Instrucción CPU X86 • Monitorización Redes Inalámbricas • Multiscope

ODROID Magazine

Año Tres
Num. #28
Abril 2016

El futuro da un paso al frente:

Realidad *Aumentada*

- Aprende a usar tu oCAM en increibles proyectos

- Un ODROID-W dentro de una Gameboy Advance SP
- Crea un armazón de madera para jugar con el CI+ y consigue la perfecta experiencia arcade en casa

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

Pollin ELECTRONIC

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

Dirección: Max-Pollin-Straße 1
85104 Pförring Alemania

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

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

EDITORIAL

OpenCV es una aplicación de software muy conocida que se puede utilizar junto con una cámara digital para analizar la información visual y tomar decisiones basadas en lo que el ordenador interpreta desde una imagen en vídeo. Hardkernel ha publicado recientemente el módulo oCAM, que es una forma económica de crear un sistema de tratamiento visual que responde a su entorno. Este mes, DongHyun Yoo nos muestra una simple aplicación de OpenCV que se puede utilizar para seguir la pista a objetos con un ODROID-XU4.

Varios usuarios del foro han diseñado recientemente consolas de juegos retro utilizando los ODROIDs y así poder recrear sus primeras experiencias con los juego, han decido compartir sus diseños con la comunidad. También presentamos algunos grandes juegos como Awkaster, un shooter 3D en primera persona basado en terminal, Cut the Rope 2, en el que tienes que resolver puzzles para alimentar con caramelos a un curioso monstruo, Re-Volt, un divertido juego de carreras y XMage, un paquete cliente/servidor de Magic: The Gathering. Venkat continúa su serie técnica con un proyecto MultiScope, y Adrian nos enseña cómo realizar pruebas de penetración en red mediante Kismet.

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 articulos, contacta con odroidmagazine@gmail.com, o visita http://bit.ly/1yplmXs.
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

NUESTRO PERSONAL DE AMAZING ODROIDIAN:

Rob Roy, Editor Jefe

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

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 especializa 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.nicolecscott.com.

James LeFevour, Editor Artístico

Soy un especialista en medios digitales que disfruta trabajando como freelance en marketing de redes sociales y administración de sitios web. Cuanto más aprendo sobre las posibilidades de ODROID más me ilusiona probar cosas nuevas con él. Me traslade a San Diego desde el Medio Oeste de los EE.UU. 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.

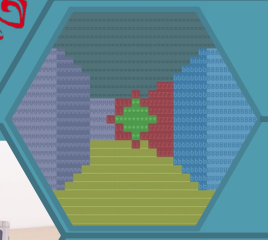
Venkat Bommakanti, Editor Adjunto

Soy un apasionado de los ordenadores desde la bahía de San Francisco en California. Procuro 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.

