

# ODROID

## Magazine

Año Uno  
Num #7  
Jul 2014

## INGENIERIA METEOROLOGICA

EL NUEVO DISPOSITIVO DE METEOROLOGÍA DE  
HARDKERNEL TE PERMITE MONITORIZAR LAS  
CONDICIONES CLIMÁTICAS CON TU ODROID



## CONFIGURA LA INTERFAZ DE TU VEHICULO TODO TERRENO AUTOMATICO

- INFORMATICA PORTATIL SOLAR CON UN ODROID U3
- FABRICA UN INTERRUPTOR DE DOBLE ARRANQUE PARA X/X2
- DESARROLLA APLICACIONES ANDROID NATIVAS CON RED

## SO DESTACADO GAMESTATION TURBO



## EMULADORES DE CONSOLA Y REPRODUCCION MULTIMEDIA PARA TU ODROID

# Qué defendemos..

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

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

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

Simple, moderno y único.  
De modo que tienes a tu alcance lo mejor



## HARDKERNEL



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

Dirección: Max-Pollin-Straße 1  
85104 Pförring Alemania

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

Nuestros productos ODROID se pueden encontrar en:  
[http://www.pollin.de/shop/suchergebnis.html?S\\_TEXT=odroid&log=internal](http://www.pollin.de/shop/suchergebnis.html?S_TEXT=odroid&log=internal)





**N**uestro columnista de Robótica Chris, acaba de lanzar la nueva entrega sobre su Vehículo Todoterreno Automático, y nos muestra cómo conectar los motores y sensores. ¡Estamos casi listos para empezar a desarrollar nuestra propia versión! Cooper y Anthony nos muestran cómo mantienen su clúster de alto rendimiento de ODROIDS, con su artículo más reciente sobre la gestión de archivos y usuarios. También echamos un vistazo al interior GameStation Turbo, la popular imagen de juegos y emuladores de consola para la serie U y X de ODROID, y aprenderemos cómo encender un ODROID usando únicamente energía solar.

Hablando de la energía solar, la placa meteorológica ODROID es el nuevo periférico para ODROID-SHOW de Hardkernel, que incluye 6 sensores ambientales por un precio muy asequible. Informes meteorológicos sobre radiación UV, presión barométrica, altitud, humedad relativa, iluminación y temperatura. Es totalmente compatible con el sistema de programación de Arduino. Puedes obtener tu placa meteorológica en la tienda Hardkernel en <http://bit.ly/lwtPdgp>.

Hardkernel estará presente en ARM Techcon del 1 al 3 octubre de 2014 en Santa Clara, California, EE.UU. Si vives en Silicon Valley o en el Area de la bahia, regístrate antes del 8 de agosto y recibirán un entrada gratuita a la Expo (59\$ en la puerta) para que vengas a pasar un rato en el stand Hardkernel y tengas la oportunidad de conocer a Justin, Lisa, Robroy o Mauro. Puedes encontrar más información sobre ARM Techcon en <http://www.armtechcon.com> y visita <http://ubm.io/lqPo8Ci> para conseguir tu entrada gratuita. ¡Nos encantaría contar con tu presencia!

Recientemente se ha unido al equipo ODROID Magazine nuestro nuevo Editor de Arte adjunto Nicole Scott, que vive en el área de la Bahía de San Francisco y tiene experiencia en video, audio, música, web y producción multimedia. ¡Bienvenida, Nicole!

¿Te gustaría escribir para ODROID Magazine? Actualmente estamos buscando varios columnistas que escriban sobre diversos temas tales como conceptos básicos de Linux, seguridad informática Kali, noticias del mundo ARM, programación Bash y cualquier otro tema que desees compartir con nosotros. Los artículos publicados son recompensados con regalos, y los columnistas regulares pueden ganar hardware de alta gama. Envía tu artículo en Google Docs o formato Office a [odroidmagazine \(at\) gmail.com](mailto:odroidmagazine@gmail.com).

El próximo número traerá consigo publicaciones muy interesantes, como la nueva generación de la familia ODROID-XU, así como un económico monitor táctil LCD USB de 9", ambos están disponibles en la tienda Hardkernel.

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



**HARDKERNEL**

# ODROID

Magazine



**Robert Hall,**  
**Editor jefe**

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



**Bo Lechnowsky,**  
**Editor**

Soy el presidente de Respectech, Inc., Consultoría tecnológica en Ukiah, CA, EE.UU. que fundé en 2001. Con mi experiencia en electrónica y programación dirijo a un equipo de expertos, además de desarrollar soluciones personalizadas a empresas, desde pequeños negocios a compañías internacionales. Los ODROIDS son una de las herramientas de las que dispongo para hacer frente a estos proyectos. Mis lenguajes favoritos son Rebol y Red, ambos se ejecutan en los sistemas ARM como el ODROID-U3. En cuanto a aficiones, si necesitas alguna, yo estaría encantado de ofrecerte alguna de las mías ya que tengo demasiadas. Eso ayudaría a que tuviese más tiempo para estar con mi maravillosa esposa de 23 años y mis cuatro hijos estupendos.



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

Está contando los días para sus vacaciones ¡bien merecidas! Mientras eso no ocurra, él estará trabajando duro para mantener las cosas en su sitio en el trabajo y jugando. Y tan chiflado como siempre.



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

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



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

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



# INFORMATICA DE ALTO RENDIMIENTO (HPC) EN CASA

## GESTION DE ARCHIVOS Y USUARIOS EN UN CLUSTER

por Cooper Filby y Anthony Skjellum -  
Runtime Computing Solutions LLC  
<http://www.runtimecomputing.com>

**E**n artículos de ODROID Magazine de los meses de febrero y marzo de 2014, describimos de forma general cómo configurar un clúster ODROID eficiente y rentable administrado por un nodo central. En este artículo continuaremos trabajando en la solidez de nuestro clúster al incorporar servicios como LDAP y NFS/AutoFS en nuestro nodo principal y nodos secundarios con el fin de ampliar sus posibles usos y usuarios. Con LDAP configurado, los usuarios podrán iniciar sesión en cualquier nodo del clúster con las mismas credenciales y permisos, y combinado con NFS/AutoFS nos permitirá configurar los directorios compartidos montados en red, creando así un usuario uniforme para conectarnos a cualquier nodo de nuestro clúster.

Para realizar todo esto, necesitaremos:

**1. 2x ODROIDS - nosotros hemos usado el modelo XU + E con Ubuntu Server 13.09. Se pueden añadir más ODROIDS con facilidad para crear un clúster más grande.**

**2. 1x Switch Ethernet (preferiblemente Gigabit, también llamado 1000-BaseT)**

**3. 3x cables Ethernet (y un cable más para cada ODROID adicional)**

**4. 1x Adaptador USB Ethernet Gigabit (ideal uno por cada ODROID)**



Un buen clúster, ¿eh? Bueno, después de leer este artículo, sabrás como desarrollar uno como este.

**5. Un PC con monitor, interfaz gráfica de usuario (GUI), conectividad red y Apache Directory studio**

### Configurar SLDAP

Hasta el momento, hemos supuesto que cualquier persona que usa nuestro clúster ODROID utiliza la cuenta de usuario `odroid` por defecto, que viene con las imágenes de Ubuntu Server pre-compiladas. Usar un único login es suficiente para uno (o quizás dos) usuarios, pero esta práctica es apenas segura y poco escalable para un clúster que se quiera ampliar. Para hacer frente a esta necesidad, vamos a instalar y configurar el Protocolo Ligero de Acceso a Direc-

torios (LDAP), servicio para gestionar los usuarios de nuestro clúster.

Por supuesto, si ya tiene un servicio LDAP ejecutándose en tu red, puedes omitir este paso y configurar los nodos para usar la autenticación de usuarios. Antes de instalar y configurar LDAP, tenemos que decidir cómo vamos a organizar nuestras cuentas de usuario. Aunque las configuraciones LDAP suelen variar según la organización, es una práctica muy común usar nombres de dominio para la base de búsqueda LDAP, como veremos a continuación. Para empezar, vamos a instalar el servicio LDAP en nuestro nodo principal de la siguiente forma:

```
sudo apt-get install slapd ldap-
utils
```

Durante la instalación, se te dará la oportunidad de elegir la contraseña de administrador para el servicio LDAP. Una vez completada la instalación, nuestro servidor LDAP aparecerá en blanco esperando ser configurado. Para empezar con nuestra configuración, ejecutaremos:

```
sudo dpkg-reconfigure slapd
```

Aunque al principio las opciones pueden resultar algo confusas, en nuestro caso podemos elegir las opciones por defecto para todo excepto los campos: Nombre de dominio DNS, Nombre de la organización y contraseña del administrador. El Nombre de dominio DNS y el Nombre de Organización dependen de ti, nosotros hemos usado “ocluster.rtcomputingsolutions.com” para ambas entradas (usa tu propio dominio y el prefijo será el nombre de tu clúster, que puede ser odroid o cualquier otro nombre que elijas).

¡Genial!, ya tiene un servidor LDAP funcionando. Desafortunadamente, todavía no tenemos ninguna información real almacenada en el servicio LDAP, como usuarios o información de Grupos. A pesar de que existen varias formas de crear esta información, como las herramientas de línea de comando. Nosotros personalmente preferimos usar Apache Directory Studio (<http://directory.apache.org/studio/>) para abordar esta configuración inicial. Después, será más adecuado utilizar scripts de línea de comandos, como el que se incluye al final de este artículo como anexo.

## Usuarios y Grupos

La siguiente cuestión que debemos afrontar es la creación cuentas de usuarios (autorización) para permitir que la gente se conecte usando sus credenciales (autenticación). Para utilizar Apache Directory Studio debemos crear una nueva conexión a nuestro servidor LDAP, que podemos hacer siguiendo estos pasos:

1. En la esquina inferior izquierda de ApacheDS, seleccione el icono LDAP y crea una nueva conexión.

2. Introduce un nombre para la conexión, luego el nombre del host o IP de tu nodo principal y selecciona “Check Network Parameter” para verificar que ApacheDS pueden comunicarse con tu servidor, y pulsa “Next”. Si ApacheDS no puede conectar, comprueba que tu máquina (PC) puede hacer ping al nodo principal y que slapd está funcionando.

3. En la pantalla de autenticación, introduce el login de administrador usando el Domain Component (dc), información que introduciste durante la instalación del slapd. Con la configuración slapd, introducimos “2cn=admin,dc=ocluster,dc=rtcomputingsolutions,dc=com” y la contraseña correspondiente que introducimos cuando ejecutamos `dpkg-reconfigure slapd` (la serie de secciones dc= serán diferentes dependiendo de la instalación). Después de verificar que “Check Authentication” es correcto, pulsa “Finish”.

Una vez más, tu configuración LDAP puede variar, pero en este caso crearemos dos unidades organizativas (UOs), denominadas “People” y “Groups” con ApacheDS. Para añadir una nueva entrada, haga clic derecho en nuestra entrada (dc, ex:dc=ocluster,dc=rtcomputingsolutions,dc=com) de Domain Component en el navegador LDAP y selecciona New -> New Entry.

Luego, crearemos nuestras UOs “People” y “Groups”:

1. Selecciona “Create Entry from scratch” y elige “Next>”

2. Busca y selecciona “organizational Unit” de la lista “Available object classes”, selecciona “Add” y luego “Next>”.

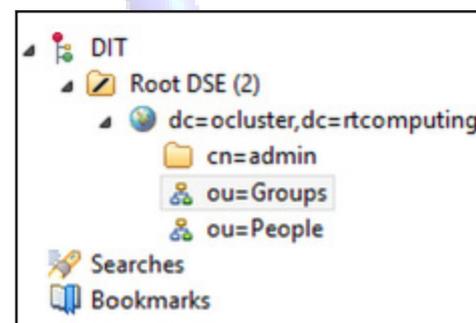
3. En la fila que contiene ‘RDN’, escribe “ou” en el primer campo, y el nombre de la Unidad Organiza-

tiva en el segundo campo después del “=” (en este caso, “People” o “Groups”) y pulsa “Next>”.

4. Revisa los Atributos y selecciona “Finish” cuando hayas terminado

5. Repite los pasos 1-4, introduciendo el otro nombre de la unidad organizativa del paso 3

Si todo ha salido bien, ahora debería ver una vista del navegador LDAP similar a la que se muestra a continuación:



**Navegador LDAP en acción**

Con estas dos Unidades de Organización en su sitio, podemos empezar a crear los usuarios y grupos para nuestro clúster. En este ejemplo, vamos a crear un usuario admin y dos grupos utilizando ApacheDS y después, usaremos un script en Python que se puede utilizar para crear nuevas cuentas desde la línea de comandos.

En primer lugar, crearemos dos grupos, uno llamado “ouser” y otro llamado “osudo”, ofreciéndonos un cierto control sobre quién tiene privilegios de administrador en nuestro clúster a través del uso ocasional de archivos sudoers. Podemos crear estos grupos siguiendo los siguientes pasos:

1. Haz clic derecho en ‘ou=Groups’ y ‘New -> New Entry’

2. Selecciona ‘Create entry from scratch’ y ‘Next >’

3. Desde la lista ‘Available object classes’, selecciona ‘posixGroup’, ‘Add’ y luego ‘Next >’.

4. En la fila RDN, introduce ‘cn’ = ‘osudo’, y selecciona ‘Next >’.

5. Después, se te solicitará un número

ro GID (identificador) para este grupo. El que elijas dependerá de ti pero te recomendamos que uses un número por encima de 1000 para evitar conflictos. Nosotros usamos “2001”. Tras introducir el valor elegido, pulsa “OK”.

6. Confirma tus atributos y pulsa “Finish” para crear el grupo.

7. Repite los pasos 1-6 para crear el segundo grupo, usa “ouser” y “2002” para el nombre y el GID respectivamente.

Con estos grupos creados, seguimos adelante y creamos una cuenta de administrador LDAP, ‘oadmin’:

1. Haz click derecho en ‘ou=People’ y selecciona ‘New -> Entry’

2. Selecciona ‘Create entry from scratch’ y ‘Next >’

3. Desde la lista ‘Available object classes’, selecciona y ‘Add’ los campos ‘posixAccount’ y ‘inetOrgPerson’, y luego ‘Next >’.

4. En la fila RDN, introduce ‘uid’ = ‘oadmin’ (donde uid corresponde al nombre de usuario) and selecciona ‘Next >’.

5. La siguiente ventana te pedirá que introduzcas el número UID (identificador) . Una vez más, te recomendamos que elija un número por encima de 1000 para evitar conflictos. Nosotros usamos “2200”. Tras introducir el valor elegido, pulsa “OK”.

6. En el menú Atributos, hay varios campos resaltados en rojo que debemos rellenar antes de crear la cuenta. En primer lugar, completamos el campo Common Name (CN) con el nombre de nuestro usuario, en este caso “Ocluster Admin”. El campo gidNumber es el grupo primario para este usuario, que en este caso será “2001” del grupo osudo. Luego, tenemos que rellenar el campo homeDirectory, que le dirá a la cuenta LDAP qué directorio usar para acceder a los

archivos específicos del usuario. En este caso, vamos a utilizar el directorio “/nethome/oadmin”, donde /nethome es la carpeta que vamos a configurar con AutoFS y NFS. El último campo es el campo Surname (sn), donde usaremos “Admin”.

7. Finalmente, añadiremos los atributos “userPassword” y “loginShell” a nuestra cuenta oadmin. Podemos añadir estos atributos haciendo clic derecho sobre la ventana y seleccionando “New Attribute”, escribiendo el nombre del atributo deseado y pinchando en “Finish”. Con el atributo userPassword, se te pedirá que introduzcas tu contraseña y tu método hash (te recomendamos que no utilices Texto plano). A continuación, puedes agregar el atributo loginShell repitiendo el paso anterior, usando ‘/bin/bash’ como valor.

8. Tras comprobar que todo está correcto, pulse “Finish” para crear tu nuevo usuario.

Si todo ha salido bien, la cuenta ‘oadmin’ “deberá aparecer bajo la Unidad “People”. A continuación, puede crear algunas cuentas más si lo desea, o puede echar un vistazo al script Python al final del artículo que se puede utilizar para crear usuarios sin el Apache Directory Studio. Antes de que podamos llegar a utilizar estas cuentas, necesitamos poner en marcha nuestro servidor NFS y configurar nuestros nodos para usar tanto LDAP como AutoFS.

