

# ODROID

## Magazine

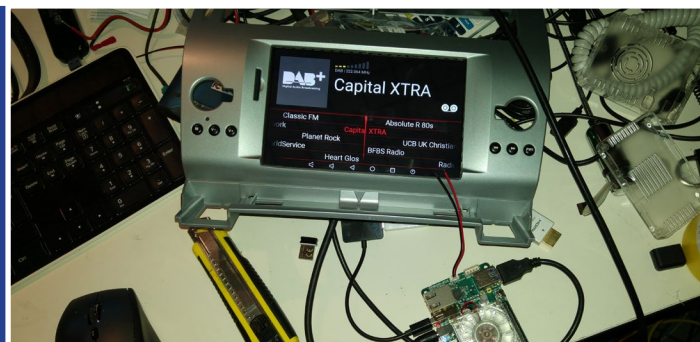
Año Tres  
Num. #29  
May 2016

# LA INTERNET PROFUNDA

- CONFIGURA UN REPETIDOR TOR EN TU ODROID
- CONOCE LA INYECCION DE CODIGO INALAMBRICA

• Reproduce  
tu música con  
Cherry Music

• Crea tu propio  
smart car con  
un ODROID-XU4



# Qué defendemos...

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

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

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

Simple, moderno y único.

De modo que tienes a tu alcance lo mejor.



## HARDKERNEL



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

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

**Teléfono & Fax**

telf : +49 (0) 8403 / 920-920

email : [service@pollin.de](mailto:service@pollin.de)

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







**N**uestro tema de este mes es la seguridad en Internet y el anonimato. La Web profunda es una gran parte de Internet que no está indexada por los motores de búsqueda tradicionales, y requiere de un navegador especial llamado Tor para acceder a ella. Tor ofrece una experiencia de navegación segura, dando a los usuarios acceso a repetidores especiales que proporcionan una entrada y salida seguras a la Web profunda. Tal y como describe David en su artículo, un **ODROID** se puede utilizar para configurar un repetidor **TOR** de bajo coste que te proporcionará servicios de navegación anónimos. Puesto que la información financiera se transfiere a través de la World Wide Web, es importante mantener una red local segura. Una técnica muy común para monitorizar maliciosamente el tráfico de un router es la inyección inalámbrica, Adrian nos detalla las formas de proteger una red usando Kali Linux, una potente suite de pruebas de penetración. Tobias analiza Atari Jaguar, Nanik nos presenta Android Support Library para mantener la compatibilidad con versiones antiguas de Android, y Marian nos cuenta cómo monitorizar el sueño de un bebé. Nuestros proyectos de bricolaje incluyen la instalación de Cherry Music, un Car PC integrado y la producción de realidad aumentada con una oCAM.

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. Para información sobre cómo enviar artículos, contacta con [odroidmagazine@gmail.com](mailto:odroidmagazine@gmail.com), o visita <http://bit.ly/lyplmXs>. 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**



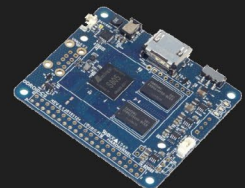
Hundreds of products available online for the professional developer and hobbyist alike



**ODROID-XU4**



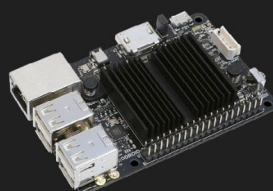
**ODROID-C1+**



**ODROID-C0**



**OWEN ROBOT KIT**



**ODROID-C2**



**VU7 TABLET KIT**





## **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 clientes 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>.

---



## **Bruno Doiche, Editor Artístico Senior**

¿Qué es lo que está haciendo últimamente nuestro amigo? Organizar todas sus colecciones de música que ha recibido en los últimos 20 años. Pensamos que es sobre todo un intento de gusto frustrado, especialmente cuando vemos a Bruno pasar de Death metal a Depeche Mode para niños. Pero, ¿Qué podemos hacer, verdad? ¡Al menos lo escucha con auriculares la mayoría de las veces!

Además, es un poco mamón con David, que le da una paliza en el juego de cartas utilizando su propio servidor XMAGE con ODROID. Pocos usuarios en el mismo servidor significa que Bruno pierde sus partidas más rápido que nunca.

---



## **Manuel Adamuz, Editor Español**

Tengo 31 años y vivo en Sevilla, España, aunque 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.

---



## **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 <http://www.nicolescott.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. Continuo muy enamorado de muchos de los 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.

---



## **Andrew Ruggeri, Editor Adjunto**

Soy un ingeniero de sistemas Biomédicos anclado en Nueva Inglaterra que actualmente trabaja en la industria aeroespacial. Un microcontrolador 68HC11 de 8 bits y el código ensamblador son todo lo que me interesa de los sistemas embebidos. Hoy en día, la mayoría de los proyectos en los que trabajo están en lenguajes C y C ++, o en lenguajes de alto nivel como C# y Java. Para muchos proyectos, utilizo placas ODROID, pero aún sigo intentando utilizar los controladores de 8 bits cada vez que puedo (soy un fan de ATMEL). Aparte de la electrónica, soy un amante de la fotografía analógica y desarrollo la película friki con la que disfruto intentando hablar en idiomas extranjeros.

---



## **Venkat Bommakanti, Editor Adjunto**

Soy un apasionado de los ordenadores desde la bahía de San Francisco en California. Procuo incorporar muchos de mis intereses en proyectos con ordenadores de placa reducida, tales como pequeños modificaciones de hardware, carpintería, reutilización de materiales, desarrollo de software y creación de grabaciones musicales de aficionados. Me encanta aprender continuamente cosas nuevas, y trato de compartir mi alegría y entusiasmo con la comunidad.

---



## **Josh Sherman, Editor Adjunto**

Soy de la zona de Nueva York, y ofrezco mi tiempo como escritor y editor para ODROID Magazine. Suelo experimentar con los ordenadores de todas las formas y tamaños: haciendo trizas las tablets, convirtiendo Raspberry Pi en PlayStations y experimentado con los ODROIDs y otros SoCs. Me encanta trabajar con los elementos básicos y así poder aprender más, y disfrutar enseñando a otros escribiendo historias y guías sobre Linux, ARM y otros proyectos experimentales divertidos.





**REPETIDOR TOR - 6**



**PRODUCCION DE ODROID - 9**



**CHERRY MUSIC - 10**



**INYECCION INALAMBRICA - 12**



**REALIDAD AUMENTADA - 19**



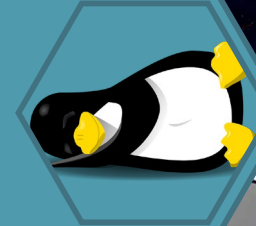
**JUEGOS LINUX: ATARI JAGUAR - 23**



**DESARROLLO ANDROID: ANDROID SUPPORT LIBRARY - 28**



**JUEGOS LINUX: BATTLE FOR THE SOLAR SYSTEM - 31**



**BABY NAP (PROGRAMA DE ACTIVIDAD NOCTURNA) - 32**



**CAR PC - 36**



**CONOCIENDO A UN ODROIDIAN - 40**



# CONVIERTE TU ODROID EN UN REPETIDOR TOR

## PROTEGIENDO A LA VEZ LA LIBERTAD DE UN ODROID

por David Gabriel



**T**or es un software gratuito que permite acceder a una red abierta muy útil en comunicaciones anónimas. El nombre deriva de las siglas del proyecto original, The Onion Router. Protege tu privacidad redireccionando el tráfico de Internet a una red de miles de repetidores, y evita las utilidades que supervisan redes y analizan el tráfico recopilando datos mientras navegas. En otras palabras, te hace “invisible”, de modo que los sitios web no saben tu ubicación por la dirección IP o por el proveedor de servicios de Internet (ISP). La gente que monitoriza tu red no será capaz de ver las páginas web y los recursos a los que accedes.

Todas las comunicaciones dentro de Tor están cifradas. Cuando se envían los datos, éstos son cifrados varias veces en la capa de aplicación, y anidados como si fueran capas de cebolla. Las rutas de los datos incluyen repetidores escogidos al azar. Cada repetidor descifra una capa de codificación revelando únicamente el siguiente repetidor y trasladando la información restante a éste. El proceso continúa hasta que el último repetidor descifra los datos originales y los envía al destinatario sin revelar la dirección IP de origen.

La desventaja de usar Tor es que tu conexión de Internet se volverá más lenta de lo normal, debido a las continuas fases de cifrado y descifrado y al paso por múltiples repetidores. La velocidad de transferencia de información sería perceptiblemente más baja.

### Instalación

En primer lugar, asegúrate de que el sistema está actuali-

**Aunque los servicios de Internet de las grandes empresas son muy buenos, en algunos se cuelan intrusos impertinentes, Tor permite crear un repetidor de acceso libre de miradas indiscretas.**

zado utilizando los siguientes comandos:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Luego, instala la aplicación Tor y sus dependencias con el siguiente comando:

```
$ sudo apt-get install tor
```

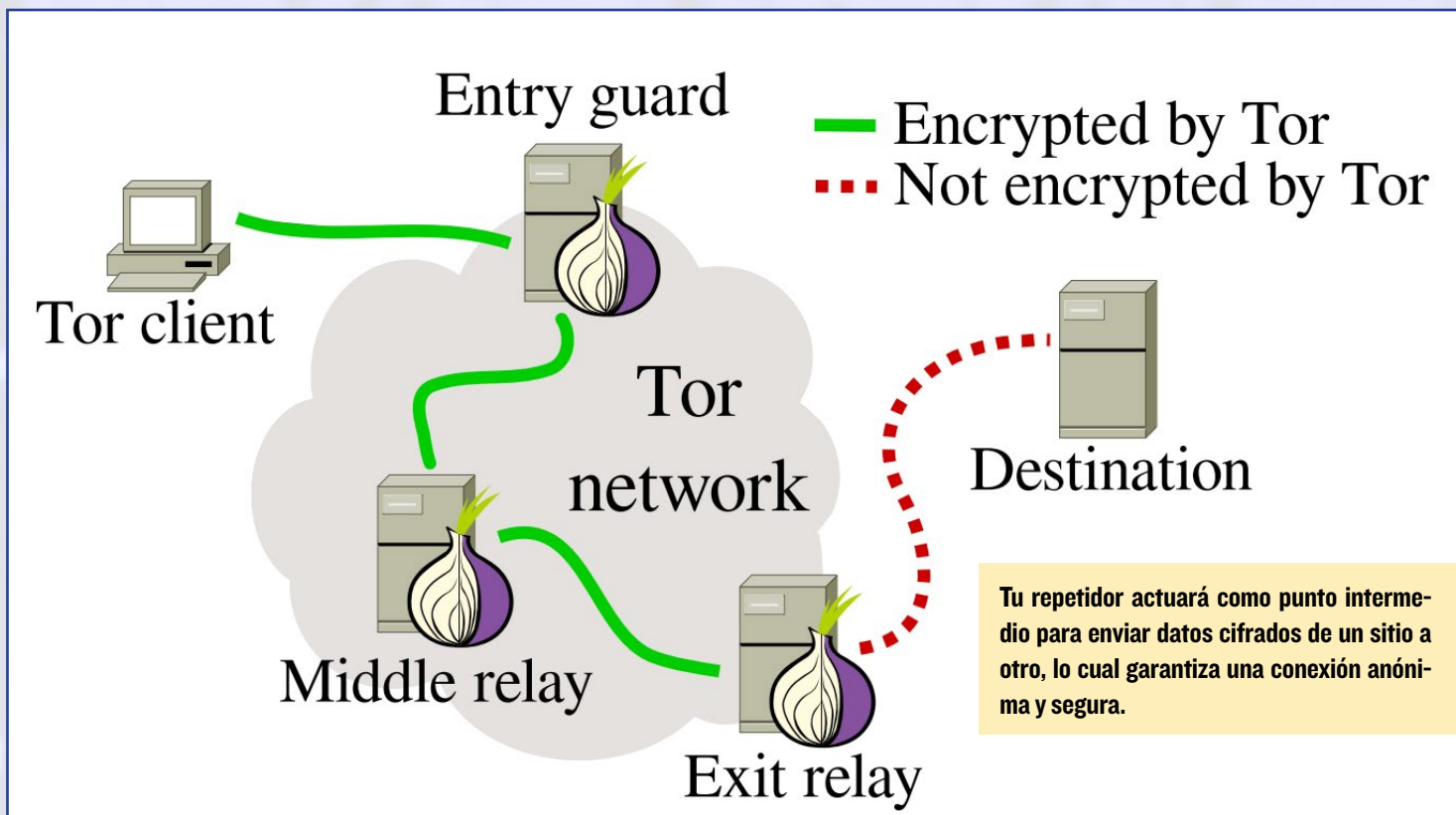
Opcionalmente, también puedes instalar Arm (acrónimo de: anonymizing relay monitor), que es una aplicación para controlar y configurar Tor. Su funcionamiento es similar a una utilidad Linux llamada top, se puede instalar con el siguiente comando:

```
$ sudo apt-get install tor-arm
```

### Configuración

Tor se puede personalizar modificando su archivo de configuración. Puedes utilizar tu editor de texto favorito para editar el archivo `/etc/tor/torrc` y añadir las opciones comentadas (con #) que se enumeran a continuación:





```
Log notice file /var/log/tor/notices.log # Log file destination
RunAsDaemon 1 # Start process in background as a daemon
ORPort 9001 # Port to be used by incoming connections
DirPort 9030 # Port to be used by directory connections
ExitPolicy reject *: * # Implies that your relay will be used for
                        # relaying traffic inside the Tor network, but
                        # not for connections to external websites or
                        # other services
Nickname odroid-tor-relay # Can be anything you like, so people
                           # don't have to refer to your relay by key
RelayBandwidthRate 100 KB # Throttle traffic to 100KB/s (800Kbps)
RelayBandwidthBurst 200 KB # But allow bursts up to 200KB/s (1600Kbps)
```

Si instalas la aplicación Arm opcional, debes añadir las siguientes líneas de configuración en el archivo que hemos mencionando anteriormente:

```
ControlPort 9051 # Port to be used by controller applica-
                 # tions.
CookieAuthentication 1 # Authentication method to be used by the
                       # controller application
DisableDebuggerAttachment 0 # Required by Arm application to be able to
                              # use
                              # commands like netstat to monitor the network
                              # traffic
```

A continuación, reinicia Tor para que la nueva configuración tenga efecto, para ello usa el siguiente comando:



```
$ sudo service tor restart
```

Si todo va bien, deberías ver una entrada en `/varlog/tor/log` como esta:

```
Jan 15 11:38:53.000 [notice] Tor has successfully opened a circuit.
Looks like client functionality is working.
```

Ten en cuenta que si tu red está protegida por un cortafuegos, tendrás que configurarlo para que permita peticiones entrantes en los puertos, 9030 (para el servicio de directorios) y 9001 (para el funcionamiento del repetidor). Puede que tengas que recurrir a la Guía del usuario de tu cortafuegos para configurar esta opción. Si has instalado Arm, puedes iniciarlo con el comando:

```
$ sudo arm
```

Aunque hay muchas opciones que se pueden configurar, las más interesantes están relacionadas con los gráficos que puedes generar para monitorizar todo el tráfico que pasa a través de tu repetidor. Consulta la ayuda de la aplicación Arm, para obtener más información sobre cómo aprovechar todo su potencial.

Por defecto, Tor también soporta el protocolo Socket Secure (SOCKS), sobre el puerto 9050. Puedes configurar tu navegador para que sea un cliente Tor y redirija todas las conexiones a través de repetidores Tor protegiendo así tu privacidad y manteniendo el anonimato. En Firefox, por ejemplo, puedes ir a Opciones > Avanzado > Red > Configuración, cambia la configuración del proxy a manual y añade 127.0.0.1 con el puerto 9050 a la línea de “Servidor SOCKS” y pincha en “Aceptar” para confirmar.

Para comprobar tu configuración, visita el sitio web del proyecto Tor <http://bit.ly/1oh1f82> usando un navegador. Te darás cuenta que la dirección IP pública que aparece en esta página será diferente a tu dirección IP real. Este es el nodo de salida de tu petición, el cual garantiza que no se rastrea tu ubicación o se muestra información personal. Ten en cuenta que los datos sólo se cifran cuando estos pasan por la red Tor. Los datos se enviarán tal como están, así que cualquier cosa que haya sido encriptada desde el principio seguirá estando así, tras dejar el nodo de salida.

Si deseas desactivar esta función SOCKS y mantener tu ODROID sólo como repetidor, agrega la siguiente línea al archivo `/etc/tor/torrc` y reinicia el servicio Tor:

```
SocksPort 0 # Disable torsocks
```

El cliente Tor también puede utilizarse en otros sistemas operativos. La configuración puede variar ligeramente dependiendo del sistema operativo y el navegador, pero las opciones que acabamos de ver es un buen punto de partida para empezar.

## References

<a href="http://bit.ly/1cqlVa3">http://bit.ly/1cqlVa3</a>	<a href="http://bit.ly/1PvVIqy">http://bit.ly/1PvVIqy</a>
<a href="http://bit.ly/1U9oXqa">http://bit.ly/1U9oXqa</a>	<a href="http://bit.ly/1U9pgkM">http://bit.ly/1U9pgkM</a>
<a href="http://bit.ly/19QYR47">http://bit.ly/19QYR47</a>	<a href="http://bit.ly/1MOsQPE">http://bit.ly/1MOsQPE</a>
<a href="http://bit.ly/1nBUETC">http://bit.ly/1nBUETC</a>	



# LA PRODUCCION DE ODROID

## UNA RETROSPECTIVA DE LOS PRIMEROS AÑOS DE HARDKERNEL

editado por Rob Roy

**P**or casualidad te ha preguntado alguna vez cómo se hacen los ODROIDS. Aunque el proceso ha ido cambiando con los años, aquí tienes un interesante análisis de cómo era la primera línea de productos de Hardkernel, el ODROID-X y el ODROID-Q, fabricados en julio de 2012.

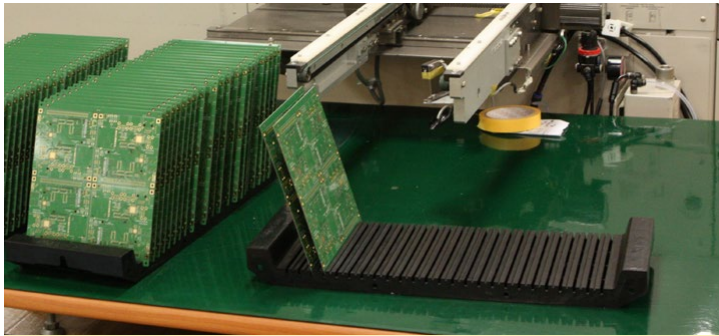


Figura 1 - El primer paso del proceso es una máquina llamada "Solder cream printer"

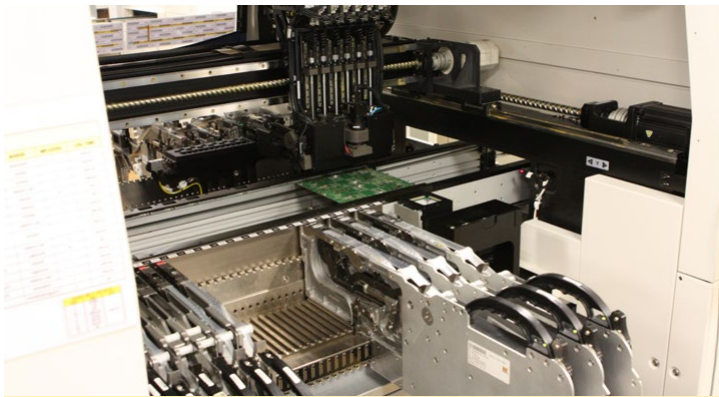


Figura 2 - La maquina Surface Mount Technology (SMT) monta los dispositivos sobre la PCB

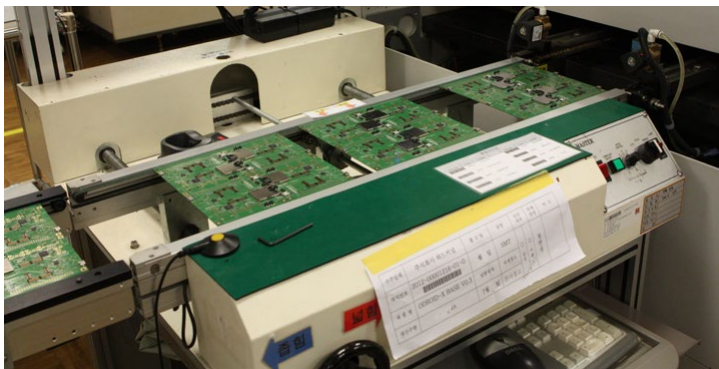


Figura 3 - Las placas se introducen en una maquina de soldadura de reflujo.