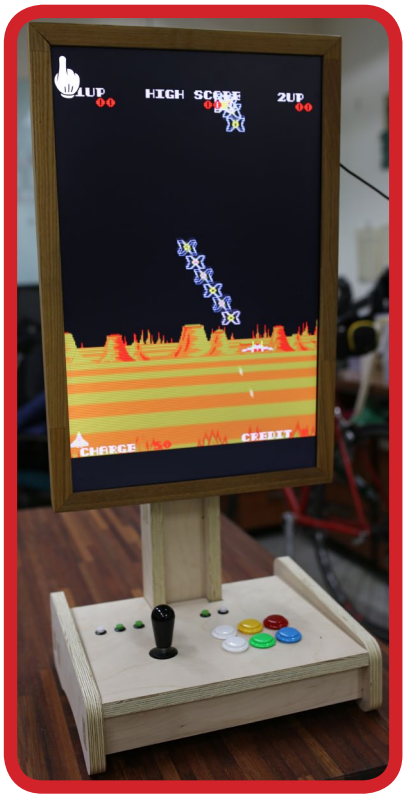
INDICE



ARMAZON DE MADERA PARA JUGAR

PARA LOS FANS RETRO

ediado por Justin Lee

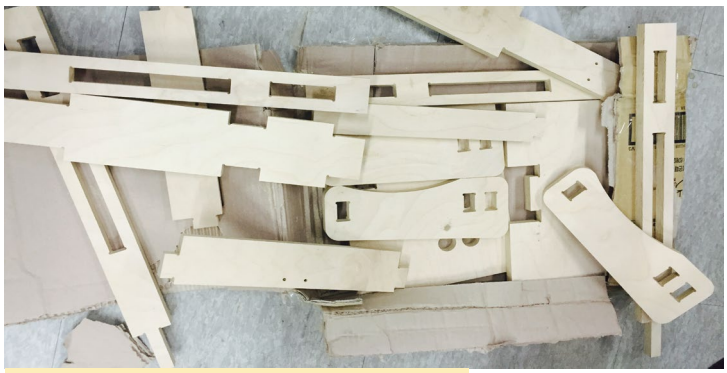


Pudiendo emular los juegos ya sea en horizontal o en vertical, una cosa que no serás capaz de conseguir es conseguir divertirte tanto sin esta consola. Garantizado.

Scott Shin, un artífice coreano, ha desarrollado una consola de juegos de madera ecológica utilizando un ODROID-C1+. Su mayor ventaja es que la consola se puede poner tanto en modo vertical como en horizontal, y así poder disfrutar de una mejor experiencia de juego. Vamos a ver cómo hizo esta gran consola usando simples elementos de diseño.

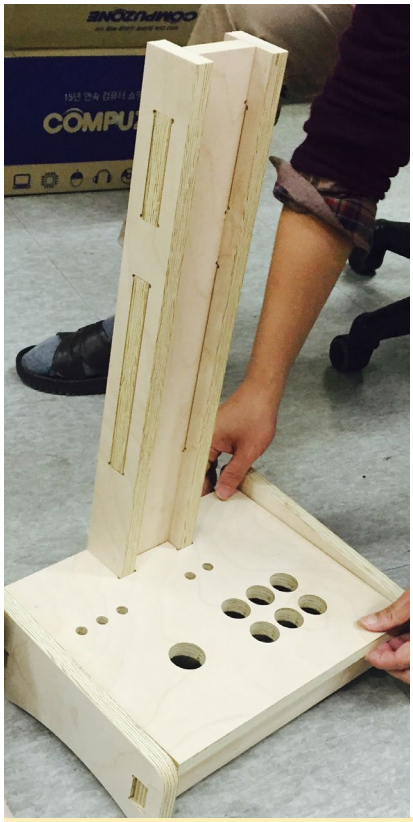
Estructura principal

La estructura principal ha sido diseñada con Google SketchUp para hacer los bloques de madera en forma de mecanizado CNC. Todo el material es de madera contrachapada con abedul de un espesor de 18 mm. Los archivos de diseño están disponibles en Github en http://bit.ly/1XDGtBp.



