

ODROID

Año Dos
Mum. #19
Jul 2015

Popcorn **Time**

Magazine

Disfruta de películas y programas de TV al instante con tu ODROID



Diet Pi

Liviana distribución para tu ODROID

- Atril Electrónico ODROID-C1
- Control Pines GPIO C1

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



HARDKERNEL



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

Dirección: Max-Pollin-Straße 1
85104 Pförring Alemania

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

Nuestros productos ODROID se pueden encontrar en: <http://bit.ly/1tXPxwe>





Uno de los usos más comunes de **ODROID** es emplearlo como centro multimedia, y **Popcorn Time** es un software todo-en-uno que puede reproducir casi cualquier tipo de película o programa de televisión. Funciona muy bien en el **U3**, convirtiéndolo en un dispositivo muy útil y barato que puedes colocar sobre el televisor.

Como siempre, contamos con divertidos artículos sobre juegos para la plataforma **ODROID**, cómo crear tus propios videojuegos para un clásico ordenador Amstrad, jugar al **Millenia: Altered Destinies**, y disfrutar de **Nubs' Adventure** y **Kung Fury** para Android. Para los apasionados de Android, **Nanik** continúa su serie de **Desarrollo Android** con una guía para compilar **Android Studio**, un entorno de desarrollo interactivo de Java.

Para los amantes al bricolaje, **Iván** presenta su innovador **Atril electrónico**, que utiliza como músico profesional para acceder a sus partituras de música y tomar notas con una moderna interfaz con pantalla táctil. También detallamos cómo acceder a los pines **GPIO** de un **ODROID-C1** usando una librería Java llamada **jOdro**, analizamos una liviana distribución llamada **DietPi**, y aprenderemos cómo compilar un kernel Linux para **ODROID** usando scripts automatizados.

ODROID Magazine, que se publica mensualmente en <http://magazine.odroid.com/>, es la fuente de todas las cosas ODROIDianas. • Hard Kernel, Ltd. • 704 Anyang K-Center, Gwanyang, Dongan, Anyang, Gyeonggi, South Korea, 431-815 • fabricantes de la familia ODROID de placas de desarrollo quad-core y la primera arquitectura ARM "big.LITTLE" del mundo basada en una única placa.

Únete a la comunidad ODROID con miembros en más de 135 países en <http://forum.odroid.com/> y explora las nuevas tecnologías que te ofrece Hardkernel en <http://www.hardkernel.com/>.



HARDKERNEL

HARDKERNEL'S EXCLUSIVE NORTH AMERICAN DISTRIBUTOR



SHOP NOW

All Hardkernel products in stock at AmeriDroid.com



USB GPS MODULE
\$26.95



ODROID-C1
\$36.95



ODROID-VU
\$119.95



C1 3.2 INCH TOUCHSCREEN DISPLAY SHIELD
\$26.95

ODROID

Magazine



**Rob Roy,
Editor Jefe**

Soy un programador informático que vive y trabaja en San Francisco, CA, en el diseño y desarrollo de aplicaciones web para clients locales sobre mi cluster ODROID. Mis principales lenguajes son jQuery, angular JS y HTML5/CSS3. También desarrollo SO precompilados, Kernels personalizados y aplicaciones optimizadas para 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 ODROID, kernel precompilados e imágenes en <http://bit.ly/1fsaxQs>.



**Robert Cleere,
Editor**

Soy un diseñador de hardware y software que actualmente vive en Huntsville, Alabama. Aunque semi-retirado del diseño de los sistemas integrados, incluyendo más de una década trabajando en el programa del transbordador espacial, continúo diseñando productos de software y hardware, y me interesa la producción de audio/video y las obras de arte. Mis lenguajes de programación son Java, C y C ++, y tengo experiencia con bastantes sistemas operativos integrados. Actualmente, mis proyectos principales son los sistemas navales de seguimiento y control, monitoreo ambiental y la energía solar. Actualmente estoy trabajando con varios procesadores ARM Cortex, pero mi ODROID-C1 es en gran medida el más poderoso de todos.



**Bruno Doiche,
Editor Artístico Senior**

Date prisa Bruno, tenemos que empaquetar la revista para enviárselas a nuestros lectores. Piensa en algo divertido podamos incluir aquí, ¡rápido!

....
....
....
....

¡Lo tengo!

“No me considero una persona fea, sino más bien un mono muy guapo!”

Además:

“La gente dice que el dinero no es la clave de la felicidad, pero siempre pensé si tienes suficiente dinero, puede tener una marca clave.”



**Nicole Scott,
Editor Artístico**

Soy una experta en Producción Transmedia y Estrategia Digital especializada en la optimización online y estrategias de marketing, administración de medios sociales y producción multimedia impresa, web, vídeo y cine. Gestiono múltiples cuentas con agencias y productores de cine, desde Analytics y Adwords a la edición de vídeo y maquetación DVD. Tengo un ODROID-U3 que utilizo para ejecutar un servidor web sandbox. Vivo en el área de la Bahía de California, y disfruta haciendo senderismo, acampada y tocando música. Visita mi web en <http://www.nicolecscott.com>.



**James LeFevour,
Editor Artístico**

Soy un especialista en medios digitales que disfruta trabajando como freelance en marketing de redes sociales y administración de sitios web. Cuanto más aprendo sobre las posibilidades de ODROID más me ilusiona probar cosas nuevas con él. Me traslade a San Diego desde el Medio Oeste de los EE.UU. Todavía estoy bastante enamorado de muchos aspectos que la mayoría de la gente de la Costa Oeste ya da por sentado. Vivo con mi encantadora esposa y nuestro adorable conejo mascota; el cual mantiene mis libros y material informático en constante peligro.



**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 las tecnologías de la información: ISO 9001, ISO 27001, ISO 20000 Soy un apasionado de la informática, especialmente de los microordenadores como el ODROID, Raspberry Pi, etc. Me encanta experimentar con estos equipos y traducir ODROID Magazine. Mi esposa dice que estoy loco porque sólo pienso en ODROID. Mi otra afición es la bicicleta de montaña, a veces participo en competiciones semiprofesionales.



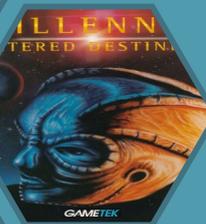
AMSTRAD - 6



KERNEL LINUX - 8



JUEGOS ANDROID: NUBS' ADVENTURE- 9



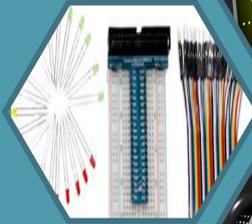
JUEGOS LINUX: MILLENIUM - 10



JUEGOS ANDROID: KUNG FURY - 15



DESARROLLO ANDROID - 16



JAVA GPIO - 19



GENERADOR DE RUIDO BLANCO - 20



MUSICA ODROID - 21



DIET PI - 25



POPCORN TIME - 30



CONOCIENDO A UN ODROIDIAN - 32

DESARROLLAR VIDEOJUEGOS PARA AMSTRAD CPC

DISFRUTA DE ESTE PEQUEÑO VIAJE AL PASADO

por Jose Cerrejon



AMSTRAD CPC es una de esas reliquias de 8 bits que siempre ocupará un sitio en nuestros corazones

Para los que no lo sepan, el Amstrad CPC era un ordenador de 8 bits que fue muy popular entre 1984 y 1990. CPCtelera es un motor que ha sido publicado recientemente en su primera versión estable, la cual facilita la creación de juegos para Amstrad usando código ensamblador o C.

Presentando CPCtelera

CPCtelera es un entorno de desarrollo integrado que permite crear contenido y juegos para Amstrad CPC y que incluye:

- Una librería de bajo nivel con soporte para: gráficos, audio, teclado, firmware, cadenas de caracteres, tratamiento de hardware de vídeo y gestión de memoria.
- Una API para desarrollar juegos y software en C y ensamblador.
- Herramientas para la creación de contenido (edición de nivel, gráficos y audio).
- Multiplataforma: Funciona con los Sistemas Operativos Windows, Ubuntu, Debian, Arch y Manjaro.

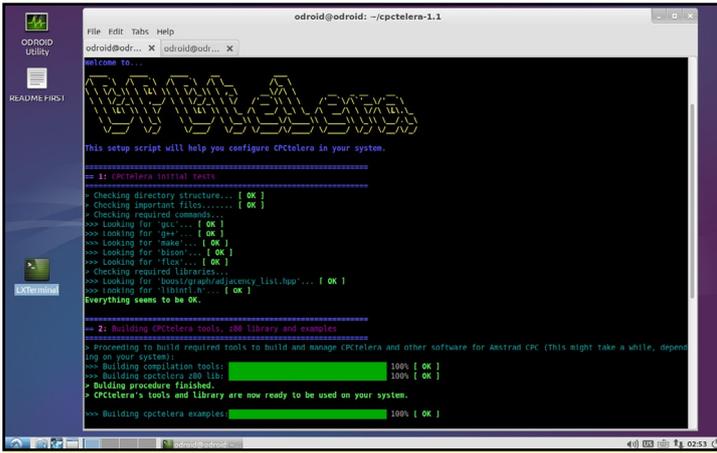
Su desarrollador principal es Francisco Gallego (@frangallegobr), un ingeniero informático, desarrollador de videojuegos y profesor de la Universidad de Alicante en España. Si quieres profundizar en este entorno de trabajo, puedes consultar los enlaces que aparecen al final de este artículo.

Instalando CPCtelera

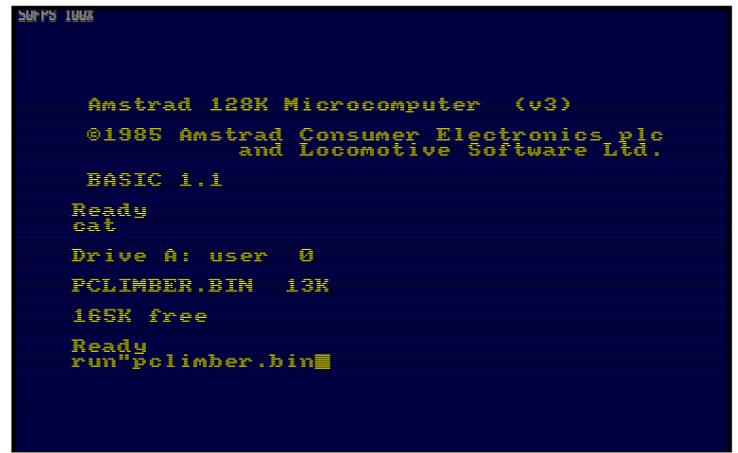
En primer lugar, necesitas descargar el código fuente para compilarlo en tu placa. Para ello, descarga la versión estable como lo hice yo, o si eres atrevido, puedes utilizar la última versión de GitHub:

```

$ wget http://bit.ly/1MMdUMA && \
  unzip -nq $(basename $_) && \
  rm $(basename $_) && \
  cd cpctelera-1.1/
# or use the last commit:
$ git clone http://bit.ly/1IPxMOF && \
  cd $(basename $_)
  
```



Ejecutando ./setup.sh tras resolver las dependencias



¡Amstrad BASIC, es hora de desempolvar unos cuantos manuales!

A continuación, tenemos que instalar las dependencias que faltan, algunas de las cuales ya pueden estar instaladas:

```
$ sudo apt-get install -y build-essential libboost-dev flex bison
```

Luego, activa el script de instalación ejecutando el archivo setup:

```
$ sudo ./setup.sh.
```

Se te advertirá de los paquetes necesarios y preparará tu sistema para ejecutar el motor. En un ODRROID-C1, la compilación suele tardar unos 20 minutos.

Iniciando el Motor

Si navegamos por los directorios, podemos ver algunas carpetas interesantes como docs/, que contiene el manual de referencia o tools/, que se utiliza para hacer sprites, componer bandas sonoras, convertidores de formatos, y mucho más. Algunas de estas herramientas solo están disponibles para Windows.

Para crear un nuevo proyecto Bash, escribe lo siguiente:

```
$ cpct_mkproject [folder_project]
```

Navega por el directorio y verá dos subdirectorios:

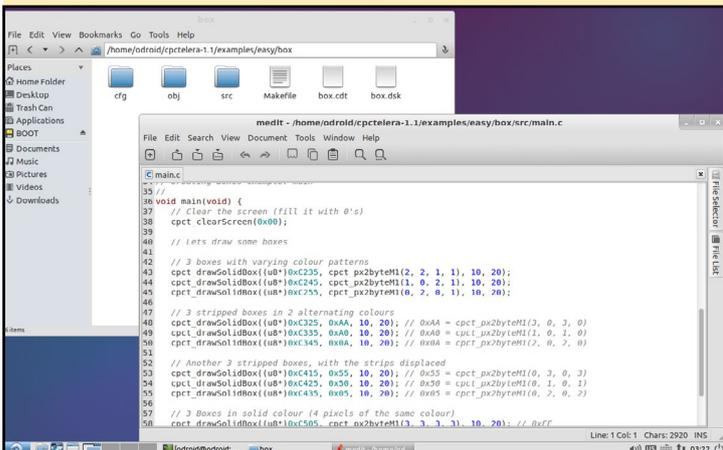
- **src/** con el código fuente (la primera vez sólo tenemos un main.c). Puede crear archivos y directorios de tu juego aquí.
- **cfg/** contiene los parámetros necesarios para compilar el juego a través del archivo build_config.mk.

También hay ejemplos que puedes analizar. Simplemente ve al directorio, ejecuta make y se crearán automáticamente los archivos .CDT y .DSK. ¡Increíble, ¿no?!

Platform Climber

CPCtelera incluye un completo juego de ejemplo llamado Platform Climber. En primer lugar, necesitas un emulador de Amstrad CPC. No he encontrado ninguno que estuviese pre-compilado para ODRROID, así que tuve que descargar y compilar uno por mí mismo. No te preocupes, es fácil:

Dispones de muchos ejemplos para aprender



Scanlines junto a la mejor nostalgia, ¡Prepárate para trepar!



```
$ wget http://bit.ly/1U2RrjZ
$ unzip caprice*
$ make -f makefile.unix
RELEASE=TRUE
```

En un minuto, conseguirás un binario llamado cap32. El uso es muy sencillo utilizando cualquier archivo .BAS o .DSK. Para cargar el juego Platform Climber, ejecuta el emulador seguido de la ruta del archivo .DSK:

```
$ ./cap32 ../cpctelera-1.1/
examples/games/platformClimber/
pclimber.dsk
```

¡Ahora ya está dentro de AMSTRAD! ¿Sientes la magia? Escribe cat para ver los archivos del disco montado previamente. Luego, ejecuta el juego:

```
cat
run"game_name.extension
```

Observa que las comillas no se cierran. Por ejemplo, para ejecutar Platform Climber, escribe lo siguiente:

```
run"pclimber.bin
```

Conclusión

Ahora es tu turno. Te recomiendo que empieces a estudiar los ejemplos de examples/easy/src, luego modifícalos y compílalos. Debajo tienes un enlace al manual de referencia con toda la información que necesitas sobre funciones y métodos disponibles. Si sabes algo de SDL, todo te será un poco más fácil.

