Crea una App para la Placa Meteorológica • Open Media Vault • Cómo instalar Node.js



# Magazine VIRTUALIZA VIRTUALIZA ALADALIZA ALADALIZA

# PLATAFORMA DE COMPUTACION DISTRIBUIDA QUE APROVECHA AL

MAXIMO EL BAJO CONSUMO DE ENERGIA DE ANDROID

- SO DESTACADO: CODE MONKEY
- JUEGOS LINUX: DOSBOX
- VEHICULO TODOTERRENO AUTOMATICO: Programar la navegación GPS

\*KVM

BOINC MONSTER: UN TREMENDO Cluster de 135W Con 96 Nucleos



# 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





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

**Dirección:** Max-Pollin-Straße 1 85104 Pförring Alemania

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

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



# EDITORIAL

La gran noticia de este mes es que ODROID está a la venta en Estados Unidos desde http://www.ameridroid.com. Con sede en California, Ameridroid ofrece envíos económicos para sus clientes nacionales y los residentes en Estados Unidos recibirán los paquetes mucho más rápido. He aquí un extracto de su sitio web:

> "Quizás tu historia es la misma. Santa Claus respondió a una carta cuando tenía 7 años, con un brillante soldador. En poco tiempo estaba desmotando aparatos electrónicos y salvando piezas para hacer una radio. Puso el cable de antena en la ventana junto a su cama (para una mejor recepción) y escucho las emisoras de radio AM con un auricular (mono, por

supuesto) hasta que se quedó dormido soñando con su próximo proyecto de electrónica. Vivimos en una época maravillosa. Un momento en el que la tecnología no es un factor decisivo para poder inventar, siempre y cuando podamos imaginar. Una era en la que se consiguen potentes ordenadores por menos de 100\$. Nos encanta inventar cosas y muchas de nuestras ideas requieren una cierta capacidad informática. Por ello, nos encantan los ordenadores de placa reducida. Cuando hablamos de este tipo de equipos con una potente capacidad de cálculo a un bajo precio, ODROID es el líder indiscutible. ¡Queremos conocer tus inventos!. Envíanos una descripción y una foto y nosotros la colocaremos e nuestra página web"

Si estas esperando una versión más económica del XU3, Hardkernel acaba de anunciar el XU3-Lite por 99 dólares. Incluye todas las características de la XU3 original, excepto el DisplayPort y los sensores de corriente y Voltaje. Utiliza un procesador Exynos 5422 con una velocidad de reloj de 1.8/1.3 GHz en lugar de 2.0/1.4 GHz de la placa original. Puedes consultar una nota de prensa en http:// bit.ly/lsf7bji.

Este mes destacamos BOINC, la aplicación que te permite participar en una red de supercomputación a nivel mundial con un ODROID, algunos emuladores X86, la instalación paso a paso de la aplicación de virtualización KVM para el XU3, así como la tercera entrega de la popular serie del vehículo todoterreno automático. Venkat nos trae unas guías sobre Node.js y Open Media Vault. El Sistema Operativo destacado de este mes es Code Monkey, una imagen todo-enuno para desarrolladores. Viene repleto de compiladores, entornos de desarrollo y herramientas de gestión de código para que puedas empezar a programar en ODROID de inmediato.

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/.

#### **ODROID** Magazine

#### Rob Roy, Editor Jefe

Soy un programador informático que vive y trabaja en San Francisco, CA, en

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



Bo Lechnowsky, Editor

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

#### Bruno Doiche, Editor Artístico

Consiguió sus habilidades informáticas después de lograr que una fi-

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





Nicole Scott, Art Editor

Nicole es una experta en Producción Transmedia Estrategia Digital espe-

cializa 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 ent http://www.nicolecscott.com.



Manuel Adamuz, Spanish Editor

Tengo 31 años y vivo en Sevilla, España,

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

# INDICE

**VIRTUALIZACION KVM - 6 APP ANDROID: INSTALACION NATIVE BOINC - 8** LA HISTORIA DE UBUNTU- II Jbunti **CONSTRUIR UN BOINC MONSTER - 12 CONEJOS Y TRUCOS: MINERIA SIN MICROSOFT CON FREEMINER - 13** FAKE86 8088/8086 VIRTUAL - 14 **JUEGOS LINUX: EMULADOR DOSBOX - 16** DOSBOX **USAR UN DOBLE MONITOR CON UN XU3 - 19 SO DESTACADO: CODE MONKEY - 20 CONSEJOS Y TUCOS: ROTAR LA PANTALLA EN ANDROID - 23 CONSTRUIR UN VEHICULO TODOTERRENO AUTOMATICO - 24 APLICACION PARA LA PLACA METEOROLOGICA - 26** 99549 8. \*C 0.04 265 Lu **COMO INSTALAR NODE.JS - 28 OPEN MEDIA VAULT - 30 CONOCIENDO A UN ODROIDIAN - 32** 

# VIRTUALIZACION KVM SOBRE EL ODROID-XU3 UN MUNDO FELIZ

**DE INFORMATICA** 

por Mauro Ribeiro

os proyectos de Máquina Virtual basados en Kernel de código abierto (KVM) permiten a un ODROID, y a muchos otros equipos alojar un segundo sistema operativo usando el mismo kernel. Con KVM se puede ejectuar múltiples máquinas virtuales con imágenes de Android y Linux sin ser alteradas. Cada máquina virtual tiene una tarjeta de red, un disco duro y un sistema gráfico vitual.

Este artículo muestra un caso práctico de KVM/Virtualización sobre un XU3. La virtualización en ARM todavía está en una etapa muy temprana y no debe aplicarse en entornos de producción. La configuración KVM requiere de algunos conocimientos de Linux puesto que necesitamos recompilar el kernel, actualizar los gestores de arranque y compilar qemu. Todas las instrucciones que se aparecen a continuación fueron escritas a partir de la imagen de Ubuntu 14.04 y se deben ejecutar sobre la propia placa.

#### Actualizar el gestor de arranque

Abre una ventana de terminal en Ubuntu 14.04 y escribe:

\$ sudo odroid-utility.sh

La utilidad ODROID viene con todas las imágenes oficiales de Hardkernel

va1f124f790d47619a2c558f9





Desplázate a la opción 2 como se muestra en la captura de pantalla, luego selecciona la opción 5 para actualizar el gestor de arranque.

#### Añadir soporte KVM

Haz una copia de seguridad de tu actual kernel y dtb:

<pre>\$ mkdir ~/backup_pre_kvm</pre>
<pre>\$ cp /media/boot/* ~/backup_pre_kvm</pre>
Compila el Kernel con soporte KVM:
<pre>\$ git clone https://github.com/hardkernel/linux.git</pre>
-b odroidxu3-3.10.y
\$ cd kernel
<pre>\$ make odroidxu3_kvm_defconfig</pre>
\$ make -j9
<pre>\$ cp arch/arm/boot/zImage /media/boot/zImage</pre>
<pre>\$ cp arch/arm/boot/dts/exynos5422-odroidxu3.dtb /me-</pre>
dia/boot/exynos5422-odroidxu3.dtb
<pre>\$ make modules_install</pre>

Llegados a este punto, deberías tener una placa con un kernel capaz de utilizar KVM.



Para conocer más sobre KVM o para contribuir con el proyecto, visita la página web de KVM: http://bitly/18isyvK

#### Compilar el kernel invitado

Para el Kernel invitado, usaremos el kernel 3.9 del Virtual Open Systems, que está diseñado específicamente para los modelos ARM Foundation. Cualquier versión del kernel que soporte opciones de invitado KVM y el hardware de modelo Foundation debería funcionar de igual forma.

Si no desear perder tiempo en compilar el kernel, puedes descargar algunos ficheros binarios precompilados que he creado escribiendo en una ventana de terminal:

```
$ cd ~/kvm
```

```
$ wget http://odroid.in/guides/kvm/kernel.tar.xz
$ tar -Jxf kernel.tar.xz
```

Para compilar el núcleo desde la fuente, escribe lo siguiente:

```
$ git clean -f -d -x
$ git remote add kvm kernel https://github.com/vir-
tualopensystems/linux-kvm-arm.git
$ git fetch kvm kernel
$ git checkout kvm kernel/kvm-vexpress-3.9
$ wget -0 .config http://odroid.in/guides/kvm/guest
config
$ make -j9 zImage dtbs
$ mkdir ~/kvm
$ cp arch/arm/boot/zImage ~/kvm
$ cp arch/arm/boot/dts/rtsm ve-cortex a15x1.dtb ~/kvm
```

#### **Compilar Qemu**

Una vez más, si no deseas compilar desde la fuente, existe una versión precompilada de Qemu disponible, escribe lo siguiente en una ventana de terminal:

```
$ wget -0 ~/kvm/qemu.tar.xz http://odroid.in/guides/
kvm/qemu.tar.xz
$ cd /usr/local && tar -Jxf ~/kvm/qemu.tar.xz && cd -
```

#### De lo contrario, escriba esto para compilar Qemu:

```
$ git clone git://github.com/virtualopensystems/gemu.
git -b kvm-arm-virtio-fb-hack
$ ./configure --target-list=arm-softmmu --audio-drv-
list="" --audio-card-list="" \
--enable-fdt --enable-kvm --enable-sdl -prefix=~/kvm
```

#### **Pruebas**

Para las pruebas, he creado dos imágenes precompiladas: una con Ubuntu 14.04 y otra con Android 4.1.2. Para descargar la versión de Android:

```
$ cd ~/kvm
$ wget http://odroid.in/guides/kvm/android.jb.img.gz
$ gzip -d android.jb.img.xz
$ wget http://odroid.in/guides/kvm/android.sh
$ chmod +x android.sh
```

\$ ./android.sh

Para descargar la versión de Linux, Lubuntu 14.04:

```
$ cd ~/kvm
$ wget http://odroid.in/guides/kvm/lubuntu.xu3.img.gz
$ gzip -d lubuntu.xu3.img.gz
$ wget http://odroid.in/guides/kvm/lubuntu.sh
$ chmod +x lubuntu.sh
$ /lubuntu.sh
```

¡Feliz hackeo! Estamos deseando ver lo que la gente es capaz de hacer con las imágenes KVM. Gracias a Suryian por sus habilidades XEN, a Fanta por su versión de u-boot con soporte Hypervisor, y a Virtual Open Systems por crear el kernel, gemu y probar las imágenes de Android.

## INSTALACION DE NATIVE BOINC UNA GUIA GRAFICA

por Uli Abromeit

OINC es una plataforma de supercomputación en red ampliamente utilizada por científicos, universidades y particulares que ayudan a explorar las fronteras del conocimiento humano. Se trata de un proyecto de voluntariado con el objetivo de mejorar el mundo descubriendo nuevos teoremas, medicinas innovadoras y otros hallazgos. Los voluntarios pueden donar el potencial de sus ordenadores que normalmente no utilizan a determinados proyectos para ejecutar cálculos científicos. Puedes empezar por descargar la aplicación NativeBOINC desde Play Store de Google o desde la página principal de Native-BOINC en http://bit.ly/109rRxg.

