Comunicación I2C entre U3 y Arduino • Emulando la PSP en Linux



Magazine

LLAMATIVO, RAPIDO Y ENFURECIDO

EL ODROID-XU3, ¡UNA HETEROGENEA MAQUINA MUTI-PROCESO DE OCHO NUCLEOS!



ODROID SMART POWER USO Y ANALISIS DE PROTOCOLO

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

El equipo Hardkernel hizo este mes un viaje a Santa Clara, California para estar prente en la convención ARM Techcon. En el stand se podían encontrar varias muestras del hardware actual, incluyendo el XU3, U3 y VU. Mauro y Justin crearon una sorprendente versión de Ubuntu 14.04 que ejectuba una máquina virtual Android

a través de KVM. Puede ver una foto en el artículo del ARM Techcon en la página 37.

También tuvimos un divertido concurso en la planta de exhibición para ver si alguien nos podía vencer a Angry Birds. ;Un participante fue con un completo kit U3 y consiguió 3 estrellas con un solo pájaro!

Gracias a todos a los que se detuvieron en el stand para compartir su afición por ODROIDs. Gran parte de la feria se centró en elementos que se llevan encima, tales como relojes inteligentes y micro-controladores. La poderosa combinación de la innovación del hardware de Hardkernel y el soporte para software libre (incluyendo Ubuntu 14.04 y KitKat 4.4.4), junto con el compromiso hacia el código abierto siempre que sea posible, hacer únicos los ODROIDs.

En este número, contamos con tutoriales de software para varios productos de Hardkernel, incluidos Arduino ODUINO y el Smart Power. Nanik también continúa con su serie de artículos de Desarrollo Android con un artículo sobre la creación de su primera aplicación personalizada de Android, un juego TicTacToe.

También tenemos una gran comparativa del PPSSPP (un emulador de Play-Station Portable) ejecutándose en Android y en Linux, instrucciones paso a paso sobre cómo logar que tus mandos inalámbricos Xbox 360 funciones con el emulador RetroArch y consejos para conseguir que tu favoritos juegos MSX se ejecuten de nuevo. Jussi una vez más encuentra una forma de combinar ODROID con el aire libre, y muestra cómo configurar el monitor de escritorio Conky Harmattan en Ubuntu para mostrar el pronóstico del tiempo local.

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, 43I-8I5 • 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 I35 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 de-

de aplicaciones web para sarrollo 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, vo 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 Artistico

Consiguió sus habilidades informáticas después de lograr que una

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

Nic Ec An Nic

Nicole Scott, Editor Artistico

Nicole es una experta en Producción Transmedia y Estrategia

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



Manuel Adamuz, Editor Español

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 la ecnología 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, y a veces participo en competiciones semiprofesionales.

INDICE



INTER-CIRCUITOS INTEGRADOS DE COMUNICACION (I2C)

ESTABLECER UNA CONEXION ENTRE U3 Y ARDUINO

por Bennyamin Bergelson



n este artículo se describe una forma de establecer comunicación I2C entre un ODROID-U3 y un módulo Arduino. Se detalla qué se debe hacer, tanto a nivel de hardware como de software para que el intercambio de información se realice correctamente. Además, se comparan brevemente los protocolos de comunicación más populares que existen en la actualidad, y explicaré qué tipos de datos se pueden transferir entre los dos módulos.

Protocolos de Comunicación

Existen varios protocolos de comunicación que permiten a los diferentes módulos electrónicos comunicarse entre sí. No me voy a detener en todos, pero si me voy a centrar en los protocolos UART, SPI y I2C:

UART

- Adecuado para la comunicación entre sólo dos dispositivos (un bus para dos dispositivos)

- Baja velocidad de transferencia (9600-115200 bits por segundo, a veces puede llegar hasta los 230.400 bps)

- Los dispositivos deben "hablar" a la misma velocidad (baudios)

- Sólo un dispositivo debe transmitir, de lo contrario los dispositivos recibirán incoherencias.

SPI

- Múltiples dispositivos "esclavos". - Requiere una línea SS separada

para cada dispositivo esclavo.

- Funciona a velocidades extremadamente altas (millones de bytes/seg.)

- Controles maestros de todas las comunicaciones (los esclavos no pueden hablar directamente entre sí).

- Las comunicaciones deben estar definidas de antemano, no es posible enviar datos aleatorios cuando se quiera.

12C

-Múltiples dispositivos "esclavos", hasta 1.008 dispositivos esclavos.

- Sistema multimaestro.

- Dispositivos pueden comunicarse a 100 kHz o 400 kHz.

- Sólo requiere dos cables.

Puesto que necesito que todos mis módulos se comuniquen mediante un bus común, sin tener que añadir cableado adicional para agregar nuevos dispositivos al sistema, me decante por el I2C. A continuación voy a explicar cómo establecer la comunicación I2C bidireccional entre mis módulos favoritos.

Conexión de Hardware I2C

Ambos módulos disponen de pines dedicados para la comunicación I2C: SDA, SCL, VCC, GND.



El asombroso periférico Arduino para U3

Pines Arduino

A4 (SDA), A5 (SCL), 5V, GND

Pines U3

pin 1: SCL (gpio199) pin 3: SDA (gpio200) pin 2: 1.8v pin 7: GND

Para establecer la conexión entre el Arduino y el U3, debemos conectar en primer lugar los pines correspondientes. Por ejemplo, el SDA de Arduino debe estar conectado al SDA del U3 y así sucesivamente. Sin embargo, no es posible conectar los pines directamente porque Arduino necesita 5 voltios y el U3 1,8 voltios. Así que, es necesario disponer de un convertidor de nivel lógico.

He probado el "Logic Level Converter BiDirectional" (http://bit.ly/1puPKJl) y el "PCA9306 Level Translator Breakout" (http://bit.ly/1BdMyZf), ambos funcionaron muy bien. Finalmente me decanté por el modelo PCA9306, puesto que ofrece un intérprete de nivel de tensión I2C dedicado.

Conexión del Software I2C

Antes de profundizar en ejemplos de código, vamos a explicar qué tipo de datos son válidos para transmitirse entre los módulos. Es importante entender que si un módulo envía datos y el otro no puede decodificarlos, los datos son desechados directamente por el receptor.

Cuando hablo de tipos de datos válidos quiero decir que su representación binaria y tamaño son los mismos en ambos módulos. Con una pequeña adaptación, puedes ejecutar el código http://bit.ly/YeO2VW en Arduino y ver el tamaño de las variables.

Tras comparar el resultado en ambos módulos, puedes ver que las variables como char, long y float tienen el mismo tamaño. Los tipos de datos "char" y "long" son simples variables enteras que sólo ocupan unos pocos bytes, así que seguramente puedan pasar sin ningún problema, pero ¿qué pasa con float?

Los números punto flotante se dividen en dos partes: mantisa y exponente. Si los tamaños (en bits) de estas partes son diferentes, es imposible enviar números punto flotante aunque tengan el mismo tamaño. No encontré la cantidad de bits que se asignan a la mantisa en un ATmega328P, así que simplemente compare la representación hexadecimal de las variables en ambos módulos usando este método:

printf("%08lx",
 (long)*((long*)&float number);

Puesto que las representaciones en ambos módulos eran similares, llegué a la conclusión que sería seguro pasar números flotantes de igual forma.

Hay dos métodos para enviar varios números. El primero es enviarlos uno a uno, y el otro es organizarlos en un "struct" y éste enviarlo como una única pieza. Si decides utilizar la segunda opción, debe asegurarte de que el relleno en ambos módulos es el mismo.

Como ejemplo, observa el siguiente

planteamiento de estructura de datos:

struct S { char c, long l; };

Los "structs" en arduino no tienen relleno, así que el tamaño de este "struct" es de 5 bytes (1 byte (char) + 4 bytes (long)). Si compruebas su tamaño en un U3 usando sizeof (S); verás que su tamaño es de 8 bytes. De modo que, necesitas envolver la definición con las siguientes funciones para compensar:

```
#pragma pack(push, 1)
:
#pragma pack(pop)
```

Puedes leer más sobre la alineación de estructura de datos en http://bit. ly/1pfjV8m. A continuación se muestra como ejemplo una parte del código del software que escribí. En el siguiente ejemplo, el U3 actúa como maestro y el Arduino como esclavo. El maestro envía un "struct" al esclavo, el esclavo lo muestra en una pantalla LCD y envía de nuevo el "struct" al maestro. Adjunto una pequeña pantalla LCD para validar que los números pasados se recibieron correctamente. Basta con configurar el dispositivo Arduino como esclavo pasando su dirección al método "Wire.begin ()" usando el siguiente código.

Código Arduino

```
#include <Wire.h> #include <Lig-
uidCrystal I2C.h>
#define I2C_ADDRESS 0x09
LiquidCrystal I2C lcd(0x27,20,4);
// set the LCD address to 0x27
for a 16 chars and 2 line display
struct Numbers {
char c; long l; float d;
};
Numbers numbers = \{0\};
int isNewData = 0;
int bytes to read;
int buf size;
char str[100];
char double_str[100];
void setup() {
lcd.init(); // initialize the lcd
```

```
lcd.backlight(); lcd.clear();
Wire.begin(I2C ADDRESS); // Start
I2C Bus as a Slave (Device Number
Wire.onReceive(receiveEvent); //
register event
Wire.onRequest(requestEvent); //
register event
}
void loop() {
if (isNewData) {
lcd.clear();
lcd.setCursor(0,0);
sprintf(str, ``%02x : %d",
numbers.c, numbers.c);
lcd.printstr(str);
lcd.setCursor(0,1);
sprintf(str, ``%08lx : %ld",
numbers.l, numbers.l);
lcd.printstr(str);
lcd.setCursor(0,2);
sprintf(str, ``%08lx : %s",
(long)*((long*)&numbers.d),
dtostrf(numbers.d, 0, 3, double
str));
lcd.printstr(str);
lcd.setCursor(0,3);
sprintf(str, "%d : %d", buf size,
bytes to read);
lcd.printstr(str);
isNewData = 0;
}
delay(100);
}
// function that executes whenev-
er data is received from master
// this function is registered as
an event, see setup()
void receiveEvent(int howMany) {
byte * buf = (byte*)&numbers;
buf size = sizeof(numbers);
for (int i = 0; i < buf size;</pre>
++i) {
buf[i] = Wire.read();
}
bytes to read = howMany;
isNewData = 1;
}
// function that executes when-
ever data is requested by master
// this function is registered as
```

ARDUINO I2C

an event, see setup()
void requestEvent() {
Wire.write((byte*)&numbers,
sizeof(numbers));
}

A continuación, actualiza el kernel U3 a la última versión (ver http://bit. ly/1rhz52C). Luego, abre el terminal en modo superusuario y escribe:

```
modprobe gpiopca953x
modprobe i2cgpiocustom
bus0=4,200,199
```

Ahora, el U3 puede utilizar los pines (200,199) en el cabezal de 8 pines para la comunicación I2C. Ejecute el código compilado con privilegios de root, de lo contrario el sistema bloquea el acceso al dispositivo I2C.

Ejemplo de Código

#include <unistd.h> #include <errno.h> #include <stdio.h> #include <stdlib.h> #include <linux/i2cdev.h> #include <sys/ioctl.h> #include <fcntl.h> #include <string.h> #define SLAVE_ADDRESS 0x09 #pragma pack(push, 1) // exact fit no padding struct Numbers { char c; long l; float d; }; #pragma pack(pop) // back to whatever the previous packing mode was const char * i2cDevName = "/dev/ i2c4"; int main() { // Open up the I2C bus int file = open(i2cDevName, 0 RDWR): if (file == 1) { fprintf(stderr, "Bad device name %s\r\n", i2cDevName); exit(1); } else {

```
printf("*** Device '%s' opened
successfully.\r\n", i2cDevName);
}
// Specify the address of the
slave device
if (ioctl(file, I2C SLAVE, SLAVE
ADDRESS) < 0) {
fprintf(stderr, "Failed to ac-
quire bus access '%x' and/or talk
to slave\r\n", SLAVE ADDRESS);
exit(1);
} else {
printf("*** Acquired bus access
to a slave device adr: %x.\r\n",
SLAVE ADDRESS);
}
Numbers numbers;
numbers.c = 117;
numbers.1 = 876543210;
numbers.d = 1234.567;
printf("*** Send to the i2c
bus.r^n;
printf("numbers.c = d.\r\n'',
numbers.c);
printf("numbers.l = %ld.\r\n",
numbers.1);
printf("numbers.d = %lf.\r\n",
numbers.d);
// Write a byte to the slave
if (write(file, &num-
bers, sizeof(numbers)) !=
sizeof(numbers)) {
fprintf(stderr, "Failed to write
to the i2c bus adr: %x.\r\n",
SLAVE ADDRESS);
exit(1);
} else {
printf("*** Wrote to the i2c bus
adr: %x.\r\n", SLAVE_ADDRESS);
}
// Read a byte from the slave
Numbers n = \{0\};
if (read(file, &n, sizeof(n)) !=
sizeof(n)) {
fprintf(stderr, "Failed to read
from the i2c bus.r^n;
exit(1):
} else {
printf("*** Read from the i2c
bus.r^n; printf("n.c = %d.
r\n", n.c);
```

```
printf("n.l = %ld.\r\n", n.l);
printf("n.d = %lf.\r\n", n.d);
}
close(file);
return 0;
}
```

Lectura Adicional

I2C http://bit.ly/1rnjxaX Comunicación Serie http://bit.ly/1v6gHtp SPI http://bit.ly/1v6gJ4w Tamaño de tipos de datos http://bit.ly/1DAyPzs Alineación de estructura de datos: http://bit.ly/1ytfn7T

OOO KEEP CALM AND USE ARDUINO

Este robot conoció el nuevo XU3



DESARROLLO ANDROID CREAR UNA APLICACION ANDROID PERSONALIZADA

por Nanik Tolaram

n este artículo, voy a tratar el proceso de creación de tu propia aplicación Android, además de cómo configurar un entorno de desarrollo para Android. Recomiendo empezar con una nueva instalación de Ubuntu 14.04 de 64 bits, aunque puedes utilizar cualquier distribución de Linux que te sea más cómoda, siempre y cuando dispongas de todas las herramientas de desarrollo necesarias. Si quieres probarlo primero, también puede ejecutar Ubuntu como una máquina virtual. En primer lugar, debe tener conocimientos básicos del lenguaje de programación Java, o al menos haber desarrollado algo de código en Java, que de hecho es el lenguaje de programación en el mundo Android.

Este artículo describe las diferentes partes de una app Android usando las aplicaciones de ejemplo Tic-Tac-Toe integradas en el código fuente de Android. Puede localizar muchas y diferentes muestras de aplicaciones en el repositorio del código fuente de Android de Google en http://bit.ly/1vkVLNE.

La app de ejemplo se puede encontrar en http://bit. ly/1ytcbsR. Hay muchos recursos en Internet dedicado a la programación de aplicaciones Android, y el mejor lugar para empezar es la web de formación de Google dedicada al desarrollo de Android en http://bit.ly/1cB6RmA.

Configuración de D<mark>esarrollo</mark>

Para este artículo, usaré Ubuntu 14.04 (64 bits) y te ensenare cómo instalar las diferentes herramientas. Existen 2 entornos de desarrollo interactivos principales (IDEs) que puedes utilizar: Eclipse o Android Studio. En este artículo me centraré en Eclipse.

