

ODROID

Año Uno
Num #2
Feb 2014

Magazine

Convierte tu ODROID en una:

GIGANTE TABLET

Y disfruta
de la última
experiencia
Android!



Informática de Alto

Rendimiento en casa

Creación y control exhaustivo

- Evaluando la interferencia en red:

Usando java multithreading

- Empieza a programar hoy mismo!

- Drivers dispositivos USB

Domina los puertos a fondo

- Conociendo a un Odroidian
Mauro Ribeiro, Programador Senior

- Juegos Linux – Emuladores

- El arte del multi-boxing

- Router 811.02ac con odroid XU
Hasta 430 mb/s



Al servicio de las comunidades “Open Source” y ODROID de todo el mundo, HardKernel tiene el placer de presentar su aportación más reciente a la tecnología ARM:

ANDROID Magazine, Una revista gratuita mensual en formato PDF.

Esta moderna publicación electrónica ofrecerá las últimas noticias de la plataforma ODROID, así como artículos destacados de la comunidad de expertos que ha surgido en torno a esta sorprendente familia de microordenadores.

Dirigida a cualquier persona desde principiantes a expertos, ODROID Magazine presenta guías paso a paso para configurar su ODROID, instalar sistemas operativos y software, además de soluciones a los problemas más comunes. Para los usuarios más expertos contará con consejos y trucos, discusiones de seguridad, proyectos especializados y artículos técnicos que permitirán explorar todas las posibilidades que ofrece ODROID.

ODROID Magazine es una gran oportunidad para que la comunidad pueda compartir sus experiencias con todo el mundo. Cada mes se publicarán una serie de temáticas para que cualquier persona pueda enviar sus aportaciones. Los artículos seleccionados que se incluirán en la revista serán premiados con regalos que se enviarán a los autores. Los mejores artículos son los que a través de una lectura sencilla, el lector pueda comprender y asimilar conceptos complejos. Es necesario enviar al menos una imagen por artículo y éste debe tener una extensión entre 500 y 2000 palabras.

En esta edición te mostramos cómo convertir tu ODROID en una enorme tableta Android de 42” pulgadas, que puede ser utilizada en quioscos, para paneles de señalización digitales o simplemente para jugar. ¡Juega al “Fruit Ninja” como nunca los has hecho!

ODROID

Magazine

Robert Hall, Editor Jefe

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

Bo Lechnowsky, Editor

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

Bruno Doiche, Editor Artístico

Pasó sus últimas vacaciones sin hacer nada. Y ni si quiera con la mitad de ellas.

Madamuz, Equipo de Traducción

Un grupo de personas fascinadas por todas las posibilidades que ofrece ODROID.

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



CONVIERTE UN MONITOR EN UNA GRAN TABLET ANDROID

Justin Lee y Charles Park

Las pantallas táctiles son muy comunes en dispositivos como teléfonos inteligentes, consolas, ordenadores “todo-en-uno” y tablet. También juegan un importante papel en el diseño de aparatos digitales como paneles de señalización, terminales punto de venta (TPV), Dispositivos GPS, videojuegos y libros electrónicos.

Android, uno de los principales sistemas operativos para ODROID, cuenta con una interfaz de usuario muy intuitiva diseñada para su uso en pantallas táctiles. Este artículo describe cómo convertir ODROID en una gigante tablet utilizando cualquier monitor o televisión.

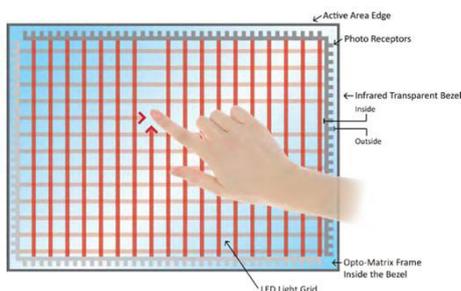
Infrarrojos vs Capacitiva

Las pantallas táctiles usan principalmente infrarrojos o tecnología capacitiva. Esta última está más entendida entre los smartphones y tablet pero también es más cara, especialmente cuando el

tamaño de la pantalla supera las 20". Una pantalla capacitiva sólo funciona con el dedo (o con un puntero) y puede presentar dificultades si el monitor no está correctamente montado. Teniendo en cuenta su facilidad de uso y bajo coste, vamos a utilizar la tecnología infrarroja para desarrollar este proyecto.

Sistema de cuadrícula por infrarrojos (IR)

Una pantalla táctil por infrarrojos utiliza una matriz LED de infrarrojos X-Y



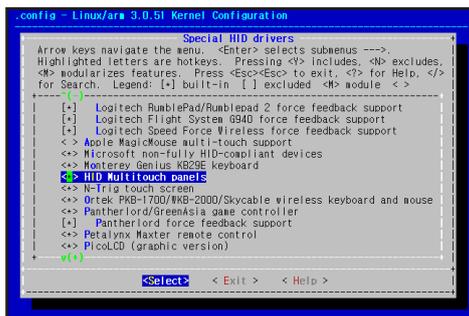
Principio de la Pantalla Táctil por IR (Infrarrojos)

y fotodetectores en los bordes de la pantalla para detectar una interrupción en el patrón LED. Estas líneas LED se cruzan entre sí en vertical y horizontal, que ayudan a los sensores a detectar la ubicación exacta cuando se toca la pantalla. Una ventaja importante del sistema por infrarrojos es que puede detectar cualquier entrada, un dedo, un guante, un lápiz o un bolígrafo. Generalmente se utiliza en dispositivos al aire libre y terminales punto de venta en los que puede presentar cierta dificultad la utilización de una pantalla táctil.

A diferencia de las pantallas táctiles capacitivas, las pantallas táctiles por infrarrojos no requieren ningún patrón en el cristal, lo que mejora la durabilidad y claridad del sistema. Sin embargo, estas pantallas son más sensibles a la suciedad y al polvo. Además, no son aptas para superficies curvas y pueden activar elementos erróneamente cuando el usuario desliza el dedo por la pantalla sin

Paso 3: Compila el kernel con la opción MULTI-TACTIL

Escriba “make menuconfig” para configurar el kernel, dirígete a Device Drivers > HID Devices > Special HID drivers > HID Multitouch panels y seleccionarlo para activar el controlador (*), tal y como se ve en la siguiente captura de pantalla.



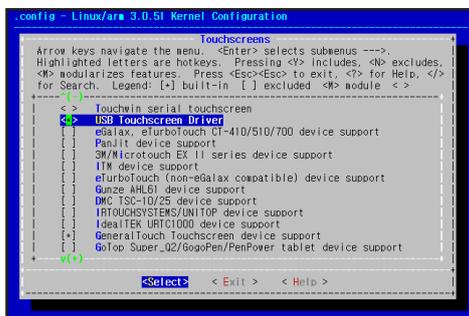
Configurando la pantalla activando el driver en la configuración del Kernel

A continuación, activa otras dos opciones (*):

Device Drivers > Input device support > touch screens > USB touch screen Driver

y:

Device Drivers > Input device support > touch screens > GeneralTouch touch screen device support



Configurando la pantalla activando driver en la configuración el Kernel

Guarde la configuración del kernel y compílolo para hacer una “zImage”.

Traslada la “zImage” a tu ODROID a través del protocolo fastboot en el “u-boot”.

Paso 4 : Crea un archivo IDC (Configuración del dispositivo de entrada)

Si no se crea correctamente el archivo IDC, la resolución de la pantalla táctil no coincidirá con la resolución HDMI. Crea un archivo de texto planto como este:

```
touch.deviceType = touchScreen
touch.orientationAware = 1
device.internal = 1
keyboard.layout = qwerty
keyboard.characterMap = qwerty2
keyboard.orientationAware = 1
keyboard.builtIn = 1
cursor.orientationAware = 1
```

El nombre del archivo debe ser: Vendor_xxxx_Product_yyyy.idc (xxxx: Vendor ID, yyyy: Device ID) . Crea dos archivos para Elitegroup y Nexio. El nombre es sensible a mayúsculas.

```
Vendor_03fc_Product_05d8.idc
and
Vendor_1870_Product_0119.idc
```

Copia los archivos IDC a tu ODROID usando los comandos correspondientes.

```
adb remount
adb push Vendor_03fc_Product_05d8.idc /system/usr/idc/
adb push Vendor_1870_Product_0119.idc /system/usr/idc/
adb reboot
```

Cómo fijar el panel táctil de 42” a la pantalla de TV.

Conecte con cuidado la pantalla táctil para alinear la ventana de visualización.



Preparando la pantalla: Fijando la cinta de doble cara al marco de la pantalla táctil

Testea la pantalla táctil. En nuestra prueba se pudo detectar hasta 6 puntos.



Prueba de la pantalla táctil - 6 puntos garantiza la integración de sensor con el software.

Esta pequeña pantalla de 23” incluía 4 correas de velcro que facilitaron bastante su montaje.



Panel táctil de 23 pulgadas con cintas de velcro, Jugando a “Fruit Ninja” en android.



Conclusión

Además de su uso personal y como plataforma para juegos, ODROID es ideal como dispositivo informático para quioscos, señalización digital, investigación de interfaces, debido a su alto potencial, bajo coste y plataforma abierta que permite modificaciones como el montaje de una pantalla táctil. Los resultados de este proyecto se pueden ver en <http://www.youtube.com/watch?v=HDSnuxchxtU>, y para un mayor efecto <http://www.you->

INSTALANDO ANDROID EN UN ODROID

CONTINUAN LAS CRÓNICAS DEL CIENTÍFICO LOCO

Bohdan Lechnowsky

Momento en el que te encuentras en tu acogedor laboratorio, encorvado sobre tu mesa, el suave resplandor de la pantalla ilumina tu rostro, los indicadores de red parpadean sin cesar, catalizando productos químicos en varios tubos de ensayo. Antes de poner en marcha tu nuevo ODROID-U3, te encuentras inverso en el proceso de creación de una especie de super-teléfono con el que dominaras el mundo

En otro ODROID se visualiza un video en YouTube del show de The Ben Heck mientras conectas tu U3 a varios dispositivos (baterías, pantalla...) con viejos cables.

Al finalizar la última conexión miras hacia arriba con una amenazante sonrisa y susurras "¡Pulsa el interruptor!". Tras un momento de silencio te das cuenta que no existe ese ayudante de laboratorio que imaginastes por un momento. Afortunadamente tu U3 dispone de un botón de encendido.

Enciendes tu U3, observas cierta inactividad y te das cuenta que la SD o eMMC no tiene sistema operativo. Tras una breve meditación decides instalar Android que parece ser la mejor opción pudiendo arrancar en paralelo Linux, que te ayudara en tu ambición por dominar el mundo tal y como se describía en el

número de Enero de ODROID Magazine. Cómodamente, imprimes el último número de ODROID Magazine para facilitar el trabajo. En ciertas ocasiones, simplemente es agradable sentir el papel entre los dedos.

Tras varios intentos fallidos en la descarga de la última versión de Android para ODROID, finalmente consigues iniciarla aunque a baja velocidad.

Como U3 es -para la mayoría de los efectos- idéntico a U2, se utiliza el mismo procedimiento de instalación. Localizas la última versión completa de Android para el U2: Android Beta 1.6. Las versiones más recientes son simplemente actualizaciones de esta versión.

Descarga la imagen Android_Beta_1.6 para U2 desde http://dn.odroid.com/Android_Beta_1.6/U2/ y trasládala a tu tarjeta SD/eMMC siguiendo el tutorial disponible en la edición de Enero de ODROID Magazine. Una vez que se haya completado el proceso, introduzca la tarjeta en tu ODROID y grita "¡Pulsar el botón!". En ese momento recordarás la ausencia de ese ayudante de laboratorio. Observarás como el logo de Android aparece en la pantalla conectada a su U3.

Localiza las instrucciones para actualizar a la nueva versión de Android en el blog odroid.com y haz algunos cambios para equipararla a la versión actual:

Compruebe que la versión del sistema operativo es superior a la Beta 1.6

(Settings > System > About Tablet)

El número de compilación tiene un código de fecha, y debe ser de abril del 2013 o posterior. La imagen utilizada es del 24-Abril-2013.

Descargue el archivo actualización de Android Visita http://dn.odroid.com/Android_Beta_1.8.0/U2/ Seleccione la imagen SD o eMMC dependiendo de la tarjeta que se use y ponga en marcha la descarga.

Cree el directorio de actualización.

Cree el directorio `/sdcard/updater` utilizando el Explorador de archivos que debería estar incluido en su lista de aplicaciones.

Mueva el archivo de actualización al directorio de actualizaciones

Copie el archivo zip descargado que contiene la versión beta de Android en el directorio `/sdcard/updater` folder.

Inicie el proceso de actualización.

(Settings > About Tablet > ODROID)

INFORMATICA DE ALTO RENDIMIENTO (HPC) EN CASA

LA INFORMATICA COMO NUNCA LA HAS VISTO

Cooper Filby y Anthony Skjellum -

Runtime Computing Solutions LLC

En el número anterior de ODROID Magazine hablamos de los beneficios de ODROID en la Informática de Alto rendimiento (HPC), por los diversos descubrimientos realizados cuando trabajábamos con plataformas XU+E y que están relacionados con la expansión de nuestro propio clúster a nivel comercial. Pero ¿Qué ocurre con el usuario particular que desea experimentar con HPC? En la primera parte de esta serie de artículos, describiremos de forma general la instalación y configuración de un clúster básico sin un nodo principal, con el fin de ejecutar programas en paralelo basados en la Interfaz de Paso de Mensajes (MPI).

En futuros artículos, ampliaremos esta instalación para crear un clúster con un sistema principal centralizado o nodo, que contará con servicios como Puppet, NFS y LDAP. Una configuración que es capaz de soportar mayor número de sistemas ODROID (nodos) configurado como un clúster HPC. En el segundo número detallaremos una configuración de red básica con un nodo principal incluyendo dnsmasq y NAT, y añadiremos más nodos al clúster. Mientras que en el tercer número (y entregas futuras) trataremos servicios adicionales como LDAP, NFS, Autofs y Puppet.

Instalación del sistema Operativo

Existen varios sistemas operativos Linux precompilados disponibles para dispositivos ODROID en <http://dn.odroid.in>. Para comenzar, descargue la imagen “Ubuntu Server” para su modelo exacto de ODROID. Para acceder al archivo IMG.XZ puede optar por la herramienta 7zip o utilizar “xz” desde la línea de comandos de Linux. Por último, debe copiar la imagen a una tarjeta SD o módulo eMMC con el comando “dd” en Linux o utilizar la herramienta Win32DiskImager.exe para ODROID en Windows. Para obtener información de la instalación del sistema operativo, consulte el artículo de Bohdan Lechnowsky titulado “Instalación de un sistema operativo en ODROID” de la edición de enero de ODROID Magazine. Recomendamos utilizar los módulos eMMC disponibles en Hardkernel, para un mejor rendimiento. Sin embargo las tarjetas SD también funcionan.

Conectando tu ODROID y ajustando configuración de usuario

Dado que hemos optado por utilizar “Ubuntu Server” en nuestros dispositivos ODROID, podemos conectarnos a



Si esto no hace que montes un clúster “Voltron” usando 5 Odroids, no nos hacemos responsables de los “robots gigantes” que pueden algún día dominar tu ciudad

nuestros sistemas XU-E (que llamaremos nodos). Utilizaremos el protocolo “ssh” usando Terminal (o Putty si estamos en Windows) para continuar con la configuración de nuestro clúster. En la siguiente sección resolveremos los conflictos de direcciones MAC y hostname. Ahora tenemos que arrancar el primer ODROID y realizar algunos ajustes antes de pasar a la siguiente fase.

[Nota: Si es posible, es recomendable disponer de una máquina con Linux o Windows que facilitará la instalación y reinicio del clúster, solucionar problemas de hardware y corregir posibles errores. Una alternativa a la utilización de un equipo independiente es conectar un teclado USB y un monitor en el primer ODROID y utilizarlo directamente para arrancar el clúster en lugar de usar SSH como se describe en la página siguiente. Presiona Ctrl-Alt-F1 para utilizar la consola framebuffer si X11 no se ejecuta]



Para conectarte a tu ODROID ha de conocer el nombre del

host o dirección IP del dispositivo. En la imagen “Ubuntu Server” utilizada en nuestro clúster XU+E el hostname por defecto es “odroid-server”, mientras que para otras imágenes que hemos utilizado, el hostname ha sido “odroid”. La mayoría de las redes domésticas deberían soportar DNS por defecto, lo que permite conectarse simplemente con el nombre de host. Si esto no funciona, se puede conectar mediante la dirección IP asignada al ODROID por el router de tu red. Si no se resuelve ningún nombre de host entre en el panel de control del router para localizar las direcciones IP que se necesitan. Esta información suele agruparse bajo la etiqueta DHCP en el panel de administración del router.

Dado que se ha utilizado copias idénticas de la imagen en ambos nodos, estos presentan un conflicto de nombres de hosts. Este problema se resuelve cambiando la configuración de la red. Si no puedes acceder al panel de control

del router, puedes utilizar el comando “nmap” para escanear la red en busca de los dispositivos ODROID. Ej.: “nmap 192.168.1.0/24”. Busca un host que tiene el puerto 22 abierto.

Encienda uno de los ODROID y a continuación introduzca “ssh odroid@ubuntu-server” (o “ssh odroid@xxx.xx.xx.xxx”, si utilizas la dirección IP) en el Terminal o con la herramienta Putty del ordenador principal. Esto establecerá una conexión segura con el dispositivo ODROID. Utiliza “odroid” como contraseña para iniciar sesión.

Una vez establecida la conexión debes ejecutar “Sudo apt-get update && sudo apt-get upgrade” para asegurarnos de que el sistema operativo este actualizado. Además, se recomienda ejecutar el comando “passwd” para cambiar la contraseña de ODROID y establecer una más segura. También puedes crear nuevas cuentas de usuario con el comando “adduser”, como por ejemplo “Sudo adduser kilroy”.

Configurando la Red

Antes de que los dispositivos ODROID estén operativos y funcionando, debemos realizar algunos cambios en la configuración de la red y eliminar los conflictos de direcciones MAC y hostname que pueden aparecer con un clúster ODROID. Para cambiar el nombre del host tenemos que editar dos archivos /etc/hostname y /etc/hosts cambiando “odroid-server” por otro nombre y reiniciar la máquina para que los cambios tengan efecto. Para establecer la diferencia, utilizaremos odroid-server0 y odroidserver1 para referirnos al primer y segundo ODROID respectivamente. De forma alternativa y si su sistema operativo lo permite, puede ejecutar “sudo odroid-config” para cambiar el nombre del host y utilizar cualquier otro nombre. Cada nombre del host debe ser único para cada nodo.

El conflicto de direcciones MAC es

una cuestión delicada que detectamos cuando configuramos varios dispositivos ODROID XU + E. Nos hemos encontrado que, por defecto, el componente Ethernet integrado en todos los dispositivos compartían la misma dirección MAC, lo que hizo imposible trabajar en un único ODROID si conectábamos múltiples en la misma red. Como ambos ODROID tienen direcciones MAC idénticas, podemos hacer dos cosas: 1) configurar uno (o ambos) ODROID para usarlo con direcciones MAC diferentes. 2) Configurar los componentes Ethernet USB para que todos tenga direcciones MAC únicas. No importa que valores específicos se elijan, siempre y cuando sean únicos en tu Red de Área Local (LAN).