## Configurar NFS

El Sistema de archivos de red (NFS) es algo esencial para cualquier clúster. Con NFS configurado, podemos acceder a los directorios de usuario montados en red, es decir, los archivos de los usuarios almacenados en los diferentes nodos del clúster. Además, podemos usar NFS para configurar las carpetas compartidas de uso general para los archivos binarios compilados o ajustar la configuración

del clúster. Para empezar a configurar nuestro servidor NFS, ejecuta:

```
sudo apt-get install nfs-kernel-server
```

Ten presente que nosotros nos hemos encontrado con problemas al intentar instalar nfs-kernel-server usando apt. Si este es el caso, intente compilar e instalar unfsd desde la fuente, que está disponible online. A continuación, necesitamos crear dos carpetas que podemos exportar desde el nodo principal y montar en los nodos secundarios, nethome y opt:

```
sudo mkdir -p /srv/nfs4/nethome
sudo mkdir -p /srv/nfs4/opt
sudo chmod 777 /srv/nfs4/nethome
```

En este caso, vamos a reemplazar el directorio existente /opt por nuestro directorio NFS y nethome será usado para almacenar los directorios de usuario. Con estos directorios creados, ahora tenemos que editar el archivo /etc/exports con nuestro editor de texto favorito (como root) y añadir las siguientes líneas:

```
/srv/nfs4/nethome
192.168.128.0/24(rw,sync,no_root_squash,no_subtree_check)
/srv/nfs4/opt
192.168.128.0/24(rw,sync,no_root_squash,no_subtree_check)
```

En resumen, el primer segmento indica la carpeta a compartir y el segundo los hosts a exportar a (nuestra red interna) y las opciones para la exportación. Con todas estas opciones en su lugar, ahora podemos reiniciar el servicio nfs:

```
sudo /etc/init.d/nfs-kernel-server restart
```

Finalmente, podemos “montar” estos directorios en el nodo principal con los siguientes comandos:

```
sudo ln -s /srv/nfs4/nethome/ /nethome
sudo rm -rf /opt
sudo ln -s /srv/nfs4/opt/ /opt
```

Con esta configuración en su lugar, podemos proseguir y configurar los nodos para usar AutoFS y LDAP.

## Configurar Autentificación Cliente LDAP

Por suerte, configurar LDAP en todos nuestros nodos del clúster es mucho más simple que configurar el servicio LDAP en el nodo principal; podemos empezar por instalar el paquete cliente ldap y nscd:

```
sudo apt-get install ldap-auth-client nscd
```

La instalación y configuración de ldap-auth-client te pedirá introducir información sobre tu servidor LDAP, te mostramos nuestras respuestas:

```
LDAP Server URI: ldap://odroid1.ocluster.rtccomputingsolutions.com
Search Base: dc=ocluster,dc=rtcomputingsolutions,dc=com
LDAP Version: 3
Local root Database admin: No
LDAP database require login?: No
```

Nos encontramos con algunas dificultades para configurar LDAP en el nodo principal al usar el nombre de dominio completo. Para evitar esto, sólo hay que poner localhost, aunque la longitud pueda variar. También hemos cambiado los nombres de host y nombres de dominio de nuestro clúster, donde odroid1 es ahora nuestro nodo principal en lugar de odroid-server0 y los nodos secundarios son nombrados con odroid2 y así sucesivamente.

Tras finalizar la instalación, ejecutaremos los siguientes comandos para completar la configuración:

```
sudo ln -s /lib/arm-linux-gnueabi/libnss_ldap.so /lib/libnss_ldap.so.2
sudo auth-client-config -t nss -p lac_ldap
sudo pam-auth-update
sudo reboot
```

Si todo ha ido bien, deberías ser capaz de iniciar sesión con las credenciales de la cuenta oadmin, o con cualquier otra nueva cuenta que hayas configurado. Por supuesto, todavía no tenemos creados los directorios de usuario (home) para nuestros usuarios. Afortunadamente, LDAP nos permite de forma automática crear estos directorios. Crea el archivo /usr/share/pam-configs/my\_mkhome como root y agrega el siguiente contenido:

```
Default: yes
Priority: 900
Session-Type: Additional
Session: required pam_mkhome.so
umask=0077 skel=/etc/skel
```

Ejecuta sudo pam-auth-update para activar la configuración. Al iniciar sesión como oadmin, tu directorio de usuario se creará de forma automática en /nethome/oadmin. Bastaría con sólo tener habilitado este directorio en el nodo principal, puesto que los usuarios tienen que conectarse al clúster a través del nodo principal (aunque tengamos este directorio habilitado en todos los nodos no nos creará ningún problema).

## Configurar Cliente AutoFS

AutoFS se usa conjuntamente con servicios NFS, ya que nos permite montar automáticamente recursos remotos. Para conseguir que nuestros recursos compartidos NFS queden montados en nuestros nodos secundarios (ya los hemos “montado” en nuestro nodo principal), tenemos que instalar autofs:

```
sudo apt-get install autofs
```

A continuación, necesitamos crear el directorio nethome que será uno de nuestros puntos de montaje NFS:

```
sudo mkdir /nethome
sudo chmod 777 /nethome
```

Por último, tenemos que modificar nuestros archivos de configuración autofs para montar los directorios opt y

## ¡BANG BASH! PORQUE LA LÍNEA DE COMANDOS NUNCA PASA DE MODA

por Bruno Doiche

**D**espués de habituarte a usar tus propios comandos a través del terminal, nada mejor que tener algunos trucos adicionales en la manga, ¿no? Tome nota de éstos que te ayudaran a ahorrar tiempo y harán que parezcas un experto de Linux.

### 1. Usa !!! para consultar el anterior comando:

```
$ apt-get install package
$ sudo !!
```

### 2. Buscar un determinado comando, pero sin ejecutarlo:

```
! <command> :p
```

## ENCUENTRA LO QUE NECESITES CON GREP UNA FORMA RAPIDA DE BUSCAR CONTENIDOS

por Bruno Doiche

**N**ecesitas editar ese archivo del sistema para una determinada serie, pero no recuerdas el archivo. Grep te puede ayudar simplemente escribiendo:

```
grep 'string` <path>
```

Puedes incluso ampliar tu búsqueda con los argumentos -i y -R. -i para ignorar caso, -R para encontrar de forma recursiva en los directorios. Sólo recuerda que grep no distingue entre los tipos de archivo, así que excluye los archivos multimedia para ahorrar tiempo.

nethome. Los archivos que hemos modificado (o creado) y sus contenidos han sido mostrados:

```
/etc/auto.master
+dir:/etc/auto.master.d
+auto.master
/nethome /etc/auto.nethome
--timeout=60
/- /etc/auto.direct
--timeout=60
```

```
/etc/auto.nethome
* odroid1.occluster.rtcomputingsolutions.com:/srv/nfs4/nethome/&
```

```
/etc/auto.direct
/opt odroid1.occluster.rtcomputingsolutions.com:/srv/nfs4/opt
```

Con esta información en su sitio, podemos guardar esta configuración autofs con el siguiente comando:

```
sudo service autofs restart
```

Si todo está configurado correctamente, deberías ser capaz de agregar archivos en /opt y /nethome (permisos pendientes) y verlos en los demás nodos del clúster que tienen montados la mismos recursos comparados. Si AutoFS no se inicia al reiniciar puedes añadir “/usr/sbin/service autofs restart” a /etc/rc.local.

## Conclusiones

Al principio es necesario un cierto esfuerzo para configurar nuestro clúster y hacerlo escalable (y poco trabajo para mantenerlo). A pesar de los pasos y las configuraciones mostradas en nuestros artículos, existe un margen bastante amplio para los fallos siendo necesaria una mayor depuración. Incluso si no se está desarrollando un enorme clúster; LDAP, NFS y autofs a menudo forma parte del conjunto de herramientas de un administrador de Linux, lo que hace de éste un esfuerzo que vale la pena en sí mismo. Con es-

tos servicios en su sitio y la incorporación de tres nodos secundarios, disponemos de una infraestructura ARM propiamente dicha. En futuros artículos, mostraremos posibles usos que podemos dar a nuestro clúster, como la utilización de MPI y otras formas de mejorarlo, como el uso de Títeres.

## Agradecimientos

Nos gustaría dar las gracias al Dr. Kenneth Sloan, director del Laboratorio de impresión 3D del Departamento de Ciencias de la Información y la Computación de la Universidad de Alabama en Birmingham (<http://bit.ly/1j2AOjI>). Con su apoyo, hemos podido imprimir en 3D el clúster que se muestra en la imagen en la página 6.

## Código de ejemplo

Ten en cuenta que esto es sólo un pequeño ejemplo que hemos utilizado. No dudes en ampliarlo para crear nuevos usuarios. Además requiere el paquete ‘python-ldap’.

```
addUser.py
#!/usr/bin/python
import base64
import getpass
import hashlib
import ldap
import ldap.modlist as modlist
import os
import random
import string
import sys

# LDAP Vars
HOST="ldap://localhost:389"
LDAP_BASE = "dc=occluster,dc=rtcomputingsolutions,dc=com"
ADMIN_CN = "cn=admin," + LDAP_BASE
PEOPLE_BASE = "ou=People," + LDAP_BASE
ADMIN_PASS = getpass.getpass("LDAP Admin Password:")
```

```
# Group Vars
OUSER_GID = "2002"

def addUser():
    """
    try:
        ldapCon = ldap.initialize(HOST)
        ldapCon.simple_bind_s(ADMIN_CN, ADMIN_PASS)

        attrs = getUserAttributes()
        ldif = modlist.addModlist(attrs)
        ldapCon.add_s("uid="+
+ attrs['uid`] + "," + PEOPLE_BASE,ldif)
        ldapCon.unbind_s()
        print "Account created."
    except ldap.LDAPError, e:
        print e
        sys.exit()
    print "Done."

def getUserAttributes():
    """
    attrs = {}
    attrs['objectClass`] = ['inetOrgPerson`, 'organizationalPerson`,
        'person`, 'posixAccount`, 'shadowAccount`, 'top`]
    attrs['cn`] = raw_input("User Full Name: ")
    attrs['uid`] = raw_input("User uid: ")
    attrs['uidNumber`] = raw_input("User uidNumber: ")
    attrs['gidNumber`] = OUSER_GID
    attrs['sn`] = attrs['cn`].split(' ')[-1]
    attrs['userPassword`] = generateLdapPassword(getpass.getpass("User Password: "))
    attrs['homeDirectory`] = '/nethome/` + attrs['uid`]
    attrs['loginShell`] = '/bin/
```

```

bash`
  attrs['mail`] = attrs['uid`]
+ "@rtcomputingsolutions.com"
  return attrs

def printUserAttributes(attrs):
    """
    for key in attrs.keys():
        if key == "objectClass":
            print attrs[key]
        else:
            print key + "= " +
attrs[key]

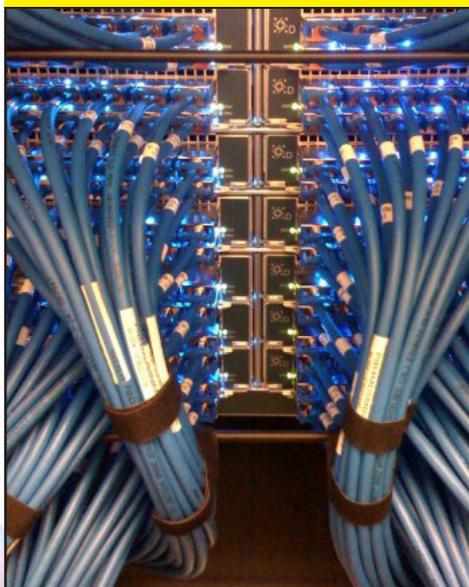
def
generateLdapPassword(password):
    """
    salt = generateSalt(6)
    m = hashlib.sha1(password)
    m.update(salt)
    return "{SSHA}" + base64.
b64encode(m.digest() + salt)

def generateSalt(size=10,
chars=string.ascii_letters +
string.digits):
    """Generate random temporary
password for web user"""
    return ''.join(random.
choice(chars) for x in
range(size))

if __name__ == "__main__":
    addUser()

```

Las cosas se nos fueron de las manos cuando creamos nuestro clúster odroid en las oficinas



## COMO USAR LOS MANDOS DE LA XBOX 360 CON ANDROID CONFIGURARLOS CON RETROARCH Y DISFRUTA CON MILES DE JUEGOS

por Rob Roy

El mando de la Xbox 360 funciona de forma nativa con el sistema operativo Android y se puede utilizar para navegar por los menús, ejecutar aplicaciones y lo más importante, jugar a los juegos. He probado varios mandos y emuladores de juegos con Android que se pueden ejecutar en ODROID, incluyendo el mando Bluetooth de PS3, el mando de la Wii y un mando de Xbox 360 con cable e inalámbrico. De todos los modelos que he probado, el modelo inalámbrico Xbox 360 es el más cómodo y compatible con la plataforma Android.

Si ya dispone de mandos inalámbricos Xbox 360, simplemente conecta un receptor USB para usarlos con tu ODROID (<http://amzn.to/1bJZv6q>). La interfaz inalámbrica soporta hasta 4 mandos por receptor USB.

Para conectar los mandos, primero pulsa el botón "connect" en el receptor USB inalámbrico hasta que la luz verde parpadee. Luego, presione el botón "connect" en el mando, que se encuentra entre los botones de disparo izquierdo y derecho junto a la conexión de carga, delante del botón "back". El indicador 1-2-3-4 parpadeará en círculo y finalmente, parpadeará muy lentamente para indicar que la conexión se ha completado. Aunque el mando normalmente permanece estable cuando se conecta a una consola Xbox, el indicador 1-2-3-4 seguirá parpadeando incluso cuando esté conectado el mando. Repite todo este proceso hasta que todos los mandos sean registrados por el receptor USB.

Para comprobar si los mandos funcionan, ve al escritorio de Android y usar el joystick para navegar por los iconos del menú principal, el botón X para lanzar aplicaciones y el botón B para volver a la pantalla anterior. El mando también funciona en XBMC sin ninguna configuración adicional. Tras comprobar que el mando responde correctamente, ya está listo para probarlo con juegos.

El emulador de consolas más versátil y completo para Android es Retroarch, compatible con los mandos de la Xbox 360. Es uno de los mejores emuladores de juegos disponibles en Play Store, e incluye los siguientes emuladores: Atari 2600, PlayStation 1, Super Nintendo, Nintendo Entertainment System, GameBoy, GameBoy Color, GameBoy Advance, Arcade, Neo Geo Pocket, Virtual Boy, Sega genesis/Mega Drive, Sega Master System, Quake, Doom, Sega Game Gear y muchos más. Retroarch puede aceptar hasta 4 mandos, lo que hace que sean más divertidos juegos como los multijugador de Mario Kart.

Para configurar Retroarch para los mandos de Xbox 360, selecciona "Input Options" en el menú principal, desactiva "Touchscreen Overlay", luego, desplázate hacia abajo hasta "Player 1 Custom Binds". Selecciona cada botón de la pantalla y pulse el botón correspondiente en el mando 360 que deseas utilizar para controlar el Jugador 1. Repita el proceso para el resto de mandos (hasta un máximo de 4). Una vez que los botones estén asignados, puedes ejecutar cualquier juego que soporte Retroarch sin tener que volver a configurar los mandos.

# CONVERTIR UNA IMAGEN S.O. PARA EJECUTARLA EN UN ODROID-XU

ACTUALIZA TU SOFTWARE LINUX PARA QUE PUEDA ACEPTAR TU NUEVO Y BRILLANTE HARDWARE

por Mauro Ribeiro

Este artículo describe cómo trabajar con imágenes con S.O. Linux de la serie X y U, tales como Ubuntu o Debian y adaptarlas para que puedan ejecutarse en el XU. Esto puede ser útil cuando actualices tu hardware a la serie XU, o si deseas una imagen que sólo está disponible para modelos anteriores de ODROID y quieres actualizarla para que funcione en los últimos modelos: XU, XU-Lite y XU +E.