¡Feliz codificación!

Más Referencias:

<http://bit.ly/1IPxMOF>
<http://bit.ly/1IpjQvV>
<http://bit.ly/1FMAkrQ>
<http://bit.ly/1IILeAT>

SCRIPTS PARA EL KERNEL DE LINUX

RETOCA TU SISTEMA

por Rob Roy

Para hacer que la compilación del kernel Linux sea más sencilla, he escrito un conjunto de scripts BASH que permiten descargar la última versión de cualquier kernel almacenado en la sección GitHub de Hardkernel, compilar el código fuente y empaquetar el kernel terminado en un paquete que pueda distribuirse. Los scripts guardan el kernel original en la máquina de compilación, de forma que el kernel para un dispositivo ODROID se puede compilar en un ODROID distinto.

El paquete resultante incluye un script de instalación de un solo clic, puede ser compartido con otros, instalarlo en cualquier ODROID compatible o añadirlo a un repositorio para su distribución vía apt-get. Los scripts dan indicaciones a lo largo del proceso, lo cual es ideal para usuarios que no tienen experiencia en la compilación de kernels.

Información general

Los scripts se encuentran en <http://bit.ly/1U6kQcU>, y se pueden descargar usando wget desde la línea de comandos o desde cualquier navegador web. Descomprime el paquete tras navegar a la carpeta de descargas:

```
$ wget \
http://bit.ly/1U6kQcU
$ tar -xvzf build.tgz
```

La carpeta "build" resultante contiene los siguientes scripts:

download.sh es usado para descargar una sección concreta del repositorio de Hardkernel

build.sh se utiliza para poner en marcha el proceso de compilación del kernel.

install.sh se copia en el paquete de instalación del kernel resultante para poder instalar el kernel en cualquier dispositivo ODROID.

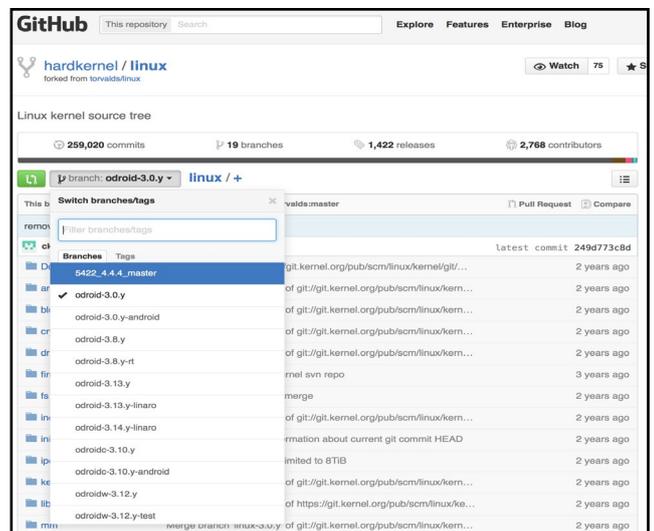
Script Download

El script "download.sh" se puede utilizar para descargar cualquier módulo disponible del repositorio GitHub de Hardkernel especificándolo como primer argumento. Por ejemplo, para descargar la versión más reciente de la sección ODROID-3.10.y, navega a la carpeta "build" y escribe lo siguiente:

```
$ sh ./download.sh odroidc-3.10.y
```

Para obtener una lista completa de todas las secciones visita <http://bit.ly/1NvVQa1> y abre el menú desplegable "branch", como muestra la imagen.

Tras completarse el script download, los archivos "download.sh" y "install.sh" se copiarán automáticamente al directo-



El lugar donde obtener el kernel más actualizado es el menú desplegable de la sección correspondiente dentro del repositorio GitHub de Hardkernel

odroidq2_android_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidq2_android_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidq2_ubuntu_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx2_android_emmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx2_android_sdmmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx2_ubuntu_defconfig	defconfigs: x, x2, u2 : REISERFS,F2FS,AFS,UHID,Joystic Stuff,Crypto S...	2 years ago
odroidx2_ubuntu_mail_defconfig	ODROID-U2: Added a defconfig to Ubuntu with Mail	2 years ago
odroidx_android_emmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx_android_sdmmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx_ubuntu_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx2_android_emmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx2_android_sdmmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx2_ubuntu_defconfig	defconfigs: x, x2, u2 : REISERFS,F2FS,AFS,UHID,Joystic Stuff,Crypto S...	2 years ago
odroidx2_ubuntu_mail_defconfig	ODROID-X2: Added Ubuntu Mail defconfig	2 years ago
odroidx_android_42inch_demo_def...	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago
odroidx_android_emmc_defconfig	Revert "odroid: Disable the exynos-mem block device on all ubuntu def...	3 years ago

Selecciona tus archivos de configuración para la mejor compilación

rio de la sección descargada. Ahora pasamos al siguiente paso.

Script Build

Una vez descargado la sección, ve al directorio en cuestión y busca el archivo en el directorio `arch/arm/configs/` correspondiente a la plataforma ODROID que utilizará el kernel:

```
$ cd odroidc-3.10.y
$ ls arch/arm/configs/odroid*
```

Por ejemplo, si seleccionas la sección `odroid-3.10.y`, los archivos de configuración están disponibles para usarse con ODROID-Q, ODROID-Q2, ODROID-X, ODROID-X2 y ODROID-U2 como muestra la imagen anterior. Otras secciones pueden incluir archivos de configuración para otros dispositivos ODROID, tales como el XU y XU3. Si compilas un kernel para el ODROID-U2/U3 usando `odroid-3.10.y`, el archivo de configuración de destino debería ser `odroidu2_ubuntu_mali_defconfig`, que es incluido como argumento principal en el script "build.sh". Cualquier cambio en el archivo de configuración se debe hacer antes de lanzar el script build. Yo prefiero editar el archivo con un editor de texto, pero también se puede usar "make menuconfig".

El siguiente comando inicia el proceso de compilación usando el archivo de configuración seleccionado. Requiere privilegios de superusuario, ya que necesita tener acceso a ciertos directorios del sistema para poder empaquetar los archivos del kernel. Asegúrate de reemplazar el nombre del archivo de configuración de destino por el argumento:

```
$ sudo sh ./
build.sh \
odroidc_ubun-
tu_mali_def-
config
```

La compilación puede tardar de 10 a 30 minutos, el resultado es una estructura de carpeta con los recursos de instalación del kernel. Esta estructura

está incluida en una subcarpeta bajo la nueva "versión" usando el nombre del archivo de configuración. El proceso de compilación no instala el kernel en local, sino que crea un paquete portátil que después se puede ejecutar en el dispositivo ODROID que elijas.

Para instalar el kernel en un equipo, copia la nueva estructura de directorios al ODROID. Luego, ve al subdirectorio que lleva el nombre del archivo de configuración seleccionado. En este ejemplo, el script de instalación se puede ejecutar escribiendo los siguientes comandos:

```
$ cd release/\
odroidc_ubuntu_mali_defconfig
$ sudo sh ./install.sh
```

Este proceso puede tardar de 3 a 10 minutos, tras lo cual el nuevo kernel estará listo. Reinicia el equipo para usar el kernel actualizado. Una backup del kernel original se guarda tal y como se indica en el resultado del script de instalación, de modo que la actualización del kernel se puede invertir si fuera necesario.

Si tiene preguntas, comentarios o sugerencias sobre los scripts, por favor crea un nuevo hilo en los foros ODROID en <http://forum.odroid.com>.

Divisiones

- odroid-3.0.y: Q, U2, U3, X, X2
- odroid-3.8.y: U2, U3, X, X2
- odroid-3.13.y: XU
- odroidx-3.10.y: C1, C1+
- odroidw-3.12.y: W
- odroidxu3-3.10.y: XU3
- odroidxu-3.4.y: XU
- odroidxu4-v4.2-rc1: XU3, XU4

NUBS' ADVENTURE DESAFIANTE JUEGO DE PLATAFORMAS 2D

por Bruno Doiche

Lo que no echamos en falta en ODROID Magazine son los juegos de plataformas en 2D con los que disfrutamos bastante.



Después de todo, emulamos todo tipo de consolas de 8 y 16 bits. Pero a menudo me pongo con una distro Android en mi U3 conectado a mi lapdock en busca de una nueva experiencia. Y para mi agrado me las ingenié para dar con Nubs' Adventure, un juego de plataformas muy ameno que me trasladó a increíbles mundos de la mejor versión de 'Metroidvania'. Gran diseño de niveles, jefes desafiantes y excelentes puzzles para mantenerte con ganas de jugar más de lo que esperas inicialmente. ¡Disfruta!

<https://play.google.com/store/apps/details?id=nubs.adventure>



JUEGOS LINUX

LAS JOYAS DE LOS JUEGOS RAROS

PARTE I

por Tobias Schaaf

Quisiera presentar algunos de mis juegos favoritos que probablemente no todo el mundo conozca, pero que son muy interesantes y divertidos. Espero que disfrutéis con los que elegidos, y os animo a que probéis y juguéis a algunos de los juegos más raros e inusuales de la historia de los juegos.

Millennia – Altered Destinies

Este juego es muy singular. Eres el piloto de un carguero en su viaje de 6 meses a la Tierra desde Júpiter. Al tiempo que te das cuenta de que podría ser una mala idea hablar en voz alta en tu primer día de viaje, eres de repente “secuestrado” por una especie alienígena. Te explican que estás en una especie de estasis temporal, y que son una especie llamada Hood. Son guardianes del tiempo que necesitan tu ayuda en su época y galaxia.

Una galaxia llamada Echelon que está actualmente invadida por una especie llamada Microids. Esta especie es muy agresiva, y la galaxia de los guardianes de tiempo está a punto de ser invadida y la próxima será “nuestra” galaxia, es por eso que se supone que debemos solucionar este conflicto. Te ofrecen una nueva nave espacial que no sólo puede viajar en el espacio, sino también en el tiempo, y eres enviado de vuelta a 10.000 años atrás, a la galaxia Echelon donde el sistema estelar sólo está ocupado por Microids.

El objetivo es establecer las cuatro especies originales que vivieron en la ga-



Esta es la versión original en CD-ROM de Millennia - Altered Destinies

laxia Echelon, así como ayudarles a prosperar hasta que sean lo suficientemente fuertes como para defenderse de los Microids, y por lo tanto salvar la galaxia. Esto además te interesa, no sólo porque no quieres que ataquen nuestra galaxia, sino porque en el momento en que empiezas en la galaxia Echelon 10.000 años atrás en el pasado, la tecnología necesaria para traerte de vuelta ya no existe, ya que fue creada por las cuatro especies que se supone que vas a ayudar a desarrollar. Así que lo que tienes que hacer es volver a tu propia época y galaxia para ayudar a desarrollar las cuatro especies hasta que sean capaces de construir las partes que faltan de tu nave, y ayudarles a defenderse de los Microids.

El juego tiene aspectos muy buenos, como el uso de voz a través de tu ordenador y AI Agnis, diferentes escenas de película, acción de combate espacial, y mucho más. Por lo que yo sé, este juego no es muy conocido, realmente es muy inusual por su estilo único y jugabilidad. Definitivamente vale la pena probarlo y sin duda deberías darle una oportunidad.

Instalación y puesta en marcha

Dado que éste es un juego para DOS y no está hecho para Linux, no puedes simplemente descargarlo y ejecutarlo. Usé DOSBox para iniciar el juego y configuré algunas opciones para que fun-

cionase correctamente. En primer lugar, necesitas el CD o un rip del mismo, aunque te aconsejo utilizar la versión del CD para una mejor experiencia.

Aquí tienes una pequeña guía de cómo instalar e iniciar el juego basada en mi imagen ODROID GameStation Turbo, aunque debería funcionar con la imagen de Ubuntu de HardKernel si tienes activadas mis listas de paquetes `all/testing` y `all/main`. Escribe lo siguiente para instalar los requisitos para DOSBox:

```
$ sudo apt-get install dosbox-odroid libgl-odroid
```

Configurar DOSBox

Inicia DOSBox para crear el archivo de configuración por defecto, luego, salte de inmediato. Abre `/home/odroid/.dosbox/dosbox-SVN.conf` con un editor de texto y cambia las siguientes líneas:

```
[sdl]
fullscreen=true
fullresolution=1920x1080
output=opengl

[render]
frameskip=3

[cpu]
core=dynamic
cputype=pentium_slow
cycles=5000
```

```
cycleup=200
cycledown=200
```

En primer lugar, he creado una carpeta donde quiero colocar mis juegos:

```
$ mkdir DOS
```

También he copiado la ISO que he creado desde mi juego Millennia – Altered Destinies, y la coloqué en una carpeta llamada CDs en mi ODROID. Para facilitar las cosas, he añadido las siguientes líneas al final del archivo de configuración de DOSBox, así no necesito escribirlas cada vez que quiera ejecutar el juego:

```
[autoexec]
mount c: /home/odroid/DOS
c:
imgmount d: /home/odroid/CDs/
Mil.iso -t iso
```

Ahora el sistema está preparado para lanzar el emulador. La carpeta DOS se monta automáticamente como mi unidad C:, y el CD será montado como D: como una unidad de CD-ROM. Instala el juego como siempre bajo DOS e inícialo. La introducción puede ser un poco lenta, pero en general con los ajustes mencionados deberías ser capaz de ejecutar el juego a toda velocidad.

Para ejecutar el juego, lanza DOS-Box de nuevo con el siguiente comando, para que DOSbox utilice glshim (libgl-droid) y así poder activar la aceleración OpenGL.

```
$ LD_LIBRARY_PATH=/usr/local/lib
dosbox
```

Los inicios del juego

Tras la introducción, te encontrarás en una galaxia muy muy lejana (te suena familiar, ¿no?). El juego no viene con un tutorial, lo que significa que tendrás mucha acción sin saber lo que debes o puedes hacer. Por eso, voy a intentar explicarte las cuestiones básicas del juego,

lo cual te ayudará a entenderlo mejor.

Tienes que utilizar las capacidades de tu nave para viajar en el tiempo y en el espacio y así poder solucionar los problemas de las cuatro especies que debes proteger, y ayudarles a evolucionar hacia una civilización avanzada que pueda ayudarte a salir de la galaxia, así como defenderse de los Microids. Para ello, tienes que entender y controlar tu nave para completar todas las tareas que tienes por delante.



