blackbox es un sistema Linux embebido en el que se ejecuta una única aplicación personalizada, como un controlador de robótica. El diseño es visualmente minimalista, ejecuta las aplicaciones de forma bastante rápida y requiere de poco espacio en memoria.

LXDE

LXDE significa Lightweight X11 Desktop Environment, y es el más adecuado para aquellos que deseen obtener el máximo rendimiento al tiempo que quieren mantener una interfaz robusta. Tiene el aspecto de Windows XP y su-

Matchbox está diseñado para PDAs, pero también se ve muy bien en monitores HD 1080p.



Lubuntu se basa en LXDE y ofrece una amplia variedad de aplicaciones de software únicas

prime composiciones gráficas y otros efectos visuales para mejorar su eficacia.

XBMBC funciona muy bien con LXDE, y es la mejor opción para aquellos que desean configurar un decodificador Linux con XBMBC, Transmission o Youtube. LXDE ofrece un administrador de archivos exclusivo denominado PCMan-FM, que aprovecha las librerías gio/gvfs para conectarse a SFTP, WebDAV, SMB y otros protocolos del sistema de archivos virtuales soportados en Gnome.

Lubuntu

Lubuntu, basado en LXDE, ofrece algunas mejoras visuales, iconos más grandes y una barra de tareas más eficiente. También cuenta con muchos y nuevos paquetes de software, incluyendo el Centro de Software de Lubuntu.

Matchbox

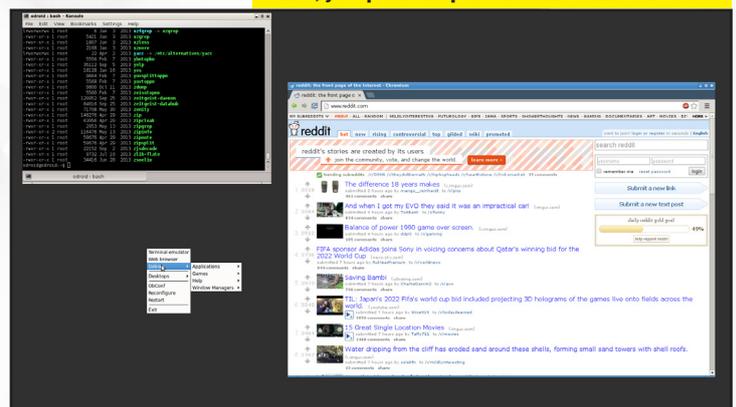
Matchbox es diferente al resto al usar todo el escritorio como menú de aplicaciones. Usa las flechas de la parte superior izquierda y derecha para navegar por las categorías, y desplázate hacia abajo para ver más aplicaciones. Matchbox es perfecto para acceder a los programas

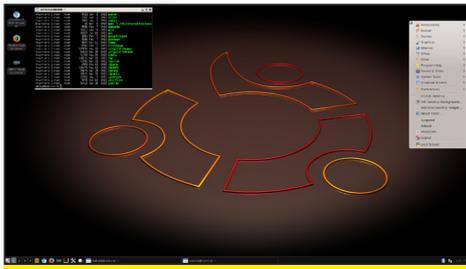
e iconos de forma rápida sin necesidad de abrir constantemente un menú, ya que el acceso directo de cada aplicación se muestra en una única página. Aunque Matchbox fue diseñado para PDAs y tablets, se ve muy bien en un monitor HDMI 1080p. Matchbox sólo muestra una ventana al mismo tiempo, en lugar de usar un diseño de múltiples ventanas.

Openbox

Openbox en realidad no es un entorno de escritorio, sino más bien un administrador de ventanas independiente. Se puede ejecutar por sí solo o dentro de otro entorno de escritorio, y usa la mínima cantidad de recursos necesarios para lanzar programas y gestionar ventanas. El tema principal de Openbox es la velocidad y es ideal para programas sensibles al tiempo, servidores de alto tráfico y aplicaciones donde el uso de la memoria y el procesador es importante. Openbox es muy pobre, compacto e incluye muy pocas dependencias.

Openbox solo tiene un terminal y navegador web, y es perfecto para sistemas embebidos.