# 331	
X NativeBOINC	# 🔟 🖛 🗰 🐙 💼 311
Do you want to install this application? It will get access to:	X Installation Wizard
PRIVACY	Welcome to NativeBOINC!
modify or delete the contents of your SD card     read the contents of your SD card	Please tap. Next: button io to download and install native BOINC client.  I have problems with dient version 0.5.9x, select 0.4.9x
	Version to install: 0.5.9.4
	Installation place Installation is not recommended, because some problems has not been solved. If application closes unexpectedly, please use "Proce close" and run again. Application Nille all clients at start up. PDINC client will be working in backgraund. Please use "Shuddown" option to shuddown client. Installation place Installation Insta
l - Tras la descarga haz clic en el archivo nativeboinc.apk y pincha en el botón Install	2 - Haz clic en "Next" para iniciar la instalación

#×米□▼								1	3:12
💓 Operations progress									
No operations in background.									
BOINC client: Operation finished									
									Next
	1	410	<del>()</del>	$\Box$	Ē	Ċ			

**3** - Haz clic en "Next" para completar la instalación



4 - Fija la contraseña de acceso para el equipo remoto, así como el nombre de host si ejecutas más de un host

#### **APP ANDROID**

# ¤ × ¥							🚅 🗎 3:1
💢 Select project							
This is a list of supported projects. To ç	pet info abou	ut project plea	ase longer tap e	ntry.			
Enigma@Home http://www.enigmaathome.net/							
MilkyWay@Home 0.4.1 http://milkyway.cs.rpi.edu/milkyway/							
OProject@Home							
theSkyNet POGS							
SubsetSum@Home 0.1.2							
WUProp@Home http://www.rop.boinc-ef.org/							
Yoyo@Home http://www.recbeokraft.net/yoyo/							
Asteroids@Home							
							Other project
	-	-E-1		0	0		

5 - Selecciona uno de los proyectos de la lista de proyectos so-

portados, coloca el e-mail y la contraseña, luego pulsa OK

c (7.0.36)	Upgraded to NativeBOINC v0.4.4.2.3b	
	Version 0.4.4.2.9 (beta) (2013-09-09): - fixed hVFs in various places - disable debug loging	
	Version 0.4. <b>4.2.2 (bet.) (2013-07-12):</b> - spdatet German transitions - some small facts for English translation	
	version 0.4.4.2.1 (bea) (2013-07-05): 	
	Version 0.4.4.2 (beta) (2013 0.6.19): - frase the devise fair CPU Limits (comparing fair CPU Limit – final screen lock (hwitwy infin) - assiming to calificative julio when calient cant detect battery - added wholge previews - - added wholge previews -	
	Version 0.4.4.1.3 (bets) (2013-05-13): - property treats intips url in "Enter project URL" dialog in "Belet project" screen.	
	Versiter 0: 4,1:2 (bat) (2013;24:22): fixed blocescapes (improve Satari dir detection) anded vestme: systime to RuyCatche : sports	
	n an	

6 - NativeBOINC se ejecutará después de hacer clic en el botón "Dismiss"



7 - Lista de tareas y ajustes del proyecto BOINC que actualmente están en ejecución



8 - Haz clic en "Manage Client" para añadir otros proyectos BOINC y ajustar la configuración

X Local preferences								
COMPUTING	NETWORK	DISK & RAM						
While device is on batteries								
Run always when plugged								
While device is in use								
Use GPU while device is in use								
Dnly after device has been idle for 0.0 min								
While processor usage is less than $0.0$ %								
Switch between applications every 600.0 min								
While battery charge level not less than 10.0 %								
While battery temperature less than 60.0 *C								
Jse at most 100.0 % of the processors								
Jse at most 100.0 % CPU time								
Computing time preferences								
		Cancel Apply default Apply						
	U 📭	🖬 💓 💓 11:49 🗎						

9 - Haz clic en "Local Preferences", configura los ajustes de "Computing" para que coincidan con los que se muestran



10 - Haz clic en "Network" y ajusta la configuración de red en base a tus preferencias de velocidad

#### **APP ANDROID**



II - Haz clic en "Disk & Ram" y ajusta las opciones para que coincida con las que se muestran en la imagen.

#### 12 - Haz clic en "Preferences" y ajusta las opciones de interfaz de usuario para que coincida con las que se muestran en la imagen.

Native BOINC direct						
		nativeboinc (7.1		SHUTDON	VN A MANAGE CLIENT A PREFEREN	CES ANTIVE CLIENT
NEWS AND UPDATES		PROJECTS				NOTICES
Latest news Displaye latest news list		Enigma@Ho Total:10249592 RAG				100 (7.69%)
Notify about news and new binaries		OProject@H	Commands			1000 (76.92%)
Notifications adout news and new binaries every o nours enabled CLIENT STARTING		Total:458703 RAC:3	Select all			100 (7.69%)
Autostart client		Total:22396713 RAG	C30429 Host 633156 Update			
Always autostart NETWORK OPTIONS		WUProp@Ho Total:903851 RAC:1	me Suspend Suspend			100 (7.69%)
Set up hostname			Detach			
Current hostname: odroidu36			No new work			
Allow remote access Allow remote access for other hosts			Properties			
Access password Change access password for client						
Access host list						
List of hosts whose have access to BOINC client						
5 C J 4 4 0 R	🖬 💢 💥 11:51 🗎	Ĵ	- • • O	₹ <sup>2</sup>		) 💢 💢 11:48 🗎

13 - Haz clic en "Native Client", luego seleccionar y activa la opción Autostart Client

14 - La pestaña "Projects" muestra una lista de todos los proyectos junto con la información detallada de cada uno de ellos. Gestiona un proyecto haciendo clic en su nombre

🐹 nativeboinc (7.0.36)		и знато			
	TASKS				
				2014-10-13	
				2014-10-1; 84.0005	
	Select all			2014-10-1: 34:7835	
	Suspend			2014-10-1; 26:316%	
Data collect version 4 4.17 WUProp@Home	Abort			2014-10-2	
	Properties			2014-10-1	
				2014-10-1	
				2014-10-13	
	e e o [	4		11:4	7 💼

15 - La pestaña "Tasks" muestra una lista de las unidades de trabajo actuales, que pueden ser gestionado de forma individual

8 Boinc auf Android-Gerät	Nati	NativeBOINC - User stati $ imes \setminus$ +								:		
Native BOINC Client for Android devices and not only :)												
START PAGE User statistics for All												
LATEST NEWS	Projec	t stat	listic	s Ho	st statistics User stat	istics Team statisti	cs					
DOWNLOAD	Pleas	e sel	ect i	ntere	esting project:	All	Sear	ch:		S	Search	
INSTALLING	Following statistics concerns only Android devices. In other words we counts credits, RAC, ranks for only											
FOR USERS	Androi are fet	a dev ched	from	comproje	putes total credits, RAC ct sites and are public	tor users and team: y available.	s based	on Android	devices. Al	i informatio	ons	
DEVELOPMENT	Last u	pdate	: 201	4-09	-27 14:11:24							
STATISTICS	Total o	ount:	731	94								
BIAHOHOG										▶100	₩500	<b>HH</b> 1000
SEND QUESTION	last month	last week	last day	Bank	Name	Team	Country	Total credits	Credit/ day	Credit/ week	Credit/ month	RAC
NOTIFY BUG	New	New	New	1	JagDoc	Planet 3DNow!		26267079.96	26267079.96	26267079.96	26267079.96	60896.77
THANKE FOR	New	New	New	2	PS_app		=	17433851.50	17433851.50	17433851.50	17433851.50	36860.32
THANKS FUR	New	New	New	3	Ray_GTI-R		***	11518350.63	11518350.63	11518350.63	11518350.63	5814.25
ABOUT ME	New	New	New	4	alez [ TSBT's Pirate ]	The Scottish Boinc T	*	5118523.43	5118523.43	5118523.43	5118523.43	23597.37
	New	New	New	5	Jerzy Gieorgijewski	BUINC.Italy	=	4521816.37	4521816.37	4521816.37	4521816.37	25680.93

16 - Lista de algunos proyectos NativeBOINC compatible con ARM. Todos los que aparecen están disponible para ARM excepto OProject, PrimeGrid, SubsetSum@Home y YAFU.



por Uli Abromeit

# CONSTRUIR UN BOINC MONSTER 96 NUCLEOS CON TAN SOLO 135 VATIOS



n enero de 2013 decidí instalar la versión de Android de BOINC en un ODROID-X2, con el fin de usarlo para ayudar a estudiar el calentamiento global, descubrir cuerpos celestes y otras investigaciones científicas. BOINC fue creado en la Universidad de California en Berkeley como una forma de reciclar el tiempo no utilizado de la CPU para resolver grandes problemas matemáticos y estadísticos de una forma eficiente. Distribuye la carga de trabajo entre muchos equipos que procesan los cálculos cuando éstos no están en uso.

Hay muchos proyectos BOINC disponibles para la plataforma ARM Android, se pueden encontrar en la lista oficial de proyectos en http://bit.ly/1r4wpzu. Mi clúster BOINC está diseñado para ejecutar BOINC constantemente 24/7, usando un total de 135W. Con el tiempo he añadido 3 X2 más y un XU, junto con algunos U2s y U3s.

Decidí que era hora de hacer algo con el caos de mi mesa de trabajo, de modo que diseñé un rack para guardar todos los ODROIDs, al que he llamado "BOINC MONSTER". Para organizar los ordenadores BOINC he usado un rack de 19" con algunas modificaciones.



Prototipo de mesa de trabajo con el clúster BOINC

#### 12 ODROID-U2

I Switch de red con 24 puertos 2 Switch HDMI con 8 puertos (Aten VSO80IH) 2 PSU 5V/20A (TDK-Lambda HWSI00-5/A)



Diseño preliminar del clúster BOINC con los ventiladores de 40x40mm instalados

Después de experimentar con la distribución, me encontré con que los U2s se calentaban y hacían mucho ruido con los típicos ventiladores. Así que cogí unos ventiladores Sharkoon 60x60x25mm de 12V (versión Power) y los conecté a los puertos para ventilador de los U2s.



Clúster BOINC con ventiladores de I2V conectados

Los U2s están conectados a un Motorola Lapdock usando switches HDMI y son controlados con un ratón conectado a dispositivos USB de 4 puertos. Prefiero controlos con periféricos USB en lugar de hacerlo de forma remota.

Cuando la distribución del rack U2 era la idónea, añadí 6 ODROID-U3 a los 12 ODROID-U2 ya montados. Una vez más, me di cuenta que con el disipador de calor más pequeño, el U3 se calentaba demasiado usando los ventiladores de 12V 60x60x25mm a 5V. Así que añadí un transformador DC-DC regulable para enviar 9V a los ventiladores y así mejorar su efectividad.



Clúster con una Combinación de U2, U3 y XU



Primer plano del transformador DC-DC regulable

El siguiente paso en el proyecto es agregar un tercer switch HDMI de manera que todos los nuevos ODROIDs puede utilizar el monitor Lapdock, y por supuesto añadir algunos equipos más al BOINC MONSTER.



### MINERIA SIN MICROSOFT CON FREEMINER AHORA QUE MOJANG HA ZARPADO HACIA REDMOND, ¿QUE TAL UN

*JUEGO DE CODIGO ABIERTO?* 

#### por Bruno Doiche

orpresa y pavor circularon por todo Internet cuando el juego favorito de todo el mundo fue comprado por Mi-



El mundo de FreeMiner se parece a Minecraft

crosoft por 2,5 billones. ¿Y ahora qué? Bueno, lo puedes celebrar con una versión de código abierto con todas las funciones de Minecraft que puede ejecutar en tu ODROID:

\$ sudo apt-get install git subversion build-essential cmake libbz2-dev libpng12-dev libjpeg8-dev libfreetype6-dev libxxf86vm-dev libg11-mesa-dev libsqlite3-dev libvorbis-dev libopenal-dev libcur14-openss1-dev libluajit-5.1-dev libleveldb-dev libsnappy-dev libgettextpol libmsgpack-dev libgles1-mesa-dev libgles2-mesa-dev

#### Descarga la aplicación irrlicht usando Subversion o Github:

\$ svn checkout svn://svn.code.sf.net/p\
/irrlicht/code/branches/ogl-es irrlicht

```
$ git clone -b ogl-es-svn --recursive \
https://github.com/freeminer/irrlicht.git \
irrlicht
```

#### Compila Irrlicht:

\$ make -j4 -C irrlicht/source/Irrlicht

Descarga el código fuente de Freeminer:

\$ git clone --recursive \
https://github.com/freeminer/freeminer.git

Compilalo y configura el driver de vídeo OpenGL ES:

```
$ cd freeminer
$ cmake . -DENABLE_GLES=1 \
-DIRRLICHT_INCLUDE_DIR=../irrlicht/include \
-DIRRLICHT_LIBRARY=../irrlicht/lib/Linux/\libIrrlicht.a
$ make -j4
$ echo "video_driver=ogles1" >> freeminer.conf
$ echo "enable_shaders=0" >> freeminer.conf
```

¡Eso es todo! Freeminer como Minecraft, también tiene servidores disponibles para disfrutar el modo multijugador. Para aprender más sobre Freeminer y unirte a los foros, visita la página principal del proyecto en http://freeminer.org

### FAKE86 UN 8088/8086 VIRTUAL EXTREMADAMENTE RAPIDO

por Jeremy (Cartridge) Kenney

En la década de 1980, era considerado un modelo portátil y pesaba tanto como una maleta llena de bolos. ¡Con Fake86 puedes llevar un 8086 en el bolsillo delantero de tu camisa!

lguna vez has pensado si hay emuladores IBM para PC y aplicaciones que permitan ejecutar pesados programas de DOS. Aunque hay opciones casi ilimitadas para la emulación Intel, puede resultar un tanto difícil a quienes no son muy expertos conseguir que funciones los programas del 8088.