Vista principal de tu nave espacial desde donde comienzas tu trabajo

La nave

Dentro de tu nave puedes ver Agnis, el ordenador de la nave y IA, que te ayudara en tu tarea. Actúa de traductor cuando hablas con las diferentes especies. Puede darte consejos y hacer comentarios sobre tus acciones. De hecho, su “yo estoy a tu servicio, humano” fue el sonido de inicio de mi máquina Windows durante mucho tiempo.

La mayoría de las pantallas de la nave se asemejan. Tienes un panel de navegación en el lado izquierdo, que se puede controlar pulsando y manteniendo el botón derecho del ratón y moviéndote sobre los diferentes botones. Un clic con el botón izquierdo selecciona el sistema que quieres visitar, una especie de ascensor que llega a tu cabina te lleva a la estación seleccionada. A la derecha, tienes los botones de acción de su estación actual. Simplemente mueve el ratón hacia el botón deseado y pulsa el botón izquierdo del ratón para activar la acción.

En la vista principal, puedes ver información sobre el planeta actual que está visitando. El nombre del planeta y el

año se muestra sobre la cruz del escáner. En el lado derecho, puedes encontrar información sobre el planeta y sus habitantes. Aquí podemos ver la especie (Raptoids), su calificación IQ actual (80), y un símbolo del evento actual (war). En la parte superior de la pantalla, vemos de nuevo un pequeño panel de información con el nombre del planeta, la especie y un reloj. El año 1600 es el siglo actual, y el símbolo 117.13 es en realidad un reloj del tiempo de juego, que tiene un formato muy extraño. Cuenta en segundos, pero llega a 100 antes de que el número anterior a los decimales cambie.

La vista principal sólo permite iniciar un viaje al siguiente siglo, o a un lugar diferente en la galaxia. Sólo puedes iniciar los motores desde esta pantalla. Otra cosa que sólo puedes hacer desde la pantalla principal es el combate espacial. De vez en cuando, tienes que defenderte y defender a los habitantes del planeta frente a los enemigos invasores. Dado que este juego gira en torno al tiempo, esto en algún momento puede llegar a ser algo confuso, ya que en realidad podrías terminar luchando contra tu otro yo en una línea temporal diferente, pero también podrías luchar contra Microids que tratan de invadir el espacio donde se asienta tu raza. Los Hoods a menudo tratan de atacar la raza que creaste desde una línea de tiempo diferente, y una de las otras razas incluso podría tratar de invadir el planeta mientras trata de expandir su territorio. Por eso, tu nave está equipada con algunas armas e incluso se puede modificar con algunos de los inventos que obtienes a partir de las razas que has creado.

Figura 2 - Luchando contra tu otro yo en Milenios



Para luchar contra los enemigos, mantén pulsado el botón derecho del ratón y muévelo en la dirección que desees. Un pequeño punto rojo indica la ubicación del enemigo más cercano. Pulsa el botón izquierdo para disparar a tu enemigo.

La lucha contra tu otro yo es bastante fácil, y normalmente escapas tras un par de golpes. El combate contra los otros enemigos es algo más difícil, ya que a menudo son más numerosos y mientras luchas contra uno, otro puede venir por detrás y atacarte. El panel de información de color rojo en la parte superior se convierte en un “espejo retrovisor”, que muestra los enemigos que te atacan por detrás. Además, tu nave puede dañarse y si la avería es demasiado grande, huirás automáticamente de la escena y te dirigirás al centro del universo, al único planeta donde no se puede crear ninguna especie. Así que en realidad nunca puedes morir.

A la vista principal se puede llegar a través del botón que apunta hacia el norte en la parte superior izquierda de la barra de navegación. La navegación es probablemente el sistema más importante que tienes. Si hace clic en el botón



Mapa de la galaxia del sistema de navegación, no sólo para moverte en el espacio, sino también para viajar en el tiempo.

que mira hacia el este, en el panel de navegación superior, se puedes ir al mapa de la galaxia y al sistema de navegación.

En el sistema de navegación, recopilas toda la información que necesitas para planificar tu próximo paso. El mapa de galaxia te muestra lo que se ve en la galaxia y cómo se distribuyen las diferentes especies. Puesto que empiezas 10.000

años atrás en el pasado, sólo existe un planeta habitado con Microids, ningún otro con otra especie. Tienes que elegir un planeta en el que quieres desarrollar las cuatro razas diferentes. Cada una requiere de un hábitat distinto.

Desde aquí puedes fijar una ruta a través del espacio y el tiempo con el fin de ayudar a las especies en su desarrollo. Sugerencia: Viajar en el tiempo y el espacio consume combustible, y sólo puedes recargar combustible en los gi-



El histograma que muestra todos los eventos importantes del desarrollo de las razas es una herramienta muy importante

gantes gaseosos. Por lo que es buena idea que crees tu raza en un planeta que se encuentre cerca de un gigante gaseoso. Hay un botón que activa o desactiva los planetas que no tienen gigantes gaseosos.

En la esquina inferior derecha de esta pantalla, puedes ver el botón del histograma, tu mejor amigo. En el histograma ves todos los eventos que tienen lugar y cada paso representa 100 años de desarrollo. Los nuevos inventos están marcados con un borde verde, los bordes rojos indica crisis.

Aquí es donde el juego se complica. Tienes que ayudar a la raza a pasar las crisis, o incluso prevenir que las crisis lleguen a producirse. A menudo hay diferentes soluciones a un problema con diferentes resultados, y aquí es donde el juego también se vuelve interesante.

Tienes diferentes formas de interacción para desarrollar una raza, y la forma más común es la de comunicarte con un representante de la especie. Este representante es conocido como Agent, que es un ser modificado de las especies que ayudas. Sólo él sabe de tu existencia y del



Dos de las diferentes especies con las que tienes que lidiar en el juego: los insectos Entomon y los reptiles Reptoides

gran plan. Sabe que puedes alterar los acontecimientos en el tiempo, y es por ello que puede ver lo que sucederá en el futuro. Aún así, el Agent es parte de la raza que estás desarrollando y tendrá creencias y necesidades similares, lo que los hace a veces difícil de tratar y cada uno requiere de una forma de persuasión distinta.

Si haces clic en el botón que mira hacia el oeste en la barra de navegación superior te llevará al centro de comunicaciones. Aquí, puedes ponerte en contacto con tus agentes e intentar solventar sus problemas. Agnis te ayudará a traducir lo que tus agentes te estén diciendo. Puedes ver las pequeñas películas de vídeo de los agentes cuando hablan contigo, haciendo que parezca una transmisión de vídeo. Con la ayuda de Agnis, intentas resolver sus problemas. Esto incluye temas como la política, donde tienes que elegir quién debe ser el líder de una raza, planificar asesinatos de líderes rebeldes con el fin de evitar levantamientos o decidir si ayudas a los rebeldes con el fin de matar a un emperador tirano.

Por otro lado, también tienes que dar consejos sobre cómo protegerse ante un duro invierno, cómo evitar que tomen decisiones equivocadas que luego puedan conducirles a catástrofes, o

ayudarles a desarrollar nuevos inventos que les permitan defenderse de los enemigos. Hay muchos temas diferentes a los que necesitas hacer frente, y esto es lo que realmente hace interesante al juego. Las decisiones que tomes pueden tener pequeñas repercusiones al principio, pero también pueden suponer un gran impacto mil años después en el futuro.

¿Debería ser radicada una mutación antes de que contamine una sociedad “pura”, debería ignorarse y posteriormente ser curada, o esta mutación menor podría ser vital en el futuro? ¿Deberías elegir un líder fuerte, un dictador que lleve a las personas a la guerra y a la dominación, o deberías elegir un pacifista que desee la paz y la cooperación con el resto? ¿Frenar al dictador el crecimiento una vez que esté satisfecho y no verá lugar para más cambios, o la paz de mil años con un pacifista dará como resultado una raza débil que simplemente no quiera aventurarse en el espacio para conquistar nuevos mundos? ¿Cómo puedes prevenir una guerra entre dos razas? ¿O deberías más bien acelerar el desarrollo, aunque sea haciendo uso de armamento militar? ¿Es mejor combatir el cambio climático con grandes fuegos en todo el planeta que mantengan un cierto nivel de calor, o esto contaminará constantemente el medio ambiente? ¿O deberías hacer pequeños incendios y reunir a muchas gente en un mismo espacio, a pesar de que esto podría conducir a la escasez de alimentos, el canibalismo, o simplemente a las falsas creencias? A veces incluso hay que resolver ciertos asuntos religiosos para que las especies vayan por la dirección correcta.

Hay muchas más cosas que tienes que decidir, pero también tienes que hacer frente a las diferentes creencias y personalidades de tus agentes. Algunos pueden ser de gran ayuda e intentan hacer lo que dices y otros necesitan que sean más convincente, con explicaciones lógicas, realizando debates o usando sus propias creencias contra ellos. A algunos incluso podrías tratar de engañar.

Si tienes éxito, una tormenta temporal sacudirá tu nave y la historia de la especie cambiará. Luego, puedes volver al sistema de navegación y comprobar el histograma para ver cómo ha cambiado el desarrollo de la especie. Sugerencia: Dado que algunos cambios pueden tener consecuencias mil años más tarde, lo mejor es guardar la partida antes de cada interacción con una especie, y así poder deshacer lo que has hecho.

Otra forma de interactuar con la especie es utilizar el transportador de tu nave. Al igual que en Star Trek, puedes utilizar el transportador para llevar objetos a tu nave y devolverlos al planeta. Se puede acceder al transportador con botón que mira al sur en la barra de navegación superior izquierda.

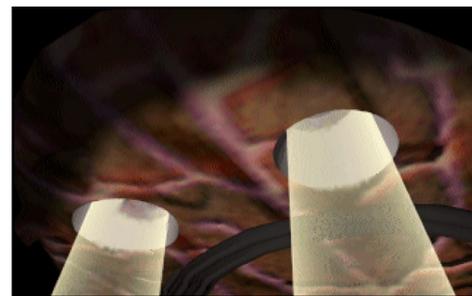
En primer lugar, haz clic en el botón



Primero entra en órbita, luego busca el Templo

scan y verás una animación de cómo el ordenador busca un templo. Este templo interactúa contigo. Los nuevos inventos son colocados allí para que puedas cogerlos, o si está vacío, puedes colocar uno.

El transporte de inventos puede ser útil de diferentes formas. A veces, un invento que parecía ser una buena idea, como son los perforadores sónicos para recursos mineros, puede causar catástrofes más adelante como terremotos. Pero, también puede acelerar el descubrimien-



Así es como transportar un invento a tu nave



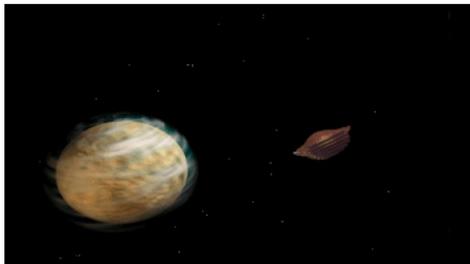
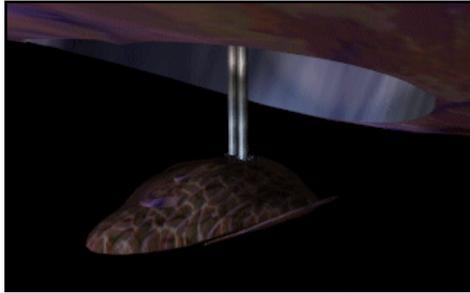
Mira esto, ahora tenemos arco y flechas

to de nuevos inventos que se remontan a cientos de años, por lo que se necesita menos tiempo para desarrollar la especie. Esto puede ser crucial, ya que cuando llegan a un IQ de 300, puedes darles los planos de las partes que le faltan a tu nave que tardarían miles de años en construir. Y tienen que hacerlo antes de que sean atacados por los Microids. Por lo tanto, acelerar el desarrollo es a menudo crucial para alcanzar tus objetivos.

Sugerencia: Si un invento da a tu gente un IQ de 100, puede extraer el invento desde su línea temporal, y puedes volver a donde tenían un IQ de 80 y darles el invento unos cientos de años antes para acelerar su desarrollo. En ocasiones, hablar no es suficiente para resolver un problema en un planeta, y no puedes detener una disputa entre los diferentes líderes utilizando un simple invento. Así que de vez en cuando tienes que tomar medidas más drásticas para que dejen de matarse entre ellos, o simplemente para ver cómo está la situación, incluso si esto significa que tienes que bajar al planeta y volar un edificio con el líder de un bando enemigo dentro. Por ello, cuentas con una pequeña nave que es capaz de bajar a la superficie del planeta donde se puede atacar un edificio y hacerlo estallar con

las armas que incorpora.

Esto puede ser necesario para matar a un líder enemigo que es inalcanzable para tu agente, o matar al agresor de una especie diferente, que vive en el planeta y amenaza con matar a tu



El descenso de la nave a la superficie del planeta es uno de los videos del juego, y va cambiando dependiendo del planeta que visitas.

gente. A veces incluso es necesario tantear el terreno para que tu agente realmente haga lo que le pides.

Sugerencia: Atacar el planeta con la nave es poco común. Si atacas cualquier edificio sin “ser necesario” corres el riesgo de matar a toda la especie, no importa lo insignificante que sea el edificio. Al instante sabrás si atacar el planeta es lo que hay que hacer, o si tienes que buscar una solución diferente. La nave se pone en marcha desde el botón de enfrente que mira hacia el este, en el panel inferior izquierda de la nave.

El último sistema de tu nave es la estación de recarga y reparación de daños, a la que se puede llegar utilizando



Una tarea muy importante es recargar tu nave en un gigante gaseoso, el líquido burbujeante representa la cantidad de combustible de tus tanques

el botón que apunta hacia el norte en el panel inferior del lado izquierdo. Viajar en el tiempo y el espacio consume combustible y hay que recargar en un gigante gaseoso. Para ello, debes sobrevolar el gigante gaseoso y recoger el combustible. El combustible no solo se utiliza para viajar en el tiempo y el espacio, sino también para reparar la nave si es dañada en combate. Hay diferentes tamaños de gigantes gaseosos, que permiten recoger diferentes cantidades de combustible cuando se sobrevuelan. Incluso podrías exceder la capacidad máxima de combustible que puedes llevar, pudiendo dañar la nave si recoger más de lo que puedes cargar. Además, volar cerca de un gigante gaseoso daña tus escudos de calor y en función del tamaño del gigante gaseoso, esto será más o menos peligroso.

Sugerencia: Es muy práctico establecer tu especie en un planeta que tenga su propio gigante gaseoso. En el mapa de galaxia, puede elegir ver únicamente los planetas que tienen gigantes gaseosos. De esta forma, no tiene que volar a otro sistema cuando tengas que reabastecerte o reparar tu nave.