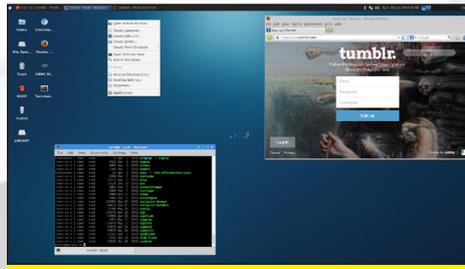




Razor está escrito íntegramente en el lenguaje de programación avanzado QT

Razor

El escritorio Razor-qt está escrito íntegramente en el lenguaje de programación QT, y es uno de los entornos de escritorio más recientes que se incluían en Ubuntu 13.10. Aunque es un proyecto nuevo, Razor-qt ya cuenta con muchos componentes claves, como un panel, escritorio, lanzador de aplicaciones, centro de configuración y sesión de usuario. Desarrollado para la versatilidad y la reutilización, Razor-qt puede ejecutarse con cualquier gestor de ventanas, aunque la mayoría de los desarrolladores de Razor prefieren usar Openbox. Es posible utilizar Razor con cualquier gestor de ventanas moderno, desde gwm2 a KWin, incluyendo KDE sin Plasma Desktop.



Xubuntu es el favorito de los aficionados a Ubuntu y ofrece muchas funciones fáciles de usar.

XFCE y Xubuntu

Xubuntu basado en el entorno de escritorio XFCE, incluye varios paquetes destinados a reemplazar muchas aplicaciones de software basadas en Gnome. Incorpora una barra de accesos directos en la parte inferior que aparece cuando pasas el cursor por encima, y un menú de inicio en el lado superior izquierdo como Gnome. XFCE es uno de los favoritos para muchos usuarios de Ubuntu debido a su equilibrio entre velocidad y funcionalidad, así como su práctica organización de aplicaciones en categorías. La función de búsqueda, representada por una lupa en la barra de acceso directo es similar a la barra de búsqueda de Unity, con la que puedes localizar cualquier programa con relativa facilidad.



XBMC, programa para reproducir todos los medios, y también puede lanzar aplicaciones.

XBMC

XBox Media Center (XBMC) puede funcionar como una aplicación independiente, pero también se puede disfrutar junto con tu propio entorno de escritorio instalando el paquete “XBMC-stand-alone” desde el Gestor de paquetes Synaptic. Es una solución todo-en-uno para reproducir audio y vídeo, soporta conexiones de servidor Samba, reproducción 5.1 DTS, complementos de terceros y curiosas visualizaciones de audio. También se puede utilizar para lanzar programas de Linux, de modo que se pueden acceder a los paquetes de software sin tener que volver al escritorio de Ubuntu. Más información sobre el uso de XBMC en <http://bit.ly/1kTXXXI>

Whisper (LXDE)

Whisper es una versión monousuario de Dream Machine, que ofrece un entorno de escritorio (LXDE) y arranca directamente el escritorio por estar configuradas las variables de autosección en /etc/LightDM/lightdm.conf. Está diseñado para el rendimiento y la velocidad, y contiene los mismos paquetes de software que Dream Machine. Whisper es una de las imá-

genes más populares disponibles en los foros. Proporciona una interfaz familiar al estilo de Windows XP que es ideal para programar, reproducir vídeo, usar aplicaciones con alto consumo de CPU, o simplemente para tareas livianas como navegar por internet y editar documentos.

Aunque Dream Machine no está todavía disponible para la serie XU, Whisper ha sido exportado con éxito



a XU, y ofrece gxm en lugar de XBMC para la reproducción de vídeo, ya que la versión XU de XBMC no se ejecuta con

X11. Abra cualquier archivo multimedia desde el Explorador de archivos para iniciar automáticamente gxm, y haga doble clic en la ventana de reproducción para activar la visualización de pantalla completa.

Ahora que las imágenes 14.04 de Hardkernel han sido liberadas, una versión actualizada de Dream Machine y Whisper será publicada en los foros en los próximos meses. Las recientes mejoras en XBMC que permiten la reproducción a 1080p sin problemas también se incluirán en las versiones X y U. Hardkernel está mejorando constantemente su rama 3.8 del kernel, y ha sustituido recientemente el script kernel-update.sh por una nueva utilidad, que también será añadida en las nuevas versiones.

Whisper, ¡Tan rápido como tu OROID!

TIENES EMAIL... O DEBERIAS!!!

SUBSCRIBETE A LA LISTA DE CORREO DE
HARDKERNEL PARA ESTAR AL DIA DE LAS
ULTIMAS NOTICIAS DEL MUNDO ODROID

por Bruno Doiche

¿Quieres saber cuándo se va a liberar una nueva versión de tu Sistema Operativo? ¿Cuáles son los contenidos de cada número de ODROID Magazine?