**1. Descarga, copia o crea un archivo de imagen de Ubuntu desde una tarjeta SD o eMMC que este diseñado para ejecutarse en las series U o X, como el U3 o X2.**

**2. Pasa la imagen como lo harías normalmente a una tarjeta SD.**

```
dd if=XXXXX.img of=/dev/sdX bs=4M
&& sync
```

**3. Descarga un nuevo Gestor de Arranque y el archivo de muestra boot.ini**

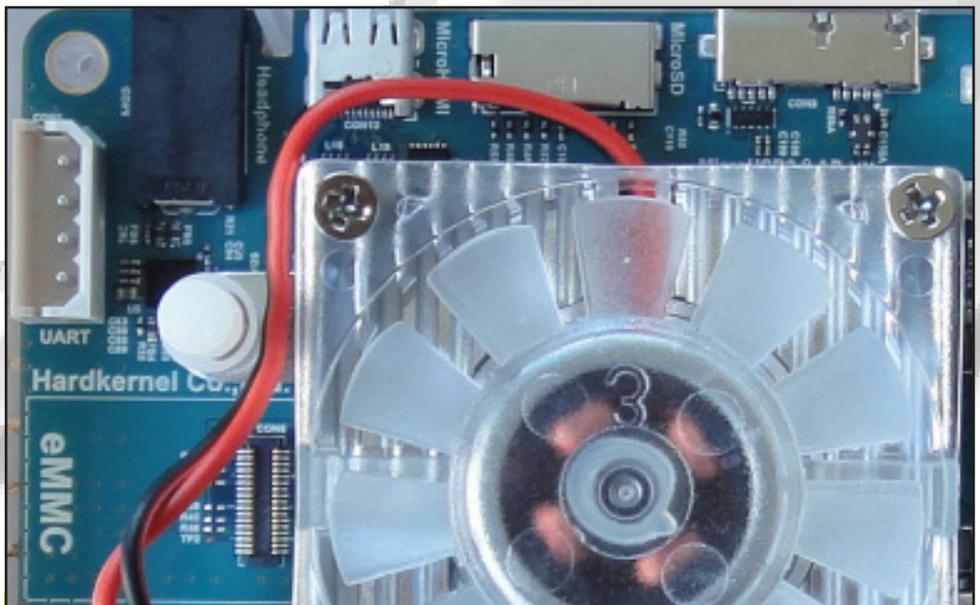
```
wget builder.mdrjr.net/tools/
uboot-xu.tar
```

**4. Descomprime el fichero uboot-xu.tar**

```
tar xf uboot-xu.tar
```

**5. Fusiona la sdcard con los nuevos Gestores de Arranque**

```
sh fusing.sh /dev/sdX
```



El ODROID-XU es el top IO de la familia de hardware SBC de Hardkernel. La mayor parte del software que se ejecuta en las series X y U puede ser adaptado para que funcione en el modelo XU

**6. Dentro del paquete del gestor de arranque, copia los archivos boot.ini y u-boot.bin a la partición de inicio FAT32 de la tarjeta SD.**

```
cp boot.ini /media/boot/
cp u-boot.bin /media/boot/
```

**7. Descarga el kernel para el XU.**

```
wget http://builder.mdrjr.net/
kernel-3.4/00-[...]xu.tar.xz
```

**8. Descomprime el kernel y copia los archivos en los directorios correctos.**

```
tar -Jxf odroidxu.tar.xz
```

**9. Copia boot/zImage a la partición fat de la SD y el directorio lib a la carpeta /lib de la partición ext4 de la tarjeta sd.**

**10. Elimina el viejo archivo boot.scr de la partición FAT32.**

**11. Intente arrancar, posiblemente funcionara con normalidad.**

**12. Activa el driver HDMI. Es posible que necesites algunas dependencias, dependiendo de tu entorno.**

```
git clone --depth 0 https://
github.com/hardkernel/linux.git
-b odroidxu-3.4.y odroidxu-3.4.y
cd odroidxu-3.4.y/tools/hardker-
nel/exynos5-hwcomposer
sh autogen.sh
./configure --prefix=/usr
make -j5 && make install
```

**13. Crea un script de inicio automático para exynos5-hwcomposer creando el archivo /etc / init/exynos5-hwcomposer.conf con el siguiente contenido:**

```
description "Exynos5 HW Composer"

start on started lightdm

# start on runlevel [2345]
exec /usr/bin/exynos5-hwcomposer
```

**14. Activa el driver HDMI. Es posible que necesites algunas dependencias, dependiendo de tu entorno.**

```
git clone --depth 0 https://github.com/hardkernel/linux.git -b odroidxu-3.4.y odroidxu-3.4.y
cd odroidxu-3.4.y/tools/hardkernel/exynos5-hwcomposer
sh autogen.sh
./configure --prefix=/usr
make -j5 && make install
```

**12. Crea un script de inicio automático para exynos5-hwcomposer creando el archivo /etc / init/exynos5-hwcomposer.conf con el siguiente contenido:**

```
description "Exynos5 HW Composer"

start on started lightdm

# start on runlevel [2345]
exec /usr/bin/exynos5-hwcomposer
```

**15. Prepara xorg.conf asegurándote que tu / etc/X11/xorg.conf es similar a esto:**

```
Section "Device"
    Identifier "FBDEV"
    Driver "fbdev"
    Option "fbdev" "/dev/fb0"
EndSection

Section "Screen"
    Identifier "Default Screen"
    Device "FBDEV"
    DefaultDepth 24
EndSection
```

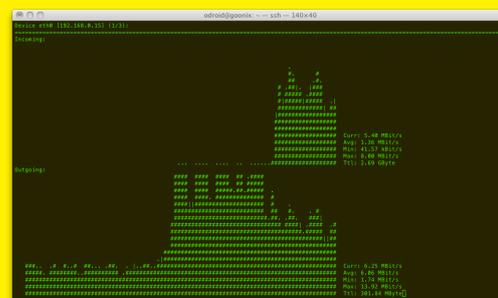
**16. Reinicia. Ahora todo debería funcionar correctamente**

Una vez que verifiques que la imagen se ejecuta correctamente en XU, asegúrate de crear una imagen del disco con el sistema operativo. Si la instalación no se ejecuta, repite los pasos con una nueva copia de la imagen de la serie U o X original. Si tienes algún problema o necesitas ayuda, puedes plantear tus cuestiones en los foros Hardkernel en <http://forum.odroid.com>.

## CONTROLA TU ANCHO DE BANDA CONOCE TU ESTADO DE INTERFAZ E/S

por Bruno Doiche

Conocer el rendimiento de tu red es tan fácil como mirar en el monitor del sistema de Gnome. Pero ¿Qué pasa con un ODROID al que solo tienes acceso a través del terminal? No te preocupes, instala nload y sigue la pista a tu interfaz de red.



## SEGURIDAD CONTROLA EL ACCESO DE USUARIOS

por Bruno Doiche

Aumenta la seguridad de tu servidor configurándolo correctamente. Si no hay necesidad de que un determinado usuario acceda a tu máquina vía SSH, asegúrate de editar el archivo /etc/ssh/sshd\_config y agrega la siguiente estrategia:

```
AllowUsers <users to be allowed to logon>
```

```
DenyUsers <users to be denied logon>
```

¿Por qué negar el acceso al usuario en lugar de simplemente eliminarlo? A veces, un usuario sólo necesita acceso por SMB o FTP a tu servidor. Esta buena práctica da tranquilidad a tu sistema.

# SO DESTACADO: GAMESTATION TURBO

## UN VISTAZO AL INTERIOR DE LA POPULAR IMAGEN DE JUEGOS Y CENTRO MULTIMEDIA

por Tobias Schaaf

GameStation Turbo para U2/U3/X/X2 puede descargarse desde <http://bit.ly/VolpVL>. Vigila los foros ODROID en <http://forum.odroid.com/> para proximas versiones actualizadas, incluyendo la última versión del emulador PPSSPP

Uno de los mayores proyectos en los que estoy trabajando para la comunidad ODROID es la imagen ODROID GameStation Turbo, con una interfaz Linux y pensada para los juegos y la reproducción multimedia. Un completo sistema de entretenimiento que te permite controlar tu ODROID sólo con mandos de juegos, sin tener que tocar el teclado para ver películas, escuchar música o jugar a tus juegos favoritos.

Para entender mejor esta imagen y todo lo que ofrece, me gustaría ofrecerte una visión interna de cómo llego a crearse la imagen, que motivo su desarrollo y cómo puedes adaptarla para satisfacer tus propias necesidades.

### Motivación

El primer dispositivo basado en ARM que tome en cuenta fue en reali-

dad Open Pandora, pero en el momento que decidí comprarlo, ya no estaba disponible. No obstante, cuando volvió haber existencias, resultaba tan caro que no podía permitírselo. Finalmente, cuando reuní el dinero suficiente, me volví más escéptico y empecé a buscar otras opciones.

La placa Pandora era un dispositivo de un único núcleo basado en ARM con sólo 1 GHz y 512 MB de RAM por 700\$, así que ¿Realmente merecía la pena? Pues bien, aunque la comunidad ha sido y sigue siendo impresionante, y es un dispositivo totalmente portátil (como la Nintendo DS), en mi opinión, era demasiado caro para lo que podría ofrecer. Para entonces había mejores alternativas, como el ODROID.

Después de ver el ODROID-X2 en una página de German IT News, me enganche al fenómeno ODROID. En el

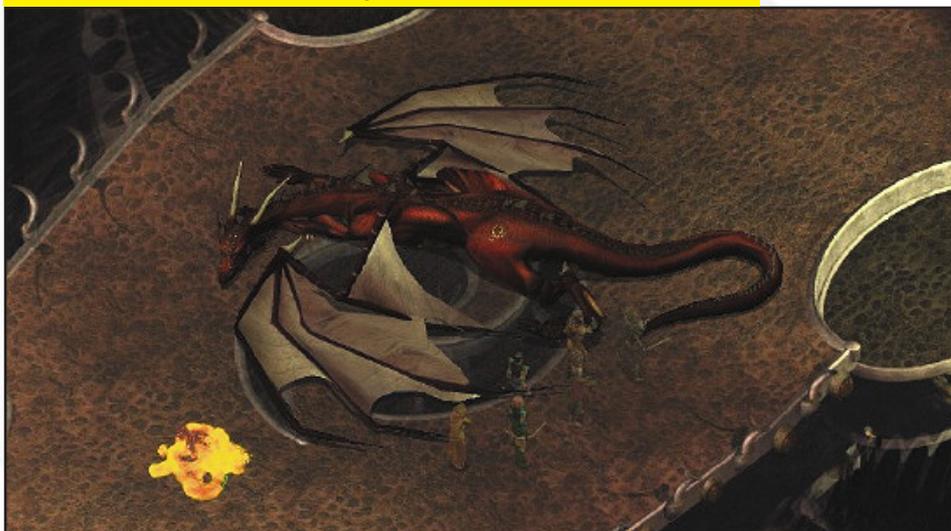
momento en el que Ubuntu fue anunciado para ODROID, me compré un X2.

Sin embargo, lo que yo quería era algo similar a Pandora para jugar y ODROID no tenía demasiados juegos en ese momento (2012). A medida que mis sobrinos se hacían mayores, pensé en que podía hacer algo para ellos y que fuese creciendo como ellos. En primer lugar, ODROID pudo ser una consola para jugar y más tarde, se convertiría en un PC para hacer los deberes y aprender Linux. Este fue mi objetivo y motivación para crear la imagen.

### Pasos para lograrlo

El primer paso para alcanzar esa meta era generar el contenido, por lo que muchos juegos y emuladores tuvieron que ser migrados a ODROID. Si ojeas la sección de Ubuntu de los Foros ODROID, encontraras muchos juegos y programas que han sido exportados por mí. Fue un trabajo duro, ya que pasé de no saber prácticamente nada sobre como exportar juegos y compilar software bajo Linux a lo que sé ahora. ODROID fue de gran ayuda en el aprendizaje de nuevas habilidades y me permitió mejorar mis conocimientos sobre Linux. Ahora, sé cómo optimizar ciertos programas, cómo configurar diferentes parámetros de optimización y cuando son necesarios esos indicadores. Aprendí cómo funcionan las CPUs ARM y especialmente, cómo trabajan las placas Hardkernel y sus diferencias.

Baldur's Gate es uno de los juegos de PSX que ahora es compatible con la última versión de GameStation Turbo, gracias a la nueva versión de PPSSPP.



Mi primer proyecto fue por tanto, exportar una gran cantidad de juegos y también compilar algunos emuladores. Si has leído mis artículos en números anteriores de ODROID Magazine, habrás encontrado una gran cantidad de información sobre los juegos que pueden ejecutarse en ODROID, y la lista sigue creciendo.

El siguiente gran paso fue facilitar, incluso para los niños, el uso de Linux y la ejecución de los juegos. Comencé a desarrollar Gamestation Turbo desde la imagen Linaro Ubuntu 12.04. Elegí este sistema por el escritorio Unity.

Unity es fácil de usar y entender, incluso para las personas que nunca antes han utilizado Linux. Tal vez no sea el mejor entorno de escritorio para todas las aplicaciones, pero es colorido y fácil de manejar. Para alguien que nunca ha usado Linux, es una muy buena manera de empezar a trabajar con él.

Lo primero que necesitaba era conseguir que los programas fuesen lo bastante fácil de ejecutar en todos los sistemas. Todas las aplicaciones y juegos que he creado están incluidos en un archivo de icono .desktop, así que puedes buscarlo en Unity o simplemente colocar un acceso directo en el escritorio. Esto funcionó muy bien para los juegos, pero no para los emuladores, puesto que éstos normalmente utilizan su propia interfaz de explorador de archivos para cargar las ROMs. Aunque los adultos somos capaces de iniciar de forma manual todos los juegos, los niños no tienen ni idea de lo que significan ciertas palabras, y les resulta difícil ver los juegos que están disponibles, o simplemente buscar en Unity. Estaba claro que necesitaba algún tipo de interfaz para iniciar los juegos.

Ya había utilizado XBMC en un viejo PC que actuaba como Equipo de Cine en Casa (HTPC), y con posterioridad descubrí un buen componente para XBMC llamado Rom Collection Browser (RCB). RCB permite organizar las ROMs del emulador de la misma forma que organizas tu colección de videos.



**GameStation Turbo te permite elegir entre una gran variedad de emuladores, además de reproducir videos a 720p, todo ello desde el XBMC usando sólo un gamepad.**

Incluso permite descargar imágenes de vista previa, portadas y una breve descripción de los juegos, al igual que lo haces para tus películas y series.

Consciente de ello, me surgió la idea de usar XBMC como interfaz y configurarlo de forma que los niños pudiesen jugar y divertirse, o mejor dicho configurarlo de forma que incluso un niño pueda utilizarlo. Durante ese tiempo, la reproducción de video por hardware no estaban disponibles, puesto que su desarrollo aún no estaba completado.

La versión de XBMC que venía con Ubuntu 12.04 era XBMC 11 (Eden) funcionaba, pero no es muy rápida a causa de la decodificación por software. Aunque el menú funcionaba sin problemas, la reproducción de vídeo no era muy fluida. Aun así, funcionaba lo bastante bien como para poder testear Rom Collection Browser y probar diversas configuración. Cuando salió la primera imagen de XBMC 12 (Frodo) para ODROID, todavía no soportaba la reproducción de películas por aceleración de hardware, pero venía con soporte para OpenGL ES 2.

Las cosas se pusieron un poco difíciles durante algún tiempo, ya que la compilación del código fuente del XBMC de Hardkernel no me servía y la versión facilitada no tenía soporte para joystick, algo que consideraba fundamental para mi proyecto. Decidí que, ya

que estaba planteada como una plataforma de juegos, la reproducción de vídeo no iba a ser demasiado importante, pudiendo reproducir cualquier video HD siempre y cuando fuese a 720p o inferior. Para los niños, por lo general, no importa si su anime o viñeta favorita está en HD o simplemente en SD.

Pues bien, fue en esa época cuando se liberó una imagen de XBMC acelerado por hardware y fui capaz de rehacer la imagen con soporte de joystick. Poco antes de que liberara la primera versión de GameStation Turbo, le incorpore una versión XBMC totalmente funcional.

## Piezas incluidas

Después decidí cómo iba a funcionar la imagen, era importante agrupar todas las piezas pequeñas en una imagen muy bien empaquetada y para ello, eran necesarios diferentes tipos de programas.

La primera prioridad era el sistema operativo, el cual tenía que ser muy estable, fácil de mantener y con una interfaz que le fuese familiar a la mayoría de la gente. La elección estaba entre Ubuntu 12.04 y Debian Wheezy. El resto de imágenes eran inestables (Debian Jessie/Sid) o no tenían soporte a largo plazo (Ubuntu 13.04 o el reciente 13.10).

Ubuntu 12.04 es una versión LTS con soporte hasta 2017, lo cual siempre es bueno, sin embargo, Debian Wheezy funciona mejor que Ubuntu 12.04.

También me di cuenta que en Debian Wheezy, los programas podían posiblemente ejecutarse en Ubuntu 12.04 o superior sin fallos, pero no al revés. Por lo tanto, decidí usar Debian Wheezy y LXDE, que usa menos de 150 MB de RAM, incluso con XBMC y un par de programas. Después, era cuestión de reunir el software correcto para convertir ODROID en una consola de juegos.

## Rom Collection Browser

Use Rom Collection Browser como base para instalar diferentes tipos de emuladores tales como Retroarch, Mednafen, PPSSPP y ScummVM. Una vez finalizada la configuración básica, resultó que no todos funcionaba con un gamepad, así que añadí la aplicación AntiMicro que es capaz de asignar ciertas teclas a botones del joystick para cubrir las lagunas en las que los drivers del joystick no funcionaban.