Piezas de madera del mecanizado CNC

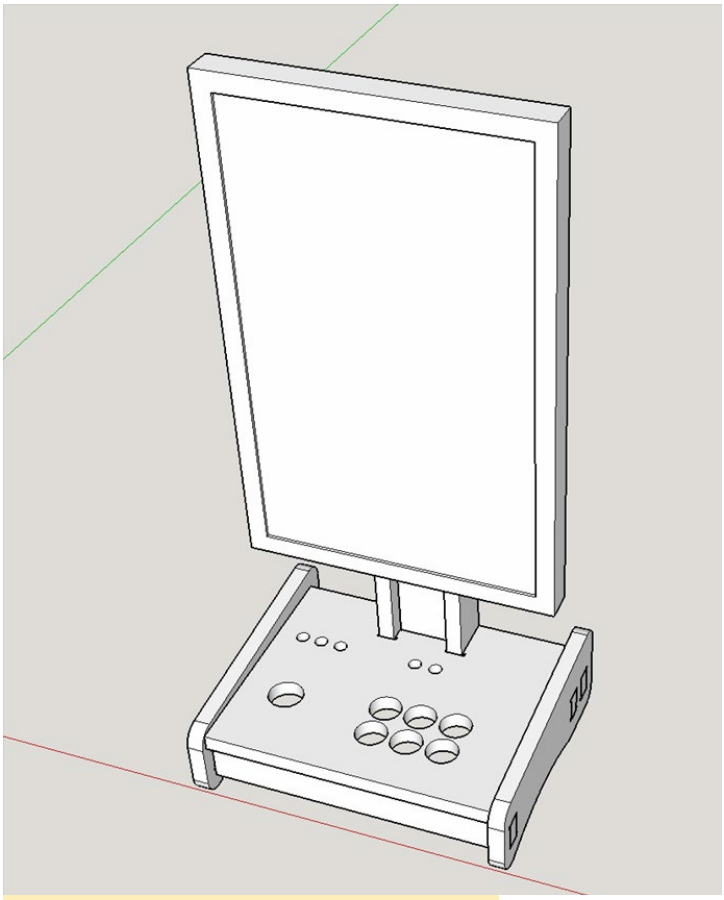
Piezas de Madera montadas



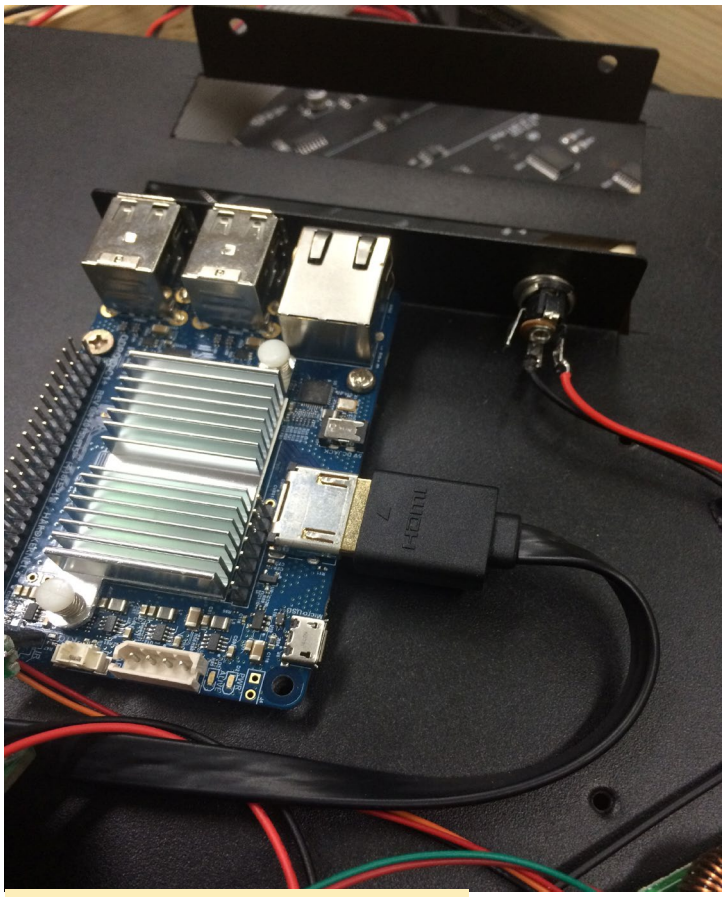
Estructura principal montada en la base

Monitor

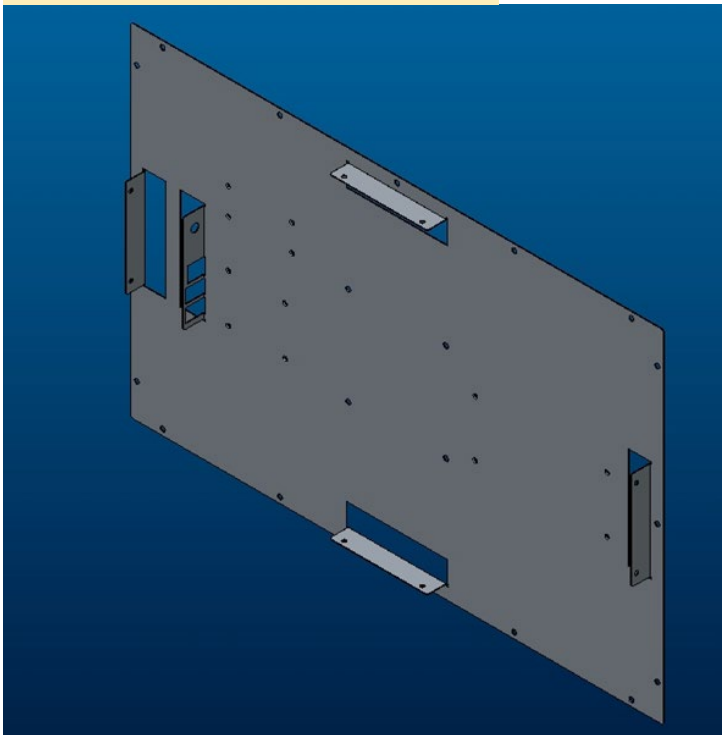
Se eligió una pantalla LCD 16:10 de 24 pulgadas para una mejor experiencia de juego. Puesto que hay muchos juegos retro que se ejecutan en una escala de 4:3, usar una pantalla 16:10 es mejor que una de 16:9 para así mantener la escala original de los clásicos juego arcade. La resolución nativa es de 1920x1200 WUXGA. El número de serie de la LCD es LTM240CL01. Está fabricada por Samsung. Se creó un marco para la pantalla usando piezas de madera con el objeto de darle un toque