Consejo estratégico

Lo más importante es guardar la

partida continuamente y en diferentes ranuras, en caso de que algo salga mal, puedes volver atrás e intentarlo de nuevo. Me he dado cuenta que lo mejor es sembrar una especie y centrarse en terminar su desarrollo antes de parar a la siguiente. Aunque aparentemente podrías ir más rápido teniendo varias especies a la vez, sólo conseguirá viajar continuamente en el tiempo y el espacio intentando solucionar sus problemas. Es mucho más fácil centrarse en una sola especie, ya que las especies son muy diferentes en cultura, y te puede resultar muy difícil intentar cambiar la forma de pensar de una mente pacífica que sólo piensa en la evolución de la especie en una raza guerrera, que sólo se nutre de los continuos conflictos y del peligro de llegar a extinguirse.

Trata de identificarte con las creencias de una especie. Una especie agresiva podría no seguir tus órdenes si le pides de forma agradable que haga lo que tú quieres, pero si los amenazas con matarlos a todos si no te obedecen, podrían convencerlos. Con una especie más bien religiosa, puede que tengas que pensar en cómo orientar sus creencias para alcanzar tus metas, o simplemente recordándoles por qué

los enviaste a ese planeta.

Si es posible, extrae inventos y colocalos al inicio de la línea temporal para que las especies se desarrollen más rápido. Mueve un puñado de inventos uno o dos siglos antes cada vez que gestiones mil años de desarrollo, que podría ser los mil años que necesitas para terminar el dispositivo que te permita salir de la galaxia antes de que los Microids intenten atacar la especie.

Echa un vistazo a todos los eventos del histograma, ya que no todos los eventos o inventos pueden ser importantes o tiene alguna influencia, pero más tarde puede causar serios problemas. Intenta sembrar las diferentes especies tan alejadas como te sea posible de los Microids, pero también alejadas entre ellas. Una especie podría desarrollar naves espaciales antes que otra raza y cuando empiecen a expandir su territorio, podrían atacar a las especies que actualmente están en desarrollo y te verás obligado a luchar contra tu propia creación. Por lo tanto, asegúrate de que no se “encuentren” unos a otros demasiado pronto.

Conclusion

Me gusta mucho este juego y ha sido uno de mis favoritos desde siempre, aunque no es muy conocido, y es por eso que lo considero una joya de los juegos raros. El juego es fascinante, tienes que asimilar decisiones políticas, así como hacer frente a creencias religiosas y otros aspectos de una sociedad. Las diferentes especies son únicas, y el modo con el que trabajes con una especie puede no funcionar con otra. El juego es rico en variedad, a menudo me absorbe durante horas tratando de conseguir que una determinada especie esté justamente donde yo quiero.

Las escenas de vídeo son muy buenas para su época, y siento un poco de pena por la persona que tenía que llevar el vestuario de cada una de las especies, pero así realmente lo incluía la configuración. Si no has jugado a este juego todavía, deberías dedicarle algo de tiempo y si ya has jugado, ¿Por qué no echas otra partida con ayuda de tu ODDROID en la gigante TV de tu salón?



¡Cuando completes este juego, echarás de menos tanto tu nave que empezarás a jugar de nuevo!

KUNG FURY: STREET RAGE

LA PEOR PELICULA JAMAS HECHA EN YOUTUBE ES AHORA EL PEOR JUEGO JAMAS CREADO

por Bruno Doiche

Si consiguieras encallar en una isla sin tu ODDROID y sin acceso a Youtube, podrías haberte perdido el fenómeno basura que fue Kung Fury.

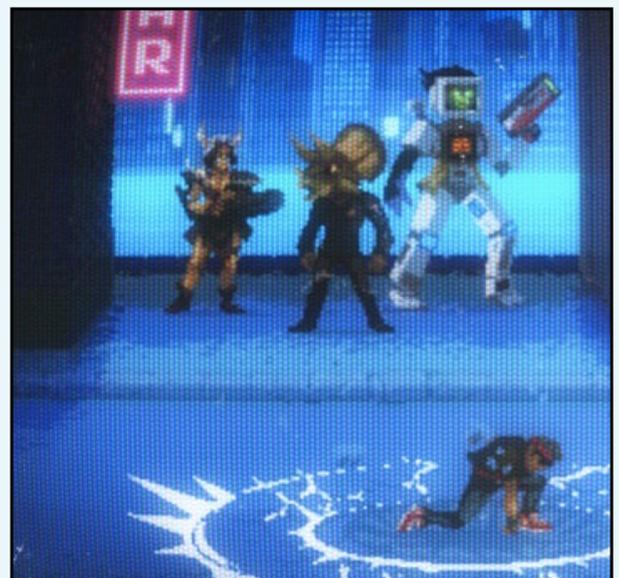
Si todavía no sabes de lo que estoy hablando,

ve a: https://www.youtube.com/watch?v=bs5P_LAqiVg.

Espera a ver y continua leyendo. ¡Lo sé! Impresionante, ¿verdad? Imagínate si hubiese un juego de esta película. ¡Bueno, aquí está! Ve a por él!



<https://play.google.com/store/apps/details?id=se.hellothere.kungfurygame&hl=en>



DESARROLLO ANDROID

COMPILANDO ANDROID STUDIO

por Nanik Tolaram

Como programadores y desarrolladores, generalmente usamos diferentes herramientas para crear aplicaciones y no es muy distinto con Android. Necesitamos herramientas que nos permitan acelerar el código, la depuración y las pruebas, ¡al tiempo que no nos volvamos locos con la depuración! Desde comienzos de año, he cambiado a Android Studio para continuar desarrollando con Android, ya que Google ha dejado de mantener ADT (Android Development Toolkit) para Eclipse. ADT era un plugin para Eclipse aportado por Google para desarrolladores como lo era el IDE de facto (Entorno de Desarrollo Interactivo) para el desarrollo de Java. Ahora cualquier persona que quiera desarrollar en Android tendrá que utilizar Studio como herramienta principal.

Al igual que cualquier otro software, Estudio a veces contiene errores que se van depurando en versiones posteriores. Puesto que es un proyecto de código abierto, los desarrolladores no tienen que esperar mucho tiempo para la siguiente versión, ya que tan pronto como se agregan parches o nuevas características al repositorio de código fuente, simplemente tenemos que descargarlo y compilarlo en local para empezar a usarlo. En este artículo, voy a explicar el proceso de compilación de Estudio desde el código fuente bajo Linux y así podrás utilizarlo como tu IDE diario. Para ello voy a usar Ubuntu 14.04 de 64 bits, sin embargo, se puede utilizar cualquier distribución de Linux o Mac OS.

Requisitos previos

Necesitas tener instalado JDK de O-



racle, que puedes descargar desde <http://bit.ly/196ebsY>. Utiliza el siguiente comando para crear los enlaces simbólicos correspondientes que apunten a java, javac y otras herramientas para la correcta ubicación de JDK:

```
$ sudo update-alternatives --install "/usr/bin/java" "java" \
"/home/nanik/Downloads/
jdk1.7.0_79/bin/java" 1071
$ sudo update-alternatives
--install "/usr/bin/javac"
"javac" \
"/home/nanik/Downloads/
jdk1.7.0_79/bin/javac" 1071
$ sudo update-alternatives
--install "/usr/bin/javaws"
"javaws" \
"/home/nanik/Downloads/
jdk1.7.0_79/bin/javaws" 1071
$ sudo update-alternatives
--install "/usr/bin/javap"
"javap" \
"/home/nanik/Downloads/
jdk1.7.0_79/bin/javap" 1071
$ sudo update-alternatives
--install "/usr/bin/javadoc"
"javadoc" \
"/home/nanik/Downloads/
jdk1.7.0_79/bin/javadoc" 1071
```

También debes instalar las siguientes herramientas. Tras la descarga, extrae el Apache Ant y colócalo en cualquier carpeta local.

- git
- Apache Ant (descargado desde ant.apache.org)

Verificar la fuente

El código fuente de Studio se encuentra alojado en el mismo lugar que el código de Android en <http://bit.ly/1GWeQwC>. El paso para la verificación del código es el mismo que Android, usando la herramienta repo, que se puede descargar usando el comando:

```
$ curl https://storage.
googleapis.com/git-repo-
downloads/repo > \
~/bin/repo
$ chmod 777 ~/bin/repo
```

A continuación, utiliza los siguientes comandos para verificar el código:

```
$ mkdir studio-1.4-dev
$ cd studio-1.4-dev
```

```
$ repo init -u https://android.googlesource.com/platform/manifest \
  -b studio-1.4-dev
$ repo sync -j4 --no-clone-bundle
```

Compilación

Compilar Studio es sencillo ya que utiliza Ant como proceso de compilación, los extractos de build.xml se muestran debajo. El archivo build.xml se puede encontrar dentro del directorio studio-1.4-dev/tools/idea.

```
<!--
  This build script compiles IntelliJ IDEA. Options
  include:
    -Dout=/path/to/out/dir, defaults to ${basedir}/
  out
    -Dbuild=123, defaults to SNAPSHOT
    -Dtestpatterns=com.foo.*, defaults to empty
  string
    -Dproduct=foo, defaults to studio
  -->
<project name="IntelliJ IDEA Community Edition"
  default="all">
  <property name="project.home" value="${basedir}"/>

  <condition property="out.dir" value="${out}"
  else="${project.home}/out">
    <isset property="out" />
  </condition>

  <condition property="build.number" value="${build}"
  else="SNAPSHOT">
    <isset property="build" />
  </condition>

  <condition property="test.patterns"
  value="${testpatterns}"
    else="org.jetbrains.android.*;com.
  android.tools.idea.*;com.google.gct.*;com.intellij.
  android.*">
    <isset property="testpatterns" />
  </condition>

  ....
  ....
  ....

  <target name="all" depends="cleanup,build,fullupda
  ter"/>
</project>
```

Cambia al directorio studio-1.4-dev/tools/ideas y asegúrate de que el directorio /bin del Apache Ant está incluido en la variable PATH. Por ejemplo, el mío se encuentra en /home/nanik/apache-ant-1.9/bin. Una vez dentro del directorio, inicia el proceso de compilación ejecutando el comando ant:

```
$ ant
```

Verás un registro log similar al de abajo cuando se completa el proceso de construcción.

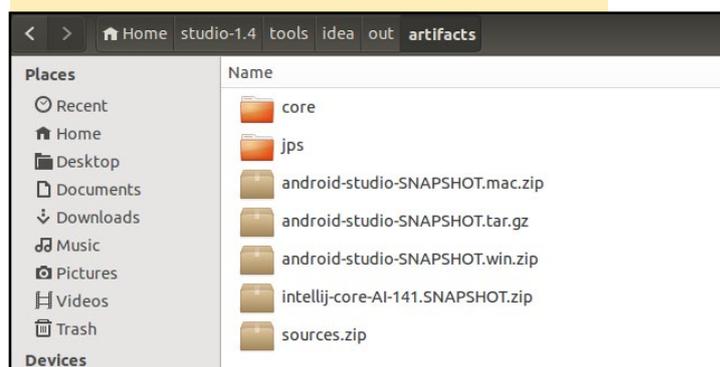
```
Buildfile: /home/nanik/studio-1.4/tools/idea/build.xml

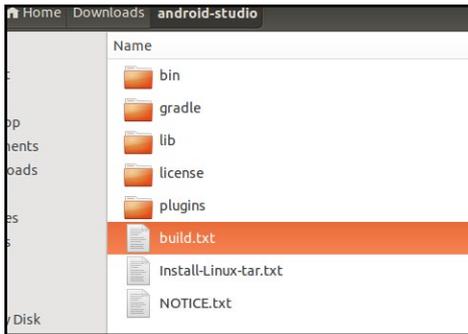
cleanup:

init:
  [mkdir] Created dir: /home/nanik/studio-1.4/
  tools/idea/out
  [mkdir] Created dir: /home/nanik/studio-1.4/
  tools/idea/out/tmp

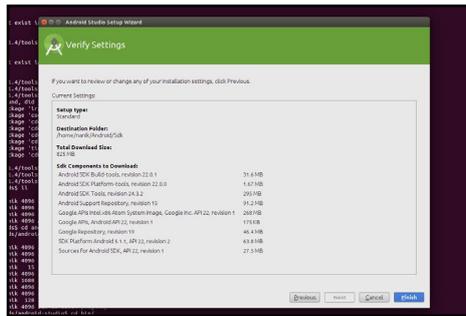
build:
  [java] Buildfile: /home/nanik/studio-1.4/tools/
  idea/build/gant.xml
  [java]
  [java] doGant:
  [java] 'home' is not defined. Defaulting to '/'
  home/nanik/studio-1.4/tools/idea'
  [java] default:
  [java] compile:
  ....
  ....
  [java] Build log (info) will be written to /
  home/nanik/studio-1.4/tools/idea/out/tmp/system/
  build-log/build.log
  [java] Loaded project /home/nanik/studio-1.4/
  tools/idea: 264 modules, 80 libraries
  [java] [mkdir] Created dir: /home/nanik/stu-
  dio-1.4/tools/idea/out/dist.win.ce
```

Figura 1: Pack de Android Studio para 3 plataformas

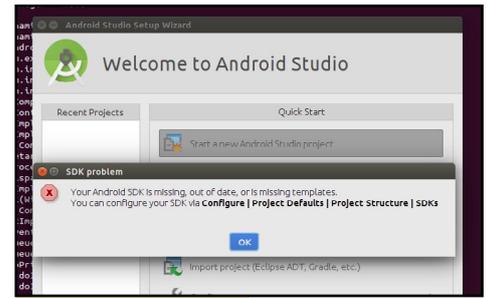




Contenido de android-studio-SNAPSHOT.tar.gz



Descargar SDK



Configurar SDK

```
[java] [mkdir] Created dir: /home/nanik/studio-1.4/tools/idea/out/dist.all.ce
....
....
....
[java] [tar] Building tar: /home/nanik/studio-1.4/tools/idea/out/artifacts/android-studio-SNAPSHOT.tar
[java] [gzip] Building: /home/nanik/studio-1.4/tools/idea/out/artifacts/android-studio-SNAPSHOT.tar.gz
[java] [delete] Deleting: /home/nanik/studio-1.4/tools/idea/out/artifacts/android-studio-SNAPSHOT.tar
....
....
[java] [jar] Building jar: /home/nanik/studio-1.4/tools/idea/out/___tmp___/_0/updater.jar
[java] [copy] Copying 1 file to /home/nanik/studio-1.4/tools/idea/out
[java] ----- default
[java]
[java] BUILD SUCCESSFUL
```

Ejecución

Si la compilación se realiza correctamente, obtendrás Studio empaquetado dentro de /studio-1.4/tools/idea/out/artifacts para 3 plataformas diferentes: Windows, Mac y Linux, como se muestra en la Figura 1.