Descarga JDK 1.7 desde la página web de Oracle en http://bit. ly/196ebsY. En mi caso, descargue el archivo "jdk-7u67-linuxx64.tar.gz".

Extrae el archivo gz en un directorio aparte. Mi archivo fue extraído en "/home/nanik/Downloads/jdk1.7.0_67".

Ejecuta el siguiente comando desde terminal, que indica a Ubuntu qué versión de las herramientas Java está instalada:



\$ sudo update-alternatives --install \
 "/usr/bin/java" "java"
 "/home/nanik/Downloads/jdk1.7.0_67/bin/java" 1071
 \$ sudo update-alternatives --install \
 "/usr/bin/javac" "javac"
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javac" 1071
 \$ sudo update-alternatives --install \
 "/usr/bin/javaws" ``javaws" \
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javaws" 1071
 \$ sudo update-alternatives --install \
 "/usr/bin/javap" ``javap" \
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javap" 1071
 \$ sudo update-alternatives --install \
 "/usr/bin/javap" ``javap" \
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javap" 1071
 \$ sudo update-alternatives --install \
 "/usr/bin/javadoc" ``javadoc" \
 "/usr/bin/javadoc" ``javadoc" \
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javadoc" 1071
 "/usr/bin/javadoc" ``javadoc" \
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javadoc" 1071
 "/usr/bin/javadoc" ``javadoc" \
 "/usr/bin/javadoc" ``javadoc" \
 "/home/nanik/Downloads/jdk1.7.0_67/bin/javadoc" 1071

Si ejecutas el comando java -version desde Terminal, obtendrá el siguiente resultado:

```
$ java -version
java version "1.7.0_67"
Java(TM) SE Runtime Environment (build 1.7.0_67-b01)
Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04,
mixed mode)
```

Tras finalizar la instalación de Java, también tendrá que descargar el IDE Eclipse Kepler desde http://bit.ly/1v5GssU.

Extrae el archivo gz en un directorio, como /home/nanik/Descargas/eclipse.

Descarga el kit de herramientas de desarrollo de Android (ADT) para Eclipse. Siga los pasos descritos en http://bit. ly/lvcZCMD.

Una vez introducida la dirección URL en la sección "Help>Install New Software" y seleccionada la Herramienta de Desarrollo para que se muestre en la pantalla "Available Soft-

DESARROLLO ANDROID

en Eclipse

🥹 🗉 Install		😣 🗉 Welcome to	Android Development		Android SDK Manager
Available Software		Walasma ta Andr	id Development		
Check the items that you wish to install.		welcome to Andr	d Development		
		Configure SDK		TV	
Work with: AndroidADT - https://dl-ssl.google.com/android	d/eclipse/ • Add				Fetching URL: https://dl-ssl.google.com/android/repository/r
Find more softw	vare by working with the " <u>Available Software Sites</u> " preferences.	To develop for And	oid you need an Android SDK and at least	one version of the Android	
type filter text	œ	APIs to compile aga	inst. You may also want additional versions	of Android to test with.	
Name	Version		-		
✓ III Developer Tools					Decearge del SDK de Andreid
🖉 🖗 Android DDMS	23.0.3.1327240	Install new SDK			Descarga del SDK de Allurold
Search Android Development Tools	23.0.3.1327240	🧭 Install the la	test available version of Android APIs (supp	ports all the latest features)	
🖉 🖗 Android Hierarchy Viewer	23.0.3.1327240				
Android Native Development Tools	23.0.3.1327240	🗌 Install Andre	old 2.2, a version which is supported by ~96%	% phones and tablets	
Android Traceview	23.0.3.1327240	(You can add	additional platforms using the SDK Manage	r.)	A Workspace Launcher
Select All Deselect All 6 items selected		Tassablesships	(heme/easily/andresid.edlys		Workspace Launchen
		Target Location	/nome/nanik/android-soks	biowse	Select a workspace
Details					Eclipse stores your projects in a folder called a workspace.
Features that add Android support to Eclipse for application	developers.	○ Use existing SDI	(s		Choose a workspace folder to use for this session.
Show only the latest versions of available software	Hide items that are already installed	O ose children good			
	What is already installed?	Existing Location	n:	Browse	Workspace: /home/nanik/workspace Browse
Group items by category	what is already installed:				
Show only software applicable to target environment					
Contact all update sites during install to find required soft	tware				
					Use this as the default and do not ask again
0	<back next=""> Cancel Finish</back>	(V)	< Back Next > C	Finish	Cancel OK
Instalación de herra	mientas de desarrollo	Instalac	ión del SDK de Andro	oid	Selección del espacio de trabajo tras l

ware". Haga clic en Next, siga las instrucciones y se completará la instalación.

Después de reiniciar Eclipse, se te pedirá que descargues e instales el SDK de Android. Tras completar la descarga, acepta la licencia para continuar la instalación.

Además del SDK, el instalador descargará las herramientas de desarrollo como se muestra en la captura de pantalla.

Haga clic en "Open SDK Manager", despues selecciona la que aparece resaltada en la captura de pantalla, al mismo tiempo que deseleccionas el resto que aparecen marcadas.

Hay 2 librerías que necesitan ser instaladas para completar la instalación en Ubuntu. Abre el terminal y ejecutar el siguiente comando:



🖷 Name	API	Rev.	Status
🔻 🗆 🚞 Tools	_		
🖉 📌 Android SDK Tools		23	Update available: rev. 23.0.2
🗆 📌 Android SDK Platform-tools		20	Installed
🗹 📌 Android SDK Build-tools		20	🗧 Not installed
🗆 📌 Android SDK Build-tools		19.1	🗖 Not installed
🗆 📌 Android SDK Build-tools		19.0.3	Not installed
🗆 📌 Android SDK Build-tools		19.0.2	Not installed
🗆 📌 Android SDK Build-tools		19.0.1	Not installed
🗆 📌 Android SDK Build-tools		19	Not installed
🗆 🦨 Android SDK Build-tools		18.1.1	Not installed
on 🖌 Android SDK Build-tools		18.1	Not installed
🗖 🖉 Android SDK Build-tools		18.0.1	Not installed
🗆 📌 Android SDK Build-tools		17	Not installed
🔻 🗆 🔁 Android L (API 20, L preview)			
🗆 🗎 Documentation for Android 'L' Preview SDK	20	1	Not installed

Selección de herramientas SDK de Android

Tras completarse la instalación de la herramienta de desarrollo reinicia Eclipse. Para comprobar si la instalación se ha realizado correctamente haz clic derecho en la pestaña "Package Explorer ", verás una pantalla de selección de Android como se muestra en la imagen anterior.

Instalar la app de muestra

Una vez completada la instalación de Eclipse, estás listo para

Selección del espacio de trabajo tras la instalación

Wizards: type filter text Image: Source of Sour	Select a wizard				\$
type filter text G Image: Solution of the second	Wizards:				
 Ceneral Android Activity Android Application Project Android Application Project Android Con Set Android Project from Existing Code Android Sample Project Jr Android XML Evout File Android XML Lyout File Android XML Values File Jr Template Development Wizard C/C++ CVS Git Java 	type filter text				×
 Android Android Activity Android Application Project Android Robject Android Object Android Sample Project Android Test Project Android Test Project Android XML File Android XML File Android XML Values File J¹ Template Development Wizard C/C+ Cit Java 	🕨 🗁 General				
Android Activity Android Application Project Android Application Project Android Object Android Object Android Sample Project Android Test Project Android Test Project Android XML Lyout File Android XML Lyout File Android XML Values File J Template Development Wizard Cost Cost Cost Cost Cost Cost Cost Cost	▼				
 Indicid project from Existing Code Android Object Android Sample Project Android Test Project Android XML File Android XML File Android XML Values File Android XML Values File J¹ Template Development Wizard Image: Cycs Git Java 	Android Activity				
Android Object Android Project from Existing Code Android Sample Project JA Android Test Project Android XML File Android XML Layout File Android XML Values File JJ Template Development Wizard b ← C/C+ b ← CVS b ← Git b → Java	Android Icon Set				
 Android Project from Existing Code Android Sample Project Android Test Project Android XML File Android XML Lile Android XML Values File C(C++ C(C+ C(S) Gilt Java 	Android Object				
 Android Sample Project Android Test Project Android XML File Android XML Layout File Android XML Values File Android XML Values File Android XML Values File C/C++ C/C++ C/S Git Java 	Android Project from Existing Code				
J ³ Android Test Project Android XML File Android XML Values File J ³ Template Development Wizard C/C++ C/S Git Java	😂 Android Sample Project				
 Android XML File Android XML Layout File Android XML Values File Graphate Development Wizard C/C++ C/C++ C/S Git Java 	J [*] Android Test Project				
 Android XML Layout File Android XML Values File J⁰₂ Template Development Wizard C/C++ C/VS Git Java 	📓 Android XML File				
Android XML Values File J ⁰ ₁ Template Development Wizard C ← (++ C ← (++ C ← (++ C ← (++ C ← (++)))) Git D → Java	🗟 Android XML Layout File				
J [°] Template Development Wizard ► 📴 C/C++ ► 🗁 Git ► 🚔 Java ► Dataset	Android XML Values File				
 ▷ C/C++ ▷ CVS ▷ Git ▷ Java ▷ Idauas 	J ^C ₀ Template Development Wizard				
 ▷ CVS ▷ Git ▷ Java ▷ Mana 	▶ 🧁 C/C++				
	CVS				
	?	< Back	Next > Ca	ancel Fini	sh

Creando un nuevo proyecto Android

trabajar con app de ejemplo. Localiza la muestra en Github, luego siga los siguientes pasos para importarla a Eclipse.

Selecciona File -> New -> Other y escoge "Android Project from Existing Code", luego pincha en "Next".

Selecciona el directorio raíz de la aplicación de ejemplo que localizaste en Github y selecciona "Copy projects into workspace", a continuación haz clic en "Finish".

Asistente de creación de proyectos Android	
8 🐵 New	
Select a wizard	->
Create one or more Android projects from existing code	
Wizards:	
type filter text	×
🕨 🗁 General	
🔻 🧁 Android	
😤 Android Activity	
😂 Android Application Project	
🌍 Android Icon Set	
📓 Android Object	
👺 Android Project from Existing Code	

Verás el proyecto en Eclipse como se muestra en las siguientes capturas de pantalla:

88					
Import Projects					
Select a directory to search fo	or existing Android pr	ojects			
Root Directory: /home/nanik	/workspace/TicTacTo	eAOSP			Browse
Projects:					
Project to Import	New Project Name				Select All
/home/nanik/workspace/	TicTacToe				Deselect All
					Refresh
🗹 Copy projects into worksp	ace				
Working sets					
Add project to working s	ets				
Working sets:				•	Select
The many second					Sector
?		< Back	Next >	Cancel	Finish

Ventana de Eclipse para Importar Proyecto

	券 ▼ 0 ▼ 9₀0 ▼ 9₀ ▼ ▼ 29 (▼ <i>A</i> ▼ {<i>A</i> ▼] ▼ 0
Image: Package Explorer 20 Image: Package Explorer 20 Image: Package Explorer 20 Image: Package E	Cartactore Manifest X Annu Version="1.0" encoding="utf-8"7≥ Copyright (C) 2010 The Android Open Source Proje I licensed under the <u>Apache</u> License, Version 2.0 (

Código del proyecto TicTacToe en el Explorador de proyectos de Eclipse

No olvides que el juegos de guerra, the Global Thermonuclear War finalizo con un simple juego TicTacToe



Para ejecutar la app en ODROID, conecte el micro USB al USB de tu PC, luego haz clic derecho en el proyecto y selecciona "Run As >Android Application". Eclipse detectará automáticamente ODROID (asegúrate que no hay otros dispositivos Android conectados) y ejecutará la aplicación en tu placa.

Android Manifest.xml

Una aplicación Android no tiene un punto de acceso "principal", pero sí que tiene un archivo XML que describe el contenido de la aplicación. Este archivo es el primer archivo que Android lee para conocer cuál es el contenido de una aplicación y cómo puede iniciarla. Este archivo se denomina AndroidManifest.xml y tiene una apariencia similar a la siguiente:

```
<manifest xmlns:android="http://schemas.android.com/
apk/res/android"
```

package="com.example.android.tictactoe"
android:versionCode="1"
android:versionName="1.0">

```
<uses-sdk android:minSdkVersion="8" />
<application android:icon="@drawable/icon"
android:label="@string/app name">
```

```
<activity android:name=".MainActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.
action.MAIN" />
<category android:name="android.in-
tent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity android:name="com.example.android.
tictactoe.GameActivity" />
</application>
</manifest>
```

Puedes leer el siguiente documento para obtener información más detallada de los diferentes elementos que componen el archivo en http://bit.ly/1msJ804. Los principales elementos que necesitamos entender se describen a continuación:

<uses-sdk..>

Este elemento te permite decidir a qué versión de Android está orientada. En nuestra muestra, nos centramos en un SDK mínimo de la versión 8 (KitKat es la versión 20).

<activity..>

Este elemento especifica el nombre de la clase Activity que hemos decidido para nuestra aplicación.

<intent-filter..>

En Android, todos los procesos se comunican con el resto a través de Intent, que es como una cola de mensajes. Este elemento indica a Android a que acción de Activity está vinculada. Para nuestra aplicación de muestra, el android.intent.action.MAIN está vinculado a nuestra Activity, lo que significa que esta clase Activity será la primera clase que será ejecutada por el entorno de trabajo cuando se inicie la aplicación.

Activity

Las Apps Android funcionan por subclases de la clase Activity, de modo que se sigue el Ciclo de vida de Activity para



Ciclo de vida de Ativity en una aplicación Android

ejecutar una aplicación. En nuestra app, tenemos 2 clases diferentes (MainActivity y GameActivity), y como se puede ver en el código, ambas amplían la clase Activity y anulan onCreate (...) y onResume (...).

Cuando una app se ejecuta por primera vez, se reclama el método onCreate (..), y posteriormente se reclama el método onStart (..). Si nuestra aplicación va al fondo como cuando el usuario cambia entre diferentes aplicaciones, se reclama el método onPause (..), y cuando cambiamos de nuevo a nuestra aplicación llevándola al primer plano, se reclama el método onResume (..). El flujo es simple y fácil de recordar ya que la app solo tiene 2 estados de ejecución: primer plano y de fondo.

Ten en cuenta que tenemos 2 elementos diferentes - Main-Activity y GameActivity. La forma en que Android decide qué clase ejecutar cuando la aplicación se inicia es analizando el elemento<intent-filter>. El MainActivity tiene un elemento <action..> android.intent.action.MAIN que dice a Android que esta es la clase principal que tendrá que ejecutar la app.

La clase MainActivity es el principal punto de entrada de toda aplicación, que a su vez reclama la GameActivity cuando la app se inicia:

```
private void startGame(boolean startWithHuman) {
    Intent i = new Intent(this, GameActivity.class);
    i.putExtra(GameActivity.EXTRA_START_PLAYER,
        startWithHuman ? State.PLAYER1.getValue())
: State.PLAYER2.getValue());
    startActivity(i); }
```

Este fragmento de código muestra como la clase GameActivityes es reclamada por el método startActivity (..) que es parte de la clase Activity. El método startActivity(..) es un método que enseña a Android a ejecutar la clase Activity especificada que ha sido enunciada. En nuestro ejemplo, hemos enunciado Intent mediante la variable "i".