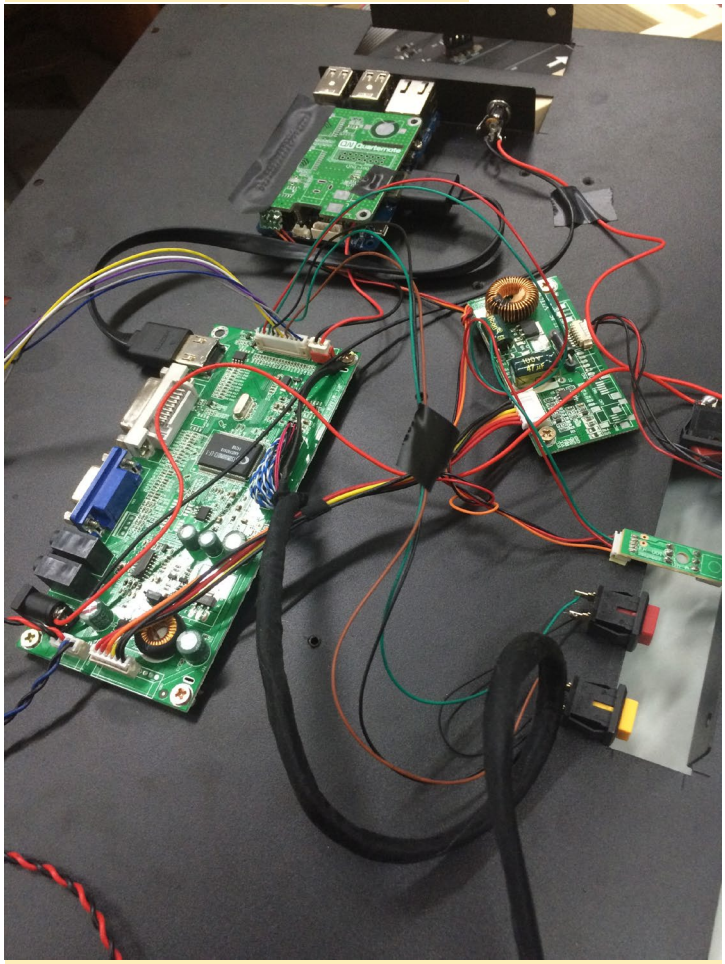


Archivos de diseño



Montaje del ODROID-CI+





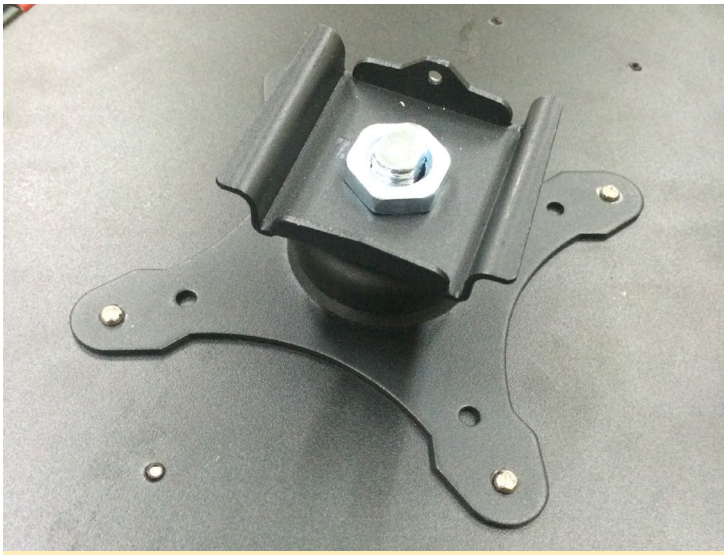
Montaje y conexión de las placas con cables e interruptores

más clásico. La consola parece un armazón de la era de los 80.

La estructura de la parte trasera del monitor cuenta con una lámina de acero con un espesor de 0,8 mm para hacer que ésta sea más resistente. Tiene unos pliegues de 90 grados y agujeros para montar la placa ODROID así como una placa de conversión de señal de video LVDS a HDMI. La alimentación para el ODROID comparte un raíl de 5V con la placa de conversión.

Montaje del cable LVDS

Montaje de la placa de acero en la estructura con tornillos



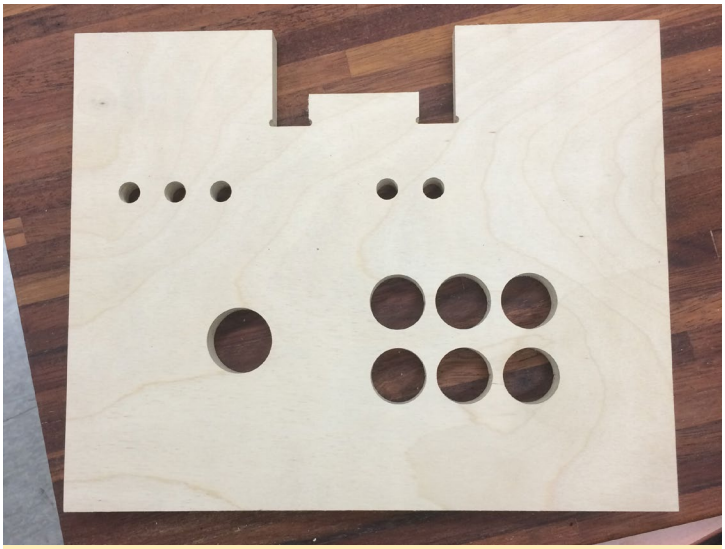
Montaje del componente de la pantalla que permite girarla sobre la estructura de acero