El popular Qemu (http://www. qemu.org) es un poco grande y se necesita un montón de cadenas de caracteres para configurarlo correctamente. Bochs (http://bit.ly/1nuFeYj) es similar a Qemu, pero con aún más cadenas de caracteres. DOSBox (http://www.dosbox.com) parece funcionar más o menos bien, pero es necesario configurar constantemente los ciclos y otras opciones de emulación cuando en el juego aparezcen errores o líneas verticales, debido a que usa un motor especial para hacer funcionar a los juegos.

¿Y si hubiese un emulador que se centrara en una determinada CPU, un par de tarjetas de video y una tarjeta de sonido Adlib/Soundblaster de la epoca? Por suerte, Fake86 (http://bit.ly/1wcIpV6) está especializado en la virtualización de las mejores especificaciones de la época, con soporte para el 8088/8086 y para el grupo de instrucciones 80186 de 16 bits. Estas son algunas de sus características:

• El motor de la CPU que se incluye con Fake86 es un intérprete puro que

todavía es capaz de ejecutarse razonablemente rápido

- El soporte de audio incluye emulación de Sound Blaster, tarjeta de música Adlib FM, Disney Sound Source y altavoces de PC estándar
- Se puede emular un ratón serial estándar compatible con Microsoft sobre COM1
- La emulación de diskettes y discos duros con interrupción de 13h utiliza archivos de imagen de disco
- Puede ejecutar cualquier sistema DOS de 16 bits (MS-DOS, DR-DOS, PC-DOS y FreeDOS)
- Compatible con la emulación de video Hércules, CGA, MCGA y VGA (excepto 640x480 a 16 colores)
- La Emulación Ethernet con libpcap y WinPcap está habilitada
- Windows 1.0 y 2.0 funcionan perfectamente, son necesarios algunos ajustes para ejecutar algunos juegos que requieren algo más que un 286 (por ejemplo, Wolfenstein)



El logotipo original de MS DOS parece que fue diseñado por un artista de graffiti



Tras leer todas esas características de Fake86, ¡Seguro que nunca habrías pensado que DOS podría ser tan grande! Es incluso mejor cuando lo ejecutas en tu ODROID en alta definición (HD). Todo lo que necesitas es un par de minutos para su instalación, en lugar de buscar en Google cómo solucionar errores y problemas técnicos de juegos.

I. Echa un vistazo al post de Fake86 del foro en http://bit.ly/1pgoyO8 y descarga el paquete de instalación desde http://bit.ly/1Dx0Kzj. Descomprimir el archivo y lee el archivo "readme" antes de continuar.

**2.** Abre una ventana de terminal y escriba lo siguiente para instalar Fake86 (el nombre de archivo puede ser diferente):

\$ sudo dpkg -i <fake86\_0.12.9.19-1\_armhf.deb>

**3.** Crea una imagen de disco en blanco para usarla como un disco duro. Abre una ventana de terminal y escribe:

\$ sudo dd bs=1M count=100 if=dev/
zero of=./nameofyourimage.img

Ten en cuenta que los parámetros "bs=1M count=100" dará como resultado una imagen de disco sin formato de 100MB. Si deseas tener más espacio en disco en tu imagen, aumenta el valor "count".

4. Para iniciar Fake86, escribe:

\$ sudo fake86 -hd0 nameofyourimage.img

Por otro lado, si deseas arrancar desde una unidad de disquete USB externa o desde una unidad de CD/DVD-ROM, inserta el disquete o DVD y escribe:

```
$ sudo fake86 -hd0 nameofyourim-
age.img \
-fd0 Path/To/Media/Floppy/Or/
CDROMDrive -boot 0
```

Tras arrancar la nueva imagen, aparecerá una ventana de Fake86 donde se muestra el Turbo XT BIOS. Se te pedirá que habilites la compatibilidad con discos grandes si resulta que has aumentado el tamaño de tu imagen. Pulsa "Yes", luego pulsa 1 para crear una partición primaria de DOS y pulsa 1 de nuevo. La verificación durará uno o dos segundos.

Crea tu partición y deja la etiqueta en blanco. Una vez creada, puede salir de fdisk y cerrar Fake86 pinchando en el botón X sobre la ventana. Los cambios se guardan automáticamente.

**5.** Formatea la unidad abriendo el Terminal y escribiendo:

\$ sudo fake86 -hd0 nameofyourimage.img -fd0 \ /Path/To/Media/Floppy/Or/CDROM-Drive -boot 0

#### En el prompt de DOS, escribe:

#### > format C:

Confirma que los datos se pueden sobrescribir, lo cual es seguro porque no borra nada fuera de la imagen que has creado. Tras completarse el formateo, la imagen está lista para DOS y Windows. La instalación de DOS 5.0.3 es muy sencilla y puede ser instalado en tu máquina virtual una vez hayas arrancado desde la parte de DOS. Una vez ejecutado DOS, está listo para jugar a cualquier juego o ejecutar un programa compatible con 8088/8086/80186.

Fake86 tiene las siguientes opciones:

-fd0 -fd1: Estas son tus unidades de disquete

-hd0 -hd1: Estos son tus discos duros

-boot 0: Arrancará desde la primera unidad de disquete

-boot 1: Arrancará desde la segunda unidad de disquete

-boot 128: Arrancará desde el primer disco duro en busca de la instalación de tu sistema operativo, siempre y cuando hayas puesto los archivos de instalación en el disco duro

-bios <nombrearchivobios>: Especifica una BIOS que puedes haber descargado de Internet

-nosound: Esto desactiva el sonido, se utiliza normalmente cuando sólo trabajamos con DOS

Wolfenstein 3-D para MS-DOS fue revolucionario por su perspectiva en l<sup>a</sup> persona -resw -resh: Esto fija la resolución de la pantalla, requiere los siguientes parámetros: "-resw 1280 -resh 720" para 720p y "-resw 1920 -resh 1080" para una pantalla de 1080p

-smooth: Esto suavizará el filtro de pantalla

-fullscreen: Usa este comando con resw y resh si deseas tener una pantalla completa a escala. Si se usa solo, obtendrá una imagen sin escala a pantalla completa.

-ssource: Este comando activa la fuente de Sonido en LPT1

-latency and -samprate: Estos comandos es mejor no tocarlos, pero si te encuentras con problemas técnicos de sonido puede utilizar estas opciones para solucionarlos escribiendo "-latency 150 –samprate 44100" (o cualquier otro número)

-console: Activa tu Terminal para que actúe como una consola interactiva para cambiar y expulsar los disquetes mientras se esté ejecutando el emulador.

¡Ahora puedes usar Fake86 para trabajar, jugar, navegar por Internet e invitar a tus amigos para que prueben el Wolfenstein 3D en un monitor a 1080p!



## JUEGOS LINUX: EMULADOR DOSBOX TUS JUEGOS ORIGINALES DE DOS EN HD

por Tobias Schaaf



DOSBox suele ser "pesado" para muchos equipos, normalmente se necesita un PC de gama alta para emular un 486 a 33 MHz. Ya que ODROID utiliza una arquitectura completamente diferente (ARM vs x86), tiene que realizar más trabajo durante la emulación. A pesar de su complejidad y múltiples capas, DOSBox funciona sorprendentemente bien en la plataforma ODROID.

Hace algún tiempo compilé una versión de DOSBox optimizada para ARMv7 a medida, parecía ejecutarse más rápido que la clásica versión de DOSBox que viene con la distribución oficial. Me llevó un tiempo comparar estas versiones y descubrir exactamente en que había mejorado la compilación optimizada para ARMv7.

A continuación encontrará una serie de pruebas que ponen de manifiesto las diferencias entre la compilación genérica de DOSBox y la compilación específica para ARM. La compilación personalizada de DOSBox para ARMv7 puede descargarse desde mi repositorio en http:// bit.ly/1DhCv6l.

#### Configuración

Configurar DOSBox puede resultar difícil en ciertas ocasiones. La mayoría

de los juegos se ejecutan con las opciones básicas. Otros en cambio, sólo funcionan con una configuración específica, así que opte por el conjunto de valores que mejor funcionan con la versión original del juego Quake, puesto que éste es muy exigente con el hardware.

Lo llamativo de Quake es que el juego en sí está en 3D sin necesidad de un entorno de escritorio gráfico. A diferencia de juegos como "Duke Nukem 3D" que cuenta con pocos objetos 3D y usa sprites 2D en muchas ocasiones, Quake ya usaba modelos en 3D similares a los que se utilizaban en juegos posteriores en Windows, algo admirable por aquel entonces.

No fue fácil conseguir la configuración correcta y tras un período de pruebas, terminé con los siguientes valores con un índice de errores y santos de fotograma prácticamente nulo:

core=dynamic						
cputype=pentium_slow						
cycles=fixed 32000						
cycleup=500						
cycledown=300						
memsize=32						
scaler=normal3x						

Se deben usar núcleos dinámicos para cualquier valor de ciclos fijos de más de 20.000. Pentium\_slow es la CPU con las mejores características y fijé los ciclos a 32.000, que son muy altos. Algunos programas de prueba mostraban una CPU Pentium a 1285 MHz. Elegí un número tan elevado porque el Quake con 32.000 ciclos ofrece una experiencia más fluida



OSBOX

#### Pruebas

Después de realizar varias pruebas, descubrí que resultaba difícil encontrar buenos indicadores de rendimiento. Me acordé de algunas aplicaciones de evaluación de cuando DOS era tan popular, pero eran difíciles de encontrar. Sin embargo, localicé un entorno de prueba para realizar diferentes pruebas de rendimiento bajo DOS denominado DOS Benchmark, disponible para su descarga desde http://bit.ly/1ttzaRR.

DOS Benchmark ofrece pruebas de CPU, GPU y de memoria, así como versiones demo de los juegos Doom y Quake que pueden ser utilizadas para la evaluación del entorno. Intente ejecutar todas las pruebas disponibles, pero no todas llegaron a funcionar. Encontré unas cuantas que funcionaron muy bien.

Por ejemplo, encontré una prueba con un cubo 3D giratorio ejecutado en DOS con grandes efectos visuales. Se ejecutaba relativamente rápido en el ODROID como se puede ver en las capturas de pantalla en la página siguiente.

#### Preuba 3DBench

La versión optimizada ARMv7 fue casi un 17% más rápida en esta prueba. Lamentablemente, esta prueba no es muy fiable si cambiamos los ciclos de la CPU. Puedes lograr resultados de más de 200 FPS con valores de 100.000 ciclos de CPU, pero incluso con estos altos números el emulador estaba lejos de funcionar mejor o incluso más rápido. Pude comprobar que la salida de vídeo

#### **JUEGOS LINUX**



Cubo giratorio bajo DOS



That is 44.8 frames per second C:NEMOI2AC3DB2NOID\_

Versión de DOSBox optimizada para ARMv7

se estaba ralentizando y aparecían saltos de imagen, aun así la prueba alcanzo una buena puntuación.

#### **Pruebas rendimiento**

Las pruebas de CPU muestra que la versión optimizada para ARMv7 funciona un poco mejor. Una mejoría de alrededor del 30% era muy común cuando se trataba de comparar la capacidad de cálculo de la CPU.

Aunque algunas pruebas de rendimiento se ejecutan mejor en la versión ARM, observé ciertos problemas importantes en varias pruebas realizadas en esta versión para ARMv7. Algunas pruebas ni siquiera se llegaban a ejecutar o tenían





La prueba 3D Bench muestra diferencias en los resultados usando la compilación ARM

un comportamiento extraño. Sólo la clásica versión de Debian ejecutaba correctamente el 100% de las pruebas.