Para cambiar la dirección MAC de un dispositivo, edita /etc/network/Interfaces con un editor de texto y añade la línea “hwaddress ether newmac”, donde Newmac es una dirección con el formato “b6:8d:67:7b:cb:e0”, debajo de las siguientes líneas.

```
auto eth0
iface eth0 inet dhcp
```

A continuación, reinicie el ODROID para que los cambios tengan efecto. Asegúrese de verificar la nueva dirección con el comando “ifconfig”. Alternativamente, se puede optar por conectar un adaptador Ethernet USB en la ranura USB 3.0 y a continuación, ejecutar “ifconfig-a | eth grep “ que debe devolver:

```
eth0 Link encap:Ethernet
HWaddr b6:8d:67:7b:cb:e0

eth2 Link encap:Ethernet
HWaddr 00:13:3b:99:92:b1
```

Por defecto, eth0 será el componente Ethernet 10/100 integrado, mientras que el segundo dispositivo de Ethernet (en este caso, eth2) será el adaptador Ethernet USB. Si sólo aparece eth0, intente



Sí, se trata de “robots gigantes” que se apoderaron de su ciudad, “Voltron” siempre gana.

colocar de nuevo su adaptador Ethernet USB y/o verificar que funcione en otra máquina. Para configurar el adaptador de forma que use DHCP en el arranque y obtener una dirección IP, modificar /etc/network/interfaces y agregar las siguientes dos líneas entre las entradas “auto lo” y “auto eth0”:

```
auto eth2
iface eth2 inet dhcp
```

Utiliza el ID correcto del dispositivo de red que has localizado previamente a través ifconfig (en este caso, eth2). A continuación, apague el ODROID, introduzca el cable de red en el adaptador Ethernet USB y vuelva a encender de su nuevo su ODROID. Si por alguna razón, no detecta conexión prueba a desenchufar y enchufar de nuevo el cable y verifica que el adaptador Ethernet USB aparece en pantalla usando el comando “ifconfig-a” También se puede dar el caso de que el ID del dispositivo Ethernet haya cambiado si el adaptador a sido desconectado. En este caso debemos actualizar /etc/network/interfaces con la nueva información

Llegado a este punto el primer ODROID esta configurado y accesible en red. Antes de adéntranos en la Interfaz de Paso de Mensajes (MPI), configure el segundo ODROID siguiendo los mismos pasos utilizados para configurar el primer ODROID.

Interfaz de Paso de Mensajes(MPI)

Ahora que tenemos ambos nodos configurados correctamente. Podemos empezar a ejecutar trabajos HPC en nuestro clúster formado por dos máquinas. Un entorno de programación paralelo como MPI le facilitará esta labor, el cual se encarga de iniciar los procesos que conforman el modelo de programación paralelo. Proporciona una Interfaz de programación de aplicaciones (API) que permite la comunicación entre procesos secuenciales usados para desarrollar trabajos de programación. Para lograr esto vamos a utilizar la Interfaz de Paso de Mensajes (MPI), que proporciona una API que permite a los nodos enviar y recibir mensajes durante la ejecución de trabajos y/o tareas. El comando “mpirun” o “mpiexec” inicia todos los procesos necesarios para controlar tus dispositivos ODROID de forma segura. Podemos descargar cualquiera de las dos implementaciones MPI de

código abierto que existen actualmente: MPICH y OpenMPI. Para nuestro cluster ODROID en red ambas opciones son igual de validas.

Para instalar MPICH, ejecuta “sudo apt-get install mpich2”, y para instalar OpenMPI ejecuta “sudo apt-get install openmpi-bin”

Cosas que podemos hacer con MPI:

- 1) Realizar pruebas con aplicaciones que usen múltiples núcleos en un único ODROID
- 2) Ejecutar programas que utilice los 8 núcleos de nuestros dispositivos ODROID
- 3) Aprender a compilar tus propios programas MPI

En este artículo nos hemos centrado en los dos primeros usos. Puedes leer ejemplos de programas desarrollados con OpenMPI y MPICH para seguir aprendiendo. También existen excelentes tutoriales en Internet y numerosos libros de programación MPI como “Using MPI” de MIT Press (Uno de los coautor de este libro).

Mejorando el sistema

El contenido de este artículo representa una pequeña fracción de lo que podemos llegar a realizar con nuestro clúster. No obstante esta configuración es más que suficiente para hacer funcionar dos nodos con unos pocos usuarios. Si queremos que nuestro clúster crezca tendremos que establecer un nodo principal que nos permitirá manejar un mayor número de usuario y nodos conectados. Además de poder ocultar el tráfico del clúster del resto de la red, este nodo principal albergara los servicios que facilitarían la gestión del clúster, como LDAP para la gestión de documentos, Pupper para la gestión de contenido, NFS para compartir archivos, y otros servicios y herramientas de redes.

En la segunda entrega de esta serie de tutoriales convertiremos un ODROID en un nodo principal.

CONTROL DE ENERGIA EXHAUSTIVO EN CLUSTER ODROID

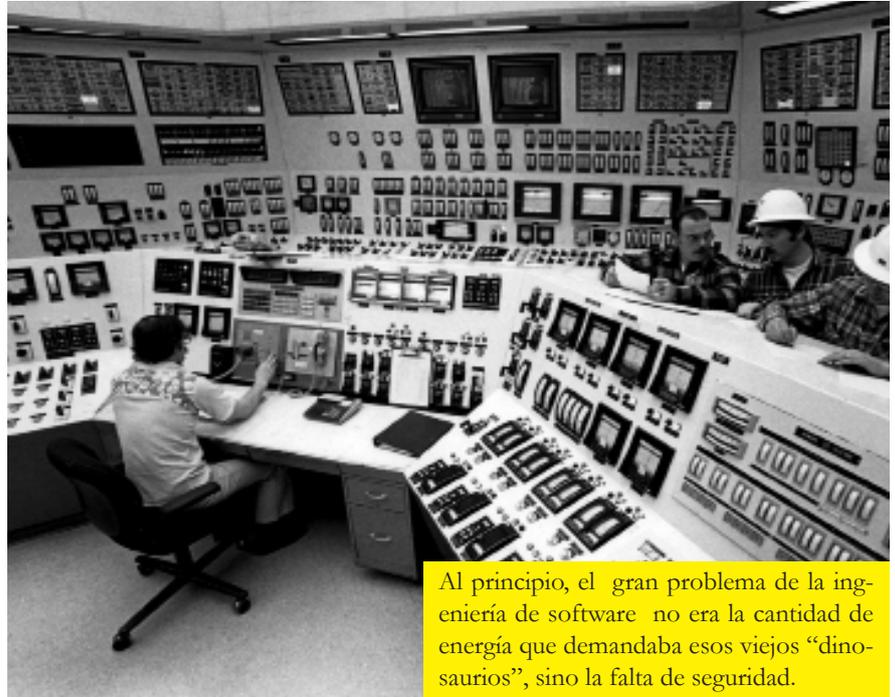
24 NUCLEOS DE ALTO
RENDIMIENTO POR
35 WATTS

Kurt Keville

Recientemente nuestro equipo ha experimentado con diversas técnicas pioneras de ahorro de energía que están a disposición de administradores de servidores y gestores de clúster de Informática de Alto Rendimiento (HPC). Nuestras experiencias vienen dadas por las pruebas realizadas a un mini-cluster de dispositivos ARM Cortex-A9 que ejecuta una versión de Ubuntu orientada a esta plataforma.

Además de la instalación recomendada por el equipo de Hardkernel del sistema operativo y suite de aplicaciones orientadas a servidores (Incluyendo MPI descrita en el anterior artículo). Hemos instalado cpufrequtils, WattsUp y PowerNap con las correspondientes actualizaciones. En este artículo nos vamos a centrar, principalmente en estos paquetes ya que nos han permitido alcanzar un mayor ahorro de energía.

El ahorro de energía es muy considerable en cluster que han reducido significativamente su tiempo de uso, que cuentan con un agresivo sistema de ahorro de energía sin que aparezcan efectos negativos en el resultado final. Incluso sin disponer de las apps (que no tenemos en nuestro cluster U2/U3) es posible ajustar los requisitos de rendimiento mediante reglas escritas frente a paquetes como Slurm, que actualmente es uno de los sistemas más populares en el mundo del HPC.



Al principio, el gran problema de la ingeniería de software no era la cantidad de energía que demandaba esos viejos “dinosaurios”, sino la falta de seguridad.

Grandes Cambios desde este proyecto

La Hibernación de Servidores es una utilidad en desuso desde que el apagado y el reinicio es más rápido. El uso de reguladores se ha convertido en un estándar

El regulador de ahorro de energía tiene un gran valor y debe ser usado por defecto en un clúster HPC

Los procesos manuales ya no son tan necesarios, si estas utilizando kernels modernos en tus ODROIDS.

Disponemos de un tiempo de arranque en U2 y U3 utilizando una eMMC de menos de 10 segundos, así que no hay razón para pensar que las posibilidades de suspender o hibernar pueda mejorar la simple acción de apagado o encendido del dispositivo.

En un centro de datos académico en el que es muy habitual la utilización de todos sus recursos en el trabajo diario, o

simplemente espera a que una tarea sea iniciada. Debido a prácticas convencionales sobre el uso de sistemas de colas y gestión por niveles, la mayoría de los sistemas pueden funcionar a tiempo completo si se organiza correctamente su carga de trabajo. Por ello, nosotros hemos analizado sistemas sin actividad, a pleno rendimiento y con un nivel de procesamiento mínimo, con el objetivo principal de ahorrar energía teniendo en cuenta todas las opciones disponibles a nuestro alcance y sin llegar a un apagado completo del sistema

El tiempo de arranque en frío de nuestro cluster de prueba era de menos de 10 segundos, por lo que verificamos este modelo. Un escenario similar a los centro de datos tradicionales que raramente se apagaban, ya que debían responder a solicitudes en un tiempo menor que el que necesitaban para iniciarse. Para esos escenarios probamos suspender y reanudar el sistema y los nuevos modelos PowerNap acpi.

Los administradores de centros de datos tienen una forma “segura” de detener sus servidores en caso de apagón eléctrico. El fabricante UPS fue líder en este campo cuando presento su línea “SmartUPS”. El software que se entregó con estos sistemas UPS usa una conexión Serie (RS - 232) que permitía iniciar una secuencia de cierre controlada cuando el servidor UPS no detecta corriente eléctrica. Desde su versión original el sistema ha ido incorporando mejoras que proporcionan una mayor flexibilidad y nuevas opciones de hardware como conexiones USB y SNMP.

La comunidad “open source” ha contribuido, de forma progresiva con nuevas opciones en forma de soporte para la base de código nut y upsd. (<http://www.networkupstools.org>). Ahora, podemos “ejecutar” este script definido por el usuario que, en nuestro caso está vinculado a un descenso en el uso del procesador por debajo del 20%. NUT es compatible con Ganglia y Nagios al ofrecer opciones de programación, lo que hace que los administradores estén familiarizados con la funcionalidad “ad-hoc” para que sus herramientas de gestión presenten mínimos problemas de integración con su configuración personalizada.

Además, “cpufrequtils” y “pmutils” son herramientas muy útiles que son compatibles con el kernel de Linux, al menos hasta la versión 3.2. PowerNap se maneja con una interfaz de comandos. Puedes bloquear tu placa en pleno funcionamiento ejecutando “cpufreq-set -g performance” desde la línea de comandos o ejecutar un script de configuración en PowerNap si quieres mayor nivel de automatización

Vea el resultado en U3 arriba en la parte derecha, cuando sus procesadores están configurados en modo “performance” (un reporte por nucleo).

Al configurar el procesador U3 a una frecuencia máxima de 1,7 Ghz y ejecutar la carga de trabajo <http://openbenchmark->

```
root@odroid:~# cpufreq-info
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0: driver: exynos_cpufreq

CPUs which run at the same hardware frequency: 0 1 2 3

CPUs which need to have their frequency coordinated by software:
0 1 2 3

maximum transition latency: 100.0 us. hardware limits: 200 MHz
- 2.00 GHz

available cpufreq governors: conservative, userspace,
powersave,performance

current policy: frequency should be within 200 MHz and 1.70 GHz.
The governor "performance" may decide which speed to use within
this range.

current CPU frequency is 1.70 GHz (asserted by call to hard-
ware).

cpufreq stats: 2.00 GHz:0.00%, 1.92 GHz:0.00%, 1.80 GHz:0.00%,
1.70 GHz:93.58%, 1.60 GHz:0.01%, 1.50 GHz:0.00%, 1.40 GHz:0.00%,
1.30 GHz:0.01%, 1.20 GHz:0.00%, 1.10 GHz:0.01%, 1000 MHz:0.00%,
900 MHz:0.00%, 800 MHz:0.00%, 700 MHz:0.00%, 600 MHz:0.00%, 500
MHz:0.00%, 400 MHz:0.00%, 300 MHz:0.01%, 200 MHz:6.38% (66)
```

ing.org/result/1401173-UT-1309189UT21, podemos controlar los procesadores a un nivel cercano a su máxima potencia alcanzando los 6,3 watts y un promedio de 5,6 por cada nodo durante el tiempo de la prueba. Es importante señalar que si bien este estándar logra mantener la CPU ocupada,

“cpufreq-info” se mostrará 4 veces, pero si buscas ahorro de energía, podrás ver los procesadores menos activos que cuando está sin uso.

El Comando top muestrea lu.A

de 4 procesos en función de si se inician como tareas MPI. Nosotros evaluamos el consumo de energía mediante

```
top - 09:00:58 up 2:52, 6 users, load average: 2.41, 0.74, 0.29
Tasks: 154 total, 2 running, 152 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.1 us, 0.0 sy, 0.0 ni, 0.8 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0
KiB Mem: 2071512 total, 884916 used, 1186596 free, 111884 buffers
KiB Swap: 0 total, 0 used, 0 free. 421392 cached Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
6306 root 20 0 69660 43248 644 R 394.9 2.1 3:59.17 lu.A
6311 root 20 0 4708 1236 836 R 0.7 0.1 0:00.11 top
```

no llega a utilizarse toda la memoria y por tanto no se refleja en el consumo de energía. Algunas pruebas de rendimiento pueden simular un uso de hasta el 400 % de un proceso en lugar del 100%

un FTDI conectado a un medidor de watts. Es evidente que el XU-E es más adecuado para llevar a cabo estas pruebas, por no mencionar que presenta un mejor rendimiento por su Arquitectura.

Config. Prueba

Queríamos un micro-cluster de prueba que capturara la energía usada por seis equipos ODROID -U2. Estos fueron conectados a nuestro medidor de potencia, que nos permitió medir las pequeñas fluctuaciones de energía que iban apareciendo. Capturamos los datos con una utilidad de Linux y además, medimos un solo nodo ODROID U3. El consumo de energía en el arranque (configurado en modo “ondemand”, utiliza sólo 200 MHz) fue insignificante. Mientras que el rango de 200 MHz a 1,7 GHz presentaban opciones más válidas para nuestra prueba (sin overclocking), en la que, por cuestiones prácticas únicamente utilizamos la velocidad más baja y más alta.

Medimos los siguientes escenarios: Todas las dispositivos a máximo rendimiento, todas las placas a mínima frecuencia con todos los núcleos activos, todas las placas con 2 CPU al mínimo y todas las placas con 1 CPU al máximo.

PowerNap controlaba la frecuencia de la CPU y la elevaba a la mínima velocidad permitida cuando detectaba un periodo de inactividad. Los periodos de inactividad pueden ser definidos por el usuarios. Permite, además controlar otros parámetros relacionados con la recogida de datos, periodos de inactividad y otras funcionalidades.

Nuestras recomendaciones pasan por ignorar la opción de hibernar en entornos de servidores. Si necesita desconectar el sistema de forma inmediata, proceda a apagarlo. El reinicio demora unos 10 segundos en cualquier de nuestras configuraciones. El U3 puede no competir con XU en FLOPs por watts, pero lo hace en coste. Es más barato que un “Intel Galileo” y ha obtenido buenos resultados en las pruebas de rendimiento realizadas.

Por otro lado, parece que la eficiencia de energía es proporcional a la frecuencia y número de núcleos. Así, si contáramos con un presupuesto de

Entorno Regulador versus Uso de watts

Regulador (freq)	Avg. Watts para 6 placas
Rendimiento (1.7 Ghz)	33.6
Espacio Usuario (700 Mhz)	21.0
Ahorro Energía w / 2 core*	3.6
Rendimiento w / 1 core	23.4

* via a PowerNap script that invokes "echo 0 > /sys/devices/system/cpu/cpu2/online" and cpu3. Actually, if you set any frequency below 700Mhz on the U3, it will turn off 2 cores.

Lower bars = less wattage usage



Clúter SoC Drawer y SoX BoX desmontados, pack asombroso

potencia con ciertas limitaciones en el tiempo, se podría crear un script que modificara la frecuencia de la CPU en determinados periodos de tiempo.

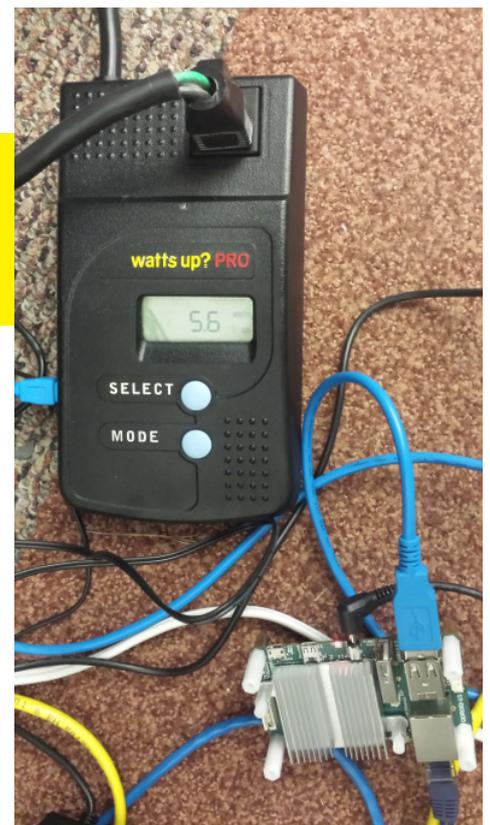
Configuración del registro de uso de energía en Odroid U3

Referencias

PowerNap Tutorial <http://tinyurl.com/PowerNapTutorial>

Slurm Integration <http://tinyurl.com/SlurmIntegration>

Power Use Logging <http://tinyurl.com/WattsUpLinuxUtil>





Atrás quedaron los días en los que los notebooks venían con puertos COM y Ethernet. ¿Qué puedes hacer si te quedas solo con un cable USB?

Cuando desarrollamos aplicaciones para ODROID, a menudo es necesario transferir archivos y comandos desde un PC principal al equipo de desarrollo. Esto se puede realizar via FTP o SSH a través del puerto Ethernet, pero también se puede hacer por USB simplificando el proceso. En este artículo vamos a describir cómo utilizar el puerto USB del ODROID para comunicarse con el PC principal. Para empezar, precisamos de un cable micro-USB para establecer la comunicación física entre ODROID y tu PC principal

Al conectar el ODROID a un PC utilizando el puerto USB, éste será detectado como otro dispositivo USB común, dependiendo del tipo de comunicación:

- Dispositivo de comunicación en Serie
- Dispositivo de red
- Dispositivo de almacenamiento masivo

También es posible utilizar el cable USB para emular cada una de estas conexiones de datos al mismo tiempo, como un enlace multifunction.