Fijar el monitor

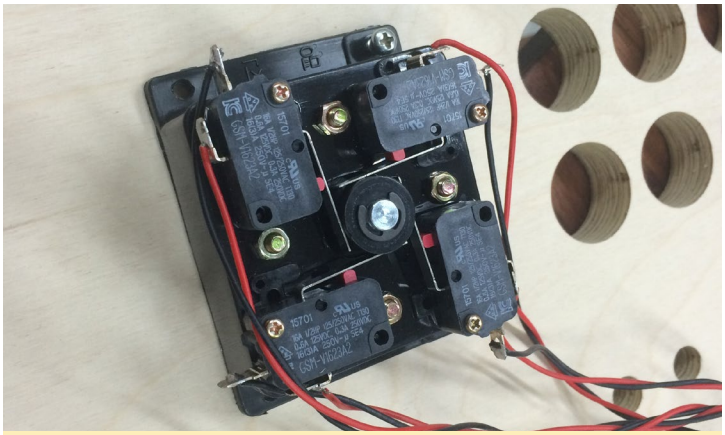
Montaje de la pantalla en la estructura principal

La función de giro funciona perfectamente

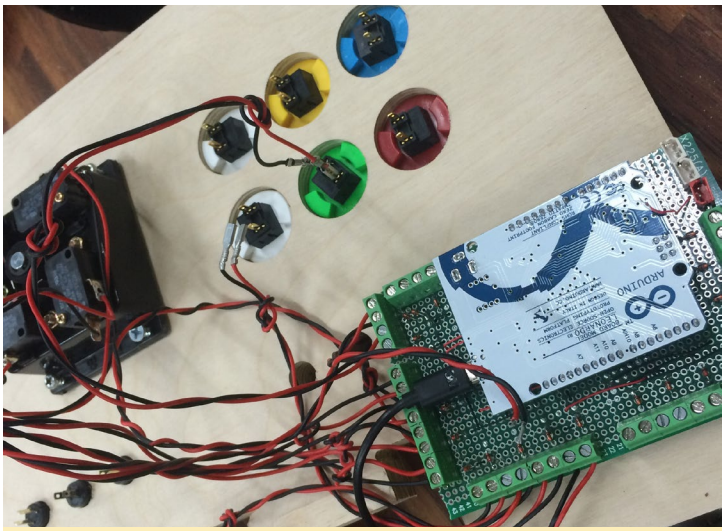
Bloque del joystick



La placa está preparada para albergar 6 botones grandes y un joystick. Es posible asignar otros botones más pequeños a teclas de Android para “Volver atrás”, “Ir al menú” o “Seleccionar un elemento”.



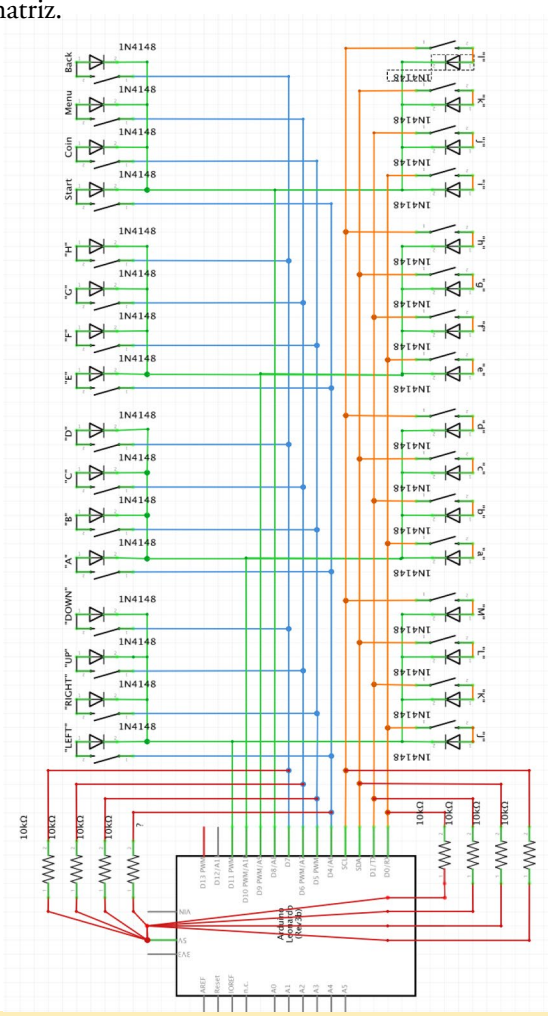
Montaje de los componentes del joystick en la estructura de madera



Arduino gestiona los eventos de entrada de teclas y envía los paquetes de datos a la placa ODROID a través de la interfaz HID USB

¡La consola está casi terminada!