Envíanos un email ahora mismo - No te molestaremos con noticias corrientes, ¡Sólo lo mejor de lo mejor!
odroid@hardkernel.com

Las últimas versiones de software para ODROIDS

Versión 14.04 de Ubuntu para las placas ODROID-U3/U2/X/X2

- * Canonical Ubuntu 14.04 LTS base
- * Drivers Mali GPU r4p0 + Drivers armsoc + patched Xorg server
- * Kernel 3.8.13.23 con drivers de aceleración por hardware 2D/3D mejorados
- * Utilidad de ODROID "Todo en uno" para la configuración de HDMI, actualizar Kernel, expandir la partición root y otras herramientas de mantenimiento
- * XBMC 13.1 Gotham con Exynos-4412 MFC/FIMC, aceleración por hardware para videos a 1080p
- * Plugins Gstreamer acelerados por hardware
- * Detalles : <http://forum.odroid.com/viewtopic.php?f=29&t=5234>

Android KitKat 4.4.3 para placas ODROID-U3/U2/X/X2

- * El último Android KitKat 4.4.3 ha sido exportado a ODROID.
- * El código fuente e imagenes actualizadas estan disponibles en: http://com.odroid.com/sigong/nf_file_board/nfile_board.php

Descubre la última tecnología ARM desde ODROID Magazine, ¡Una revista electrónica en PDF de tirada mensual!

EJEMPLAR DE MAYO



- Proceso de Inicio de Android: Entender los entresijos de como arranca android en tu ODROID
- Pipe Viewer: Consigue más interactividad con tus progresos de datos
- La fuerza es poder con Traceroute
- Cómo compilar Doom en tu ODROID: Juega a este clásico de siempre adaptado y compilado para ODROID
- Recompila Drivers de video Mali: Corregir los problemas de la "Pantalla en blanco" y "Ventanas retardadas" al actualizar a Ubuntu 14.04
- 2 Sistemas, 1 ODROID, ¡Pura diversión! Cómo hacer un doble sistema de arranque con Android y Ubuntu
- Primeros pasos con tu ODROID: Cómo copiar un archivo de imagen a una tarjeta SD o eMMC
- Consigue un poco más de personalidad en tu Sudo
- Localiza los archivos más grandes de un directorio
- Cómo dividir un gran archivo
- Sobre el comportamiento térmico de los ODROIDS: La diferencia de rendimiento entre el XU y el U3 con mayor detalle
- Una campaña de Indiegogo promete compatibilidad ODROID con un ambicioso objetivo
- ODROID-SHOW: Una potente Mini Pantalla LCD para el U3
- ODROID-UPS Kit: Una solución para los tiempos de inactividad de tu ODROID
- SO Destacado: Fully Loaded Ubuntu 12.11 con el entorno de escritorio Unity 2D
- Monitoriza tu Linux con Nmon
- Contruir un vehículo todoterreno automático con ODROID: Parte 1 - Esquema general, Montaje de plataforma y Distribución de energía
- Conociendo a un ODROIDian: Simone (@Serf00), Un apasionado de ODROID desde hace tiempo y un experto informático

EJEMPLAR DE ABRIL



- Desarrollar Android en ODROID-U3: Partiendo de cero, Hazte con el control total de tu Sistema Android
- Shairport: Convierte tu ODROID en una estación de sonido con Itunes
- Backup portable de una imagen: Crea un archivo de recuperación para tu Sistema Operativo favorito
- Renombra tus archivos de mayúscula a minúscula en una línea de comandos
- Protejerte de los accidentes de Superusuario
- Cómo desarrollar tu propio Ubuntu desde cero: Usando RootFS de Linaro ¡Compila Linux como un profesional!
- Cómo instalar la versión 8 del kit de desarrollo de Java (JDK) de Oracle: Ahorra tiempo con la arquitectura "de código único y multiplataforma" de java
- Usar ODROIDS en la Informática de Alto Rendimiento: Qué diferencia a un kernel compilado
- Juegos Android: Vector, repleto de acción Parkour
- Cómo configurar un Servidor Minecraft: ¡Enredate!
- Descargar videos de Youtube para verlos sin conexión
- Crear un muñeco Papercraft para llevarlo junto a tu Servidor Minecraft
- Conocer Rebol: Escribir programas útiles con un código asombrosamente pequeño y facil de entender
- Ser escuchado con Ubercaster: Un transmisor de audio en tiempo real
- Comunicación I2C ODROID-U3: Circuitos integrados para el resto
- Tablet Linux resistente y portatil con Router LTE
- Cómo crear un CAR PC para mi camioneta usando ODROID: No importa los productos que haya en el mercado
- Conociendo a un ODROIDian: Marian Mihailescu, Uno de nuestros principales colaboradores del foro

CONOCIENDO A UN ODROIDIAN

RUPPI KIM, UNO DE LOS MIEMBROS FUNDADORES DE HARDKERNEL

por Robert Hall



Por favor, Háblanos un poco sobre ti.