Android utiliza mucho la comunicación entre procesos con aplicaciones iguales o diferentes. Esto permite a las aplicaciones "reutilizar" otras partes de una aplicación como si fuera parte de sus propias librerías internas utilizando la infraestructura Binder. Para utilizar este servicio, cualquier aplicación que quiera hacer la comunicación entre procesos utiliza un objeto Intent. Esto se puede ver en nuestro trozo de código anterior donde definimos la variable "i" para crear una nueva clase Intent con GameActivity como uno de los parámetros.

View

Cuando se ejecuta una app Android, lo que aparece en pantalla está basado en una clase llamada View. Esta clase es la piedra angular de tu interfaz de usuario, ya que todo lo que necesites colocar en pantalla se debe montar sobre View. Para explorar más a fondo el mundo de la interfaz de usuario de Android, puedes echar un vistazo a la documentación disponible en http://bit.ly/1ss708p.

Un juego TicTac-Toe en curso

La vista estándar no es adecuada para nuestra aplicación de ejemplo en Android, ya que sólo ofrece una vista de un widget en forma de cuadro de texto o casilla de selección. El control de la vista



ofrece flexibilidad a la app en relación a cómo se verá en la interfaz de usuario

En nuestra aplicación de muestra tenemos una clase llamada GameView que amplía la clase View:

public class GameView extends View {

public static final long FPS_MS = 1000/2;

onDraw()	This is the main method that contains most of the code, and will be executed every time Android needs to draw on the screen. You can think of this function as the 'paint' method.
onMeasure()	This method is called to set the view measurement.
onSizeChanged()	This method is called when view size has changed, or when it is beginning to initialize the view hierarchy.
onTouchEvent()	This method is called when there has been a touch or mouse click event initiated by the user.
onSaveInstanceState() onRestoreInstanceState()	These 2 methods are used to save and restore the state of the game.

DESARROLLO ANDROID

La lógica para desarrollar los cubos, junto con el estado del jugador, está dentro de la función OnDraw (..). Cada vez que Android tiene que actualizar la vista reclamará este método, por lo que es importante que el código que se ejecute en este método sea tan rápido y eficiente como sea posible, con el fin de evitar el retraso cuando los usuarios interactúan con la app.

Handler

Android está diseñado para utilizar el procesamiento asincrónico, donde los mensajes circulan de unas a otras aplicaciones con facilidad. Debido a este diseño, las aplicaciones utilizan una clase Handler, que actúa a modo de "devolución de llamada", donde se procesan los mensajes entrantes. La simple aplicación hace uso de esta clase para resaltar la celda que contiene la selección del usuario.

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    ...
    if (state != State.EMPTY) {
        // Start the blinker
        mHandler.
sendEmptyMessageDelayed(MSG_BLINK, FPS_MS);
    }
```

El handler envía un mensaje (MSG_BLINK) en un intervalo concreto, definido por la variable FPS_MS. La clase Callback que la aplicación define para recibir los mensajes MSG_ BLINK está definida del siguiente modo:

```
private class MyHandler implements Callback {
    public boolean handleMessage(Message msg) {
        if (msg.what == MSG_BLINK) {
            ... }
        }
        return true;
    }
    return false;
    }
}
```

Recursos (Gráficos y ca<mark>denas</mark> de caracteres)

Las Aplicaciones Android guardan cadenas de caracteres, imágenes y otros recursos binarios en una carpeta independiente que hace referencia a la aplicación. Todos los recursos se almacenan dentro de la carpeta res/. Durante el proceso de compilación y empaquetado, los archivos de esta carpeta se almacenan como se muestra en la siguiente captura de pantalla.

El archivo R.java generado contiene un ID para cada uno recursos definidos en nuestra aplicación. La forma en que la



aplicación accede a los recursos es mediante el uso de la API Android integrada:

```
public GameView(Context context, AttributeSet attrs)
{
       . . .
        mDrawableBg = getResources().getDrawable(R.
drawable.lib bg);
        setBackgroundDrawable(mDrawableBg);
        mBmpPlayer1 = getResBitmap(R.drawable.lib
cross);
        mBmpPlayer2 = getResBitmap(R.drawable.lib
circle);
       . . .
}
       . . .
       . . .
    private Bitmap getResBitmap(int bmpResId) {
       . . .
        Resources res = getResources();
        Bitmap bmp = BitmapFactory.
decodeResource(res, bmpResId, opts);
        if (bmp == null && isInEditMode()) {
            . . .
            Drawable d = res.getDrawable(bmpResId);
             . . .
    }
}
```

APK

Tu aplicación será empaquetada en un único archivo APK. Si deseas saber más sobre el funcionamiento interno de un .APK, consulta mi artículo de la edición de septiembre de 2014.

- FicTacToe [TicTacToe master]
 - 🖻 🔠 src
 - Generated Java Files]
 - 🕨 🛋 Android 4.0
 - Android Dependencies
 assets
 - 🔻 😓 bin
 - dexedLibs
 - 🕨 🗁 res
 - 🖻 AndroidManifest.xml
 - 📄 classes.dex
 - resources.ap

🗐 TicTacToe.apk

- 2 res
- 📑 _index.html
- 🖪 AndroidManifest.xml
- 📝 changeAuthor.sh
- project.properties

El archivo resultante TicTacToe.apk tras su desarrollo

Instalación

Puedes ejecutar tu aplicación desde Eclipse haciendo clic derecho sobre el nombre del proyecto, a continuación, selecciona Run As -> Android Application. Si tienes problemas al ejecutar la aplicación desde Eclipse, como los que yo he experimentado en algún momento cuando la he ejecutado con una máquina virtual, también puedes ejecutarla desde la línea de comandos mediante adb:

I. Escribe adb devices para asegurarte que tienes conexión con ODROID.

2. Usa el comando adb install TicTacToe.apk para instalar la aplicación una vez que hayas establecido la conexión. Normalmente el apk se encuentra en la carpeta /bin dentro del entorno de trabajo de Eclipse.

EPOCA DE BONANZA DE LA INFORMATICA 8-BIT DIVIÉRTETE JUGANDO A TUS JUEGOS FAVORITOS DE MSX

por Bruno Doiche

isfruta de lo último en informática moderna desde 1983 hasta 1995 emulando el único e incomparable MSX. Antes de que apareciera el gran éxito de Nintendo, MSX fue la plataforma para la cual los principales estudios de juegos japoneses, como Konami y Hudson Soft, desarrollaron videojuegos. La serie Metal Gear, por ejemplo, fue escrita originalmente para el hardware MSX.

```
$ wget http://sourceforge.net/projects/openmsx/\
files/openmsx/0.10.1/openmsx-0.10.1.tar.gz
```