Debido al número limitado de puertos E/S en Arduino, es necesario implementar una matriz de conexiones. Se utilizaron resistencias de 10k ohmios y diodos de conmutación 1N4181 para crear la matriz.



Asignación de botones y joystick

El código fuente de Arduino de este proyecto está disponible en http://bit.ly/1RtSRzp. También puedeS encontrar más información sobre los esquemas del joystick y de los botones en http://bit.ly/1Rs4qgs (Coreano). Hay una guía muy buena para desarrollar un sistema para dos jugadores.

Jugabilidad

Arranca Android en ODROID-C1+ y configura la opción de orientación de la pantalla con la aplicación ODROID-Utility.

Pantalla principal del lanzador personalizado

Juego arcade en modo vertical

Tal y como muestran las imágenes, MAME emula un gran juego de disparos en modo vertical, y la placa de conversión activa automáticamente la escala 3:4. El emulador usado es la versión 0.1398u1 de MAME4droid, disponible de forma gratuita en la tienda de Google Play. Esperamos ver una plataforma con dos jugadores con el potente C2 en lugar del C1+ original.

Un clásico emulador de DOS ejecutanto el famoso juego de Tetris en modo horizontal

La mayoría de los juegos nativos para Android también se ejecutan sin problemas

DESARROLLO ANDROID

COMPILA TU PROPIO SDK DE ANDROID

por Nanik Tolaram

Personalizar tus herramientas es el primer y mejor paso para hacer todos los experimentos con Android que desees.

Lo divertido de Android es la posibilidad de que cualquiera puede añadir lo que quiera y experimentar con diferentes configuraciones. Hay un montón de versiones de Android para diferentes plataformas, cada una es única en sí misma y ofrece algo que el resto no tienen. Por ejemplo, Google ofrece Android pre-instalado en sus dispositivos con Apps de Google como YouTube y Google +. Tú También puedes ofrecer tu propio versión de Android junto con tu SDK para que los usuarios desarrollen y experimenten con ella.

En este artículo, vamos a analizar brevemente cómo añadir un nuevo servicio dentro de Android, y cómo dar a los usua-rios la posibilidad de acceder mismo desde su aplicación. También te enseñare cómo añadir nuevas funcionalidades a un SDK Android ya existente. En este ejemplo, nosotros vamos a utilizar Android Lollipop, aunque las diferentes versiones de Android utilizan el mismo proceso.

Servicio del sistema

En algún momento del desarrollo de Android, te encontrarás con que usas servicios que están expuestos a través del SDK, como LocationManager (GPS), SensorManager y muchos más. Vamos a crear nuestro propio servicio llamado SampleService. El servicio sólo proporcionará un simple método que devuelve la cadena “ValueFromServer” al cliente:

```
package com.android.server;

import android.content.Context;
import com.android.internal.os.ISampleService;

public final class SampleService extends ISampleSer-
vice.Stub {
    private static final String TAG = “SampleService”;

    private final Context mContext;

    public SampleService(final Context context) {
        mContext = context;
    }

    public String getHello(){
        return “ValueFromServer”;
    }
}
```

La clase será creada dentro del directorio frameworks/base/servicies/core/java/com/android/sever. Ten en cuenta que se amplia a ISampleService.Stub, pero la pregunta es: ¿dónde reside esta clase? Será generada por el proceso de compilación basado en la interfaz que vamos a definir dentro del archivo llamado ISampleService.aidl, tal y como se muestra a continuación. El .aidl debe encontrarse dentro del directorio frameworks/base/core/java/com/android/internal/os

```
package com.android.internal.os;

interface ISampleService {
    String getHello();
}
```

El archivo .aidl será usado por el proceso de compilación para generar todos los archivos necesarios del entorno de trabajo y así exponerlo como un servicio remoto que permita a las aplicaciones externas acceder a él, tanto dentro como fuera del

entorno de trabajo.