Además, mantengo mi propio kernel que también incluye los archivos cabecera, ya que algunas partes del kernel proporcionado por Hardkernel no satisfacían mis necesidades y no incluían los archivos cabecera. Además, había una enorme diferencia de espacio entre el Kernel de hardkernel y el que yo desarrollé. El tamaño del Kernel de Hardkernel era de 300MB y el mío era de tan sólo 16MB, que logre eliminado los módulos

innecesarios. Mis scripts también permiten a los usuarios instalar o desinstalar paquetes del kernel, en lugar de limitarse a copiar el kernel directamente sobre los ficheros existentes.

## Complicaciones

El mayor problema fue conseguir que todas las piezas encajaran, con el fin de facilitar a la gente el uso de la imagen, incluso sino disponían de conocimientos sobre configuración. Rom Collection Browser es un poco complicado de usar para un principiante, ya que tienes que elegir el emulador, configurar parámetros y dar a los archivos ROM la extensión estándar para conseguir que funcionen. De modo que, tuve que buscar un sistema que facilitase esta labor al usuario.

Había otra cuestión. Yo desea tener el control total del Joystick (GamePad), pero algunos emuladores requerían teclas adicionales, como Retroarch y Mednafen que necesitaban la tecla ESC para finalizar el juego y regresar a XBMC, y los juegos de MAME que necesitaban pulsar "OK" para continuar.

## Configuración

Un problema con la configuración previa del Rom Collection Browser era que necesitaba la ruta completa de la ubicación del emulador y las ROMs, y la extensión de archivos para buscar las

ROMs. Esto puede resultar complicado para alguien que nunca ha trabajado con Rom Collection Browser. Esa es la razón por la que he preseleccionado el emulador y los juegos, y he creado una estructura de carpetas donde serán ubicadas las ROMs, para que el Rom Collection Browser pueda localizar los juegos.

Se pueden añadir emuladores adicionales presionando la tecla C en el Rom Collection Browser y seleccionando "Add a new ROM collection". Tienes que proporcionar la ruta de acceso al emulador y a la ROM, la ruta donde se almacena la información y las imágenes, y la extensión de las ROMs.

El archivo de configuración de ROM Collection Browser se almacena en:

```
/home/odroid/.xbmc/userdata/addon_data/script.game.rom.collection.browser/config.xml
```

Editando este archivo, puede modificar varias opciones, tales como, si un archivo .zip debe ser extraído en una carpeta temporal, o si hay que buscar una ROM dentro de un archivo .zip (que, por ejemplo, tiene que ser desactivado para los juegos de MAME). Si tienes experiencia suficiente, puedes incluso agregar nuevas colecciones directamente en este archivo.

## Iniciar un Emulator

Aunque es posible iniciar una ROM directamente a través del emulador, tiene algunos inconvenientes. En primer lugar, XBMC continuará ejecutándose en segundo plano y usará algunos recursos que son necesarios para mejorar la experiencia de juego. En segundo lugar y como ya he mencionado, algunos emuladores necesitan teclas adicionales que no están asignadas a botones. Si usas un joystick que no es compatible, necesitaras la aplicación antiMicro para asignar los botones correspondientes. Si es así, tienes que asegurarte de que antiMicro se inicia cuando lo necesites, que no siempre es el caso. Iniciar directamente

**DOOM3 (PSX) presume de los gráficos de ODROID dando rienda suelta a esta criatura dentro de tu ordenador, y luego sólo te dan una palanca para la defenderte.**



antiMicro junto con el emulador tampoco funciona.

Para resolver estas y otras cuestiones, hice que XBMC ejecutara un pequeño script en lugar de iniciar directamente el emulador. En este script, se inicia el emulador y luego ejecuta el archivo ROM como un parámetro de Línea de Comandos desde XBMC. De esa forma, pude definir diferentes pasos para asegurarme que el emulador funcionase de la mejor forma.

Ejemplo: Este script permite ejecutar un juego de SNES con Retroarch:

```
#!/bin/sh
/usr/bin/killall -STOP xbmc.bin
if [ `ps aux | grep antimicro |
grep -v grep | wc -l` -lt 1 ];
then
    antimicro --tray --profile
/home/odroid/joydev.xml &
else
    /usr/bin/killall -CONT
antimicro
fi
/usr/local/bin/retroarch -L /
usr/local/share/retroarch/cores/
working/snes9x_next_libretro.so
"$1"
/usr/bin/killall -CONT xbmc.bin
/usr/bin/killall -STOP antimicro
```

Si leemos el código anterior, observamos que XBMC está en modo suspensión, lo que significa que no realiza ningún proceso mientras ejecutamos nuestro emulador. Después, compruebo que antiMicro se está ejecutando, y o bien se carga con el archivo del perfil requerido, o se reanuda en caso de que aún este ejecutándose. Luego, activo el emulador en sí. Aquí puedo usar los parámetros de comando que me permiten configurar el emulador. Una vez que el emulador ha terminado y nos salimos de éste, XBMC se reanuda y antimicro se suspende. Justo después de que el script haya finalizado por completo, cambia el control a XBMC. Esto te permite algo



**Homeworld 2 (PSX) ¡El mejor 3D RTS con naves espaciales futuristas! Este juego es muy divertido, pase varias horas jugando cuando salió por primera vez.**

de limpieza que suele venir muy bien.

Escribí unos pocos scripts para adaptar las diferentes circunstancias. Por ejemplo, el script para ScummVM y Amiga es un poco más complicado, pero por lo general es casi siempre el mismo.

1. Suspende los procesos que no necesitas (por ejemplo XBMC)
2. Configura tu entorno preparando el sistema con lo que necesitas (ejemplo: carga antimicro con el perfil adecuado).
3. Activa el emulador y le proporciona los parámetros que necesita. El "\$1" representa el archivo de ROM que está recibiendo XBMC como un parámetro.
4. Haz algo de Limpieza y reanuda los procesos que dejaste en suspensión al principio.

Todos los scripts que he usado para lanzar emuladores se encuentran en **usr/local/bin/**, donde puedes revisar, mejorar o añadir tus propios script.

## FAQ

De vez en cuando, recibo algunas preguntas sobre mi imagen a las que me gustaría responder desde aquí:

*¿Dónde tengo que colocar los archivos ROM para mis juegos?*

Navega hasta **/home/odroid/ROMS**, donde encontrarás una estructura de carpetas ya creada para cada

tipo de ROM al que quieras jugar, como GBA y SNES. Por favor, consulta el post del foro en <http://bit.ly/1nVvQqz> para obtener más información sobre las extensiones de archivos soportadas.

*¿Hay alguna forma de cargar ROMs desde un dispositivo de almacenamiento externo?*

Copia el contenido de **/home/odroid/ROMS** a tu dispositivo de almacenamiento externo y luego auto-monta el dispositivo externo en **home/odroid/ROMS** añadiéndolo a **/etc/fstab**, o usando **/etc/rc.local** para hacerlo permanente.

*¿Qué joysticks/gamepads son compatibles?*

Desarrolle la imagen para usarse con un mando inalámbrico Xbox 360 y el receptor USB Xbox 360. Así que si tienes este hardware, la imagen debe funcionar sin tener que realizar modificaciones, a menos que se me olvidara algo.

Además, todos los joystick/gamepad compatible con Linux deberían funcionar, pero tendrás que adaptar la configuración para tu dispositivo. Por lo tanto tienes que cambiar la configuración del joystick en los emuladores.

Con Mednafen, sólo tiene que pulsar ALT + SHIFT +1 para configurar los controles de tu dispositivo. El programa de configuración es fácil de entender. El segundo jugador, si es compatible con el emulador se puede configurar con ALT

# GENERADORES FRACTAL PARA LINUX

por Rob Roy

El mes pasado, presentamos algunas de las aplicaciones matemáticas disponibles para la plataforma ODROID a través del Centro de Software de Ubuntu. Puesto que los fractales han sido siempre uno de mis favoritas ramas de las matemáticas, he probado algunos disponible a través del Gestor de Paquetes de Synaptic y apt-get. Para empezar, escriba lo siguiente en una ventana de terminal

```
sudo apt-get install xaos \
mandelbulber xmountains xfractint
```

## Xaos

Genera clásicos fractales bidimensionales y tiene un curioso botón "Load Random Example" en el menú archivo. Puede seleccionar diferentes fórmulas desde el menú y ajustarlas para ver qué efecto tiene en el diseño general.

## Mandelbulber

Crea bonitos fractales en 3D con sólo pulsar el botón "Render" en la parte superior de la pantalla. Cambiar el tipo de fractal seleccionando "Fractal Formula Type" desde la pestaña Fractal. Haz clic en cualquier área de la imagen del fractal para acercar y recalcular la vista a una resolución más alta. Asegúrate de que tu ODROID tiene un ventilador, ya que éste requiere de una alta capacidad de procesamiento.

## Xmountains

Dibuja cadenas de montaña realistas, con nieve. No hay nada más que hacer que disfrutar de las montañas matemáticas.

## Xfractint

Es un generador fractal basado en teclado que puede producir todo tipo de fractales. Para obtener más información sobre cómo configurar Xfractint, visita <http://bit.ly/1s00x1x>.

+ SHIFT + 2, y así sucesivamente.

Para Retroarch, es un poco más complicado. Salte del XBMC, abre un terminal, escriba `retroarch-joyconfig` y siga las instrucciones que aparecen en la pantalla. Al final, recibirás una larga lista de parámetros de configuración. Copia esta lista y abre el archivo `/home/odroid/.config/retroarch/retroarch.cfg`, donde localizarás los mismos parámetros. Reemplaza los parámetros ya existentes por lo que obtuviste con `retroarch-joyconfig`. Ahora el dispositivo debería funcionar en Retroarch.

XBMC lamentablemente no es compatible con muchos dispositivos con joystick. Aunque los mandos de la Xbox 360 funcionan muy bien, otros en cambio no hacen nada.

Con PPSSPP, puedes cambiar la configuración del mando con sólo entrar en el menú. Sin embargo, en PPSSPP, el sistema de controles implementado es bastante lento por lo que sólo unos pocos llegan a funcionar. En el peor de los casos, ni siquiera se puede utilizar el teclado. Los ajustes del mando no te permitirán pulsar ciertas teclas de dirección. Si eso ocurre, elimine el archivo:

```
/home/odroid/.config/ppsspp/PSP/
SYSTEM/controls.ini
```

Y vuelta a empezar. Si todo lo demás falla, elimina cualquier asignación del mando y mantén la configuración de solo teclado. A continuación, utiliza tu mejor aliado, antiMicro. Si usas un dispositivo de joystick diferente y tienen problemas para que funcione con PPSSPP o XBMC, antiMicro te ayudara. Usando antiMicro, puede asignar comandos de teclado a los botones de tu gamepad/joystick del mismo modo en que simplemente asignas las teclas de flecha del teclado a tus gamepads.

*¿La imagen tiene soporte para CEC?*

No en principio. La imagen tiene libcec instalado el cual funciona en HDMI 1, pero elimine CEC de XBMC

porque causaba problemas. Sin embargo, puede instalar XBMC con soporte CEC desde mi repositorio en [image http://oph.mdrjr.net/meveric/](http://oph.mdrjr.net/meveric/)

*Al salir de un emulador, la ventana XBMC es muy pequeña ¿cómo puedo solucionar esto?*

Si sales del XBMC y lo reinicias de nuevo, volverás a la pantalla completa. Te aconsejo "maximizar" la ventana XBMC, en cualquier caso te resultará más fácil seleccionar las opciones.

*¿Hay algún entrono además del XBMC en la imagen?*

Sí, tras XBMC hay una distribución completa de Debian con LXDE. Esto significa que puedes instalar todo lo que quieras desde el repositorio de Debian. Además, puedes hacer todo lo que se puede hacer en las imágenes de Ubuntu, como la navegación web, edición de documentos y diseño gráfico.

*¿Cómo es el rendimiento en 3D y XBMC?*

ODROID GameStation Turbo utiliza los drivers más recientes (Mali r4p0) que son facilitados por Hardkernel, junto con los nuevos drivers de arm-soc framebuffer y parches de Xorg que dan muy buen resultado. El programa de pruebas de rendimiento es2gears se ejecuta a más de 250fps y glmark2-ES2 se ejecuta a más de 90fps. XBMC se funciona a gran velocidad con 60fps y soporta la reproducción a 1080p de películas H.264. Incluso permite vsync en la reproducción de películas y cambia la frecuencia de tu televisor para que coincida con la tasa de fotogramas de las películas. Al hacer esto se utiliza muy poca CPU y memoria RAM superando la imagen original de Lubuntu 13.10 que publiqué con anterioridad.

Si tienes más preguntas sobre GameStation Turbo, o quieres realizar cualquier consulta, visita los foros ODROID en <http://forum.odroid.com/> También puedes encontrar un montón de juegos pre-compilados en mi repositorio personal en <http://oph.mdrjr.net/meveric>

# INFORMATICA PORTATIL SOLAR CON UN ODROID-U3

## MONTA TU SISTEMA DE 8W USANDO ENERGÍA SOLAR

por Rob Roy

**S**e me ha ocurrido un sencillo proyecto para ejecutar un equipo únicamente con luz solar, siendo la forma más barata y ecológica de usar un ordenador. Los requisitos de potencia pueden estar entre 50-150W, incluso para la mayoría de portátiles de alta eficiencia energética y modelos de escritorio como un PC estándar. Sin embargo, el sistema de energía de 5-8W del ODROID U3 lo convierte en un candidato idóneo para la denominada *Informática Solar*. De este modo puedo configurar un servidor web Apache inalámbrico para mi negocio de desarrollo web, que no me cueste nada ejecutar, se pueda cargar en cualquier lugar con luz solar y disponga de una duración estimada de 6 horas sin necesidad de carga.

Ya tenía el U3 y compré los componentes solares en Amazon por alrededor de 240\$. El modelo de 3 paneles tiene suficientes células solares como para recoger la energía necesaria para que el ODROID funcione durante aproximadamente 6 horas diarias, con 10 horas de recogida de energía solar. Si sólo vas a usar el ordenador durante una hora al día, puede comprar un kit de paneles solares más pequeño, con un coste total estimado de 100\$.

Use varias ventosas para fijar el panel solar portátil GoalZero (http://amzn.to/1z0RUIV) a la entrada de carga de la batería portátil EasyAcc (http://amzn.to/1sQ17mM) usando un cable microUSB a USB. Cada vez que sale el sol, se enciende el indicador de carga del sistema portátil a una intensidad correspondiente a la cantidad de luz recibida. El botón de encendido/apagado del sistema portátil también sirve como un interruptor de encendido remoto para el propio ODROID.

Para ampliar el sistema, puedes conectar en cadena múltiples baterías y conectar la última batería a un panel solar más grande. A la primera batería se enchufa ODROID. Se necesitarían alrededor de 4 de estas baterías para suministrar suficiente energía para un sistema de 24/7.

Si vives en una zona con mucho sol, tiene sentido utilizar la energía solar puesto que la tienes disponible de una forma continuada. Además, ¡En el caso de un ataque gigante de robots, puede llevarte tu campamento de ODROIDs y configurar tu servidor web en el desierto!



Trata de conseguir una huella de carbono cero como esta con una plataforma X86 y vas a tener que comprar una enorme batería!



# DESARROLLAR APLICACIONES ANDROID NATIVAS CON RED

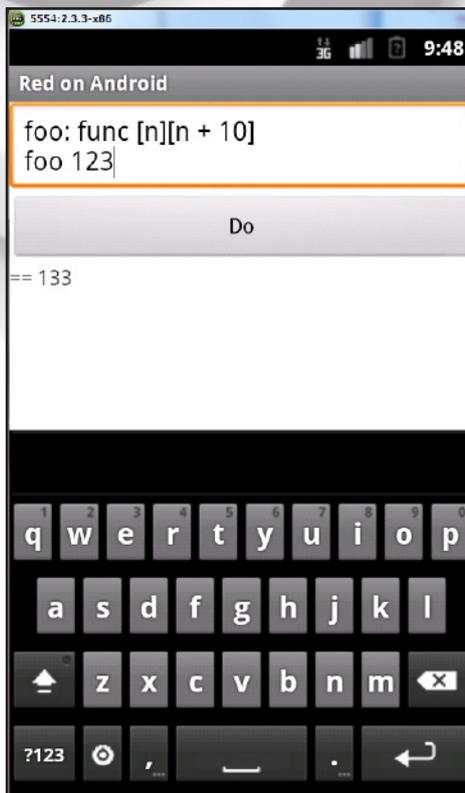
## PARTE 2 - COMPILAR UNA APP ANDROID

por Gregory Pecheret

Para desarrollar una aplicación para Android, puede descargar el SDK de Android de 480MB comprimido desde android.com, que requiere más de 1 GB de espacio en el disco duro. Por otro lado, pronto podrás descargar en su lugar, un SDK de alrededor de 1 MB para el lenguaje RED. Incluso si el espacio en disco necesario, no es importante para ti, la diferencia en el número de líneas necesarias para codificar una aplicación es realmente significativa, ya que afecta directamente al tiempo de desarrollo, depuración y mantenimiento de un proyecto. Todas las comparaciones tienen sus límites, pero si únicamente consideramos el tamaño del código fuente, el código fuente del servidor Apache es de 10 MB, mientras que el código fuente de Cheyenne (Apache alternativo de calidad comercial escrito en Rebol) ocupa menos de 1 MB.