```
$ tar -zxvf openmsx-0.101.tar.gz && \
cd openmsx-0.10.1
```

\$ sudo -s ./configure && make -j4 && make install

Puede que tenga que instalar las siguientes dependencias:

GLEW, libao, libogg, libpng, libtheora, libtrorbis, libxml2, OpenGL, SDL, SDL, TCl, zlib



Copia tu archivo.rom MSX favorito a~/.openMSX/share/ software/, luego escribe openmsx <gamename.rom>.



JUEGOS LINUX COMPARANDO LA EMULACION DE LA PSP EN LINUX Y ANDROID

por Tobias Schaaf

ay muchas ventajas al ejecutar juegos en Linux más que en Android. Para ello voy a comparar el rendimiento del famoso emulador para PlayStation Portable (PSP) PPSSPP, disponible para Andoid y Linux. Como yo no tengo experiencia con Android, todas las capturas de pantalla que aparecen en este artículo han sido tomadas de la versión para Linux de PPSSPP.

Ajustes PPSSPP

Para realizar una comparación adecuada entre los dos sistemas, he usado la misma configuración en ambos con una resolución de 1080p:

Frameskipping:	3
Auto frameskip:	ON
Rendering Resolution:	2x PSP
Mipmapping:	OFF
Hardware transform:	ON
Software skinning:	ON
Vertex Cache:	ON
Lazy texture caching:	ON
Retain changed textures:	ON
Disable slower effects:	ON
Spline/Bezier curves quality:	Medium

El resto de opciones la he dejado por defecto en su configuración estándar, sin modificaciones. Para tener una referencia, he activado en ambos la opción "Show FPS counter ", ofreciéndome así una visión detallada del rendimiento del emulador con ratio y porcentaje: 40/60 (100%). El primer número representa los frames por segundo (FPS) actuales, cómo de rápido se está ejecutando el emulador. El segundo número son los FPS esperados, que varía de un juego a otro así como de una escena a otra. Algunos juegos sólo necesitan ejecutarse



Asphalt: Urban GT2 (PSP) es uno de los muchos y excelentes juegos para el emulador PPSSPP

a 30 o 40 fps en determinadas escenas (por ejemplo, durante la reproducción de vídeo), mientras que otros siempre necesitan ejecutarse a 60 FPS.

El último número en % proporciona la velocidad actual del emulador, comparada con la velocidad requerida. 100% significa que el juego se está ejecutando a toda velocidad. Esto es posible incluso si el emulador sólo se ejecuta a 40 de los 60 FPS que son los que realmente debería tener, debido a que está activada la opción frameskip. Si el valor se reduce significativamente (80% o menos) experimentarás cierta lentitud en el juego.

Información general

Utilicé mi imagen ODROID GameStation Turbo para probar los juegos en Linux y Android, ejecutando la última versión 4.4.4 KitKat proporcionada por Hardkernel. La configuración no es muy distinta de los parámetros por defecto, excepto por algunas opciones de aceleración activadas, que probablemente ni siquiera sean necesarias y frameskip que aumenta notablemente el rendimiento.

Me he dado cuenta que la resolución

con la que se ejecutan los juegos es principalmente responsable de lo bien que funciona el emulador. La configuración por defecto sólo utiliza la resolución 1x PSP que es el parámetro más bajo. Esto significa que los juegos serán dimensionados a 480x272 que como puedes imaginar, se verán muy mal cuando ampliamos a una resolución de pantalla de 1920x1080. Con la configuración a 2x, puedes conseguir una resolución de 920x544, que presenta buen aspecto a 1080p. Algunos juegos son, incluso capaces de ejecutarse a resoluciones 3x en Linux a una velocidad aceptable.

Sólo para ver la diferencia, intenta ejecutar un juego a 1x PSP, ve cambiando la configuración de la resolución a 2x y 3x, y entenderças por qué esto también afecta al rendimiento. Si eliges "1:1 Auto", PSP se mostrará a una resolución cercana a la resolución de pantalla nativa, la cual utiliza muchos recursos.

Juegos

He elegido 4 juegos para comparar el rendimiento de Linux y Android con la PSP:

JUEGOS LINUX

*Tekken 6 – Uno de los más representativo juegos de lucha, ya que a mucha gente parece que le gusta.
*Ultimate Ghosts 'n Goblins – Un juego de plataformas, uno de los juegos arcade más difíciles.
*Asphalt Urban GT2 – Un juego de carreras, por su alto consumo de recursos
*Naruto Shippuuden: Kizuna Drive – Otro juego de lucha con misiones en lugar de combate cuerpo a cuerpo.

Decidí probar Tekken 6 en primer lugar, por ser tan popular y mencionarse constantemente en los foros ODROID. Aunque los gráficos 3D no son excepcionales, tiende a ser pesado y utiliza una gran cantidad de recursos, siendo una buena opción para la comparar Android y Linux.

Tekken 6 - Android

La versión de Android de Tekken 6 funciona sin mayores problemas, y se puede disfrutar al completo a una velocidad aceptable. La introducción se ve un poco lenta y oscila entre los 20 y 40 FPS (de los 60 FPS), ya que usa frameskip. Aparecen algunas líneas horizontales durante la reproducción de la película, lo que significa que el juego no está sincronizado durante la reproducción.

El menú se ejecuta a 30 de los 60 FPS, se nota cuando te mueves rápidamente por el menú y a veces se distorsiona el sonido. La experiencia de juego varía entre 12 y 20 FPS, con una velocidad en general entre el 95 y el 100%, con casos raros de retraso. La experiencia global en Android es aceptable y con algunos retoques en el emulador, debería ser suficiente como para jugar con normalidad.

Tekken 6 - Linux

Tekken funciona notablemente mejor en Linux que en Android. La introducción se ejecuta a una velocidad constante de 60 FPS, sin líneas horizontales u otros problemas durante la reproducción del video. La velocidad del menú también se mantiene en 60 FPS, sin problemas de sonido y movimientos muy fluidos. Los



La jugabilidad del Tekken 6 es muy rápida y los gráficos muestran el potencial de ODROID



Te lo has ganado si puedes vencer a Ling Xiaoyu en Tekken 6 - ¡No es fácil derrotarla!

gráficos se ven muy bien, se muestran las sombras y los efectos de luz que se ejecutan a toda velocidad. La tasa de fotogramas durante el juego permanece entre 15 y 25 FPS. La acción es muy fluida y da una sensación de combate rápido. En general la experiencia de Tekken 6 en Linux es mejor que en Android.

Tekken 6 es un buen juego de lucha para PSP, aunque prefiero juegos como Soul Calibur más que el Tekken o Street Fighter. Tanto las versiones de Android como de Linux son completamente jugables, con pequeños problemas gráficos de poca importancia. Ambos se ejecutan lo suficientemente rápido como para eliminar cualquier ralentización en el juego, aunque la versión de Linux es un poco más rápida que la versión de Android, en particular durante la reproducción de vídeo y navegando por el menú.

En general, le doy un nivel de compatibilidad en ODROID al Tekken de un 8 sobre 10 para Android y un 9 sobre 10 para Linux. Este juego es, sin duda un TRIUNFO

Ultimate Ghosts 'n Goblins

Como juego de plataformas, es más un juego en 2D que en 3D sin que llegue a usar una gran cantidad de gráficos. En consecuencia el juego hace un uso muy eficiente de la CPU y la GPU. Es un buen ejemplo como juego liviano que se ejecuta en PPSSPP.

Ultimate Ghosts 'n Goblins es un remake del viejo juego de arcade Ghosts' n

JUEGOS LINUX



Ultimate Ghosts 'N Goblins es uno de los juegos de arcade más difíciles de todos los tiempos



La secuela de G'n'G incluye mejoras en gráficos y sonido, ¡y es muy divertido!

Goblins, que fue uno de los juegos más difíciles de su tiempo. La nueva versión no es mucho más fácil que su predecesor, pero es un juego bastante divertido. Hay que señalar que el juego original tenía un problema con un efecto gráfico en el primer nivel que hacía que fallase la pantalla, y el rendimiento se reducía drásticamente hasta alcanzar el nivel 2. Esta cuestión de hecho fue solucionada en el código de emulador PPSSPP, y desde la versión 0.9.8 el juego se puede ejecutar sin experimentar problemas.

UG'nG - Android

Al igual que en Tekken 6, la intro se ejecuta a 30/60 FPS, pero al menos no presenta líneas horizontales u otras anomalías gráficas. Este juego es muy difícil y se debe utilizar un gamepad con Android, ya que jugar con un teclado es casi imposible. El control del menú funciona muy bien, al igual que lo hace el sonido y la música.

Sin embargo, jugar al juego en Android es casi imposible. PPSSPP para Android sólo es capaz de mostrar de 8 a 13 FPS, provocando que el juego sea muy lento. Es extraño, ya que el juego no es tan exigente, e incluso funciona muy bien con versiones anteriores de PPSSPP en Linux, aún cuando todavía tenía el problema del efecto gráfico y a pesar de que no había sido optimizado para imágenes hardfloat.

UG'nG - Linux

La experiencia con Linux de Ultimate Ghosts 'n Goblins es perfecta, sin ningún tipo de problemas. La intro, el menú y la música funcionan a plena velocidad (60/60 FPS). El juego se ejecuta sin ninguna caída de velocidad notable. No hay mucho más que decir de la versión para Linux excepto que, ¡Es muy divertido!

No entiendo por qué Android ejecuta tan mal este juego. Aún así, la introducción, la música y el menú parece funcionar bien y con algunas modificaciones, la versión para Android probablemente pueda ejecutarse a 15/30 FPS. En Android, doy al UG'nG 5 puntos sobre 10, mientras que la versión Linux obtiene 10 sobre 10 por la compatibilidad y la experiencia de usuario.

Asphalt Urban GT2

Este fue uno de los primeros juegos de carreras al que jugué en la PSP. No es tan difícil como juego de carreras, aunque no tienen tantas opciones como el Need for Speed o Midnight Club. Sin embargo, dispones de un montón de coches y pistas para jugar. Se empieza con una cantidad muy alta de nitro y es muy divertido echar a tus oponentes (y a la policía) fuera de la carretera por algo de dinero extra.

Aunque este juego es bastante pequeño (unos 400 MB del archivo rom), es uno de los juegos más difíciles de conseguir que funcione correctamente en ODROID. Desde que tengo funcionando mi primera versión de PPSSPP, este juego ha estado lejos de ejecutarse a toda velocidad. Siempre ha sido un reto para mí, ya que a veces ni siquiera mi portátil, que tiene una moderna tarjeta gráfica NVIDIA, ha sido capaz de ejecutarlo a toda velocidad.

Asphalt Urban GT2 es un juego de muy bueno con una amplia variedad de pistas y coches. Sin embargo, la experiencia de juego puede variar mucho dependiendo de la configuración y la versión del emulador que utilices. Una configuración errónea puede hacer que la versión funcione a menos de 10 FPS.

También parece utilizar muchos efectos especiales que todavía no son totalmente compatibles con PPSSPP, por lo



Asphalt Urban GT (PSP) cuenta con una gran cantidad de coches y pistas desafiantes



Asphalt Urban GT permite deslizar los coches por las curvas, imitando a un coche real

que es un juego realmente útil para comprobar las mejoras y parches a medida que los desarrolladores vayan liberando nuevas versiones de PPSSPP.

Asphalt Urban GT2 -Android

Como era de esperar y debido a la carga de gráficos, Android tiene problemas cuando ejecuta este juego. Incluso los logos 2D durante la introducción no se muestran a toda velocidad (30/60 FPS). El menú sorprendentemente funciona bien en la pantalla de inicio (entre 25 y 40 FPS), y si entras en un submenú (por ejemplo, "Arcade - Single Race") o en a vista previa de un coche, se reduce a unos 15 FPS, pero sigue funcionando a toda velocidad.

Sin embargo, la experiencia de juego

en Android no es tan buena como en el menú. Aunque la prueba de rendimiento varía desde los 20 FPS a los 7 FPS, apenas se percibe cuando se ejecuta a toda velocidad alcanzando los 20 FPS y el 100%. Es jugable, pero Asphalt Urban GT2 en Android realmente no ofrece una experiencia de juego agradable.

Asphalt Urban GT2 -Linux

En general la experiencia con Asphalt Urban GT2 es algo mejor en Linux: los logos se muestran a 60 FPS, el menú a 40-50 FPS y los submenús a 15-20 FPS. Durante el juego, la tasa de fotogramas también se reduce a 10 a 25 FPS, pero se muestra más fluido que en Android. A menos que utilices tu nitro, se ejecuta como en una auténtica consola PSP. El juego tiene un comportamiento extraño en ocasiones: por ejemplo, cuando se conecta el multi-threading, en realidad se vuelve más lento y el frameskip también puede tener un efecto negativo en su rendimiento. Los efectos de luz a menudo parecen como si estuviesen fuera de lugar.

Aunque este juego es bastante pequeño ofrece algunas características interesantes, tiene una muy buena banda sonora y el juego es bastante divertido. Sin embargo, funciona muy lento y tiene muchos problemas técnicos. Será interesante ver cómo evoluciona el emulador y estoy deseando poder jugar pronto a este juego a toda velocidad en ODROID.

Naruto Shippuuden: Kizuna Drive

Otro juego de lucha en el que completas misiones en lugar de luchar contra enemigos individuales, como Tekken. A menudo luchas con varios enemigos a la vez y se puedes luchar en equipo. Cuenta con llamativos gráficos estilo cómic y ya que el juego se muestra a 30 FPS (en no a 60 FPS), es asumible por el hardware.

Los combates contra jefes son muy interesantes y a menudo necesitan a todo tu equipo para vencer a un enemigo. En las llamadas "free missions" se puede luchar con cualquier personaje, no está restringido a usar sólo a Naruto.

Kizuna Drive -Android

El juego, el menú y película de inicio funcionan bien en Android a 30 FPS. La jugabilidad varía entre 20 a 30 FPS, pero la mayoría del tiempo se mantiene por encima de 25 FPS sin recesos.

Kizuna Drive - Linux

Kizuna Drive se ejecuta a toda velocidad sin ningún problema. La versión de Linux, de hecho tiene un muy buen rendimiento y se ve de forma asombrosa cuando se juega a una resolución 3x PSP o incluso en modo1:1 Auto.

Naruto Shippuden - Kizuna Drive es

JUEGOS LINUX

uno de los juegos que mejor se ejecuta en el emulador PPSSPP. Funciona perfectamente tanto en Android como en Linux, aunque en Linux se ejecuta extraordinariamente bien, haciendo que el juego se vea a una resolución de 1080p.

Resumen

Hablando en general, PPSSPP se ejecuta más rápido y con menos problemas en Linux que en Android, aunque PPSSPP nunca fue pensado para ejecutarse en sistemas hardfloat. Sin embargo, cuando se usa PPSSPP en Debian Wheezy (como mi imagen ODROID GameStation Turbo), los juegos se detienen unos segundos de vez en cuando durante su ejecución.

No estoy muy seguro que causa este comportamiento, pero creo que es un problema de E/S que tiene lugar cuando se accede a las imágenes .cso.

El retraso parece sólo suceder con Debian Wheezy, y no ocurre cuando se utiliza Ubuntu. Algunos juegos lo hacen con más frecuencia. En general, PPSSPP es un emulador bien escrito que funciona muy bien en dispositivos ODROID usando Linux o Android. Tiene la ventaja de utilizar OpenGL ES 2.0, que activa todo el potencial gráfico de ODROID.

Encontró mi copia de GameStation Turbo, y ha estado jugando durante tres días hasta vencer al jefe final de Kizuna Drive





Naruto Shippuden: Kizuna Drive (PSP) es como el Mortal Kombat pero con misiones de combate y un mundo en 3D para explorar



Luchar con el jefe en Naruto Shippuden: Kizuna Drive es un desafío para horas y requiere mucha práctica

El ganador entre las versiones de Linux y Android de PPSSPP es claramente Linux, que cuenta con una alta tasa de fotogramas y movimientos fluidos



SODESTACADO: QUIET GIANT UN LIVIANO SERVIDOR LAMP, SAMBA Y MINECRAFT

los cuales abren el escritorio Blackbox.

Haga clic derecho en el escritorio y sele-

cciona "xterm" desde el menú para abrir

una ventana de terminal. La mayoría del

mantenimiento del servidor Linux se

realiza desde la línea de comandos, cual-

quier servidor que no necesitemos puede

ser eliminado con el comando tasksel.

Hay otras aplicaciones disponibles al uti-

lizar tasksel, incluidos los paquetes de es-

critorio como Kubuntu y Unity. Sin em-

bargo, la imagen está adaptada para un

alto rendimiento con el pequeño escrito-

rio Blackbox, que cuenta con una escasa

interfaz y un bajo consumo de memoria.

Se utilizan aproximadamente 350MB de

RAM cuando los 8 servidores están fun-

cionando, dejando 1.65GB libre para el

uso del servidor.

Quiet Giant

para las series X, U y XU se puede descargar desde http://bit.ly/IrhHymu. Mira los foros ODROID en http:// forum.odroid.com para las versiones I4.04 ¡Disponibles muy pronto!

por Rob Roy

uiet Giant, una imagen descargable para las series X, U y XU basada en Ubuntu Server, ofrece varios y diferentes servidores, incluyendo Apache, Tomcat, MySQL, FTP, Samba y Minecraft. Está pensado para ser un servidor LAMP con un sencillo entorno de desarrollo, pero también puede servir como una plataforma liviana para un sistema embebido que requiere estabilidad a largo plazo. LAMP, que significa Linux, Apache, MySQL y PHP/ Perl es una opción muy común para las aplicaciones de Internet, El Quiet Giant funciona muy bien como plataforma de aprendizaje para desarrolladores web.

Una vez arrancado, el nombre de usuario y la contraseña son "odroid",

Commando Tasksel

laces System 🏟 🔽 🙆 🗖		: a 4a
	jagdeep@jagdeep-desktop: ~	: , , , , , , , , , , , , , , , , , , ,
erminal <u>H</u> elp		
ition		
	Software selection You can choose to install one or more of the following predefined collections of software choose software to install:	



Blackbox es el escritorio por defecto para Quiet Giant

Aunque Quiet Giant incluye el escritorio Blackbox, es más común acceder a la máquina para ejecutar actualizaciones utilizando un cliente Secure Shell (SSH). Las transferencias de archivos se realizan con el Protocolo de Transferencia de Archivos (FTP) a los directorios especificados, donde pueden ser recuperados por los procesos programados que se ejecutan en el servidor. Por otro lado, el protocolo Samba también está disponible para compartir archivos con otros equipos de la red, se puede utilizar para crear un simple servidor multimedia.

SSH

Secure Shell (SSH) es el servicio de comunicación más básica disponible en Quiet Giant. SSH monta una ventana de comandos de terminal sobre Ethernet para que los comandos pueden ser lanzados de forma remota.

HTTP/Apache

Utilizo Quiet Giant como servidor

web en mi red local de casa, permitiéndome desarrollar sitios web sin exponerlos a Internet. Sin embargo, también es posible utilizarlo como un servidor público si se hacen algunas modificaciones.

Cuando se elige que el servidor sea de acceso público, es fundamental instalar un router entre el servidor e Internet, y que el firewall del router pueda proteger el servidor ODROID de cualquier acceso no autorizado. Sólo es seguro usar un ordenador como servidor de Internet si los datos personales importantes no se almacena en la misma red que el servidor. Además, asegúrate que las contraseñas para cada servicio sean seguras antes de configurar el router.

Servidor web

Para crear un servidor web de uso público, desinstala por seguridad el resto de servicios, excepto Apache y MySQL. A continuación, configura el router local para redirigir las solicitudes web entrantes al servidor Quiet Giant, como se detalla en los siguientes pasos.

- Consigue la dirección IP (privada) del servidor web local escribiendo en la ventana de terminal del servidor o por SSH:

```
$ ifconfig | grep inet | grep Bcast
inet addr:192.168.1.10
Bcast:192.168.1.255
Mask:255.255.255.0
```

- Asigne una dirección IP fija, que en este ejemplo es 192.168.1.10 al servidor web a través del panel de administración del router.

- Una vez reservada la dirección IP. El puerto 80 que es el estándar para HTTP, debe apuntar a la dirección IP. Para ello utiliza de nuevo el panel de administración y accede a la parte de asignación de puertos.

- Tras configurar correctamente el router, escribe el siguiente comando en una ventana de terminal del servidor para conocer tu dirección IP pública:

```
$ curl -s checkip.dyndns.org|sed
-e `s/.*Current IP Address: //'
-e `s/<.*$//'
79.211.83.113
```

En este ejemplo, la dirección del servidor es 79.211.83.113, al que se puede acceder desde cualquier navegador usando "http://79.211.83.113" siempre que ODROID y el router estén conectados a Internet. Servicios similares como el Protocolo de Transferencia de Archivos (FTP) y SSH, también pueden hacerse públicos con la misma técnica de reenvío de puertos, usando el puerto correspondiente para esos servicios, como es el puerto 21 para FTP.

MySQL

MySQL utiliza el puerto 3306 por defecto, y permite a los sitios web y otras aplicaciones acceder a la información almacenada en una base de datos utilizando un lenguaje de programación especial llamado Lenguaje de Consulta Estructurado (SQL). Cuando se combina con otro lenguaje de programación como el PHP: la entrada de datos del usuario, los datos de registro y otra información se puede registrar, recuperar y archivar.

La versión de MySQL instalada en Quiet Giant viene con una contraseña de administrador en blanco. Por seguridad, la contraseña por defecto debe cambiarse inmediatamente. El procedimiento para actualizar la contraseña se describe en http://help.ubuntu.com:

- En primer lugar, detén el proceso mysql escribiendo en el terminal:

sudo /etc/init.d/mysql stop

- Luego, debes reiniciar el demonio mysqld:

sudo /usr/sbin/mysqld --skipgrant-tables --skip-networking &

- A continuación, reinicie el proceso del cliente mysql:

```
mysql -u root
```

- Desde el prompt MySQL, ejecuta este comando para restablecer la contraseña de root:

FLUSH PRIVILEGES; SET PASSWORD FOR root@'localhost' = PASSWORD('password'); FLUSH PRIVILEGES; exit;

- Por último, detén el proceso de mysql y lánzalo de nuevo:

sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql start

Samba

Samba es un protocolo de intercambio de archivos basado en Windows, Quiet Giant lo tiene preconfigurado para compartir los ficheros ubicados en /var/www/ (Apache), /var/lib/tomcat7/ webapps/ROOT (Tomcat) y /home/ ODROID/Documents (uso general). Accede a Samba escribiendo la dirección IP local del servidor Quiet Giant en un explorador de archivos desde cual-quier otro ordenador de la red, introduce el nombre de usuario y contraseña por defecto "odroid". Esto te dará acceso al directorio compartido.

Para configurar Samba, edite el archivo /etc/samba/smb.conf en una ventana de terminal. También existe una interfaz gráfica de usuario para la administración de usuarios y carpetas disponible si usas el escritorio Blackbox:

sudo system-config-samba

Tomcat

Tomcat es un servidor web de código abierto basado en Java que se puede utilizar para desarrollar casi cualquier tipo de aplicación web. A una aplicación instalada se puede acceder visitando http://127.0.0.1:8080 desde el servidor, o escribiendo la dirección IP (privada) interna del servidor seguido de ":8080", desde cualquier ordenador de la red local. Si va a configurar un servidor público, el puerto 8080 debe ser redireccionado a ODROID. Para obtener más información sobre programación y contribuir con el proyecto Tomcat, visita http://tomcat.apache.org

Spigot (Minecraft)

¡Todo el mundo adora Minecraft! Quiet Giant viene con una versión optimizada del servidor de Minecraft denominada Spigot version 1.6.4 instalada en /home/odroid/público/ Se puede iniciar escribiendo los siguientes comandos en el Terminal:

por Justin Lee