Instalación

Con el fin de utilizar los drivers del dispositivo, En primer lugar compruebe que los siguientes módulos están presentes en el sistema ODROID:

Emular Almacenamiento

```
/lib/modules/`uname -r`/kernel/
drivers/usb/gadget/
g_mass_storage.ko
```

Emular Serie

```
/lib/modules/`uname -r`/kernel/
drivers/usb/gadget/g_serial.ko
```

Emular Ethernet

```
/lib/modules/`uname -r`/kernel/
drivers/usb/gadget/g_ether.ko
```

Almacenamiento+Serie+Ethernet

```
/lib/modules/`uname -r`/kernel/
drivers/usb/gadget/g_multi.ko
```

Si no disponemos de los controladores apropiados, debemos actualizar el kernel y módulos de drivers. Escriba los siguientes comandos en Terminal:

```
$ mkdir update-kernel && cd
update-kernel

$ wget http://builder.mdrjr.
net/tools/kernel-update.sh

$ chmod +x kernel-update.sh

$ sudo ./kernel-update.sh
```

En los siguientes ejemplos de código, los caracteres verdes muestran la ventana de Terminal ejecutada en el ODROID y los caracteres azules muestran la Terminal en el PC principal

Emulando un dispositivo de almacenamiento masivo:

Almacenamiento sobre USB

Cargue el módulo “g_mass_storage” con un nodo de almacenamiento que será montado en el PC principal.

```
# mount
/dev/mmcblk0p1 on /media/
boot type vfat (rw,nosuid,no
dev,flush,umask=000)

# sudo umount /dev/mmcblk0p1

# sudo modprobe g_mass_stor-
age file=/dev/mmcblk0p1
```

Inicie “dmesg” y busque la función Mass_Storage en los mensajes del driver:

```
... gadget: Mass Storage Func-
tion, version: 2009/09/11
... gadget: Number of LUNs=1
... lun0: LUN: file: /dev/
mmcblk0p1
... gadget: Mass Storage Gad-
get, version: 2009/09/11
... gadget: userspace failed to
provide iSerialNumber
...gadget:g_mass_storage ready
... g_mass_storage
gadget: high-speed config #1:
Linux File-Backed Storage
```

Conecte el ODROID a su PC a través del cable micro-USB. Pasamos a utilizar el PC con Linux

En el PC principal con Linux:

Si la distribución de Linux instalada es reciente, el dispositivo debería ser montado automáticamente. De lo contrario, siga estos pasos para añadirlo manualmente.

Como usuario root, ejecute “dmesg” y busque el dispositivo /dev/sdX asignado al ODROID:

```
.. scsi 56:0:0:0: Direct-
Access Linux File-CD Gadget
0308 PQ: 0 ANSI: 2

.. sd 56:0:0:0: Attached scsi
generic sg2 type 0

.. sd 56:0:0:0: [sdc] 262144
512-byte logical blocks: (134
MB/128 MiB)

.. sd 56:0:0:0: [sdc] Write
Protect is off

.. sd 56:0:0:0: [sdc] Mode
Sense: 00 00 00 00

.. sd 56:0:0:0: [sdc] Asking
for cache data failed

.. sd 56:0:0:0: [sdc] As-
suming drive cache: write
through

.. sd 56:0:0:0: [sdc] Asking
for cache data failed

.. sd 56:0:0:0: [sdc] As-
suming drive cache: write
through

.. sdc:

.. sd 56:0:0:0: [sdc] Asking
for cache data failed

.. sd 56:0:0:0: [sdc] As-
suming drive cache: write
through
```

Luego, monte el dispositivo y podrá compartir archivos en ambos sistemas.

```
$ sudo mount /dev/sdc /mnt/
tmp

$ cd /mnt/tmp
```

Emular una conexión Serie:

Serie sobre USB

Cargue el módulo “g_serial” en ODROID:

```
# sudo modprobe g_serial
```

Inicie “dmesg” y busque la función serie entre los mensajes del driver:

```
# dmesg

... gadget: Gadget Serial v2.4
... gadget: g_serial ready

... g_serial gadget: high-speed
config #2: CDC ACM config

# ls -al /dev/ttyGS*

crw-rw---- 1 root dialout
248, 0 Dec 31 1999 /dev/
ttyGS0
```

Utilice el cable Micro-USB para conectar el PC principal con ODROID, observaremos que aparece el siguiente mensaje en el PC

```
$ dmesg

... usb 1-1.2.4: new high-speed
USB device number 3 using
ehci_hcd

... cdc_acm 1-1.2.4:2.0: This
device cannot do calls on its
own. It is not a modem.

... cdc_acm 1-1.2.4:2.0: tty-
ACM0: USB ACM device

$

$ ls -al /dev/ttyACM*

crw-rw---- 1 root dialout
166, 0 Jan 22 00:29 /dev/
ttyACM0
```

Los datos podrán ser transferidos entre ambos dispositivos: /dev/ttyACM0 (PC) y /dev/ttyGS0 (ODROID):

```
$ sudo chmod 666 /dev/tty-ACM0
$ echo qwerty > /dev/ttyACM0
# cat /dev/ttyGS0
qwerty
```

```
TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 tx-queuelen:1000
RX bytes:22645 (22.6 KB) TX bytes:18987 (18.9 KB)
```

```
HWaddr 62:19:ce:95:e0:9f
inet addr:192.168.100.1 Bcast:192.168.100.255 Mask:255.255.255.0
inet6 addr: fe80::6019:ceff:fe95:e09f/64 Scope:Link
```

Emulando una conexión Ethernet:

Ethernet sobre USB

Cargue el módulo “g_ether” en ODROID:

```
# sudo modprobe g_ether
```

Inicie “dmesg” para recuperar la función Ethernet entre los mensajes del controlador del dispositivo, así como los detalles de la conexión Ethernet.

```
# dmesg
... gadget: using random self ethernet address
... gadget: using random host ethernet address
... usb0: MAC e2:53:36:a5:8f:38
... usb0: HOST MAC 76:42:28:9e:9f:eb
... gadget: Ethernet Gadget, version: Memorial Day 2008
... gadget: g_ether ready
... g_ether gadget: high-speed config #1: CDC Ethernet (EEM)
# ifconfig
usb0 Link encap:Ethernet HWaddr e2:53:36:a5:8f:38
inet6 addr: fe80::e053:36ff:fea5:8f38/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:106 errors:0 dropped:0 overruns:0 frame:0
```

Conecte el cable Micro-USB entre el PC principal y ODROID y aparecerá el siguiente mensaje en el PC al escribir “dmesg”:

```
$ dmesg
... usb 1-1.2.4: new high-speed USB device number 12 using ehci_hcd
... hub 1-1.2:1.0: unable to enumerate USB device on port 4
... usb 1-1.2.4: new high-speed USB device number 13 using ehci_hcd
... cdc_eem 1-1.2.4:1.0: usb0: register 'cdc_eem' at usb-0000:00:1a.0-1.2.4, CDC EEM Device, 62:19:ce:95:e0:9f
$ ifconfig
usb0 Link encap:Ethernet HWaddr 62:19:ce:95:e0:9f
inet6 addr: fe80::6019:ceff:fe95:e09f/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3 errors:0 dropped:0 overruns:0 frame:0
TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 tx-queuelen:1000
RX bytes:488 (488.0 B) TX bytes:624 (624.0 B)
$ sudo ifconfig usb0 192.168.100.1
$ ifconfig
usb0 Link encap:Ethernet
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:21 errors:0 dropped:0 overruns:0 frame:0
TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 tx-queuelen:1000
RX bytes:4677 (4.6 KB) TX bytes:8394 (8.3 KB)
```

Ahora el ODROID será capaz de enviar y recibir datos a través del puerto usb0:

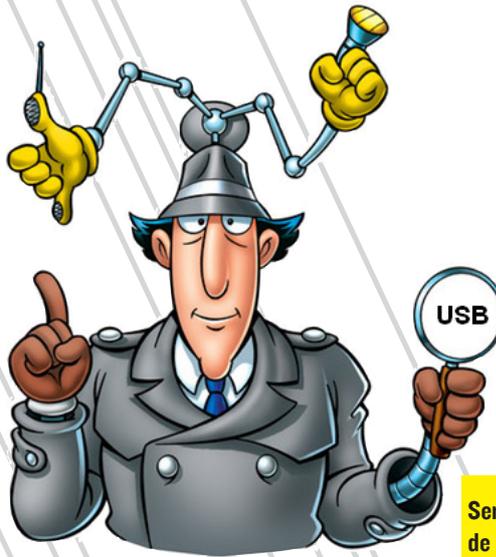
```
# sudo ifconfig usb0 192.168.100.2
# ifconfig
usb0 Link encap:Ethernet HWaddr e2:53:36:a5:8f:38
inet6 addr: fe80::e053:36ff:fea5:8f38/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:594 errors:0 dropped:0 overruns:0 frame:0
TX packets:434 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 tx-queuelen:1000
RX bytes:120002 (120.0 KB) TX bytes:98616 (98.6 KB)
```



Ethernet over USB



Serial over USB



Storage over USB

Ser capaz de trabajar en red, con sistemas de almacenamiento y en serie, te salvará la vida más amenudo de lo que imaginas.

```
PING 192.168.100.1
(192.168.100.1) 56(84) bytes
of data.

64 bytes from 192.168.100.1:
icmp_seq=1 ttl=64 time=0.348
ms

64 bytes from 192.168.100.1:
icmp_seq=2 ttl=64 time=0.254
ms

^C

--- 192.168.100.1 ping statis-
tics ---

2 packets transmitted, 2 re-
ceived, 0% packet loss, time
999ms

rtt      min/avg/max/mdev    =
0.254/0.301/0.348/0.047 ms

$ iperf -s

-----
-----

Server listening on TCP port
5001

TCP window size: 85.3 KByte
(default)

-----
-----

[ 4] local 192.168.100.1
port 5001 connected with
192.168.100.2 port 34764
```

```
[ ID] Interval      Transfer
Bandwidth

[ 4] 0.0-10.0 sec   193
MBytes  162 Mbits/sec

# iperf -c 192.168.100.1

-----
-----

Client connecting to
192.168.100.1, TCP port 5001

TCP window size: 20.7 KByte
(default)

-----
-----

[ 3] local 192.168.100.2
port 34764 connected with
192.168.100.1 port 5001

[ ID] Interval      Transfer
Bandwidth

[ 3] 0.0-10.0 sec   193
MBytes  162 Mbits/sec
```

Emulando un enlace multifunción:

El pack completo

Un enlace multifunción emula un dispositivo de almacenamiento masivo, una conexión serie y una ethernet utilizando sólo el módulo `g_multi`:

```
# mount

/dev/mmcblk0p1 on /media/
boot type vfat (rw,nosuid,no
dev,flush,umask=000)

# sudo umount /dev/mmcblk0p1

# sudo modprobe g_multi file=/
dev/mmcblk0p1
```

Después de conectar el cable Micro-USB del ODROID al PC, las tres funciones podrán ser utilizados simultáneamente.

¡Guau! 162Mbps es mas rápida que una Ethernet normal de 100Mbps!



JUEGOS LINUX EN ODRROID

LA MEJOR PLATAFORMA PARA TU JUEGOS

Tobias Schaaf

Cuando hablamos de videojuegos, son tantos los gustos como colores. El abanico de géneros es muy amplio: Primera persona, rol, arcade, aventura, simulación, estrategia, por mencionar algunos. Algunos están más orientados a ser utilizados en consolas mientras que otros requieren la utilización de un ratón y un teclado. Desde la plataforma ODRROID es posible ejecutar una gran variedad de videojuegos, permitiendo elegir entre diversos sistemas de juego dependiendo de tus preferencias personales.

Tu elección también dependerá de si te gusta jugar sólo (modo jugador único), con otros jugadores (modo equipo o competición), por Internet (juego en red), o con amigos en la misma habitación (modo multijugador en pantalla dividida). También dependerá de si optar por los clásicos videojuegos con sonido basado en simples pitidos, o si prefieres juegos más modernos con sonido estereo y gráficos en 3D.

Este mes te ofrecemos una visión general de las posibilidades que la placa ODRROID ofrece como plataforma de juego y te daremos algunas ideas interesantes que podrás poner en práctica.

Buscando el sistema adecuado

ODROID permite muchas posibilidades para emular y cargas videojuegos. A priori, resulta complicado decantarse entre uno u otro. Por ello vamos a evaluar diversos sistemas de juego

que ODRROID es capaz de emular y compararemos sus pros y contras para ayudarte a decidir cuál es la mejor opción. En este análisis se verán reflejadas mis opiniones personales que podrán ser distintas a las de otras personas, así que considera este artículo y su contenido como una simple sugerencia.

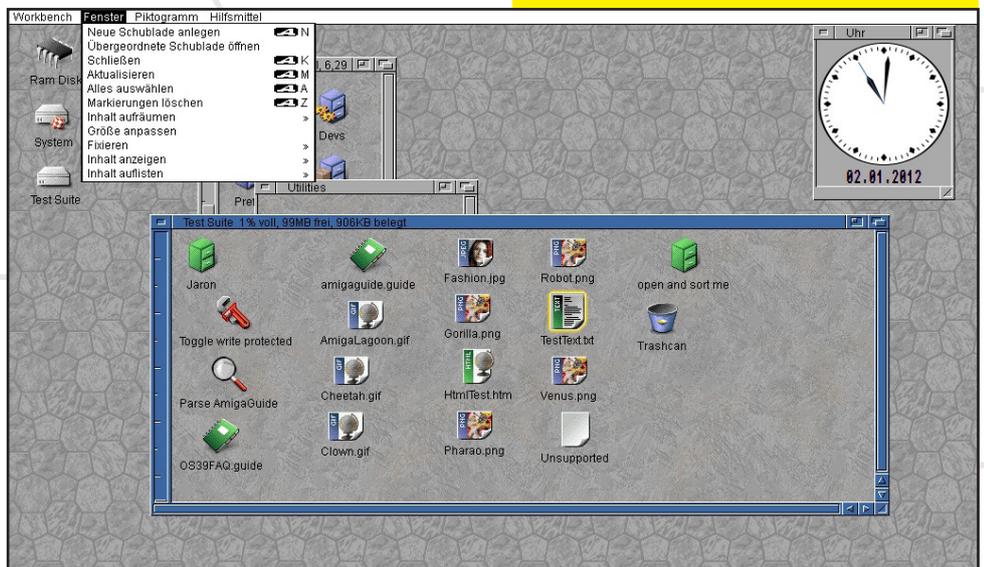
Si ya ha descargado e instalado la imagen GameStation Turbo para ODRROID disponible gratuitamente en los foros, te habrás hecho una idea de lo que sistem ODRROID es capaz de emular.

Empezamos por el primero de la lista: Amiga. En su día tuve varios PC Amiga y me encantaba jugar con ellos. En la década de los 90 fue muy popular. Comparado con otros sistemas de la época (Apple//GS, Color Macintosh y IBM PC) presentaba mejores gráficos y sonido que la mayoría de los PC de aquellos tiempos. Nunca he llegado a tener una consola real como una Super

Nintendo o Sega Genesis, así que no puedo comparar Amiga con estos sistemas de entretenimiento más modernos.

En mi opinión, Amiga estaba muy por delante a su tiempo. Con juegos

El escritorio Amiga, se convertía en la envidia de todos tus amigos. Te sentías como el chico mas "cool" de la escuela.





<-- Banshee, un gran Shooter Amiga
The Chaos engine -->

que dispone de juegos con lo que poder jugar con tus amigos. La considero un gran sistema multi-jugador. Juegos como "Banshee" o "The Chaos Engine" son muy buenos para compartir con ami-

SNES supuso un gran avance con respecto a NES y otras consolas. Sorprendente si comparando las especificaciones de las dos consolas: NES solo tenía 2 KB de RAM y 2KB de RAM de vídeo con un máximo de 25 colores simultáneamente (un total de 48 colores y algunos grises). Su sucesor, SNES, podría utilizar hasta colores de 15bit (32768 colores disponibles), con 128 KB de RAM y una gran cantidad de RAM de vídeo (64 kB de RAM principal, 512 + 32 bytes RAM sprite, 256 × 15 bits RAM palette). El tamaño de la ROM de los juegos en NES iban desde 8 KB a 1 MB. En SNES el tamaño estaba entre 256KB y 6MB.

de 15 o más discos, multitarea, capacidad para discos duros y con una interfaz grafica muy intuitiva.

Hay emuladores que pueden ejecutar juegos de Amiga, pero es necesario realizar varias configuraciones para que funcionen y con frecuencia debemos hacer cambios en "Kickstart" (BIOS de Amiga). Simplemente el hecho de tener que realizar continuamente estos cambios puede resultar algo molesto, aunque reconozco que los gráficos de Amiga dificilmente pueden ser comparados con consolas de hoy en día.

Utilizo retroarch de la imagen GameStation Turbo instalada en mi ODROID para ejecutar juegos de Amiga. Por desgracia, no todos funcionan. Considero que Amiga es inestable para ser utilizada en ODROID. Existen otros emuladores que se ejecutan directamente (Ej., E-UAE y FS-EAU) con mejores resultados que retroach, pero requieren configuraciones avanzadas y presentan problemas con Gamepads. En cambio retroarch funciona correctamente con Gamepads. Hay juegos muy buenos y muchos han sido trasladados a otras consolas o al PC que pueden ejecutarlos con mejores resultados.

En realidad, lo bueno de Amiga es

gos.

Los siguientes en la lista son Nintendo Entertainment System (NES), SuperNintendo Entertainment System (SNES) y la GameBoy Advance (GBA). Si escuchas a la gente hablar de Nintendo y de juegos retro, es que se están refiriendo a las consolas NES y SNES. NES es una de las consolas más antiguas, pero introdujo juegos clásicos que han sido remasterizados con el paso del tiempo. Títulos como "Legend of Zelda" o "Super Mario" son juegos que nunca se olvidaran y representan una de las más importantes licencias de Nintendo.

SNES introdujo varias mejoras con respecto a NES, mayor número de colores y ofrecía mas posibilidades de juego. Se convertido durante mucho tiempo en la consola por excelencia para muchos, incluso para jugadores profesionales.

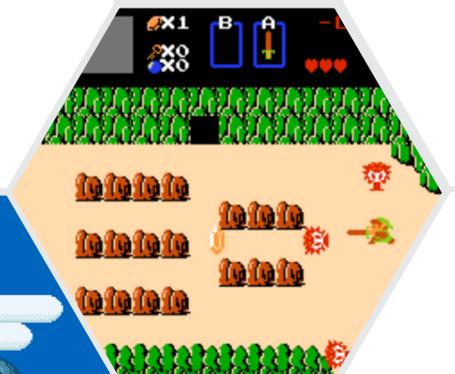


Era posible introducir muchos más datos en un cartucho de SNES, lo que permitía más contenido y desarrollar mejores juegos, tal y como se aprecia en la comparación entre NES y SNES:

SNES también utilizaba capas de vídeo permitiendo más efectos y contenido en una única escena, y contaba con una resolución más alta que su predecesor.



Zelda, Mario and Kirby, Los mejores juegos de la edad de oro de Nintendo



Otra ventaja de estas consolas es que se hicieron con dos mandos, lo que dio lugar a la creación de una gran biblioteca de videojuegos en la que amigos podían jugar bien en modo cooperación o compitiendo entre ellos mismos. Personalmente, considero que SNES es la mejor consola para videojuegos multijugador.

En mi opinión, GBA se suele pasar por alto al hablar de juegos retro y Nintendo. GBA fue la consola portátil de Nintendo, muy parecida a la GameBoy (GB) y GameBoy Color (GBC), que era compatible con los juegos de GB y GBC. Incluso tenía un segundo procesador que otorgaba mayor compatibilidad, así los usuarios de GameBoy podían lanzar sus viejos juegos en la consola más reciente. A parte de eso era muy parecida a la SNES, lo que hizo que se convirtiera en un dispositivo portátil de entretenimiento referente en su época.

La GBA cuenta con 15 bits de color y CPU de 32 bits, en comparación con el procesador de 16 bits de la SNES. Ofrecía 32KB + 96KB de RAM de vídeo (interna en la CPU) y 256 KB WRAM (fuera de la CPU).

Si comparamos sus características técnicas, GBA es mejor sistema que SNES en casi todos los aspectos. La única ventaja que SNES tenía sobre GBA era la resolución, que era de 240x160 píxeles. Aunque la resolución era más baja, el tamaño de los cartuchos GBA tenían entre 1 y 32 MB.

**Crew: Captain
we are about
Cecil: Good.**



Más tamaño es igual a más contenido y una vez más, GBA presentaba una gran mejora en cuanto a contenido. Ofrecía pequeñas escenas de vídeo, música y gráficos muy buenos. Son las razones por las que yo prefiero emular GBA sobre SNES. Muchos juegos que existían en SNES fueron trasladados a GBA y no sólo se veían mejor, sino que presentaban contenido adicional.

Aunque la GBA es, en mi opinión mejor consola que SNES, tenía una gran desventaja. La GBA era un dispositivo portátil, que aunque tenía función multijugador conectando las consolas a través de un cable, se hacía imposible jugar a un videojuego con un amigo en la misma máquina.