Por ejemplo, había una prueba de memoria que usaba bloques de diferentes tamaños, se hacian algunas operaciones de modo que al final los diferentes

Circuit Analysis Benchmark Test							
Computation Speed							
Time to run a 500 frequencies, circuit analysis: 0.22 seconds.							
Performance Index							
3.50 times FASTER then a 33 Mhz 486.							
13.64 times FASTER then a 25 Mhz 386.							
41.82 times FASTER then a 16 Mhz 286.							
Press amu keu to exit.							
Prueba de rendimiento de la CPU con la							
clásica compilación							



#### La prueba 3D Bench muestra diferencias en los resultados usando la compilación ARM

bloques añadían un total de 24 MB. Funcionaba con 384 bloques de 64 KB y el resultado indicaba la rapidez con la que la memoria hacia el cálculo.

La misma prueba en la versión ARM dio lugar a resultados muy diferentes. No sólo la prueba necesitaba 10 veces más de tiempo para ejecutarse, sino que los valores fueron completamente inexactos. En lugar de aparecer 24MB como resultado, se añadieron bloques de 512 MB o más a una velocidad ridícula.

Algunas pruebas tuvieron resultados muy elevados, se salían de la escala y da-

```
e(s) 48,615 Bytes.
(s) 262,111,744 Bytes free.
```

```
        CACHEORK vit 2:7/26
        Copuright (c) 1995 by Bay Uwn Tassle. (-h for help)

        OHDS reports: conv.pees 640K, ext.pmes 31,744K, Total BMH= 32,384K

        "GeminiseIntel"
        Pettium Clocked at 1285.0 Hiz

        Reading from memory.
        Hearsy Access Block sizes (KB)-----

        1
        2
        4

        0
        3
        4
        5

        2:
        0
        8
        8
        8

        2:
        0
        0
        0
        0
        0

        3:
        4
        6
        0
        0
        0
        0

        2:
        0
        0
        0
        0
        0
        0
        0
        0
        0

        2:
        0
        0
        0
        0
        0
        0
        0
        0
        0

        2:
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0

        2:
        0:
        2:0
        2:0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        0
        <
```

#### CACHECHK sólo funciona en la clásica versión de DOSBox para Debian, identificó correctamente la CPU.

ban lugar a una velocidad negativa o con altos resultados, calculando decenas de miles de megabytes por segundo. Otras pruebas ni siquiera se iniciaron o provocaban que el emulador se colgase.

#### Herramientas

He testeado algunas otras herramientas de prueba para evaluar el rendimiento de los gráficos del sistema como el cubo giratorio y VideoDOS, a veces tenían resultados muy raros. Puesto que las pruebas gráficas son sólo indicadores de referencia y no están directamente relacionados con la capacidad de respuesta del juego, hice algunas pruebas con algunos de mis juegos favoritos.

#### Juegos

El paquete de pruebas de rendimiento incluye dos juegos, Doom y Quake. Fueron muy comunes durante la edad de oro de DOS y ofrecen buenas características para las pruebas de rendimiento en modo demo. Sin embargo, el indicador de referencia incorporado en Doom no funcionaba correctamente. Indicaba que casi siempre se ejecuta a toda velocidad, a pesar de que realmente no era así.

#### Prueba gráfica sobre la clásica compilación de DOSBox para Debian



#### **JUEGOS LINUX**



Prueba gráfica de la compilación de DOSBox ARM. Esta pruebas gráficas dieron resultados extraños: algunas pruebas parecían ejecutarse muy rápido, con más colores y en resoluciones muy altas, mientras que otras parecían normales.

Pixels 640 × 480	× 256 Colors 8	bits/pixel	
Putimare	10.280 seconds	24 AA4M nivels/second	
Ellinse	10.100 seconds	0.946M nixels/second	
Bectangle Fill	10.050 seconds	40.644M nixels/second	
Set Pixel	10.000 seconds	A A42M nivels/second	
Draw Line	10.060 seconds	0.889M nixels/second	
Get/Put Image	10.050 seconds	64.654M nixels/second	
deeriter mage		orroom practo occona	
C:\BENCH2\DOSTES	TS>		
Bi1- (40 400		14	
P1XEIS 640 X 480	x 256 Colors 8	bits/pixel	
Putimage	10.220 seconds	13.682M mixels/second	
Ellipse	10.490 seconds	0.911M pixels/second	
Rectangle Fill	10,000 seconds	18.049M pixels/second	
Set Pixel	10.550 seconds	0.042M pixels/second	
Draw Line	10.110 seconds	0.885M pixels/second	
Get/Put Image	10.210 seconds	13.811M pixels/second	
C:\BENCH2\DOSTES	TS>		

#### Resultados de VideoDOS de la versión ARM optimizada (arriba) y la clásica versión de Debian (abajo)

En lugar de usar los indicadores de rendimiento predefinidos, hice mi propia prueba y comparé el tiempo que necesitaron los juegos para ejecutar la demo por completo. Los resultados fueron muy sorprendentes: Doom 3 necesito cerca de 108 segundos para completar la ejecución de la demo por completo en la versión de DOSBox optimizada para ARMv7, mientras que en la clásica versión de DOSBox para Debian tardó 156 segundos. Eso supone un incremento de casi un 45% en la velocidad con respecto a la versión ARM.

Aún más espectacular, podría verse la diferencia con el Quake. La ejecución de la demo necesito 147 segundos en la



La versión Doom de DOS es jugable, pero no anda fina en DOSBox, funciona mucho mejor en una versión nativa de Linux.

versión optimizada, un 70% más rápido que en la clásica versión de Debian que tardo 248 segundos.

Después de las pruebas de rendimiento quise comprobar que tal funcionaba el emulador con un juego real, pronto descubrí que la configuración que yo había elegido originalmente no funciona para cualquier otro juego, así que cambié la configuración de nuevo y ejecute un par de juegos de prueba. Después afine los ciclos a 6000 en lugar de 32.000, El Dune 2 se ejecutaba perfectamente con un buen rendimiento. El sonido, la música y las voces se escuchaban bien y no tuve ningún problema.

También probé un par de juegos más exigentes, como Prisoner of Ice, que es un juego de aventura muy bueno, con algunas escenas de película y una opción para ejecutarse a una resolución de 320x240 o 640x480. La segunda incluso permitía algunas características tales como mejorar las fuentes. Ambas versiones funcionaban muy bien en DOSBox. También detecte un rendimiento similar al ejecutar Space Quest 6.

#### Resultados

La versión ARMv7 optimizada funciona significativamente mejor que la clásica versión de DOSBox para Debian. Haciendo un cálculo aproximado, la versión optimizada es de un 10 a un 15% más rápida que la versión de Debian. A veces era incluso mucho más rápida, por ejemplo cuando ejecutaba el Quake.

Los resultados más rápidos parecen estar relacionados con algunas optimizaciones matemáticas dentro del emulador que además, pueden crear problemas como un efecto secundario, especialmente con operaciones de memoria. Esto a su vez, puede causar errores en algunos juegos o evitar que se ejecuten correctamente. Además, la versión ARM-v7 optimizada es la mejor versión en términos de velocidad.

Con mi anterior prueba, puedo decir que es lo suficientemente rápido como para ejecutar Windows 3.11 o incluso Windows 95. La mayoría de los juegos

Game	Cycles	Infos	Comments
Sid Meier's Colonization	1,500-3,000	Game runs best with rather low cycles. Besides that, it's running very fine with no issues or sound drops. However, the intro on the first game start takes a long time to play through.	
Shadow Warrior	15,000-20,000	The game is laggy, and not playable	
Terry Pratchett's Discworld 1	3,000-6,000	Game ran fine without any issues	
Syndicate	6,000-10,000	Game ran fine without any issues	Does not run with glshim
Wing Commander I	2,000-4,000	Game ran fine without any issues. In my opinion, the Amiga version has a much better soundtrack	You should use a 3x scaler here
Prisoner of Ice (640x480)	2,000-8,000	Game ran fine with only a minor issue with the sound cracking occasionally	
Space Quest 6	~12,000	Game mostly runs at full speed, but has some slight stuttering in the music, and the text is too fast	
Dune 2	3,000	Game seems a little slow but generally good and without issues	
XCom Series	1,000-15,000	Works well with only slight speed issues	
Dark Legions	~20,000	Works well with only slight speed issues	

deberían funcionar bien en ambos emuladores, aunque la ejecución es algo mejor en la versión ARM optimizada.

#### Condiguración adicional

Cuando terminé con la pruebas de algunos juegos, cambié mi configuración con las siguientes valores que funcionaba muy bien con muchos juegos:

```
core=auto or dynamic
cputype=auto
cycles=fixed 3000
memsize=31
```

También descubrí que DOSBox es capaz de utilizar glshim junto con su intérprete OpenGL usando la opción "outpu":

output=opengl

Por último, cambie la configuración sdl:

```
fullscreen=true
fulldouble=true
fullresolution=1280x720
windowresolution=original
output=opengl
```

Estas opciones inician el juego en modo de pantalla completa y cuando se utilizan junto con LD\_LIBRARY\_ PATH=/usr/local/lib/, puedes ejecutar el emulador con soporte OpenGL.

#### Otros juegos

Como se puede ver en la tabla de la página anterior, los juegos varían mucho en funcionalidad, y no hay un único archivo de configuración que funcione para todos los juegos. Sugiero comenzar con un valor de ciclo de 3000 e ir aumentando el valor hasta que el juego se vuelva lento, luego volver a unos cuantos valores a atrás. Esto debería dar como resultado una óptimo rendimiento de tus juegos favoritos de DOS.

# USAR UN DOBLE MONITOR CON UN ODROID-XU3

por Justin Lee



n este artículo voy a descibir cómo utilizar dos monitores para ampliar el escritorio de un ODROID-XU3, tal y como vimos en la ARM Techcon 2014. Cada monitor puede mostrar la misma imagen o ser configurado para que aperezca imágenes distintas, permitiendo que el escritorio 1920 x 1080 sea ampliado a 3840 x 1080.

#### Actualizar el kernel

Abre la utilidad ODROID, existe un acceso directo en el escritorio de todas las imágenes oficiales de Ubuntu de Hardkernel. Si ha sido eliminado, abre una ventana de Terminal y escribe:

sudo odroid-utility.sh

Selecciona la opción 2, después la opción 1 para actualizar el kernel, necesitará unos minutos para instalarse.

#### Instalar herramientas de desarrollo

apt-get install device-tree-compiler

#### Activar DisplayPort

fdtput -t s /media/boot/exynos5422-odroidxu3.dtb /dpcontroller@145B0000 status "okay" sudo reboot

#### Configuración

Tras completarse el reinicio, observarás que las salidas HDMI y Display-Port son los mismas. Esto se llama salida "clonada", la misma imagen aparece en ambos monitores. Para conseguir que cada monitor muestre una imagen distinta, escriba lo siguiente en el Terminal:

```
xrandr --output HDMI-1 --pos
2080x0
```

Esto habilita el DisplayPort en el XU3 añadiendo un monitor por defecto de 1080p. Si por alguna razón necesitas una resolución distinta a 1080p, consulta el post correspondiente en el foro, http://bit.ly/1zdGd4m, donde se explica cómo configurar correctamente el DisplayPort para otras resoluciones.

# SO DESTACADO: CODE MONKEY

TODO EN UNO PARA DESARROLLADORES DE SOFTWARE Y HARDWARE Code Monkey para las series X y U se pueden descargar desde http://bit.ly/IrhHymu. Vigila los foros de ODROID en http:// forum.odroid.com/ para las versiones 14.04 y las de la serie XU, ¡Disponibles muy pronto!

por Rob Roy

a imagen de la comunidad Code Monkey está dirigida a desarrolladores de aplicaciones software y hardware. Basada en LXDE (Lubuntu) y con Thunar como explorador de archivos por defecto, incluye muchos de los conocidos entornos de desarrollo interactivos (IDE) y lenguajes de programación disponibles desde el Centro de Software de Ubuntu:

**Bluefish Web Editor** Code::Blocks IDE CodeLite Geany **Monkey Studio IDE Netbeans IDE** Ninja IDE **JRuby** PHP 5 **QT** 5 **Arduino IDE** Scratch Squeak Android Debug Bridge (ADB) GCC **Oracle JDK 8 Python** 

Code Monkey también incluye aplicaciones de escritorio estándar, como son Firefox y Chromium, un reproductor de vídeo (Xine), un programa de manipulación de imágenes de GNU (GIMP), el cliente de torrent Transmisión, PuTTY Telnet, el cliente SSH y una aplicación FTP de código abierto llamada Filezilla.