```
cd ~/Public/spigot
./spigot.sh
```

Memoria virtual

Swap está habilitado en el kernel, permite ampliar la memoria más allá de los 2GB escribiendo bloques de memoria en el disco, ya sea con un único archivo, o con una partición dedicada aparte. Para aprender más sobre la creación de archivos swap, consulte http:// bit.ly/1pYfWSY. Para obtener información sobre cómo crear una partición swap, visite http://bit.ly/1rdONWu.

Sincronización del reloj

Si la imagen se inicia sin estar conectada a la red LAN, el reloj puede no estar sincronizado. En Linaro, esta diferencia de tiempo puede hacer que el sistema de archivos raíz sea montado como sólo lectura. Si esto sucede, escribe "fsck/" como root, luego reinicia con el cable LAN conectado. Esto desbloquear el sistema de archivos y sincronizará el reloj con la hora de Internet vía NTP.

Asegúrate de actualizar tu servidor con los últimos parches de Ubuntu



ODROID-XU3 ¡EL ORDENADOR MÁS RÁPIDO CREADO POR HARDKERNEL HASTA LA FECHA!



El ODROID-XU3 es un ordenador de placa reducida con 8 nucleos ARM big.LITTLE

1 ODROID-XU3 es un nuevo micro ordenador de placa reducida (SBC) de 8 núcleos con tecnología ARM[®] big.LITTLE[™] que usa el denominado Heterogeneous Multi-Processing (HMP). Pertenece a una nueva generación de dispositivos informáticos más potentes, con un hardware de alta eficiencia energética y un tamaño reducido. Con soporte para código abierto, la placa puede ejecutar varias distribuciones de Linux, como el último Ubuntu 14.04 y el Android 4.4. Utiliza eMMC 5.0 y la interfaz USB 3.0 que ofrecen una alta velocidad de transferencia de datos, una característica que cada vez es más demandada por la capacidad de procesamiento avanzado de los dispositivos ARM y que permite a los usuarios experimentar las últimas mejoras, como son un arranque más rápido, la navegación web y la experiencia de juegos en 3D.

- Samsung Exynos 5422 Cortex[™]-A15 2.0Ghz quad core and Cortex[™]-A7 quad core CPUs
- Mali-T628 MP6 (OpenGL ES 3.0/2.0/1.1 y OpenCL 1.1 Full profile)
- 2Gbyte LPDDR3 RAM a 933MHz (14.9GB/s de ancho

- de banda) PoP stacked
- eMMC5.0 HS400 Flash Storage
- USB 3.0 Host x 1, USB 3.0 OTG x 1, USB 2.0 Host x 4
- HDMI 1.4a y DisplayPort1.1 para la visualización
- Herramienta integrada de control de consumo de energía

Monitorización integrada de energía

El ODROID-XU3 tiene una herramienta integrada de análisis de energía, con 4 sensores de corriente/voltaje para medir el consumo de energía de los núcleos Big A15, núcleos Little A7, GPUs y DRAM de forma individual. Los desarrolladores profesionales pueden monitorizar el consumo de energía de la CPU, GPU y DRAM usando el circuito de medición de energía integrado.

Con la herramienta de análisis de energía integrada, el XU3 reduce la necesidad de realizar continuas pruebas relacionadas con el consumo de energía. Los usuarios tienen la oportunidad de mejorar y optimizar el rendimiento de sus aplicaciones por CPU/GPU, manteniendo el consumo de energía tan bajo como le sea posible.

La información sobre la potencia

43MHz 64°C 1000MHz, 20% 1000MHz, 5% 1000MHz, 6% 1000MHz, 6% 1600MHz, 0% 64°C 1600MHz, 0% 61°C 1600MHz, 81% 61°C wer : 1.1V, 1.139A, 1.273W wer : 1.0V, 0.159A, 0.171W ower : 1.0V, 0.446A, 0.544W

El ODROID-XU3 ejecutando la aplicación de Monitorización de energía

eléctrica, amperaje, voltaje y frecuencia se muestra superpuesta sobre la pantalla en Android. Asimismo puedes controlar la temperatura de la GPU y de los núcleos, tal y como se muestra en la captura de pantalla.

Heterogeneous Multi-Processing (HMP)

El ODROID-XU3 está equipado con cuatro grandes núcleos (ARM [®] Cortex[®] -A15 [™] de hasta 2,0 GHz) y cuatro pequeños núcleos (ARM [®] Cortex[®] -A7 [™] de hasta 1,4 GHz), ofreciendo mayores capacidades de procesamiento al tiempo que mantiene un consumo de energía más eficiente. Con la solución big. LITTLE[™] HMP, Exynos-5422 puedes utilizar los ocho núcleos al máximo para ejecutar intensas tareas desde el punto de vista informático.

Un ODROID-XU3 con Ubuntu 14.04 LTS y la última versión del Kernel 3.10. ¡Es extremadamente rápido!

OpenGL ES 3.0 y OpenCL 1.1

La GPU ARM [®] Mali [™] - T628 MP6 ofrece una API con soporte para OpenGL ES 1.1, OpenGL ES 2.0

y OpenGL ES 3.0, OpenCL 1.1 Full Profile y Google Render-Script. El chip Mali-T628 es la GPU elegida para la nueva generación de dispositivos líderes del mercado, optimizado para ejecutar los impresionantes gráficos de las aplicaciones que se consumen hoy día, tales como gráficos en 3D, efectos visuales, realidad aumentada, generación de texturas y reconocimiento de voz. Puede descargar el SDK completo para OpenGL ES y OpenCL desde el sitio web del desarrollador ARM Mali sin ningún coste.

e**MMC 5.0**

eMMC utiliza la tecnología de memoria flash inteligente que no sólo ofrece la capacidad de almacenar contenido digital, sino que también hace frente a los altos y estrictos requisitos de rendimiento secuenciales y aleatorios para asegurar una adecuada experiencia de usuario. Esto hace posible el rápido inicio del sistema operativo y de las aplicaciones, una multi-tarea totalmente integrada y el rápido acceso a la nube.

En octubre de 2013, JEDEC pub-



licó la última versión de su popular estándar eMMC llamado JESD84-B50: Embedded **MultiMedia** Card Electrical Standard (5.0). eMMC v5.0 incluye nuevas

funcionalidades y mejoras para la memoria flash de almacenamiento masivo ampliamente utilizada en los teléfonos inteligentes y otros dispositivos móviles; llegando a alcanzar los desafiantes objetivos de rendimiento demandados por la nueva generación de sistemas móviles al introducir la modalidad HS400 que ofrece una mejora adicional en términos de velocidad (hasta 400 MB/s frente a los 200 MB/s de la versión anterior). JESD84- B50 está disponible para su descarga gratuita desde el sitio web JEDEC en http://bit.ly/1uQKfZC.

Para ver una demostración de las capacidades de XU3, accede al video http://bit.ly/1CvJBWv

Especificaciones

Procesador

Samsung Exynos5422 ARM® Cortex™-A15 Quad 2.0GHz/ Cortex™-A7 Quad 1.4GHz

Memoria

2Gbyte LPDDR3 RAM PoP (933Mhz, 14.9GB/s de ancho de banda, 2x32bit bus)

Acelerador 3D

Mali[™]-T628 MP6 OpenGL ES 3.0 / 2.0 / 1.1 y OpenCL 1.1 Full profile

Monitor de energía

Control del consumo de energia de nucleos big.LITTLE, GPU y DRAM

Audio

Códec de audio / Clavija de 3.5 mm estándar para auricular con salida de audio digital HDMI Salida SPDIF óptica USB opcional

USB

1 Puerto USB 3.0 Host maxima velocidad USB standard Conector tipo A 1 puerto USB 3.0 OTG máxima velocidad USB Micro Conector tipo A-B 4 puertos USB 2.0 Host de alta velocidad standard Conector tipo A **Visualización** HDMI, DisplayPort

Almacenamiento

eMMC 5.0 Flash (hasta 64GB) Tarjeta MicroSD (hasta 64GB)

LAN

Fast Ethernet LAN 10/100Mbps Ethernet con clavija RJ-45 (Soporta Auto-MDIX) Gigabit Ethernet LAN (Optional) Adaptador USB 3.0 a Gigabit Ethernet (módulo USB opcional) WiFi USB IEEE 802.11b/g/n 1T1R WLAN con Antnna (módulo USB opcional)

Almacenamiento

Interfaz SATA para HDD/SSD opcionales. Adaptador USB de máxima velocidad (USB 3.0) a Serial ATA3 para HDD y SDD de 2.5"/3.5"

Fuente de Alimentación (incluida)

Potencia 5V 4A

Software del sistema

Ubuntu 14.04 + OpenGL ES + OpenCL con Kernel LTS 3.10 Android 4.4.2 con Kernel LTS 3.10 El código fuente esta disponible desde nuestro Github

РСВ

Tamaño: 94 x 70 x 18 mm



con Kernel LTS 3.10 4.4.2 con Kernel LTS 3.10 Para empezar, descarga la última versión de RetroArch desde http://bit.

ly/1uP6ejM. El APK funciona con cualquier versión reciente de Android, incluida Kit Kat. Asegúrate que el mando de la Xbox 360 está conectado pudiendo controlar el escritorio de Android. A continuación, iniciar RetroArch. Si utilizas un mando inalámbrico, usa el botón del receptor USB para conectar los joysticks en primer lugar.

RETROARCH

CON RETROARCH VI.0.0.2+

n las más recientes versiones de

desarrollo de RetroArch para An-

droid (1.0.0.2r34 y superior), la

forma en que se configuran los mandos

de la Xbox 360 ha cambiado. En lugar de

utilizar el menú de opciones de la prim-

era pantalla, la configuración se realiza

en el interior del juego usando la inter-

faz RGUI. Los mandos de la Xbox 360

(inalámbricos y conectados por cable)

funcionan de forma nativa en el sistema

operativo Android sin drives adiciona-

les, aunque otros mandos USB también

se puede conectar del mismo modo.

por Rob Roy

CONFIGURAR LOS MANDOS XBOX 360

En la pantalla de opciones inicial de RetroArch, selecciona "Settings", haz clic en la pestaña "Input" y asegúrate que "Onscreen Overlay" esta activado. Haz clic con el botón derecho del ratón y utiliza la opción "Load Content (Detect Core)" para iniciar tu emulador favorito. Una vez dentro del juego, haz clic en el símbolo RetroArch con el botón izquierdo del ratón, a continuación, pulsa el botón "A" sobre la superposición dpad. Utiliza el dpad sobre la superposición para resaltar "Settings", luego presiona "A" de nuevo. Selecciona "Input Options" y pulsa "A" una vez más, se mostrará el menú de configuración del mando.

ETRO fIRCH

Utiliza dpad en pantalla para que coincidan las opciones que se muestran en la pantalla, asegúrate de ajustar el "Analog D-pad Mode" a "Left Analog", para que los movimientos del joystick sean reconocidos. Por último, haz clic en "Configure All" y pulsa los botones solicitados en el mando Xbox 360. Repite este proceso para cada mando.

Una vez configurados los mandos, pulsa el botón derecho (B) del mando de Xbox 360. Selecciona "Resume Content", utiliza el botón del ratón para reducir al mínimo la superposición de RetroArch, y ¡Ya estás listo para jugar!



ODROID SMART POWER USO Y ANALISIS DE PROTOCOLOS

Editado por Venkat Bommakanti

na de las principales áreas de desarrollo de los sistemas embebidos es el Análisis de Requisitos de Energía del Sistema. Además de los muchos test de validación, es esencial para garantizar que el sistema en su conjunto se ejecute dentro de los parámetros del diseño, una potencia adecuada. La solución Smart Power para ODROID es una herramienta ideal para esta finalidad. Se trata de una fuente de alimentación inteligente regulable que, periódicamente puede recopilar/ mostrar/enviar cargas de voltaje, corriente y tensión al sistema, para el análisis y la optimización del consumo de energía.

Este artículo te enseñara a usar esta solución (hardware y software desarrolladoa por Hardkernel), tanto en Windows (7 +) y Lubuntu (kernel3.8.13) y a utilizar el popular software analizador de protocolos de código abierto Wireshark (y la línea de comandos equivalente Tshark) con un módulo USB de análisis del protocolos, específico para Lubuntu. Analizar el tráfico de datos por USB es útil para estudiar el protocolo de comunicación, mejorar el firmware y depurar errores si fuese necesario. Para Windows también existen varias herramientas de análisis de protocolos.

Requisitos

I. El paquete completo Smart Power vI.O

2. Un ordenador de placa reducida ODROID, como un U3, cuyos requisitos de energía se pueden analizar. La fuente de alimentación y cables incluido en el paquete Smart Power deberían ser suficientes para inicial el U3.

3. Puede ser necesario un adaptador para utilizar el dispositivo en tu país.

4. Una tarjeta MicroSD 8+ GB o módulo eMMC que contenga la última imagen Lubuntu disponible desde el sitio web de Hardkernel en http://www.hardkernel.com.

5. Cable microUSB a USB para la transferencia de datos entre el U3 y el ordenador principal.

6. Un ordenador para recopilar los datos trasmitidos por el Smart Power, como un PC con Windows 7+. Este ordenador también puede ser un ODROID U3 (incluso uno de los analizados) ejecutando Lubuntu.

7. Un monitor HDMI compatible para ser utilizado con U3 o acceso VNC al U3 a través de utilidades como TightVNC desde el PC principal.

8. Software Wireshark y tshark Ver. 1.10+, disponible para Ubuntu y Debian 9. Código fuente de la aplicación de monitorización Smart-Power Ver. 1.1.0 de Hardkernel
10. Librerías MinGW 0.6., QT 4.8.6, y Qwt 6.1.0 para Windows (si usas una máquina con Windows)

Configuración del dispositivo

Asegúrate que la alimentación principal está estabilizada y protegida contra subidas de tensión, y que todos los dispositivos están conectados a tierra correctamente. Conecte el PC principal o U3 a una red cableada funcional para que pueda acceder a Internet. Coloque la fuente de alimentación proporcionada 12V 3A al Smart Power y encienda el dispositivo. Espere a que la pantalla muestre los parámetros de potencia, tales como el voltaje. Ajusta el regulador de voltaje de salida a 5.01V (lige-ramente superior a 5 V), puesto que vamos a analizar los requisitos de potencia de un U3.

Conecte el cable de datos micro USB a la PC principal o al propio U3, luego conecte la pantalla HDMI al U3 si la hay. Conecte los terminales rojo y negro del cable DC (2,5 mm/0,8 mm para U3) al dispositivo Smart Power y a la clavija de energía en el U3. Conecte la alimentación y asegúrate que el U3 realiza un proceso de arranque adecuado. Si no dispones de un monitor HDMI, acceder al U3 a través de VNC o SSH para analizar el proceso.

Aplicación de Monitorización para Windows



Hardkernel ha desarrollado una aplicación de monitorización de código abierto llamada SmartPower que funciona con el dispositivo Smart Power. Aunque se facilita un ejecutable precompilado, voy a describir el proceso de desarrollo de la aplicación a partir de su código fuente. Siga los pasos para

SMART POWER

crear el entorno de desarrollo y compilar la aplicación.

MinGW es un entorno de desarrollo minimalista (GNU) para aplicaciones de Windows. Accede al sitio web http://www. mingw.org/ y haga clic en el botón Download Installer en la parte superior derecha de la página web para descargar y ejecutar la utilidad mingwget-setup.exe. Selecciona las opciones por defecto donde proceda y MinGW se instalará en C:\MinGW. Por último, añade C:\MinGW\bin a la variable de entorno PATH, que es la ubicación de la herramienta mingw32-make.exe

Qt es una aplicación multiplataforma y entrono UI para desarrollar aplicaciones C++. Descarga la última versión 4.8.x desde http://bit.ly/1ru4Jsk. Instálalo en C:\Qt y añade C:\ Qt\4.8.4\bin a la variable de entorno PATH para que la herramienta qmake.exe puede ser localizada.

La librería Qwt contiene componentes GUI y clases utility que son principalmente útiles para los programas que presentan datos técnicos. La última versión 6.1.0 se puede descargar desde http://bit.ly/1quAoaY, extráela a C:\qwt-6.1.0 y añade C:\qwt-6.1.0\lib a la variable de entorno PATH, que debería coincidir con las variables PATH que se muestran en la imagen.



Compila qwt-6.1.0 desde una nueva instancia cmd, usando los siguientes comandos:

```
> cd C:\qwt-6.1.0
> qmake
> make
```

- > make install
- > qmake -set QMAKEFEATURES C:\qwt-6.1.0\features

Descargar el código fuente de la aplicación de monitorización SmartPower para PC desde http://bit.ly/1DKSGw0. Extráela en C:\smartpower_source, luego compílala desde una nueva instancia cmd, utilizando los siguientes comandos:

```
> cd smartpower_source\HIDAPI
```

- > qmake
- > make -f MakeFile.Release

```
> cd ..\smartpower_source\smartpower
```

- > qmake
- > make -f MakeFile.Release

Tras una correcta compilación, la aplicación de monitorización que has creado se encuentra en la siguiente ubicación:

C:\smartpower_source\smartpower\windows\SmartPower.

Dado que el dispositivo Smart Power ya ha sido configurado, esta aplicación ya puede ser iniciada. Fíjate en la captura de pantalla -Vista Amperios- que muestra el estado de iniciación y la versión de firmware. Al marcar la casilla de verificación de registro se iniciará el registro de los datos capturados.

Al hacer clic en el botón Watt Graph te lleva a la Vista Vatios, como se muestra en la captura de pantalla.

Si aparece un error durante la configuración inicial, verás un mensaje indicando por qué se ha producido el error:

- El cable de datos USB entre el Smart Power y el ordenador principal está desconectado

- Se está usando un cable de datos USB defectuoso

Vista Vatios



Vista Amperios



- El puerto USB en el equipo principal o en el Smart Power no está funcionando correctamente

- Ejecución defectuosa del firmware o del Sistema Operativo

El uso de Wireshark como programa de utilidades (tratado más adelante) pueden ayudar a reducir algunos de estos errores.

Aplicación de Monitorización para Lubuntu

La aplicación de monitorización SmartPower necesita un adecuado entorno de desarrollo, que se puede instalar ejecutando el siguiente comando (una línea):

```
$ sudo apt-get install qt4-default qt4-designer
libqwt-dev libusb-1.0-0-dev
```

Descargar y descomprimir el código fuente de SmartPower usando los siguientes comandos:

```
\ cd \sim && mkdir src && cd src && mkdir sp && cd sp
```

- \$ mv ~/Downloads/smartpower_source.zip .
- \$ unzip smartpower_source.zip
- \$ cd smartpower_source

Las versiones de las herramientas de desarrollo instaladas se puede comprobar escribiendo:

```
$ uic -version
Qt User Interface Compiler version 4.8.6
$ qmake -version
QMake version 2.01a
Using Qt version 4.8.6 in /usr/lib/arm-linux-gnueabi-
hf
$ make -version
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying
conditions.
There is NO warranty; not even for MERCHANTABILITY or
FITNESS FOR A
PARTICULAR PURPOSE.
This program built for arm-unknown-linux-gnueabihf
* Unce
* Note
*
```

La aplicación SmartPower puede ser compilada:

\$ cd HIDAPI

\$ qmake

- \$ make
- \$ cd ../smartpower
- \$ uic smartpower.ui > ui_smartpower.h
- \$ qmake
- \$ make

Crea el tan importante archivo udev con el siguiente comando y valores. Sigue los consejos de los comentarios.

\$ sudo vi /etc/udev/rules.d/99-hiid.rules

- # This is a sample udev file for HIDAPI devices which
- # changes the permissions
- # to 0666 (world readable/writable) for a specified
- # device on Linux systems.
- # If you are using the libusb implementation of hi
- # dapi (hid-libusb.c), then
- # use something like the following line, substituting
- # the VID and PID with
- # those of your device. Note that for kernels before
- # 2.6.24, you will need
- # to substitute "usb" with "usb device". It
- # shouldn't hurt to use two lines
- # (one each way) for compatibility with older
- # systems.
- # HIDAPI/libusb
- # SUBSYSTEM=="usb", ATTRS{idVendor}="04d8",
- # ATTRS{idProduct}=="003f", MODE="0666"
- # If you are using the hidraw implementation, then do
 # something like the
- # following, substituting the VID and PID with your
- # device. Busnum 1 is USB.
- # HIDAPI/hidraw
- KERNEL=="hidraw*", ATTRS{busnum}=="1",

```
ATTRS{idVendor}=="04d8", ATTRS{idProduct}=="003f", MODE="0666"
```

Once done, _optionally_ rename this file for your # device, and drop it into # /etc/udev/rules.d and unplug and re-plug your # device. This is all that is # necessary to see the new permissions. Udev does # not have to be restarted. # Note that the hexadecimal values for VID and PID

```
# are case sensitive and
# must be lowercase.
```

If you think permissions of 0666 are too loose, then see:

SMART POWER

```
# http://reactivated.net/writing_udev_rules.html for
```

- # more information on finer
- # grained permission setting. For example, it might
- be sufficient to just #
- # set the group or user owner for specific devices
- # (for example the plugdev
- # group on some systems).

Cambia los privilegios de este archivo mediante el comando:

\$ sudo chmod 0666 /etc/udev/rules.d/99-hiid.rules

Tras reiniciar el sistema, la aplicación de monitorización SmartPower se puede ejecutar mediante los comandos:

\$ cd ~/src/sp/smartpower_source/smartpower/linux/ SmartPower

La herramienta de monitorización para Lubuntu tiene una interfaz de usuario idéntica a la de Windows. Las imágenes del ejemplo de Windows son aplicables en este caso.

La información del dispositivo USB y los detalles del dispositivo Smart Power pueden ser reportados con estos comandos (suponiendo bus 1, dispositivo 19):

\$ odroid@u3-2:/etc/udev/rules.d\$ lsusb Bus 001 Device 019: ID 04d8:003f Microchip Technology, Inc.

Use Bus/Device information from previous command \$ lsusb -D /dev/bus/usb/001/019 output: Device: ID 04d8:003f Microchip Technology, Inc. Device Descriptor:

	bLength	18		
	bDescriptorType		1	
	bcdUSB	2.00		
	bDeviceClass		0	(Defined at Interface
1	evel)			
	bDeviceSubClass		0	
	bDeviceProtocol		0	
	bMaxPacketSize0		8	
	idVendor	0x04d8	Mi	crochip Technology,
I	nc.			
	idProduct	0x003f		
	bcdDevice	0.02		
	iManufacturer		1	Microchip Technology
I	nc.			
	iProduct		2	Simple HID Device Demo
	iSerial		0	
	bNumConfigurations	1		
	Configuration Descr	iptor:		

bLength		9	
bDescriptorType		2	
wTotalLength		41	L
bNumInterfaces		1	
bConfigurationVa	lue	1	
iConfiguration		0	
bmAttributes		02	<c0< td=""></c0<>
Self Powered			
MaxPower		10) OmA
Interface Descr	iptor:		
bLength		9	
bDescriptorType		4	
bInterfaceNumbe	r	0	
bAlternateSetti	ng	0	
bNumEndpoints		2	
bInterfaceClass		3	Human Interface Device
bInterfaceSubCl	ass	0	No Subclass
bInterfaceProtocol	0 None		
iInterface	0		
HID Device Descriptor	:		
bLength	9		
bDescriptorType	33		
bcdHID	1.11		
bCountryCode	0 Not si	up	ported
bNumDescriptors	1		
bDescriptorType	34 Repo	rt	
wDescriptorLength	28		
Report Descriptors:			
** UNAVAILABLE **			
Endpoint Descriptor:			
bLength	7		
bDescriptorType	5		
bEndpointAddress	0x81 E	P	1 IN
bmAttributes	3		
Transfer Type	Interru	pt	
Synch Type	None		
Usage Type	Data		
wMaxPacketSize	0x0040	1	x 64 bytes
bInterval	1		
Endpoint Descriptor:			
bLength	7		
bDescriptorType	5		
bEndpointAddress	0x01 E	P	1 OUT
bmAttributes	3		
Transfer Type	Interru	pt	
Synch Type	None		
Usage Type	Data		
wMaxPacketSize	0x0040	1	x 64 bytes
bInterval	1		
Device Status:	0x0001		
Self Powered			

Instalación de Wireshark

El Wireshark precompilado y los paquetes de línea de comandos TShark pueden instalarse tecleando los siguientes comandos, luego reinicia el sistema:

- \$ cd ~/
- \$ sudo apt-get install build-dep wireshark
- \$ sudo dpkg-reconfigure wireshark-common
- \$ sudo apt-get install tshark

La versión instalada de estas utilidades se debería comprobar después de reiniciar:

\$ wireshark --version
wireshark 1.10.6 (v1.10.6 from master-1.10)

Copyright 1998-2014 Gerald Combs <gerald@wireshark. org> and contributors. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compiled (32-bit) with GTK+ 3.10.7, with Cairo 1.13.1, with Pango 1.36.1, with GLib 2.39.91, with libpcap, with libz 1.2.8, with POSIX capabilities (Linux), without libnl, with SMI 0.4.8, with c-ares 1.10.0, with Lua 5.2, without Python, with GnuTLS 2.12.23, with Gcrypt 1.5.3, with MIT Kerberos, with GeoIP, with PortAudio V19-devel (built Feb 25 2014 21:10:47), with AirPcap.

Running on Linux 3.8.13.27, with locale en_US.UTF-8, with libpcap version 1.5.3, with libz 1.2.8, GnuTLS 2.12.23, Gcrypt 1.5.3, with-out AirPcap.

Built using gcc 4.8.2.

\$ tshark --version
TShark 1.10.6 (v1.10.6 from master-1.10)

Copyright 1998-2014 Gerald Combs <gerald@wireshark. org> and contributors. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. Compiled (32-bit) with GLib 2.39.91, with libpcap, with libz 1.2.8, with POSIX capabilities (Linux), without libnl, with SMI 0.4.8, with c-ares 1.10.0, with Lua 5.2, without Python, with GnuTLS 2.12.23, with Gcrypt 1.5.3, with MIT Kerberos, with GeoIP.

Running on Linux 3.8.13.27, with locale en_US.UTF-8, with libpcap version 1.5.3, with libz 1.2.8.

Built using gcc 4.8.2.

Para la versión de Lubuntu seleccionada, el driver usbmon ya está presente y se puede cargar y comprobar expresamente:

\$ sudo mount -t debugfs none_debugs /sys/kernel/debug mount: none_debugs already mounted or /sys/kernel/ debug busy

mount: according to mtab, none is already mounted on
/sys/kernel/debug

\$ sudo modprobe usbmon

\$ sudo ls /sys/kernel/debug/usb/usbmon
0s 0u ls lt lu 2s 2t 2u

Privilegios de acceso

El acceso a los dispositivos usbmon normalmente solo es posible para superusuarios tales como root. Sin embargo, no es aconsejable ejecutar analizadores de protocolos utilizando privilegios de root. Para garantizar el acceso seguro de las interfaces usbmon para Wireshark y tshark, se debe dar al usuario los privilegios adecuados mediante los siguientes comandos:

\$ sudo addgroup -system wireshark

dev/usbmon0

\$ sudo usermod -a -G wireshark odroid