Sin embargo, si eres de los que juegas en solitario, debería echar un buen vistazo a los juegos de GBA. También puede comprobar si tus juegos favoritos de SNES se trasladaron a GBA y de esta forma conseguir una experiencia mejorada del juego en sí. Si estás interesado en probar un nuevo juego, te recomiendo ojear la biblioteca de juegos de GBA.

En su conjunto me gusta bastante

**Final FantasyIV
SNES vs GBA**



Annation!



**GBA's
Riviera
Promised
Land**

GBA y habitualmente juego con ella, ya que presenta unos gráficos muy buenos y una gran jugabilidad. Todos los sistemas de Nintendo (GB, GBC, GBA, NES, SNES) funcionan correctamente en ODROID usando los emuladores que contiene la imagen GameStation Turbo.

Yo prefiero ejecutar SNES en retroarch y jugar al resto de emuladores de Nintendo utilizando Mednafen ya que ejecuta mejor los juegos de esas plataformas. Además permite la integración de escaladores de software como hq2x o super2xsai, lo que hace mejorar la calidad de imagen y hace que el juego parezca más “moderno” de lo que realmente es.

Si eres un fan de máquinas recreativas, probablemente disfrutarás con MAME, NeoGeo y NeoGeo Pocket, que están disponibles para el ODROID. MAME y NeoGeo están disponibles en retroarch y NeoGeo Pocket está disponible en Mednafen.

NeoGeo Pocket es similar a la GBA, se podría situar entre GBC y GBA, pero no era tan popular y no cuenta con tantos juegos como sus competidores. Aún así, es muy similar al resto de máquinas Arcade ofreciendo algunos juegos de acción interesantes.



GBA's
Summon
Night
Swordcraft
Guard

MAME y NeoGeo presenta una gran cantidad de juegos de acción como Street Fighter 3, 3rd Strike y 1944 The Loop Master.

Los juegos Arcade varían desde aquellos que presentan gráficos muy simples, graficos en 3D, o



con tecnología "sprite" como es el caso del mencionado 1944 TheLoop Master. Si te gusta bastante la acción y eres un jugador experimentado ODDROID es lo que estás buscando, especialmente si juegas con tus amigos.

Sin embargo, el género Arcade está limitado a juegos de acción. Rara vez encuentras juegos de rol, aventuras o simulación en una máquina recreativa.

Resulta difícil comparar los juegos Arcade con SNES o Sega, ya que los estos han evolución bastante y lo mismo ha hecho el hardware en el que se apoya. Algunos juegos Arcade presentan gráficos de calidad NES mientras que otros

tienen una resolución tan buena como una PlayStation.

ODROID puede ejecutar la mayoría de estos juegos , y por ello merece la pena hacerse con una de estas plataformas. Incluso existe un proyecto documentado en el foro en el que alguien ha desarrollado su propia maquina recreativa con un U2: <http://bit.ly/1cmrgjK>.

Otra de las grandes compañías de consolas de juegos fue Sega, que compitió directamente con Nintendo. El equipo de Sega tenía muy buenas ideas, pero no se llevaba muy bien con los desarrolladores y editores. Ahora sólo produce juegos para otras consolas.

La Sega MasterSystem, Sega Genesis (Mega Drive) y la Sega GameGear son los emuladores que he incluido



Street Fighter 3rd Strike
1944 The Loop Master
(NeoGeo)
Sonic the Hedgehog
Alex Kidd in Miracle World
(Sega Master System)
Double Dragon
(NES)
Double Dragon
(SMS)

en la imagen GameStation turbo para ODDROID, ya que son sus consolas más conocidas. Crearon otros sistemas como la Sega 32X , Sega Mega - C D, Sega Saturn y la Sega Dreamcast , que en realidad no eran más que consolas con un aspecto llamativo, pero resulta difícil encontrar emuladores para ejecutar estos sistemas en Linux y ARM por lo que no se incluyen en la imagen, todavía.

Si comparas los diferentes sistemas de Sega con otros de su tiempo,

llegarás a la conclusión de que la Sega Master System es comparable a NES, la Sega Genesis es similar a SNES y Sega GameGear era parecida a GBA o GBC.

Sega siempre han intentado superar a las consolas de Nintendo en prestaciones. De hecho, Sega Master System tenía más RAM y más colores que NES y era, ligeramente más rápida. Su juego



más popular no fue Sonic the Hedgehog, sino Alex Kidd in Miracle World. Sin embargo, Sonic fue la respuesta de Sega a Mario de Nintendo y su estrategia tuvo muy buenos resultados.

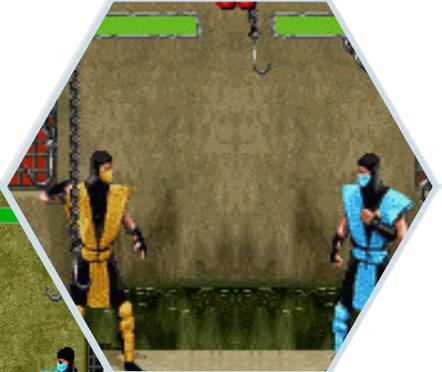
Si comparamos los juegos disponibles para NES y Sega Master System puedes apreciar que Sega es bastante mejor que NES. Si tu juego favorito de NES también está dis-



mente, ilustradores de Disney participaron en la creación de animaciones en la versión de Sega Genesis, que le otorgo una mejor apariencia.

Sega intentó llegar a jugadores más adultos, así como llamar la atención de jugadores profesional. Para ello, publicó la versión no censurada de Mortal Kombat que contaba con más sangre y otros efectos que se habían retirado de la versión para SNES.

sobre 300 juegos para la GameGear. Si continuamos en el mundo de Sega, es una buena ocasión para conocer esta consola. Incluso algunos de los Juegos



Compara los gráficos de SNES y Genesis. Pudes decir a que sistema corresponden cada uno?
Earthworm Jim 2
Aladdin vs Aladdin
Mortal Kombat

ponible para Sega Master System, te recomiendo jugar a la versión de Sega.

La siguiente generación, Sega Genesis no era tan buena como SNES, ya que presentaba menos colores y los desarrolladores tuvieron dificultades

Génesis también publicó muchos juegos de deportes que eran muy popular entre la gente mayor.

En su conjunto, SNES tenían mejor hardware y los juegos te dejaban más impresionado que en la versión de Sega Genesis, pero Génesis intentó contrarrestar

de Genesis se publicaron para GameGear, por lo que podemos darle una oportunidad.

La siguiente consola que tuvo un impacto significativo en el mercado fue Sony con PlayStation, aunque ya considerada una consola retro. Tiene una gran ventaja sobre las consolas mencionadas, ya que marca una nueva generación aunque ofrece juegos clásicos que fueron remasterizados.

En cuanto a los juegos podemos decir que tienen muy buenas escenas de video, una trama elaborada y gráficos de alta definición. La PS1 supera con creces a todas las consolas anteriores, y si localizas juegos desarrollados para SNES o Sega Genesis, te darás cuenta que en su versión para PS1 aparece con un montón de contenidos adicionales, escenas de video y mejor banda sonora.

Aunque el sistema retroarch para PS1 utiliza un software adaptado, los juegos se ejecutan con cierta fluidez en ODROID. Clásicos como Final Fantasy 7, 8 y 9 son algunos ejemplos en los que podemos observar esos gráficos de alta definición. PS1 fue diseñada para el modo multijugador, lo que significa que podemos jugar con amigos en el mismo televisor.

Por último, llegamos al emulador más impresionante de ODROID, el PPSSPP que es capaz de ejecutar



este aspecto con una mecánica de juego más elaborada.

La Sega GameGear pretendía ser un dispositivo portátil mejorado y el competidor directo de la GameBoy (GB), pero era mucho más que eso. De hecho, la Sega GameGear fue, incluso más potente que la Sega Master System y contaba con un adaptador para ver la TV.

La Sega GameGear se encuentra en algún lugar entre el Sega MasterSystem y la Sega Genesis. Disponía de más colores que la Génesis, pero su desventaja era su baja resolución, incluso para su época. Punto que no era tan importante al tratarse de una consola portátil. Había

para crear juegos desde que Nintendo prohibió que sus empresas pudiera desarrollar juegos para cualquier consola que no fuese Nintendo. Aún así, había bastantes juegos disponibles para Génesis y parecen bastante aceptables.

Aunque los juegos presentaban más detalles en su versión para SNES, Sega Génesis disponía de mejor mecánica de juego incluyendo “una espada”, no disponible en la versión de SNES. Curiosa-

juegos de Play-Station Portable (PSP) en nuestro ODROID. La PSP cuenta con una gran colección de juegos que ofrecen graficos en 3D, con escenas de video y voces muy reales. Aunque PPSSPP presenta mejor rendimiento en un PC con Windows, Linux o Android, también lo he podido arrancar en ODROID. Hay numerosos juegos de PSP que se ejecutan correctamente en ODROID en un entorno X11.



La Edad de Sony, Final Fantasy VIII de PS One, y juegos de PSP, Soul Calibur, Fifa Soccer 2011, Wipeout Pure y Naruto Shippuden estan al alcance de tu ODROID

¿QUÉ EMULADOR Y QUE SISTEMA ES EL MEJOR?

Si comparamos los diferentes sistemas por su capacidad para hacernos ver y sentir, considerando la profundidad de color, contenido del juego, calidad de sónico y hardware utilizado, podemos enumerar los sistemas de menor a mayor teniendo en cuenta el potencial que nos ofrecen.

gran variedad de juegos desde clásico Pac-Man al Tekken 6 para la PS1. De todas estas consolas me quedo con NeoGeo, ya que sus juegos te hacen ver y sentir una mejor experiencia de juego, en mi opinión.

Otra forma de decidir qué juegos ejecutar es ver si los emuladores ofrece

y que normalmente carecen de la posibilidad de jugar con otras personas, a menos que se conecten a través de Internet o en una red doméstica con ordenadores.

Cada sistema analizado en este artículo tiene su propio encanto. Todo ellos tienen juegos únicos e in-

GB < GBC < NES < SMS < GAMEGEAR < GENESIS < SNES < GBA < PSI < PSP

No se incluye en este ranking Amiga, que se trata más bien de PC de su época a pesar de que ofrecía una gran variedad de juegos y opciones. Juegos desde los más simples a los más espectaculares, por su sistema de sonido y configuración de colores, incluso mayor a la ofrecía por una SNES o GBA llegado hasta los 24 bits

También he omitido las máquinas de recreativas como NeoGeo Pocket, NeoGeo y MAME, que ofrecían una

la posibilidad de jugar con otras personas frente al Televisor. Para juegos multijugador las mejores consolas son MAME, NeoGeo, SNES y la PS1. También existen juegos multijugador para Sega Master System y Sega Génesis, pero me resulto difícil hacerme con ellos. Recomiendo SNES como mejor opción para jugar con tus amigos.

Los dispositivos portátiles tales como la GBA es la mejor opción para aquellos que suelen jugar en solitario

teresantes mas por la experiencia de juegos que por los graficos en sí, lo que hace que cada juego en particular valga la pena probar. Con ODROID tienela posibilidad de experimentar con todode ellos cambiando simplemente de un emulador a otro. Y con GameStation Turbo, tienes en tu mano todo lo que necesitas para jugar con tu mando favorito.

EVALUANDO LA INTERFERENCIA DE COMUNICACION

CON JAVA MULTI-THREADED

Jussi Opas

En este proyecto hemos intentando realizar una estimación de la interferencia de comunicación usando Java “multi-threaded” (múltiples hilos de ejecución). ¿ODROID, en comparación con otros sistemas puede hacer el mismo trabajo?

Si algunas de las tareas se pueden ejecutar en paralelo, cada “thread” (hijo de ejecución) adicional recorta el tiempo de ejecución de todo el trabajo. Puesto que hay un proceso de ejecución para cada hilo que se añade, vamos a estimar que se necesita 8 unidades para realizar todo el trabajo de forma secuencial. Añadiendo un nuevo hilo se acortará el tiempo en 4 unidades, mientras que al añadir un tercer hilo se recortará 3.67 unidades y con un cuarto disminuirá el tiempo en 2 unidades. Si el trabajo no se ejecuta en paralelo completamente, siempre habrá una parte que debemos hacer en secuencia por hilo y este trabajo no podrá acelerarse con hilos adicionales.

Empezamos con la suposición de que el trabajo no ejecutado en paralelo requiere de 2 unidades y se supone que agregar más hilos no representa un coste adicional. Entonces, agregamos hasta 12 hilos. Todo el trabajo debe hacerse en un periodo de tiempo constante desde el 4º al 12º hilo. Con esta información podemos hacer una primera estimación, tal y como se muestra en la imagen de la derecha.

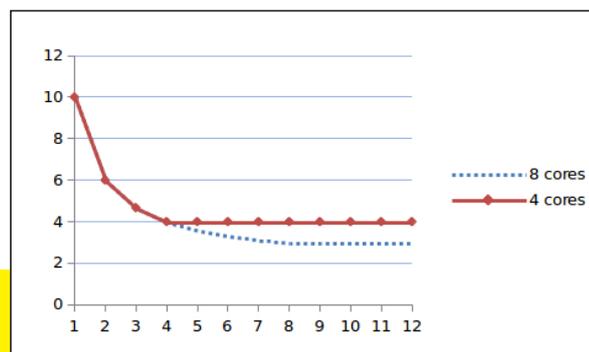
Modelo expectativa: Curvas de rendimiento teórico de procesadores con varios núcleos

Esto es interesante para comprobar que al añadir más núcleos se acorta el tiempo, pero no demasiado, tal y como muestra la línea azul discontinua. El trabajo de dos unidades en forma secuencial mejora el rendimiento general. Si el objetivo es buscar el beneficio de todos los núcleos, debemos, previamente ejecutar en paralelo el trabajo secuencial. El tiempo de compilación del kernel Linux en ODROID XU demuestra que este modelo es totalmente válido.

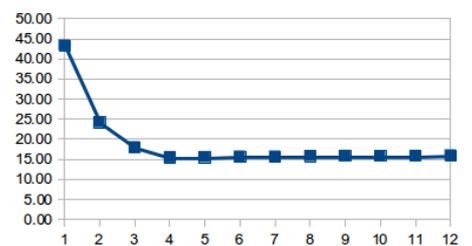
Nuestra aplicación de ejemplo, “Poiju” calcula la estimación de interferencia de comunicación (ver imagen pag siguiente). Para realizar la estimación, es necesario calcular el área de servicio y cobertura de cada célula. De esta forma, en cada píxel, se calcula la intensidad del campo de las células interferidas. Esto se hace sobre el área de servicio de las células portadoras. El cálculo es realizado en píxeles de un tamaño entre 30 y 350 metros. El tamaño de las células de radio varían entre los 200 metros y 50 km.

La dificultad de cálculo tiene una naturaleza exponencial, la cantidad de píxeles a procesar crece por segundos cuando más precisa es la resolución utilizada.

La particularidad del programa es



Linux kernel compilation with many threads [min]

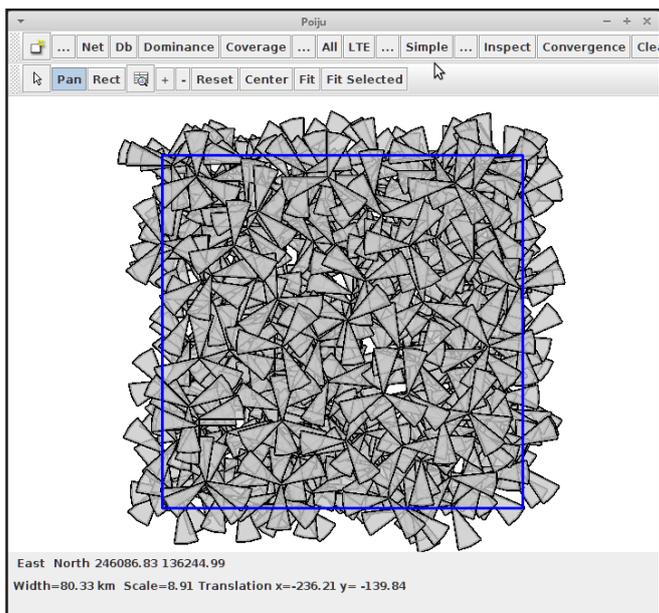


Tiempo de compilación (en minutos) del kernel Linux en XU

que solo utiliza memoria RAM (sin acceso a archivos) y todo el trabajo puede ser realizado por la CPU, ya que la GPU no realiza ninguna tarea. La ejecución en paralelo ha sido diseñada de manera que, para cada célula, el área de servicio se calcula en primer lugar de forma secuencial, y después esa interferencia es calculada con varios hilos por candidatos más cercanos a la célula.

Los núcleos físicos y/o los hiper-hilos son visibles como procesos activos. [LEA99]. Los hilos Java quedan ocultos bajo el sistema operativo y hardware, por lo que el mismo programa se puede usar en todas las plataformas que disponga de una máquina virtual. Para implementar hilos hemos usado un paquete moderno de Java.

Puedes ver en la siguiente página una sesión típica de desarrollo con la aplicación Poiju. La Leyenda de parámetros se muestra en la parte frontal, debajo de la ventana principal de la aplicación Poiju. También se muestran el terminal http y el



entorno de desarrollo Iltellij IDEA situado en la parte inferior.

El procesador ARM del ODROID-XU ofrece 4 núcleos A7 de gran potencia y 4 núcleos A15 de alto rendimiento [ARM13]. En cualquier momento, se puede poner en marcha el cluster de pequeños núcleos o el cluster de grandes núcleos, conocido como Arquitectura “big.LITTLE”. De modo que haya 4 procesos disponibles. La aplicación Poiju solo utiliza los grandes núcleos para realizar cálculos de forma constante.

El rendimiento de aplicación es medido por el mismo sistema por medio del periodo de tiempo que el núcleo necesita para realizar alguna tarea hasta que finaliza la misma.

Hemos realizado pruebas con 10.000 células en cada momento. El resultado de la medición de rendimiento con OpenJDK y OracleJDK se puede

```
final long t1 = System.nanoTime();

final double distance = SpatialMath.computeProjectedDistance(carrier, candidate);

instrument.
```

ver abajo a la derecha .

Observamos que el modelo teórico al que hemos hecho referencia, está siendo seguido parcialmente. Cada hilo agregado esta acelerando la ejecución del programa, hasta

el cuarto hilo. El primer hilo añadido optinene el mejor rendimiento. Este está parcialmente afectado por la compilación Just In Time (JIT) que mejora el funcionamiento de ejecución cuando se ejecuta el mismo código en repetidas ocasiones.

Al añadir el 3º y 4º hilo, previsible mejora el rendimiento, mientras que al añadir el 5º disminuye. Suponemos que la programación del hilo y el cambio del contexto no estan libres, cuando hay muchos hilos esperando ser ejecutados. Cuando nos dimos cuenta de esta cuestión, lo mejor era limitar el número de hilos a 4 en el programa de ejecución. Eso se puede hacer a través del método Runtime.getRuntime().availableProcessors().

También observamos que OracleJDK tiene mejor funcionamiento que OpenJDK. OpenJDK se encuentra

instalador por defecto, Java en Ubuntu. Por lo que el usuario debe instalar manualmente Oracle JDK. Afortunadamente, está a libre disposición desde el sitio web de Oracle Developer .

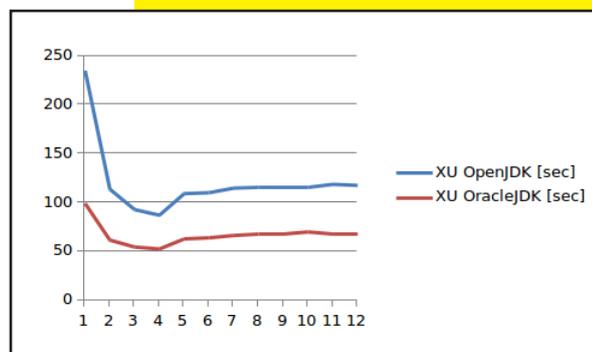
El rendimiento

de la aplicación en XU es prácticamente idéntica entre las 4 distribuciones de Linux: Debian, Xubuntu, Fedora y Ubuntu con el escritorio Mate. Existe una pequeña pero

visible mejora en la versión liberada de JDK 1.6, 1.7, y 1.8. IIT, que mejora el rendimiento cuando la frecuencia pasa de 1,6 a 1,8 GHz, como se muestra al final de esta página.