Hay días en que un programador se considera especial, pero hay otros días en los que va a la tienda de tatuajes y se hace un tatuaje similar a esta imagen, para humillarse a si mismo

#### Servidores

Es posible codificar tanto el back-end como front-end desde el ODROID de desarrollo. Los entornos de desarrollo de Code Monkey trabajan conjuntamente con varios servidores locales, como Tomcat, Samba, MySQL, VNC y Apache 2. Las populares herramientas de gestión de código Git y Subversion también están disponibles a través del menú del botón derecho del ráton en File Explorer.

Aunque Code Monkey arranca directamente el escritorio sin solicitar una contraseña, el nombre de usuario por defecto es "odroid" y su contraseña es "odroid". La contraseña de acceso root necesaria para los comandos "sudo" es también "odroid".

#### Bluefish

Para el desarrollo web, Bluefish Editor (http://bit.ly/1xAGg4Q) soporta muchos lenguajes web como JavaScript, Ruby y jQuery, a la vez que sirve como un gran entorno de aplicación para lenguajes más tradicionales como C++ y Python. Personalmente lo uso para escribir páginas web usando jQuery, Angular JS, CSS3, HTML5 y para mi gusto es un editor de texto rápido, simple y directo.

#### SO DESTACADO



Code monkey viene totalmente equipado para empezar a programar

#### Code::Blocks, CodeLite y Geany

Los proyectos de código abierto Code::Blocks (http://www.codeblocks. org) y CodeLite (http://codelite.org) ofrecen excelentes herramientas C++, cada uno con un entorno de trabajo ampliable que puede ser modificado y recompilado desde el código fuente. Geany (http:// bit.ly/132QrpP) también compila C/ C++ así como muchos otros lenguajes, incluidos C #, Go y Perl.

38833 5 5	1 6 6 ( <b>4 6 )</b> 0 × 5 0 0 ( <b>64119</b> 7 )   1   1 6 位化元化化 6 0 ( <b>8</b> 元
stapping, B	August Hone M L
Angente Speakets	
Quertaux.	Contraction of the second seco
	r (10) Hit Vein das Vein Ten marka (, Barring, Francous), (, Brange)
	Nachatari Manana Wangari U Mangari U Mangari U Mangari

Interfaz de aplicaciones Code Blooks

#### **Monkey Studio**

El IDE Monkey Studio está pensado principalmente para el desarrollo de Qt (http://qt-project.org), es un lenguaje en desarrollo que se centra en la compatibilidad entre plataformas. El escritorio Razor incluido en Ubuntu 13.10 fue escrito totalmente en lenguaje Qt.

#### Ninja

El IDE Ninja (http://ninja-ide.org) es un editor de código abierto que se puede usar para desarrollar aplicaciones en Python. Significa "Ninja Is Not Just Another IDE" y esta mantenido por un

,	_		Templates Wizard			,
				No Te	mplate Selec	ted
Language :	C++	0		Type :	Projects	0
Empty f C XUP Pro Q QMake ( Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q	le iect C-C++ Applica C-C++ DII C-C++ Static Lil Qt Console Qt DII Qt Gui Qt Plugin Qt Static Lib		Informations			
Add To P	roject	~				
Expert						
Destination :	droid/Documer	ts/M	onkey Studio/Projects	Codec : UTF	-8	0
Open File:	Upon Create			Cre	ate Can	cel

**Aplicación IDE Monkey Studio** 

pequeño equipo de desarrolladores apasionados. Python (https://www.python. org) es uno de los principales lenguajes usados en Linux, es muy fácil de aprender y tiene miles de módulos de terceros disponibles para su descarga gratuita desde Python Package Index.

#### Netbeans

El IDE Netbeans (https://netbeans. org) es uno de los entornos de desarrollo más completos que hay disponibles, compatible con una amplia variedad de lenguajes modernos como SASS, Java, HTML 5, PHP, C / C ++ y muchos más. También cuenta con una enorme cantidad de plugins y se puede utilizar para desarrollar grandes sitios web de calidad y aplicaciones de escritorio profesionales.

Antes de iniciar NetBeans por primera vez se debe seleccionar la máquina virtual Java y las librerías. Para ello, actualiza el archivo "/etc/netbeans.conf" para usar java-7-openjdk-armhf. También puede ser necesario instalar el paquete "openjdk-7-jre".

netbeans\_jdkhome="/usr/lib/jvm/
java-7-openjdk-armhf/"

#### **Aplicación IDE Ninja**



	NetBearstoe		
Learn & Decever	My Roteboard	What's horse	
Take a Texa	Dennis & Tudentals	Federed Derve	
Try a hample Project	Inter 16 April 14 Days	_	
What's New	Jaco III & Jeve Web Applications	-	
Commanity Corner	and applications		
	Holds and Calculated Applications		
	all define becamentation on	Value: Anywherd, and Jama HI in Reliting	
OPAGIE	Real Street Conception	4	

#### **Aplicacion IDE Netbeans**

#### Arduino

El IDE Arduino (http://www.arduino.cc) es el entorno de desarrollo ideal para domótica, robótica y sistemas de control. Permite que cualquier productos Arduino pueda ser actualizado con programas personalizados para su uso en proyectos interactivos. Tiene su propio lenguaje de programación que es similar a C++. El IDE Arduino está especialmente indicado para los ODROIDs, ya que incluye herramientas para escribir, compilar y cargar programas en los periféricos I/O Shield y Oduino (Arduino Uno) de Hardkernel.

Butters_Hunddiy_avd_Temperature   Articise 3 3.0.5 + dbg2-1	
Lie Ed Sotra Toola Hep	
	5
Butters_Humidty_and_Temperature	
Rectain standards and an	
This enormal will next the Lil park. The bottons and the Heavier terms are a second to the terms of terms of the terms of t	
Weight ( 3.3 1998 + 28-001-2013	
le Hardaniat	
27 for English Headship and Temperature memor (DHEE) det OPT Addition Optimiz, JPR 3	
22 Kitosi, sersalasi unspend Tong elaped, times	
27 select the pure used on the LCD parel Logislitythmi Matth, M. S. W. 71;	
<pre>/ distance was values and by the partial distance in the target of the partial distance are provided in the partial distance are provided in the partial distance are provided in the partial distance of the partial are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided in the partial distance of the partial distance are provided are partial distance of the partial distance are partial distance of the partial distance of the partial distance are partial distance of the part</pre>	
// well the sections	
(a) Apply a - extended in: () - () - () - () - () - () - () - ()	
recent bradde, or dea all others field, record that	

**Aplicación IDE Arduino** 

#### Scratch y Squeak

Scratch (http://scratch.mit.edu) es un lenguaje de programación educativo que enseña una metodología y una técnica de resolución de problemas a través una interfaz intuitiva y fácil de usar. Con el objeto de hacer el aprendizaje divertido, Scratch se suele utilizar para crear historias, juegos y animaciones que pueden ser compartidos con otros usuarios. Squeak (http://www.squeak.org) es una versión personalizada de Scratch que implementa el lenguaje Smalltalk, uno de los primeros lenguajes orientados a objetos.

#### SO DESTACADO



Scratch (arriba) y Squeak (derecha) son quizás las herramientas más importantes disponibles en Code Monkey. ¿Por qué? Bueno, teniendo en cuenta a nuestros jóvenes ODROIDians es quizás la imagen más importante que hemos hecho, así que estamos seguros que esta distribución es tan buena y educativa como accesible.

#### Desarrollo de Juegos

Code Monkey incluye drivers para la mayoría de joysticks y mandos como el Xbox 360, PS3 y Wiimote. Usa el programa "joytest" disponible en el menú de aplicaciones para determinar el puerto serie del dispositivo USB, al que luego puedes acceder para configurar los botones del mando y los movimientos del joystick. En http://bit.ly/1sAsmeP encontrarás una guía completa sobre la configuración y la instalación de joysticks en Ubuntu.

Los Wiimotes también son compatibles via Bluetooth y pueden ser utilizados como sensores infrarrojos o giroscópicos. Para obtener más información sobre cómo utilizar Wiimote en Linux, consulte la página 8 de la edición de junio 2014 de ODROID Magazine.



	MySQL Workbench	- * *
*		
Ele Edit View Database Blugins Scripting	Help	
Workbench Central		
Welcome to MySQL • What's New in This Release? Read about all changes is this Myte	Norkbench	Lusides Expansion Association and the second
# Workspace		
SQL Development Correct to existing databases and run SQL queries, SQL torget, edit data and manage database objects.	Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.	Server Administration Configure your database server, sing user accounts, broase status variables and server logs.
Open Connection to Start Querying Or etch & DB connection to span the SDL balance	Open Existing EER Model           Image: Stating and	Server Administration
New Connection		Rew Server Instance
Edit Table Data	Create New EER Model	Manage Import / Export Cruste a damp for or restore data from a file.
Edit SQL Script Open an evening COL Dorpt No for edites	Create EER Model From Existing Datab	Manage Security Manage see account and another prickets
Manage Connections	Create EER Model From SQL Script	Manage Server Instances Act: delete and codale server instance settings.
Ready		

#### MySQL y PostgreSQL

El Lenguaje Estructurado de Consulta (SQL) es el estándar en la industria del software para la lectura y escritura de información en base de datos, y la plataforma Linux lo soporta muy bien. Una de sus implementaciones más famosas, llamada MySQL esta preinstalada y se puede gestionar ejecutando la aplicación "MySQL Workbench" desde el menú Aplicaciones. La contraseña root de MySQL es "odroid".

PostgreSQL es otra conocida implementación de SQL para la plataforma Linux, es muy conocida por su fiabilidad. Para instalar PostgreSQL en Code Monkey, consulta la guía oficial de Ubuntu en http://bit.ly/1bsJQo4.

#### **Git y Subversion**

Tanto Git (www.github.com) como Subversion (http://subversion.apache. org) ofrecen amplias herramientas de colaboración y gestión de código para la gestión de proyectos a gran escala. Hardkernel utiliza GitHub como su repositorio de código (http://bit.ly/ZX834P). Todas sus fuentes de software y Kernels puedes ser descargadas desde GitHub con el comando de terminal "git clone".



Para obtener más información sobre cómo acceder a los repositorios de Hardkernel, visita http://bit.ly/1wb9ity.

Subversion es una suite de gestión de proyectos de software similar que permite el uso de servidores locales, en lugar de depender de un servidor en la nube para almacenar las versiones del código. A la funcionalidad de Subversion se puede acceder haciendo clic derecho en una carpeta vacía usando Thunar File Explorer.

#### Apache Web Server y PHP

Apache 2 es el estándar en la industria de los servidores web, aunque su popularidad ha sido cuestionada recientemente por el servidor liviano Nginx. Los sitios web se almacenan en "/var/www/ html/<nombresitioweb>/", también deberían añadirse a la lista de sitios web dis-

#### Git y Subversion



#### SO DESTACADO

ponibles en "/etc/apache2/sites-available/". Para acceder al sitio web visita http://127.0.0.1/<nombresitioweb>. Apache detectará automáticamente el lenguaje de la web en función de la extensión de archivos. Por ejemplo, index.html indica que la página se basa en HTML5 y index.php indica que la página está escrita en PHP.

PHP es un lenguaje muy conocido, fácil de aprender y puede usarse para escribir potentes aplicaciones web. Por su estabilidad y amplia aceptación, algunos de los sitios web con mayor tráfico de Internet están escritos en PHP como Facebook y Wikipedia. Para desarrollar en PHP, recomiendo usar los entornos Netbeans o Bluefish. Para aprender más sobre PHP, visite el sitio oficial de desarrollo en http://bit.ly/1zeFFdp.

#### **Tomcat Java Server**

Tomcat (http://bit.ly/1wzLbUz) mantenido también por la Apache Software Foundation es otro tipo de servidor web que gestiona aplicaciones basadas en Java. También usado para desarrollar robustas aplicaciones web. A menudo se ejecuta junto con Apache para consultar y guardar datos vía JSONP. Para acceder a las aplicaciones de Tomcat, visite http://127.0.0.1:8080/ <nombresitioweb> tras copiar la aplicación web en el directorio "/var/lib/tomcat7/webapps/ROOT/nombresitioweb".