Si ya estás convencido del beneficio que tiene usar un lenguaje de programación de alto nivel semántico como REBOL o Red, entonces puede que estés dudando entre escoger REBOL o Red para desarrollar una aplicación para Android. Ya sabemos que REBOL interpreta el lenguaje mientras que Red lo compila. En cuanto a la interfaz de usuario, REBOL tiene un dialecto para desarrollar widgets partiendo de cero, mientras que Red reutilizará widgets nativos ya existentes.

La mejor elección dependerá del proyecto. ¿La aplicación sólo usa widgets estándar o permite visualizar grá-



ficos personalizados? ¿Será rápida o pretende ahorrar energía? El desarrollo de una interfaz de usuario con Red está próximo a cómo se hace con el Visual Interface Dialect (VID) de REBOL. Debido a que la implementación del dialecto en Red para Android está todavía en curso, sólo podemos ver (por ahora) una prueba de concepto en el que un ejemplo de widget no está terminado.

Para comenzar, descarga el archivo zip desde <https://github.com/red/red>.

Inicia una consola REBOL y ejecuta `build.r`, ubicado en `bridges/android/eval/build.r`:

```
== %/home/odroid/red-master/  
bridges/android/
```



### Consola Red de Línea de Comando en Android

```
>> do %build.r  
Choose CPU target (ENTER = de-  
fault):  
1) ARM (default)  
2) x86  
3) both  
=> 1  
  
==== Red Compiler 0.4.1 ====  
  
Compiling /home/odroid/red-mas-  
ter/bridges/android/samples/eval/  
eval.red ...  
...compilation time:      637 ms  
  
Compiling to native code...  
. .  
Verification successful  
...all done!
```

Una vez que los pasos se han completado con éxito, tu primera aplicación para Android, escrita y compilada en Red, está ubicada en:

`home/odroid/red-master/bridges/android/builds/eval.apk`

El APK puede ser instalado en cualquier sistema Android y se ejecutara como una aplicación nativa de Android.

# DOBLE ARRANQUE CON ODROID-X/X2

## ELIGE ARRANCAR DESDE ANDROID O LINUX PULSANDO UN SIMPLE BOTÓN

por Rob Roy y Venkat Bommakanti

Los dispositivos ODROID X y X2 permiten un doble arranque desde la tarjeta SD o el módulo eMMC a través de un conector en la placa base. Mientras que el conector es bastante accesible en la placa descubierta, no se puede alcanzar tan fácilmente cuando se monta la placa en una caja. Este artículo describe como crear un cable especial que te permitirá seleccionar el arranque deseado, incluso si el X/X2 está dentro de una caja.

### Requisitos

**1. Placa ODROID X/X2, con el adaptador de corriente apropiado.**

**2. Tarjeta MicroSD (con lector de tarjetas SD) que contenga cualquier imagen de Android o Linux**

**3. Cable conector a placa base:**

<http://www.amazon.com/gp/product/B009CWY8PA>

**4. Mini interruptor de 3 vías:**

<http://www.amazon.com/gp/product/B009752DE0>

**5. Tubo termo retráctil:**

<http://www.amazon.com/gp/product/B005W42SW2>

**6. Pelacables (o utiliza la muesca pequeña de una cuchilla de corte)**

**7. Fuente de calor sin llama portátil como un secador de pelo**

**8. Soldador y estaño (opcional)**

**9. Una caja de protección como la siguiente (opcional)**

<http://bit.ly/1jIkp3Z>

**10. Pegamento Superglue (opcional)**

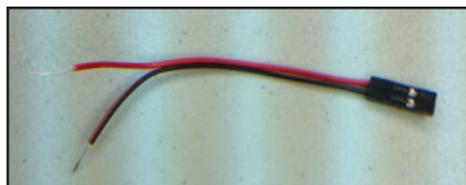
### Preparar el cable conector

Coge el cable conector a la placa base, como se muestra en la primera foto, mide alrededor de 5-6" desde final con el conector y corta el cable en ese punto.



Cable conector a la placa base

Luego, tira alrededor de 1/2" de los extremos de los dos cables, dejando al descubierto el núcleo metálico.



Cable pelado

### Crear el cable a medida

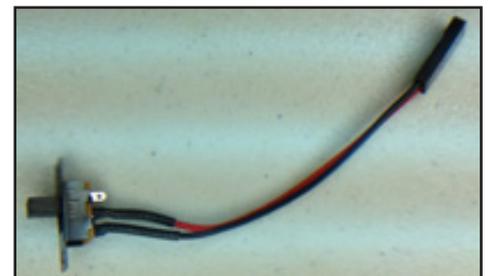
Selecciona un pedazo pequeño de tubo termo retráctil de aproximadamente 1" de longitud que se desliza cómodamente sobre el cable pelado. Inserte un tubo en cada alambre del cable

conector. Conecte los 2 cables a los 2 terminales continuos en el interruptor haciendo un bucle alrededor del interruptor. Puedes soldar con cuidado las conexiones si lo deseas.

A continuación, mueve el tubo para cubrir completamente el terminal y el cable descubierto. Usando un secador a su máxima potencia, desplázalo por encima del plástico de un lado a otro con el aire caliente y mantenlo durante 1-2 segundos. El tubo de goma se encogerá y quedara encajado alrededor de la conexión.



Piezas para conectar el cable



Cable especial montado

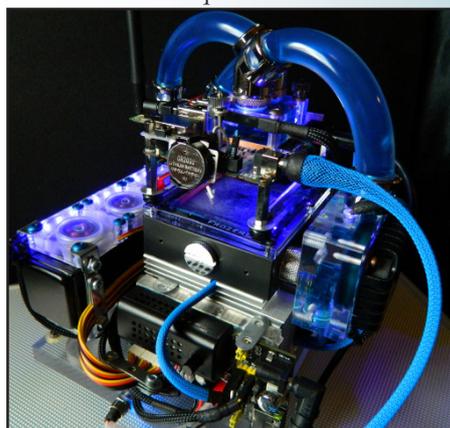
### Conectar el nuevo cable al X/X2

Fija el cable recién creado para establecer la conexión apropiada en el dis-

# REFRIGERACION LIQUIDA PARA ODRROID-XU

por Rob Roy

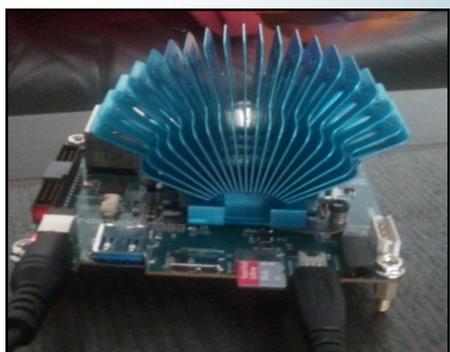
**O**DRROID-XU viene con un silencioso ventilador de refrigeración, pero varios usuarios de ODRROID han añadido sistemas de refrigeración completamente nuevos. Estas son algunas fotos de los asombrosos sistemas que han utilizado.



El H2ODROID, refrigeración por agua y un disipador de calor extremadamente complejo

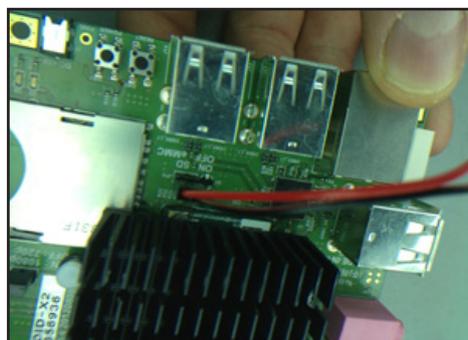


Otra vista del H2ODROID

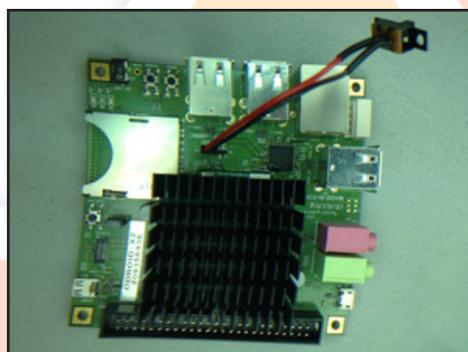


Refrigeración pasiva usando un disipador de calor Zalman ZM-NBF47.

positivo, como se muestra en las figuras 5 y 6. En este ejemplo, el conector de SD / eMMC está siendo conmutado, pero este también se podría conectar al conector 720/1080P igualmente.

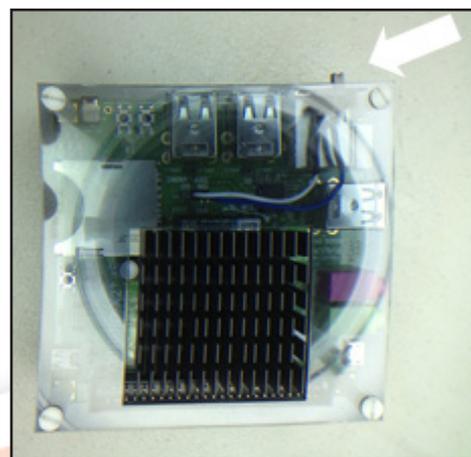


Ubicación del conector en la placa



Cable conector fijado

Los últimos dos fotos indican una posible ubicación para el interruptor cuando utilizamos una caja. Si no dispones de una caja, puede utilizar una pequeña gota de pegamento Superglue para fijar el interruptor en la parte superior del puerto ethernet, donde encaja muy bien. ¡No tendrás que preocuparte más si pierdes la tapa! Al mover el interruptor puedes elegir si arrancas con el eMMC o la tarjeta SD.

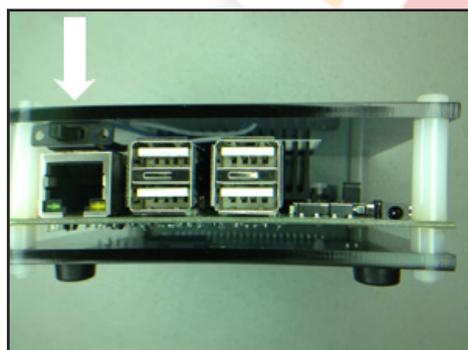


Vista superior de la ubicación del interruptor

Para completar el proyecto, puede marcar una "U" (para Ubuntu) y una "A" (para Android) en la caja, en base a la posición del interruptor apropiada, para la fácil elección del sistema operativo. Si vas a utilizar un diseño de caja diferente, la posición del interruptor puede variar con respecto a la que se muestra aquí. Para más información o plantear tus cuestiones, puedes visitar: <http://bit.ly/1iyfB69>



Ejemplos de cajas a medida para el U2 y X2



Vista lateral de la ubicación del interruptor en una caja a medida



# ESCRITORIO REMOTO PARA ODROID U3

## USAR LOS PROTOCOLOS VNC Y XRDP

por Stangri, Editado por Venkat Bommakanti



**P**or diversas razones, puede ser necesario acceder al escritorio de un U3 de forma remota desde otro equipo.

Un método muy común para este acceso es utilizar el protocolo VNC (Virtual Network Computing) que permite el control remoto de cualquier escritorio de un ordenador. En este artículo se describen los métodos de acceso basados en VNC para conectarse a un U3 desde varios clientes, tales como Windows 7, OSX y Linux.

Estos pasos son válidos para el servidor x11vnc en instalaciones de Ubuntu 13.10 o 14.04, y no se aplican a otros servidores VNC sobre el U3, como TightVNCserver.

```
$ sudo aptitude install x11vnc
```

Es muy posible que x11vnc no esté empaquetado en la imagen por defecto. Valida la instalación comprobando la versión:

```
$ x11vnc --version
x11vnc: 0.9.13 lastmod: 2011-08-10
```

```
Set x11vnc password
```

Para asegurar el acceso VNC al U3, configura una contraseña usando el siguiente comando:

```
$ sudo x11vnc -storepasswd {your_password_here} /etc/x11vnc.pass
```

Usa una contraseña segura sin incluir los corchetes {}. A continuación, actualiza los privilegios de acceso para este archivo de contraseñas recién configurado, con el comando:

```
$ sudo chmod 744 /etc/x11vnc.pass
```

Sin este cambio de privilegios, el comportamiento de una sesión VNC es impredecible. Pueden aparecer problemas como una pantalla en blanco/gris en el visor VNC, o que el servidor VNC falle con una conexión entrante, u otros problemas.

## Configurar x11vnc

Antes de establecer la configuración, es recomendable determinar las resoluciones de pantalla soportadas por el U3. Comprueba las resoluciones disponibles con el comando:

```
$ xrandr
Screen 0: minimum 320 x 200, current 1920 x 1080,
maximum 4096 x 4096
HDMI-1 connected 1920x1080+0+0 1440mm x 810mm
1920x1080 60.0*+ 30.0
1920x1080i 60.1
1280x1024 60.0
1280x960 60.0
```

## Requisitos

**1. Una placa ODROID U3 con un adaptador de corriente adecuado**

**2. Cualquier tarjeta MicroSD o módulo eMMC con Ubuntu, Debian o imagen similar que incluya un escritorio de ventanas (X11).**

**3. Una red en la que el dispositivo tenga acceso a Internet y a los foros ODROID.**

**4. Acceso ssh para el U3 a través de utilidades como PuTTY (MS Windows 7) o terminal (Mac, Linux) desde el escritorio remoto.**

**5. TightVNC viewer 2.7.1 para Windows 7**

**6. Remmina 0.0.99.1 para Ubuntu o Debian**

**7. Microsoft Remote Desktop Connection Windows 7**

## Instalar x11vnc

El primer acceso al U3 vía ssh, usando cualquier herramienta de SSH compatible con tu máquina cliente. Desde una sesión de Terminal en Linux, el comando sería el siguiente:

```
$ ssh root@odroid
```

En una imagen típica de Hardkernel, el nombre de host del U3 será odroid. Conectando por SSH a través de PuTTY aparecerá un segundo prompt desde la máquina remota. Usando este prompt, instala x11vnc con los comandos:

```
$ sudo aptitude update
```

|           |      |
|-----------|------|
| 1152x864  | 75.0 |
| 1280x720  | 60.0 |
| 1024x768  | 60.0 |
| 1440x480i | 60.1 |
| 720x480   | 59.9 |
| 640x480   | 60.0 |

Es posible que quieras seleccionar una resolución diferente en función del monitor conectado al escritorio remoto. La resolución 1920x1080 ha sido la elegida para este artículo.

La instalación de x11vnc ahora puede ser configurada usando un archivo de configuración, que se encuentra en:

`/etc/init/x11vnc.conf`

Crear el archivo si no está, usando un editor como vi, medit o nano. Para el típico acceso VNC, el contenido del archivo de configuración debe ser similar a este:

```
start on login-session-start
stop on runlevel [016]

emits vnc-server-start

script

/usr/bin/x11vnc -forever -bg -geometry 1920x1080
-usepw -auth /var/run/lightdm/root/:0 -display :0
-rfbauth /etc/x11vnc.pass -shared -nopri-
mary -o /var/
log/x11vnc.log

initctl emit vnc-server-start

end script
```

Usa el editor (con privilegios de root) para actualizar el archivo de configuración. Ten en cuenta que la línea de configuración x11vnc (que empieza por /usr/bin/x11vnc) tiene que ser una única línea.

Si tu escritorio remoto resulta que ejecuta OSX o Linux (por ejemplo, Ubuntu 14.04 LTS), puedes acceder a éste activando los servicios de avahi en el U3. Esto facilita la detección de escritorios avahi-aware del servicio VNC ofrecido por el U3 través de la red local. El archivo x11vnc.conf en este caso sería el siguiente:

```
start on login-session-start
stop on runlevel [016]

emits vnc-server-start
```

```
script

/usr/bin/x11vnc -forever -bg -geometry 1920x1080
-usepw -auth /var/run/lightdm/root/:0 -display :0
-rfbauth /etc/x11vnc.pass -shared -gui tray -nopri-
mary -noxdamage -zeroconf -avahi -env X11VNC_AVAHI_
NAME=odroid -o /var/log/x11vnc.log

initctl emit vnc-server-start

end script
```

La opción X11VNC\_AVAHI\_NAME debe reflejar el nombre de host de tu U3, en el caso de que se haya modificado el valor por defecto de odroid. Cualquier aplicación de pantalla compartida, como la aplicación integrada de pantalla compartida en OSX o la aplicación cliente Remmina VNC para Ubuntu 14.04 se puede utilizar para ver el escritorio del U3.