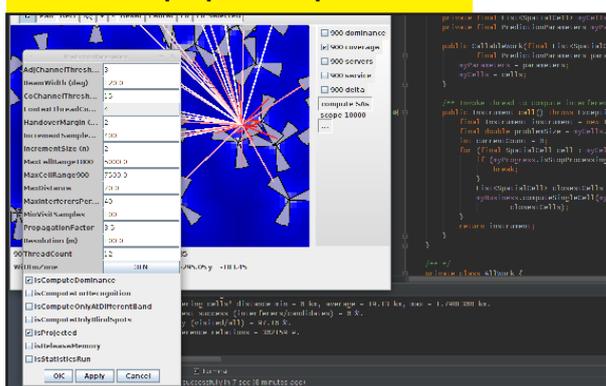
Con htop y vmstat vemos que más

Periodo de ejecución en 4 núcleos A15 de XU en dos JDK diferentes.

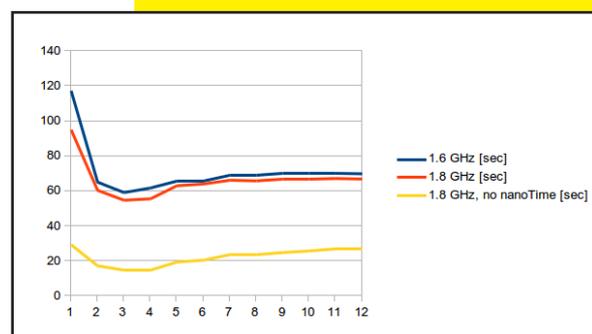


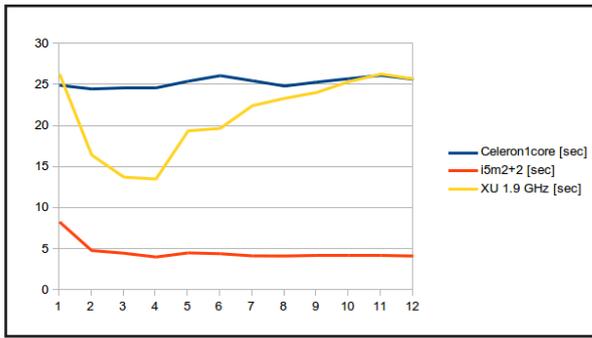
de la mitad del tiempo es utilizado por el núcleo en lugar del espacio de usuario. Este es un problema importante que nos dimos cuenta sólo en XU. ¿Qué es lo que falla en la aplicación? Si hay algún error interno en un programa y se ejecutan gran cantidad de excepciones, afectaría al rendimiento de manera significativa. Sin embargo, eso no es visible durante el tiempo que esta trabajando el kernel. Después de varias conjeturas y juicios que no nos llevaron a ninguna

Sesión desarrollo en Poiju en ODROID XU. Motinor por puerto de pantalla.



Over-clocking mejora el rendimiento, pero el descenso en el uso de la instrumentación lo mejora significativamente





parte. Finalmente, modificamos una serie de instrucciones que habían sido generadas por el método “nanoTime” de Java. Como resultado, desaparecieron dos tercios del tiempo de cálculo.

Esperamos que la ineficiencia de “nanoTime” y cualquier característica similar de Java en ARM se haga más útil en el futuro, ya que se escuchan noticias en este sentido. [Jac13]. Desde que utilizamos el procesador ARM para que hacer el trabajo que tradicionalmente ha sido realizado por ordenadores portátiles, hemos realizado una comparación con dos portátiles. Uno de ellos con un procesador Celeron a 2GHz y otro con un i5-2520M a 2.5GHz. El Celeron cuenta con un núcleo, mientras que el i5m presenta 2 núcleos con 2 hiperhilos. Nos referimos a estos procesadores como x64 (de 64 bits) y x86. La comparación se puede ver en la imagen de arriba.

Con esta diferente de escala en el gráfico es obvio que algo empieza a limitar el rendimiento del quinto hilo. Agregando más hilos hasta 12, no parece tener problemas de funcionamiento el núcleo Celeron. Por otro lado, vemos que el funcionamiento de la aplicación en XU esta perdiendo valor cuando se utilizan

5 y más canales. Con la salida vmstat podemos crear un gráfico, donde aparecen el tiempo de uso, el tiempo de inactividad y el rendimiento. Vease la imagen de la parte inferior izquierda.

Desconocemos la razón real de este comportamiento, pero podemos realizar algu-

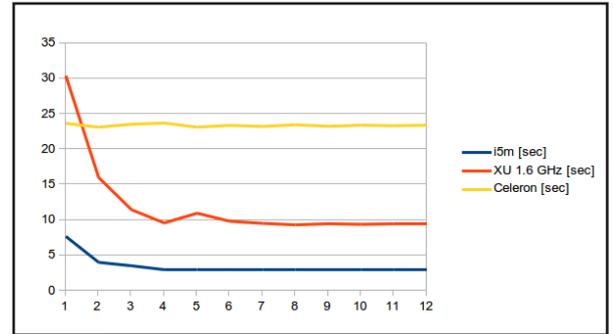
nas conjeturas como que el cambio de contexto no es eficiente, la carga de trabajo no es uniforme, o que la parte de trabajo secuencial es demasiado amplia. La primera suposición se puede solucionar con el siguiente razonamiento: 1) no hay problemas en la compilación del kernel de Linux con múltiples hilos, 2) Según ciertos estudios el tiempo de cambios de contexto debe ser como máximo el 0,4% del tiempo total [dav07], y 3) el entorno de desarrollo -IntelliJ IDEA- utiliza múltiples hilos sin fallos. Esta conducta es un rompecabezas que todavía no ha sido resuelto.

Con un enfoque alternativo, implementamos hilos de forma distinta y también hicimos un par de cambios más. Ahora, las células se organizan en tantas listas de trabajo como hilos son usados. Ambas áreas de servicio e interferencias se calculan en paralelo, así hay N células son calculadas al mismo tiempo. El trabajo es ejecuta totalmente, o casi totalmente en paralelo y en el rendimiento de la

aplicación se observa claramente una mejoría. El gráfico de Rendimiento sigue nuestro modelo. El sistema multi-hilo usa toda la capacidad de la placa XU, como se muestra en la Figura 9. En esta comparación, se utilizó el ODROID XU a 1,6 GHz.

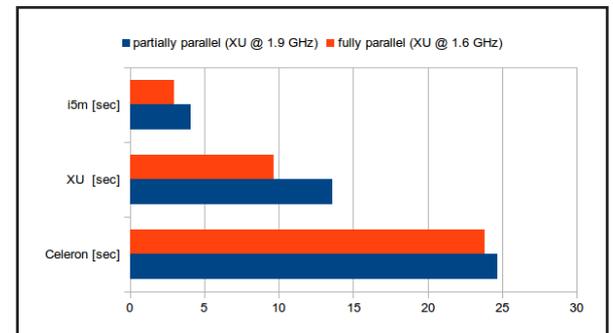
Para hacer una comparación más sencilla. Empezaremos en el momento del 4º hilo y colocaremos un número de ejecuciones paralelas parciales y totales

En esta prueba, el procesador multi-core ARM está realizando sus trabajos mucho más rápido que el procesador Celeron de un núcleo. Con 4 hilos, el rendimiento es 1,8 veces mejor. Mientras tanto, el procesador i5



Perfil de rendimiento cuando la aplicación se ejecuta en paralelo completamente.

Comparación de i5, XU y Celeron con 4 hilos en ejecución.

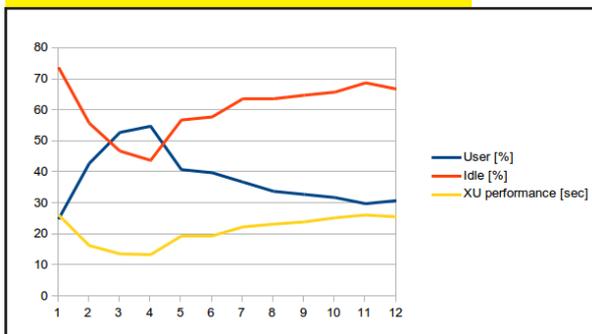


se mueve a 3.3 veces más rápido que el XU. La Configuración de hilo en paralelo cambia estos números a 2.5x y 3.2x respectivamente. En esta comparación, el XU hace 1/3 del trabajo que el i5 hace en el mismo periodo de tiempo.

Resumen

La placa ODROID XU puede realizar las mismas tareas de Java que los ordenadores portátiles utilizados tradicionalmente. Aunque no puede competir con nuevas marcas (y más caras) de procesadores x64, que puede ser utilizado como sustitutos a los i5. El ODROID-XU es claramente mejor que los viejos ordena-

Procesador no esta cargado, hay capacidad libre



El valor por defecto "on-demand" en el regulador de Linux afecta a los resultados de la prueba de rendimiento, debido a que la frecuencia del reloj de la CPU se altera automáticamente. Por lo tanto, en Linux, se debe utilizar el regulador ajustándose a la frecuencia de rendimiento durante las pruebas.

El método System.nanoTime consume mucho tiempo del kernel. El problema del método nanoTime afecta a los procesadores Celeron y ARM, mientras que el procesador mobile i5 no sufre tanto. La llamada a nanoTime debe comentarse en el código de producción.

Cuando la intensa carga de trabajo es ofrecida a hilos paralelos, la cantidad de hilos es igual al número de procesadores disponibles usados. El método Runtime.getRuntime().availableProcessors() se puede utilizar para este propósito.

Utilizar herramientas de Linux como vmstat y htop para validar el tiempo de carga y del núcleo. Para perfilar una aplicación puede usarse jvisualvm en el directorio bin del JDK. No confíe en la instrumentación sin una herramienta de Backup.

dores x64 de un solo núcleo y las aplicaciones Java se benefician significativamente del uso de múltiples hilos.

A través de este ejercicio, el rendimiento del Poiju en otras plataformas mejoró un 40%. Solo en ODDROID-XU, el avance fue enorme desde su diseño inicial. En general, las aplicaciones Java de escritorio también puede mejorar cuando se ejecutan en el XU basado en ARM. El ODDROID calcula la estimación de interferencia de un radio de 10,000 células en 9,6 segundos utilizando pixel basados java multihilo. Este ejercicio se ha realizado como un adhoc, Proyecto de tipo BYOD. Su contenido, como tal, no implica productos reales de NSN.

versity of Illinois at Urbana-Champaign 201 N Goodwin Ave Urbana, IL. 5 pp. 2007. <http://www.cs.huji.ac.il/~feit/exp/expcs07/papers/136.pdf>

[LEA99] Lea Doug. Concurrent Programming in Java, Design Principles and Patterns, Second Edition. Addison Wesley. ISBN 0-201-31009-0. 411 pages. November 1999.

[ARM13] ARM Limited. Multithreading technology and the challenges of meeting performance and power consumption demands for mobile applications. 9 pages, 2013.

[JAC13] Jackson Joab. Oracle and ARM to tweak Java. 2 pages. 2013. <http://www.java-world.com/article/2078833/enterprise-java/oracle-and-arm-to-tweak-java.html>

Referencias

[DAV07] David F. M., Carlisle J. C., and Campbell R. H. Context Switch Overheads on Mobile Device Platforms. Uni-

Dear reader: I honestly have to do a better cheat sheet than this one, but this fellow one helped me on countless times, and believe me it will help you immensely for our next page article. Sincerely Bruno Doiche

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Esc normal mode	~ toggle case	! external filter	@, play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) sentence	"soft" bol down	+ next line
· goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format	
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.	
q record macro	w next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	· misc	· misc		
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	· ex cmd line	" reg. spec	bol/ goto col		
a append	s subst char	d delete	f find char	g extra cmds	h ←	j ↓	k ↑	l →	· repeat t/T/f/F	· goto mk. bol	· not used!		
Z quit	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid	< un-indent	> indent	· find (rev.)				
Z extra cmds	X delete char	c change	v visual mode	b prev word	n next (find)	m set mark	· reverse t/T/f/F	· repeat cmd	/ find				

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`

WORDS: `quux(foo, bar, baz)`

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)

:e f (open file f), :%s/x/y/g (replace 'x' by 'y' filewide), :h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim), CTRL-F/-B: page up/down, CTRL-E/-Y: scroll line up/down, CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay\$ to copy rest of line to reg 'a')
- type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5l, d4j)
- duplicate operator to act on current line (dd = delete line, >> = indent line)
- ZZ to save & quit, ZQ to quit w/o saving
- zt: scroll cursor to top, zb: bottom, zz: center
- gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

COMO INSTALAR REBOL

UNA GUÍA PARA PRINCIPIANTES

por Bohdan Lechnowsky, Editor

Rebol (Relativo Expresión Basado Object Language) es un avance revolucionario en los lenguajes de programación que surge tras más de treinta años de investigación. Ofrece una gran flexibilidad y potencia, centrado en patrones de lenguaje intuitivo que promueven nuevas formas de desarrollo de software.

que desea que su script sea guardado. (vea “vi cheat sheet” para conocer algunos conceptos básicos sobre su uso).

Después de editar su script, tendrá que cambiar los permisos, a los que se pueden acceder por Rebol 3 (ej.: `chmod 755 myscript.r`)

vi Cheat Sheet

Vi fue creado en su época cuando había muy pocas teclas especiales en los teclados de ordenador. Las teclas como “Esc” (escape) y la flecha del cursor ni siquiera existían. Una de las grandes ventajas del editor vi es que está disponible en casi cualquier sistema operativo basado en Unix, incluyendo Android.

Debido a las limitaciones de los primeros teclados, vi fue diseñado para tener dos modos. El modo entrada y el modo comando. En la versión de vi para Android, estos dos modos se pueden alternar presionando CTRL y [(más conocida simplemente como CTRL + I) En el modo de entrada, cualquier tecla que escribe se introduce con su valor nominal. En el modo comando te presentamos algunos de los comandos más utilizados :

i Insertar en la posición actual
I Insertar al principio de la línea
a Añadir a la posición actual
A Añadir al final de la línea
dd Eliminar la línea actual
yy Copiar línea actual
. Repetir el último comando
cw Cambiar palabra actual
dw Eliminar palabra actual



Siga estos pasos para instalar Rebol 3 open source con soporte GUI:

Abra el navegador Web y escriba la siguiente URL

<http://development.saphirion.com/experimental/builds/android/>

Descargue `r3-droid.apk (2MB)`.

Cuando haya terminado, haga doble clic sobre el icono de descarga y la instalación se iniciará

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

Notas adicionales

Cuando usamos la app “Terminal” de Android, es necesario tener permisos de root. Para iniciar sesión como root, escriba el comando “su” en el terminal. Esto debe realizarse cada vez que se abra el terminal.

En mi instalación de Android, normalmente aparece un mensaje cuando intento crear un nuevo archivo, “sistema de archivos de solo lectura”. Para solucionar esto introduzca el comando “mount -o remount/”. Esto debe realizarse cada vez que se abra el terminal.

Para editar un script utilizando Terminal, introduzca “vi myscript.r” donde “myscript.r” es el nombre con el

p Pegar datos

: Entrar en modo de comandos extendido

Algunos comandos extendidos:

w Escribir (guardar) el archivo

q Salir (sin guardar) el archivo

wq Escribir y salir

q! Salir, incluso si el archivo ha cambiado desde la última modificación

/ Búsqueda hacia adelante en el texto

? Búsqueda hacia atrás en el texto

Si desea profundizar en vi cheat sheets pueden buscar manuales en línea, pero los comandos ya detallados debería permitirte editar casi cualquier cosa que necesites en vi. Aunque al principio parezca incómodo su uso, los usuarios con experiencia aseguran que te permite editar como cualquier otro editor gráfico.

En Ubuntu/ARM

Como Rebol 3 open-source está todavía en desarrollo, el puerto gráfico de Rebol para ARM aún no ha sido liberado. Según los desarrolladores de Saphirion, el puerto ARM de Rebol 3 con gráficos estará disponible en febrero o marzo de 2014. Visite <http://development.saphirion.com/experimental/builds> para conocer las novedades. La versión gráfica para Linux

PROGRAMANDO CON REBOL

FACILITANDO EL DESARROLLO

por Nick Antonaccio y Bohdan Lechnowsky, Editores

¿Por qué Rebol?

Las modernas herramientas de desarrollo de software son demasiado complejas. La creación de un pequeño programa que permita a los usuarios completar la tarea más básica en informática requiere de la instalación de un complejo entorno de escritorio integrado (IDE), Kit de desarrollo de software (SDK) y otras herramientas de apoyo. Los desarrolladores necesitan herramientas de diversas ramas del conocimiento, una Interfaz de Programación de Aplicaciones (API), sistemas de bases de datos, entre otros instrumentos.

Para los desarrolladores, la creación de apps móviles, programas de escritorio GUI y aplicaciones Web requiere del dominio de patrones de dinámica de trabajo, formatos de datos y metodologías de desarrollo inconsistente que terminan agotando día a día. Para ellos, es una parte inevitable de su trabajo.

Vivimos en una época en la que dispositivos de bajo coste y muy comunes entre la población (entre ellos ODROID) nos conecta a un enorme mundo de recursos. Estos dispositivos permiten la gestión de datos, comunicaciones, transacciones comerciales, entretenimiento, etc. que afectan, profundamente a la naturaleza y la calidad de nuestra vida cotidiana.

El hecho de no disponer de aparatos electrónicos ha sido sustituida por las nuevas generaciones de tecnologías. Los usuarios no necesitan saber cómo funciona internet, cámaras, micrófonos, altavoces, GPS y otros sensores integrados en tabletas, teléfonos, netbook y

PC de escritorio que trabajan de forma transparente y sin tener que instalar controladores adicionales.

Aparatos diminutos, como ODROID y las conexiones de red móvil de bajo coste hacen posible nuevos tipos de aplicaciones que no eran prácticas incluso, en ordenadores portátiles hace unos cuantos años. Los procesadores ODROID de cuatro núcleos de alta velocidad, sin olvidar teléfonos y tabletas modernas superan a los mejores PC de escritorio de hace una década. Datos de todo tipo y de cualquier disciplina están disponibles en línea para cualquier persona en todo el mundo, al instante. El hardware que integran es minúsculo, interfaz muy intuitivas, formatos estándar (imágenes, audio, vídeo, tablas estructuradas de texto y datos numéricos, etc), con sistemas operativos conocidos y software uniforme y familiar, de forma que todo se ejecuta tal y como esperan los usuarios.

Capaz de ponernos las posibilidades que ofrece la informática de red al alcance de nuestros bolsillos. Acceso a un enorme potencial cultural que nunca antes había existido, y nos brinda la capacidad de crear aplicaciones personalizadas que aprovechan ese tremendo potencial que seguirá creciendo en impotancia y en generaciones futuras.

Esta situación es fantástica para los usuarios de "apps", pero para los desarrolladores de software el panorama actual de herramientas es un completo desorden. Los residuos que han dejado las soluciones de software y hardware a lo largo del tiempo tienden a infectar el proceso de desarrollo de código moderno.

x86 ya está disponible, pero no funciona en ODROID. Las adaptaciones de Linux están siendo impulsadas en gran medida por un acuerdo de Saphirion y con otros profesionales, por lo que esperamos mucho movimiento en los próximos meses.

Mientras tanto, Rebol 3 está disponible en su variedad no-GUI en <http://rebolsource.net>, con el que puedes hacer maravillas, como sustituto de bash, perl, PHP, Python y otros lenguajes que no tienen GUI nativo

Siga estos pasos para instalar Rebol 3 en Ubuntu / ARM :

Abra un navegador y introduce la URL <http://rebolsource.net>.