Soy el director de tecnología de Hardkernel, uno de sus miembros fundadores. Vivo en Corea del Sur desde hace 39 años y trabajo en la industria de la ingeniería de hardware y software desde hace unos 20 años. El diseño de PCB fue mi primer trabajo hasta que hice el servicio militar a los 18 años de edad. Tras éste, entré en el desarrollo de hardware y software.

¿Qué es lo que haces en Hardkernel?

Soy un ingeniero de sistemas de hardware. Diseño los circuitos de los ordena-

dores ODROID, así como de las placas principales. En mi tiempo adicional, trabajo con la última versión del kernel de Linux, junto con compras, fabricación y todo lo relacionado con esto. También me ocupo del servidor odroid.com y del repositorio GitHub de Hardkernel

¿Cómo fueron tus inicios con los ordenadores?

Empecé en el colegio en 1998. Recuerdo la primera vez que se mostró un equipo en clase para un curso piloto. Era en primer grado (creo que tenía 13 años), y el ordenador era un DAEWOO

IQ1000 (MSX). Antes de descubrir los ordenadores, yo sólo los conocía por los nombres de marca en las salas de juegos arcade.

Conocí primero un lenguaje de programación que permitía el diseño de formas con caracteres y aritmética denominado MSX BASIC. Esta es la hoja de especificaciones para la IQ1000 que usé en la escuela: <http://bit.ly/1p1e1kM>

Después de 3 años, me compré un modelo NEWTEK 286 (AT) de 1500 \$ como primer ordenador. Tenía un microprocesador Intel 80286 (Código negro P2) a una



이제, 대우퍼스컴 아이큐 1000으로 세계 어느 MSX방식 컴퓨터의 소프트웨어도 실행 수 있습니다.

소프트웨어를 알아본지 바깥에 할 수 있는
대우퍼스컴 아이큐 1000은
당시의 유명한 소프트웨어—
그리고 소프트웨어를 운영 가능한 수 십만개의 프로그램은, 그 기능을
확실히 증명합니다. 이제 소프트웨어를 마음대로 구입하여 쓸 수 있는
대우 퍼스컴 아이큐 1000이 널리 보급되었습니다.

국내에서 처음으로 선보인 16MHz의 고성능 마이크로 프로세서에서 개발된 MSX
방식은 세계, 영인력까지 앞으로 영인되는 소프트웨어를 마음대로 할 수 있
습니다. 이제 여러분이 영인력 받으실 수 있는 컴퓨터를 구입하십시오.

소프트웨어는 항상 풍부해지고
가치는 높아져갑니다.
안타깝게도 MSX방식을 위한 대우퍼스컴 아이큐 1000은 다양한
소프트웨어를 공급하고 소프트웨어를 무료로 할 수 있는 기회를
제공합니다. 이 기회를 놓치지 않으셨습니까? 이제 영인력 받으실 수
있는 대우 퍼스컴 아이큐 1000이 널리 보급되었습니다.

대우 퍼스컴 아이큐 1000의 주요 사양은 다음과 같습니다.
* CPU: Z8001 (16MHz) * RAM: 128KB * ROM: 40KB * 플로피 디스크 드라이브: 5.25인치 1.2MB * 그래픽: 320x200 * 오디오: 16KHz * 키보드: MSX방식 * 모니터: MSX방식 * 전원: AC 100V * 크기: 400x300x150 (mm) * 무게: 10kg

대우퍼스컴 아이큐 1000

Nuestro editor de arte estaba fascinado con esta imagen de Bakyeon Falls en invierno, junto con la oportunidad de publicar una foto de un viejo MSX! Este artículo no podía ser mejor ...

... A menos, por supuesto, llegamos a incluir las fotografías de los adorables bebés! Bienvenidos Kyung A y Kyung Euny!



velocidad de 16MHz. Con Turbo llegaba a una velocidad de 21MHz. Utilizaba una tarjeta gráfica Trident 1MB SVGA con un monitor Goldstar (LG Electronics) de 14", y tenía 1 MB de memoria. Aprendí principalmente lenguaje ensamblador y mecano-grafía con este equipo.

¿Qué tipos de proyectos has realizado con tus ODROIDs?

En el futuro, mis ODROIDs serán utilizados con fines educativos para mis nuevas hijas gemelas que nacieron en el

otoño del año pasado. El cuidado de las niñas se ha convertido en mi nuevo hobby. Sus nombres son Kyung A y Kyung Eun. Cuando no estoy con mis hijos, estoy muy implicado en el desarrollo de Linux y el próximo nuevo kernel de Linux: <http://bit.ly/SehCTb>