Algunos usuarios pueden tener problemas de inicio con la configuración de interfaz gráfica de usuario (GUI). Si es así, elimina la siguiente opción en el archivo de configuración:

`-gui tray`

Además, asegúrate de que no haya otras instancias de servidor VNC usando Display #0.

## Ejecutar x11vnc

Para iniciar el servidor x11vnc en el arranque, escriba el siguiente comando en una única línea en una ventana de terminal y a continuación, reinicie el U3:

```
$ sudo /usr/bin/x11vnc -geometry 1920x1080 -usepw
-auth /var/run/lightdm/root/:0 -display :0 -forever
-bg -rfbauth /etc/x11vnc.pass -o /var/log/x11vnc.log
```

Después de reiniciar, comprueba que el servidor x11vnc está funcionando analizando el archivo log del x11vnc en:

`/var/log/x11vnc.log`

Debe aparecer la información de vnc como la siguiente:

```
18/06/2014 14:11:27 Avahi group odroid established.
The VNC desktop is:      odroid:0
```

Ten en cuenta que en este caso, tendremos que añadir `display#0` en odroid (es decir, `display:0`).

## Instalar Servicio XRDP

Si desea acceder al escritorio del U3 desde un sistema re-

moto con Windows 7, usando la aplicación Microsoft Remote Desktop Connection, tendrás que instalar también el servicio xrdp en U3. Instálalo utilizando el siguiente comando:

```
$ sudo apt-get install xrdp --no-install-recommends
```

Actualiza el archivo de configuración xrdp recientemente creado para reflejar las siguientes opciones:

```
/etc/xrdp/xrdp.ini

[globals]
bitmap_cache=yes
bitmap_compression=yes
port=3389
crypt_level=high
channel_code=1
max_bpp=24

[xrdp1]
name=local-xvnc-console
lib=libvnc.so
username=odroid
password=ask
ip=127.0.0.1
port=5900
```

El parámetro de nombre de usuario tiene que reflejar el usuario odroid por defecto creado en una instalación limpia. Si va a utilizar otro nombre de usuario, asegúrate que tenga las mismas asociaciones de grupos que la cuenta odroid. El siguiente comando facilita la información del grupo:

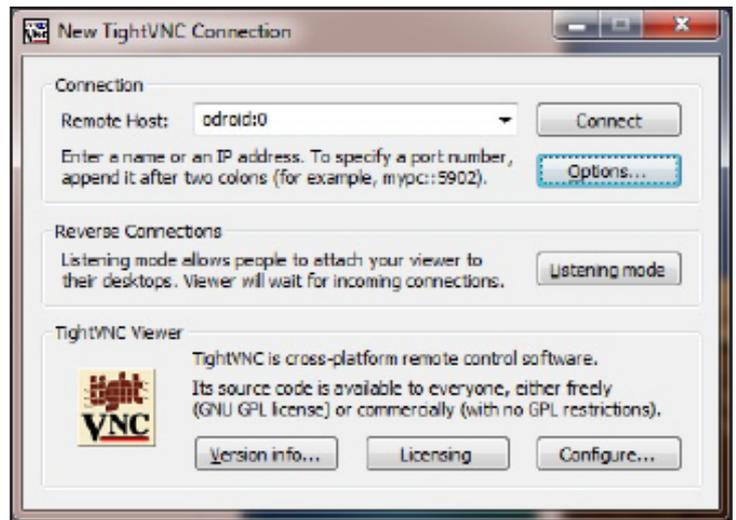
```
$ groups odroid
odroid : odroid adm dialout fax cdrom floppy tape
sudo audio dip video plugdev netdev nopasswdlogin
lpadmin scanner fuse
```

Para mejorar la seguridad, es útil forzar la visualización del cuadro de diálogo de inicio de sesión en el cliente. La posibilidad de solicitar la contraseña habilita esta opción. La contraseña real no tendrá que se usada en el archivo de configuración, aumentado así la seguridad y permitiendo cambiar la contraseña a tu antojo, sin tener que alterar el archivo config para nada.

El servicio xrdp es un servicio adicional del servidor, que requiere del servidor VNC totalmente funcional en el U3.

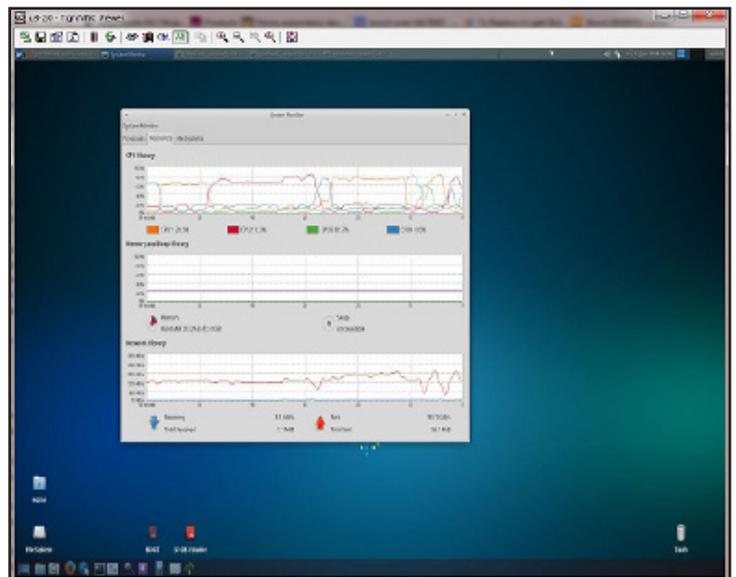
### TightVNC Viewer en Windows

Inicia TightVNC Viewer e introduce la información del servidor VNC del U3, como se muestra en la imagen TightVNC Viewer Login.



TightVNC viewer login

Luego haga clic en el botón “Connect” para conectar y ver el escritorio de U3 de forma remota. El escritorio del U3 que se muestra es similar al que aparece en la siguiente imagen:

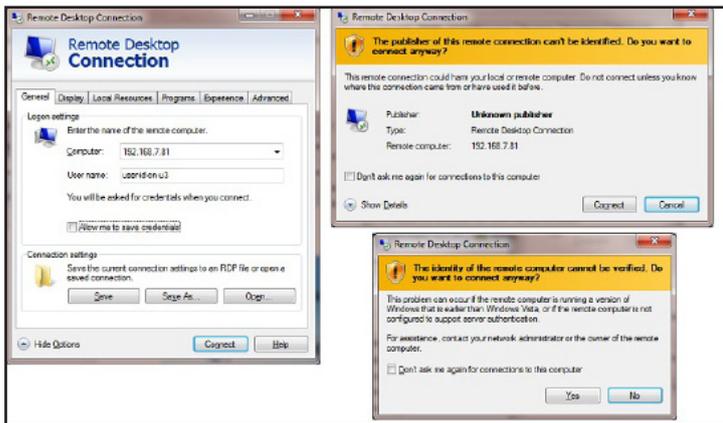


Escritorio U3 via TightVNC viewer

### Remote Desktop Connection

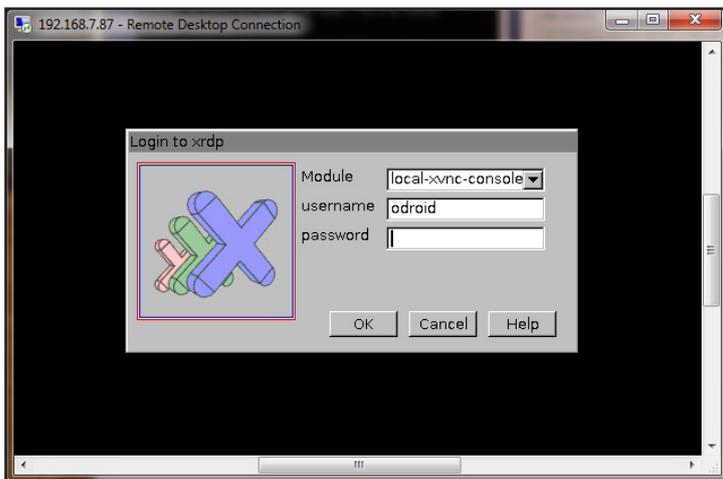
Inicie la aplicación Remote Desktop Connection. Introduzca la información del servidor VNC del U3 y rellene el formulario como se muestra en la imagen Pantalla Inicio de Sesión del Remote Desktop Connection. En el cuadro de diálogo de inicio de sesión preliminar, haga clic en “Opciones” para ver más opciones. Introduzca todos los detalles en la pestaña “General”, selecciona la pestaña “Mostrar” y seleccionar la opción de pantalla completa. Elija la opción del color verdadero de 24 bits, si fuera necesario.

Si la conexión tiene éxito, aparecerá la pantalla de inicio de sesión xrdp de la figura 4. Introduzca la contraseña del servidor vnc y haz clic en “OK” para continuar. Ten en cuenta que el módulo local-xvnc-console usado en el



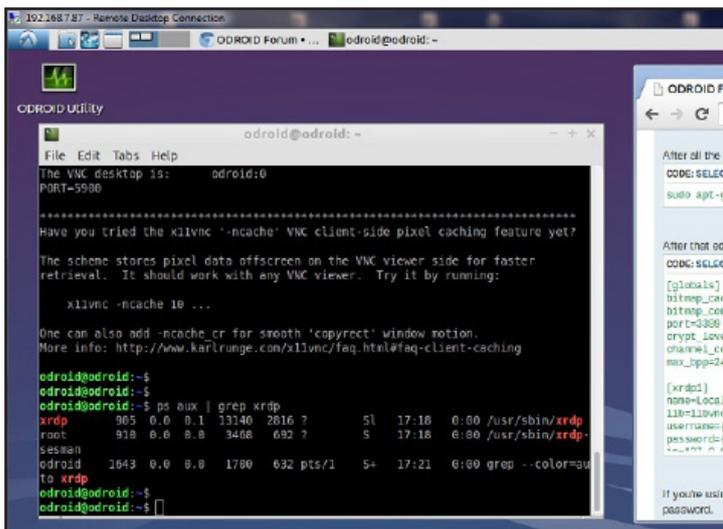
Pantalla inicio del Remote Desktop Connection

el cuadro de diálogo de inicio de sesión ha de coincidir con la entrada de configuración del xrdp.ini mencionada anteriormente.



Pantalla de login xrdp para Remote Desktop Connection

Un inicio de sesión correcto da como resultado la pantalla del escritorio de U3, similar a la que se muestra en el imagen escritorio U3 via Remote Desktop Connection.



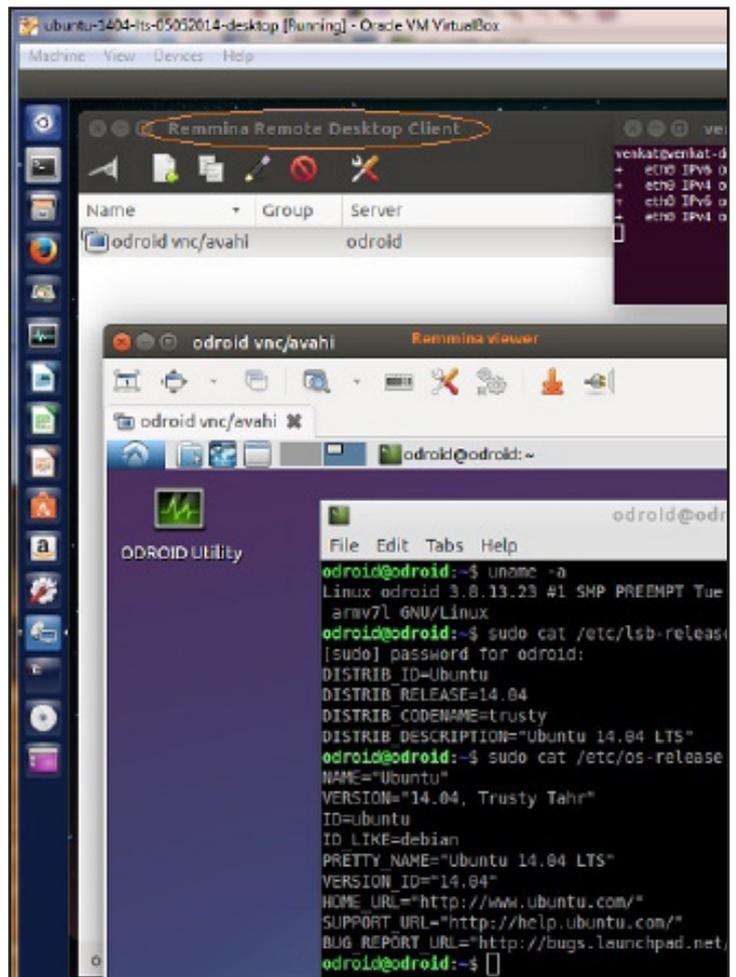
Escritorio U3 via Remote Desktop Connection

## Remmina para Ubuntu 14.04 LTS

Instala Remmina desde Centro de Software de Ubuntu. Inicie el cliente de Escritorio remoto Remmina. Seleccione la entrada para el U3 y conectarte.

También puede buscar los servicios de avahi-aware disponibles ejecutando el comando:

```
$ avahi-browse -all
```



Escritorio U3 via Remmina

Para obtener más información o realizar preguntas sobre como conectarse a un ODROID usando VNC o XRDP visite los siguientes recursos:

- <http://bit.ly/1qS6AWc>
- <http://www.tightvnc.com/>
- <http://avahi.org/>
- <http://remmina.sourceforge.net/>
- <http://bit.ly/1z93Bxd>
- <http://bit.ly/1mVyEF2>

# VEHICULO TODOTERRENO AUTOMATICO IMPULSADO POR ODROID

## PARTE 2: CONEXION CON MOTORES Y SENSORES

por Christopher D. McMurrough

Continuamos nuestra serie sobre cómo construir un vehículo todoterreno automático (VTA) usando el ODROID-XU. En esta segunda entrega analizaremos la comunicación entre ODROID y los sensores y controladores del motor de la plataforma, que usaremos más adelante en la Parte 3 para que se desplace de forma autónoma usando puntos GPS. La aplicación del controlador se ejecuta sobre la imagen de Ubuntu 12.04 Robótica Edition para ODROID-XU, que se puede descargar desde los foros Hardkernel.

### Introducción

En la Parte 1, nos centramos principalmente en los aspectos mecánicos y eléctricos de nuestro robot. Ahora que disponemos de una plataforma estable para trabajar, vamos a ver como configurar ODROID-XU para comunicarnos con los sensores y controladores de motor. Los controladores de motor que hemos elegido para este proyecto son los Parallax HB-25, que estarán gestionados por un microcontrolador USB Teensy [1]. Teensy acepta paquetes de comando en serie desde el ODROID-XU, que luego se utilizarán para generar los impulsos servo-control necesarios para fijar las velocidades del motor y la dirección.

Además, nos comunicaremos con un dispositivo Android (una tablet Nexus 7) para obtener datos de rumbo y GPS. Esto nos permitirá controlar las velo-

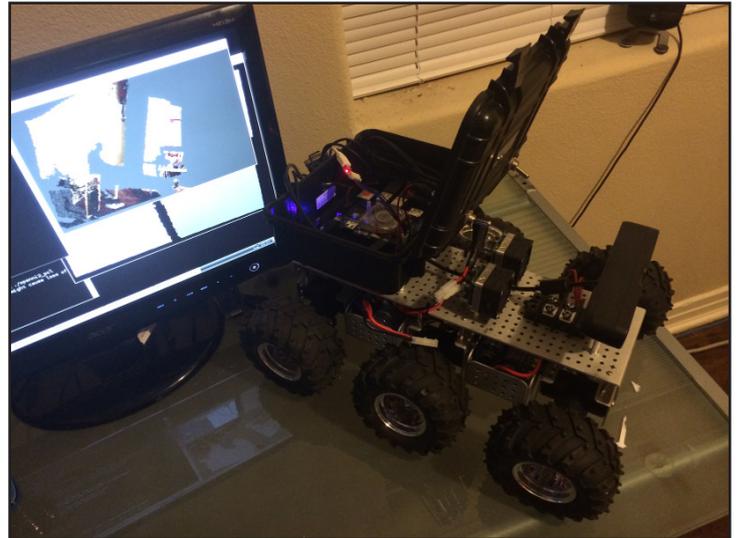
idades del motor para abordar el sistema de movimiento en la parte 3. El dispositivo Android se comunica vía USB tethering, lo que nos permitirá hablar con el ODROID como un si estuviéramos en lamisma red.

Por último, vamos a probar Robotic Operating System(ROS) para enviar y recibir datos. Este sistema modular, nos hará más fácil la posibilidad de ampliar las capacidades de nuestra plataforma en el futuro. El código de ejemplo se puede descargar en <http://bit.ly/1jfykOU>.