Localiza la descarga para Linux ARM. Hay disponible dos variantes: hard float y soft float. Cualquiera de los dos debería funcionar correctamente, pero si tiene problemas, la soft float es probable que presente mayor compatibilidad con la mayoría sistemas.

Después de la descarga, abre Terminal y cambiar el nombre del archivo descargado a "r3" para la facilidad su uso (ej: `sudo mv r3-linux-arm-g4d9840f r3`)

También, cambiar los permisos para permitir la ejecución de la archivo (ej: `sudo chmod 755 ./r3`)

Para ejecutarlo, escriba `sudo ./r3`.

Para ejecutar un script ubicado en un archivo escriba `sudo ./r3 myscript.r`.

Podemos, incluso ejecutar scripts que se encuentran en Internet mediante

`: sudo ./r3 http://mysite.com/myscript.r`.



Numerosos intentos por estandarizar los diferentes formatos y plataformas con fines comerciales han dado lugar a la necesidad de mantener una enorme variedad de estructuras de datos, lenguajes y conjuntos de herramientas

Para desarrollar una sencilla app para android, es necesario instalar diverso software en un ordenador de escritorio, incluyendo un Entorno de desarrollo Integrado (IDE), un Entorno de desarrollo de software (SDK), un dispositivo emulador, y herramientas utiles que te permitan aliviar ese complejo proceso de desarrollo de una aplicación.

Incluso para desarrolladores expertos, el actual desarrollo de aplicaciones resulta una tarea pesada. Si también tienes la intención de trasladar tus aplicaciones a móviles, ordenadores y plataformas web, necesitas poseer cierta experiencia, habilidad y un “montón” de tiempo.

No es raro ver como equipos de profesionales se dedican al desarrollo y soporte de una única aplicación. A pesar de que Android es una plataforma abierta, los dispositivos actuales son muy cerrados o al menos engorrosos para la mayoría de los desarrolladores. Las barreras para aprender a programar son demasiadas, incluso para aquellos que se dedican a tiempo completo. Atrás han quedado los días en los que aficionados escribían aplicaciones para sus dispositivos personales y profesionales. Hubo un tiempo en el que esta actividad fue tan popular, accesible e incluso divertida para los usuarios que desarrollaban aplicaciones comerciales. Es una lastima, porque nunca antes ha habido tanto conocimiento y tantos recursos informáticos a nuestro alcance .

Rebol en todas las plataformas

Durante más de una década, un pequeño grupo de desarrolladores que programan en Rebol han conocido lo que significa ser verdaderamente productivo, usando una simple y unica herramienta para crear aplicaciones de todo tipo y para todas las plataforma: mas de

40 plataformas de hardware, sistemas operativos y entornos web.

Todos los componentes principales que hacen posible la informática del presente: Interfaz grafica de usuario, redes de comunicación, gestión de formatos de datos, etc. son manejados en Rebol usando una Sintaxis simple y poniendolos en practica usando un pequeño intérprete que se ejecuta de forma similar en todos los sistemas operativos.

Rebol ha sido diseñado partiendo de cero para eliminar la complejidad causada por distintos parches y herramientas, y podemos decir que realmente es excepcional en este sentido. Rebol evita la mayoría del desorden que ha hido creciendo a lo largo de los años dentro de la tecnología de desarrollo de software, y para una gran variedad de tareas de desarrollo simplemente funciona bien, rápido y más simple de lo que puede imaginar cualquier persona envuelta en ese desorden de herramientas tradicionales de programación.

Rebol tiene una habilidad especial para incorporar nuevas capacidades dentro de las estructuras de lenguaje simples. (también conocidos como “dialectos”, lenguajes de dominio específicos, o “DSL”) que controlan múltiples capas de la operatividad. Va mucho más allá de las capacidades que ofrecia los antiguos y rigidos métodos de Programación orientada a objetos (POO) y que reduce ampliamente la complejidad en diversas tareas de desarrollo y programación.

Recientemente, la versión 3 de Rebol (“Rebol 3”o “R3”) fue lanzado por Carl Sassenrath como un proyecto de código abierto, con una versión para Android denominada “Saphir“ desarrollada y mantenida por Saphirion group. Saphirion es una empresa de desarrollo de aplicaciones comerciales que utiliza Rebol como su principal herramienta. Las versiones R3 Saphir gratis y de código abierto también existen para Windows, Mac y Linux, y las versiones de servidor de R3 hacen posible fáciles estrategias de desarrollo web.

En la mayoría de los casos, la descarga Saphir R3 tiene un tamaño de 0,5 a 1,5 mb con GUI, sistema de redes, entre otros componentes necesarios incluidos y es relativamente fácil de aprender y utilizar. El conjunto de herramientas de desarrollo de R3 requiere menos de un minuto para su instalación, ya que no es necesario instalar ningún SDK o IDE robusto.

Rebol trabaja de forma similar en todas las plataformas, sin cambios en el código o flujo de trabajo. Incluso las personas que pasan la mayor parte de su tiempo haciendo otras cosas, puede aprender a crear potentes aplicaciones personales y profesionales a medida, de forma rápida y sencilla.

A pesar de su naturaleza simple, R3 no es un juguete o entorno para aficionados. Rebol es una poderosa herramienta que ha demostrado su valía en numerosos proyectos profesionales, en una amplia variedad de contextos de demanda comercial en todo el mundo y en mas de una década.

Ejemplos de aplicaciones Rebol

Aplicaciones de calendarios

Programas de correo electrónico

Aplicaciones profesionales para manejar el inventario , ventas y gestión de empleados

Programas informáticos específicos destinados reventa internacional

Aplicaciones para organizar actividades de grupo en su su escuela local

Aplicaciones web para la gestión de suscripciones a un club online

Interfaces de hardware para el control de los dispositivos de su hogar o negocio.

Sistemas para gestionar la robótica en instalaciones de cadenas de montaje

Sistemas para gestionar la robótica en instalaciones de marcas comerciales

Evolución de Rebol

Durante la mayor parte de su vida, Rebol 2 fue una herramienta comercial de código cerrado que prospero como

una “arma secreta” entre una pequeña comunidad de usuarios que se comunicaban de forma privada y secreta usando canales de comunicación por invitación desarrollados con Rebol 2.

Con el versión R3 del código abierto, el nuevo potencial para el desarrollo en Android y con un legado de más de 40 plataformas de desarrollo web, Rebol ha atraído a un nuevo grupo de programadores interesados en sus capacidades de producción para la creación de todo tipo de software. En futuros artículos te mostraremos lo fácil que es empezar.

Red en todas las Plataformas

Durante la época del código cerrado de Rebol 2 y 3, varios lenguajes fueron desarrollados como variantes de Rebol como Boron, Topaz, World y Red. El formato de intercambio de datos, JavaScript Object Notation (JSON) fue inspirado en el formato de datos de Rebol.

En la actualidad, El mejor resultado de todos estos esfuerzos es Red, que también cuenta con un subsistema a bajo nivel denominado Red/System. Mientras que Rebol es sólo un lenguaje interpretado, Red y Red/System pueden funcionar tanto como un lenguaje interpretado (ejecuta directamente las instrucciones) como un lenguaje compilado. El fundador y principal desarrollador de Red es Nenad Rakocevic, también conocido como “DocKimbel”.

El lenguaje Rebol ha sido recientemente reconocido como el lenguaje multifunción más expresivo, como se ve en este gráfico <http://bit.ly/1iikG1r>. Es una característica que se logra codificando cierta cantidad de código.

Los únicos lenguajes consideramos como más expresivos que Rebol son Augeas y Puppet, ambos no son de uso general. Red pertenece a la misma categoría que Rebol en lo que se refiere a expresividad.

Red junto a Red/System, es el primer competidor real para conver-

tirse en una solución de programación “full-stack”. Varios lenguajes comunes con comparados y contrastados en <http://bit.ly/1tHdcbS>, por su “Ambito natural de aplicación”. Red y Red/System cubren la casi totalidad de niveles de aplicación.

¿Otro language?

Antes de comenzar con esta sección, asegúrese de haber completado la instalación de Rebol 3, como se describe en el artículo de Bohdan Lechnowsky. Las capturas de pantallas mostradas en éste y futuros artículos son tomadas de un ordenador con Windows. Para Ubuntu, android o cualquier otro sistema operativo que soporte Rebol 3, las capturas son prácticamente idénticas.

Durante años, los desarrolladores han estado escuchando que una “nueva” tecnología de lenguaje, supuestamente recortaría en diez veces su tiempo de desarrollo. Esa forma de hablar era tan habitual que ya no significaba nada. En esta sección introductoria proporcionaremos algunos ejemplos que confirman algunas de las capacidades productivas de las que disfrutaban los expertos en Rebol. Realmente fáciles y simples.

Usando R3, puedes desarrollar una aplicación GUI “hello world” para Android, Linux, Windows y Mac de forma muy sencilla:

```
load-gui
view [text "Hello World!"]
```

En Android, no existe el botón “cerrar” como en Windows, por lo que presiona la tecla “Escape” del teclado para volver a la Línea de comandos Rebol.

El ejemplo de arriba es mucho más que una app que muestra el típico texto en consola “hello World”. En realidad es un completo formato de ventana, capaz de mostrar toda clase de “widgets” útiles de interfaz gráfica de usuario (GUI).

Aquí hay otro sencillo programa que muestra algunas entradas de texto, botones, listas y otros “widgets” GUI

comunes. Tenga en cuenta que cada palabra única representa directamente algo que aparece en pantalla. No existe sintaxis oculta, realmente no necesitas entender de programación para comprender que es lo que hace este código.

```
load-gui
view [
  field
  area
  check
  radio
  text-list
  text-table
  drop-down
  button
]
```

En Android, la línea de comandos Rebol sólo admite comandos de una sola línea. Esto no es un problema, ya que Rebol es de sintaxis libre. Introduce el siguiente comando en Android:

```
load-gui
view [field area check radio
text-list text-table drop-down button]
```

Alternativamente, los programas multi-línea pueden ser escritos con un editor de texto y ejecutados mediante Rebol. Por ejemplo, si creas un archivo llamado `gui-demo.r`, puede ejecutarlo desde la línea de comando Rebol 3.

```
do %gui-demo.r
```

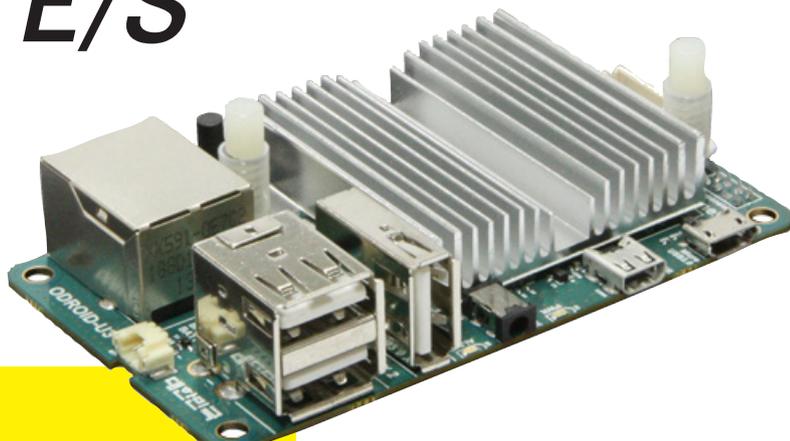
Eso es todo!

Tengaa en cuenta que Rebol tiene numerosas variantes GUI. Los ejemplos mostrados en este y futuros artículos usan la variante “R3-GUI” que es muy similar a Rebol 2 Visual Interface Dialect (VID). En el próximo artículo, veremos como crear un editor de texto totalmente funcional, calculadora gráfica y otras aplicaciones.

PROTECCION E/S ACCESO

USANDO LENGUAJE C/C++
PARA ODROID-U3

Justin Lee y Kevin Kim



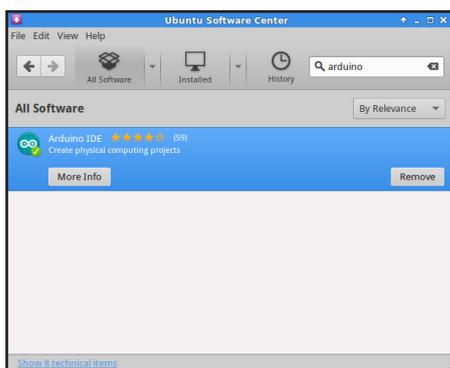
Uno de los mejores accesorios disponible para la ODROID-U3 es el protector E/S, que esta diseñado como un flexible modulo de control y toma de datos que proporciona una conexión directa al Conector E/S de ODROID-U3.

La protección E/S incluye una amplia gama de conexiones con las que podrás conectar chips DIP, sensores, entre otros. A lo largo de los bordes, todos los GPIO / ADC / PWM y conectores de energía estan divididos 0.1 pins para facilitar el acceso. También hay dos cabezales de 3 pines para las pequeñas conexiones de “servomotores” y un conector de 10 pines de I2C/GPIO para una mayor expansión.

Este artículo te mostrará como usar el protector E/S con ejemplos en C++.

Instalación

Los drivers vienen instalador en Ubuntu 13.10, que está disponible en los foros ODROID. Comprueba que tienes los siguientes módulos (T's TCA6416A I2C to Parallel Port Expander) en tu kernel: `/lib/modules/`uname-r`/kernel/drivers/gpio/gpio-pca953x.ko`



Uso

Descargue el software Firmado para el protector de E/S, y realice la configuración de la siguiente forma:

Configurar Puerto :
Tools > Serial Port > /dev/tty-ACM99

Abra un ejemplo de código y cargelo en tu IO_Shield

File > Examples > Firmata > StandardFirmata

Descargue el código fuente del protector E/S: http://dn.odroid.com/U3_Accessory/u3_IOshield_example.tgz

1. Extraiga el código fuente usando el comando "tar -xvzf".
2. Compile la biblioteca libfirmata.a
3. Compile el código de ejemplo con enlace a libfirmata.a:

ledBlink.ex es un ejemplo de protección E/S 'D13' en puerto LED

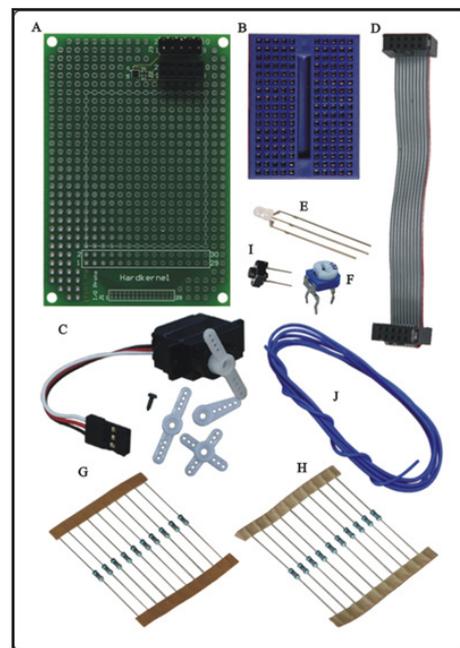
servotest.ex es un ejemplo de protección E/S 'D3' en puerto servo motor

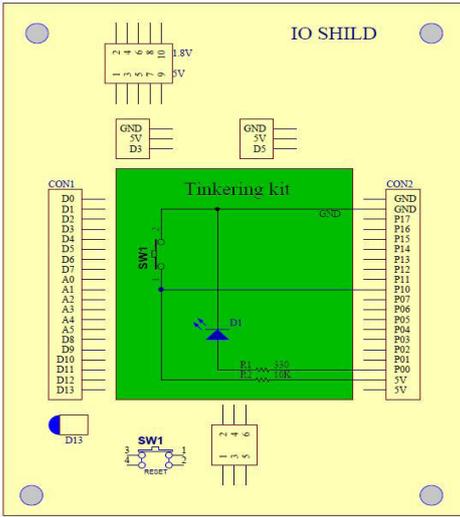
Código de Ejemplo

Ejecuta el centro de Software Ubuntu, Localiza "arduino" e instalalo:

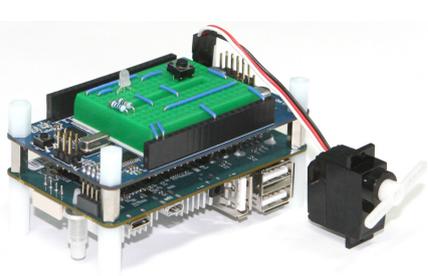
Prueba con el protector E/S del U3 y el kit de ajustes

Utilice las siguientes partes del kit, también disponible en la pagina web de Hardkernel: B, C, E, G, H, I, J:





Coincidir con este circuito, y colocar U3 como se muestra a continuación:



Acceder GPIO con el entorno de usuario de Linux

GPIO significa “General Purpose Input/Output” y es un pin especial presente en algunos chips que se pueden configurar como entrada o salida. Es utilizado para mover una señal alta o baja (en modo salida) o para obtener el estado actual de la señal (en modo de entrada). Generalmente, estos pin son gestionados directamente por los módulos del núcleo, pero igualmente hay otra forma simple de gestionar estos pins dese el entorno de usuario.

Los Kernels de Linux estándar tienen una interfaz especial que permite acceder a los pins GPIO. Al ejecutar el comando menuconfig del núcleo antes de compilar el kernel, puedes comprobar si esta interfaz está activada o no y habilitarla. La ruta es:

```
Device Drivers > GPIO Support > /sys/class/gpio/... (sysfs interface)
```

Si activas el driver puede acceder a GPIO de este modo :

```
# /sys/class/gpio
```

Si quieres trabajar con un GPIO específico, primero debe reservarlo, establecer la dirección de entrada/salida y podrás empezar a gestionarlo. Una vez que reserves el GPIO y termines de usarlo, también necesitas liberarlo para permitir que otros módulos o procesos puedan usarlo. Este sistema debe aplicarse si utilizas el GPIO desde el kernel o a nivel de usuario .

Exportar GPIO

Al introducir “sudo modprobe” en el Terminal después de arrancar el núcleo de Linux, el driver GPIO y dispositivos TCA6426 serán conectados al bus i2c-gpio.4, 16 puertos son identificados GPIO # 289 ~ # 304 (Pin protector E/S P00 ~P07 , P10 ~ P17)

```
# sudo modprobe gpio-pca953x
# sudo su -c 'echo tca6416 0x20 > /sys/devices/platform/i2c-gpio.4/i2c-4/new_device'
# dmesg
... i2c i2c-4: new_device: Instantiated device tca6416 at 0x20
```

A continuación, compilamos el u3_shield_GPIO_sysfs.c, localizado en la carpeta de ejemplos, usando g++ . Tenga en cuenta que la función GPIOsys_init () necesita permisos de superusuario para hacer “modprobe” y añadir el dispositivo i2c . La contraseña por defecto en el imágenes oficiales es “odroid” .

```
P00 port : LED blink
P10 port : key press, then GPIO
297 read value = 0, otherwise GPIO
297 read value = 1
```

Tu aplicación en el arranque

Hay un área en el kernel para cargar los módulos a medida que se necesite. Sin embargo, no se puede utilizar para enviar co-

mandos “echo” o valores a “sysfs” sin ser root o tener permisos de superusuario.

Afortunadamente, hay una forma de superar esta limitación mediante el uso de un archivo de configuración para lanzar el módulo como un “demonio”, usaremos un “script Shell” para iniciar el proceso:

1) Crea el archivo /etc/init/tca6416.conf, que puede requerir privilegios root:

```
description "TCA6416 Module Initialization"
start on runlevel [2345]
exec /usr/sbin/tca6416init.sh
```

2) Crea el archivo /usr/sbin/tca6416init.sh:

```
#!/bin/bash
# Load the Kernel Module
modprobe gpio-pca953x
# Set the I2C address of the module
echo tca6416 0x20 > /sys/devices/platform/i2c-gpio.4/i2c-4/new_device
# Enable all GPIO's on the board
for gpio_n in `seq 289 304`
do
echo $gpio_n > /sys/class/gpio/export
done
```

Al añadir estos dos archivos, todos los Puertos GPIO se habilitarán de forma permanente al reiniciar. También puedes añadir tu aplicación en “tca6416init.sh “ como método alternativo .