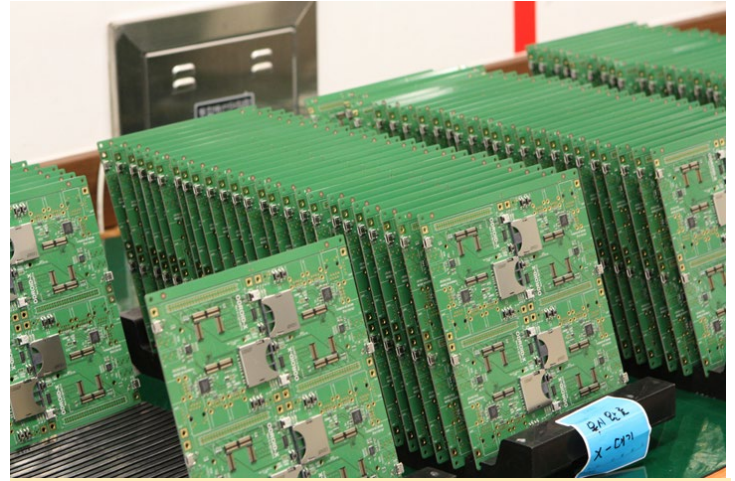


Figura 4 - La soldadura SMT está terminada, y las placas están listas para someterlas al proceso de inspección

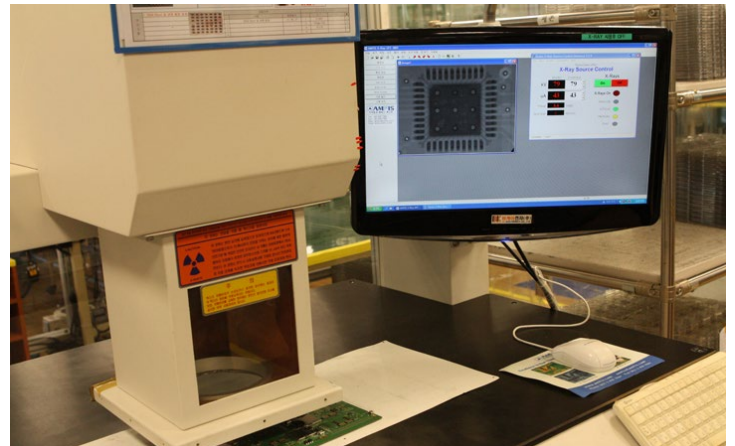


Figura 5 - Inspección por rayos X para comprobar la calidad de la soldadura



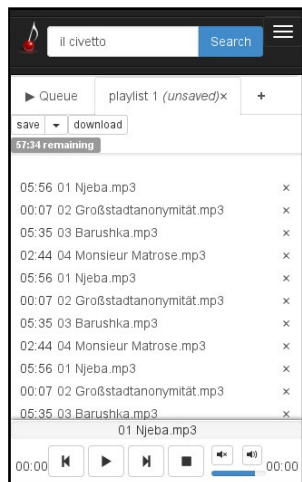
Figura 6 - Líneas de fabricación, donde las PCBs son trasladadas a otra máquina de soldadura para añadirle diferentes conectores tras el proceso de análisis, y pasar por todas las pruebas de funcionalidad.

# CHERRYMUSIC

## TU PROPIO SISTEMA STREAMING DE MUSICA PRIVADO

por @synportack24

CherryMusic es un servidor de streaming de música basado en CherryPy y jPlayer. Reproduce archivos de música almacenados en tu PC, teléfono inteligente, tablet, ODROID o cualquier dispositivo que tenga instalado un navegador compatible con HTML 5. CherryMusic está basado en AJAX, de modo que no requiere actualizaciones de página, haciendo que sea muy rápido. La interfaz de usuario basada en navegador funciona en cualquiera dispositivo móvil o de escritorio. Con multitud de funciones y características permite crear listas de reproducción, navegar por la música, configurar múltiples cuentas de usuario y mucho más.



### Vista web móvil

Un sistema basado en CherryMusic se puede combinar con cualquier sistema HTPC que encuentres. Es ligero y se puede ejecutar fácilmente en segundo plano, sin sobrecargar los procesadores. Para la configuración que se describe en este artículo, he utilizado un ODROID-XU4 como centro multimedia y un disco duro de gran capacidad para almacenar todos los archivos de música. Me permite transmitir música a cualquier dispositivo de la casa, mientras que otros están libres para visualizar simultáneamente vídeo por streaming utilizando Kodi.

## Requisitos previos

Primero, debes reunir las siguientes dependencias de software:

**Python (version 2.6 o 3.2)**  
**git**  
**screen**

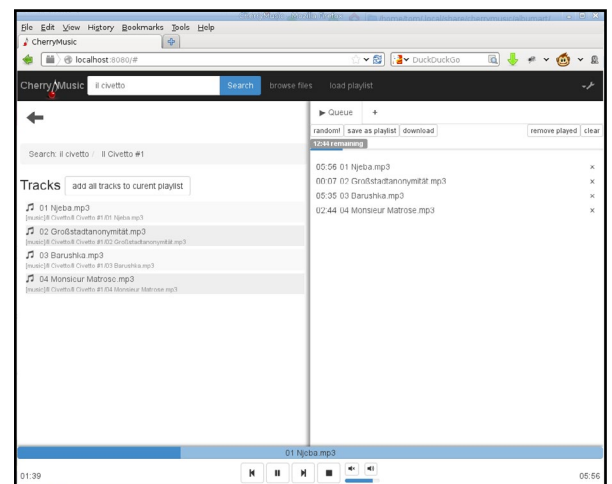
Las pueden instalar con el siguiente comando:

```
$ sudo apt-get install python git screen
```

## Instalación

La instalación es simple y fácil. En primer lugar, accede al terminal (ssh o local) del dispositivo que contiene los archivos de música. Luego, instala CherryMusic usando los siguientes comandos:

```
$ cd ~
$ git clone -b master \
```



### Página web de escritorio



```
https://github.com/devsnd/cherrymusic.git
```

El cual se instala en un subdirectorio llamado `~/cherrymusic`.

La mejor manera de iniciar CherryMusic es utilizar la utilidad `screen`, que te permite ejecutar CherryMusic en segundo plano. Esto se puede conseguir con el siguiente comando:

```
$ screen -mS cherrymusic ~/cherrymusic/cherrymusic --setup --port 8080
```

El entorno de trabajo CherryPy incluye un servidor web, que se utiliza para ejecutar la interfaz de usuario basada en navegador. Si el puerto 8080 ya es utilizado por otra aplicación, puedes seleccionar otro puerto cambiándolo en el anterior comando.

Puede que se te solicite descargar `cherrypy`, así que presiona “Y”. La instalación finalizará cuando aparezca un `prompt`. Tu pantalla debería ser similar a la siguiente Figura, momento en el que debes pulsar `Ctrl + A + D` para asegurarte de que CherryMusic se está ejecutando en segundo plano.

Abre un navegador, ya sea en el dispositivo que tienes instalado CherryMusic, o en un dispositivo conectado a la red e introduce la siguiente dirección. Ten en cuenta que es necesario utilizar el puerto HTTP correcto si no es el 8080:

```
https://<hostname-of-cherrymusic-device>:8080
```

Ahora puedes configurar CherryMusic. El “Media base directory” es el directorio que contiene los archivos de música. El resto de ajustes se pueden dejar con los valores por defecto. Haz clic en “Save Configuration and start CherryMusic” para pasar a crear una cuenta de administrador.

¡Ya está! La próxima vez que te conectes, se te pedirá el nombre de usuario y contraseña. Tras iniciar sesión con éxito, podrás transmitir a cualquier dispositivo toda tu música.

Aquí tienes unos cuantos comandos para experimentar:

#### Cambiar a CherryMusic que se estaba ejecutando en segundo plano

```
screen -r cherrymusic
```

#### Detiene la ejecución de CherryMusic

```
kill cherrymusic
```

## Lecturas recomendadas

#### GitHub de CherryMusic

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

#### Página de CherryPy

<http://bit.ly/1NIW5t3>

#### Página de CherryMusic

<http://bit.ly/1r1eeTf>

#### Librería jPlayer

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



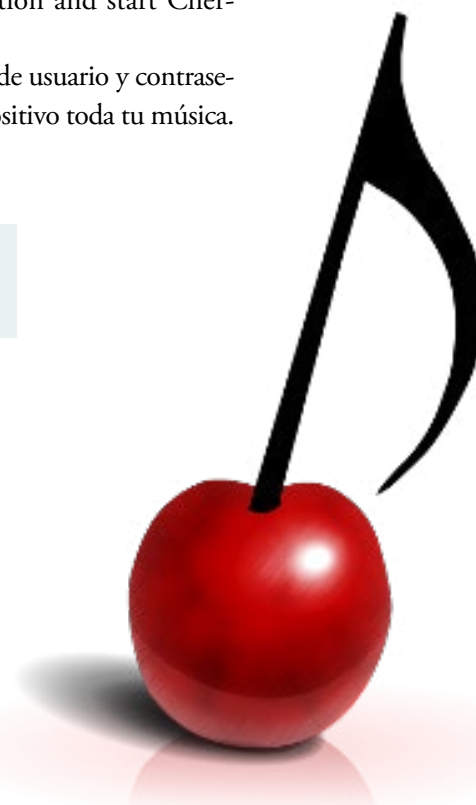
```
odroid@odroid: ~
File Edit View Search Terminal Help

CherryMusic needs the module "cherrypy" to run. You should install it
using the package manager of your OS. Alternatively cherrymusic can
download it for you and put it in the folder in which currently
CherryMusic resides.

Download cherrypy now? (y/n)y
Downloading cherrypy...
Extracting /tmp/cherrypy.tar.gz
Copying cherrypy module inside cherrymusic folder (/home/odroid/cherrymusic)...
Cleaning up temporary files...
Successfully installed cherrymusic dependencies! You can now start cherrymusic.
[160417-15:43] Migrating saved album art filenames...
[160417-15:43] Album art filename migration done.

Starting setup server on port 8080 ...
Open your browser and put the server IP:8080 in the address bar.
If you run the server locally, use: localhost:8080.
```

## Instalación



# INYECCION INALAMBRICA

## UNA MANERA PRACTICA DE CONOCER 802.11

por Adrian Popa

**E**n mi anterior artículo, aprendimos cómo configurar los diferentes módulos WiFi ODROID en modo monitor, lo cual nos permitía escuchar el tráfico inalámbrico mediante Kismet. En este artículo, probaremos la inyección de tráfico y analizaremos redes abiertas que utilizan diversos métodos de protección incluida la encriptación. Terminaremos con algunos ataques que no implican romper el cifrado de la red. Aunque este artículo es de carácter técnico, su objetivo es el de mostrarte cómo funcionan las redes inalámbricas. No está diseñado para convertirte en un “script kiddie” que no entienda lo que están haciendo. Si no has leído el artículo sobre Kismet de ODROID Magazine del mes pasado, disponible en <http://bit.ly/20YG7Yg>, hazlo ahora, ya que vamos a partir de esa experiencia previa. Como siempre, penetrar en la red de alguien sin su consentimiento está totalmente prohibido, de modo que sólo hacer esto con tus propias redes.

### Preparación

El término “inyección” se utiliza para indicar que se está generando tráfico inalámbrico, basado en paquetes especialmente diseñados que pasan por alto el tráfico regular de Internet del adaptador inalámbrico. Esto significa que un programa es capaz de generar paquetes con determinados campos y enviarlos al controlador a través de la interfaz de monitor, incluso los envía aunque no sean compatible con el protocolo utilizado.

Para hacer nuestra primera prueba de inyección, necesitaremos una interfaz de monitor y aireplay-ng, el cual cubrimos en el anterior artículo sobre Kismet. Opcionalmente, puedes especificar el canal en el que quieres operar cuando crees la interfaz mon0:

```
$ sudo airmon-ng start wlan0 6
$ sudo aireplay-ng -9 mon0
```

El programa inicialmente envía peticiones de sondeo por radiodifusión. Estas son las peticiones que solicitan a cualquier punto de acceso (PA) que está escuchando para que responda con una descripción de sí mismo. No todos los PA responderán a este tipo de solicitudes. Se creará una lista con todos los puntos de acceso que respondieron, ésta será utilizada en los siguientes pasos. Si un PA responde, aparece un mensaje en pantalla indicando que la tarjeta puede inyectar con éxito.



**Las pruebas de inyección pueden ayudarte a mantener tu red segura de intrusos desconocidos**

Al mismo tiempo, cualquier PA identificado a través de un paquete beacon también es añadido a la lista de puntos de acceso para ser procesado posteriormente. Si un PA específico es añadido por línea de comandos, BSSID y SSID, también se agrega a la lista. Después, para cada punto de acceso de la lista, se envían 30 peticiones de sondeo directas. Una petición de sondeo directa siempre será dirigida a un punto de acceso específico. El número de respuestas de sondeo recibidas más el porcentaje aparecen en pantalla, esto te indica si puedes comunicarte con el PA y a qué nivel.

```
adrianp@qp06:~$ sudo aireplay-ng -9 mon0
15:25:00 Trying broadcast probe requests...
15:25:00 Injection is working!
15:25:02 Found 9 APs

15:25:02 Trying directed probe requests...
15:25:02 72:55:9C:DF:98:80 - channel: 1 - 'Ro...381'
15:25:04 Ping (min/avg/max): 5.536ms/76.800ms/130.476ms Power: -38.80
15:25:04 30/30: 100%

15:25:04 72:55:9C:DF:98:81 - channel: 1 - 'Ro...381'
15:25:07 Ping (min/avg/max): 29.852ms/76.253ms/119.094ms Power: -38.73
15:25:07 30/30: 100%

15:25:07 72:55:9C:DF:98:82 - channel: 1 - 'i...b'
15:25:09 Ping (min/avg/max): 38.904ms/83.900ms/135.193ms Power: -38.73
15:25:09 30/30: 100%

15:25:09 3C:F8:08:43:86:34 - channel: 1 - 'H...'
15:25:11 Ping (min/avg/max): 15.031ms/70.436ms/136.017ms Power: -67.67
15:25:11 30/30: 100%
```

**Figura 1 - Una prueba de inyección en ejecución**

### Redes abiertas

Vamos a examinar con más detenimiento cómo funcionan las redes inalámbricas en condiciones normales, creando una red abierta y analizando su tráfico. En mi router, configure una red de prueba llamada “NASA-HQ-Guests”, puesto que “FBI-Surveillance-Van-3” ya estaba cogida.

Para monitorizar una red específica puedes utilizar airodump-ng o Kismet, ambas herramientas hacen el mismo trabajo. Para ver una lista de redes disponibles y sus clientes, ejecuta



el siguiente comando, se supone que tu interfaz de monitorización ya está activada:

```
$ sudo airodump-ng mon0
```

Debería ver una lista de redes con sus ESSID, nombre de la red, BSSID y las direcciones MAC del PA, junto con su potencia, tipo de cifrado y el canal. En este caso, queremos capturar todo el tráfico de la red con el ESSID “NASA-HQ-Guests”, que tiene el correspondiente BSSID “9C:C1:72:3A:5F:E” y que opera en el canal 1:

```
$ sudo airodump-ng --write open-network-NASAHQ --output-format pcap --bssid 9C:C1:72:3A:5F:E1 --channel 1 mon0
```

Para mis pruebas, tuve un cliente, mi smartphone conectado a la red inalámbrica, ejecuté los siguientes comandos:

```
$ ping -c 3 8.8.8.8
$ ping -c 3 www.google.com
$ ping -c 3 www.hardkernel.com
$ wget -p http://www.hardkernel.com/main/main.php
```

Los comandos realizan algunas pruebas básicas de conectividad y simulan un navegador cargando la página de Hardkernel, suponiendo que no está almacenada en caché. Lo mejor de todo, es que se puede repetir y genera unos 10 MB de tráfico.

Si echas un vistazo a la captura de paquetes, quizás observes uno de estos dos escenarios.

**Escenario 1:** verás un montón de tráfico de gestión, pero poco o nada de tráfico de datos. Esto me pasó la primera vez que lo probé y me esforcé bastante en intentar comprender la causa. Sospeché problemas con la modulación, las interferencias de los vecinos, las transmisiones direccionales y de trayectoria múltiple, aliens, ¡lo que fuese! Tras varias pruebas en el ODROID-C1 con el Módulo 3 Wifi, realice la misma prueba en un PC con la misma tarjeta wifi y fue capaz de capturar el tráfico milagrosamente. Esto significa que posiblemente el problema estaba en la combinación del kernel y el driver. He ejecutado el kernel estándar en el C1 y he vuelto a hacer las pruebas, pero sigo sin tener resultados. El Módulo 0 y el Módulo 4 Wifi no tienen estos problemas y permiten capturar el tráfico en el C1 sin problemas.

**Escenario 2:** verás un montón de tráfico de gestión y algo de tráfico de datos. La cantidad de tráfico de datos que ves puede variar en función de la posición de la antena, en relación con el punto de acceso y el cliente, su patrón de interferencia inalámbrica actual, además de otros factores. Basándome en las

pruebas realizadas, el módulo 0 WiFi parecía tener un menor número de paquetes perdidos. No obstante, las pruebas fueron realizadas en la misma habitación en la que estaba el punto de acceso, de modo que no puedo hacer comentarios sobre el nivel que puedas conseguir.

Guardé una captura de paquetes, está disponible en GitHub en <http://bit.ly/242cdEq>. Si la analizas la captura en Wireshark, te darás cuenta de lo siguiente:

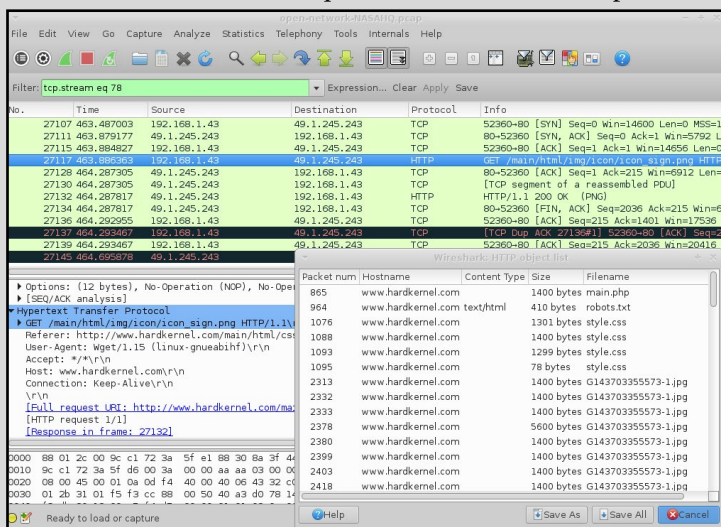
- Los paquetes 254, 256 y 258 representan el tráfico de asociación y autenticación de IEEE 802.11.
- Una vez asociado, el cliente realiza peticiones DHCP para obtener una dirección IP: paquetes 268, 269. La operación DHCP completa no está capturada.
- Tráfico ARP para detectar la dirección MAC de la puerta de enlace: paquetes 431, 435.
- Peticiones Echo ICMP a 8.8.8.8, paquetes 437. Como puedes observar se enviaron tres pings, pero sólo fuimos capaces de capturar una solicitud y sin respuesta.
- Respuesta DNS para una petición a [www.google.com](http://www.google.com): paquete 484.
- Respuesta Echo ICMP para un ping a Google, paquete 486 y la segunda petición, 616, seguida poco después por la tercera petición: paquetes 627-629. Puedes observar que los paquetes de peticiones han sido retransmitidos varias veces por la capa MAC. Esto es transparente para los protocolos de capa 3, pero puede introducir latencia y ruido adicional.
- Petición DNS para [www.hardkernel.com](http://www.hardkernel.com), retransmitida 4 veces: paquetes 638-641
- Petición Echo ICMP a la IP de Hardkernel, dos paquetes de tres: 728, 737, 738.
- Por último, una consulta DNS y la respuesta de [www.hardkernel.com](http://www.hardkernel.com), esta vez nos pide 8.8.8.8: paquetes 814, 816.
- Tráfico HTTP, empezando con un triple handshake - sólo dos paquetes capturados: paquetes 818, 824, y un HTTP GET, paquete 825.
- Tráfico HTTP con una gran cantidad de retransmisiones, como por ejemplo los paquetes: 925, 927, 929, 931.
- Desvinculación con la red Wi-Fi: paquetes 27219, 27223.

Para hacer el análisis HTTP en Wireshark, puede buscar peticiones GET y utilizar la opción “Follow TCP Stream”, o bien puedes hacer un tratamiento en masa de todo el tráfico HTTP entrando en **File -> Export Objects -> HTTP**. Aquí puede ver todas las consultas realizadas y puedes potencialmente extraer datos, como las imágenes. Por desgracia, si intentas guardar los datos del sitio, te darás cuenta que la mayoría de las imágenes están dañadas y el texto está cortado. Esto se debe a que el receptor no fue capaz de capturar todo el tráfico del medio inalámbrico. Esto es muy diferente en la captura de paquetes en Ethernet, donde se evitan las colisiones y el medio generalmente es muy fiable. El análisis resumido del tráfico

capturado te puede mostrar las limitaciones a las que te enfrentas cuando intentas capturar tráfico cifrado, de modo que puedes hacerte una idea del resultado final.