### Configurar el Sistema Operativo

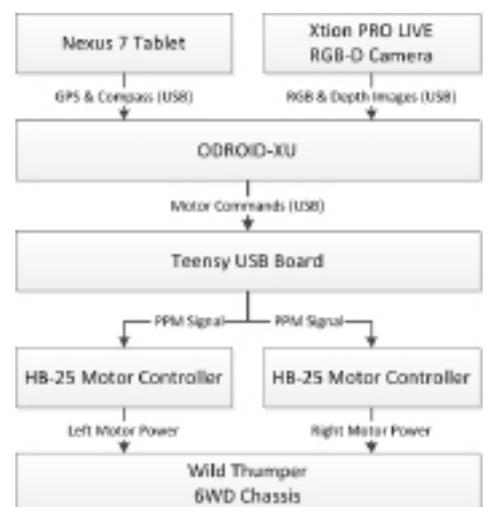
Primero vamos a configurar nuestro ODROID-XU para ejecutar Ubuntu 12.04 Robótica Edition [2]. Esta imagen del sistema operativo ha sido precargada con Robot Operating System(ROS) y otro software de robótica como OpenCV, Point Cloud Library y OpenNI. Descarga la imagen de los foros ODROID y cópiala a tu tarjeta SD o módulo eMMC. Aunque nosotros usamos específicamente el ODROID-XU, las imágenes de robótica están disponibles de igual modo para X2 y U2/U3

Una vez que la imagen con el sistema

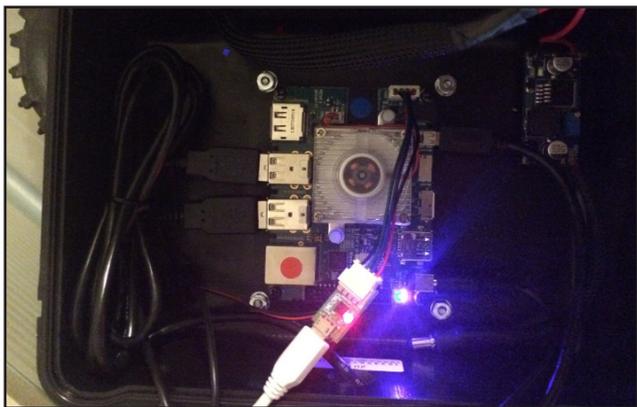


Este robot se ejecuta en un ODROID-XU, está hecho de piezas fácil de conseguir, y adora los terrenos difíciles.

operativo ha sido cargada en ODROID, arranca el dispositivo con un teclado, ratón y monitor conectados. Serás llevado directamente al entorno de escritorio LXDE. Desde allí abre una ventana de comandos y escriba “roscore”. Pulsa intro y ¡Gualal!, ahora estás ejecutando ROS, que usaremos más adelante.



Arquitectura del Sistema VTA



**Este VTA es lo bastante listo como para tener un XU como cerebro, y sólo usa unos cuantos vatios.**

Como ya se ha mencionado, el microcontrolador Teensy se usa para conectar ODROID con los controladores de motor HB-25. El HB-25 espera señales servo PPM, que cualquier dispositivo compatible con Arduino (como nuestro Teensy) puede generar con facilidad. Vamos a conectar los pines de señal de HB-25 a los pines 20 y 21 de Teensy, y los configuraremos como pines de servo control. Con la librería Arduino Servo, podemos controlar nuestros motores enviando un número entero con el comando “write”.

El valor enviado a este comando determina un ángulo servo de entre 0 y 180 grados, siendo 90 grados el punto neutral. En el caso del HB-25, el valor 0 no aplicará energía al motor, mientras que un valor de 0 a 180 aplicará potencia al motor en cualquier dirección. Como los valores se desvían desde 90 hasta 0 o 180, el motor obtendrá más potencia y empezará a girar más rápido.

Al escribir un número entero en cualquier pin de salida usando la librería Servo, podemos controlar independientemente la velocidad y dirección de los motores. La siguiente tarea será la de establecer comunicación con ODROID usando paquetes de datos en serie. El ODROID transmitirá un único paquete de datos para agrupar los dos bytes de control. Estos paquetes se envían continuamente a una velocidad de 20 Hz, de modo que ODROID puede sin problemas controlar el movimiento de la

plataforma. Cuando este cargado, el firmware Teensy permitirá que el dispositivo se pueda enumerar como un puerto serie virtual. Luego escribiremos nuestros paquetes de control en este puerto usando nuestro nodo demo ROS.

## Estructura Paquetes de Comandos

Los puertos serie facilitan la transmisión de datos en serie, que son simplemente cadenas de bytes. Para aplicaciones muy simples, como el envío de forma continua de un único valor de byte, una aplicación puede simplemente leer o escribir solamente el byte más reciente e ignorar cualquier alineación o validación de datos. La mayoría de las aplicaciones de robótica requieren de la transmisión de varios bytes de datos (por ejemplo, escribiendo comandos a múltiples servos), por lo que se debe tener cuidado y asegurarse que el receptor conoce el significado de cada byte.

En nuestro ejemplo, vamos a definir un paquete con un byte inicial, un byte de longitud, la carga y la suma de control. El byte de inicio indicará al receptor que se está transmitiendo un paquete. La longitud especificará la longitud total del paquete, incluyendo los datos de estructura en número de bytes (para soportar paquetes de tamaño variable). La carga se compondrá de 2 bytes que especifican los comandos de motor, y la suma de control permitirá a Teensy comprobar si hay datos dañados o mal alineados. La estructura de paquete resultante será:

```

Packet [0]
Packet [1]
Packet [2]
Packet [3]
Packet [4]
Start byte
Packet length
Payload byte 1
Payload byte 2
Checksum
    
```

El byte de inicio se define arbitrariamente como el byte 0xAA. La longitud del paquete, para este paquete, se define como 0x05 (3 bytes de estructura + 2 bytes de carga). La suma de control se calcula individualmente para cada paquete basándose en el resto de bytes. Definimos la suma de control para que sea el valor obtenido por todos los otros bytes, de modo que:

```

Packet [4] = Packet [0] XOR
Packet [1] XOR Packet [2] XOR
Packet [3]
    
```

Cuando Teensy recoge todos los bytes del paquete, se calcula la suma de control y la compara con el paquete [4]. Si estos dos valores no son iguales, puede ser porque los datos estén dañados o porque hay un conflicto con el número de bytes recibidos. De cualquier manera, esta función permite al Teensy detectar y recuperarse de un error (o detener los motores si fuese necesario). Aunque sólo tenemos que transmitir 2 bytes de carga a Teensy para este proyecto en particular, el método puede ser usado para enviar cuantos bytes se quiera siempre ajustando la longitud del paquete. En tu propia aplicación, tendrás que ajustar la longitud del paquete para que coincida con la longitud de los datos que van al vehículo con el fin de controlar los motores y sensores que estés utilizando.

## Firmware Microcontroller

El firmware del microcontrolador Arduino está en la carpeta “ugv” con el código de ejemplo. Para usar el firmware, simplemente compila y actualizarel archivo “teensy\_firmware.ino” utilizando el Arduino IDE, siguiendo las instrucciones que se encuentran en el sitio web Teensy [1]. Una vez que el firmware haya sido instalado, estarás listo para ejecutar la demo. Ten en cuenta que las reglas udev apropiadas deben estar instaladas en ODROID para que la aplicación sea capaz de abrir la conexión serie hacia Teensy sin usar el comando “sudo”.

## Interfaz Motor Controller ROS

En nuestro código de ejemplo hay un nodo ROS que conecta al puerto serie especificado y agrupa paquetes en el formato adecuado. Este nodo se suscribe a los mensajes "Twis" ROS, que es el método estándar para especificar la velocidad deseada de un robot. En nuestro caso, sólo nos preocupamos por la velocidad lineal de avance y la rotación. La rotación nos permite calcular la cantidad de energía necesaria para abastecer a los motores izquierdo y derecho, y la velocidad lineal define la velocidad en general. En la Parte 3 entraremos en más detalles sobre cómo calcular las velocidades del motor, pero por ahora, sólo queremos ejecutar el nodo ROS y verificar que el puerto serie USB se abre correctamente. Puede ser necesario cambiar el nombre del puerto /dev/ttyUSB/ para que coincida con tu configuración.

## Datos de Navegación

Nuestro sistema usa datos de navegación desde una tableta Android para planificar el movimiento. Para ello, primero debemos establecer una conexión entre la tablet y ODROID, y vincular estos datos a ROS. Por suerte, esto es muy fácil si usamos la app de android "ROS Android Sensors Driver" [4]. Tras su instalación en el dispositivo Android, los datos de la brújula, GPS y acelerómetro son enviados a ROS con la dirección IP especificada. Al conectar la tablet a ODROID vía USB y activando tethering, Android y ODROID estarán ubicados en la misma red y pueden comunicarse vía IP. Es importante tener en cuenta que el paquete ROS android\_sensors\_driver también se debe instalar para recibir datos de la aplicación.

Una vez que la tablet este conectada, ejecuta el programa de línea de comandos roscore en ROS, a continuación, iniciar los programas android\_sensors\_driver en la tablet Android. Introduce la dirección IP y el puerto de roscore en la aplicación Android, y pulsa conectar.

Los datos de GPS se muestran como temas ROS, que podemos utilizar para nuestras tareas de navegación.

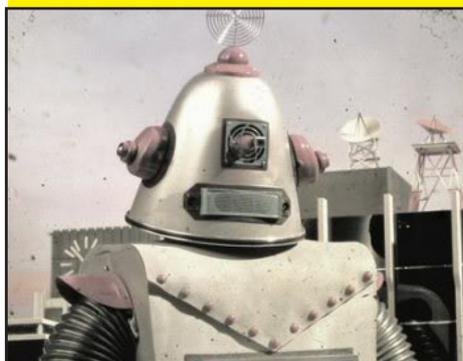
## Conclusion

En este punto, disponemos de todos los componentes de software para acceder a nuestros motores y sensores, y lo hemos vinculado todo a ROS, lo que hará que sea más fácil desarrollar nuestro software de aplicación. En la Parte 3, introduciremos las coordenadas GPS en ROS, calcularemos la trayectoria de movimiento, y proporcionaremos los comandos de control de motores para alcanzar el objetivo. Luego, ampliaremos nuestra base de software añadiendo nodos ROS adicionales para el dominio y control. Finalmente, probaremos el sistema en un escenario al aire libre.

## Referencias

- [1] P. Stroffregen, "Teensy USB Development Board," PJRC Electronics, 2014. <http://www.pjrc.com/teensy/>
- [2] C. McMurrugh, "Ubuntu 12.04 Robotics Edition v1 (ROS+OpenCV+OpenNI+PCL) XU," ODROID Forum, 2014. <http://forum.odroid.com/view-topic.php?f=61&t=5216>
- [3] C. McMurrugh, "Odroid Code Examples and Support Files," GitHub, 2013. <https://github.com/cmc-murrugh/odroid-development>
- [4] C. Rockey, "Ros Android Sensors Driver," [http://wiki.ros.org/android\\_sensors\\_driver/](http://wiki.ros.org/android_sensors_driver/)

**Hemos seguido las instrucciones, pero nuestro VTA se ve muy diferente a la que Chris desarrollo**



## CAMBIA TU LATIDO CONTROLANDO EL LED DE TU U3

por Mauro Ribeiro

**C**ambiar el comportamiento del LED azul en el ODROID-U3 es relativamente fácil con un único comando BASH. Escriba cualquiera de los siguientes en la ventana de Terminal de Linux:

1. Desactivar completamente el led:
 

```
$ echo none > \
/sys/class/leds/led1/trigger
```
2. Dejarlo encendido (sin parpadear):
 

```
$ echo default-on > \
/sys/class/leds/led1/trigger
```
3. Mostrar la actividad de eMMC:
 

```
$ echo mmc1 > \
/sys/class/leds/led1/trigger
```
4. Mostrar la actividad de la SD:
 

```
$ echo mmc0 > \
/sys/class/leds/led1/trigger
```
5. Hacer que parpadee con un patrón personalizado:
 

```
$ echo timer > \
/sys/class/leds/led1/trigger
$ echo 1000 > \
/sys/class/leds/led1/delay_on
$ echo 1000 > \
/sys/class/leds/led1/delay_off
```

El último ejemplo permite especificar cuántos milisegundos debe permanecer la luz en cada estado. También puede utilizar el LED para indicar otra actividad. Por ejemplo, puede realizar un seguimiento de la actividad SSH monitorizando las tablas IP escribiendo lo siguiente en una ventana de terminal de Linux como usuario root:

```
$ iptables -A INPUT -p tcp \
--dport 22 -j LED \
--led-trigger-id ssh \
--led-delay 1000
$ echo netfilter-ssh > \
/sys/class/leds/led1/trigger
```

Para hacer el cambio permanente, agregue la línea en el archivo /etc/rc.local. El nuevo patrón del LED tendrá efecto después del siguiente reinicio.

# PLACA METEOROLOGICA ODROID

## VIGILA EL TIEMPO DE FORMA INTELIGENTE CON TU ODROID

por Justin Lee

La placa meteorológica es un complemento para ODROID-SHOW fácil de usar que te permite acceder a las mediciones sobre radiación UV, presión barométrica, altitud, humedad relativa, iluminación y temperatura. Sus dimensiones son de tan sólo 20 x 20 mm e incorpora un conector de 6 pines para el ODROID-SHOW.

La placa informa de la humedad Si7020, presión barométrica BMP180 y el índice Ultravioleta (UV) Si1132 con sensores de luz ambiental, y se apoya en algunas librerías de Arduino basadas en sensores. Todos los datos de los sensores van por comunicación I2C entre los ICs de los sensores y MCU ATMEGA328P en el ODROID-SHOW.

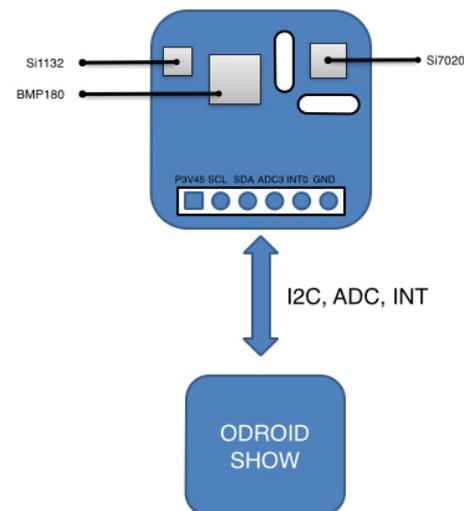
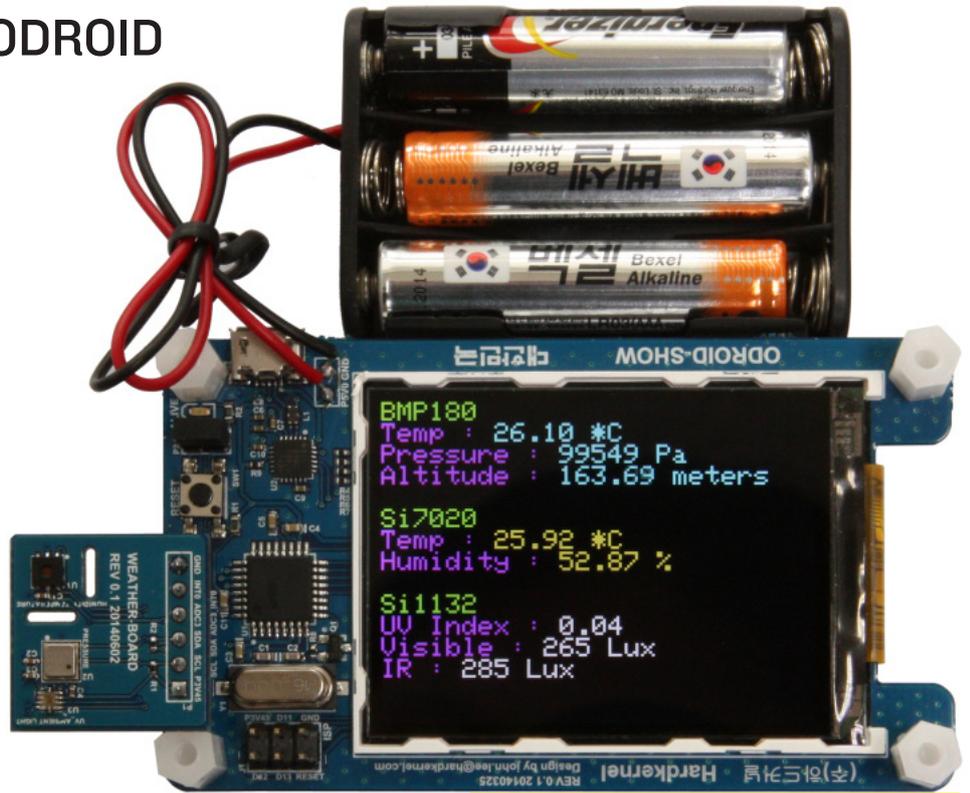


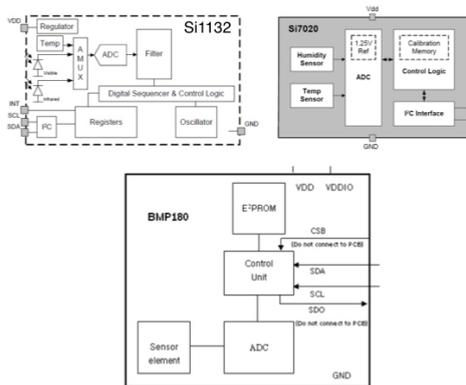
Diagrama por bloques de la placa meteorológica

Con algunas pilas, se puede hacer un sistema de captura de datos meteorológicos exterior portátil. También se puede conectar a tu PC o ODROID a través del ODROID-SHOW con el fin de registrar



La placa meteorológica es su herramienta personal para comprobar si es seguro salir a la calle, o mejor quedarse en casa y trabajar en un proyecto ODROID.

diversos datos ambientales, como lo hace una estación meteorológica.