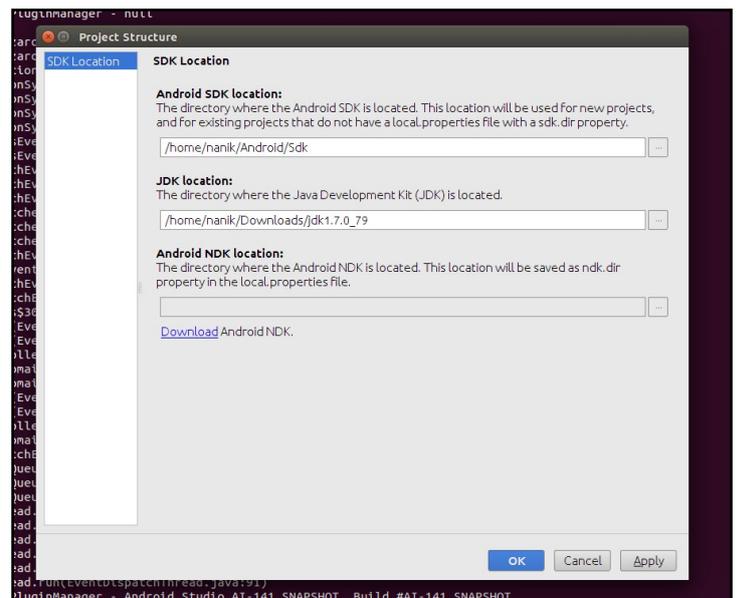
Puesto que uso Linux, extraje el archivo android-studio-SNAPSHOT.tar.gz, cuyo contenido se muestran en la Figura 2. Para iniciar Studio, basta con ejecutar el studio.sh dentro del directorio /bin.

Si tienes el SDK instalado, será detectado automáticamente por Studio. De lo contrario, verás una pantalla que te indica que tienes que descargar el SDK tal y como se muestra en la Figura 3.

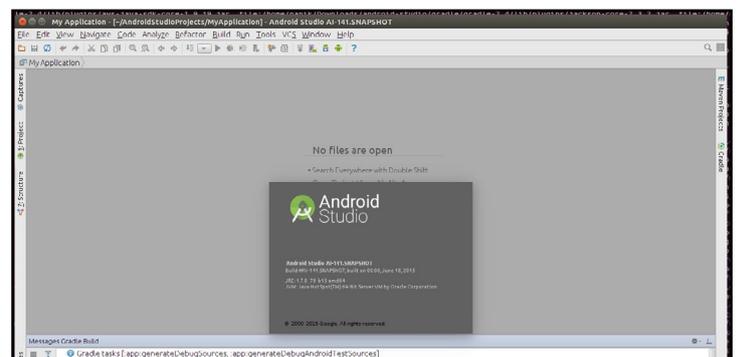
Una vez finalizada la descarga, aparecerá una pantalla simi-

lar a la Figura 4. Todo lo que tienes que hacer es seguir las instrucciones para localizar tu JDK y SDK cuando se te soliciten. Completa la pantalla que se muestra en la Figura 5 con la ubicación correcta.

Habilitar por ti mismo el IDE te permitirá mantenerte siempre al día de los últimos cambios, lo cual te proporciona un extra que te ayudará a desarrollar tu aplicaciones mucho más rápido.



Configuración de la ubicación correcta del SDK y JDK

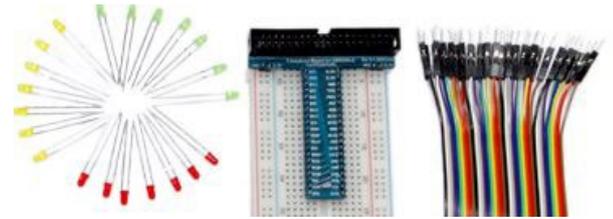


Listo para crear tu impresionante aplicación Android

JODRO

LIBRERIA JAVA PARA CONTROLAR LOS PINES GPIO DEL ODROID-C1

por @ChromoDev
editado por Rob Roy



Empecé a escribir esta librería porque no lograba encontrar una librería Java como Pi4j para ODROID-C1. Este proyecto se encuentra en desarrollo, si tienes alguna idea para mejorarlo o incluir nuevas funciones ponte en contacto conmigo a través del enlace incluido al final del artículo.

Instalación

Descarga el proyecto desde <http://bit.ly/1RROajs> y agrega jOdro.jar desde la carpeta dist a tu proyecto. Ejecuta el siguiente comando en tu Odroid para dar a la librería los permisos necesarios.

```
$ sudo chmod 222 /sys/class/gpio/
export /sys/class/gpio/unexport
```

Uso

De momento puedes ajustar y leer un pin. Aquí tienes un proyecto de ejemplo para que empieces a usar jOdro:

```
public class Tester{
    private static final int delay
    = 500;

    GPIOPin led;
    GPIOPin in;

    public void startTest() {
        led = new
        GPIOPin(OdroPin.GPIO_24, PinMode.
        OUT, PinState.LOW);
        in = new GPIOPin(OdroPin.
        GPIO_23, PinMode.IN);

        Runtime.getRuntime().
        addShutdownHook(new Thread(() ->
        {
```

```
        led.shutdown();
        in.shutdown();
    });

    while(true){
        led.toggle();
        System.out.
        println(in.read());
        try {
            Thread.
            sleep(delay);
        } catch
        (InterruptedException ex) {
            Logger.
            getLogger(Main.class.getName()).
            log(Level.SEVERE, null, ex);
        }
    }
}
```

En primer lugar, tienes que definir un pin GPIO, que es una representación software de un pin hardware. Para este pin tienes que definir qué pin hardware quieres seleccionar, por ejemplo (OdroPin.GPIO_24), en qué dirección debería funcionar el pin (pinMode.OUT) y opcionalmente, el valor por defecto (como PinState.LOW). Después puedes manipular o leer el pin conforme a la figura 1.

Al final del programa, tienes que cerrar los pines que se restablecen por defecto (bajo y entrada) y sin exportarlos.

Clase GPIOPin

La clase GPIOPin representa el pin hardware en el código.

OdroPin	GPIO #	Funktion	WiringPi
GPIO_00	#74	SDA1	--
GPIO_01	#75	SCL1	--
GPIO_02	#83	--	7
GPIO_03	#88	--	0
GPIO_04	#116	--	2
GPIO_05	#115	--	3
GPIO_06	#107	MOSI	12
GPIO_07	#106	MISO	13
GPIO_08	#105	SCLK	14
GPIO_09	#76	SDA2	--
GPIO_10	#101	--	21
GPIO_11	#100	--	22
GPIO_12	#108	--	23
GPIO_13	#97	--	24
GPIO_14	#113	TXD1	--
GPIO_15	#114	RXD1	--
GPIO_16	#87	--	1
GPIO_17	#104	--	4
GPIO_18	#102	--	5
GPIO_19	#103	--	6
GPIO_20	#117	CE0	10
GPIO_21	#118	--	11
GPIO_22	#77	SCL2	--
GPIO_23	#99	--	26
GPIO_24	#98	--	27

Figura 1 – Distribución GPIO

```
// Constructor without default
state (set to LOW)
public GPIOPin(OdroPin pin,
PinMode mode)
```

SENSACION DE PAZ

GENERADOR DE RUIDO BLANCO

por Bruno Doiche

Tiene compañeros de trabajo insensibles que piensan que los teclados ruidosos están de moda en el 2015. Un robot Roomba limpia tu piso de forma automática, mientras que tú intentas gestionar un par de hojas de cálculo. ¡Si este es tu caso, pillate un generador de ruido blanco!

```
$ sudo apt-get install sox
```

Ruido blanco

```
$ play -n synth 60:00 whitenoise@
```

Ruido marrón

```
$ play -n synth 60:00 brownnoise
```

Ruido rosa

```
$ play -n synth 60:00 pinknoise
```

Disfruta de tu propio espacio personal durante toda una hora. Si te gusta y quiere mantenerlo indefinidamente, sólo tienes que usar el temporizador de 60 minutos, aunque los tímpanos pueden molestarte de vez en cuando.



```
// Constructor with default State
public GPIOPin(OdroPin pin,
PinMode mode, PinState state)

// Sets the state of the pin to
low
public void low()

// Sets the state of the pin to
high
public void high()

// Reverses the state of the pin
public void toggle()

// Reads the state of the pin
public PinState read()

// Shuts down the pin
public void shutdown()

// Returns the constant for the
pin
public OdroPin getPin()

// Returns the mode of the pin
public PinMode getMode()
```

PinState

La clase PinState representa el valor del pin en el código.

```
// Constant for a low value
PinState.LOW

// Constant for a high value
PinState.HIGH

// Returns the value which is
used to control the GPIOs
public string getCode()

// Returns the state state as a
boolean
public boolean toBool()

// Returns the state as an
integer
public int toInt()

// Returns the state as a string
```

```
public string toString()
```

PinMode

La clase pinMode representa el modo del pin en el código.

```
// Constant for input
PinState.IN

//Constant for output
PinState.OUT

// Returns the value which is
used to control the GPIOs
public string getCode()

// Returns the mode state as a
boolean
public boolean toBool()

// Returns the mode as a int
public int toInt()

// Returns the mode as a String
public String toString()
```

OdroPin

La clase OdroPin representa la dirección del pin en el código.

```
// Returns the value which is
used to control the GPIOs
public int getOdroidCode()

// Returns the function of a
pin. If there is no function, it
returns the number.
public string getLabel()

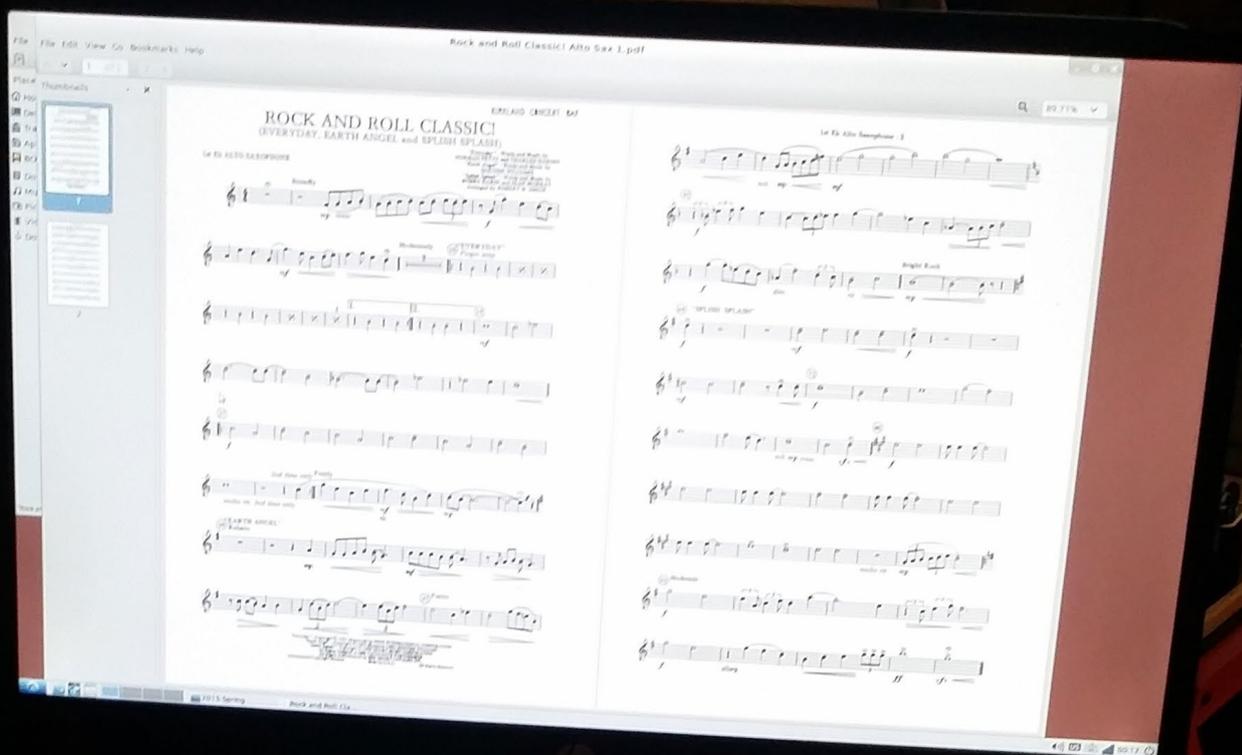
// Returns the GPIO number in the
WiringPi Protokol
public int getWiringPin()
```

Para preguntas, comentarios o sugerencias, por favor visita el repositorio jOdro de GitHub en <http://bit.ly/1HweC2B>.

ATRIL CON ODRID-C1

SESION DE MUSICA CON ESTILO

por Ivan Reede



Al formar parte de unas cuantas bandas de música y una orquesta, tengo que llevar muchos cuadernos con partituras. Me resulta muy incomodo transportar tantos cuadernos, averiguar qué partituras necesitaba en cada caso y no olvidar los cuadernos específicos que necesito para practicar en cada banda en concreto. Para mí, el papel es un medio muy primitivo a la hora de transportar partituras que tiene varias dificultades añadidas. Como soy ingeniero y músico, decidí crear un atril electrónico.

Primero intenté usar una tablet de 10" para reemplazar las hojas de papel, pero la pantalla era demasiado pequeña. Tenía una buena aplicación en la tablet que era perfecta para reproducir la música, pero la imagen aún no era lo suficientemente grande. Quería algo que me permitiera mostrar dos páginas de lado a lado, como mi atril. Al tocar el saxofón, ambas manos las tengo ocupadas y utilizar una pantalla táctil para pasar las hojas me representa un problema. Intente usar un pedal de pie, pero eso sólo suponía tener que acarrear con otra cosa más.

Mi aventura con la tablet llegó a su fin cuando mi frágil atril se balanceo y la tablet se cayó al suelo, agrietándose la pantalla. Definitivamente necesitaba algo mejor. Empecé escribiendo las especificaciones del nuevo atril. Me imaginaba utilizando un soporte estándar, un monitor, un ordenador y unas baterías recargables, todo mecánicamente acoplado en una sola unidad funcional.

Especificaciones Soporte

Placa posterior: Placa de 13.5" x 19" con una balda de 2" con bordes redondos. Orificios para sujetar el soporte del micrófono. Tiradores de inclinación para adaptar los ángulos de visión.