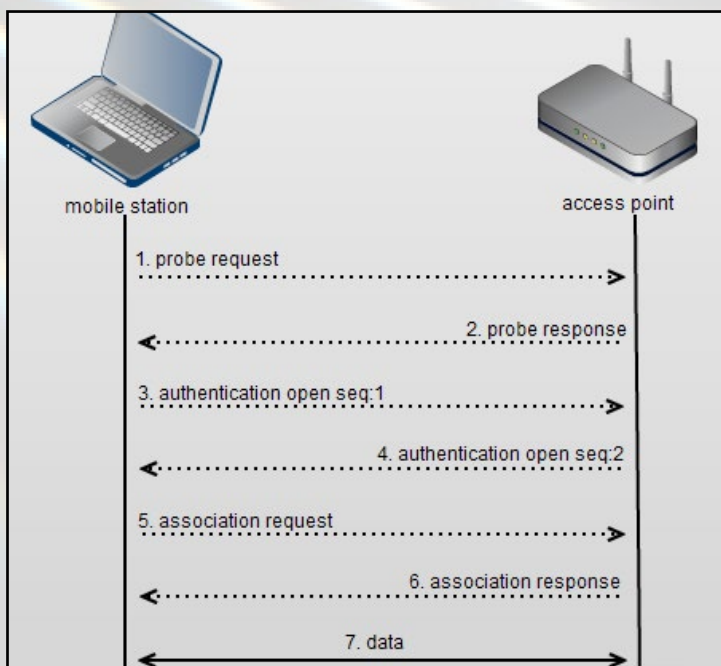
Si escuchas una red abierta que no tenga tráfico activo, posiblemente veas cosas interesantes de las estaciones activas de la red. Por ejemplo, observarás algo de tráfico ARP, transmisiones de NetBIOS de hosts Windows, paquetes multidifusión UPnP/SSDP de dispositivos DLNA, como reproductores multimedia o routers e incluso DNS multidifusión (puerto 5353) de hosts Linux y Apple que anuncian sus capacidades.

Incluso si consigues pocos datos, todavía pueden serte útiles ya que puedes analizar las peticiones de usuario y recopilar contraseñas sin cifrar y cookies enviadas con las peticiones POST. Te animo a que analices mis datos capturados,



**Figura 2 – Análisis de una red abierta**

o intentar capturar tus propios datos en una red abierta. ¡Recuerda lo que puedes llegar a ver la próxima vez que te conectes a una red abierta en un restaurante o aeropuerto!



## Cómo es la asociación de red

La asociación a una red inalámbrica conlleva básicamente dos pasos, como se muestra a continuación:

Una estación móvil envía peticiones de sondeo para descubrir las redes 802.11 de sus proximidades. Estas solicitudes de sondeo anuncian la tasa de datos soportada por las estaciones móviles y sus capacidades 802.11 tales como 802.11n. Puesto que la solicitud de sondeo se envía desde la estación móvil a la dirección de destino de capa 2 y BSSID ff:ff:ff:ff:ff:ff, a veces representado como "Broadcast", todos los PA que la reciban responderán.

Los PA que reciben la petición de sondeo la examinan para ver si la estación móvil tiene al menos una tasa de datos compatible. Si es así, se envía una respuesta de sondeo con el SSID, nombre de la red inalámbrica, tasas de datos soportadas, tipos de cifrado si se requiere y otras capacidades 802.11 del PA.

Una estación móvil elige las redes compatibles a partir de las respuestas de sondeo que recibe. Además, la compatibilidad también se puede basar en el tipo de cifrado. Una vez descubiertas redes compatibles, la estación móvil intentará la autenticación 802.11 de bajo nivel con los puntos de acceso compatibles. Recuerda que el mecanismo de autenticación 802.11 no es igual que el WPA2 o el 802.1X, que se lleva a cabo una vez que la estación móvil se autentifica y se asocia. Originalmente, la autenticación 802.11 fue diseñada para el cifrado WEP, sin embargo, se ha demostrado que este sistema de seguridad es inseguro y por lo tanto está en desuso. Por esto, los cuadros de autenticación 802.11 están abiertos y casi siempre tienen éxito.

Una estación móvil envía un cuadro de autenticación 802.11 de bajo nivel a un PA, cambia la autenticación a abierta y la secuencia a 0x0001.

El PA recibe la información de autenticación y responde a la estación móvil con el cuadro de autenticación fijando en abierto indicando una secuencia de 0x0002.

Si un punto de acceso recibe cualquier dato que no sea una petición de sondeo o autenticación desde una estación móvil, ésta no será autenticada. Entonces, responderá con una desautenticación colocando a la estación móvil en un estado de no autenticada y no vinculada. La estación tendrá que iniciar el proceso de asociación desde el paso de autenticación de bajo nivel. Llegado a este punto, la estación móvil se encuentra autenticada pero no vinculada todavía. Algunos funciones 802.11 permiten que una estación móvil tenga autenticados de bajo nivel a múltiples puntos de acceso. Esto acelera el proceso de asociación cuando nos movemos entre varios puntos de acceso, conocido como "roaming". Una estación móvil puede estar autenticada 802.11 por múltiples puntos de acceso. Sin embargo, sólo puede estar vinculada de un modo activo y transfiriendo datos a un único punto de acceso a la vez.

**Figura 3 – Proceso de Autenticación/asociación**



Una vez que una estación móvil determina a qué PA desea asociarse, envía una solicitud de asociación al PA. La solicitud de asociación contiene los tipos de cifrado elegidos si fuera necesario y otras funciones 802.11 compatibles. Si un punto de acceso recibe un cuadro de una estación móvil que está autenticada, pero aún no asociada, responderá con un cuadro de desvinculación, que coloca a la estación en un estado de autenticada pero no vinculada.

Si los elementos de la petición de asociación se ajustan a las características del PA, el AP crea un ID de asociación para la estación móvil y responderá con una respuesta de asociación que contiene un mensaje de éxito permitiéndole el acceso a la red.

La estación móvil entonces se asociará con éxito al PA y la transferencia de datos podrá empezar.

En WPA, WPA2 o 802.1x, se toman medidas adicionales después de la fase de asociación y antes de que los datos empiecen a transmitirse.

Incluso después de que un cliente se conecte a un PA, éste continúa enviando peticiones de sondeo, tanto “dirigidas” como por radiodifusión, con el fin de descubrir nuevos y potencialmente mejores puntos de acceso en sus inmediaciones. Las solicitudes de sondeo dirigidas contienen el SSID de los puntos de acceso conocidos por el cliente con la esperanza de que éstos estén cerca. Alguien que esté escuchando en el medio inalámbrico puede utilizar esto para obtener una lista de SSIDs fiables de un cliente y utilizarlos para seguirles la pista, o simplemente para crear puntos de acceso “dobles malicioso”. Puedes encontrar un video sobre este tema en <http://bit.ly/1WLAkVH>.

## SSIDs ocultos

Una forma de intentar proteger una red es ocultando su SSID. La mayoría de los routers tienen una opción para que puedas configurar tu SSID en modo “no-radiodifusión”. Esto significa que el router emite señales, pero éstas contendrán un SSID en blanco. Los clientes que deseen conectarse a la red necesitan conocer el SSID de antemano y enviar peticiones de sondeo. En teoría, esto es genial, ya que el atacante tendría que adivinar un SSID posiblemente complicado. Sin embargo, en la práctica ofrece muy poca protección.

Cuando un cliente envía una petición de sondeo, éste retransmite los SSIDs que conoce. Esta transmisión también incluye los SSIDs de los puntos de acceso ocultos. Cualquiera que escuche pueden recopilarlos, ya que están sin cifrar incluso en redes encriptadas. Ni siquiera es complicado revelar las redes ocultas, airodump-ng lo hace por defecto.

```

Before
CH 1 ][ Elapsed: 12 s ][ 2016-03-07 13:21
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
9C:C1:72:3A:5F:E1 -47 25 127 4 0 1 54e OPN <length: 14>
After
CH 1 ][ Elapsed: 28 s ][ 2016-03-07 13:21
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
9C:C1:72:3A:5F:E1 -50 29 271 12 1 1 54e OPN NASA-HQ-Guests
    
```

Vamos a configurar nuestra red en modo no radiodifusión y vamos a tener airodump-ng escuchando el tráfico, sólo para redes abiertas en nuestro caso. Te darás cuenta que las redes ocultas aparecen generalmente como “< length: xx >” en lugar

**Frame 1: AP broadcasts its presence with a hidden SSID**  
Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)  
Destination address: Broadcast (ff:ff:ff:ff:ff:ff)  
Transmitter address: HuaweiTe\_3a:5f:e1 (9c:c1:72:3a:5f:e1)  
Source address: HuaweiTe\_3a:5f:e1 (9c:c1:72:3a:5f:e1)  
BSS Id: HuaweiTe\_3a:5f:e1 (9c:c1:72:3a:5f:e1)  
IEEE 802.11 wireless LAN management frame  
Fixed parameters (12 bytes)  
Tagged parameters (84 bytes)  
Tag: SSID parameter set: Broadcast  
Tag Number: SSID parameter set (0)  
Tag length: 14  
SSID: [truncated]  
Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]

**Frame 2: Client sends a probe request asking if network NASA-HQ-Guests is around**  
Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)  
Destination address: Broadcast (ff:ff:ff:ff:ff:ff)  
Transmitter address: MurataMa\_3f:44:b7 (88:30:8a:3f:44:b7)  
Source address: MurataMa\_3f:44:b7 (88:30:8a:3f:44:b7)  
BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)  
IEEE 802.11 wireless LAN management frame  
Tagged parameters (106 bytes)  
Tag: SSID parameter set: NASA-HQ-Guests  
Tag Number: SSID parameter set (0)  
Tag length: 14  
SSID: NASA-HQ-Guests  
Tag: Supported Rates 1, 2, 5.5, 11, [Mbit/sec]

**Frame 3: AP sends a unicast probe response with its SSID, giving itself away**  
Receiver address: MurataMa\_3f:44:b7 (88:30:8a:3f:44:b7)  
Destination address: MurataMa\_3f:44:b7 (88:30:8a:3f:44:b7)  
Transmitter address: HuaweiTe\_3a:5f:e1 (9c:c1:72:3a:5f:e1)  
Source address: HuaweiTe\_3a:5f:e1 (9c:c1:72:3a:5f:e1)  
BSS Id: HuaweiTe\_3a:5f:e1 (9c:c1:72:3a:5f:e1)  
IEEE 802.11 wireless LAN management frame  
Fixed parameters (12 bytes)  
Tagged parameters (78 bytes)  
Tag: SSID parameter set: NASA-HQ-Guests  
Tag Number: SSID parameter set (0)  
Tag length: 14  
SSID: NASA-HQ-Guests

Figura 5 - Análisis de paquetes de una conexión con SSID oculto

de SSID. Tan pronto como un cliente se encuentre en las inmediaciones, ni siquiera necesita conectarse ya que emitirá peticiones de sondeo dirigidas, airodump mostrará el nombre de la red. Es muy fácil, lo único que tienes que hacer es esperar.

```
$ sudo airodump-ng -t OPN --channel 1 mon0
```

Puedes ver cómo funciona esto echando un vistazo a la captura de paquetes (figura 5). Para tu comodidad, la captura está disponible en <http://bit.ly/1U9L2F4>.

## Listas de acceso MAC

Otra forma muy común de aumentar la seguridad en las redes es usar una lista de acceso MAC en el punto de acceso. Esta lista especifica qué dispositivos pueden conectarse a la red. Puesto que la dirección MAC es única y no cambia, permite al administrador controlar los dispositivos que pueden unirse a la red. Un atacante tendría que encontrar una dirección MAC válida dentro de esa lista, y luego reemplazar su propia MAC por una de la lista para poder unirse a la red.

Figura 4 - airodump completa el SSID cuando lo escucha

Si esto fuera un juego de adivinanzas, en el peor de los casos el atacante tendría que pasar por las  $2^{48}$  direcciones, aunque en términos de probabilidad sería suficiente la mitad. Si se tar-

Vendor	Number of OUIs	Max search time (years)
Apple	471	250
Samsung	398	211
LG	127	67
Lenovo	50	26
HTC	29	15
Xiaomi	26	13

**Tabla 1 - Tiempo de búsqueda máximo en años para unos proveedores**

da 2 segundos en probar una dirección, serían necesarios unos 17 millones de años para pasar por todas ellas. Si el atacante no está dispuesto a esperar tanto tiempo, podría probar con las MAC de un proveedor en particular, como Apple, Samsung, HTC, etc, con la esperanza de conseguirla de un modo más rápido. Los tres primeros bytes, 24 bits, de una dirección MAC representan el ID del proveedor, se puede conseguir una lista de proveedores desde IEEE en <http://bit.ly/1Idlxf2>.

```
$ wget standards-oui.ieee.org/oui.txt
$ grep '(hex)' oui.txt | grep -i Samsung | wc -l
```

Incluso si aún sigue siendo demasiado tiempo, escanear un OUI, 24 bits, requiere algo más de 6 meses. Si aceleramos el proceso usando múltiples tarjetas, la búsqueda se reduce considerablemente. Se podría llevar a cabo un ataque que podría tener éxito en unos cuantos meses.

Es evidente que esta forma no es la usada por los hackers para entrar en tu sistema. Vamos a ver una opción mucho más rápida, escuchar una dirección MAC que ya tiene acceso. El modo más simple para escuchar una red y ver qué clientes están asociados a ella es utilizar Kismet. Si hay poca actividad, tendrás que esperar un tiempo, pero si estamos ante un punto de acceso concurrido, los clientes se darán a conocer al instante. Disponer de una lista de clientes conectados significa tener una lista de direcciones MAC autorizadas, de modo que puedes dar por finalizada tu búsqueda rápidamente. Sin embargo, tener dos dispositivos con la misma MAC en la misma LAN es un gran problema, de modo que para entrar en la red tienes que

```
root@iqp06:~# ifconfig wlan0
wlan0    Link encap:Ethernet  Hwaddr 7c:dd:90:73:5a:3c
         BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@iqp06:~# macchanger --mac 88:30:8a:3f:44:b7 wlan0
Current MAC: 7c:dd:90:73:5a:3c (Shenzhen Ogemray Technology Co., Ltd.)
Permanent MAC: 7c:dd:90:73:5a:3c (Shenzhen Ogemray Technology Co., Ltd.)
New MAC: 88:30:8a:3f:44:b7 (Murata Manufacturing Co.,Ltd.)
root@iqp06:~# ifconfig wlan0
wlan0    Link encap:Ethernet  Hwaddr 88:30:8a:3f:44:b7
         BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

echar fuera a un usuario real, mediante el envío de un montón de paquetes de desautenticación, como veremos más adelante.

Sin embargo, Conocer una MAC autorizada y contar con un dispositivo con esa Mac en concreto son cosas diferentes. Por suerte, es posible cambiar la dirección MAC de un adaptador WiFi, con NetworkManager o a través de macchanger:

```
$ sudo apt-get install macchanger
$ ifconfig wlan0
$ sudo macchanger --mac 88:30:8a:3f:44:b7 wlan0
$ ifconfig wlan0
```

Hasta aquí mayoritariamente hemos analizado conceptos teóricos de las redes inalámbricas. Estos te ayudarán a convertirte en un buen ingeniero de redes, pero aún no eres un “haxtor”. Estudiada la teoría podemos pasar a la parte “133t”.

## Paquetes de desautenticación

Existe una forma sencilla y eficaz de causar problemas en tu sistema inalámbrico: inundar la red con paquetes de desautenticación. Estas señales dirigidas al punto de acceso provocarán que un cliente abandone la red. No existe ningún mecanismo de protección e encriptación alrededor de los paquetes de datos, por lo que cualquier persona con un transmisor potente puede suplantar estos paquetes y hacen que los clientes se desconecten. Hay algunos casos legítimos en los que conviene hacerlo:

Los sistemas de prevención de intrusiones inalámbricas (WIPS) envían paquetes de desautenticación falsos para desconectar clientes conectados a otros puntos de acceso. Esto se hace para obligar a determinadas empresas o agencias gubernamentales a conectarse a una serie de PA dentro de un área geográfica específica, tal como se describe en <http://bit.ly/1T5INi6>.

Expulsar a tu suegra o al niño de la red. Probablemente sea más sencillo desde el propio punto de acceso, pero técnicamente no es ilegal con los paquetes de desautenticación.

### Existen algunos casos ilegítimos para usar los paquetes de desautenticación:

- Para revelar los SSID ocultos
- Para capturar datos WPA y WPA2, obligando a los clientes a volver a conectarse
- Llevar a cabo ataques de Denegación de Servicio

Para enviar un paquete de desautenticación necesitas aireplay-ng:

```
$ sudo aireplay-ng -0 10 -a 9C:C1:72:3A:5F:E1\
-c 88:30:8A:3F:44:B7 mon0
```

### Los parámetros son los siguientes:

- 0 : Significa desautenticación
- 10 : El número de intentos a realizar. Cada intento envía 64 paquetes al cliente y 64 paquetes al PA. Si se coloca un cero el

**Figura 6 - Cambiando la dirección MAC**



```
root@qp06:~# aireplay-ng -0 10 -a 9C:C1:72:3A:5F:E1 -c 88:30:8A:3F:44:B7 mon0
15:56:20 Waiting for beacon frame (BSSID: 9C:C1:72:3A:5F:E1) on channel 11
15:56:21 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:21 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:22 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:22 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:23 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:23 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:24 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:24 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:25 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:25 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:26 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
15:56:26 Sending 64 directed DeAuth. STMAC: [88:30:8A:3F:44:B7] [ 0] 0 ACKs]
root@qp06:~#
```

Destination	Protocol	Info
MurataMa_3f:44:b7	(R 802.11	Acknowledgement, Flags=.....
Broadcast	802.11	Beacon frame, SN=3714, FN=0, Flags=....., BI
MurataMa_3f:44:b7	802.11	Deauthentication, SN=0, FN=0, Flags=.....
MurataMa_3f:44:b7	802.11	Deauthentication, SN=0, FN=0, Flags=...R...
MurataMa_3f:44:b7	802.11	Deauthentication, SN=0, FN=0, Flags=...R...
MurataMa_3f:44:b7	802.11	Deauthentication, SN=0, FN=0, Flags=...R...
MurataMa_3f:44:b7	802.11	Deauthentication, SN=0, FN=0, Flags=...R...
HuaweiTe_3a:5f:e1	802.11	Deauthentication, SN=1, FN=0, Flags=.....
MurataMa_3f:44:b7	(R 802.11	Acknowledgement, Flags=.....
MurataMa_3f:44:b7	802.11	Deauthentication, SN=2, FN=0, Flags=.....
MurataMa_3f:44:b7	802.11	Deauthentication, SN=2, FN=0, Flags=...R...
MurataMa_3f:44:b7	802.11	Deauthentication, SN=2, FN=0, Flags=...R...

Figura 7 - Ataque de Desautenticación

envío de paquetes de desautenticación será constante.

-a : es la dirección MAC del punto de acceso al que está conectado el cliente.

-c : es la dirección MAC del cliente. Si se omite, aireplay enviará cuadros de radiodifusión para desconectar todos los clientes conectados al PA. Esta es la forma de hacer un ataque de denegación de servicio.

Aunque aireplay es una herramienta muy potente, hay mejores. Por ejemplo, si instalas mdk3, descrita en <http://bit.ly/1psIKUw>, tienes la posibilidad de desconectar todos los clientes inalámbricos desde todos los puntos de acceso visibles, o hacer otros tipos de ataques. Para instalarla, tienes que descargarla y compilarla por ti mismo:

```
$ git clone \
https://github.com/wi-fi-analyzer/mdk3-master
$ cd mdk3-master
$ make
$ sudo make install
```

Para desconectar todos los clientes que están conectados a cualquier punto de acceso en el canal 6, puedes ejecutar el siguiente comando:

```
$ sudo mdk3 mon0 d -c 6
```

¡Pero hay más! Mdk3 también puede emitir transmisiones de puntos de acceso falsos. Por ejemplo, yo he hecho público puntos de acceso falsos con el cifrado WEP en el canal 11 con nombres aleatorios. Si esto no llega a bloquear o ralentizar los ordenadores de tus vecinos, al menos, los desconcertará al intentar conectarse a una nueva red. Ten en cuenta que la función de radiodifusión del PA no funciona correctamente con el Modulo 3 Wifi de ODDROID, aunque sí que funciona muy bien con los otros dos módulos wifi.



Figura 8 - ¡Hay muchas redes donde elegir!