```
$ sudo chgrp wireshark /dev/usbmon*
$ sudo ls -lsa /dev/usbmon*
0 crw------ 1 root wireshark 248, 0 sep 14 15:23 /
dev/usbmon0
0 crw------ 1 root wireshark 248, 1 sep 14 15:23 /
dev/usbmon1
0 crw------ 1 root wireshark 248, 2 sep 14 15:23 /
dev/usbmon2
$ sudo chmod g+r /dev/usbmon*
$ sudo ls -lsa /dev/usbmon*
0 crw-r---- 1 root wireshark 248, 0 sep 14 15:23 /
```

SMART POWER

0	crw-r	T	root	wireshark	248,	Ţ	sep	14	15:23	/
de	ev/usbmon1									
0	crw-r	1	root	wireshark	248,	2	sep	14	15:23	/
de	ev/usbmon2									

Uso de Wireshark

La utilidad Wireshark puede iniciarse directamente (sin sudo) con el siguiente comando para mostrar las interfaces usbmon como se muestra a continuación.

\$ wireshark

Para averiguar la interfaz usbmon asociada al dispositivo Smart Power, haz clic en la opción "Interface List" en la sección "Capture" de la pantalla principal de Wireshark. Se puede observar inmediatamente que una de las interfaces usbmon muestra una alta tasa de transferencia (USB). Como puede verse en la pantalla resulta ser la interfaz usbmon1, que está asociada al dispositivo Smart Power (suponiendo que ningún otro dispositivo USB esté conectado al ordenador principal).

Haga clic en el botón "Start" e iniciar el proceso de captura.

● ● 📕 🦉 🖆 📋 × ୯ । प्	✓ > ♥ 示 ⊻ □ □ □ □ ✓ Expression Clear Apply Save	1 🔛 🎬 🖾 🏥 🌣 🤉					
WIRESHARK The World's Most Popular Network Protocol Analyzer Version 1.10.6 (v1.10.6 from master-1.10)							
Capture	Files	Online					
Interface List Less of the capture interfaces tocurs incoming packets) Start Choose one or more interfaces to capture from, then Start those one or more interfaces to capture from, then Start tho g for queue integrad thomas interface integrad	Open Open a previously captured file Open Recent: Sample Captures A rich assortment of example capture files on the wiki	Website Visit the project's website User's Guide The User's Guide (online version) Security Work with Wireshurk as securely as possible					
Ready to load or capture No Packets	Profile: Default						

Pantalla principal de Wireshark

Ten en cuenta que el proceso de captura puede resultar muy rápido en archivos muy grande. Desplázate por la parte superior, donde una línea corresponde a la captura de un conjunto de datos. Puedes ver que algunas de las líneas corresponden a los datos sobre voltios, amperios, vatios y vatios-hora.

Tras 100 puntos de datos, puedes detener el proceso de captura y guardar los datos en un archivo para su análisis posterior. Un estudio más extenso dará detalles sobre el protocolo. Este conocimiento puede ser útil para comprender el funciona-

	Device	Descripti	on IP	Packets	Packets/s	
	eth0		none	0	0	
	nflog		none	0	0	
	nfqueue	High packat	none	0	0	
	usbmon1 💳 >	capture from	□ > none	1826	52	
	usbmon2	Sinarcrower	none	10	0	
	any		none	0	0	
	lo		127.0.0.1	0	0	
? Help	Start	Stop	Opti	ons	X Close	

miento del dispositivo Smart Power y depurar actualizaciones de firmware si fuese necesario.

Tshark

El equivalente de línea de comandos a Wireshark llamado tshark, se puede activar con el comando:

nuer	÷ (Expression	Clear Appl	y Save					
ο.		Time	Source			D	estina	tion	Pro	tocol Length	Info					
2	247	5.096485000	host			8.	1		USB	128	URB_INT	TERRUPT	out			
2	248	5.097921000	8.1			ho	ost		USB	64	URB INT	TERRUPT	out			
2	249	5.098927000	8.1			ho	ost		USB	128	URB INT	TERRUPT	in			
2	250	5.098991000	host			8.	1		USB	64	URB INT	TERRUPT	in			
2	251	5.175920000	7.2			ho	ost		USB	69	URB INT	TERRUPT	in			
	252	5 175970000	host			7	-		LICE		UDD THE	TERRUPT	4.4			
ram ISB eft	URB	247: 128 byte 8 er Capture Da	es on win sta: 813	e (10)	24 bi1 330562	s), 1 02a30	28 byt 283236	es cap 332041	035 tured (1024 20312e333433	64 bits) on inte 5720	erface 6)	TU			
ram ISB .eft	URB OVe	247: 128 byte ar Capture Da	es on win ata: 813	e (10)	24 bi1 330562 01 01	s), 1 02a30	28 byt 2e3236	es cap 332041 d 00	tured (1024 20312e333433	64 bits) on into 5720	erface 6)	10			
ram SB eft	002 002 002 002	247: 128 byte er Capture Da e2 d3 e2 06 56 15 54 06	es on win ata: 813 00 00 0	e (10)	24 bi1 33056; 01 01 04 0f	s), 1 02a30 08 0 08 0	28 byt 2e3236 1 00 2 1 ff f	es cap 332041 d 00 f ff	tured (1024 20312e333433	64 bits) on int 5720	erface 6)	10			
ram SB eft 90 10 20	00 00 00 00 00 28 40	e2 d3 e2 00 56 15 54 00 00 00 00 40	es on wi ata: 813 00 00 0 00 00 0	e (10)	24 bi1 330562 01 01 04 0f 00 00	s), 1 02a30 08 0 00 80 00 00	28 byt 2e3236 1 00 2 1 ff f 9 00 0	es cap 332041 d 00 f ff 0 00	tured (1024 20312e333433 S. (V.T @@	54 bits) on inte	erface 6)	10			
ram SB eft 90 10 20 30	002 002 002 002 002 002 002 002 002 002	e2 d3 e2 00 56 15 54 00 00 00 00 00	es on wi ata: 813 00 00 0 00 00 0 00 00 0	e (10) 2e313 0 53 0 53 0 65 0 00 0 00	24 bi1 330562 01 01 04 0f 00 00 00 00	s), 1: 02a30: 08 0: 00 8: 00 8: 00 0:	28 byt 2e3236 1 00 2 1 ff f 9 00 0 9 00 0	es cap 332041 d 00 f ff 0 00 0 00	tured (1024 20312e333433 	5720	erface 6)	10			
ram SB eft 10 20 30 40	00 00 00 28 40 01 81	e2 d3 e2 06 56 15 54 06 00 00 06 46 35 2e 31 33	es on win ata: 813 00 00 00 00 00 00 00 00 00 00 00 00 00 00	e (10) 2e313 0 53 0 d5 0 00 0 00 0 2a	24 bi1 330562 01 01 04 0f 00 00 30 20	(1) (1) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	28 byt 2e3236 1 00 2 1 ff f 3 00 0 3 00 0 5 33 2	es cap 332041 d 00 f ff 0 00 0 00 0 41	20312e333433 	5720	erface 6)	10			
eft 00 10 20 30 40 50	00 00 00 00 00 00 00 00 00 00 00 00 00	e2 d3 e2 00 56 15 54 00 00 00 00 40 35 2e 31 33 31 2e 33 34	es on win ata: 813 00 00 00 00 00 00 00 00 00 00 00 00 00 00	e (10) 2e313 0 53 0 d5 0 00 0 00 0 2a 0 20 2 20	24 bi1 330562 01 01 04 0f 00 00 30 26 30 26 30 26	s), 1 02a30 00 80 00 80 00 00 32 36 34 31	28 byt 2e3236 1 00 2 1 ff f 9 00 0 5 33 2 1 37 5	es cap 332041 d 00 f ff 0 00 0 00 0 41 7 68		64 bits) on intr 5720 	captured		>			
eft 00 10 20 30 40 50 60 70	00 00 28 40 01 20 01	(47: 128 byt) (47: 128 byt) (47: 128 byt) (47: 128 byt) (56 15 54 00 (00 00 00 40 (00 00 00 04 40 (00 00 00 00 04 40 (00 00 00 00 00 04 (00 00 00 00 00 00 (00 00 00 00 00 (00 00 00 00 00 (00 00) (00 00 00 (00 00) (00 00 00 (00 00) (00 00)	es on win ata: 813 0 00 00 0 0 00 00 0 00 00 0 00 00 0 00 0	e (10) 52e313 0 53 0 05 0 00 0 2a 0 20 2 98 0 00	24 bi1 330562 01 01 04 0f 00 00 30 26 30 26 52 6	s), 1 02a30 08 01 00 80 00 00 32 30 34 31 aa 10	28 byt 283236 2 00 2 1 00 2 1 ff f 3 00 0 3 00 0 5 33 2 1 37 5 9 20 0 2 00 0	es cap 332041 d 00 f ff 0 00 0 00 0 41 7 68 2 12 0 00	035 tured (1024 20312e333433 (V.T 00 (V.T 00 00 1.343W 0 	64 bits) on inte 5720 263 A Data 263 A Data	captured Smart Powe		>			

Wireshark con datos capturados desde el Smart Power

\$ tshark -D
1. eth0
2. nflog
3. nfqueue
4. usbmon1
5. usbmon2
6. any
7. lo (Loopback)

Ten en cuenta que la invocación basada en sudo no necesita mostrar las interfaces usbmon, debido a la correcta configuración de privilegios de acceso. Tshark se puede utilizar para capturar los datos a través de la interfaz usbmon2 (véase notas de Wireshark) con el siguiente comando para que el archivo 1.pcap se pueda ver en Wireshark directamente.

\$ tshark -i usbmon2 -w 1.pcap

Otras aplicaciones de monitorización

El miembro de la Comunidad @muehlbau ha proporcionado una útil utilidad de línea de comandos para monitorizar el dispositivo Smart Power, que se puede instalar escribiendo la siguiente serie de comandos en una ventana de terminal:

```
$ sudo apt-get install build-dep pkg-config libusb-
1.0-0-dev
$ cd ~/src && mkdir mu && cd mu
$ git clone https://github.com/muehlbau/odroid-smart-
power-linux
```

Compila la utilidad:

Interfaz Wireshark

```
$ cd odroid-smartpower-linux
$ make
```

Luego puede ser lanzada con el comando smartpower (que requiere privilegios de root) y los resultados se escriben en un archivo log:

```
$ sudo ./smartpower capture-logfile
```

El miembro de la comunidad @pcat ofrece otra útil herramienta de línea de comandos para monitorizar el dispositivo Smart Power. Se puede obtener su código fuente desde github con los comandos:

```
$ cd ~/src && mkdir sp-cll && cd sp-cll
$ git clone https://github.com/polarcat/smartpower
$ cd smartpower
$ make
```

Es muy util para detectar los dispositivos hidraw con el comando:

\$ sudo find /dev/hid*
/dev/hidraw0
/dev/hidraw1
/dev/hidraw2
/dev/hidraw3

El comando anterior se puede ejecutar con o sin el dispositivo Smart Power conectado. en nuestra prueba, indicó que hidraw3 corresponde al dispositivo Smart Power. Se puede comprobar con el siguiente comando:

```
odroid@u3-2:~/src/sp-cl1/smartpower$ cat /sys/class/
hidraw/hidraw3/device/uevent
DRIVER=hid-generic
HID_ID=0003:000004D8:0000003F
HID_NAME=Microchip Technology Inc. Simple HID Device
Demo
HID_PHYS=usb-s5p-ehci-3.2.6/input0
HID_UNIQ=
MODALIAS=hid:b0003g0001v000004D8p0000003F
```

El binario SmartPower generado se puede iniciar usando uno de los comandos para observar el resultado captado periódicamente.

```
$ cd ~/src/sp-cl1/smartpower
$ sudo ./smartpower
$ sudo ./smartpower -v -d /dev/hidraw3
```

Las opciones pueden ser listadas usando el comando help:

\$ sudo ./smartpower --help

Para hacer frente a un bug en el resultado y algunos otros problemas menores de formato de salida, he creado un parche para que el Smart Power sea autodectado, eliminado la necesidad de especificar al ruta del dispositivo hidraw (opción -d). El parche incorpora una relación de los cambios en smartpower.c, que se detallan a continuación:

```
85d85
< static char detected dev[32] = { '\0', };
97c96
<
              printf("0.%06u%c%s\n", 0, sep, data);
___
              printf("0.0%c%s\n", sep, data);
169c168
      printf("Version: %s\n\n", buf);
<
>
      printf("Version: %s\n", buf);
188,191d187
<
      if (errno == 22) {
              printf("(=) Check invocation
syntax\n");
<
              exit(errno);
<
     }
294a289
> i++;
302,303c298
<
              i++;
<
             snprintf(detected dev, sizeof(detected
dev), "%s", name);
              fprintf(stderr, "Detected smartp at
%s\n", name);
309,312c304,305
     if (i == 0) {
      printf("(=) Smart Power device is disconnect-
<
ed. Please connect and retry\n");
      exit(1);
<
<
             }
____
      if (i == 0)
      printf("(=) smart power device is not
connected\n");
371c364
<
      printf(" -d, --dev <dev>
                                     path to hidraw
device node (/dev/hid*) \n");
>
      printf(" -d, --dev <dev>
                                    path to hidraw
device node\n");
387,389d380
<
    csv = 0;
```

SMART POWER

CONSEJ<u>OS Y TRUCOS</u>