Infomación adicional

Los ejemplos y librería FirmataC son realizados por jdourlens, que han sido modificados para su uso en ODROID: <https://github.com>.

USAR UN ODROID-XU COMO UN ROUTER WIFI

LLEGA A 802.11AC CON ESTILO

por Mauro Ribeiro, Desarrollador Hardkernel

Si deseas usar tu ODROID XU como router inalámbrico, necesitarás varios elementos, todos ellos disponibles en la web de Hardkernel

ODROID -XU (+E o Lite)

Adaptador USB 3.0 a LAN Gigabit

Cable MicroUSB 3.0 a USB OTG

Adaptador USB 3.0 a 802.11ac

MicroSD o eMMC con Ubuntu Server instalado

El ODROID-XU es el que mejor funciona en este proyecto, ya que cuenta con un puerto USB 3.0 por lo que podemos utilizar todo el ancho de banda disponible para Wifi y porque los modelos XU permiten, además el uso del adaptador USB 3.0 LAN Gigabit. Para un mejor rendimiento, usaremos nuestra LAN 10/100 integrada como puerto WAN.

El adaptador Wifi USB 3.0 usado es un Netis WF2190, que incluye el chip Realtek 8812AU. Si tienes problemas para encontrar el adaptador Netis WF2190, hay una lista de adaptadores equivalentes que utilizan el mismo chip en: <http://goo.gl/dNwdnY>.

Antes de empezar, asegúrese de que tener al menos 4 GB de espacio libre en disco. Los siguientes valores son los utilizados en este tutorial (su red local puede ser diferente) :

La dirección de LAN 10.10.10.0/24

La puerta de enlace LAN (ODROID) 10.10.10.254

El rango DHCP desde 10.10.10.1 a 10.10.10.253

Creame, esta configuración es buena

La interfaz LAN es eth1 (adaptador Gigabit)

La interfaz WAN es eth0 (Conectado a Internet)

La interfaz de Wifi es wlan2

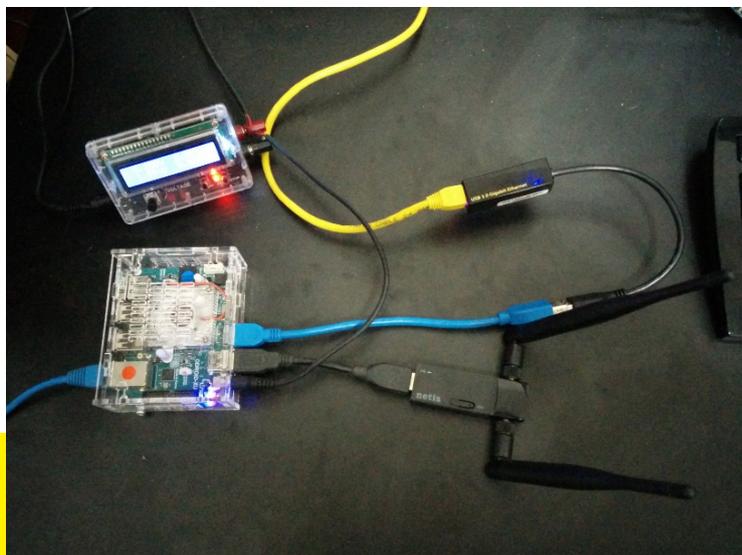
Aunque es poco probable, el valor de la interfaz Wifi (el último elemento de la lista) puede ser diferente en su LAN, pero en este tema los trataremos más adelante. También es muy importante por seguridad que cambie la contraseña de root y la del usuario "odroid", especialmente si va a conectarse a internet con tu ODROID.

Instala las dependencias y paquetes necesarios

```
apt-get install git build-essential bridge-utils
```

Compile el kernel

```
git clone --depth 0 https://github.com/hardkernel/linux.git -b odroidxu-3.4.y linux
cd linux
make odroidxu_ubuntu_defconfig
make -j5
```



```
make modules_install
cp arch/arm/boot/zImage /media/boot
sync && reboot
```

Esto compila un nuevo kernel XU desde el último código fuente de GitHub. Ahora dispones de una copia actualizada del kernel en tu placa ARM.

Gestiona los drivers Realtek Wifi.

La configuración de los drivers Realtek wifi presenta cierta dificultad.

Use los drivers facilitados

Netis envía un mini-cd que incluye los controladores para Linux. Localiza una carpeta con el siguiente nombre: RTL8812AU_8821AU_linux_v4.2.0_6952.20130315. Copia esta carpeta a tu ODROID, estos son los drivers que necesitamos para compilar.

Compila los drivers Wifi

```
cd RTL8812AU_8821AU_linux_v4.2.0_6952.20130315/driver
tar zxvf rtl8812AU_8821AU_linux_v4.2.0_6952.20130315.tar.gz
cd rtl8812AU_8821AU_linux_v4.2.0_6952.20130315
```

* edita el fichero en la linea 583 ARCH ?= cambia a ARCH ?= arm

```
make && make install
modprobe 8821au
```

Ahora los controladores Wifi deberían funcionar correctamente.

Probar el controlador Wifi

```
ifconfig wlan0 up
iwlist wlan0 scan
```

Tenga en cuenta que si wlan0 no aparece, compruebe wlan1 o wlan2 tecleando “ifconfig-a “. Guarde esta información para después, ya que necesitaremos conocer el nombre de la tarjeta wifi más adelante.

Llegados a este punto, verás en orden la lista de Redes Wifi. Ahora puede utilizar tu ODROID como cliente Wifi al conectarte a otras redes inalámbricas.

Compilar hostapd y wpa_supplicant

Debido a que Realtek 8812 todavía no tiene soporte para Linux, tenemos que compilar los controladores incluyendo hostapd desde el código del fabricante disponible en su página web.

Compilar hostapd desde el código del fabricante

```
cd RTL8812AU_8821AU_linux_v4.2.0_6952.20130315/wpa_supplicant_hostapd
tar zxvf wpa_supplicant_hostapd-0.8_rtw_r6747.20130222.tar.gz
cd wpa_supplicant_hostapd-0.8_rtw_r6747.20130222/hostapd
```

Antes de compilar e instalar hostapd, vamos a asegurarnos de que no tenemos instalado hostapd de Ubuntu:

```
apt-get remove hostapd
make && make install
```

Compilando wpa_supplicant

```
cd RTL8812AU_8821AU_linux_v4.2.0_6952.20130315
cd wpa_supplicant_hostapd
cd wpa_supplicant_hostapd-0.8_rtw_r6747.20130222
cd wpa_supplicant
```

Tenemos que asegurarnos que Wpa_supplicant de Ubuntu no este presente:

```
apt-get remove wpasupplicant
make && make install
```

Esto instala todos nuestros controladores wifi, junto con las herramientas facilitadas por el fabricante. Vamos a continuar con la configuración

Configuración

Configura el enlace de red Ubuntu para crear la integración entre las conexiones inalámbricas y por cable.

```
cd /etc/network
```

También necesitará editar el archivo de interfaz, utiliza la siguiente captura de pantalla como ejemplo:

```
-- BEGIN --
# setup loopback interface
auto lo
iface lo inet loopback
# setup eth0 as DHCP for our WAN
auto eth0
iface eth0 inet dhcp
# setup eth1 as manual and leave it empty
auto eth1
iface eth1 inet manual
# Create our bridge with eth1 on it and use the IP 10.10.10.254
auto br0
```

```
iface br0 inet static
bridge_ports eth1
address 10.10.10.254
netmask 255.255.255.0
network 10.10.10.0
pre-up /sbin/ifconfig wlan2 up
```

Tras el reinicio, la interfaz LAN se llamará br0 con eth1 siendo parte de ella.

Punto de acceso inalámbrico Configuración mediante hostapd

El adaptador inalámbrico no cuenta con un sistema dual, tienes que elegir entre la frecuencia de 5 GHz o 2,4 GHz. Si estás en una red con dispositivos de 5 Ghz en su mayoría, utiliza configuración de 5GHz que te permite velocidades de hasta 800 Mb/s. Si necesita compatibilidad adicional con los ordenadores más antiguos o con dispositivos que no cuenta con 5GHz, utiliza la configuración de 2,4 GHz.

Crea el siguiente archivo /etc/hostapd.conf:

```
5GHz version
# Change this to match your config
interface=wlan2
ctrl_interface=/var/run/hostapd
# Your network name
ssid=ODROID-NET 5Ghz
channel=36
wpa=2
# Your network password
wpa_passphrase=testtest
bridge=br0
eap_server=0
wps_state=0
driver=rtl871xdrv
```

```

beacon_int=100

hw_mode=a

ieee80211n=1

wme_enabled=1

ht_capab=[SHORT-GI-20]
[SHORT-GI-40][HT40+]

wpa_key_mgmt=WPA-PSK

wpa_pairwise=CCMP

max_num_sta=8

wpa_group_rekey=86400

2.4GHz version

interface=wlan2

ctrl_interface=/var/run/hostapd

ssid=ODROID-NET 2.4Ghz

channel=6

wpa=2

wpa_passphrase=testtest

eap_server=0

wps_state=0

driver=rtl871xdrv

beacon_int=100

hw_mode=g

ieee80211n=1

wme_enabled=1

ht_capab=[SHORT-GI-20]
[SHORT-GI-40][HT40+]

wpa_key_mgmt=WPA-PSK

wpa_pairwise=CCMP

max_num_sta=8

wpa_group_rekey=86400

```

Debe adaptar los parámetros de interfaz “ssid” y “wpa_passphrase” de su red.

Autoinicio hostapd

Edita /etc/rc.local y añade las siguientes líneas antes de la fila “exit 0”:

```

/usr/local/bin/hostapd /etc/hostapd.conf &

sleep 3

/sbin/brcctl addif br0 wlan2

```

DNS y DHCP

Nuestra red necesita tanto un servidor DNS como DHCP para configurar automáticamente nuestros clientes. Vamos a utilizar “dnsmasq” instalado como dependencia en el primer paso .

```
-mkdir /etc/dnsmasq
```

Crea el fichero /etc/dnsmasq.d/odroid.conf:

```

resolv-file=/etc/resolv.dnsmasq

addn-hosts=/etc/dnsmasq/hosts

dhcp-hostsfile=/etc/dnsmasq/dhcp

expand-hosts

min-port=4096

stop-dns-rebind

rebind-localhost-ok

interface=br0

# Here on this line we'll
configure the LAN interface
(br0), initial IP Address

# last IP address, Network
netmask and the time that a
IP will be kept to a client

# that is known as lease time

dhcp-range=tag:br0,10.10.10.1,10.10.10.253,255.255.255.0,1440m

# This line we configured the
IP address of our gateway
(the odroid)

dhcp-option=tag:br0,3,10.10.10.254

dhcp-lease-max=255

dhcp-authoritative

```

Ahora que los dispositivos pueden conectarse a nuestra nueva red, necesitan ser capaces de conectarse a Internet. Vamos a solucionar este tema:

Crear tablas IP para el acceso a Internet

Edita /etc/rc.local, y antes de “exit 0”, añade las siguientes líneas:

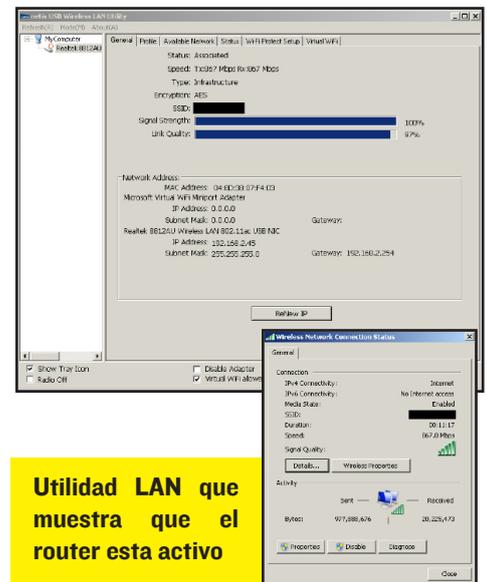
```

echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o eth0 -j MASQUERADE

```

Ahora disponemos de un router inalámbrico totalmente funcional



Utilidad LAN que muestra que el router está activo

Para hacer su router más útil, puedes conectar discos duros a los puertos USB. A continuación, crea un servidor de Torrent, un servidor de archivos samba, o cualquier otro tipo de recurso de red compartido.

He probado el router ODROID-XU con mi teléfono Nexus 5, y su rendimiento es de 433Mb/s . Bastante aceptable para una red Wifi, podría ir más rápido, pero 433Mb/s es el límite del Nexus 5 .

El mes que viene, hablaremos de como configurar graficamente el monitoreo de datos de entrada y salida en la red.

EL ARTE DEL MULTI-BOXING:

CENTRO MULTIMEDIA 1080p

USANDO POCKET ROCKET Y WHISPER

por Rob Roy, Editor Jefe

Multi-boxing es una técnica de red avanzada para conectar dos o más ordenadores para crear un unico sistema virtual. Originalmente hacia referencia a jugadores que dominaban partidas en línea mediante el control de varios jugadores simultaneamente. Esta técnica también se puede utilizar para otras aplicaciones como una forma de utilizar informática paralela para mejorar la capacidad de respuesta global de la experiencia de usuario.

Comparado con un sistema single-box que requiere de un reinicio para cambiar entre plataformas, la combinación de teclas Alt-Tab permite cambiar entre los diferentes escritorios de las máquinas. Tareas que normalmente ralentizarían la interfaz de usuario como la descarga directa, la conversión de vídeo o torrents; se trasladan a un segundo equipo que realiza perfectamente este trabajo de fondo, pero aparece en el primer equipo como si se tratara de una aplicación nativa. Resulta más económico conectar varios Odroids juntos, accediendo por un solo monitor y un teclado que comprar equipamiento por separado para cada equipo.

La principal ventaja de la delegación de tareas entre ordenadores es que, si la actividad de la CPU o del disco es alta en una máquina el rendimiento del resto de sistemas no se ven afectado, ya que cada equipo tiene sus propios recursos locales, sin necesidad de competir por el uso del procesador o acceso al disco duro. Por ejemplo, cuando visualizamos videos el proceso de descarga activo de fondo puede afectar a la velocidad de

reproducción de fotogramas, generando saltos en el video y audio. Usando un segundo sistema que realiza el trabajo de gestión de archivos multimedia, se permite liberar al sistema principal para que pueda reproducir video sin problemas.

Dos de las imágenes disponibles en los foros ODROID : Android Pocket Rocket y Lubuntu Whisper, pueden trabajar conjuntamente para ejecutar dos Odroids como un sistema virtual octa-core. Dos imágenes trabajando simultáneamente mejora la estabilidad y la eficiencia de ambos entornos, con la ventaja de duplicar la capacidad de procesamiento, RAM y espacio disponible en disco. Alternativamente, La imagen "Dream Machine" podrá utilizarse en lugar de Whisper y la imagen Couch Potato para SD se puede sustituir por Pocket Rocket sino tenemos eMMC.

Equipamiento

Activar un equipo virtual multi-boxed con ODROID requiere de dos Odroids de la serie X o XU y un router wifi, módulos eMMC disponibles de Hardkernel que ofrecen el mejor rendimiento y como medio de almacenamiento, discos externos USB o tarjeta SD de alta capacidad, que permite el intercambio de archivos entre los equipos sin agobiar la partición principal del sistema operativo

La siguiente guía describe los pasos para configurar el típico Media Center con dos Odroids, utilizando uno como el servidor de medios y otro como el cliente multimedia. Puesto

que XBMC para Android ofrece el mejor rendimiento en resoluciones de 1920x1080, he Elegido Pocket Rocket Android como cliente, que cuenta con los protocolos necesarios (SSH , VNC y Samba) instalado. Para el servidor de medios, Lubuntu Whisper funciona bastante bien debido al escritorio compartido preconfigurado, Transmisión y Samba. Ambas imágenes disponibles en <http://forum.odroid.com/>

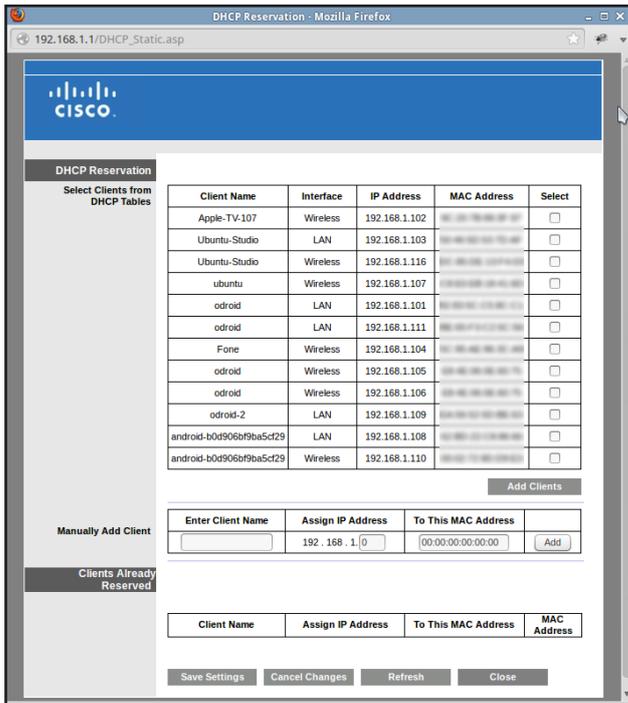
En este sistema de doble placa, la descarga y/o almacenamiento de medios se realiza con el servidor Linux, que puede manejar con fluidez un alto volumen de tráfico en red y lectura de disco duro. Cuando los archivos multimedia están listos para su visualización, el sistema Android los carga desde un directorio compartido y los reproduce usando un decodificador de video por hardware como XBMC o MXPlayer.

El primer paso, después de descargar y copiar las imágenes a una eMMC o SD, es configurar la red para que los Odroids puede localizarse entre ellos.

Configurando Router

El sistema de servidor que ejecute Whisper necesita una dirección IP estática local para que el otro ODROID puede localizarlo en la red con facilidad. Sin dirección IP estática, las conexiones tendría que ser reconfiguradas con una nueva IP después de cada reinicio.

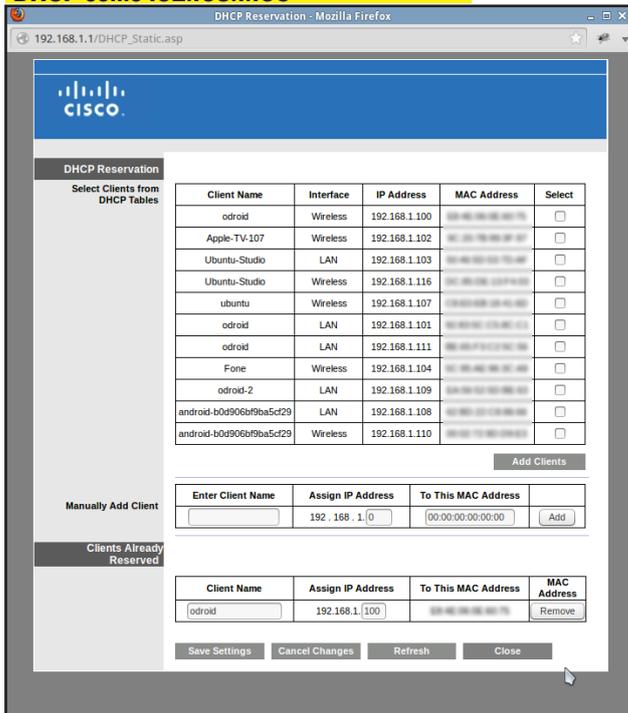
La opción para asignar una IP estatica, también denominada "DHCP reservation", se encuentra en el panel de administración del router. Por ejemplo, usando



Panel administración del router típico con opción para asignar dirección IP estática

un router cisco estándar, se accede a la página de administración en la dirección 192.168.1.1 con el nombre de usuario en blanco y contraseña "admin". Algunos routers utilizan 192.168.0.1, 10.0.0.1 u otra dirección similar. Para obtener instrucciones específicas, consulte la web del fabricante o recurra al manual de usuario para

Panel de administración del router con el cliente "ODROID" agregado a la lista de DHCP como 192.168.1.100



obtener detalles sobre el acceso al panel de administración y configuración de una IP estática.