Sistema de ajuste a media altura: botón de bloqueo, ajustable en altura desde 24" (sentado) a 45" (de pie)

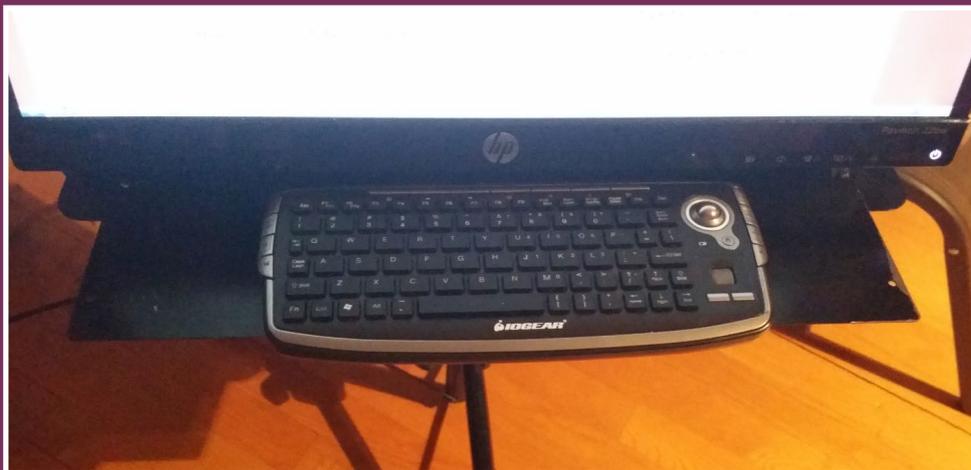
Base: trípode robusto con patas de goma antideslizantes y brazo extensible que se pliega fácilmente para los viajes.

Especificaciones Monitor

Quería que fuese capaz de mostrar al menos 2 páginas de música al mismo tiempo, con un tamaño aproximado de 8.5" x 11". El monitor tenía que encajar muy bien en la plataforma y ser montado con firmeza. Necesitaba una pantalla anti-reflejos para evitar los reflejos del sistema de iluminación del escenario y de la luz de sol. La imagen tenía que ser visible fuera, a plena luz del día y el monitor tenía que ser capaz de funcionar con baterías durante al menos 6 horas hasta tener que volver a recargarlas.

Especificaciones Ordenador

El equipo tenía que ser pequeño y ligero, con suficiente espacio de almacenamiento para guardar una gran cantidad de música, con una salida HDMI para conectar al monitor y puertos USB que permitiesen la fácil transferencia de



Para el control Iván ha elegido un impresionante teclado con rueda de desplazamiento.

datos, capaz de soportar WiFi y periféricos Bluetooth. Al igual que el monitor, tenía que ser capaz de funcionar con baterías durante al menos 6 horas.

He encontrado un monitor HDMI de 22 pulgadas con una fuente de alimentación externa de 20V que tiene aproximadamente la misma anchura que la parte posterior del atril. Para asegurar el monitor, le quité su pedestal y todas sus piezas de plástico manteniendo el marco de metal interior. Lo comprobé y me aseguré de que el monitor se podía sujetar fuertemente al pedestal sin las piezas de plástico. Con un poco de paciencia, perforé la placa posterior del atril para acoplar la base del monitor. Esto fue bastante fácil, ya que el pedestal utiliza tornillos para sujetar el plástico que normalmente se usan para un atril con pantalla LCD.

El sistema operativo de la tablet era en la práctica un impedimento. Los sistemas operativos de las Tablet están orientados hacia el consumo de información en lugar de la producción de información. Por lo tanto, me puse a buscar

Iván sin duda hizo un gran montaje y puede gestionar la fuente de alimentación



un sustituto apropiado. Linux parecía ser un SO interesante, el cual llevo usando desde hace muchos años. En su mayor parte, es de código abierto y dedicándole algo de tiempo, puedes hacer que haga lo que tú quieras en lugar de lo que quiere la mayoría.

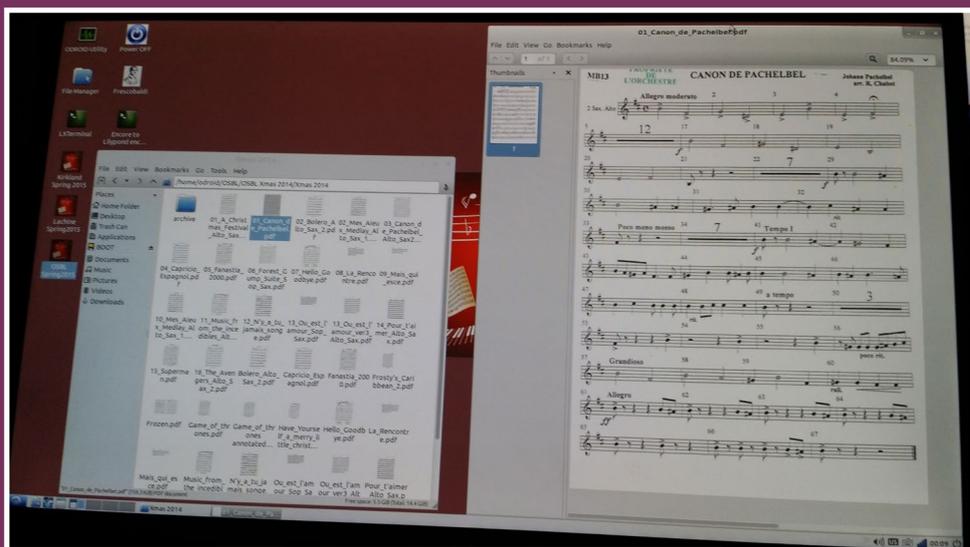
Al principio, intente usar una Raspberry Pi, que no llevo a funcionar muy bien. Se quedó sin memoria bastante rápido, y tras cargar unas diez piezas musicales se volvía muy lenta. Me llevaría unos 20 minutos cargar las hojas de música que necesitaba para un concierto y el cambio de página llegaba a tardar hasta 30 segundos.

Era suficiente para practicar en casa, pero seguramente no lo utilizaría en una sala de conciertos. Aún así lo probé, con mis compañeros músicos que se reían de mí lento artillugio con toneladas de cables, fuentes de alimentación y un ordenador realmente lento. Se preguntaban cómo podría llegar alguna vez a sustituir las hojas en papel.

Luego apareció un nuevo candidato, el ODROID-C1. Tenía el doble de memoria, cuatro núcleos de CPU, aproximadamente el doble de velocidad de reloj y una bonita carcasa rectangular, así que compré tres y les di una oportunidad. De repente, mi atril electrónico se convirtió en algo mucho más viable. Los 20 minutos que necesita para cargar mis piezas musicales para un concierto se redujeron a 45 segundos, lo cual fue genial. Finalmente el atril comenzaba a tener un sentido práctico. Realmente podía asumir los 45 segundos de tiempo de configuración en un espectáculo. El cambio de páginas era bastante rápido, pero no lo suficiente todavía.

Tras varios intentos, finalmente encontré un buen visor PDF, rápido y simple. Tan sólo tenía que poner toda mi música en una carpeta, abrirla y elegir el archivo en cuestión con un simple doble clic. Luego añadí un icono en el escritorio por concierto. Cambie los nombres de los archivo por un número de 2 dígitos (00, 01, 02) para que mi música apareciese por orden de reproducción, estaba empezando a ser divertido. El ratón, en cambio, ha resultado ser un problema en los conciertos.

Todavía tenía un problema de movilidad, el hecho de cómo alimentar la unidad. Tenía que cargar con las fuentes de alimentación, cables y demás.

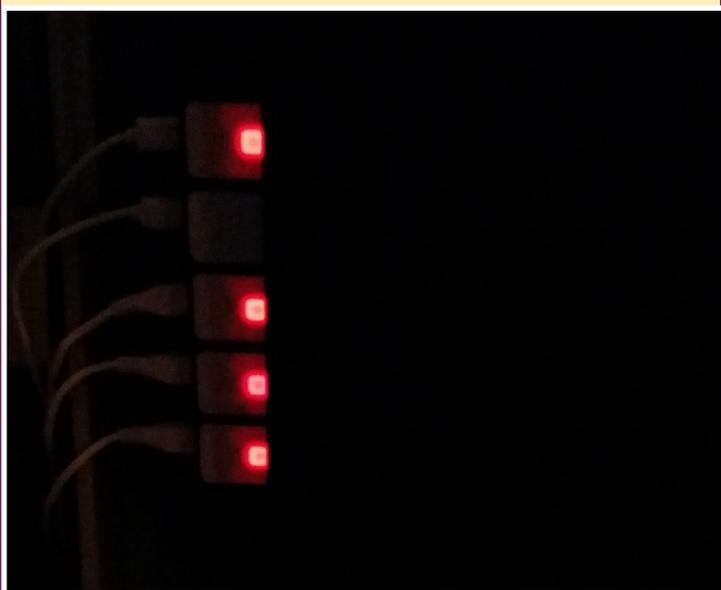


Todas sus partituras al alcance de sus manos en formato pdf.

Corriendo en cada concierto en busca de un enchufe, alargaderas, cinta adhesiva para fijar los cables al suelo y así evitar que la gente tropezara. Tenía que buscar una solución mejor, así que compré un pack de baterías USB de potencia de 10Ah y 2.1 amperios. Las pruebas mostraron que el ODRROID-C1 podía aguantar 19 horas encendido. ¡Excelente! Ahora pasaríamos al monitor, que resultó ser un gran reto.

El monitor necesita 20 voltios, así que pensé en utilizar cuatro baterías USB de 5 voltios en serie, pero me equivoqué. El monitor tira de una impresionante cantidad de energía cuando se enciende, cuando se ilumina y cuando se pasa la página. Sin embargo, cuando la imagen está fija y no cambia nada, el consumo de energía se reduce al mínimo. Las fuentes de alimentación se apagaban de forma aleatoria mientras estaba reproduciendo una partitura, aun cuando nada cambiaba en pantalla.

Los puertos de carga iluminados son una gran ventaja



El resultado final era que las fuentes de alimentación se apagaban mientras estaba tocando o cuando pasaba las páginas. Descubrí que podría deberse a que no recibían carga o, se producía una sobrecarga.

La solución fue retirar por completo la regulación electrónica de las baterías USB. Así que, destroce las fuentes de alimentación USB y quité las baterías de Li-ion. Coloque las células en paralelo para crear una batería con la capacidad necesaria amperios-hora. Conectando 5 de estas baterías alcanzaba 21 voltios con carga completa y 17 voltios cuando se descargaba. El regulador interno del monitor puede trabajar con eso. Por último, fabrique una carcasa para las baterías y la fije a la parte trasera del atril. Este sistema funcional me permitía disponer de 6 horas de autonomía, y no tener cables de alimentación molestos de por medio. Lo que es mejor aún, la mayoría de las conexiones seguían en su lugar, por lo que la configuración del atril era muy similar a la de un atril para papel normal.

Como sistema de emergencia, amplíe la balda inferior con un clip, de modo que ahora si quiero, puedo poner las hojas de papel en el soporte. También es muy útil para colocar mi teclado. Mis compañeros músicos están empezando a ver este sistema bastante útil. Mejor aún, con un dongle WiFi conectado al puerto USB y mi teléfono como punto de acceso, puedo coger cualquier pieza de música que necesite desde mi servidor personal. El C1 puede almacenar toda la colección de música de la banda sin tener que necesitar impresoras y sin papeles que salen volando en las actuaciones al aire libre. ¡Realmente es genial!

Añadiendo las aplicaciones Lilypond y Frescobaldi, puedo escribir música y hacer correcciones sobre la marcha. El atril puede reproducir música y grabar tanto las actuaciones como las prácticas, permitiéndome escucharlas más tarde y así estudiar donde puedo mejorar. También puede ayudarme a afinar los instrumentos.

Con todo esto, quisiera dar las gracias a Hardkernel por esta pequeña maravilla técnica. Habéis hecho que mi atril se haga realidad. Está listo para acompañarme a los conciertos como si fuese un atril electrónico comercial a tamaño real. Con mi último experimento, usando VNC, un ayudante podía mostrar hojas, partituras musicales y mensajes directamente en mi pantalla. Lo cual me permitía concentrarme en la reproducción de música, en lugar de pasar las páginas. El siguiente paso es añadir el software necesario que me permita tener una red inalámbrica de atriles para bandas y orquestas.

SO DESTACADO

DIETPI PARA ODROID-C1

por Daniel Knight



¿Qué es DietPi?

En esencia, DietPi es una “imagen” con una instalación mínima de Ubuntu:

Hemos desmontado la imagen oficial de Hardkernel eliminando prácticamente todo, para ofrecer una imagen básica a la que hemos llamado DietPi-Core.

Con Ramlog, un servidor SSH Dropbear y ajustes para reducir el uso de memoria/CPU, la imagen DietPi viene pre-optimizada y lista para ejecutarse.

Estadísticas básicas

Expansión automática del sistema de archivos

DietPi ampliará automáticamente tus sistemas de ficheros en el primer reinicio.

Esto garantiza que podrás acceder a toda tu tarjeta MicroSD.

Ramlog

Reduce el E/S del Sistema de Archivos y ahorra el tener que escribir en la tarjeta SD al mover /var/log a la RAM

Dropbear

Liviano servidor SSH instalado por defecto. Se puede cambiar por OpenSSH-Server usando DietPi-Software si necesitas SFTP/SCP.

Soporte Wifi

Usando DietPi-Config, puedes conectarte con rapidez y con facilidad a tu red Wifi.

Bajo consumo de memoria Ubuntu

< 98 MB de RAM en el arranque.

Bajos recursos

Un total de 11 procesos en el arranque.

Swapfile

100mb con swappiness ajustado en 1 (para evitar errores por falta de memoria).

Unidad dedicada USB opcional

Si tienes previsto usar una unidad USB con tu instalación, DietPi pondrá en mar-

```

1 [ 0.0%] Tasks: 12; 2 running
2 [ 0.0%] Load average: 0.04 0.07 0.04
3 [ 1.3%] Uptime: 00:02:45
4 [ 0.0%]
Mem[|||||] 97/958MB
Swp[ ] 0/99MB

USER NI CPU% MEM% TIME+ Command
root 0 1.3 0.1 0:00.19 htop
root 0 0.0 0.2 0:03.43 /sbin/init
root 0 0.0 0.1 0:00.44 mountall --daemon --fsck-fix
root 0 0.0 0.1 0:00.61 upstart-udev-bridge --daemon
root 0 0.0 0.1 0:00.34 /lib/systemd/systemd-udevd --daemon
root 0 0.0 0.1 0:00.10 upstart-socket-bridge --daemon
root 0 0.0 0.1 0:00.10 upstart-file-bridge --daemon
root 0 0.0 0.1 0:00.00 cron
root 0 0.0 0.0 0:00.00 /usr/sbin/dropbear -d /etc/dropbear/dropbear_ds
root 0 0.0 0.1 0:00.00 /sbin/getty -8 38400 tty1
root 0 0.0 0.1 0:00.50 /usr/sbin/dropbear -d /etc/dropbear/dropbear_ds
root 0 0.0 0.2 0:00.20 -bash

```

97 Megabytes usados, 12 procesos con HTOP, esta es una super imagen slim

cha tu unidad USB y configurará automáticamente todo el software que se instale en un futuro con DietPi-Software para que use tu dispositivo USB en lugar de la SD.