La arquitectura interna de cada sensor

### Instalar Hardware

Conecta tu placa meteorológica al ODROID-SHOW. Luego, conecte un cable micro-USB a tu PC principal o placa ODROID con Linux. Si la conexión se realiza correctamente, verás que aparece un nuevo dispositivo llamado / dev/ttyUSB0

(o algo similar). También puede conectar 3-4 pilas alcalinas o baterías NiMH / NiCD para crear un dispositivo portátil.

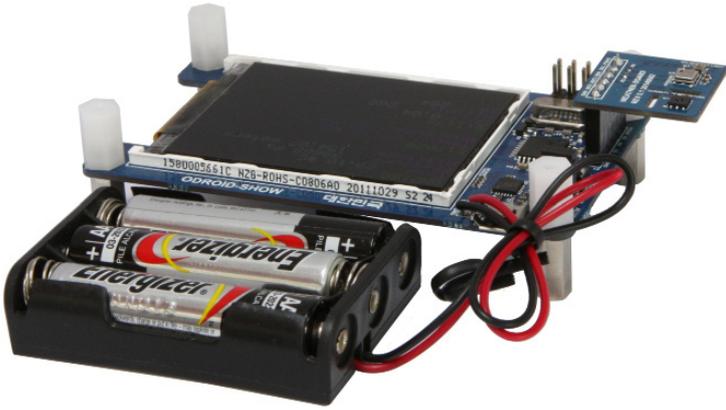
Por último, lanza una copia de IDE Arduino , que está disponible para su instalación desde Gestor de paquetes Synaptic o el Centro de Software de Ubuntu.

### Instalar Firmware

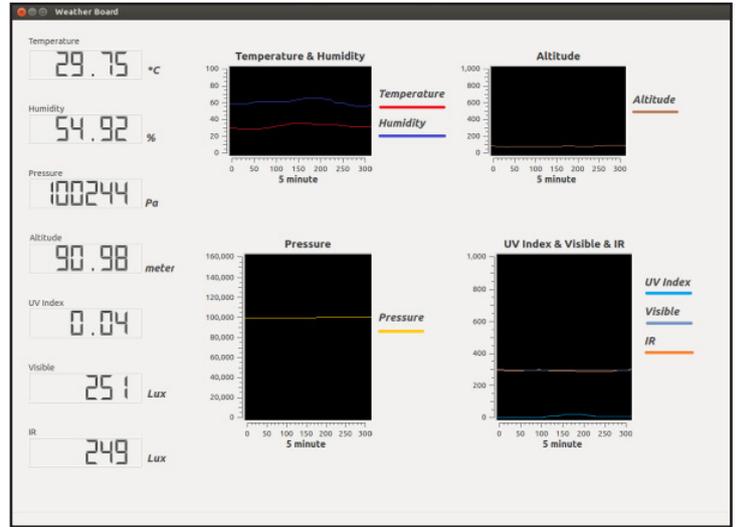
Descarga el fichero de esquema y librerías de <http://bit.ly/1iuz4VK> Después, abra el archivo weather\_board.ino usando Arduino, que se encuentra en:

```
ODROID-SHOW/firmware/weather_board/weather_board.ino
```

Selecciona "Preferences" en el menú Archivo de IDE Arduino y escoge ODROID-SHOW como carpeta de es-



Otra perspectiva de la placa meteorológica sobre un U3.



Una aplicación GUI basada en AQT puede monitorear y mostrar los datos meteorológicos en un entorno gráfico.

quemas. Agrega las librerías seleccionando Import library->Add library y selecciona ODROID-SHOW/libraries. Es hora de desarrollar y cargar el archivo de esquemas. La tapa del conector, cerca del botón de reinicio, debe instalarse en primer lugar.

Un PC o ODROID pueden registrar los datos, así como mostrarlos gráficamente en tiempo real. Para nuestro ejemplo, usando lenguaje de programación QT.

La Placa meteorológica para ODROID se puede comprar en la tienda de Hardkernel: <http://bit.ly/1wtPdgP>.

## Desarrollar una aplicación QT para Windows

1. Descarga e instala la versión Qt 4.8.4(MinGW) en <http://bit.ly/1nAxheV>

2. Añade C:\Qt\4.8.4\bin a la variable path de nuestro sistema (qmake.exe esta localizado aquí)

3. Añade C:\MinGW\bin a la variable path de tu sistema (mingw32-make.exe. esta localizado aquí)

4. Descarga y extrae qwt-6.1.0 at <http://bit.ly/1quAoaY>.

5. Copia el qwt-6.1.0 a C:\qwt-6.1.0

6. Abra una línea de comandos (cmd) y vaya a "C:\qwt-6.1.0":

```
# cd C:\qwt-6.1.0
# qmake
# make
# make install
# qmake -set QMAKEFEATURES C:\qwt-6.1.0\features
```

7. Añade C:\qwt-6.1.0\lib a la variable path de tu sistema (qwt.d.dll esta localizado aquí)

8. Descarga el código fuente de la aplicación PC en <http://bit.ly/1pgo0yf>.

9. Compila el ejecutable weather\_board:

```
# cd qt_weather
# qmake
# make -f MakeFile.Release
# qmake
# make -f MakeFile.Release
```

## Desarrolla una aplicación x86 para Ubuntu

1. Instala paquetes

```
# sudo apt-get install qt4-default qt4-designer libqwt-dev
```

2. Compila weather\_board

```
# git clone https://github.com/hardkernel/ODROID-SHOW/qt_weather
# cd qt_weather
# qmake
# make
# uic weather_board.ui > ui_weather_board.h
# qmake
# make
# cd linux
# ./weather_board
```

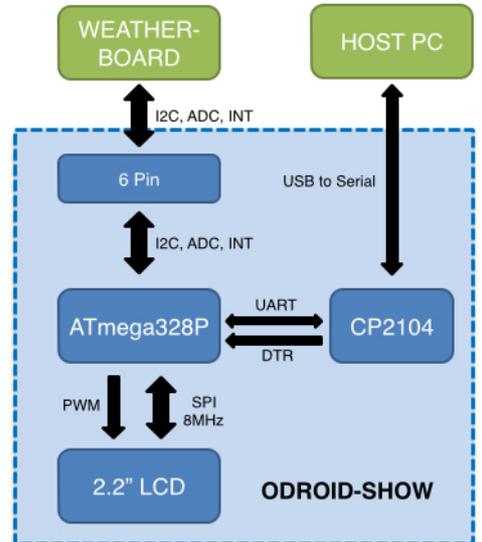


Diagrama por bloques de la placa meteorológica usando ODROID-SHOW



Sea un héroe y salvar a tu pueblo con una estación portátil con U3 y una placa meteorológica

# CONOCIENDO A UN ODROIDIAN

DENIS ODINTSOV (@OVERSUN)

PROGRAMADOR AVANZADO Y EXPERTO EN XBMC

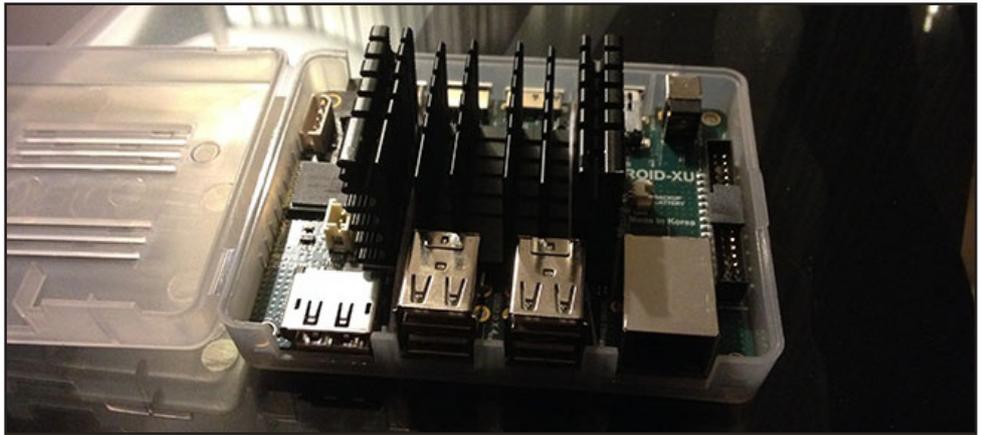
*Por favor, háblanos un poco sobre ti.*

Mi nombre es Denis Odintsov. Soy de Rusia y vivo en Polonia, trabajo para una compañía de EE.UU. llamada Cisco Systems. En cuanto a mis inquietudes, me encantan los juegos de ordenador, los libros, los lugares tranquilos, el aire limpio y la buena comida. Aunque suene un poco raro, estas son las cosas que me interesan a mis 34 años.

*¿Cómo fueron tus inicios con los ordenadores?*

Para que te hagas una idea de cuánto tiempo llevo con los ordenadores, puedo decirte que recuerdo un artículo que hablaba de una tecnología que cambiaría el mundo denominada “World Wide Web”. Sí, lo leí en una revista en papel! Yo no tenía ordenador por aquel entonces, fue hace mucho tiempo, pero me me pareció muy interesante. Recibí mi primer ordenador al poco tiempo. Se trataba de un puntero Intel 286 de 25Mhz de principios de los noventa. No recuerdo la fecha exacta, era muy joven y ya hace mucho tiempo de ello.

Pero vale ya de viejas historias, vamos lo más interesante: ¿Cómo empecé con ODROID? Básicamente te puedo decir que ya conocía este tipo placas, ya que tenía una Raspberry pi, y me encantaba. La utilizaba como un pequeño centro multimedia. Era realmente increíble, puesto que conocía el mundo de los PC desde mucho antes del 2012, y ni siquiera podía imaginar algo así por aquel entonces. El único problema era la velocidad de los menús. La Pi era demasiado lenta.



**Aunque Denis terminó con una caja para el XU que no podía cerrar, se las ingenió para personalizar su propia placa usando un disipador sin ventilador para un funcionamiento totalmente silencioso.**

Así que, como usuario de x86 desde hace mucho tiempo entre en Internet y compré “algo rápido” sin tener ningún conocimiento sobre el mundo ARM. Realmente, quiera comprar algo más pequeño pero con más Mhz. En este punto, debo decir ODROID era la única opción. Pienso que el U2 es inigualable, tanto en su diseño interior como exterior. Es la pieza más perfecta de hardware que the tenido en mis manos.

Y luego vino la realidad, que de hecho es la parte más increíble de la historia: Acababa de enterarme de que no podía coger algo compilado para Raspberry Pi e instalarlo en otra placa ARM (¡No se rían!) . Fue como una ducha de agua fría para alguien que sólo conocía el mundo x86. Por suerte, ya existía una distro Linux en desarrollo y ese es mi terreno. Si algo puede ejecutar Linux, puedo hacerle frente.

*Has sido uno de los principales desarrolladores de la comunidad para XBMC duran-*

*te más de un año. ¿Qué tipo de experiencias has tenido al trasladar con el popular centro multimedia para ODROID?*

La gente que lleva con ODROID bastante tiempo puede recordar la situación de Linux en ese momento: XBMC 11 era una versión estable, pero obsoleta, y XBMC 12 sólo mostraba una pantalla en negro. Pero lo bueno de Linux es que siempre hay recursos. El hardware soporta aceleración 3D y existían unas cuantas demos GL para eso. Lo que hice fue simplemente atar los cabos sueltos. Agilice la carga inicial de visualización usada en las demos GL para XBMC 12, y logre iniciarlo y ejecutarlo sin problemas. Necesite alrededor de una semana para trasladar 500 líneas de código de un programa a otro. No fue un trabajo difícil.

Y entonces comenzó la segunda parte de mi experiencia con XBMC: un nuevo mundo de hardware que ni siquiera existía en la arquitectura x86. Realmente necesite varios días para comprender que, en el mundo ARM, la descodifi-

cación de video es una cuestión totalmente diferente. Y así comenzó mi reto.

Si miro atrás, lo curioso es que probablemente, el código en el que eventualmente trabajaba estaría listo en un par de semanas. Sin embargo, existía un molesto fallo de Malí en el kernel 3.8 que estaba haciendo que la GPU cambiara toda la pantalla a 6-10 fotogramas por segundo (FPS). Y nadie sabía realmente donde estaba el problema. Pasé literalmente varios días intentando comprender cómo funcionaban las cosas, para entender por qué la salida de vídeo era tan lenta, pensando que la cuestión estaba en algún lugar de la decodificación por hardware. Esta actividad probablemente contribuyó al desarrollo de mis conocimientos como programador más que en cualquier otro momento de los últimos años.

De hecho no soy programador como tal, ya que mi área de conocimiento se centra mayoritariamente en la resolución de problemas y diseño de sistemas. La programación es un hobby para mí. Y quisiera destacar que no creo que mi mayor contribución fuese el código que permitiera a XBMC ejecutarse correctamente en ODROID. Sinceramente, creo que fue el efecto que tuvo mi trabajo en la propia comunidad ODROID.

Recuerdo como de pequeño y entusiasta era el foro a comienzos del 2013, y ahora puedo ver en lo grande y poderoso que se ha convertido. Y me enorgullece pensar que mis humildes esfuerzos para ejecutar XBMC pueden haber animado a la gente a comprar más placas, así como atraer a personas que realmente saben programar. Puedo ver como el código que escribí, cogido por personas para las que la programación es más que un hobby, se ha convirtió en algo increíble desde que era un simple y plano código fuente. Y a partir de ese momento fueron mucho más allá, donde ni siquiera ya sirvo de ayuda para esos programadores. Este es el verdadero logro para mí y de lo que estoy más orgulloso.

*¿Cómo te convertiste en un experto en desarrollo Linux?*



**Nuestro hombre Denis disfrutando de un momento de tranquilidad en la playa.**

*¿Cuándo puede ser más fácil convertirse en un experto en desarrollo teniendo Internet en tus manos? Linux es abierto, esta increíblemente documentado y te permite controlarlo todo. Simplemente elige una tarea, abre un editor de texto y ¡Adelante! En la primera dificultad que tengas, incluso si es en la primera línea del programa, la buscas en google y encontraras cientos de páginas con información. Crea pequeños programas que no pueden hacer nada en especial, crea programas que hagan cosas tontas, y en algún momento te sentirás capaz de programar todo lo que puedas imaginar. Y poco después, verás que algunas de las cosas que creas son buscadas por miles de personas en todo el mundo. Realmente es así de simple.*

*¿Qué otras aficiones e intereses tienes?*

Fitness, libros, crear cosas ... ser feliz es probablemente la descripción más común de la única afición que tengo. A veces por los lugares donde estoy, a veces por la gente que conozco y a veces por las cosas que hago y el efecto que tienen.

*¿Está involucrado con cualquier otro proyecto de software o hardware?*

Últimamente me está gusta la programación para dispositivos Apple, me da un poco de diversión. iOS tiene sus pro y sus contras, y no te ofrece tanta libertad como Linux, pero me es relevante y puede ser rentable.

*¿Qué tipo de novedades en hardware te gustaría ver en futuros ODROIDS?*

La última vez que visite el sitio de ODROID, encontré algunas cosas increíbles - como la placa UPS. ¡Así se hace, Hardkernel, Es realmente impresionante! De modo que, lo único que se me ocurre ahora es un módulo de GPS/GSM, ya que uso mi U2 como una puerta GSM. En cuanto a los chips en sí mismos, cualquier cosa que soporte Linux podría llegar a hacerse. Hardkernel está haciendo grandes placas - pequeñas, sólidas y hermosas

**¡El trabajo de un ODROID nunca termina! Usar el U2 como una puerta GSM es un modo de hacer una nube personal.**