```
$ sudo mdk3 mon0 b -c 11 -h 11 -w -g
```

## Visitar la sede de Hardkernel

Hasta aquí hemos escuchado redes WiFi, expulsado gente y testeado el alcance de las transmisiones. Ahora, vamos a probar algo más atrevido; vamos a emitir puntos de acceso específicos e influir en dispositivos cercanos que piensen que están en un lugar diferente. Vamos a visitar la sede central de Hardkernel en Corea del Sur, ¡Abróchate el cinturón de seguridad!

La geolocalización basada en red en dispositivos móviles funciona recopilando las direcciones MAC de los puntos de acceso cercanos y enviando los datos a un servicio proveedor de localización, que generalmente es Google si estás en Android. El servicio de localización busca la lista de direcciones MAC en una base de datos interna e informa de una ubicación aproximada. Es curioso que la ubicación de red no se basa para nada en la dirección IP. Tal vez porque podrías conectarte a Internet a través de una serie de túneles y el punto final podría estar muy lejano geográficamente. Lo que vamos a hacer ahora es generar lo suficientes puntos de acceso usando mdk3 como para engañar al dispositivo móvil.

Para extraer la información del punto de acceso podemos utilizar wgle.net, disponible en <http://wgle.net/search>. Rellena las coordenadas y hacer clic en Search, que sólo está disponible para usuarios registrados. Puede copiar y pegar los resultados en un archivo de texto. He probado dos métodos, pero el área pequeña es la que me funcionó:

### Area grande:

**320 puntos de acceso:** <http://bit.ly/1NHtIjX>

### Area pequeña:

**45 puntos de acceso:** <http://bit.ly/217LBAA>

Puedes procesar el archivo y extraer sólo los BSSID y SSID:

```
$ cat hardkernel-small.txt | grep 'infra\'
| sed 's/infra.*//\' | sed -r 's/^map\+s//\'
| sed -r 's/\\t/ /\' > hardkernel-small-ssid.txt
```

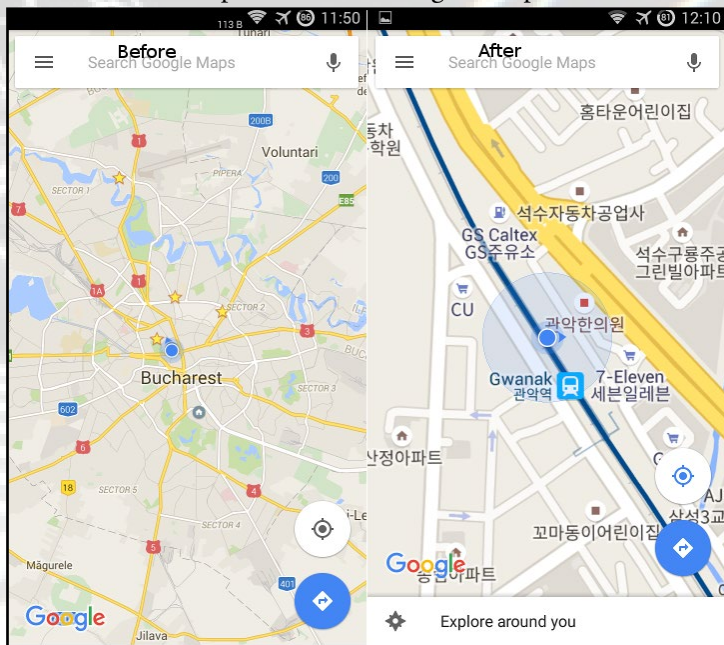
Para que el ataque tenga éxito, la víctima tiene que tener activada la localización por red, ajustando el WiFi en ahorro de batería y no tener otras fuentes de ubicación disponibles.

Esto significa que el GPS debe estar apagado o no tener satélites GPS a la vista. Si el dispositivo está en modo de avión, ayudará a mantener el bloqueo más tiempo, pero no es obligatorio. El Wifi tiene que estar encendido y la víctima tiene que ser capaz de acceder a Internet para poder comunicarse con el servicio de localización por red. Una típica víctima podría ser una tablet sin conexión de datos 3G.

**Para empezar emitir redes con mdk3, utiliza el siguiente comando:**

```
$ sudo mdk3 mon0 b -v\  
hardkernel-small-ssid.txt -g -t
```

Debería ver las redes que van apareciendo en tu lista de redes, pero si esperas y esperas un poco más ... parece que no sucede nada. Tuve que abrir Google Maps en mi teléfono y apenas se movió. Esto ocurría porque las nuevas redes entraban en conflicto en términos de ubicación, con las redes actuales de mi zona y el servicio de ubicación no se aclaraba. En este estado de “confusión”, el servicio de localización prefería mantener la ubicación actual original. Si no tienes muchas redes inalámbricas a tu alrededor, puedes omitir el siguiente paso.



**Figura 9 - Experimentando con la ubicación**

Para silenciar las redes de mi alrededor necesitamos algo más drástico - una jaula de Faraday. Se trata de una jaula metálica que protege de las ondas electromagnéticas y puede eliminar redes inalámbricas y datos 3G. ¿Cómo se construye una jaula de Faraday casera? Probé con papel de aluminio, pero no funcionaba. Necesitaba algo más robusto, como un microondas. Si pongo mi teléfono en el interior del microondas, con la alimentación desconectada y cierro la puerta, puedo comprobar que la señal de la red de los puntos de acceso wifi disponibles disminuía significativamente.

Por lo tanto, coge tu ODROID, conéctalo a una batería, inicia mdk3 y mételo en el microondas junto con un teléfono. ¿Suena como si fuera una historia de terror? ¡No te alarmes, aunque asegúrate de desconectar el microondas!

Dejé el teléfono y el ODROID en el interior durante unos minutos y no pasó nada. Todavía tenía la misma ubicación en el mapa. Sin embargo, cuando prácticamente me había rendido y abrí la puerta del microondas, la ubicación cambió rápidamente a la sede de Hardkernel. Entonces, ¿qué había ocurrido? Resulta que la conexión de datos inalámbrica que estaba usando había fallado, la señal no se recibía y el teléfono no podía ponerse en contacto con el proveedor de ubicación. Sin embargo, mientras se encontraba en el microondas, había recopilado suficientes puntos de acceso coreanos como para pensar firmemente que se encontraba en otra ubicación. Más interesante aún, el teléfono mantuvo su ubicación incluso cuando se saco de la jaula de Faraday. Todavía podía ver los mismos puntos de acceso y estaba algo confundido con el resto de su alrededor, prefirió guardar la ubicación hasta que el ODROID dejó de transmitir.

## Conclusión

En vista de las pruebas que hemos realizado, podemos ver que 802.11 está muy lejos de ser seguro incluso cuando se utiliza la robusta encriptación AES. Un atacante puede fácilmente desconectar cualquier objetivo, y con suficientes intentos de desconexión puede forzar al objetivo a conectarse a un punto de acceso diferente. El atacante puede realizar ahora ataques man-in-the-middle contra la víctima, sin que la víctima lo sepa. Un mecanismo de defensa contra algunos de estos ataques es el estándar 802.11w (<http://bit.ly/1rrivQh>) que añade cuadros de gestión protegida, pero no suele ser compatible por todos los clientes.

En términos de privacidad, las redes inalámbricas son horribles. Cada dispositivo transmite sus redes conocidas cada pocos segundos, incluso si ya está conectado. Para solucionar esto, se puede eliminar las redes conocidas de la configuración inalámbrica o mantener el wifi apagado cuando no se esté utilizando.

Las redes ocultas y listas de acceso MAC sólo son eficaces si a la red que estás intentando asegurar sólo se accede en raras ocasiones. Por ejemplo, la wifi de tu casa de vacaciones. De lo contrario, estas medidas se pueden considerar poco eficaces, ya que se pueden evitar con muy poco esfuerzo.

Los servicios basados en puntos de acceso inalámbricos, tales como la ubicación son poco fiables ya que pueden ser engañados con facilidad. Por ejemplo, si el teléfono de tu objetivo tiene programas como Tasker que ejecuta acciones basándose en datos de localización, como desbloquear el teléfono cuando está cerca de un punto de acceso específico, puedes utilizar un ataque inalámbrico para abrir nuevas vías de ataque. Si tiene preguntas, comentarios y deseas más información, visita el hilo de soporte en <http://bit.ly/1r7wLNv>.



# REALIDAD AUMENTADA

## USANDO LA OCAM Y EL ODRROID-XU4

por [nowdac@withrobot.com](mailto:nowdac@withrobot.com)



La realidad aumentada o RA, es una parte de la realidad virtual, RV que se han convertido en algo muy de moda hoy día. En pocas palabras, La RA superpone objetos artificiales o información sobre una imagen real. A pesar de que ha existido desde hace mucho tiempo, la RA se ha hecho muy popular con el tan reciente y amplio uso de los dispositivos móviles.

En este tutorial mostraremos código de ejemplo, lo compilaremos y lo ejecutaremos. La aplicación intentará colocar un curioso personaje animado sobre un marcador especial.

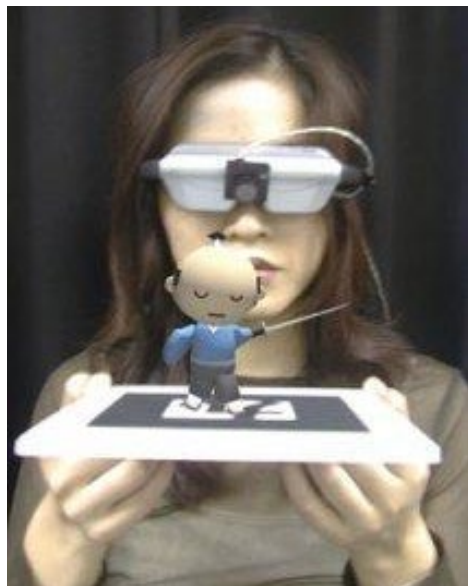


Figura 1 - Ejemplo de configuración de RA con RV (Fuente: ARToolkit)

### Funcionamiento

Para ejecutar la demo de RA, necesitamos un fotograma de vídeo que contenga el marcador especial. El programa detecta el marcador y calcula la posición

y la orientación del mismo. Usando estos datos, podremos colocar un personaje virtual en la posición y postura correctas.

Para el marcador especial, utilizaremos ArUco que es una parte de la librería OpenCV. Para más detalles sobre ArUCO, puedes consultar documentación en <http://goo.gl/Ao6hBg>. Para crear el personaje animado, usaremos

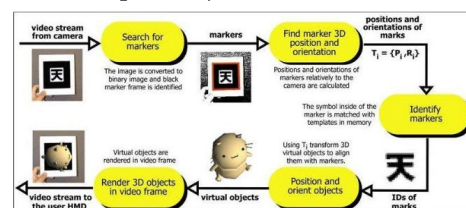


Figura 2 - Principio de funcionamiento de RA (Fuente: ARToolkit)

Ogre3D que es un motor de renderizado gráfico de código abierto. Puede encontrar más información sobre Ogre3D en <http://goo.gl/k1bMO4>.

Puesto que vamos a usar varias librerías en esta aplicación, puede parecerse un poco complicado. Sin embargo, cuando veas al pequeño y gracioso personaje ejecutándose en RA, estarás de acuerdo conmigo en que vale la pena.

### Configuración

Necesitamos los siguientes elementos para ejecutar la aplicación:

- ODRROID-XU4
- oCam
- Impresora para imprimir el tablero marcador ArUco
- Librerías OpenCV, ArUco, y Ogre3D

Empieza introduciendo el siguiente comando, que nos asegura que contamos con el último listado de paquetes:

```
$ sudo apt-get update
```

Prepara un directorio "Project" para gestionar el código de ejemplo para ArUco y Ogre3D introduciendo los siguientes comandos:

```
$ cd ~
$ mkdir Project
$ cd Project
```

Después, instala la última versión de OpenCV, que es la 2.4.9:

```
$ sudo apt-get install libopencv-dev
```

### Compilación

Para compilar ArUco, necesitamos instalar las librerías con el siguiente comando:

```
$ sudo apt-get install cmake
build-essential\
libicu-dev freeglut3 freeglut3-dev\
Libgstreamer0.10-dev\
libgstreamer-plugins-base0.10-dev libxine2-dev
```

Ahora estamos listos para descargar y descomprimir el código fuente de ArUco. Para ello, escribe los siguientes comandos:

```
$ cd ~/Project
$ mkdir aruco && cd aruco
$ wget http://sourceforge.net/projects/aruco/files/1.3.0/aruco-1.3.0.tgz
$ tar xzvf aruco-1.3.0.tgz
$ cd aruco-1.3.0
```

Por último, compilamos ArUco:

```
$ mkdir build && cd build
$ cmake ..
$ make -j 4
$ sudo make install
```

El argumento “-j 4” tras el comando “make” se usa para crear 4 tareas de compilación. Con 8 núcleos disponibles en el ODROID-XU4, esto hará que proceso de compilación sea mucho más rápido.

## Pruebas

Una vez finalizada con éxito la compilación, es el momento de probar ArUco. Para ello, necesitaremos un tablero marcador ArUco, oCam y una serie de parámetros esenciales de la cámara.

Para crear un tablero marcador ArUco, podemos utilizar estos comandos:

```
$ cd utils
$ ./aruco_create_board 6:4 board.png board.yml
```

Los argumentos para aruco\_create\_board son:

```
aruco_create_board
[cols:rows]
[board image filename]
[board information filename]
```

El archivo de información “board.yml”, contiene la información sobre los marcadores, como son los identificaciones y las posiciones de las esquinas. Si imprimimos “board.png”, podemos hacernos con un tablero ArUco, como el que se muestra en la Figura 3.

Si sabes cómo calibrar la cámara, puede ejecutar la calibración y usar los

correspondientes parámetros en formato XML/YAML de OpenCV. Si no está familiarizado con este proceso, en cualquier editor de texto crea un “camera.yml” y pega el siguiente contenido.

```
%YAML:1.0
```

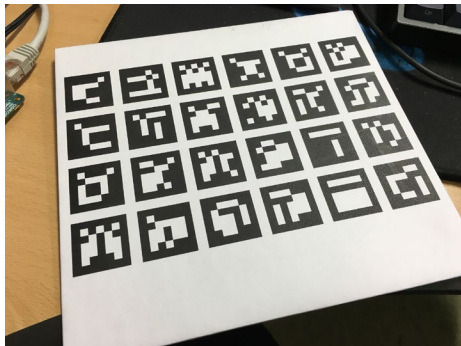


Figura 3 – Tablero ArUco impreso con 6 x 4 marcadores

```
image_width: 640
image_height: 480
camera_matrix: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  data: [ 500., 0., 320.,
         0., 500., 240.,
         0., 0., 1. ]
distortion_coefficients: !!opencv-matrix
  rows: 1
  cols: 4
  dt: d
  data: [ 0., 0., 0., 0. ]
```

Es posible que quieras usar algún editor con interfaz gráfica, como Pluma, similar al Bloc de notas de Windows.

```
$ pluma camera.yml
```

Puedes encontrar más información sobre la calibración de la cámara en <http://goo.gl/j3j0Xn>.

## Ejecución

Una vez que la oCam esté conectada al ODROID-XU4, estamos listos para probar ArUco con el siguiente comando:

```
$ ./aruco_test live camera.yml
```

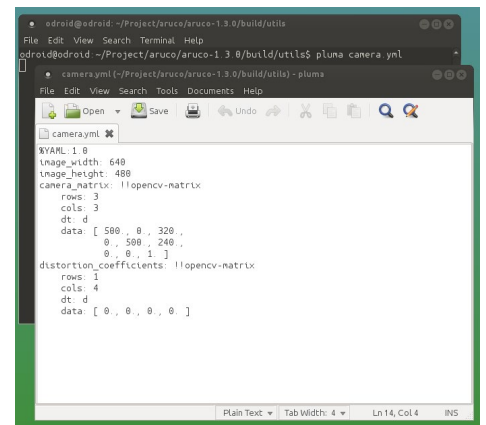


Figura 4 - Creación de camera.yml usando el editor de Pluma

```
0.025
```

Cuando veamos el tablero de marcadores ArUco con la OCAM, el programa detectará los marcadores y mostrará sus IDs y posiciones en 3D.

El siguiente paso es comprobar si podemos detectar el tablero:

```
$ ./aruco_test_board live board.yml camera.yml 0.025
```

Aquí verás que hemos especificado “board.yml” y “camera.yml” como parámetros para detectar el tablero.

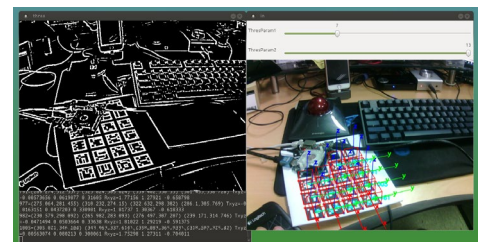
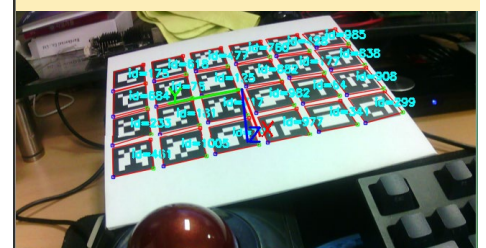


Figura 5 - Pantalla del programa de prueba del marcador ArUco

Al igual que el programa de prueba de marcadores ArUco, este programa de prueba del tablero ArUco detecta el tablero y muestra también la posición del tablero en 3D

Figura 6 - Pantalla del programa de prueba del Tablero ArUco





Casi estamos en el punto donde vamos a poder añadir nuestra personaje animado al tablero. Para utilizar Ogre, necesitamos instalar la librería:

```
$ sudo apt-get install libo-
gre-1.8-dev libois-dev
```

Para conseguir el código fuente que contiene ArUco y Ogre, utiliza los siguientes comandos:

```
$ cd ~/Project
$ mkdir ogre-test && cd ogre-test
$ wget https://sourceforge.net/
projects/aruco/\
files/ogre-test/ogre-test-
0.0.3.tgz
$ tar xzvf ogre-test-0.0.3.tgz
$ cd ogre-test-0.0.3
```

Desafortunadamente, este código fuente de uso público tiene algunos errores. Necesitas abrir tres fuentes, archivos “\*.cpp” y editarlos para corregir los errores. La solución es la misma para cada archivo, la línea donde se debe realizar la modificación aparece junto al nombre del archivo.

```
aruco_test_ogre.cpp (línea 164)
aruco_test_board_ogre.cpp (línea 161)
aruco_test_board_ogre_mask.cpp
(línea 179)
```

Cambia la siguiente línea:

```
if (argv[1]=="live") TheVideoCap-
turer.open(0);
```

Por esta:

```
if (string(argv[1]).
compare("live")==0) TheVideoCap-
turer.open(0);
```

También tenemos que corregir “CMakeLists.txt” en la línea 41 añadiendo “boost\_system”, tal y como se muestra a continuación:

```
set (REQUIRED_LIBRARIES ${OpenCV_
LIBS} ${aruco_LIBS} ${OGRE_LI-
BRARY} ${OIS_LIBRARY} boost_sys-
tem)
```

Tras hacer estas correcciones, estamos listos para compilar el código editado:

```
$ cmake -DCMAKE_BUILD_
TYPE=Release
$ make -j 4
```

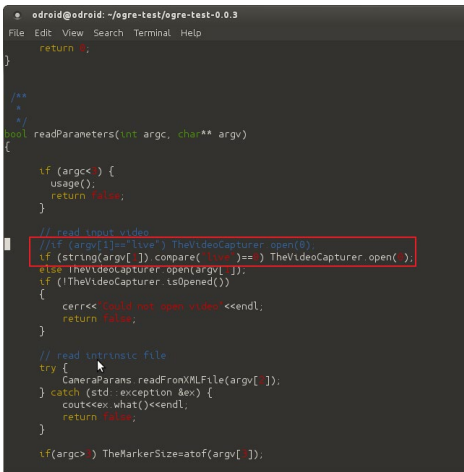


Figura 7 - Corrección del código fuente

Ahora tenemos que hacer dos cosas más. La primera consiste en modificar la ruta de acceso a la librería de Ogre definida en “plugins.cfg” en la línea 4, para que sea la adecuada para nuestro entorno ODROID.

```
PluginFolder=/usr/lib/i386-linux-
gnu/OGRE-1.7.4
```

Por esto:

```
PluginFolder=/usr/lib/arm-linux-
gnullabihf/OGRE-1.8.0
```

Lo último es añadir los parámetros del tablero y los parámetros esenciales de la cámara. Para ello, podemos utilizar los archivos que ya están preparados para las pruebas de detección.

Copia los archivos “board.yml” y “camera.yml” usando los siguientes comandos.

```
$ cp ~/Project/aruco/aruco-1.3.0/
build/utils/board.yml
$ cp ~/Project/aruco/aruco-1.3.0/
build/utils/camera.yml
```

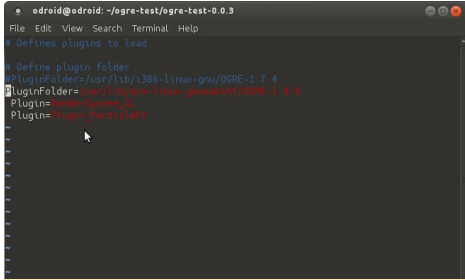


Figura 8 - La modificación de la ruta de acceso a la librería Ogre

Finalmente, estamos listos para iniciar nuestra demo de RA.