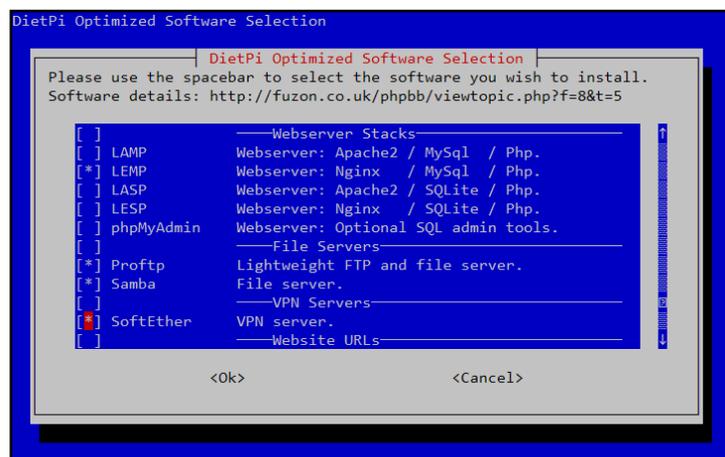
Posibilidades

Compilado desde cero, DietPi-Software permite instalar aplicaciones opcionales, las cuales están pre-configuradas y “listo para ejecutarse” con todas las optimizaciones y configuraciones ya realizadas.

Si estás buscando un servidor web LÁMP (Nginx / MySql / Php), o la combinación Kodi y servidor BitTorrent, DietPi-Software los instala, configura y optimiza. Las optimizaciones aplicadas incluyen tamaño del cache php, tamaño de caché del servidor bittorrent, cantidad de subprocesos Nginx/php5-fpm y muchos más.

Con la optimizaciones específicas que se aplican automáticamente a tu hardware, DietPi te asegura el máximo rendimiento de tu dispositivo ODROID y del software que decidas instalar.

La lista completa de opciones de DietPi la puedes encontrar online, por favor visita <http://fuzon.co.uk/phpbb/viewtopic.php?f=8&t=11#p11>



DietPi tiene su propia librería de software optimizado

Configuración

Dentro de DietPi-Config puedes cambiar fácilmente las opciones de visualización, conectar a un red wifi, configurar una dirección IP estática, modificar los parámetros de la CPU entre otras opciones. Con la integración del cliente Samba, NoIp y CurlFtpFs, puede acceder fácilmente a archivos compartidos en red o asignar a tu dispositivo una dirección web fija con facilidad. DietPi-Config es un paraíso para los que le gusta hacer ajustes.

Instalación

Descarga la imagen DietPi, configura un disco duro USB dedicado e instala el siguiente software optimizado para DietPi con DietPi-software:

- Owncloud - Tu propio sistema personal de backup
- Transmission - Servidor BitTorrent con interfaz web
- Kodi - La cúspide de los centros multimedia
- Servidor web LAMP - Apache2, MySql, PHP-5 (usado por ownCloud).
- Servidor Samba - Para acceder a las descargas de BitTorrent y a los archivos del dispositivo de forma remota.

Instalación

También cubriremos la configuración de NoIp con DietPi-Config. Dale a tu ODROID-C1 un poco de formalidad. Todo lo que necesitarás es:

- **ODROID-C1.**
- **Tarjeta MicroSD de 2 GB o superior.**
- **Acceso a Internet (Ethernet o Wifi, para completar la configuración DietPi)**
- **Se recomienda un disco duro USB para las instalaciones del servidor Transmission BitTorrent u ownCloud. DietPi moverá automáticamente los datos al disco duro USB si está instalado.**

Las guías y documentación online están disponibles aquí:

<http://fuzon.co.uk/phpbb/viewtopic.php?f=8&t=9#p9>

Descarga DietPi para ODROID-C1 en el siguiente enlace:

<http://goo.gl/UF6I0F>

Escribe la imagen en tu tarjeta MicroSD

- **Descomprime/extrae la imagen DietODROID.7z.**
- **Escribe el archivo de imagen DietODROID_vxx.img en tu tarjeta MicroSD.**

La documentación on-line incluye los métodos para escribir la imagen con Windows y Linux.

- **Inserta la tarjeta microSD en tu dispositivo ODROID-C1 y enciéndelo.**
- **DietPi ampliará automáticamente el sistema de archivos y se reiniciará dos veces cuando haya terminado.**
- **Cuando aparezca la pantalla de inicio de sesión, escribe "root" como nombre de usuario y "raspberrypi" como contraseña.**

DietPi también viene con un liviano servidor SSH (Dropbear) pre-instalado. Simplemente tiene que usar la dirección IP de tu dispositivo ODROID y los datos de acceso anteriores.

DietPi comprobará ahora si hay actualizaciones. Si se aplican las actualizaciones, aparecerá un aviso para reiniciar el sistema, pulse Intro. Cuando vuelva a aparecer la pantalla de inicio de sesión, vuelve a entrar.

Unidad USB

DietPi te pedirá que respondas a las preguntas sobre tu instalación, pulse Intro para continuar.

- **Ahora aparecerá la pantalla de disco duro USB. Si tienes un disco duro USB, selecciona USB Install, pulsa intro y sigue las instrucciones en pantalla.**

Si el disco duro USB ya está formateado en ext4 o NTFS, se te dará la opción de mantener los datos o formatear en ext4.

Software

En el menú principal, seleccione la opción DietPi Optimized Software y presiona Intro.

- Usa la barra espaciadora para seleccionar Kodi, Transmisión, ownCloud. Luego, pulsa Intro.
- No es necesario seleccionar LAMP, ya que DietPi lo instalará automáticamente
- Cuando aparezca el aviso “File Server Recommended”, pulsa Intro.
- Cuando aparezca el aviso “Boot Options”, selecciona Yes y pulsa Intro. Desde aquí, puedes elegir qué software se iniciará automáticamente en el arranque. Selecciona Kodi de la lista y presione intro. Cuando hayas terminado, pulse ESC para volver al menú principal.

Seleccionar un Servidor de archivos

En el menú principal, selecciona la opción de File Server y pulsa Intro.

- **Seleccione Samba de la lista de servidores de archivos disponibles y pulsa Intro. Confirma cuando aparezca el aviso para volver al menú principal.**

Los servidores SSH y servidores de archivos se pueden intercambiar fácilmente en cualquier momento simplemente ejecutando Dietpi-software desde el terminal. DietPi instalará automáticamente la nueva opción eliminando la anterior.

Iniciar la instalación

Cuando estés listo instala tus elecciones:

- **Simplemente selecciona Go Start Install desde el menú principal y pulsa Intro.**

DietPi comenzará el proceso de instalación y automáticamente instalará, configurará y optimizará tus opciones.

Una vez que DietPi haya terminado de instalar las opciones seleccionadas, el sistema se reiniciará. Esto completa la instalación de tu software.

Usar el software instalado

Obtener tu dirección IP:

En esta guía estamos utilizando la dirección IP 192.168.0.100. Esta tiene que ser reemplazada por la dirección IP de tu dispositivo ODROID.

Puedes obtener tu dirección IP ejecutando dietpi-config. Selecciona el menú de opciones de red, luego, escoge ethernet o wifi.

Usar Kodi:

Al seleccionar Kodi con la opción de auto arranque, éste se cargará automáticamente. Si no ha seleccionado Kodi para que se inicie durante el arranque, puede ejecutarlo escribiendo startkodi. También puedes cambiar las preferencias de arranque ejecutando dietpi-config desde el terminal y seleccionando la opción AutoBoot.

Acceso a unidad USB:

Todos las unidades montadas por DietPi se pueden localizar en la carpeta /mnt/. Si desea hojear tu unidad USB, simplemente ve a /mnt/usb_1.

Usar Transmission (BitTorrent):

```
url = http://192.168.0.100:9091
username = root
password = raspberry
```

Acceso a los datos descargados:

Puesto que instalamos el servidor Samba, podemos acceder a las descargas de forma remota. Desde Windows, simplemente guarda la clave de Windows y pulsa R. Luego, introduce la siguiente dirección.

```
address = \\192.168.0.100\dietpi or \\dietpi\dietpi
username = root
password = raspberry
directory = downloads
```

Usar Owncloud:

Acceso a la interfaz web:

```
url = http://192.168.0.100/owncloud
```

La primera vez que te conectes:

Crea tu cuenta admin escribiendo un nuevo nombre de usuario y contraseña.

Haz clic en Storage & Database para expandir el submenú.

Database type = Seleccionar MySQL

DataFolder = Cambiar a /var/lib/owncloud/data

Database User = root

Database Password = raspberry

Database Name = owncloud

Haga clic en Finish Setup para completar la configuración de ownCloud

Usar LAMP Webserver:

Acceso a la página web:

```
url = http://192.168.0.100
```

```
directorio local = /var/www
```

Acceso a phpinfo:

```
url = http://192.168.0.100/phpinfo.php
```

Acceso PHP cache info:

```
url = http://192.168.0.100/apc.php
```

MySQL Details:

```
username = root
```

```
password = raspberry
```

Instalar NoIp:

Usar NoIp te permitirá dirigir una dirección URL (ejemplo: <http://MyWebsite.noip.biz>) a tu dispositivo ODROID, independientemente de tu dirección IP pública.

Registrarse en NoIp:

Crea tu cuenta NoIp gratis en <https://www.noip.com/sign-up> y elije una dirección web para tu cuenta.

Activar tu cuenta con DietPi:

DietPi-Config es una herramienta de configuración con muchas funciones para tu dispositivo. Una de sus principales características es la posibilidad de instalar y configurar el cliente NoIp, el cliente Samba y el cliente FTP con gran facilidad.

Desde terminal, ejecuta dietpi-config
 Selecciona el menú **Networking Options**
 Selecciona **NoIp** de la lista y selecciona **Install** para instalarlo. DietPi instalará ahora automáticamente **NoIp**.
 Cuando se complete la instalación, selecciona **NoIp** desde el menú.
 Desde aquí puedes introducir tu dirección de correo electrónico **NoIp** y contraseña.
 Pulse **Intro** para el retos de opciones.

Si has introducido tus datos correctos, el estado actual **NoIp** cambiará a **Online**.

Abril puertos del router:

Si desea acceder a tu sitio web desde fuera de la red local, tendrá que habilitar el reenvío de puertos en tu router. Esto permitirá el acceso externo a tu sitio web. Simplemente habilitar el puerto **TCP 80** y dirígelo a tu dispositivo **ODROID**.

Figura 1 - "DietPi - Figura 1 - DietOdroid_htop.png"

```

----- DietPi - Performance Options -----
Hardware : oDroid C1
ARM Temp: 39.0 'c
RAM: 981 MB | Used: 155 MB | Free: 826 MB

1 Change Overclocking Profile
2 CPU Governor      : ondemand
3 CPU Throttle Up  : 60%

<Ok>                <Back>
  
```

Figura 1 - "DietPi - Figura 1 - DietOdroid_htop.png"

```

DietPi - Config
--- DietPi - Network Options ---

Ethernet : Available | Disabled | Disconnected
Wifi      : Available | Enabled | Connected
Internet  : Online
Samba client : /mnt/samba | Size 932G | Used 297G | Free 636G
FTP client  : Not Installed
NoIp status : Not Installed

Ethernet  Change Wired Network Settings
Wifi      Change Wireless Network Settings
Test      Run the Internet Connection Test
Samba Client Input/Modify Details
Ftp Client Install now to access FTP as filesystem mount
NoIp      Install No-IP now

<Ok>                <Back>
  
```

POPCORN TIME

EL SUEÑO DE LOS AMANTES DEL MUNDO MULTIMEDIA

por László Leonard

La aplicación original Popcorn Time es un cliente BitTorrent de código abierto multiplataforma que incluye un reproductor multimedia. El programa y sus derivados con el mismo nombre son alternativas gratuitas para servicios de video en streaming con suscripción como Netflix. Popcorn Time utiliza la descarga secuencial para reproducir las películas listadas en el sitio web yts.to (antes yify-torrent.com y yts.re), también conocido como YIFY (aunque se pueden añadir otros trackers y utilizarse de forma manual).

Tras su lanzamiento, Popcorn Time llamo rápidamente la atención de varios medios de comunicación, llegándose a comparar con la aplicación de Netflix, debido a su facilidad de uso. Tras este aumento de popularidad, el programa fue repentinamente eliminado por sus desarrolladores el 14 de marzo 2014, debido a la presión de la MPAA. Desde entonces, Popcorn Time se ha separado en varios equipos de desarrollo para mantener el programa y ofrecer nuevas características. Una de estas divisiones está disponible en <https://popcorn-time.io/>. Las plataformas soportadas son Mac, Windows, Linux (32 y 64 bits) y Android.

Gracias a su moderna y sencilla interfaz de usuario, la aplicación se hizo muy popular en todo el mundo. Las nuevas versiones de Popcorn Time pueden descargar y reproducir contenido facilitado por otros trackers, además utilizar los reproductores multimedia instalados en el sistema para reproducir el contenido. También puedes

Popcorn Time no sólo tiene un gracioso logotipo, sino que es una revolucionaria forma de conseguir contenido multimedia



configurar la aplicación para guardar los archivos eliminados y verlos más tarde, incluso con otros reproductores multimedia.

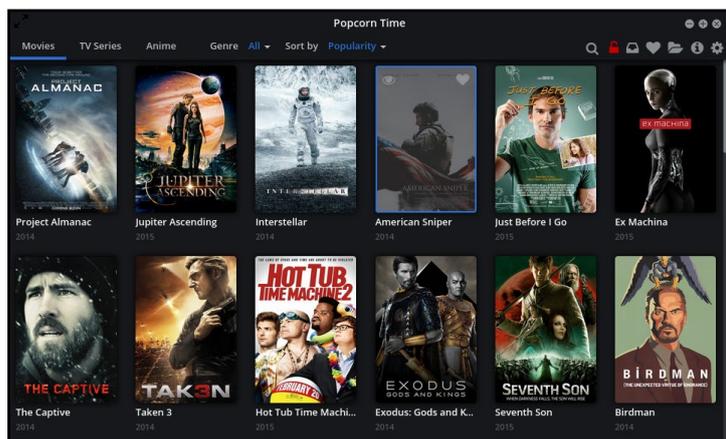
Tenía en mi poder un ODROID U3 cuando descubrí esta aplicación, lo primero que hice fue comprobar si estaba disponible para las plataformas armhf. En los foros ODROID y Popcorn Time sin embargo, descubrí que Popcorn Time no es compatible con dispositivos Linux armhf. Tras investigar un poco por Internet, averigüé que en teoría no había razón por la que esta aplicación no se ejecutara en mi ODROID-U3. Todo lo que necesitaba era encontrar los binarios de la aplicación nw.js (nodo-webkit) para armhf. Esta tarea representaba un gran reto porque ninguno de los binarios que encontré era capaz de ejecutar hardware acelerado por WebGL (en otras palabras usar OpenGL ES 2.0), aplicaciones demo, ni decodificar archivos de vídeo y audio. Así

que traté de desarrollar mis propios binarios basándome en los tutoriales que encontré en diferentes foros, pero con cada intento fracasaba porque algunas dependencias no eran las adecuadas, o aparecía algún error durante el proceso de compilación.