#### Samba

El directorio compartido Samba (SMB) esta preconfigurado en "/ home/ODROID/Public", al que se puede acceder con los usaurios root y odroid, la contraseña por defecto es "odroid". Para configurar directorios Samba adicionales o cambiar las contraseñas, escriba lo siguiente en una ventana de terminal:

\$ sudo system-config-samba

#### Vino y Re<mark>mmina</mark>

Code Monkey incluye Vino VNC Server que se inicia en el arranque, permite el acceso al escritorio de la máquina de desarrollo de forma remota. Tienes información adicional sobre el control del servidor Vino en http://bit.ly/1wbbhOG. Para acceder a otros escritorios de la máquina de desarrollo Code Monkey utiliza la aplicación Remmina, proporciona soporte para conectarse a servidores de VNC y RDP.

#### **Desarrollo Android**

La familia de equipos de placa reducida ODROID son máquinas ideales para el desarrollo con Android, principalmente porque pueden ejecutar tanto Linux como Android. Esto permite a los desarrolladores crear apps de Android utilizando un ODROID que ejecute Code Monkey, luego subir la aplicación via Android Debug Bridge (ADB) a otro ODROID que ejecute una versión root de Android. Para más información sobre el uso de ADB con Linux visita la Wiki de ODROID en http://bit.ly/1u4L2uq.

#### ROTA FACILMENTE TU PANTALLA EN ANDROID DERROTA A TUS ENEMIGOS VIR-TUALES SIN TENER QUE GIRAR TU MONITOR

#### por Bruno Doiche

*e vez en cuando queremos jugar a un juego en nuestro* ODROID con Android, pero el juego requiere que utilices la pantalla en modo vertical como esta:

iFighter 2: The Pacific 1942 es mucho más divertido de jugar en modo vertical



Así que para hacerlo mas llevadero, sólo tiene que instalar la app **Screen Rotation Control** o **Ultimate Rotation Control** desde Google Play y ¡A disfrutar de tus juegos en vertical!



# CONSTRUIR UN VEHICULO TODOTERRENO AUTOMATICO CON ODROID

**PARTE 3: PROGRAR LA NAVEGACION GPS** 

por Christopher D. McMurrough

on este artículo, concluimos nuestra serie sobre la construcción de un vehículo todoterreno automático con odroid (VTA) usando la plataforma ODROID-XU y centrándonos en guiarlo hacia unas coordenadas GPS predefinidas, utilizando para ello los datos de navegación proporcionados por un dispositivo Android externo. Usaremos la imagen de Ubuntu 12.04 Robotics Edition para el ODROID-XU, que se puedes localizar en los foros Hardkernel en http://bit.ly/1vK6TWD.

Las partes 1 y 2 de la serie se centraron principalmente en la configuración del hardware de nuestro sistema. Esto incluía el chasis mecánico, la distribución de potencia, los controladores de los motores y el montaje de los dispositivos electrónicos. En la Parte 2 tratamos la recogida de datos y el control del motor usando el software Robotic Operating System (ROS). Desarrollamos nodos ROS para cada una de las entradas y salidas necesarias de nuestro sistema. Nuestro último artículo nos permitirá desarrollar las reglas de control necesarias para poder guiar la plataforma hacia unas coordenadas GPS específicas. Al igual que antes, seguiremos facilitando ejemplos de código desde el repositorio del proyecto en http://bit.ly/1jfykOU.

#### Navegación con Waypoint

En esta demostración, realizaremos una simple tarea de navegación en la que continuamente calcularemos la posición y la orientación actual del robot, deduciendo de esta forma los comandos de control de movimiento necesarios para mover la plataforma hacia un objetivo concreto. El "objetivo" se define como una constante en nuestro código de navegación y representa la localización GPS de un punto marcado en nuestra área de prueba, aunque podría fácilmente ampliarse con otros puntos o configurarse de forma dinámica durante la propia ejecución. Como nuestra zona de pruebas es un campo abierto, no vamos a incluir detección de obstáculos.

Nuestra simple estrategia de navegación consta de dos partes: en primer lugar, la distancia lineal entre nuestra localización GPS actual y la de destino ha sido calculada y usada para estimar una velocidad máxima de avance. Con ello se pretende disminuir la velocidad de la plataforma al acercarnos al destino de forma que evitemos rebasarlo. En segundo lugar, calculamos el ángulo entre el rumbo del robot actual (proporcionado por el subsistema de brújula del dispositivo Android) y la dirección de la ubicación del objetivo en relación a la localización del robot. Este ángulo será usado para calcular las velocidades de giro necesarias para corregir el rumbo del robot para que avance hacia la meta.

En el artículo anterior, tratamos el control del motor utilizando el microcontrolador Teensy y su nodo ROS. El nodo del controlador de motor espera un mensaje del tipo "Twist", que agrupa velocidades lineales (Vx, Vy, Vz) y velocidades de rotación sobre los ejes del robot (Rx, Ry, Rz). Puesto que nuestro



vehículo sólo puede avanzar sin poder deslizarse de izquierda a derecha, sólo necesitamos tener en cuenta una única velocidad de avance/retroceso Vx.

De igual modo y puesto que nuestro robot no puede "rodar" hacia los lados o "inclinarse" hacia arriba y abajo, sólo consideraremos una única velocidad de giro, Rz (cogeremos los puntos del eje x para avanzar y el eje z para orientarse, elegiremos por tanto Vx y Rz). Nuestro nodo ROS de controlador del motor ya ha sido diseñado para convertir los valores Vx y Rz en velocidades individuales de giro de izquierda/derecha, por lo que crearemos un nodo ROS llamado "navegación" que acepte los mensajes del GPS y de la brújula procedentes del nodo "android\_sensors\_driver" y que publique los mensajes "Twist" que espera el nodo del controlador del motor.

#### Cálculos

El nodo de navegación calculará la velocidad de avance Vx, usando la distancia entre las posiciones GPS actual y de destino. Con la solución trigonométrica planteada en http://bit.ly/1FzriC6, podemos calcular la distancia usando:

```
dlat = lat2 - lat1;
dlon = lon2 - lon1;
distance = sqrt(dlat*dlat +
dlon*dlon);
```

Una vez que tenemos la distancia, podemos calcular el ángulo usando:

```
y = sin(lon2-lon1) * cos(lat2);
x = \cos(lat1) * \sin(lat2) -
sin(lat1)*cos(lat2)*cos(lon2-
lon1);
if(y > 0)
{
      if (x > 0)
              angle =
arctan(y/x);
      else
              angle = 180 -
\arctan(-y/x);
}
else
{
      if (x > 0)
              angle = -arctan(-
y/x);
      else
              angle =
\arctan(y/x) - 180;
}
```

Una vez calculada la distancia (metros) y el ángulo (grados), podemos calcular los valores de Vx y Rz que necesitamos para nuestro mensaje "Twist". Queremos que nuestro valor Vx reduzca la velocidad de la plataforma cuando se acerque a una cierta distancia del objetivo, de lo contrario lo rebasaremos. Vamos a elegir una distancia de aproximación arbitraria de unos 20 metros. Fijaremos los límites de Vx en (-1,1), donde -1 es la velocidad máxima marcha atrás y 1 es la velocidad máxima hacia adelante. Nuestro valor Vx es calculado así:

```
if (distance > 20.0)
{
     Vx = 1.0;
}
else
{
     Vx = distance/20.0;
}
```

Esto hará que la velocidad de avance del robot disminuya a medida que se acerque al destino, una vez que esté dentro del radio de los 20 metros. Podemos entonces calcular la velocidad de giro Rz:

}

Un Rz positivo hará que el robot gire hacia la derecha, mientras que uno negativo provocará un giro hacia la izquierda. La velocidad de giro disminuye a medida que el robot corrige su ángulo de orientación, de forma similar a la forma en que la velocidad de avance disminuirá a medida que el robot se aproxime a su destino. Si estas funciones se combinan cuando el robot está en marcha, debe girar hacia la localización de destino y aplicar pequeñas correcciones a medida que va avanzando. Este es el comportamiento que queremos cuando ignoramos los obstáculos en un robot guiado. Una vez combiandas las funciones Vx y Rz en nuestro nodo ROS de navegación, ¡Estamos listos para la prueba!

#### Pruebas

Nuestro sistema fue probado en un campo abierto sin obstáculos relevantes como árboles y agujeros. Antes de iniciar una prueba, seleccionamos una localización de destino en medio del campo y medimos su posición con la tablet Android. A continuación, añadimos este valor a nuestro nodo de navegación como objetivo. Tras introducir este valor, ¡El único propósito del robot es dirigirse a la posición de destino a toda costa! Una vez que el robot detecta que su posición está cerca de la meta, la velocidad disminuye hasta el punto en que el robot no llega a moverse. Esto nos permitía girar el interruptor del motor para desconectarlo cuando alcanzara el objetivo.

Un claro inconveniente de nuestras reglas de control relativamente simples es que no manejan muy bien las imprecisiones del GPS. Los datos en grados del GPS proporcionados por la tablet Android sólo tienen una precisión de entorno a 10 metros más o menos. Esta inexactitud aumenta en presencia de árboles, nubes, etc. Cuando el robot está lejos del objetivo, esta deficiencia prácticamente no se aprecia.

El problema aparece cuando el robot se acerca a su destino, ya que los datos GPS pueden "saltar" 10 metros o más entre las mediciones. El resultado es que el robot puede pensar que de repente la posición de la meta está por detrás, provocando un giro repentino similar a la orientación inicial. De hecho, una vez que el robot está dentro de los 10 metros, su movimiento se vuelve bastante caótico. Esto podría solucionarse con un filtrado de señal más avanzado o con un GPS de mayor precisión, pero el hecho de llegar a los 10 metros está bastante bien para nuestra prueba inicial.

Otra desventaja al usar la tablet Android como único sensor de navegación es que los datos de la brújula se pueden ver afectados por campos magnéticos externos. Puesto que nuestra plataforma consta de 6 motores magnéticos, es crucial que montemos la mesa lo más lejos posible. Los elementos de metal que pueden estar presentes en el suelo también pueden causar interferencia, por lo que colocamos la tabla en un poste que se extendía aproximadamente 0,5 metros desde el chasis superior de la placa. Aunque esto no elimina completamente las interferencias magnéticas, funciono muy bien en nuestra prueba inicial.

#### Conclusiones

Con esta serie, hemos dado una visión global de cómo montar un vehículo todoterreno automático (VTA) con la plataforma ODROID-XU. Hemos cubierto los aspectos mecánicos, eléctricos y de software del sistema con el fin de proporcionar a los aficionados de los robots ODROID ideas sobre cómo montar sus propios proyectos. En nuestro siguiente proyecto nos centraremos en evitar obstáculos usando la cámara RGB-D y la capacidad de procesamiento del ODROID-XU.

# DESARROLLAR UNA APP PARA LA PLACA METEOROLOGICA CREA UN SISTEMA EN MINIATURA PARA

REDOGER DATOS METEOROLOGICOS

#### por Justin Lee

n este artículo se presenta el ejemplo de una aplicación que recibe datos procedentes de la Placa Meteorológica de Hardkernel (http://bit. ly/1wtPdgP), y los muestra en tiempo real en una pantalla táctil ODROID-VU (http://bit.ly/UmZEod) utilizando un ODROID-SHOW basado en Arduino (http://bit.ly/1wyo5MZ) como controlador integrado.

Para este ejemplo, Se utilizo un ODROID-U3 con Ubuntu 14.04 instalado como equipo de desarrollo. La app Android se puede programar usando Eclipse y el firmware de ODROID-SHOW se puede actualizar utilizando el IDE de Arduino. Para instalar Eclipse consulta http://bit.ly/1pMAAAJ. Para instalar el IDE de Arduino, escriba lo siguiente en una ventana de terminal: \$ sudo apt-get update && sudo apt-get install arduino arduinocore \$ sudo arduino

Todo el software que aparece a continuación está disponible para su descarga gratuita desde el repositorio Github de Hardkernel en http://bit.ly/1snZCG0.