Para reservar la dirección IP estática, arranca Whisper con su puerto ethernet conectado al router y accede al router desde cualquier ordenador de la red. Siga las instrucciones del fabricante y asigne una dirección IP estática al ODROID que esta activo. Utilice una secuencia fácil de recordar, como 192.168.1.100.

Escritorio compartido, Samba y SSH

Los dos ODROID tiene tres protocolos de comunicación entre sí. Cada uno ofrece un servicio diferente que permite tipos específicos de acceso remoto al disco duro del servidor principal, memoria y recursos perifericos :

VNC significa Virtual Network Computing, un protocolo que comunica un escritorio gráfico con un sistema remoto y procesa entradas del teclado y ratón remotos como si estuvieran conectados a nivel local.

Servidores y clientes VNC están disponibles para casi todos los sistemas operativos modernos (iOS , Android , Windows , Linux y OSX). Su gran potencial permite combinaciones únicas como el acceso a un escritorio de Windows desde Macintosh, ejecutar un escritorio Linux en Windows , o controlar de forma remota múltiples Odroids desde un teléfono con Android o un iPad .

SSH es un interprete de comandos seguro que crea una conexión cifrada

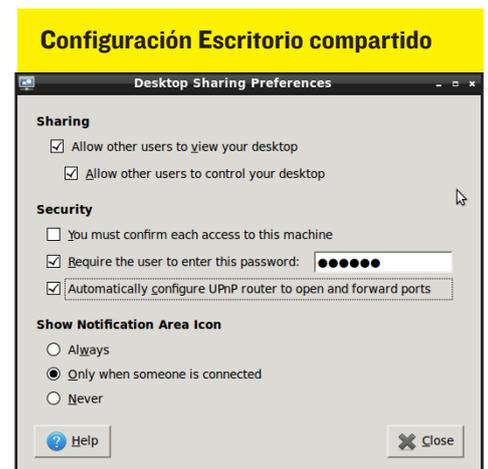
basada en texto entre dos ordenadores que utilizan un terminal en red. Con SSH, los comandos remotos se pueden enviar al servidor usando la interfaz estándar de línea de comandos para iniciar y detener aplicaciones, reiniciar el sistema, y ejecutar otras funciones administrativas. En el Media Center de ejemplo se utiliza para iniciar el servidor VNC.

Samba es una implementación de software libre del protocolo de red SMB/CIFS, permitiendo a los servidores de archivos remotos ser montados como unidades locales. Para el propósito de compartir archivos el servidor Samba permite el acceso a video, audio y otros archivos a los que se acceden por el cliente multimedia Android.

Para mejorar la seguridad del sistema, los tres protocolos requieren nombres de usuario y contraseñas por separados. Por comodidad, Samba, SSH y VNC se sincronizan al inicio del sistema y usan "odroid" como nombre de usuario y contraseña. Estas contraseñas deben ser cambiadas. En las siguientes capturas de pantalla, observarán que los parámetros seguridad no ha sido modificados y todos las contraseñas contienen el valor por defecto "ODROID".

Configurar servidor

Para la configuración inicial, conecte el ODROID con Whisper al teclado, ratón y monitor. Para activar el Escritorio Compartido seleccione "DesktopSharing" en el menú "Internet". La pantalla ha de coincidir con la que se muestra a continuación. Elija una contraseña de se-



```

odroid@odroid: ~
File Edit Tabs Help
collisions:0 txqueuelen:0
RX bytes:41046 (41.0 KB) TX bytes:41046 (41.0 KB)

lxcbr0 Link encap:Ethernet HWaddr ce:a9:6f:e3:04:63
Inet addr:10.0.0.1 Bcast:10.0.3.255 Mask:255.255.255.0
Inet6 addr: fe80::cca9:6fff:fe3:403/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 carrier:0
TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:12998 (12.9 KB)

vlan7 Link encap:Ethernet HWaddr e8:4e:06:0e:00:75
Inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
Inet6 addr: fd7c:7c9e:5497:0:ea4e:6ff:fe0e:6075/64 Scope:Global
Inet6 addr: 2601:9:6880:18c:ea4e:6ff:fe0e:6075/64 Scope:Global
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:171 errors:0 dropped:54 overruns:0 frame:0
TX packets:225 errors:0 dropped:1 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:23469 (23.4 KB) TX bytes:48189 (48.1 KB)

odroid@odroid:~$
    
```

Uso del comando ifconfig en el servidor Whisper

guridad para el acceso al escritorio que más tarde se utilizara en la configuración de bVNCFree de Android

Comprueba que la dirección IP del sistema Whisper coincide con la seleccionada en la opción DHCP de la configuración del router ejecutando Terminal desde el escritorio de Whisper y escribiendo "Ifconfig". Si las direcciones no coinciden, puede que sea necesario reiniciar el sistema Whisper con el fin de que coga la dirección IP reservada en el router.

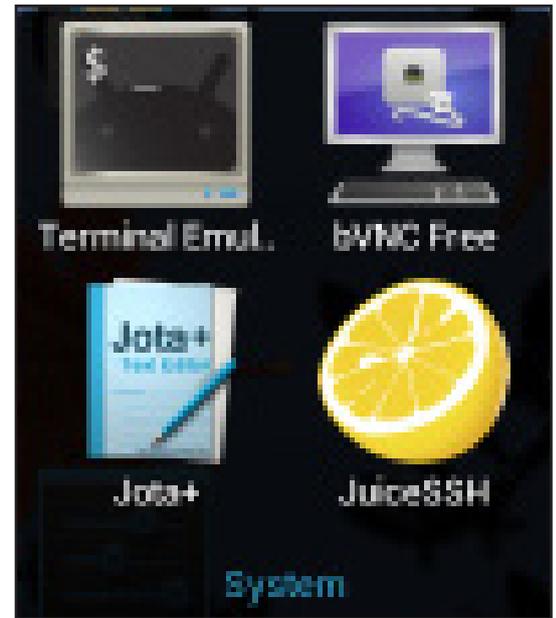
Por último, conecte un disco duro al servidor Whisper para utilizarlo como directorio compartido que contendrá los archivos y ejecute "Samba" desde el menú "Preferencias". Las opciones ha de coincidir con la que se muestra en las siguientes figuras, sustituyendo el

directorio de almacenamiento por el que se muestra en la captura de pantalla. Agregue el directorio compartido "/ home / ODROID / Downloads" ya que será el directorio por defecto de Transmisión, Firefox y Chromium. Si utilizas una unidad externa, configure las tres aplicaciones para guardar y compartir por samba el directorio de almacenamiento del disco externo.

Configurar Cliente

Ahora que el servidor está preparado, desconecte el teclado, el ratón y el monitor, y los conectamos al cliente con Pocket Rocket Android. A partir de ahora, el servidor Whisper no necesitará ningún periférico excepto en caso de caída de la red o fallo del router. El escritorio del servidor, la interfaz de línea de comandos, archivos y la CPU están disponibles desde el cliente a través de los protocolos SSH, VNC y Samba.

La conexión inicial entre los dos ordenadores se realiza a través de SSH para que el servidor VNC pueda ser iniciado, lo que permitira que el escritorio X11 sea compartido. Después de cargar el servidor Vino, JuiceSSH permanecerá en segundo plano, mientras que la aplicación para An-



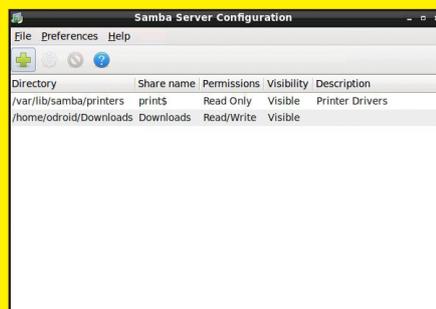
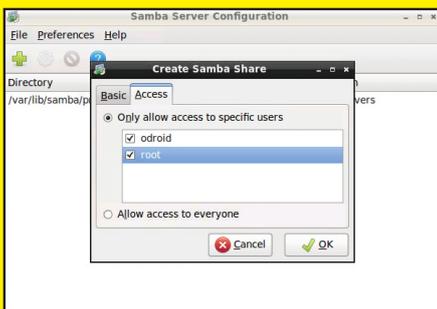
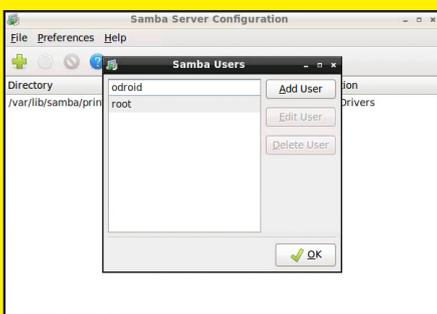
Escritorio Pocket Rocket contiene JuiceSSH y bVNCFree.

droid denominada bVNCFree dara acceso remoto al escritorio Whisper.

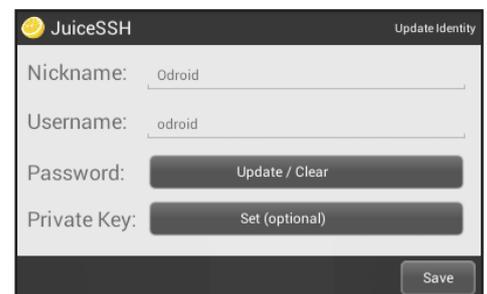
Conectando cliente al servidor

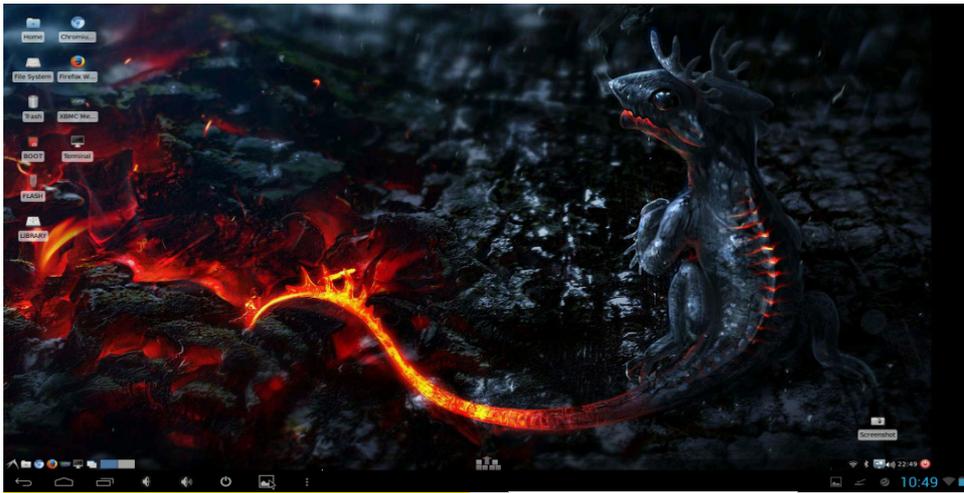
Inicie JuiceSSH y seleccione "Manage Your Connections". Las opciones han de coincidir con las que se muestra en las siguientes captura de pantalla, utilizando la dirección IP estática

Configurando samba en Whisper



Opciones de configuración JuiceSSH





Escritorio Whisper, visto desde la aplicación bVNCFree de android

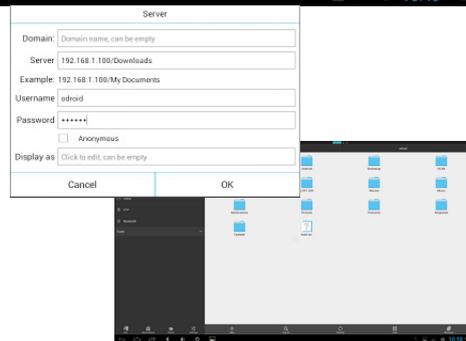
vídeo donde la pantalla parpadee o el audio no se sincronice con la imagen por los continuos picos de acceso al disco. Debido a que la eMMC o la unidad externa tienen muchas aplicaciones que requieren continuamente espacio de almacenamiento, la experiencia de reproducción se degrada ligeramente en intervalos aleatorios.

Para mejorar la estabilidad y la reproducción de vídeo, las tareas de descarga y gestión de grandes volúmenes que requieren constante acceso al disco son ejecutadas por el servidor Whisper en la configuración dual-box. Esta separación libera carga de trabajo al procesador y al disco del Cliente Android para que el reproductor de vídeo no tenga que esperar a que finalice otra aplicación antes de cargar el siguiente vídeo desde el disco duro.

Para realizar las pruebas de rendimiento del sistema dual-box, cierre todas las descargas y torrents en el sistema Android de manera que sólo este activo el reproductor de vídeo. Presione Alt-Tab para cambiar a la aplicación bVNCFree. A continuación utilice el escritorio Whisper para descargar varios archivos de gran tamaño y torrent, usando el navegador de Internet y Transmisión al mismo tiempo.

Después de que haya finalizado la descarga de algún archivo multimedia, pulse Alt-Tab para volver al reproductor de vídeo de Android, como XBMC y reproduzca el archivo desde el reciente directorio compartido por Samba.

La forma más sencilla de cargar un



Pestaña LAN ES File Explorer's

archivo multimedia, es localizarlo a través de un explorador de archivos con soporte LAN ("ES File Explorer"), o iniciando XBMC y acceder al directorio compartido a través de su interfaz.

Observamos que al mover las descargas al servidor permite que los vídeos se reproduzcan sin problemas en Android, sin el parpadeo o la congelación momentánea del vídeo. La reproducción en el cliente Android sigue siendo estable a pesar de la carga de trabajo en el servidor mejorando la experiencia global del sistema.

Mira bajo el capo

Puesto que Android y el reproductor de vídeo son los únicos que acceden al disco duro local desde el cliente, el rendimiento general mejora notablemente. Con acceso directo al disco duro los archivos multimedia son dirigidos hacia el puerto Ethernet sin espera. El resultado de la coordinación entre los dos sistemas es que, por un lado el cliente sigue respondiendo sin cortes, y por otro el servidor asume la mayor carga de trabajo en segundo plano sin bloquear los recursos del sistema Android.

La versión Dual-box del Media Center es notablemente mejor que la versión single-box, debido al uso compartido de recursos de cada sistema.

Los archivos también pueden ser copiados y compartidos de un sistema, por lo que pueden ser editados en el cliente Android mientras se mantiene una copia original en el servidor Whisper. Además, bVNCFree permite que las cadenas de texto puedan ser transferidos entre entornos de escritorio distintos (Ej., entre Windows y Linux) usando la función de copiar y pegar del Escritorio Android, en lugar de recurrir a un archivo intermedio o correo electrónico.

Desarrollando Cluster de Odroids

Siga los pasos de nuevo para conectar más equipos en el clúster, con una sesión SSH y VNC independiente para controlar cada sistema adicional. Para mi sistema multi-boxing personal, conecté varios ordenadores ODROID que proporciona un servidor web dedicado, paquetes de desarrollo de aplicaciones y una suite para chequeos de seguridad, todos ellos gestionan sus propios recursos coordinados a través del escritorio Pocket Rocket de Android

Multi-boxing Avanzado

Los usuarios expertos pueden añadir automatización al Media Center mediante cron jobs, añadiendo RSS y lanzar descargas con la interfaz de línea de comando rtorrent. Además de otros ajustes como activar la interfaz web en XBMC y transmission en remoto, y la instalación de un sitio web local para gestionar y presentar el contenido de la biblioteca multimedia en un formato atractivo y de fácil lectura. Handbrake también ha sido exportado a ODROID, que puede ser incluido en el proceso de descarga para convertir archivos multimedia a la resolución deseada y ser codificados antes de ejecutados por el cliente Android.

CONOCIENDO A UN ODROIDIAN

MAURO RIBEIRO: EL GENIO QUE HAY DETRAS DEL KERNEL LINUX DE ODROID

por Robert Hall

¿Cuál es su cargo oficial y papel en Hardkernel?

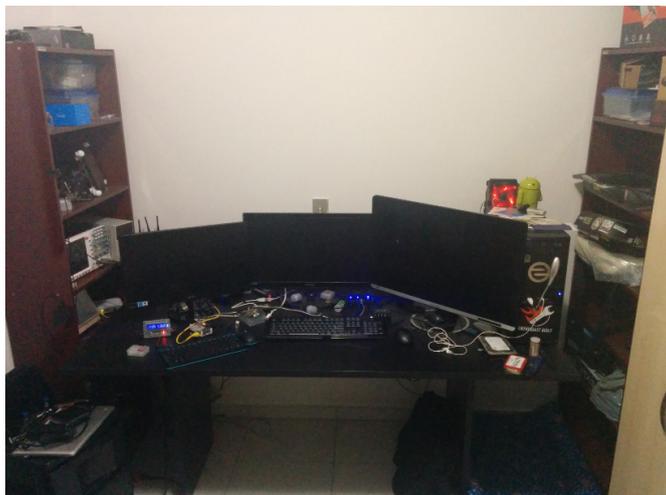
Ingeniero de Software Senior, principalmente trabajo con el núcleo, Soporte Linux y participo en los foros. Intento también llegar a Mundo del hardware. Es muy divertido :)

Como uno de los moderadores de los foros ODROID, ¿Qué es lo que mas te gusta de la Comunidad ODROID?

Creo que lo que más me gusta es la cantidad de información que hemos logrado compartir y como usuarios con apenas conocimientos de Linux, se han convertido en usuarios avanzados

¿Cómo eras tus inicios on los ordenadores?

Desde que tenía 2 años (hasta donde puedo recordar), he mostrado una cierta curiosidad por los aparatos eléctricos y electrónicos, debido sobre todo a la influencia de mi tío, quien me enseñó la electrónica básica desde que era niño. Yo tenía 5 o 6 años cuando mi tío me mostró un Apple 2+. Esto fue en 1989 o 1990.



ODROID MAGAZINE 42

Un año más tarde, mi tío me regalo un impresionante ordeanador 386. Todavía recuerdo las especificaciones exactas: “386-DX40, 4 MB de RAM, tarjeta VGA, Monitor 14” (CRT), Doble disquetera 5 1/2 y 3 1/4, y un destacado HDD Seagate de 260MB”

Conservo esta máquina. El CRT y el HDD se rompieron, pero todo lo demás funciona sin problema.

Realmente llegué a Linux en 1998. Un amigo me trajo un CD-ROM de EE.UU. que contenía “Slackware 97”. Ahí es donde empezó todo.

¿Cuáles son tus proyectos favoritos en los que has visto utilizar ODROID?

Me encantan los proyectos que tienen que ver con el control de la iluminación y el control de iluminación ambiental. No creo haber visto alguno que haya utilizado Odroid hasta el momento. He visto algunos bonitos robots impulsados por ODROID, y tengo en mente un proyecto personal que permita controlar la iluminación con Odroid

Tienes algún proyecto personal en el que está trabajando y utiliza ODROIDs?

Sí, el control de la iluminación ambiental. Se trata de utilizar una



Mauro vestido con un traje típico coreano. Recuerdo de su último viaje.

placa U3 para controlar la luz de mi habitación con 5 leds de 10W. Ahora mismo lo tengo completado al 70%. Falta algunos agustes menores relacionados con el cableado y el software de control remoto.

¿Qué otros intereses y aficiones tienes?

Los coches de carrera es probablemente mi otra gran afición aparte de la informática. He formado parte del mundo de las ECU. Los motores de combustión son muy divertidos.

Por otro lado mis perros, pertenece a la raza Lhasa-Apso. Meg es la más vieja, ahora tiene 6 años. Junior es el último nacimiento de su madre, ahora tiene 4 años. Laika y Res fueron mis primeros perros. Ellos fueron la pareja que dio a luz a todos mis perros actuales. Desafortunadamente, Laika murió a principios del pasado año debido a complicaciones derivadas de una cirugía para eliminar piedras de la vejiga. Res, lamentablemente falleció un par de meses mas tarde. Laika fallecio a la edad de 7 años y Res a la edad de 9.