```
$ ./aruco_test_board_ogre live\
camera.yml board.yml 0.025
```

Aceptar todos los ajustes por defecto y haz clic en “Accept”. Visualizando el tablero de marcadores ArUco con la oCam, por fin podemos encontrarnos con nuestro pequeño personaje Simbad.

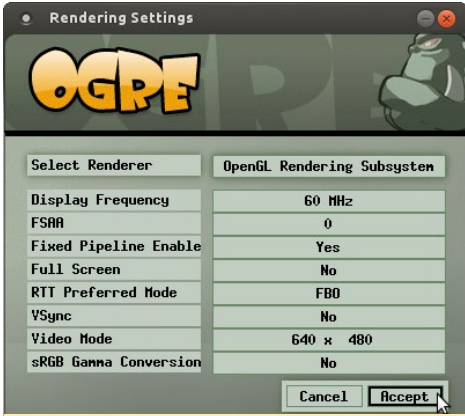


Figura 9 - Lanzador para ejecutar la demo RA

Por favor, echa un vistazo al video de <http://goo.gl/PYRQpw> para ver esta demo de RA en acción utilizando una oCam y un ODROID-XU4.

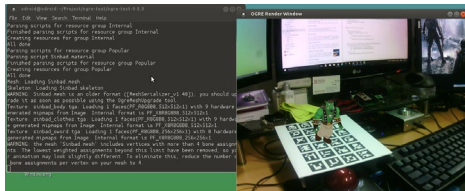


Figura 10 - Personaje de RA, Simbad

# JUEGOS LINUX

## ATARI JAGUAR

### SOBRE ODROID-XU3/XU4

por Tobias Schaaf

**R**ecientemente, he trabajado mucho en mis imágenes, especialmente con los emuladores. Intenté mejorar el rendimiento y realice un montón de pruebas. Durante las pruebas, conversé con @ptitSeb de los foros OpenPandora, y hablamos del núcleo Virtual Jaguar del proyecto libretro, el cual es bastante lento en todos los dispositivos ODROID, incluso en los potentes ODROID-XU3 / XU4. Me comento que ellos utilizan el emulador independiente Virtual Jaguar para ejecutar juegos de Atari Jaguar en OpenPandora, que es una placa ARM de un sólo núcleo con una CPU Cortex-A8 a 1 GHz. Según él, debería funcionar mucho mejor en placas como el ODROID-XU3 / XU4 de lo que ya lo hace en OpenPandora. Revisamos el código juntos, y con algunos parches creados por @ptitSeb para OpenPandora, fuimos capaces de exportar el emulador independiente Virtual Jaguar a ODROID usando QT5 para poder activar la aceleración OpenGL ES. Después realicé un montón de pruebas para ver de lo que realmente era capaz este emulador y qué tal funcionaba en los ODROIDS, precisamente es de lo que voy a hablar en este artículo.

### Introducción

La Atari Jaguar fue la última consola de Atari, se comercializó como una consola de 64 bits para competir con las consolas de 16 y 32 bits ya existentes, como la SNES, la Sega Genesis, o la 3DO. Fue lanzada en noviembre de 1993 en América del Norte, y más tarde en Europa y Japón. Contaba con múltiples procesadores para diferentes tareas y era bastante rápida (13.295MHz) en comparación con una SNES (3,58 MHz) o Sega Genesis (7.6MHz). Más tarde, se le añadió un CD y unos cascos de RV a modo de complemento. Aunque no es una mala consola, dejó de fabricarse en 1996, ni siquiera llegó a los 3 años de vida tras su aparición. Una de las razones de sus bajas ventas podría estar relacionada con el mando que presentaba un aspecto algo extraño, no estaba bien diseñado.

### Juegos

El diseño de la Atari Jaguar con diferentes chips para diferentes tareas hace que sea muy difícil desarrollar juegos para la consola, debido al hecho de que Atari realmente no impulsó su desarrollo, sólo se publicaron 67 títulos bajo licencia para la Atari Jaguar. El complemento CD de Jaguar consiguió otros 15 títulos llegando a un total de 82 juegos, la mayoría de los cuales



**Nuestro editor artístico, Bruno, trabajo de adolescente en una tienda de juegos y tuvo el sospechoso privilegio de poder jugar con una Jaguar, de modo que no tuvo que comprarla realmente.**

fueron liberados por la propia Atari. Aunque sólo hay unos cuantos juegos con licencia para la consola, a finales de 1990, la consola pasó a ser de dominio público y fue declarada como una plataforma abierta. Algunos de los nuevos juegos fueron desarrollados como juegos caseros, creados por aficionados y expertos. Incluso hoy en día, aparecen nuevos juegos de vez en cuando, como Alice's Mom's Rescue para la Jaguar CD, que fue lanzado para Atari Jaguar en 2015.

Aunque la biblioteca de Atari Jaguar es bastante limitada, tiene algunos títulos muy buenos, como Raiden, Rayman, Pitfall y Cannon Fodder, que también son conocidos en otras consolas.



**El juego de plataformas 2D Alice's Mom's Rescue lanzado en 2015 para Atari Jaguar CD**





## Soporte ODROID

Entonces, ¿Qué tal se ejecuta la Atari Jaguar sobre los ODROIDS? Me temo que en realidad no es compatible. Hay un núcleo libretto para Virtual Jaguar que se puede utilizar con retroarch, pero me pareció muy lento y muy pocos juegos llegaron a funcionar. Algunos se ejecutaban casi a plena velocidad en un XU3 / XU4 como Cannon Fodder, pero estos juegos son muy raros. Otros juegos como Pitfall o Alien vs Predator funcionan con una velocidad de fotogramas muy baja (por debajo de 30 FPS) y muchos otros no funcionan para nada. Por ejemplo, Rayman no se inicia con el núcleo libretto.

La alternativa es un emulador independiente. El emulador Virtual Jaguar funciona bien, pero no lo suficientemente como para ejecutarse en todos los ODROIDS. El C1 nunca será capaz de ejecutar un juego con fluidez. No estoy seguro si funcionará con el C2 ya que no sé si incluso se ejecutaría sin drivers X11. Ni tampoco conozco la calidad de los drivers, ya que en el momento en el que escribí este artículo no existían todavía drivers X11 para el C2. El U3, aunque es el doble de rápido que el C1, probablemente también sea demasiado lento para muchos juegos. Algunos juegos como Cannon Fodder podrían funcionar, pero por lo general yo no contaría con ello. Eso significa que Atari Jaguar probablemente esté limitada únicamente al XU3/XU4. El emulador de hecho cuenta con un filtro que te permite utilizar el filtrado bilineal para gráficos, pudiendo aumentar en gran medida la calidad de la imagen:

Me llevo un tiempo probar los 56 ROMs diferentes para Atari Jaguar y confeccionar una lista de compatibilidad para todos ellos. Me gustaría destacar algunos de los juegos y comentar un poco lo que pienso de ellos.

**El mando de Atari Jaguar no estaba bien diseñado, y era tan malo que incluso si eras secuestrado durante mucho tiempo, no llegarías a desarrollar el síndrome de Estocolmo**

La Atari Jaguar CD añadió más juegos a la biblioteca que fueron mejorados con escenas de películas, mejores gráficos y mejor sonido. Sin embargo, sólo se publicaron unos cuantos títulos, de modo la Atari Jaguar CD no podía competir con la Playstation 1 o la Sega Saturn.



Rayman para Atari Jaguar vs Rayman para la Playstation 1



19.9 | Frames: 14798



Diferencias entre no tener filtro y un filtro bilineal - la mitad superior utiliza el filtro y la mitad inferior no lo hace

## Cannon Fodder

Cannon Fodder es un juego de acción estratégica en el que envías un pequeño grupo de soldados a la batalla para luchar contra otros soldados y destruir edificios. El juego es muy divertido, y aunque el juego original se controlaba con el ratón, como en Amiga, funciona sorprendentemente bien en Atari Jaguar. El ratón virtual es muy sensible utilizando el mando, y puesto que sólo necesitas dos botones para controlar el juego es



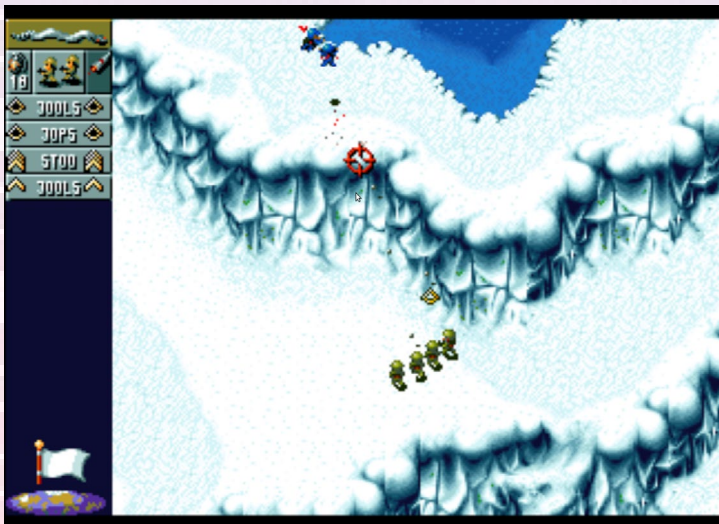


Como su nombre indica, tus soldados son básicamente Carne de cañón y son muy reemplazables

relativamente fácil de manejar.

Dispones de un montón de soldados llamados reclutas, esperando a que los envíes a la batalla, si uno de tus soldados muere simplemente es reemplazado por uno nuevo. El simple número de reclutas que recibes debería mostrarte con qué frecuencia puedes morir. Además pone de manifiesto que los soldados son muy reemplazables. Se combate en diferentes escenarios: la Jungla, el Artico, y el Desierto.

Me gusta mucho el juego, aunque hay algunas alternativas



A nice arctic level on Cannon Fodder where you fight with your soldier

donde elegir. El juego salió para diferentes consolas. Solía jugar en el Amiga, donde encuentras incluso un Cannon Fodder 2 y algunos otros juegos derivados de la serie. Aún así, la versión de Atari Jaguar es muy buena y deberías probarla.

### Flashback

Flashback - The Quest for Identity es otro simpático juego que me encantaba en Amiga. Esta versión para el Atari Jaguar también es muy buena, básicamente es un juego de acción y



La Jungla es donde comienza tu viaje

plataformas muy refinado para Atari Jaguar.

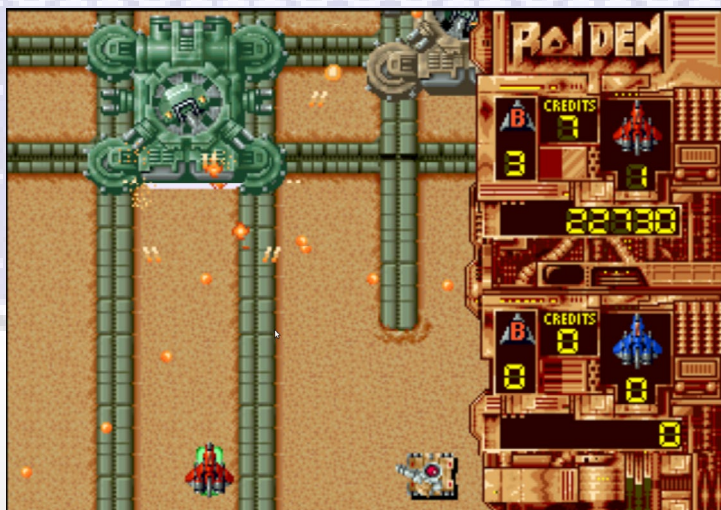
Aunque han pasado por alto la música de Amiga, que tenía mucha mejor calidad en mi opinión, también echo en falta algunas de las escenas cinemáticas de otras consolas. Sin embargo, el juego está completo y es tan jugable como cualquier otra versión del mismo. Creo que esta es la razón por la que este juego tiene el Récord Guinness como juego francés más vendido de todos los tiempos, desde mi infancia he jugado a este juego muchísimas horas sobre Amiga, te puedo decir que está muy bien conseguido.



Luchar en entornos futuristas, las minas y el teletransporte de enemigos son sólo algunos de los obstáculos a los que te enfrentas

El juego puede resultar muy difícil en ocasiones, especialmente cuando descubres la forma correcta de cómo hacer frente a los enemigos. A veces es mejor distraerlos en lugar de enfrentarse a ellos directamente, y otras veces simplemente espera a que el escudo te proteja. Yo prefiero la versión de Amiga a la





Raiden para Atari Jaguar es muy divertido, permite 2 jugadores

de Atari, aún así un es gran juego sin importar el sistema desde el cual los ejecutes.

También hay algunos shooters arcade muy buenos para Atari Jaguar, por ejemplo Raiden y Tempest 2000. Raiden de hecho tiene unos gráficos bastante pulidos y una música muy buena, se puede decir que me gusta bastante. El sistema de juego por el contrario, es algo difícil, al menos para mí. Hay algunos potenciadores repartidos por todo el nivel, aunque son poco frecuentes y si mueres, los pierdes todos y tienen que empezar de nuevo a recopilarlos, lo cual no significa que vayas a conseguirlos todos de nuevo en el mismo nivel. Sin potenciadores serás extremadamente débil. Incluso te llevará mucho tiempo matar a los enemigos "midbos", sino cuentas con ningún potenciador para tus armas, lo cual hace que el juego sea muy difícil en mi opinión. Haces muy poco daño y sin bombas las posibilidades de matar a un enemigo se reducen bastante. Aún así, el juego es muy divertido y supongo que el modo multijugador es la mejor forma de pasarse el juego.

Tempest 2000 es muy interesante, aunque no lo parece demasiado al primero, de hecho es bastante divertido. Tiene una



Tempest 2000 fue un esfuerzo por parte de Atari para ponerse al día con los juegos con entornos 3D, pone de manifiesto la diferentes con respecto a lo que había disponible para sistemas PS1/Sega Saturn

música agradable y los gráficos no son tan malos, a pesar de su simpleza. Tiene buenos efectos con las explosiones y los objetos que recoger. Puedes reunir algunos potenciadores e incluso conseguir un defensor NPC que te ayuda a con los enemigos. Presenta algo de desfase en el ODROID, pero se puede jugar perfectamente. En realidad es bastante adictivo.

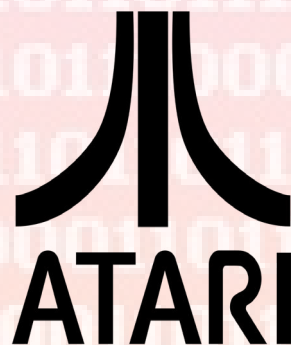
Existen más shooters arcade famosos, como Defender 2000 y Protector. Atari Carts es un juego de carreras divertido similar a Mario Carts. Pinball Fantasies es un buen juego de pinball sobre Atari. También me encanta la versión de Worms, que se ejecuta bastante bien en la Atari Jaguar, aunque se controla en un principio con el ratón y el teclado.

## Reflexiones generales

En general, Atari Jaguar es una buena consola similar a la Sega Génesis o la SNES. Por desgracia, sólo hay algunos juegos disponibles y no todos llegan a funcionar. Sólo el XU3 y el XU4 son actualmente capaces de ejecutar Atari Jaguar a una velocidad aceptable. El núcleo Libretto de Virtual Jaguar no funciona bien y son muy pocos los juegos que llegan a funcionar. El emulador independiente funciona bastante mejor.

Muchos de los juegos son en realidad versiones de otras consolas y son igual de buenos en Atari o incluso mejor, por lo que merece la pena probar estas versiones para ver las diferencias. Muchos de los juegos que conozco de Amiga son mejores en su versión original. Otros juegos como Pitfall, también existen para otras consolas menos exigentes, como la Sega 32X, de modo que se pueden ejecutar en más dispositivos aparte del XU3/XU4. Los juegos de Atari Jaguar CD parecen no funcionar, así que ni siquiera se pueden probar. Afortunadamente, ahora es posible jugar con mandos como el de la Xbox 360 en lugar de usar ese mando tan poco elegante de la Atari Jaguar. Configurar el mando es muy fácil gracias a la interfaz gráfica de usuario basada en QT5 y los drivers SDL2 para soporte de joystick.

Es una consola interesante para el XU4, aunque no es un problema si tu ODROID no es capaz de ejecutar algunos juegos, ya que probablemente existan versiones de éstos para otras consolas. Si deseas probar Atari Jaguar en tu ODROID XU3/XU4 puedes descargarlo desde mi repositorio. Espero que pases un buen rato con este trozo de historia de los videojuegos en tu ODROID.



Todos tenemos un lugar en nuestros corazones para los juegos de la vieja escuela de Atari. Podría haber sido una consola mejor, pero le gana la partida otras consolas, como la Sega Dreamcast

Compatibility List

Air Cars	not working	
Alien vs Predator	working	a little laggy
Atari Karts	working	feels slow (not laggy but slow)/ Menu seems off screen
Attack of the Mutant Penguins	not working	
Battle Sphere Gold	not working	only works until menu (but you can listen to nice music)
Brutal Sports Football	not working	
Bubsy - Fractured Furry Tails	working	
Cannon Fodder	working	Main Theme sounds slightly too fast
Checkered Flag	working	
Club Drive	partly working	graphics seems off and it crashes
Cybermorph	not working	
Defender 2000	working	
Doom - Evil Unleashed	not working	
Double Dragon V	working	
Dragon - The Bruce Lee Story	partly working	Background and Menu are not working correctly
Evolution - Dino Dudes	working	Button Layout?
Fever Pitch Soccer	working	
Fight For Your Life	working	issues in Background
Flashback	working	
Flip Out	not working	
Hover Strike	not working	crashes when you start a game
Hyper Force	not working	stops when you start a game
International Sensible Soccer	not working	crashes when you start a game
Iron Soldier	not working	stops when you start a game
Iron Soldier 2	not working	terrible sound issues in menu, stops when you start a game
I-War	working	
Kasumi Ninja	working	may require multiple starts, slow menu, missing ground



Missile Command 3D	working	all very slow (frameskipping?)
Music Demo	not working	hangs emulator
Native Demo	working	no sound
NBA Jam Tournament Edition	working	
Pinball Fantasies	working	odd controls
Pitfall - The Mayan Adventure	working	
Power Drive Rally	not working	crashes when you start a game
Protector - Special Edition	working	
Raiden	working	(only starts with BIOS)
Rayman	working	very fun to play
Ruiner Pinball	not working	
Skyhammer	partly working	terrible sound issues, you can get in game, but runs unstable
Soccer Kid	working	
Space War 2000	working	confusing game could have been nice
Super Burnout	not working	Only menu works, stops when going into game
Super Cross 3D	not working	partly working without BIOS, but with no sounds and many glitches
Syndicate	working	not a good game for Consoles
Tempest 2000	working	takes a long time to load
Theme Park	partly working	graphical glitches, runs unstable
Total Carnage	not working	crashes soon after you're in the game
Trevor McFur in the Crescent Galaxy	working	hard game (no music)
Troy Aikman NFL Football	working	sometimes crashes
Ultra Vortek	not working	
Val D'Iserre Skiing & Snowboarding	working	Graphical issues, too vague controls
White Men Can't Jump	not working	freezes in various situation
Wolfenstein 3D	not working	
Worms	working	menu slightly off screen
Zero 5	working	hard to maneuver
Zool 2	working	

# DESARROLLO ANDROID

## ANDROID SUPPORT LIBRARY

por Nanik Tolaram



Un problema muy común relacionado con el desarrollo de aplicaciones Android es la cantidad de diferentes versiones de Android que necesitan soporte, lo cual puede resultar desalentador en muchas ocasiones. En los primeros días de Android, la mayoría de los desarrolladores se esforzaban en asegurarse de que sus apps pudieran ejecutarse en la versión anterior de Android. Con el paso del tiempo, Google publicó una librería para facilitar el desarrollo de aplicaciones. En los últimos años, Google está animando encarecidamente a los desarrolladores a usar Android Support Library. El objetivo final de la librería es reducir el nivel de código necesario para que las aplicaciones puedan ejecutarse en las diferentes versiones de Android. Google ha ido publicando constantemente cambios en esta librería. Gracias a Android Support Library, los desarrolladores sólo tienen que centrarse en escribir sus aplicaciones, dejando el trabajo duro de la portabilidad de la versión de Android a la librería. En

este artículo vamos a echar un vistazo a esta librería y cómo utilizarla.

### Configuración

Lo primero que debemos hacer es asegurarnos de que tienes la librería instalada en Android Studio. Utiliza la aplicación “android” que se encuentra dentro del directorio <android\_sdk\_folder>/tools. En la sección Extras, selecciona “Android Support Library” para descargarla en tu SDK. Consulta la Figura 2 para ver cómo debería verse tu directorio SDK una vez descargada la librería.

### Versiones de librería

Como puedes ver en la Tabla 1, existen varias librerías disponibles, pero sólo tendrás que utilizar la correcta para tu proyecto. El directorio que aparece en la tabla contiene todos los archivos importantes, como son el código fuente, los recursos y los archivos de configuración de la librería.

Si miras la tabla, podrás comprobar que la mayoría de las librerías están rela-

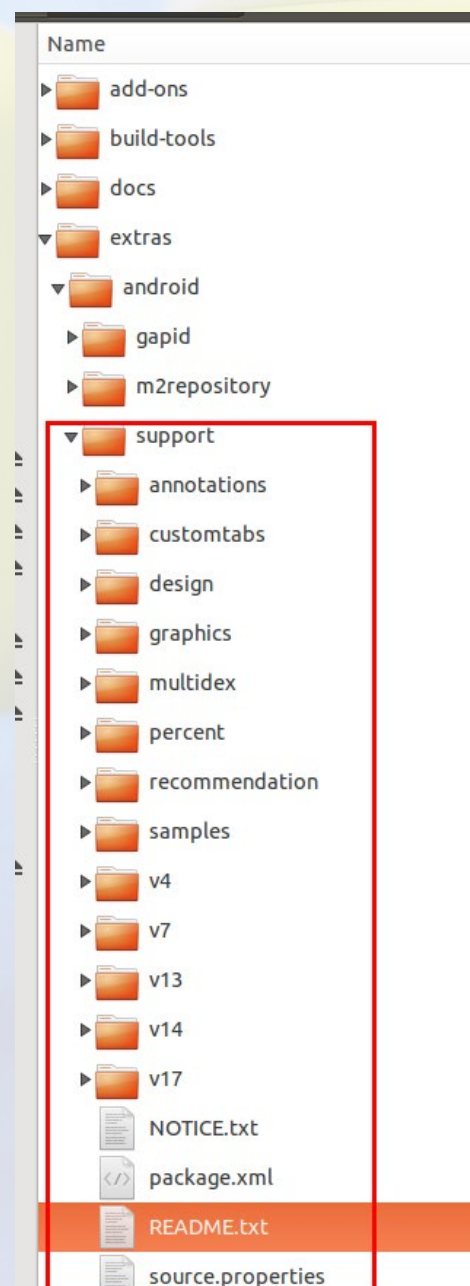


Figura 2 - Librería de soporte dentro de la carpeta SDK

Figura 1 - Gestor de descargas Android SDK

Nombre	Versión	Estado
GPU Debugging tools	1.0.3	Instalado
Android Support Repository	30	Instalado
Android Support Library	23.2.1	Instalado
Android Auto Desktop Head Unit emulator	1.1	Instalado
Google Play services for Fit Preview	1	Instalado
Google Play services for Froyo	12	Instalado
Google Play services	29	Instalado
Google Repository	25	Instalado
Google Play APK Expansion Library	3	Instalado
Google Play Billing Library	5	Instalado
Google Play Licensing Library	2	Instalado



Support Library	SDK Directory
v4 Support Library	<sdk>/extras/android/support/v4/
Multidex Support Library	<sdk>/extras/android/support/multidex/
v7 Support Libraries	<sdk>/extras/android/support/v7/appcompat/
v7 cardview library	<sdk>/extras/android/support/v7/cardview/
v7 gridlayout library	<sdk>/extras/android/support/v7/gridlayout/
v7 mediarouter library	<sdk>/extras/android/support/v7/mediarouter/
v7 palette library	<sdk>/extras/android/support/v7/palette/
v7 recyclerview library	<sdk>/extras/android/support/v7/recyclerview/
v7 Preference Support Library	<sdk>/extras/android/support/v7/preference
v13 Support Library	<sdk>/extras/android/support/v13/
v14 Preference Support Library	<sdk>/extras/android/support/v14/
v17 Preference Support Library for TV	<sdk>/extras/android/support/v17/
v17 Leanback Library	<sdk>/extras/android/support/v17/leanback
Annotations Support Library	<sdk>/extras/android/support/annotations
Design Support Library	<sdk>/extras/android/support/design
Custom Tabs Support Library	<sdk>/extras/android/support/customtabs
Percent Support Library	<sdk>/extras/android/support/percent
App Recommendation Support Library for TV	<sdk>/extras/android/support/recommendation

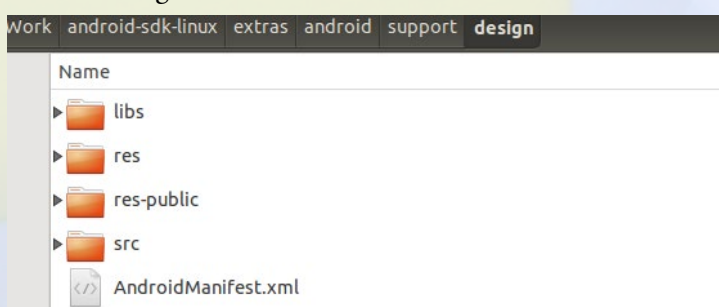
**Tabla 1 – Librerías de soporte y Localización SDK**

cionadas con la interfaz de usuario, una gran fuente de problemas a la que se enfrentan los desarrolladores a la hora de diseñar aplicaciones Android. Observa también que la librería utiliza numeraciones tales como, V7, V13, V14, etc. La Tabla 2 resume el significado para cada una de las versiones.

v4	Android 1.6, API level 4 and higher. The library includes a large set of APIs, compared to the other libraries, for: data handling, network connectivity, and programming utilities.
v7	Android 2.1, API level 7 and higher
v13	Android 3.2, API level 13 and higher.
v14	Adds support for preferred interface
v17	Targeted for Android TV devices.

**Tabla 2: Versiones de Android Support Library**

Vamos a echar un vistazo al contenido de Design Support Library que se encuentra dentro del directorio “<sdk>/extras/android/support/design”. El contenido del directorio puede verse en la Figura 3.



**Figura 3 – Directorio de Design Support Library**

Hay un directorio /src, que está vacío. El código fuente de la librería de soporte se puede descargar desde el repositorio AOSP en <http://bit.ly/1Ua2clG>.

## Demo

Vamos a echar un vistazo a un código demo al que puedes acceder desde el repositorio GitHub en <http://bit.ly/1QuHhMk>.

android / platform / frameworks / support

Clone this repo:

```
git clone https://android.googlesource.com/platform/frameworks/support
```

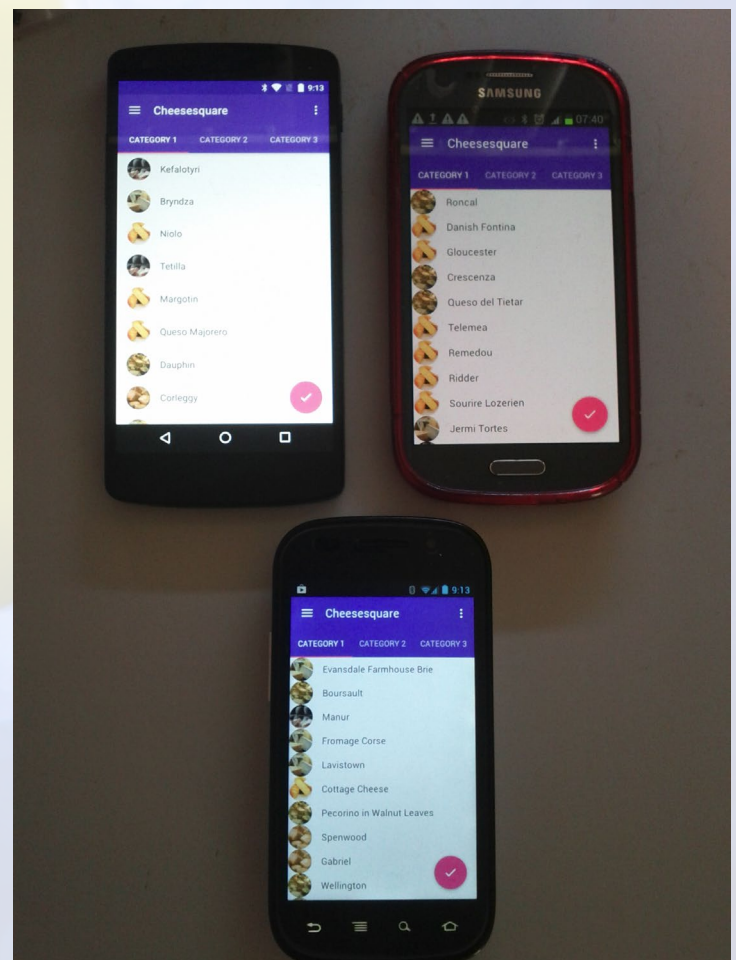
Branches

- 795681d Merge "Fix typos for the word "Unsupported"." by Stephen Hines · 8 days ago master
- 83b686c Fix typos for the word "Unsupported"; by Stephen Hines · 8 days ago
- 1c3-factoryrom-2-release 1235718 Merge "Avoid destroying Device separately." by Yang NI · 2 weeks ago android-n-preview-2
- 1c3-mr0 56e7023 Avoid destroying Device separately by Yang NI · 2 weeks ago
- 1c3-mr0-release aa83832 Merge "Removed unused variables" by Yang NI · 3 weeks ago
- 1c3-mr1 83e2547 Merge changes I91b92056,I596e00fa,I89fda42d by Miao Wang · 3 weeks ago
- 1c3-mr1-release cd76b43 [RenderScript] Update documentation for Allocation.setAutoPadding() by Miao Wang · 3 weeks ago
- 1c3-plus-aosp fca65fc [RenderScript] Fix IntrinsicConvolve documentation, by Miao Wang · 3 weeks ago
- idea133 1a2b667 [RenderScript] Update the documentation of ScriptLaunchOptions by Miao Wang · 3 weeks ago
- idea133-weekly-release cfc67d1 Merge "Turn FORTIFY back on for the RS support library." by Stephen Hines · 3 weeks ago
- jb-dev 82ad184 Turn FORTIFY back on for the RS support library, by Stephen Hines · 3 weeks ago
- More... 2312887 Removed unused variables by Yang NI · 4 weeks ago

**Figura 4 – Repositorio fuente de la librería**

La figura 5 muestra la aplicación ejecutándose en tres dispositivos Android diferentes: Nexus S - Android 4.1.2, Nexus 5 - Android 6.0, y Samsung Galaxy Express - Android 4.1.2. Observa cómo la aplicación se ve y se comporta de modo similar en todos los dispositivos.

Una cosa que tienes que decidir a la hora de escribir una



**Figura 5 - Tres dispositivos diferentes con dos versiones diferentes de Android ejecutándose**

aplicación para Android es la versión de Android más baja para la cual la aplicación tendrá soporte. No hay una norma general, ya que todo depende del mercado al que te dirijas, recursos de los que dispones y muchas otras cosas. La aplicación demo que estamos viendo es compatible con dispositivos Gingerbread,

Android 2.3 API nivel 9 y posteriores. Puede ver los dispositivos de las Figuras 6a y 6b.to support. There is no rule,

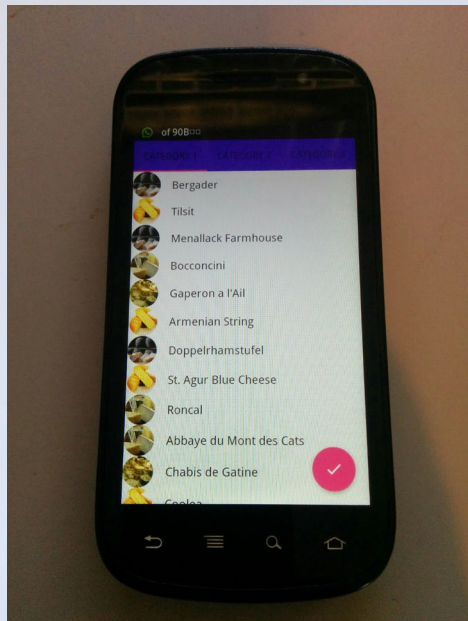


Figure 6a – Nexus S Gingerbread

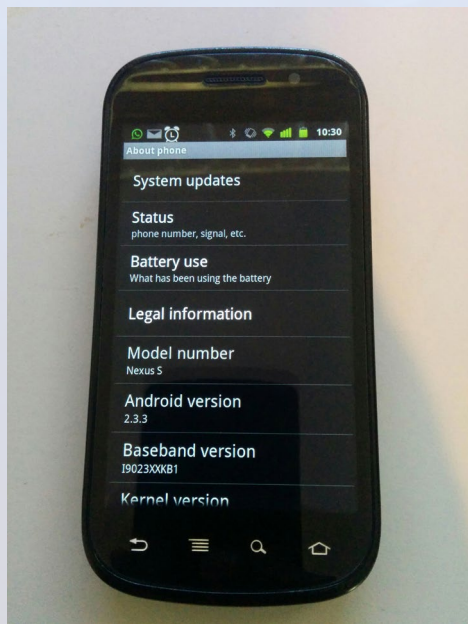


Figura 6b – Configuración del Nexus S

since it all depends on the market that you are targeting, resources that you have, and many other things. The demo app we are looking at supports devices from Android 2.3 API nivel 9 y posteriores. Puede ver los dispositivos de las Figuras 6 y 6b.

La configuración que especifica la versión de Android más baja donde quieres que se ejecute tu aplicación se

encuentra en el archivo build.gradle. El código siguiente muestra el atributo minSdkVersion que controla esto. La API de Android y la información de la versión la puedes encontrar en <http://bit.ly/1Ua4mlk>.

```
defaultConfig {
    applicationId "com.support.
    android.designlibdemo"
    minSdkVersion 9
    targetSdkVersion 23
    versionCode 1
    versionName "1.0"
}
```

El siguiente código muestra las dependencias declaradas dentro del archivo build.gradle que indican al sistema de compilación que librería puede utilizar. La aplicación demo utiliza Design Support y la librería Cardview v7, ambos usan la versión 23.1.1.

```
dependencies {
    compile 'com.android.
    support:design:23.1.1'
    compile 'com.android.
    support:cardview-v7:23.1.1'
    compile 'com.github.bumptech.
    glide:glide:3.6.0'
    compile 'de.hdodenhof:circlei
    mageview:1.3.0'
}
```

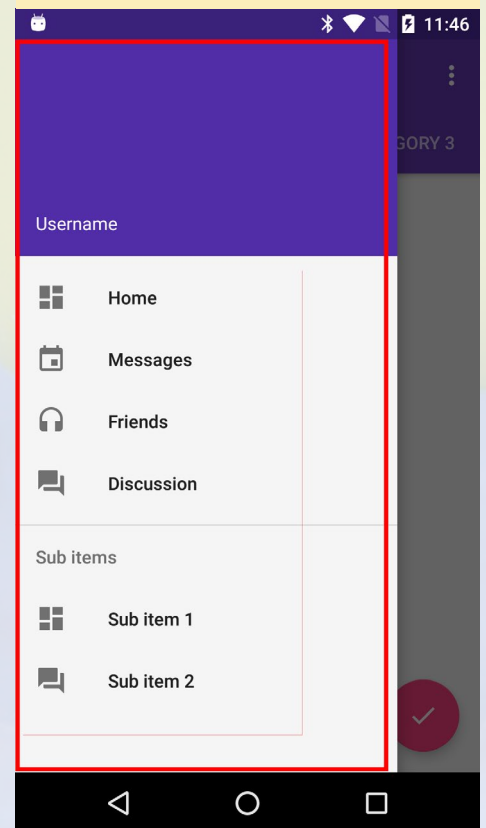
La vista personalizada que la aplicación usa es la siguiente, que se puede encontrar dentro del archivo res/layout/activity\_main.xml:

```
<android.support.v4.widget.DrawerLayout xmlns:android="http://
schemas.android.com/apk/res/an-
droid"
    xmlns:app="http://schemas.
    android.com/apk/res-auto"
    android:id="@+id/drawer_lay-
    out"
    android:layout_height="match_
    parent"
    android:layout_width="match_
```

```
parent"
    android:fitsSystemWindows="t
    rue">
    <include layout="@layout/in-
    clude_list_viewpager"/>
    <android.support.design.wid-
    get.NavigationView
        android:id="@+id/nav_
    view"
        android:layout_
    height="match_parent"
        android:layout_
    width="wrap_content"
        android:layout_
    gravity="start"
        android:fitsSystemWindows
    ="true"
        app:headerLayout="@lay-
    out/nav_header"
        app:menu="@menu/drawer_
    view"/>
</android.support.v4.widget.Draw-
erLayout>
```

La figura 7 muestra qué parte de la aplicación está usando la nueva vista personalizada de la librería, llamada android.support.design.widget.Naviga-

Figure 7 - widget NavigationView





tionView.

Hay 2 parámetros en el interior del nuevo widget: `app:headerLayout` y `app:menu`. Estos apuntan a los archivos `res/layout/nav_header.xml` y `res/layout/drawer_view.xml`. El `drawer_view` contiene el texto del submenú junto con los iconos mientras que `nav_header` contiene la parte superior de la vista. En la Figura 7, es el texto "Username".

La otra vista personalizada utilizada es un `droid.support.v4.widget.DrawerLayout` que es el diseño principal para toda la aplicación agrupando el `android.support.design.widget.NavigationView`. La vista `DrawerLayout` conoce que vista activar basándose en el siguiente código:

```
switch (item.getItemId()) {
    case android.R.id.home:
        mDrawerLayout.openDrawer(GravityCompat.
START);
        return true;
}
```

El `GravityCompat.START` le dice a la librería qué buscar dentro de la vista `DrawerLayout` para iterar hasta sus vistas hijas para encontrar qué vista ha fijado el `layout_gravity` como "start", la cual se define dentro de `activity_main.xml` en el `NavigationView`

Esto debería darte una visión general sobre el uso de Android Support Library. Veremos más a fondo la librería de soporte en futuros artículos.

## Recursos

Características de Android Support Library  
<http://bit.ly/1ei6hQ5>

Android Support Library  
<http://bit.ly/1qIpxiK>

# BATTLE FOR THE SOLAR SYSTEM: PANDORAN WAR

## UN DIVERTIDO SHOOTER ESPECIAL 2D BASADO EN MISIONES

por Tobias Schaaf

El Pandoran War ofrece una experiencia de juego muy simple para tu ODROID. El juego es un sistema de misiones no lineales, lo que significa que simplemente tienes que elegir una de la docena de misiones que más te interese a lo largo de todo el juego. Tiene lugar entre los libros 2 y 3 de la trilogía, por lo que podría valer la pena leer algunas de las muestras gratuitas en la página web de Pandoran War en <http://bit.ly/1YRi6Rj>, para conocer mejor la historia del juego. El juego se ejecuta sobre SDL2 y OpenGL ES, por lo que se aprovecha de la GPU Mali y los potentes procesadores de 4 o 8 núcleos disponible en dispositivos como el ODROID-XU4 o ODROID-C2.

### Instalación

Para instalar Pandoran War, tienes que añadir el repositorio de @meveric a tu distribución, si aún no lo has hecho:

```
$ su
# cd /etc/apt/sources.list.d/
# wget http://oph.mdrjr.net/meveric/sources.lists/meveric-all-main.list
# wget -O- http://oph.mdrjr.net/meveric/meveric.asc | apt-key add -
```

Luego, simplemente instálalo como los haría con cualquier otro juego o aplicación a través de apt-get:

```
$ apt-get update
$ apt-get install tbftss-odroid
```



Un divertido shooter en 2D, una joya para los jugadores Linux en ODROID

Para preguntas, comentarios y sugerencias, por favor visita el hilo de los foros ODROID en <http://bit.ly/1prxGY1>.

# BABY NAP (PROGRAMA DE ACTIVIDAD NOCTURNA)

## PARTE I - CONFIGURACION DEL HARDWARE

por Marian Mihailescu



**S**er padre por primera vez es un gran reto. Tener un bebé llorando y no saber cómo calmarlo rápidamente puede ser extremadamente agotador, especialmente en los casos de padres trabajadores que se turnan para cuidar al bebé. Espero que el Programa de Actividad Nocturna (Baby NAP) ayude a los padres con esta difícil situación.

Imagina una cuna con un sensor de presión en forma de almohadilla que iría debajo del niño. Un Lector de identificación por radiofrecuencia (RFID) conectado a la cuna, combinado con unas pulseras o anillos RFID para los padres, puede determinar cuando el bebé esta reclinado, por cuánto tiempo y qué padre lo hace. Se puede utilizar un sensor de sonido para registrar la amplitud del sonido del bebé y determinar si el bebé está llorando, por cuánto tiempo y lo fuerte que es el llanto. Se pueden utilizar un sensor de infrarrojos (pasivo) piroeléctrico (PIR) y un sensor de luz para registrar el movimiento y la luz ambiental. Todos estos sensores se pueden conectar a su vez a un ordenador de placa reducida (SBC) como el ODROID C1 /C1+. Esta placa puede subir los datos de los sensores a un servicio basado en la nube como los Amazon Web Services (AWS), para tu posterior tratamiento.

Para aquellos que quieran ampliar esta configuración, pueden por ejemplo, utilizar una bombilla Philips Hue para controlar el color de la luz de la habi-

tación del bebé, y un micrófono para filtrar el ruido del bebé y determinar, si los padres están cantando una nana, cómo de efectivo es poner al bebé a dormir, etc.

Con una función lambda AWS y un panel de control con las lecturas de los sensores, los padres pueden sacar conclusiones muy interesantes sobre las preferencias de sueño de su bebé, tales como:

- ¿Tiene preferencia por un determinada luz ambiental?
- ¿Le gusta que lo mezan cuando se va a dormir?
- ¿Se duerme más rápido cuando le cantamos una nana?
- ¿Suele despertarse cuando hay luz en la habitación?

Los padres también pueden comprobar el tiempo que pasan por la noche con el bebé, y así saber rápidamente a quien le toca dormir al bebé. Aunque este proyecto no pretende sustituir un monitor de bebé, podría enviar notificaciones a los padres si el bebé empieza a llorar usando Amazon Simple Notification Service (SNS).

### Componentes usados

- ODROID C1/C1+
- Placa controlador Adafruit PN532 NFC/RFID
- <http://bit.ly/1O0wK30>
- Pulsera Adafruit RFID de 13,56 MHz/NFC
- <http://bit.ly/1pP4Q3G>
- Anillo inteligente Adafruit RFID

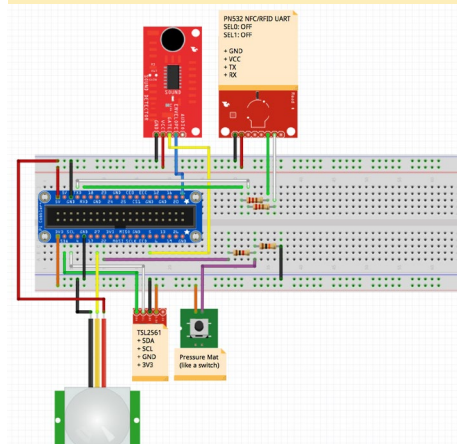
/NFC

- <http://bit.ly/1WXBrsi>
- Sensor Adafruit TSL2561 Digital Luminosity/Lux/Light
- <http://bit.ly/1rFj0WX>
- Sensor de movimiento PIR (Ebay)
- Detector de sonido Sparkfun
- <http://bit.ly/1HZN6Y1>
- Almohadilla de Presión PM1 (Ebay)

### Conexiones de hardware

El ODROID C1 tiene un conector de 40 pines que en su mayor parte es compatible con el conector de 40 pines de la Raspberry Pi (excepto las 2 entradas analógicas), de modo que vamos a utilizar un Pi Cobbler+ con los esquemas que incluye este artículo. Existen 2 railes de alimentación en la placa, uno para 5V (cables rojos) y otro de 3,3V (cables

Figura I - Primer paso de la instalación





aranja). También hay 2 railes para la toma a tierra (cables negros) a cada lado de la placa. Vamos a utilizar algunos programas de prueba para cada sensor escritos en Python y la librería WiringPi2, disponibles para ODROID en <http://bit.ly/23Rwf7T>. El mapeo GPIO de WiringPi para el C1 está disponible en <http://bit.ly/1Ejubsm>. La figura 1 muestra el circuito utilizado.

También es posible utilizar los componentes del kit Tinkering C1 de Hardkernel (<http://bit.ly/1YNPN6k>). Si utilizamos este kit, tenemos que prestar atención a la asignación de pines en el PCB TBreakout y modificar la información correspondiente para que coincida con los cambios.

### Sensor de movimiento PIR

El sensor de movimiento PIR tiene 3 pines: VCC, que va en el raíl de 5V, GND y DATA (cable amarillo). Vamos a utilizar el cabezal #13 para el pin DATA, exportado por WiringPi como GPIO 2. La entrada de datos desde el sensor PIR será 1 cuando haya movimiento y 0 en caso contrario. El sensor tiene 2 potenciómetros, uno para la sensibilidad y

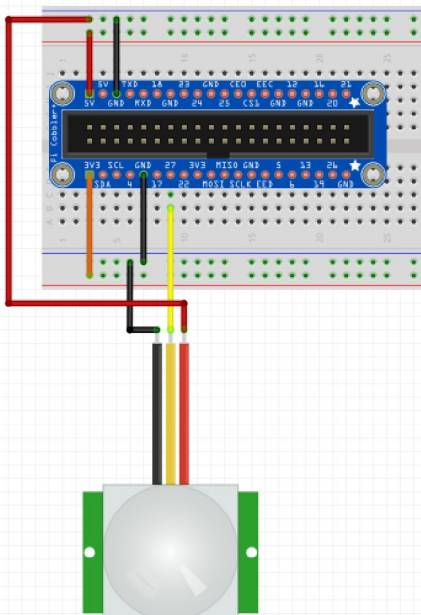


Figura 2 - Segundo paso de la instalación

otro para el tiempo (el tiempo tras el cual es detectado el movimiento - la salida de DATA será 1). Usando el método de en-

sayo y error, ajusté el potenciómetro para enviar 1 para una duración de detección de 10 segundos. La Figura 2 muestra la sección correspondiente del circuito.

El programa de pruebas utilizado es un simple script Python que sondea la entrada GPIO cada segundo:

```
import wiringpi2 as wpi
import time
wpi.wiringPiSetup()
# GPIO pin setup
wpi.pinMode(2, wpi.GPIO.INPUT)
while True:
    i=wpi.digitalRead(2)
    if i==0:
        print "no motion ", i
    elif i==1:
        print "motion detected ", i
        time.sleep(1)
```

Ejecutando el programa y moviéndose cerca del sensor debería aparecer un resultado similar al siguiente:

```
# python pir.py
no motion 0
no motion 0
no motion 0
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
motion detected 1
no motion 0
no motion 0
```

### Almohadilla de presión

La Almohadilla de presión actúa como un interruptor: cuando se presiona, el interruptor se enciende. Hay 2 cables a conectar, - uno a 3,3V (cable naranja) y el otro al pin de entrada #15 GPIO (GPIO 3 WiringPi, cable morado). Utilizaremos una resistencia de 10k para conectar el cable de datos a la toma

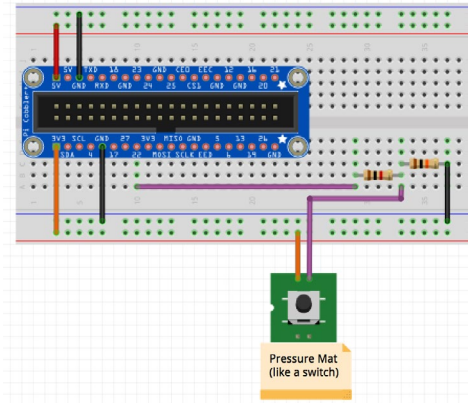


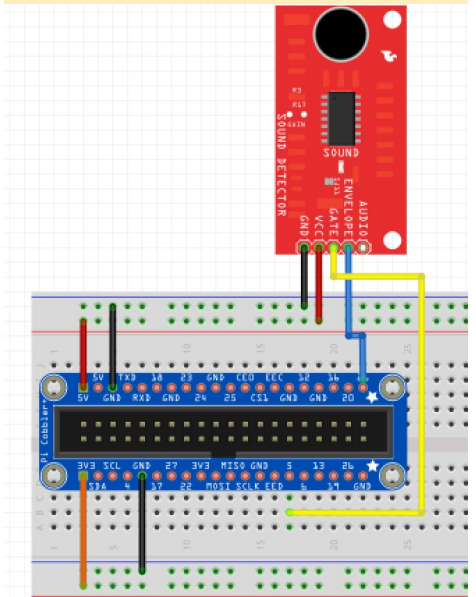
Figura 3 - Tercer paso de la instalación

de tierra (cable negro), lo que significa que cuando el interruptor este abierto habrá acceso a tierra y el GPIO leerá 0. Cuando se presione la alfombrilla, el pin conectado a 3.3V abrirá una ruta de menor a mayor resistencia y el GPIO leerá 1. Vamos a utilizar también una resistencia limitadora de corriente 1k en el cable de datos, para asegurarnos de que la placa controla la corriente cuando el interruptor este en ON. La Figura 3 muestra esta configuración.

El programa de pruebas utilizado aquí también es muy simple:

```
import wiringpi2 as wpi
import time
wpi.wiringPiSetup()
# GPIO pin setup
wpi.pinMode(3, 0)
```

Figura 4 - Cuarto paso de la instalación



```
while True:
    i=wpi.digitalRead(3)
    if i==0:
        print "not pressed ", i
    elif i==1:
        print "pressed ", i
        time.sleep(1)
```

## Sensor de sonido

El detector de sonido Sparkfun que vamos a utilizar tiene 5 pines VCC (5V), GND, GATE, ENVELOPE y AUDIO. Usaremos la salida digital GATE, que será 1 cuando se detecte sonido y 0 en caso contrario, y la salida analógica ENVELOPE que representará la amplitud del sonido. Conectaremos el pin GATE al cabezal #29 (WiringPi GPIO 21, utilizando el cable amarillo) y el pin ENVELOPE a ADC.AIN0 en el cabezal #40 (cable azul). La Figura 4 muestra esta parte de la configuración

Puedes utilizar el siguiente script para probar la detección del sonido:

```
import wiringpi2 as wpi
import time
wpi.wiringPiSetup()
# GPIO pin setup
wpi.pinMode(21, 0)
while True:
    i=wpi.digitalRead(21)
    if i==0:
        print "no sound ", i
    elif i==1:
        print "sound detected ", i
        time.sleep(1)
```

El volumen del sonido se puede determinar por el método ensayo y error. En mi caso, he reducido la salida a los valores 0-127 y utilicé los niveles 10-30 como volumen de sonido moderado (a nivel de conversación). Una nivel por debajo de 10 significa que la habitación está tranquila y un valor por encima de 30 indica que hay sonido con cierto volumen, con una alta probabilidad de que el bebé esté llorando.

La siguiente es otra aplicación de prueba, que se debe guardar en un ar-

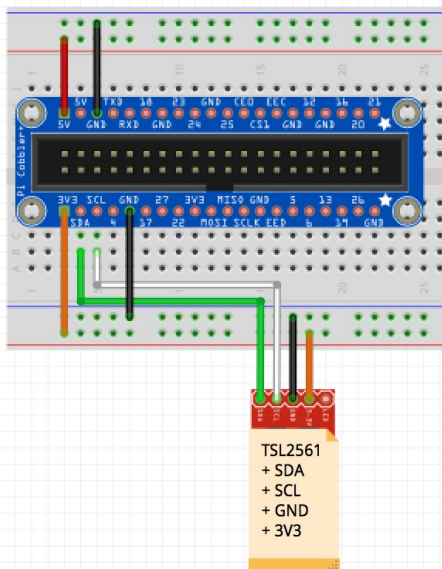
chivo llamado sound-env.py:

```
import wiringpi2 as wpi
import time
wpi.wiringPiSetup()
while True:
    i=wpi.analogRead(0)
    ampl = i*255/2047 # not
    sure this is correct
    if ampl <= 10:
        print "quiet: ", ampl, ",
        ", i
    elif ampl <= 30:
        print "moderate: ", ampl,
        ", ", i
    else:
        print "loud: ", ampl, ", ",
        i
        time.sleep(0.5)
```

Cuando se ejecuta, produce un resultado similar al siguiente:

```
# python sound-env.py
quiet: 10 , 88
quiet: 3 , 31
moderate: 27 , 220
moderate: 24 , 199
moderate: 28 , 229
moderate: 16 , 130
loud: 52 , 419
moderate: 16 , 129
moderate: 30 , 244
```

Figura 5 - Quinto paso de la instalación



```
quiet: 4 , 33
quiet: 6 , 49
quiet: 3 , 31
quiet: 5 , 44
```

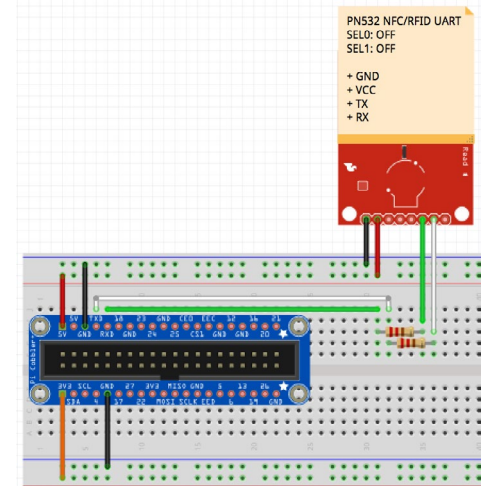
## Sensor de luz

El Adafruit TSL2561 se puede conectar a la placa ODROID a través de I2C. Las conexiones son simples: 3V3 al raíl 3,3V (cable naranja), GND a tierra (cable negro), SDA a SDA (cable verde) y SCL a SCL (cable blanco). La figura 5 muestra esta parte de la configuración.

Para activar I2C en el ODROID C1, es necesario cargar el módulo del kernel aml\_i2c. Esto se puede hacer añadiendo aml\_i2c al fichero /etc/modules. También vamos a utilizar el módulo python tentacle\_pi, disponible en <http://bit.ly/1AiDZfk>. Este permite la comunicación con TSL2561 desde una aplicación Python enviando la intensidad de la luz en unidades lux:

```
from tentacle_pi.TSL2561 import
TSL2561
import time
tsl = TSL2561(0x39, "/dev/i2c-1")
tsl.enable_autogain()
tsl.set_time(0x00)
while True:
    print "lux %s" % tsl.lux()
    time.sleep(1)
```

Figura 6 - Sexto paso de la instalación





## Lector RFID

El lector de tarjetas RFID se puede conectar al ODROID C1 usando la interfaz UART, SPI o I2C. Vamos a utilizar la librería Linux nfc, que es la opción más sencilla para utilizar la conexión en serie UART. Para esto, tanto SEL0 como SEL1 en la placa PN532 RFID necesitan estar fijado en OFF. Al igual que con cualquier conexión en serie, vamos a utilizar 5 cables: VCC (5V, cable rojo), tierra (cable negro), RX y TX (verde y blanco) - conectado a TX y RX de UART1 (/dev/ttyS2) en la placa sobre los cabezales #8 y #10. La Figura 6 indica esta parte de la instalación.

La librería libnfc y los componentes relacionados se pueden instalar fácilmente en Ubuntu con el siguiente comando:

```
$ sudo apt-get install \
  libnfc-bin libnfc-examples \
  libnfc-pn53x-examples
```

El empaquetador python se puede descargar e instalar desde <http://nfcpy.org>. Para utilizar libnfc, es necesario configurar la conexión mediante la creación de un archivo llamado /etc/nfc/devices.d/pn532\_uart.conf que contiene el siguiente código:

```
## Typical configuration file for
PN532 device connected using UART
name = "PN532 board via UART"
connstring = pn532_uart:/dev/
ttyS2
allow_intrusive_scan = true
```

A continuación, puedes probar la conexión con el programa pn53x-diagnose:

```
$ sudo pn53x-diagnose
pn53x-diagnose uses libnfc 1.7.0
NFC device [pn532_uart:/dev/
ttyS2] opened.
Communication line test: OK
ROM test: OK
RAM test: OK
```

La aplicación nfcpy está bien documentada y es fácil de usar. Para leer una etiqueta, sólo necesita unas cuantas líneas de python. Ten en cuenta la cadena de conexión tty:S2:pn532 que es necesario utilizar.