#### **Descargar fuente**

La fuente original de puerto serie sobre la que se basa esta aplicación meteorológica está disponible desde Google en http://bit.ly/1zubmQX. El código fuente completo para la versión ODROID-SHOW puede descargarse escribiendo:

\$ git clone git@github. com:codewalkerster/Weather.git Crea un nuevo proyecto de aplicación para Android y llamalo "Weather". Copia el directorio "jni" y el archivo "SerialPort.java" desde el código fuente descargado en el nuevo proyecto.

Luego, copia el archivo Application. java en el proyecto, después cambia el puerto a "/dev/ttyUSB0" y la tasa de baudios a 500.000.

```
public SerialPort getSerialPort()
throws SecurityException, IOEx-
ception, InvalidParameterExcep-
tion {
    if (mSerialPort == null) {
        /* Open the serial port */
            mSerialPort = new
SerialPort(new File("/dev/tty-
USB0"), 500000, 0);
    }
    return mSerialPort;
```

Conectando al ordenador al ODROID-SHOW a través de USB





}

#### **PLACA METEOROLOGICA**



Archivos que se copian en el proyecto weather

Descarga el archivo GraphView-3.1.3.jar desde http://bit.ly/1wyuibY y cópialo en la carpeta "libs" de tu proyecto. Luego, crea el flujo de entrada de datos en serie:

```
mSerialPort = mApplication.getSe-
rialPort();
mInputStream = mSerialPort.getIn-
putStream();
```



Copiando el archivo GraphView.jar

Los datos raw están en este formato:

```
ESCw0[bmp180 Temperature] +
ESCw1[bmp180 Pressure] +
ESCw2[bmp180 Altitude] +
ESCw3[si7020 Temperature] +
ESCw4[si7020 Humidity] +
ESCw5[si1132 UV Index] +
ESCw6[si1132 Visible] +
ESCw7[si1132 IR]
```

```
...
mInputStream.read(buffer,
0, 1);
if (buffer[0] == `w') {
    int i = 0;
    while (buffer[0] !=
0x1b) {
        mInputStream.
read(buffer, 0, 1);
        ...
        switch (index) {
        ...
        case `1': //bmp180
Pressure
        String str = new
String(buf).split("\0")[0];
```

buffer) throws IOException {

```
mPressureData[mGraphX] = new
GraphViewData(mGraphX, Double.
parseDouble(str) / 100);
```

Despué de haberse rellenado la secuencia GraphViewData[] con nuevos datos, activa la función GraphViewSeries::resetData()

```
mPressureSeries.
resetData(mPressureData);
mAltitudeSeries.
resetData(mAltitudeData);
mTemperatureSeries.
resetData(mTemperatueData);
```

mHumiditySeries.
resetData(mHumidityData);
mUVIndexSeries.
resetData(mUVIndexData);
mVisibleSeries.
resetData(mVisibleData);
mIRSeries.resetData(mIRData);

#### Firmware ODROID-SHOW

Descarga el código fuente de ODROID-SHOW desde Github, luego añade las librerías tras instalar el IDE Arduino. Consulta nuestra wiki para más detalles en http://bit.ly/ZKD7UM.

```
$ sudo apt-get install git
$ git clone https://github.com/
hardkernel/ODROID-SHOW
```

Tras añadir el conector, compila el proyecto y cárguelo. Una vez en funcionamiento, el ODROID-SHOW mostrará los valores de los sensores y enviará los valores a través del puerto serie.

Para solucionar el tiempo de latencia de los valores de los sensores para la transferencia en serie, editar el archivo

Diagrama por bloques d ODROID-VU + ODROID-U3 + ODROID-SHOW + Placa meteorológica



public boolean updateData(byte[]

#### NODE.JS

"weather\_board.ino". Si cambias el tiempo de detección, asegúrate de ajustar también el valor de Timer1.initialize

```
# File: ODROID-SHOW/weather_
board/weather_board.ino
void setup() {
...
// Timer one setting
Timer1.initialize(200000); //
200ms
Timer1.
attachInterrupt(timerCallback);
}
```

#### Lecturas recomendadas

Wiki ODROID-SHOW http://bit.ly/1toe7Pl

Wiki Placa Meteorologica http://bit.ly/ZKD7UM

#### Resultado de la aplicación de la Placa Meteorológica sobre un ODROID-VU



Una Placa meteorológica puede ayudarte a predecir la humedad y evitar que tu pelo tenga un mal día



# COMO INSTALAR NODE.JS

UNA MODERNA PLATAFORMA DE APLICACION WEB BASADO EN JAVASCRIPT

editado por Venkat Bommakanti



ode.js una plataforma liviana y eficiente, ideal para aplicaciones que gestionan muchos datos en tiempo real. Está basada en el lenguaje JavaScript en tiempo de ejecución de Chrome, orientada a eventos es un modelo E/S libre de bloqueos para aplicaciones web. Este artículo describe el proceso de instalación de Node.JS sobre la ODROID-U3.

#### Requisitos

1. Cualquier placa ODROID, con el adaptador de corriente apropiado.

2. Una tarjeta MicroSD 8GB o Módulo eMMC de arranque con la última imagen Lubuntu para el U3 disponible desde el sitio web Hardkernel.

3. Acceso por SSH al U3 opcional a través de utilidades como PuTTY (MS Windows 7 +) o Terminal (Mac, Linux).

#### Actualizar el sistema

Para empezar, instala el entorno y las herramientas de desarrollo esenciales con el comando apt-get. Todos los comandos que se muestran de aqui en adelante deben escribirse en una sola línea.

\$ sudo apt-get install python \
g++ make auto-apt checkinstall \
fakeroot build-essential

#### Conseguir el último código fuente

Cuando escribí estas líneas, la última versión de Node.js era la 0.10.32. Los pasos que se indican a continuación pueden tener que ajustarse para las versiones más recientes. Para instalar la versión de Node.js simplemente escriba lo siguiente e ignora la sección "Create Test Sample":

sudo apt-get install nodejs

Para instalar la última versión de nodo.js, crea los siguientes subdirectorios en el directorio de inicio y dirígete el directorio src:

```
$ mkdir nodejs && cd nodejs/
$ mkdir src && cd src/
```

Consigue el último tar-ball con el código fuente y descomprimirlo, usando los siguientes comandos:

```
$ wget -N http://nodejs.org/dist/
node-latest.tar.gz
$ tar xzvf node-latest.tar.gz
$ cd node-v0.10.29/
```

#### **Compila el paquete**

\$ sudo auto-apt run ./configure
--without-snapshot

Entering auto-apt mode: ./configure --without-snapshot Exit the command to leave auto-apt mode.

```
{
```

`target defaults': { `cflags': [] 'default\_configuration' 'Release', `defines': [] `include dirs': [], `libraries': [] }, 'variables': { `arm fpu': `vfpv3', `arm neon': 0, `armv7': 1, `clang': 0, 'gcc version': 48, `host arch': `arm', `node install npm': `true', 'node prefix': '' `node shared cares': `false', 'node shared http parser': 'false', 'node shared libuv': 'false', `node shared openssl': `false', 'node shared v8': 'false', 'node shared zlib': 'false', `node tag': `', `node unsafe optimizations': 0, `node use dtrace': `false', `node use etw': `false', 'node use openssl': 'true', `node use perfctr': `false', `node use systemtap': `false', 'openssl no asm': 0, 'python': '/usr/bin/python', `target\_arch': `arm', 'v8 enable gdbjit': 0, 'v8 no strict aliasing': 1, 'v8 use arm eabi hardfloat' 'true', 'v8 use snapshot': 'false', 'want separate host toolset': 0}} creating ./config.gyp creating ./config.m

# the command below should be
typed on a single line

```
$ sudo fakeroot checkinstall -y
--install=no \
--pkgversion $(echo $(pwd) | sed
-n -re's/.+node-v(.+)$/\1/p')
make -j$(($(nproc)+1)) install
```

[...]

Done. The new package has been saved to node-v0.10.32/ node\_0.10.32-1\_armhf.deb You can now install it in your system anytime using:

dpkg -i node\_0.10.29-1\_armhf.deb

#### Instalar el paquete

En primer lugar, comprueba que el paquete de Debian fue creado correctamente escribiendo:

```
$ ls -ltr
...
-rw-r--r-- 1 root
root 3688430 Jul 30 15:03
node_0.10.29-1_armhf.deb
```

\$ sudo dpkg -i node\_0.10.29-1\_
armhf.deb

Si todo ha ido bien, puedes saltarse esta sección. En caso de tener problemas con el proceso anterior, el archivo .deb puede compilarse manualmente escribiendo los siguientes comandos:

```
$ sudo apt-get install python g++
make checkinstall fakeroot
$ src=$(mktemp -d) && cd $src
$ wget -N http://nodejs.org/dist/
node-latest.tar.gz
$ tar xzvf node-latest.tar.gz &&
cd node-v*
$ ./configure
# the command below should be
typed on a single line
$ sudo fakeroot checkinstall -y
--install=no \\
--pkgversion $(echo $(pwd) | \
sed -n -re's/.+node-v(.+)$/\1/p')
make -j$(($(nproc)+1)) install
$ sudo dpkg -i node *
```

Después, comprueba la instalación manual usando los comandos:

```
$ which node
/usr/local/bin/node
$ node --version
v0.10.32
```

#### Crea un ejemplo de prueba

Crea un subdirectorio y ejecuta el archivo JavaScript:

\$ mkdir sample

- \$ cd sample/
- \$ touch hello-world.js
- \$ chmod +x hello-world.js
- \$ medit hello-world.js

Añade el siguiente contenido al archivo JavaScript de ejemplo:

```
var http = require('http');
http.createServer(function (req,
res) {
  res.writeHead(200, {'Content-
Type': 'text/plain'});
  res.end('Hello ODROID
World\n');
}).listen(8090, 'your-u3-ip-ad-
dress');
console.log('Server run-
ning at http://your-u3-ip-ad-
dress:8090/');
```

Ten en cuenta usar el puerto 8090 en este caso.

#### Probar el ejemplo

En una ventana de terminal, inicia Node.js con el archivo JavaScript:

\$ node hello-world.js

Para comprobar que la instalación funciona correctamente usa otro dispositivo en red local como un PC, tablet o ODROID-U3 y dirige un navegador a la URL http://<direccionipU3>: 8090.

Para obtener información adicional o realizar preguntas, vista el sitio web de Node.js en http://nodejs.org.

# OPEN MEDIA VAULT SISTEMA DE ALMACENAMIENTO EN RED DE CODIGO ABIERTO

PARA DEBIAN **GNU/LINUX** 

por Venkat Bommakanti

uando queremos crear una solución de almacenamiento en red (NAS), Open Mdia Vault es una opción muy común en los contextos de pequeñas oficinas en casa (SOHO). Permite que los usuarios de una red común puedan acceder a los archivos compartidos. Open Media Vault ofrece servicios y módulos, tales como:

- BitTorrent
- Secure Shell (SSH)
- File Transfer Protocol (T/FTP)
- Network File System (NFS)
- Samba (SMB)
- Common Internet File System (CIFS)
- Lightweight Directory Access Protocol (LDAP)
- Uninterruptible Power Supply (UPS)
- Digital Audio Access Protocol (DAAP) Servidor multimedia
- Sincronización remota (Rsync)
- Utilidades de administración web

Este artículo describe el proceso de instalación de Open Media Vault sobre un ODROID-U3. Los pasos también se puede aplicar a un ODROID-XU3.

#### Requisitos

1. Una placa ODROID-U3, con un adaptador de alimentación apropiado

2. Una Tarjeta MiscroSD clase 10 (con un adaptador SD a USB) con la The open network attached storage solution

última imagen de escritorio Debian Wheezy específica para U3.

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

Se necesita un servidor web para alojar la interfaz gráfica de usuario basada en web del Open Media Vault. Instala una versión básica de nginx utilizando los siguientes comandos. Se puede realizar una instalación de nginx más completa usando los pasos descritos en el artículo de la página 25 de la edición de agosto 2014 de ODROID Magazine.