```
sep = ' ';
<
<
416,418c406,410
                continue;
<
<
                if (opt(arg, "-c", "--csv")) {
<
>
                csv = 0;
                sep = ' ';
>
                continue;
>
>
       }
>
       if (opt(arg, "-c", "--csv")) {
430,438c422,430
      fd = smartp_probe();
<
      if (dev) {
<
       fd = smartp open(dev);
<
       if (fd < 0)
<
<
                return fd;
<
      }
<
<
      if (verbose == 1) {
       printf("Detected dev: [%s], requested dev:
<
[%s]\n", detected dev, dev);
___
>
      if (dev)
       fd = smartp_open(dev);
>
>
      else
       fd = smartp_probe();
>
>
      if (fd < 0)
>
>
       return fd;
>
      if (verbose == 1) {
>
```

Para información adicional o realizar preguntas, por favor visitealas fuentes de información originales en:

http://bit.ly/lvpEdkh
http://bit.ly/loomVis
http://bit.ly/luwjVaP
http://www.mingw.org
http://www.qt-project.org
http://qwt.sourceforge.net
http://code.wireshark.org
http://bit.ly/lBriMjG
http://bit.ly/llh6G8v

TODO SOBRE DEBIAN UNA GRACIOSA INFOGRAFIA



editado por Nicole Scott

cha un vistazo a una asombrosa infografía sobre el funcionamiento interno del sistema operativo Debian en http:// bit.ly/1vJHXgB. Ilustra todo lo relacionado con el proceso de desarrollo, y la forma de participar en el proyecto Debian

Understanding Debian The universal operating system



ADAPTA TU ESCRITORIO DE LINUX PARA MONITORIZAR EL RENDIMIENTO Y EL TIEMPO USANDO CONKY Y HARMATTAN

por Jussi Opas

s importante para muchas aplicaciones monitorizar los recursos del sistema, como el uso del procesador, la actividad del disco, espacio de almacenamiento y el tráfico de red. Este último es especialmente importante cuando se transfieren grandes archivos, tales como la descarga de una imagen del sistema operativo o una actualización del sistema. Aunque Gnome System Monitor se puede utilizar para este propósito, si uno quiere medir las estadísticas de tráfico de red tanto en la red inalámbrica como por cable de forma individual, no es posible. El Monitor de sistema consume recursos, de modo que sería bueno tener una alternativa más configurable y liviana, como conky. También queremos medir las características específicas de ODROID como el uso de clúster Big.little. En este artículo describimos nuestra experiencia al usar Conky en Linux en un ODROID-XU y en un ODROID-U3.

Introducción

Conky existe desde hace varios años, como se puede ver al revisar los registros de desarrollo en http://bit.ly/1mk0gHu. Puedes encontrar en Internet muchas capturas de pantalla donde Conky se incluye como parte del escritorio. En nuestros ensayos, hemos adaptado la monitorización del rendimiento y el tiempo con Conky en dos ODROIDs y tres sistemas operativos, como se muestra en la siguiente tabla:

OS	version	ODROID
Xubunu	13.10	XU
Debian	7, Ezy	U3
	Wheezy	
Lubuntu	14.04	U3

Instalación

Conky puede ser descargado y iniciado escribiendo el siguiente comando en una ventana de terminal:

\$ sudo apt-get install conky && conky

Tras cargar el programa, se muestra el monitor por defecto Conky, donde se puede definir qué y cómo mostrar los elementos a monitorizar. El archivo de configuración se guarda en el directorio ~/.conkyrc. Para saber más sobre .conkyrc, visite http://bit.ly/1CSX6Qd.

Monitor de rendimiento

Cuando observamos la CPU de un ODROID, nos interesamos por la frecuencia, el regulador de nivel de frecuencia, la temperatura, el clúster usado y la utilización. El regulador puede ser cogido fácilmente desde los archivos del kernel. La función nativa conkyfreq_g se utiliza para obtener la frecuencia actual, aunque también podría ser obtenida directamente del archivo.

Para mostrar la temperatura en la escala correcta, tenemos que dividir el resultado por 1000. Podemos usar la calculadora binaria bc para ello:

\${exec echo "scale=1; \$(cat /
sys/class/thermal/thermal_zone0/
temp)/1000" | bc}

Tuvimos que instalar be en Debian por separado con apt-get, ya que no lo incorpora por defecto. Como alternativa, también se podría usar **awk** para dividir por 1000 y obtener te temperatura correcta.

Para ODROID, añadimos definiciones especiales para mostrar la utilización de los 4 núcleos. El uso de cada núcleo puede mostrarse por separado en forma de barra con el siguiente ajuste:

core1 \${cpu cpu0}% \${cpubar
cpu0}

Con el XU en especial, también queremos saber qué clúster de núcleos se está usando (big o LITTLE). La siguiente instrucción determina qué clúster de



temperatura y la información de red

núcleos está en uso:

\${if_match \${exec cat /dev/bL_
status | grep A7 | cut -c18} < 1}
big\${else}LITTLE\${endif}</pre>

El ejemplo también muestra, cómo expresar una sentencia if-then en un archivo de definición de conky. Lo malo de esta definición es que conky debe ser invocado con sudo, ya que el archivo / dev/bL_status no se puede acceder sin privilegios de root.

Conky por si mismo ofrece funciones como downspeedgraph para mostrar por separado el trafico Ethernet y el tráfico inalámbrico de descarga y de subida. También es posible mostrar el tráfico de forma conjunta. Nosotros hemos elegido mostrar el tráfico de eth y wlan por separado. Cuando se configura definiciones personalizadas, es útil buscar los nombres que deben ser utilizados en el archivo de definición con el comando **ifconfig** (por ejemplo, eth0 y wlan6).

La imagen de fondo se ve a través del monitor Conky, puesto que el own_ window se ha escrito en el archivo de parámetros .conkyrc. Si el monitor está



Monitorización Conky en Xubuntu de un ODOID XU con el clúster big en uso

definido para ser transparente, el fondo de pantalla delimita si la fuente utilizada es legible y distinguible. Para cambiar los colores usados, se puede definir en el archivo de configuración de conky como sigue:

color8 888888

Debido a que el color puede variar a lo largo y ancho del fondo de escritorio, esto no garantiza que el texto sea siempre de fácil lectura. En cambio, cuando se utiliza una ventana, el fondo definido siempre es el mismo y el contenido del monitor siempre será visible.

Monitor de tiempo

En Internet, hay un montón de ejemplos de cómo usar Conky como parte de un escritorio. Además, puede mostrar el tiempo actual y alguna previsión meteorológica. Hemos experimentado con Harmattan Conky y lo hemos adaptado para que funcionase en las imágenes de Xubuntu, Lubuntu y Debian.

Para variar, el pack Harmattan Conky ofrece 15 estilos diferentes con diferentes modos, que se pueden descargar desde http://bit.ly/1rrxV20.

Los monitores de tiempo Harmattan vienen con varios tamaños, y la imagen de fondo del tiempo cambia de forma dinámica con la temperatura y el tipo de clima. Si algo sale mal o no consigues que Harmattan funcione a la primera, es mejor desinstalar e instalar de nuevo.

Cuando se instala por primera vez es



en Conky Harmattan

necesario buscar el código del tiempo para la ciudad deseada. Esto implica visitar la URL de Yahoo y copiar el código de ciudad.

Primero hay que ir a http://tiempo. yahoo.com y buscar la ciudad, luego copia el código de la barra de direcciones. Por ejemplo, la siguiente dirección contiene el número 44418, que es el código requerido para Harmattan.

https://weather.yahoo.com/unitedkingdom/england/london-44418/

La otra parte de la configuración de Harmattan pasa por hacer que el widget se muestre de forma apropiada y visible en cada distribución de Linux. Como no teníamos instrucciones explícitas, hemos

CONKY

usando el método de ensayo y error para encontrar nuestra propia configuración.

Los ajustes comunes que parecen funcionar en todas las plataformas se muestran en la siguiente lista:

double_buffer yes
update_interval 5
own_window yes
own_window_transparent yes

La opción double_buffer se utiliza para reducir el parpadeo y reajustamos el intervalo de actualización cada 5 segundos. En Xubuntu también usamos:

own_window_type override

Para Debian y Lubuntu, la definición específica adicional es:

own_window_hints
undecorate,sticky,skip_
taskbar,skip_pager,below

Con estas configuraciones, logramos un Harmattan estable y bien configurado, como se muestra en la captura de pantalla de Debian.

Puesta en marcha

La instalación de Harmattan agrega un archivo para iniciarse con el arranque dentro del fichero ~/.start_conky. El monitor de rendimiento personalizado puede ser invocado de manera similar. El contenido del archivo es la siguiente:

#!/bin/sh

sleep 20

Harmattan Conky funcionando en la imagen Debian Ezy Wheezy en ODROID U3



conky -d -c ~/.conkyrc
sleep 5
conky -d -c ~/conky/.conkyrc_XU
exit

Esta definición activa el monitor de tiempo 20 segundos después del arranque. El monitor conky personalizado especifico para XU se activa 5 segundos después. Harmattan usa Internet para obtener los datos de pronóstico del tiempo, es necesario dejar tiempo suficiente para que el sistema inicie y establezca una conexión inalámbrica o por cable.

Múltiples monitores

Se pueden añadir a único escritorio varios monitores de rendimiento y estadísticas. Por ejemplo, podemos estar interesados en el tiempo y el rendimiento del sistema al mismo tiempo. Una posible configuración podría ser mostrar todos los datos en un widget combinado. Si decidimos en su lugar, utilizar dos ventanas podemos cerrar el monitor de rendimiento cuando no lo necesitemos.

Notas técnias

Con el cuadro del monitor transparente, el grupo de iconos del escritorio se puede ocultar si los ajustes de la ventana son incorrectos. Los iconos ocultos sólo son visibles cuando pasamos el ratón sobre ellos, puede ser solucionado adaptando la configuración y reduciendo el tamaño de la ventana del widgets.

Archivo de configuración de muestra

Monitores de rendimiento y tiempo juntos en una imagen de Lubuntu, mostrando el tiempo brumoso y un procesador inactivo



background yes cpu avg samples 2 net avg samples 2 out to console no font 7x13 use xft no own window yes ### black own window monitor own_window_transparent no own window colour black ### transparent background monitor #own window transparent yes #own window type override #own window hints undecorate, sticky, skip taskbar, skip pager, below double buffer yes # double buffering removes flicker on bottom yes update interval 1 minimum size 5 5 draw shades no draw outline no draw borders no stippled borders 0 border margin 10 border width 2 default color white default shade color white default outline color white alignment bottom_right gap x 40 # 20 gap y 100 # 20 use spacer yes no buffers no uppercase no color2 CCCCCC color8 888888 TEXT \${color8}\${time %a %d.%b %y} \$alignr \${color green}\${time %k:%M:%S} \${color8}\$sysname \$kernel \$alignr \$machine Uptime \$alignr \$uptime \${color white}\${hr 2} \${color green}cpu \${color slate gray}frequency \${color2}\${freq g } \${color slate

CONKY

gray}GHz \${color slate gray}governor \${color2}\${exec cat /sys/devices/ system/cpu/cpu0/cpufreq/scaling governor} \${color slate gray}cluster \${color2}\${if match \${exec cat /dev/bL_status | grep A7 | cut -c18} < 1}big\${else} LITTLE\${endif} \${color slate gray}temperature \${color2}\${exec echo "scale=1; \$(cat /sys/class/thermal/thermal zone0/temp)/1000" | bc}\${color slate gray} C \${color slate gray}ulitization \${color2}\${cpu}% \${color8}\${cpugraph 25 ff0000 ff00ff} core1 \${cpu cpu0}% \${cpubar cpu0} core2 \${cpu cpu1}% \${cpubar cpu1} core3 \${cpu cpu2}% \${cpubar cpu2} core4 \${cpu cpu4}% \${cpubar

cpu3} \${color green}wlan\${color8} DOWN \${color2}\${downspeed wlan6} \${color8}UP \${color2}\${upspeed wlan6} \${color8} \${downspeedgraph wlan6 25,100 ff0000 0000ff} \${color8}\${upspeedgraph wlan6 25,100 0000ff ff0000} \${color8} TO-TAL \${color2}\${totaldown wlan6} \${color8}TOTAL \${color2}\${totalup wlan6} \${color green}eth\${color8} DOWN \${color2}\${downspeed eth0} \${color8}UP \${color2}\${upspeed eth0} \${color8} \${downspeedgraph eth0 25,100 ff0000 0000ff} \${color8}\${upspeedgraph eth0 25,100 0000ff ff0000} \${color8} TO-TAL \${color2}\${totaldown eth0} \${color8}TOTAL \${color2}\${totalup eth0} \${color green}disk \${color2}

\${diskiograph 30,220 fef7b2
e18522}
\${color8}root \${color2}\${fs_
size /} \${color8}\${fs_free_perc
/}% free \${fs_bar /}
\${color8}boot \${color2}\${fs_
size /media/boot} \${color8}\${fs_
free_perc /media/boot}% free
\${fs_bar /media/boot}

I THOUGHT YOU SAID IT WAS COLD OUT HERE



Harmattan Conky en la imagen Debian Wheezy Ezy en un ODROID U3, muestra las estadísticas de red, sistema y software en tiempo real



HARDKERNEL EN ARM TECHCON 2014

por Rob Roy

¡Tuvimos mucha diversión en la ARM TechCon 2014! Varios miembros de Hardkernel hicie-ron el viaje desde Corea del Sur, y Mauro vino desde Brasil para el gran espectáculo tecnológico de 2 días en Santa Clara, California. Hemos puesto en marcha juegos, demos e incluso, tuvimos una graciosa mascota robot Android que cantó y bailó para nosotros. Como de costumbre, el stand Hardkernel tuvo un montón de visitantes que estaban interesados en ver de lo que era capaza el XU3.



De izq. a der.: Bo, Rob Roy, Justin, Mauro, Ryan and Lisa en el stand de Hardkernel en ARM TechCon 2014

Presentamos un adelanto de la revista, además de una competición de Angry Birds.



Mauro preparó una demo de la versión 14.04 de Ubuntu para XU3, in-

> cluía una máquina virtual KVM que ejecuto Android dentro de Ubuntu y al mismo tiempo ejecutaba varias demos de gráficos y software en 3D.

> ¡Gracias a todos los que se detuvieron en el stand! Asegúrate de conseguir tus entradas a principios del año que viene para tener la oportunidad de reunirte con el equipo Hardkernel.

CONOCIENDO A UN ODROIDIAN BRUNO DOICHE: EDITOR ARTISTICO DE ODROID MAGAZINE

editado por Rob Roy

Por favor, háblanos un poco sobre ti.

Como bromee en la sección "Acerca de mí" del Número 1 de ODROID Magazine, ¡Soy una persona estupenda! como la mayoría de nuestros lectores. Creo que soy un tipo que espera con interés conocer y usar equipos que no sean de la familia Wintel (Windows).

Soy un administrativo normal que trabaja en la industria de TI gestionando servidores Unix y Linux, almacenamiento SAN y gestión de base de datos. Antes de eso, solía trabajar como fotógrafo después de ser ayudante de un fotógrafo durante un par de años. Luego trabaje durante algún tiempo en el área de edición de revistas haciendo revistas de informática y juegos. Así fue como me involucré en la revista: un día, Rob escribió en los foros preguntando si alguien le podía ayudar, pensé "en fin, puede ser divertido" y el resto los puedes leer desde entonces por aquí. Cuando tengo tiempo y mientras hago el diseño de la revista, me gusta insertar bromas tontas/

De izquierda a derecha, los 3 primeros equipos que utilice alguna vez: MSX Hotbit, TK3000 clon Apple II y MSX Expert



cómicas que utilizáis posiblemente como escusa para leer cada número.

¿Cómo fueron tus inicios con ordenadores?

Bueno, cuando yo tenía 9 años, visité a un primo y vi por primera vez en mi vida un ordenador. Era un hotbit MSX, un clon brasileño del modelo japonés Hitbit y recuerdo jugar (y superar) un juego llamado Yie Ar Kung Fu. Luego, en el siguiente año de colegio empecé a recibir clases de informática con un clon de Apple II llamado TK3000, usando LOGO y BASIC. Les di la lata a mis padres hasta que me consiguieron un ordenador por Navidad.

Siguiendo el consejo de un amigo, me conseguí un MSX. Por aquel entonces, no entendía el concepto de diferentes plataformas informáticas, y me llevo algún tiempo entender que el ordenador de mi primo era de hecho, la misma plataforma que la mía. Las cosas eran muy diferentes en aquel entonces, tenía un





amigo que contaba con un Amiga Commodore, otro que tenía un PC IBM, y otro un Apple ¡Era un ZOO! Tratamos de ejecutar software de uno a otro y era decepcionante, hasta que nos dimos cuenta de estas incompatibilidades.

En 1993 tuve mi primer PC IBM y me quedé con él hasta que llegué a la universidad, donde estudie Ciencias de la Computación, en el laboratorio trabajé con Macintosh. Abandone la universidad y mi fui directamente a trabajar con diseño y acabe graduándome como fotógrafo y diseñador. Trabaje mayoritariamente con edición, resultaba muy dinámico trabajar con material gráfico y material relacionado con los ordenadores.

Conseguí mi trabajo actual y por fin me gradué en Ciencias de la Computación y trabajé principalmente en la industria de TI. Así que me conseguí un servidor IBM Power4, luego un IBM Power5 para obtener mi certificado AIX. Pero eran demasiado ruidosos y



CONOCIENDO A UN ODROIDIAN



Arriba: Una máquina IBM Power4 que tuve una vez, tenía la mitad de potencia de procesamiento que un ODROID XU3 pero pesaba 35,5 kg (78,0 libras) frente a los 100 g (0,22 libras) del XU3. Derecha: Una matriz de almacenamiento de la misma familia Power4 con una enorme capacidad de 540GB, hoy te cogen 640 GB de eMMCs en el puño.

consumían bastante energía como para tenerlos trabajando en casa. Un amigo me habló de los ODROIDS, así que compré uno para utilizar en casa y hacer todas las cosas que nunca quería que dejaran de hacer mis equipos de escritorio/ ordenadores portátiles. Compré un X2, lo arranqué con una de las distribuciones Linux de Rob Roy y todavía está funcionando. Si no fuera por los cortes de luz ocasionales cuando llueve o cosas por el estilo, el X2 estaría encendido permanentemente desde entonces.

¿Qué tipos de proyectos has realizado con tus ODROIDs?

No puedo vivir sin una red en casa, así que se hace imprescindible un servidor de archivo personal. Conseguí el X2 para transcodificar vídeo a mi playstation 3, fue un proyecto divertido. Otro fue la creación de un Shairport para





transmitir audio desde iTunes a un buen altavoz estéreo. Recientemente, tengo un nuevo Mac y exporto todos mis mp3 al X2. Ahora también funciona como servidor de iTunes con miles de canciones.

Uso el X2 como máquina torrent, y ejecuto un programa llamado Sick Rage para obtener todos los programas de televisión que quiero seguir. También hice un simple script con crontab para buscar los subtítulos de mis programas favoritos. Hay algunos contenedores y KVM que necesitan virtualización pero necesito algo de tiempo para hacerlo. Openstack con mi U3s es otro de mis proyecto, y todavía estoy buscando la forma de poner mis ODROIDs en una caja compacta alimentada por una sola fuente de alimentación ATX, pero no le estoy dedicando demasiado tiempo debido a mi habitual trabajo. Conectar el U3 a un Atrix Motorola también puede resultar divertido.

¿Qué otras aficiones e intereses tienes ?

Me encanta montar en bicicleta! Tengo 3 bicicletas, y cuando voy en bicicleta regularmente y escucho música al mismo tiempo hace que me sienta realmente feliz. Colecciono discos de vinilo, tanto antiguos como nuevos, y aunque actualmente estoy muy centrado en una dieta

Una de mis fieles bicicletas que compré a un chico en Kansas por alrededor de 300\$,

Atkins, me gusta cocinar mucho, ya que mi pareja no es el más idónea para cocinar risotto. Aún me gusta bastante la fotografía pero sólo como hobby. Y por supuesto, me paso mucho tiempo analizando material de diseño para las publicaciones.

¿Qué novedades te gustaría ver en futuros ODROIDs?

Pues bien, un controlador SATA y Ethernet son obvios. Conozco a los chicos de Hardkernel y realmente no es una cuestión de si pueden o no. Están haciendo un gran producto que realmente revienta la competencia. Hay diferentes formas de ver lo que se puede esperar de la próxima generación de ODROIDs:

Cómo servidor de archivos, incruir SATA y 2 ethernets gigabit

 Como máquina de juegos, una mejor GPU y un puerto estándar DVI/VGA

 Para virtualización, incruir 2 procesadores físicos, más RAM, y Linux ejecutando PAE

Qué ODROID es tu favorito?

El X2 es mi servidor domestico, ejecuta un servidor de archivos SMB y un servidor multimedia para Playstation. Maneja mis torrents y hace casi todo lo que necesito. Pero una vez que tuve el U3 enganchado en un Atrix Motorola ejecutando OpenMSX, me hizo pensar si el U3 podría convertirse en mi nuevo favorito, pero sólo fue un truco para llamar mi atención.

Yie Ar Kung-Fu y todos los juegos de MSX se ejecutan en uno de mis U3, Finalmente adapte un U3 sobre una caja horbit!