```
>>> def connected(tag):
print(tag); return False
...
>>> clf = nfc.ContactlessFrontend
('tty:S2:pn532')
>>> clf.connect(rdwr={'on-connect':
connected}) # now touch a
tag
Type3Tag IDm=01010501b00ac30b
PMm=03014b024f4993ff SYS=12fc
<nfc.tag.tt3.Type3Tag object at
0x7f9e8302bfd0>
```

Puesto que la función de conexión esta bloqueada, el código final será escrito para leer la etiqueta NFC en un hilo separado.

En la Parte 2 de esta serie, disponible en el próximo número, detallaré los componentes de software necesarios.



## Referencias

- ODROID-C Tinkering Kit
- <http://bit.ly/1YNPN6k>
- Repositorio GitHub WiringPi2 de Hardkernel
- <http://bit.ly/23Rwf7T>
- ODROID-C2
- <http://bit.ly/1oTJBya>
- Drivers Python para los sensores TWI/I2C
- <http://bit.ly/1AiDZfk>



# ODROID Magazine está en Reddit!



## ODROID Talk Subreddit

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



# MONTA TU PROPIO COCHE INTELIGENTE CON EL ODROID-XU4

## INCORPORA UNA NUEVA FUNCIONALIDAD A TU VEHICULO

por Jon Westgate

Soy un ingeniero de electrónica que trabaja además como ingeniero de sistemas informáticos, y estoy certificado en el Reino Unido por la asociación de servicios electrónicos en el nivel tres. Me he sentido atraído por la electrónica desde que mi padre montara un televisor en color desde cero en su tiempo libre. Este proyecto pretende llevar el potencial de un ODROID-XU4 a tu coche añadiéndole más funcionalidades y características. Esta es mi aventura de empezar un proyecto y conseguir que todo funcione correctamente.

Siempre he querido construir un PC para usarlo en mi coche. Es algo que he visto con mucho potencial desde que los MP3s se hicieron digitales y pueden reproducirse desde un PC, llevando la posibilidad de escuchar música y otras características a mi coche. He hecho pruebas en el pasado, pero la mayoría de los dispositivos simplemente no cumplían con mis necesidades. Las máquinas basadas en x86 que utilicé eran demasiado lentas, y el ordenador Android que probé no podía utilizar una pantalla táctil o el GPS para lo que quería. También lo intenté con un Nook HD, que parecía prometedor, pero su batería me dio un montón de problemas.

### Ensayo y error

Todas estas cuestiones me llevaron a probar este proyecto con un ODROID. Mi primer ODROID fue el U3, y me lo compré con el objeto de no tener que hacer demasiado para lograr que todo

funcionase. La pantalla táctil china de 7 pulgadas que encontré tenían una resolución de 1280x800, y el ODROID-U3 sólo podía manejar una resolución de 1920x1080 o de 1280x720, creando una imagen distorsionada. También tuve algunos problemas con la pantalla táctil cuando intentaba conseguir posiciones precisas con el controlador táctil. Supuso un reto aún mayor hacerlo con Android, donde lo primero que esperas es una pantalla capacitiva.

Necesitaba algo que simplemente funcionase con esta pantalla táctil, de modo que investigué una alternativa, incluyendo el XU3. El XU3 parecía encajar muy bien por su mayor compatibilidad para resoluciones y otras características, pero era muy caro. Pero cuando el ODROID-XU4 apareció a un precio mejor, pude finalmente costearme un dispositivo que cubriera mis necesidades.

### Buscando un dispositivo visual

Tras recibir mi ODROID-XU4 dos semanas más tarde, seguí adelante y empecé con una imagen de Android sobre mi eMMC. Afortunadamente Android arranco y la pantalla táctil llegó a funcionar en el acto. Sin embargo, todavía no estaba contento con la calidad de la imagen y la pantalla táctil, de modo que decidí ver qué más había por ahí. Mientras tanto, descubrí la pantalla táctil de 7 pulgadas que ofrecía Chalkboard Electronics. Me proporcionaba la experiencia multi-táctil capacitiva que necesitaba

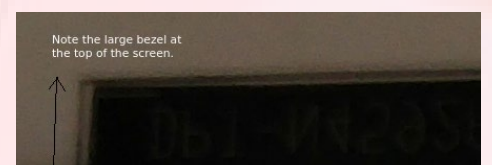
para Android, además de una mejor calidad de imagen en general.

Cuando la pantalla táctil llegó, me di cuenta que el marco de la pantalla táctil de 7 pulgadas de Chalkboard era muy grande. Además hay que tener en cuenta que está invertida, con las conexiones en la parte superior, en lugar de la parte inferior como era de esperar en otras pantallas. Esto es algo a tener muy presente a la hora de diseñar tu propio proyecto



Car PC, ya que necesitarás aproximadamente 7 mm de espacio en la parte superior cuando montes la pantalla.

Me di cuenta de que el firmware que viene con esta pantalla táctil emula un puntero de ratón, en lugar de una experiencia táctil que cabría esperar de una tablet o dispositivo Android similar. De esta forma el dispositivo registra un salto cuando deslizas el dedo, más que mover un cursor de ratón sobre la pantalla. Existe otro firmware diferente que necesitaremos grabar en esta pantalla táctil para conseguir que funcione, según la documentación que he leído.





## Actualizar el firmware de la pantalla táctil

Para este proyecto con mi ODROIDXU4, utilicé CyanogenMod 12 de @voodik, con Android 5.1.1 (<http://bit.ly/1VyXFKW>). Lo mejor es empezar con un sistema pre-configurado e ir añadiendo características a medida que vas consiguiendo que el hardware básico funcione. Tras encontrar el driver correcto para actualizar el firmware, necesitas utilizar un PC con Windows para actualizar el firmware de la pantalla táctil y así poder trabajar bien con este proyecto. Puedes encontrar instrucciones detalladas de este proceso en el sitio web de Chalkboard Electronics en <http://bit.ly/1MULs0y>. Una vez que termines con la actualización del firmware, deberías ser capaz de usar la función de zoom y otras funciones intuitivas que haran que este proyecto sea fácil de instalar y usar en tu coche.

## Conseguir que el GPS funcione

Para que este proyecto tuviera éxito, tenía que conseguir también que funcionara el GPS para la navegación por satélite. Esto era algo que aún no tenía en mi Citroen C4 Coupe. Intenté sin éxito incluir la navegación por satélite usando otros métodos de terceros, la tecnología GPS del fabricante era demasiado cara y no valía la pena instalarla en mi coche. Antes de empezar con este proyecto, tenía entre otros proyectos, conseguir un dispositivo WinCE que funcionase con una unidad GPS de 7 pulgadas junto con un sistema de entretenimiento Bluetooth conectado todo a un cuadro de mandos. Pero tras varios fallos, errores y un bajo rendimiento en general, decidí que ya era hora de pasar a mi nuevo XU4 con su potente procesador de 8 núcleos y el sistema operativo Android.

Cuando cambié al XU4, arrojé a la basura la vieja pantalla china integrada en el dispositivo WinCE, en su lugar utilicé el marco y el cuadro de mandos para encajarlos en mi nueva pantalla táctil

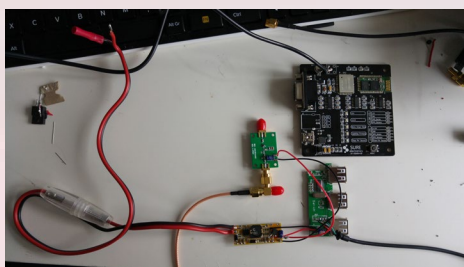
de alta calidad Chalkboard. También utilicé un Denison Gateway Pro para programar los botones y funciones de mi coche con la pantalla a medida y XU4 instalados. Esto me proporcionó una conexión Bluetooth para el manos libres y poder escuchar música usando los altavoces del coche.

En lugar de utilizar un módulo GPS ODROID, acabé usando un módulo GPS chino sin marca, conectado a una antena 4-en-1 para ahorrar espacio y



facilitar las cosas. Todo esto estaba conectado a mi XU4 con un adaptador serie a USB. El objetivo era utilizar una única antena con la que poder manejar no sólo mi GPS, sino también mi conexión de radio DAB, AM, FM, a través de este conector de antena 4-en-1. Terminé colocando la antena cerca del espejo retrovisor después de experimentar con varios sistemas de antenas, incluyendo una potente antena GPS que no llegó a funcionar con mi configuración actual.

El software GPS que solía usar en WinCE se llamaba iGO Primo. Pero ahora que estoy usando Android, todo se puede manejar usando una app de



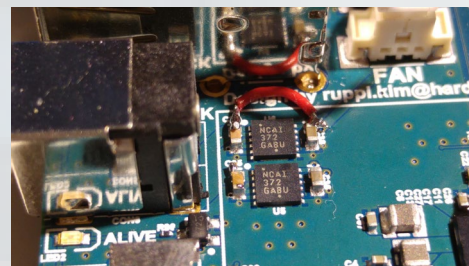
Play Store, que incluye numerosas aplicaciones GPS offline, así como también Google Maps. Sin embargo, había una última cuestión con el GPS, asegurar que el reloj de tiempo real pudiera sincronizarse con los satélites GPS para la precisión horaria, lo cual requería de una batería

RTC, así la hora no se perdería cada vez que el dispositivo se desconectara de la alimentación. Contaba con una batería de repuesto y simplemente hice algo muy similar a lo que Hardkernel ofrece en su página web. También descubrí una forma curiosa de apagar el XU4 a través del conector PIO de la placa.

## Un percance menor

Mientras trabajaba en conseguir que el conector PIO pudiera apagar correctamente mi XU4, terminé accidentalmente ocasionando un cortocircuito en la línea de 5V, dejando inservible mi XU4 y con el LED parpadeando. Por si alguien se lo está preguntando, fui capaz de solucionarlo provocando un cortocircuito en los circuitos de protección (ICs). Si alguna vez cometes este error, puede hacer esto para intentar resucitar a tu XU4.

Este percance provocó que arruinase mi circuito RTC, de modo que la batería quedaba inservible. Tras realizar



una búsqueda en Plays Store, encontré una aplicación llamada "Smart Time Sync" (<http://bit.ly/1rw76yA>), que usa el GPS para fijar la hora y ajustar el reloj de Android después de cada arranque. Puede ser útil si no quieres usar una batería, o no puedes por haber tenido un tropiezo como el mío. Esta app también es muy útil para aquellos interesados en que su XU4 funcione con un GPS en Android.

## Lograr que la radio funcione

¿Qué es un coche sin radio? Puesto que el objetivo es crear una única experiencia desde mi XU4, también quería lograr que funcionase la radio con ayuda

de Software Defined Radio (SDR). El RTL-SDR es un gadget de 25\$, que te permite coger un amplio rango de señales de radio, incluyendo estaciones de radio FM con DAB que permite mostrar información de la estación.

Sin embargo, la app DAB que es-



taba utilizando necesitaba una conexión a Internet para comprobar su licencia, aún cuando no necesita Internet para funcionar por sí sola. Esto requería un poco de creatividad y otro gadget para conseguir que todo funcionase: una conexión de datos móvil. Lo bueno de esto es que Google Maps funcionaría también, aunque suponía mucho trabajo. Finalmente me decante por un dongle 3G/4G Huawei que es compatible con mi compañía de telefonía británica local, Everything Everywhere (EE).

El Dongle que elegí era uno que se



conectaba vía Wi-Fi a mi XU4. Podría diseñarse para conectarse de forma nativa a través de Android, pero esto significaba una cosa menos que cablear, ya que podía hacerse bien desde la toma de 12V del coche. Terminé mejorando la antena para potenciar la conectividad.

## Poniéndolo todo junto

A medida que continuaba realizando



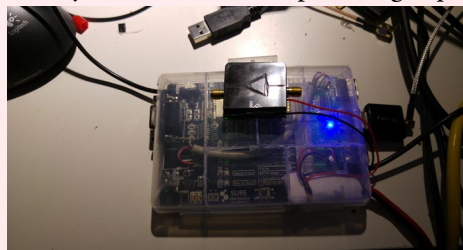
pruebas, todo empezaba lentamente a encajar. Tenía un XU4 funcionando con Android, una pantalla táctil, conectividad GPS, radio FM con DAB y acceso a Internet. Con todos estos dispositivos tan diferentes conectados, necesitaba un Hub con alimentación para maximizar la potencia disponible al XU4 sin que afectara a las demandas de energía de los periféricos.

Como el espacio en el panel de instrumentos del coche está limitado, intente minimizar el espacio que ocupa todos los



dispositivos. Tuve especial cuidado a la hora soldar y cortar, para no estropear el módulo de alimentación, el XU4 y los periféricos aglutinados en un único dispositivo compacto, ocupando así mucho menos espacio que cuando todo estaba conectado para realizar pruebas.

Dado que el RTL-SDR que he comprado incluye una modificación de muestreo directo, es capaz soportar radio AM y otras frecuencias. En total, el dispositivo puede cubrir señales entre 100 KHz y 1,8 GHz, un amplio rango que



incluye radio AM y FM, radio CB e incluso señales de radio privadas, como la de la policía y los bomberos. Disponer de escáner tan amplio resulta muy útil si alguna vez quiero localizar una señal específica cercana.

## Conseguir que todo entre en el coche

Ahora que todo funciona y está compactado, era hora de configurarlo y dejarlo listo para instalarlo en mi coche. Esto requiere la instalación de la antena 4-en-1, así como colocar el cableado adecuado entre mi nuevo panel de instrumentos y los propios sistemas de alimentación de mi coche. Nunca me cansare lo suficiente de decir lo importante que es usar fusibles para evitar un mal funcionamiento eléctrico que puede causar daños a tus equipos electrónicos o peor aún, a un amigo o a los miembros de tu familia. También es importante la correcta conexión a tierra de tus equipos electrónicos, ya que limitas una conexión DC de 12V a una de 5V que es la que necesita tu XU4 y los periféricos. Incluso a estas bajas potencias, puede ser muy peligroso un fallo eléctrico con estos dispositivos electrónicos. Hay que aportar la potencia que realmente se necesita.

Puedes hacerlo todo a través de USB, pero en mi caso fue más fácil suministrar la energía desde la fuente de alimentación de 12V a un componente CPT que pasa la conexión de 12V a una conexión de 5V 10A. Luego todo se transmite a través de una conexión directa al hub USB, y finalmente el ODROID-XU4. Esto te ahorra el paso redundante de utilizar un convertidor u otro adaptador AC cuando todo termina en el adaptador DC y además, facilita la alimentación de la antena mejorando la señal. Los componentes CPT son muy útiles,

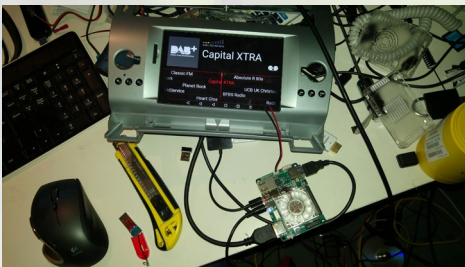




ya que tienen protecciones integradas para prevenir cortocircuitos.

En este momento, todo se inicia cuando arrancamos el coche. Esta es la forma en la que todo está configurado actualmente, aunque espero que cambiar esto con un temporizador para los casos en los que apague rápidamente el coche para hacer una parada o repostar combustible. El USB será la clave, aunque me llevará algún tiempo entender todo esto y efectivamente controlar la forma en que el dispositivo recibe alimentación de la batería incluso cuando el coche no esté arrancado.

Mientras tanto, instale cuidadosamente todos los componentes en el panel de instrumentos de mi coche, asegurándome de que funcionaba todos los bo-



tones y componentes una vez instalado.

## Conseguir que el audio funcione

Lograr que el audio funcionara en esta nueva configuración Car PC con mi XU4 fue mucho más difícil de lo que se esperaba. Había un montón de ruido a causa de la conexión a tierra utilizando las salidas normales por defecto. La solución pasaba por utilizar un extractor de audio HDMI para evitar todo el ruido que de otro modo tendría que oír al reproducir música o escuchar la radio.

Llegue a probar diferentes opciones, incluyendo un controlador de audio digital USB en lugar de un extractor de audio HDMI, pero el objetivo era que este nuevo Car PC pudiera soportar cualquier dispositivo de audio, como un iPod, una llamada con el manos libres de mi smartphone o el sistema de navegación GPS. Todo el proceso sigue estando en desarrollo con planes de comprar una mejor DAC con un amplificador.

También hay un desajuste de impedancia en la clavija de audio de mi Denison Gateway Pro, provocando una especie de sonido metálico muy fino. Continuare investigando esta cuestión a medida que vaya retocando mi Car PC.

## Otros cambios

Hay una serie de pequeñas modificaciones que he llevado a cabo para poner a punto este proyecto. Esto incluye el uso de un Dremel para perforar la pantalla táctil Chalkboard, de manera que encajara en el marco y en el salpicadero del coche. También he añadido un nuevo conector al XU4 para poder añadir en el futuro un botón de encendido usando controles externos sobre una placa de circuitos. El objetivo es tener soporte para futuros joysticks USB y otras conexiones sin tener que abrirlo todo de nuevo.

Para este proyecto, terminé usando una tarjeta SD de gran tamaño para disponer de una mayor y más barata capacidad de almacenamiento. Esto necesito una alargadera debido al limitado espacio entre el marco y al conector HDMI que bloqueaba el lugar donde tenía que ir mi SD.

Otra idea innovadora que tenía en mente era utilizar una cámara trasera. El sistema GPS chino que utilicé soportaba una cámara trasera, aunque no había manera de conectarla directamente a mi pantalla táctil Chalkboard. La cámara funciona detectando una señal de vídeo compuesto y reenviándola a la pantalla, la cual se podía convertir a la entrada HDMI de Chalkboard. El único problema es que la señal compuesta se veía realmente mal y tenía una relación de aspecto muy pobre sobre la pantalla.

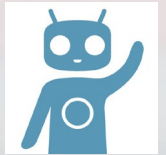
La solución era doble. En primer lugar, era necesario instalar un convertidor para llevar la señal compuesta hasta una señal HDMI adecuada y sin problemas de relación de aspecto. Sin embargo, esto añadiría otro periférico a tu coche. También tendrá que buscar una cámara que sea lo suficientemente buena para conseguir una señal de calidad HD

que mejore la visibilidad. Recomiendo usar una cámara SDI-HD, que convertirá la señal HD compuesta del cable coaxial al convertidor, que a su vez se conecta a la pantalla táctil Chalkboard. Esto te permitirá obtener resoluciones hasta UHD 4K con un único cable coaxial con conectores BNC, junto con un enchufe de alimentación DC de 12V, y no tendrás problemas de relación de aspecto. Esto continúa estando en desarrollo, ya que estoy actualmente montando un circuito que detectará la marcha atrás del coche y hará que cambie la pantalla Chalkboard del XU4 automáticamente.

## Software

Lista de todo el software usado:

- CyanogenMod 12.1 de Voodik que ejecuta Android 5.1.1
- App de Google incluyendo el Google Play Store
- Smart Time Sync Pro
- Google Chrome, Firefox no funciona bien en el XU4 y necesita ser activado con el botón de encendido
- iGo GPS, aunque existen muchas aplicaciones GPS offline
- Autorun (FIT009C Soft Tools)
- K@mail, recibirás un error de certificado de correo electrónico que debe pasar por alto, debido a Smart Time Sync Pro
- Google Maps/Google Streetview



Para aquellos que lo necesiten, aquí tenéis los cambios que realicé en mi archivo build.prop:

```
config.disable_bluetooth=false
wlan.modname=8192cu
ro.kernel.android.gps=tttyUSB0
ro.kernel.android.gps.speed=9600
```

Gracias por acompañarme en mi viaje para montar un Car PC, ¡Y diviértete experimentado!

# CONOCIENDO UN ODROIDIAN

## MILTADIS MELISSAS

editado por Rob Roy

*Por favor, h́ablanos un poco sobre ti*

Mi historia personal comienza en 1965 en el norte de Grecia y en la hermosa ciudad de Xanthi. A los 18 años, me trasladé a Atenas para cursar mis estudios académicos, y 5 años más tarde me decante por la isla de Corfú, donde trabajo y me desarrollo.

Conseguí mi licenciatura como ingeniero en electrónica en 1988. Después, estuve trabajando casi 5 años con Barclays como operador de sistemas Unix online y luego otros 2 años y medio con Scotiabank en un puesto similar. En 1997, abandone este campo y me convertí en un empleado del estado especializado en tecnología de enseñanza secundaria, una posición que aún sustento hoy día. En 2006, finalice mi Máster en Tecnología de la Información y en 2012, me ascendieron a Sub-director de la escuela secundaria donde enseño Tecnología y Electrónica. Mi Máster sobre Educación está pendiente, y mis intereses académicos se centran en el campo ontológico de la información.

Mi carrera se ha ido desarrollado junto a mi vida personal. Me casé con mi encantadora esposa Dori en 1995, su continuo apoyo y el estímulo es el que ha hecho posible todas las cosas en nuestras vidas. Mi hijo Apostolos fue aceptado el año pasado en la Universidad de Glyndwr en Gales, para especializarse en el desarrollo de juegos: “De tal palo, tal astilla”. Su sueño es trabajar con grandes nombres de la industria. El último miembro de nuestra familia, pero no menos importante es mi hija Eri que está estudiando educación infantil en el instituto.

*¿Cómo fueron tus inicios con los ordenadores?*

En 1985, compré mi primer “ordenador personal”: el legendario Commodore 64. Con 64K de memoria y un precio inferior a los 600\$, Commodore 64 fue realmente un gran descubrimiento por aquel entonces. Esta máquina de 8 bits era capaz de dibujar gráficos fantásticos usando sprites, y supero al resto de equipos del mercado con su circuito de tarjeta de sonido de 3 canales. Me viene a la mente un sinfín de momentos con los que disfrutaba con juegos como “Head over Heels” y el clásico insuperable “Pacman”. Pero el verdadero potencial de Commodore 64 estaba en su facilidad de programación. Dominar BASIC se convirtió en un reto de la década de 1980. Algunos de mis primeros intentos de programación fueron una libreta de direcciones, una agenda y una base de datos de recetas. Commodore 64 era en realidad una escuela en y por sí mismo. Adquirí todos mis conocimientos de programación estructurada con esta máquina, e incluso llegue aún más lejos con algunos conocimientos básicos de lenguaje ensamblador. En los años



**Miltadis, conocido como Miltos, ha estado programando y usando los ordenadores desde los primeros días del Commodore 64**

posteriores, fui observando la evolución de la línea de ordenadores Intel con Windows, y fue en 2006 cuando cambié a Linux. La principal razón del cambio fue la sensación de libertad que la programación de código abierto te ofrece, junto con la potente comunidad que existe tras Linux. Además, Linux es un desafío para todo ingeniero de electrónica, ya que te puedes encontrar con muchos dispositivos aparte de los ordenadores, tales como descodificadores, electrodomésticos, módems, teléfonos, máquinas de juegos y sistemas embebidos. Fue Linux quien me llevó a la línea de productos de Hardkernel.

*¿Qué te atrajo a la plataforma ODROID?*

Siempre dispuesto a aprender, encontré por casualidad un manual online de los creadores de makeuseof.com (<http://bit.ly/SInllr>) en el cual se abordaba la Raspberry Pi. Al final del libro, el autor proponía algunas alternativas, y el U3 de Hardkernel era, por supuesto, el primero de la lista. Contaba con mejores especificaciones de hardware, más implementaciones de software, y podías ejecutar diversas versiones de escritorio y de servidor Linux, incluyendo Android, OpenVault, y Media Center. Para mí, fue “amor a primera vista”. Encargué un U3 casi inmediatamente, y tras seis meses de uso compré otro. Como ingeniero en electrónica me impresionó la calidad de la placa y su durabilidad por no hablar de su reducido tamaño y bajo consumo de energía.

Ese fue el momento en el que las excelentes imágenes de Rob Roy hicieron su entrada. Disfruté con “Fully Loaded: Ubuntu 12.11”, “Quiet Giant: Ubuntu Server 13.05”, y “Pocket Rocket: Android Jelly Bean 4.1.2”. Descubrí la potente comunidad (los ODROIDians) cuando empecé a desarrollar mis imágenes y compilar algunos drivers específicos. ¡Sin su ayuda no habría llegado tan lejos! El siguiente que compre fue el ODROID-C1 y luego adquirí 2 unidades del XU4, que es mi favorito.

*¿Cómo utilizas tus ODROIDS?*

Tengo OpenELEC funcionando perfectamente en mi ODROID-U3, el cual me permite escuchar música y ver vídeos



por streaming, así como ver los canales de YouTube, películas, canales de TV en vivo y programas de todo el mundo. Por otro lado, mi ODROID-C1 se encuentra en mi escritorio ejecutando “Pocket Rocket II: Android 4.4.4 KitKat” con todos los extras que este sistema operativo nos ofrece. El XU4 es mi PC de escritorio principal y francamente, han pasado varios meses desde que abrí por última vez mi PC Windows. El XU4 satisface la mayoría de, si no todas, mis necesidades como estación de trabajo e incluso puedo ejecutar juegos con la excelente imagen GameStation Turbo proporcionada por @meveric, otro ODROIDian. El emulador PPSSPP funciona muy bien en esta imagen, y la mayoría de los juegos de PSP se ejecuta a 60 fps sin problema.

Por último, he creado mi propio entorno de escritorio basado en Arch Linux, e instalé el escritorio Mate como interfaz gráfica de usuario. No hay duda que detrás de este excelente entorno de escritorio hay mucho trabajo realizado. Sorprendentemente, esta imagen no tiene el error de la pantalla en negro y el error de los iconos de la suite LibreOffice, todo parece funcionar a la perfección. Con suerte, pronto publicaré una guía en el foro para todo aquel que desee tener el mismo tipo de entorno. El ODROID-XU4 es una verdadera joya, y si estos usos no son suficientes, mi segundo XU4 lo utilizo como servidor. Las posibilidades son infinitas.

*¿Qué innovaciones te gustaría en futuros productos Hardkernel?*

No debería ser difícil para Hardkernel incluir un circuito wifi y bluetooth en su próxima placa de desarrollo. Por otro lado, creo que existe mucho margen para el desarrollo de software de las placas. Por supuesto, el sistema operativo Android está bastante pulido para la línea C de productos (es decir, C1, C1 + C2), así como el XU4, aunque el sistema operativo Linux es probable que evolucione en los próximos años para ofrecer más y mejores posibilidades. Más aún, el reciente desarrollo de sistemas operativos como “Remix OS” parece muy prometedor, aunque aún no está implementado en la línea de productos de la Hardkernel, y no sé si alguna vez se podrá instalar en los ODROIDS. Por último, algo de memoria (RAM) adicional podría añadirse a todas las placas, por supuesto con el inevitable coste adicional que supone. No creo que añadir 1 o 2 GB de RAM extra suponga un incremento del coste de 100 dólares, incluso para el ODROID más caro.

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

“Una mente sana en un cuerpo sano” es un dicho griego muy famoso y antiguo. Además de los ordenadores, me encanta nadar y caminar. Nado mucho, sobre todo tras haber tenido algunos problemas serios con mi espalda hace un año. El hecho de vivir en una isla me da la oportunidad de nadar casi todo el año. Suelo cubrir una distancia de unos 35 km por semana a pie y otros 8 nadando. También me motiva mi muñequera inteligente y el ODROID-C1, que ejecuta una aplicación para Android que me ayuda a hacer un seguimiento de mis actividades, recopilar datos y estadísticas, y ver información sobre mi salud y estado físico.

Además, me encanta leer libros de educación y filosofía, y ver películas en blanco y negro del viejo cine estadounidense. También suelo tomar fotos durante mis caminatas diarias por la ciudad. Me he redescubierto mi propia ciudad después de 25 años viviendo, con paseos y actividades que he ido realizando. También estoy pensando seriamente en comprarme una bicicleta.

*¿Qué consejo le daría a alguien que quiere aprender a programar?*

La programación es como aprender un nuevo idioma. Siempre lleva su tiempo. Es más, puede verse como una curva personal de aprendizaje, ya que cada uno aprendemos y comprendes el conocimiento de un modo diferente. Ha habido períodos en los que yo aprendía a pasos agigantados, y otros en los que avanzaba muy lentamente. Empecé con BASIC hace mucho tiempo en el Commodore 64. BASIC me ayudó a entender los fundamentos de la programación estructurada. Continué con Visual Basic en el año 2000 sumergiéndome en el conocimiento de la programación óptica, un lenguaje de programación que todavía uso con fines educativos. Hace unos diez años, realice el “gran salto” a la programación orientada a objetos usando Python. Me encanta Python porque es un lenguaje de programación educativo, versátil y de aplicación general, con una gran librería de código y una vasta comunidad.

Aconsejo empezar con la programación estructurada, y poco a poco pasar a la programación orientada a objetos. Python (<http://bit.ly/1oDM6iq>) es un lenguaje de programación que te permite trabajar de un modo rápido e integrar sistemas con mayor eficacia. Proporciona una sintaxis limpia y una estructura fácil de aprender. Animo a mis estudiantes de secundaria a que aprendan Python y a tomarlo como un conjunto básico de habilidades. El entorno de programación Python se puede instalar con facilidad en todos los equipos Linux ODROID, lo cual minimiza en gran medida el coste de implementación. Además, existen muchos cursos gratuitos en línea ofrecidos por diversos desarrolladores. Otros recursos online incluyen Coursera (<http://bit.ly/18HdJkD>), Edx (<http://bit.ly/1bxqyhn>), y Udacity (<http://bit.ly/1dy71Lz>). Como digo a mis alumnos, con ODROID, la World Wide Web y Python, el cielo es el límite y las posibilidades son infinitas.



**Miltos y su esposa, Dori, viven en una isla en Grecia**