\$ sudo	apt-get	instal	ll nginx
\$ sudo	service	nginx	stop
\$ sudo	service	nginx	start

#### Imágenes disponibles

Visita http://bit.ly/1zwjhxc para ver una lista de las imágenes Open Media Vault disponibles:

Name	Last modified
omnius/	29-Apr-2012
	23:08
ix/	15-May-2012
	21:04

omnius-proposed/	09-Sep-2012
	00:45
fedaykin/	25-Feb-2013
	21:05
sardaukar/	20-May-2013
	16:34
kralizec/	11-Jan-2014
	21:32
sardaukar-proposed/	16-May-2014
	23:25
kralizec-proposed/	16-May-2014
	23:25

Para este ejemplo, vamos a utilizar la última imagen Kralizec, que seleccionaremos para su instalación escribiendo:

- \$ cd /etc/apt/sources.list.d/
- \$ sudo touch openmediavault.list
- \$ sudo medit openmediavault.list

Agrega la siguiente línea en el archivo openmediavault.list y guárdalo:

deb http://packages.openmediavault.org/public kralizec main

#### Instalar paquetes

Instala los paquetes con los siguientes comandos, ignorando cualquier error o advertencia:

\$ sudo apt-get update

```
Get:1 http://packages.openmedi-
```

4. Acceso SSH al U3 a través de utilidades como PuTTY (MS Windows 7 +) o Terminal (Mac, Linux)

#### Instalar nginx

#### **OPEN MEDIA VAULT**

avault.org kralizec Release.gpg [181 B]

Get:2 http://packages.openmediavault.org kralizec Release [9,696 B]

Ign http://packages.openmediavault.org kralizec Release ....

Get:3 http://packages.openmediavault.org kralizec/main armhf Packages [6,678 B] Ign http://packages.openmediavault.org kralizec/main Translation-en\_GB

Ign http://packages.openmediavault.org kralizec/main Translation-en

• • •

Reading package lists... Done W: GPG error: http://packages. openmediavault.org kralizec Release: The following signatures couldn't be verified because the public key is not available: NO\_ PUBKEY 7E7A6C592EF35D13

#### Instalar keyring

Instala el archivo de claves de Open Media Vault con el siguiente comando:

\$ sudo apt-get install openmediavault-keyring postfix

El proceso de instalación te pedirá información adicional:

Mail Server: Local Only Name: <nombredehostdetuU3> Mail-recipient: odroid Synchronous mail updates: No

Ejecuta el siguiente comando para actualizar la instalación:

\$ sudo apt-get update
...
Get:1 http://packages.openmedi-

avault.org kralizec Release.gpg [181 B]

- . . .
- Hit http://packages.openmedi-

avault.org kralizec Release

Hit http://packages.openmediavault.org kralizec/main armhf Packages

• • •

. . .

Ign http://packages.openmediavault.org kralizec/main Translation-en\_GB Ign http://packages.openmediavault.org kralizec/main Translation-en

#### Instalar Open Media Vault

Instala Open Media Vault usando el comando:

\$ sudo apt-get install openmediavault

El proceso de instalación le pedirá información adicional:

Run Samba as: Daemon Install beep as: Usable for all (users, not just root) MD arrays: All Monthly mdam checks: Yes MD monitoring daemon: Yes MD events notification: odroid resolve.conf: No Reboot: No Run ProFTPD as: Standalone Quota reminders: Yes Over quota notification: odroid@<your-u3's-hostname> Phone #: <blank> Watchdog module: None Smart watchdog @bootup: Yes Restart watchdog on updates: No

Inicia Open Media Vault y luego reinicia usando los siguientes comandos:

\$ sudo OpenMediaVault-initsystem
\$ sudo reboot

La instalación se puede actualizar con la página de gestión de actualizaciones.



Página de acceso a Open Media Vault

#### Acceder a Open Media Vault

Tras completarse el reinicio, accede a la instalación de Open Media Vault escribiendo en el navegador Iceweasel http://<direcciónIPdetuU3>, debe mostrarse la página de inicio de sesión de Open Media Vault. El ID de usuario por defecto es "admin" con la contraseña "openmediavault".

Por defecto, se utiliza el puerto estándar 80, pero Open Media Vault puede ser configurado para usar un puerto diferente si hay otro servidor web en la máquina. Tras iniciar sesión, aparece la página de información del sistema y la funcionalidad de red puede ser probada navegando hasta la página de interfaces de red, como se muestra en las capturas de pantalla que aparecen a continuación.

La página de administración web del sistema de información de Open Media Vault

File Edit View History Bookm	OpenMediaVault web ac arks Tools Halo	Imii	nistration i	nterface • I	a3+3 • Ice	wease	d .	- * *
OpenMedia\ault web administra								
						• 67	Second Second	۵ 🕹 🙆
Search: C b o Dookmarks Toolbar ♥ Dookmarks Menu		20	diaVa	ult				=.
P Becently Bookmarked Recent Tage			Overview	Processes	Sugar	Report		-
Mazilla Firefox     Index of /public/dets     Melcome to more)     Unserted Bookmarks	A for yourse     A for the systems     A for the systems     A for the systems     A for the system of the sy	A	Hostname Version Processor Kernel System time Uptime Load average CPU usage		1010 0 (1010) 1013 3 (1010) 1013 3 (1010) 1013 3 (1010) 8 4 Lio 3 (1010) 8 4 Lio 3 (1010) 8 4 Lio 3 (1010) 8 4 Lio 3 (1010) 1010 0 (1			
	VC Bysteen Information ■ Systeen Logs ≪ Services ©■ information © Durase ? Support	C I						

La página de interfaces de red del sistema de información Open Media Vault

	Opena	sediava	off web administ	ration interrace -	.03-3 -	ceneare					
File Edit View History Bookm	arks jools	Help									
🗐 OpenMediaVault web administr	a 🔶										
③ 192.168						<b>☆</b> ♥ (?)	🔂 🛩 Go			Q	1
	· 0	System   1	Nerwork								=-
C System	Gene	ral inte	erfaces DIES Server	Service Discovery	Hours	Fireval					
e <sup>O</sup> General Settings	+ 44	d Pice	Querery X Delen								
Oce 6 Time	Name	Metod	Address	Tetmask	Garrow		MAC and	***	MTU	Link.	
TT Relevant	ettő	Pot-	Pv4: 292.168	Pv4:255,255,255,0	Pv4:15	2.568	18.64	36.26	1500	1000	
Controllers		Pv6-	P.4.	Pv6:-	Pv6-						
3. Pugins											
Prysical Disks     Prysical Disks     SMART     SMART     File     File Systems	L										
Om Access Fights Management											
L Uber											
It Group				-							
C Shared Polders	- H -	Page	1   of 1   > >    -	2						Displaying bea	is 1-1 of 1

## CONOCIENDO A UN ODROIDIAN ALEXEY GUSEYNOV

#### (@KIBERGUS): UN EXTRAORDINARIO INGENIERO DE SOFTWARE

editado por Rob Roy

Por favor, háblanos un poco sobre ti.

Mi nombre es Alexey Guseynov. Soy ingeniero de software en el gigante buscador ruso Yandex, donde trabajo con el equipo Yandex.Maps. Uso Linux desde hace más de 10 años, así que cuando me enteré de las placas ODROID de Hardkernel y me di cuenta que son casi tan potentes como mi viejo escritorio, no pude evitar comprar unas cuantas. A partir de ese momento, los ordenadores de casa han superado en número a las personas con un margen inalcanzable.

#### ¿Cómo fueron tus inicios con ordenadores?

Debo agradecérselo a mi abuelo, que insistió en que sus nietos debían tener ordenadores a mediados de los 90. Durante algún tiempo lo usé para los juegos, pero siempre estuve interesado en conocer cómo funcionan las cosas, así que mi hermano y yo desmontábamos constantemente muchos dispositivos. Como resultado, adquirí algunas habilidades básicas sobre programación durante 8º grado. Después de eso, me trasladé desde la escuela normal a una escuela de tecnologías de la información, donde los profesores me dieron una muy buena base de conocimientos.

#### ¿Cuál es tu ODROID favorito?

Mi ODROID favorito sigue siendo el



U2. Comprendo que el U3 tiene un diseño mucho más práctico, es más pequeño, incluye GPIO y también es más barato, y los modelos de la serie XU son mucho más potentes. Sin embargo, el U2 tiene un aspecto elegante, sólido y completo. A diferencia del resto de placas que he visto, la PCB básica del U2 da la impresión de elementos de diseño innovadores en lugar del típico enfoque de "por qué gastar dinero en una caja". Me gusta mucho cuando las meras decisiones de diseño técnico dan como resultado cosas realmente buenas.

#### ¿Qué tipos de contribuciones has hecho a los esfuerzos de desarrollo del XBMC?

La historia es bastante simple. Mi receptor de satélite está situado en el armario de una antesala. Pero yo quería ver la televisión en el lado opuesto de mi apartamento. Mientras que los esfuerzos de otros desarrolladores se concentraron en la reproducción de vídeo progresivo, siempre me mantuve fiel al hecho de que el vídeo entrelazado necesitaba algún procesamiento especial. Yo era el chico que siempre se quejaba de que MFC no funciona bien.

También hubo momentos en los que un error de los drivers Mali impidió que el XBMC funcionara correctamente, se creía que la causa estaba en el MFC. Por aquel entonces, @OverSun aún no se había ganado su gran reputación como mago del MFC. Así que, realice algunas pruebas y confirmé que el código de @ OverSun funcionaba muy bien y que el problema estaba en el subsistema de vídeo. ¡Lo curioso es que nunca veo programas de televisión!

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

No tengo ninguna afición especial, pero disfruto probando nuevas actividades. Tengo una bicicleta con la que paseo todo el año, siendo aún más divertido en invierno. Cuando hay bastante nieve, me gusta esquiar a campo través y cuesta abajo. Además, mi esposa es especialmente aficionada al esquí. También he saltado en paracaídas. Después de saltar se te pide que cuente "341", "342", "343" y luego tiras de la anilla.

Me quedé atónito al olvidar cómo se contaban los números de tres cifras y al mismo tiempo pensar rápidamente como encontrar una solución. Este verano, diez años después de que mi padre me enseñara a bucear con el equipo de buceo, pasé el programa de certificación PADI (Asociación Profesional de Instructores de Buceo).



Alexy está usando su fiel U2 como un centro de audio. Lo ha conectado en red para usarlo como una tarjeta de sonido universal

¿Está involucrado con otros proyectos no relacionados con ODROID?

He estado centrado en Open Street Maps (OSM) y he pasado mucho tiempo caminando por las zonas cercanas a mi casa. Pero ahora, los datos OSM están muy bien detallados, es difícil encontrar un lugar sin identificar. Así que centre mi atención en la electrónica y como resultado, las luces de mi apartamento están controlados con Arduino.

Hace dos meses recibí una carta de una empresa de crowdfunding ("¿No comprarías un kit con un superconductor de alta temperatura?") Yo no me pude resistir. Así que ahora, estoy construyendo un tipo de máquina de Rube Goldberg que utilizará un carruaje en suspensión como una de sus fases.

### ¿Qué tipo de innovaciones de hardware te gustaría ver en futuras placas Hardkernel?

Prefiero dispositivos silenciosos, así que voto por disipadores más grandes y más eficaces en ODROIDs. Entiendo que aumentaría el coste pero no obstante, sería gastar dinero en un disipador sin ventilador.

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

Durante mi primera lección de programación, mi profesor nos sugirió que jugáramos a un juego llamado "Artista chiflado". Necesitas urgentemente hacer un dibujo y tu única opción es llamar por teléfono a tu amigo (un artista chiflado con una mente desequilibrada y con amnesia), y le pides que lo dibuje. Siempre empiezas con: "Hola, por favor lee un manual de dibujo", luego le das órdenes simples. Si te equivocas en algo, te insulta, cuelga el teléfono y se olvida de todo. Fue muy divertido e interesante.

No conocíamos los bucles, las expresiones if, las variables, pero empezamos a escribir programas. Queríamos conocer cómo parpadean las luces de un coche que habíamos dibujado. Así conocimos los bucles y otras técnicas de programación.

Mi consejo es que busques un problema interesante que desees resolver y luego aprender las habilidades que necesitas para completar la tarea. De esta forma, tendrás una mayor motivación porque entiendes cómo utilizar las cosas que aprendes.

#### Alexey y su esposa en las montañas Mallorcian



PUBLICIDAD

















Big excitement, small packages Thrill your inner geek

ODROIDS AHORA ESTA DISPONIBLE EN LOS ESTADOS UNIDOS WWW.AMERIDROID.COM ENVIOS ECONOMICOS