NW.js es una aplicación en tiempo de ejecución basada en Chromium y node.js. Puedes escribir aplicaciones nativas en HTML y JavaScript con NW.js. También permite activar módulos Node.js directamente desde el DOM y hace posible una nueva forma de escribir aplicaciones con todas las tecnologías Web. Fue creada en el Intel Open Source Technology Center.

Reproducción de un vídeo usando HTML 5 con NW.js





Portada de Popcorn Time

Puesto que NW.js está basado en Chromium y mi navegador Chromium es capaz de ejecutar WebGL acelerado por hardware, pensé que la manera en el que el entorno de trabajo utiliza OpenGL ES 2.0 en lugar de OpenGL 2.0 en NW.js sería similar a Chromium. Así que ejecute aplicaciones demo con el flag `--usegl=egl` y apareció magia. De repente, mi aplicación empezó a ejecutarse con aceleración hardware. Era capaz de ejecutar algunos juegos desarrollados con WebGL en mi ODROID.

Aunque mi entorno de trabajo se ejecuta en modo de aceleración por hardware, no era capaz de decodificar archivos de vídeo y audio así que continué investigando. Uno de los desarrolladores de Popcorn Time señaló que el problema estaba en el archivo `libffmpegsumo.so`, el cual no tenía implementados la mayoría de los códecs. Puesto que mi navegador

Popcorn Time reproduciendo un vídeo



Chromium era capaz de decodificar un montón de vídeos, intente usar el archivo `libffmpegsumo.so` localizado en el directorio de instalación de Chromium. Además de sustituir este archivo también

tuve que reemplazar el fichero `icudtl.dat` en el archivo `node-web-kit`.

Esta vez el entorno de trabajo `node-webkit` fue capaz de reproducir vídeos HTML 5 como muestra la imagen de abajo:

Llegado a este punto, contaba con el entorno de trabajo necesario para ejecutar la aplicación Popcorn Time con aceleración por hardware, que era capaz de decodificar algunos archivos de vídeo y audio. Todo lo que tenía que hacer ahora era exportar el proyecto a la plataforma `armhf`. Tras descargar el proyecto desde los repositorios Git intente compilarlo, pero la compilación falló debido a la arquitectura desconocida. Para hacerlo funcionar, tuve que añadir soporte `arm` al proyecto de escritorio Popcorn Time y a su proyecto de dependencia `Nodewebkit-builder`. Para hacer esto, afortunadamente, tan sólo tenía que modificar dos archivos de mi proyecto Git.

El resultado fue una versión funcional de Popcorn Time para mi ODROID-U3. Debido a la falta de codecs para Chromium, algunos vídeos todavía no se podían decodificar, o la decodificación era lenta en especial con los vídeos Full HD. Espero que los drivers actualizados para la GPU Malí resuel-

van estos problemas y que las próximas versiones de Chromium permitan más codecs. Además, descubrí que cuando intentas abrir los archivos torrent de otros trackers no funciona correctamente, aunque estoy trabajando en este problema. Realmente espero que este problema quede resuelto antes de que leas este artículo.

Los aficionados pueden encontrar dos proyectos en mi repositorio Git. Uno de los proyectos contiene los archivos y tutorial necesarios para la instalación de la aplicación, y el otro contiene un tutorial para la compilación del proyecto.

Estos repositorios están disponibles en las siguientes direcciones:

```
- https://git.popcorn-time.io/laslaul/popcorn-time-\
installation-guide-armv7
- https://git.popcorn-time.io/\
laslaul/popcorn-time-\
building-guide-armv7
```

Esta compilación solo la probé en un ODROID-U3 ejecutando Ubuntu 14.04.2, así que apreciaría las opiniones y comentarios de los usuarios, sobre todo de aquellos que ejecutan la aplicación en un dispositivo o sistema operativo diferente.

Antes de descargar e instalar Popcorn Time debes comprobar la legislación en propiedad intelectual de tu país, o asegurarte de que el contenido que subes o descargas es gratuito y legal. Ten en cuenta que Popcorn Time suele ser conocido a menudo como la versión pirata de Netflix.

Referencias

Wikipedia, Popcorn Time, http://en.wikipedia.org/wiki/Popcorn_Time (2015. may)
 GitHub, Nw.js project, <https://github.com/nwjs/nw.js/> (2015. may)

CONOCIENDO UN ODROIDIAN

CHRIS MCMURROUGH, EXPERTO EN ROBOTICA

editado por Rob Roy

Por favor, hablemos un poco sobre ti.

Soy un ingeniero de percepción robótica, profesor universitario, y creador de Texas. He trabajado con un gran número de plataformas robóticas (aéreas, terrestres, submarinas, acuáticas e industriales) y con sistemas integrados (ODROID, Raspberry Pi, Intel Atom y microcontroladores). La mayor parte de mi experiencia procede de la investigación y el desarrollo, tanto en el mundo académico como industrial. Siempre ando interesado en el cambiante mundo de la robótica y de los modelos informáticos necesarios para que hagan un trabajo de un modo útil.

En estos momentos estoy centrado en la formación, sobre todo a nivel universitario para estudiantes que están a punto de iniciar su carrera profesional. Disfruto enseñando las técnicas multidisciplinarias más demandadas que no se suelen aprender necesariamente en clase. También participo en programas de divulgación para que los niños empiecen a interesarse en la ingeniería a temprana edad. La mejor forma que conozco de hacer esto es mostrando a los estudiantes un robot que haga algo interesante, y luego explicarles todas las cuestiones mecánicas que son necesarias para diseñar una plataforma robótica, los conocimientos en electricidad que se necesita para establecer la comunicación y el control, y las habilidades informáticas que hacen falta para añadir la inteligencia y la toma de decisiones..

¿Cómo fueron tus inicios con los ordenadores?

Crecí con Oregon Trail. La mayoría de mis amigos murieron de disentería.

En serio, todos los ordenadores que he ido teniendo siempre han sido máquinas de segunda mano. Tenía que ir ac-



Aunque Chris aún no se dedica a crear armaduras de Iron Man, sin duda es la persona que más cerca está de hacerlo.

tualizando los componentes para lograr que funcionasen, y siempre andaba con problemas que tenía que ir resolviendo por mí mismo. El primer equipo “nuevo” que compré fue un PC con Windows ME, que es el peor SO que jamás se ha hecho. También tenía un defecto de diseño que provocaba que se sobrecalentase y se bloqueara de vez en cuando, así que me vi obligado a poner en práctica mis propios trucos y hacer ajustes para ejecutar el Starcraft original. Básicamente, he aprendido mucho sobre los ordenadores simplemente jugando al Starcraft.

¿Qué te atrajo de la plataforma ODROID?

Hago mucho reconocimiento de imágenes por ordenador y trabajos con robótica. Básicamente, necesito Linux y librerías informáticas para hacer mi trabajo. Las plataformas ODROID te proporcionan una serie de ventajas en comparación con el resto. Todas las placas de Linux integradas tienen problemas cuando intentas exportar el código desa-

rollado en un ordenador de sobremesa o portátil, pero he sido capaz de solventar todos los problemas con lo que me he ido encontrando gracias a la gran cantidad de información disponible en los foros.

¿Cuál es tu ODROID favorito?

Mi favorito insuperable es el ODROID-U3. Es pequeño, rápido, y ofrece un extra en comparación con los que están actualmente disponibles en el mercado. También siento predilección con el X2, porque tuve una muy buena experiencia con él siendo mi primera placa ODROID.

¿Cómo llegaste a ser un experto en robótica?

El primer robot que desarrollé fue un sencillo solucionador de laberintos utilizando el kit original de Lego Mindstorms en mi clase de ciencias de la computación del instituto. Esta fue mi primera experiencia con la programación integrada, el uso de sensores y el control de motores. Después de esto,



Un poco de turismo por el mundo entero, aquí está Chris en el Partenón en Grecia

me enganché. Cuando empecé mi diplomatura, me uní a un equipo de robótica de la universidad y desarrollé un sencillo robot de campo con estudiantes de otras carreras de ingeniería. Disfrute mucho con la experiencia interdisciplinaria y competitiva que te proporciona estas competiciones, y hasta ahora creo que la forma “ más correcta” de aprender robótica es participar y desarrollar.

Hacia el final de mis estudios universitarios, empecé a trabajar en un laboratorio de investigación de sistemas de control en prácticas. Adquirí una valiosa experiencia sobre teorías de control y sistemas electrónicos y decidí quedarme y hacer la licenciatura. En el siguiente par de años, terminé el desarrollo de un micro vehículo aéreo como prueba de concepto para una ley de control teórico que alguna gente investigaba. Aquí es donde adquirí habilidades de ingeniería eléctrica y mecánica, que junto con la ingeniería informática, completan mis conocimientos en robótica.

Llegado este momento, disfrutaba mucho con lo que hacía y decidí que otros 4 años con estas cosas no sería mala idea. Consideré apropiado hacer un doctorado tras terminar mi Licenciatura y continué trabajando en plataformas robóticas. Por ese tiempo, a un familiar cercano le fue diagnosticado esclerosis lateral amiotrófica, una enfermedad neu-

ro degenerativa que hace que los pacientes sean incapaces de moverse o hablar. Le dieron un equipo de seguimiento ocular, que en ese momento necesitaba muchas intervenciones para calibrarlo y mantenerlo en funcionamiento. He aprendido mucho con las limitaciones del dispositivo, y empecé a desarrollar mis propios prototipos y algoritmos con la esperanza de reducir los costes y mejorar su rendimiento y facilidad de uso. Hacia el final de mi doctorado, me encontraba experimentando con el control de robots terrestres (sillas de ruedas y pequeñas UGV) a través de la mirada, fue cuando me compré un X2.

Tras graduarse, desarrollé un software de percepción para los robots que manipulasen material industrial. Luego acepté un puesto en la facultad de mi universidad, y ahora me dedico a la enseñanza. Me encanta mi trabajo, y me da motivos para seguir aprendiendo nuevas técnicas y experimentar con nuevos productos comerciales. Después de todo, ningún estudiante quiere aprender el funcionamiento interno de un sistema integrado que fue popular hace 10 años.

¿Qué te motivó a crear la popular imagen de edición Robótica en los foros ODROID?

La mayor parte de mi trabajo requiere Linux, OpenCV, Point Cloud Library y ROS. Cuando compré el X2,

me di cuenta que configurar mi entorno base era más complicado de lo que yo estaba acostumbrado. Investigue bastante y logre encontrar soluciones a cada uno de los problemas de compilación e instalación que me iba encontrando en el camino, y al final disponía de una buena y estable imagen con toda la configuración que deseaba. Hice una backup de la tarjeta SD “por si acaso”, y un día decidí publicarla en los foros con el fin de darle las gracias a la comunidad por haberme ayudado con sus consejos e ideas. Empecé a recibir gran cantidad de respuestas en el hilo original, y seguí respondiendo a las preguntas como lo haría hecho la comunidad conmigo. Poco después de publicar la imagen X2, compré un U2 y modifiqué la misma imagen, y la publiqué como lo hice con la anterior. La gente de Hardkernel continuaba haciendo productos nuevos e innovadores, y en poco tiempo recibía solicitudes para trasladar la imagen a otras placas.

Me sorprendió bastante que mis imágenes crecieran en popularidad de la forma en que lo hicieron. Hardkernel me envió algunas cosas gratis como parte de su regalo mensual, y todo empezó desde allí. Hardkernel y la comunidad ODROID en general han sido de gran ayuda para mi trabajo, e intento liberar nuevas imágenes como versiones oficiales de Ubuntu y nuevos productos que están accesibles y puestos a disposición de todos.



Como ha viajado por todo el mundo, aquí podemos encontrar a nuestro experto en robótica en Shanghai

¿Qué aficiones e intereses tienes aparte de los ordenadores?

Soy un mecánico amateur y aficionado a la creación de prototipos. Recientemente, he estado trabajando en varias máquinas CNC (impresoras 3D, routers, fresadoras, tornos). Estoy muy contento con las creaciones que he realizado hasta el momento, y teniendo un garaje lleno de equipos de fabricación se están volviendo más y más realistas cada día. También estoy experimentando con métodos de fabricación tecnológicamente menos avanzados, como la madera y fundición de metales. Básicamente, lo único que hago es hacer cosas.

¿Está involucrado en otros proyectos informáticos a parte de ODROID?

He hecho algunas contribuciones mínimas (un par de correcciones de errores menores) a Point Cloud Library. Como parte de mi trabajo como profesor universitario, asigno y oriento nuestros proyectos de alto nivel. Estos cambian cada semestre, y tienden a impulsar proyectos basados en sistemas integrados y de robótica. Pongo mucho énfasis en las habilidades prácticas en mis clases y me paso la mitad del tiempo haciendo demostraciones y dando charlas técnicas sobre nuevos y excitantes aparatos (ODROIDS, impresoras 3D, visión por ordenador). Me gusta ver los conceptos

que aprenden mis alumnos implementando sus proyectos finales, y continuo haciendo y probando sistemas en mi tiempo libre y así siempre tengo algo nuevo que enseñar.

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

Me gustaría ver algunos progresos en la GPU, en particu-

lar con Linux. Una de mis inquietudes es la programación en paralelo con las GPU multinúcleo, mi única experiencia real en esta área es el kit de herramientas NVIDIA CUDA. Hardkernel ha hecho buenos progresos en esta área, y estoy deseando ver cómo se desarrolla esto.

¿Qué consejo le darías a alguien que quiere aprender más sobre programación?

Comenzar con un lenguaje que tenga muchos ejemplos online, y tal vez algo de herramientas GUI. La mayoría de las personas que se inician en la programación quieren hacer una aplicación GUI como primer programa tras la aplicación por consola "Hello World" por excelencia, pero esto puede ser complicado y engorroso para alguien que no está familiarizado con los IDEs, makefiles, compiladores, etc. Me gusta mucho el entorno de trabajo .NET con C#, por lo que hacer una GUI y la mayoría de las tareas en un entorno de Windows es bastante simple. Python es un buen lenguaje de partida, pero creo que una buena formación en C++ es algo que separa a un profesional del resto.

Lo más importante es no preocuparse por romper algo. Cuando estás empezando con la programación o incluso con el desarrollando integrado, es fácil romper cosas. A veces se rompe el código, a veces se rompe el hardware, pero si no rompes algo, es que no estás aprendiendo nada.



¡ODROID Magazine está ahora en Reddit!



**ODROID Talk
Subreddit**

<http://www.reddit.com/r/odroid>