Android.mk

Para que el archivo .aidl sea procesado por el sistema de compilación, hay que indicarlo expresamente en el makefile de Android.mk dentro del directorio framework/base. Busca la declaración LOCAL_SRC_ FILES dentro del archivo, a continuación agrega la línea que se muestra en la siguiene Figura. Asegúrate de que no haya espacios después del carácter '\', ya que estos bloquearán la compilación.

```
core/java/com/android/internal/backup/IObbBackupService.aidl \
core/java/com/android/internal/policy/IFaceLockCallback.aidl \
core/java/com/android/internal/policy/IFaceLockInterface.aidl \
core/java/com/android/internal/policy/IKeyguardShowCallback.aidl \
core/java/com/android/internal/policy/IKeyguardExitCallback.aidl \
core/java/com/android/internal/policy/IKeyguardService.aidl \
core/java/com/android/internal/policy/IKeyguardStateCallback.aidl \
core/java/com/android/internal/os/IDropBoxManagerService.aidl \
core/java/com/android/internal/os/ISampleService.aidl \
core/java/com/android/internal/os/IParcelFileDescriptorFactory.aidl \
core/java/com/android/internal/os/IResultReceiver.aidl \
core/java/com/android/internal/statusbar/IStatusBar.aidl \
core/java/com/android/internal/statusbar/IStatusBarService.aidl \
core/java/com/android/internal/textservice/ISpellCheckerService.aidl \
core/java/com/android/internal/textservice/ISpellCheckerSession.aidl \
core/java/com/android/internal/textservice/ISpellCheckerSessionListener.aidl \
```

Declaración aidl dentro de Android.mk

Administrador del sistema

El siguiente paso es crear una clase administradora. Esta clase será el principal consumidor de nuestro servicio y además, será la clase a la que tendrán accedo todos los usuario de nuestro servicio. Para que el administrador sea accesible es necesario hacer algunas cosas:

```
package android.os;

import com.android.internal.os.ISampleService;

public class SampleManager {
    private static final String TAG = "SampleManager";
    private final ISampleService mService;

    /** {@hide} */
    public SampleManager(ISampleService service) {
mService = service; }

    public String getHello() {
        String returnValue = "NOTHING";
        try {
                returnValue = mService.getHello();
            }
        catch (RemoteException e) {
            e.printStackTrace();
         }
         return returnValue;
    }
}
```

Esta clase administradora será el punto de entrada a nuestro servicio, y como en cualquier otro administrador de Android, la forma de obtener una referencia es utilizando el método getSystemService. Veremos esta cuestión paso a paso más adelante.

Servidor del sistema

El entorno de trabajo inicia todos los servicios usando un archivo llamado SystemServer.java dentro del directorio frameworks/base/services/java/com/android/. Añadiremos fragmentos de código en esta clase para iniciar nuestro SampleService:

```
….
try {
Slog.i(TAG, "Sample Service");
ServiceManager.addService("SampleService",new
SampleService(context));
} catch (Throwable e) {
reportWtf("starting SampleService", e);
}
….
```

El código debe colocarse dentro del metodo startOtherServices(), como muestra la siguiente figura, que mapeará la cadena SampleService con nuestra clase. Esto será absorbido por la clase administradora que se define en la siguiente sección.

```
if (!disableNonCoreServices) {
    try {
        Slog.i(TAG, "Assets Atlas Service");
        atlas = new AssetAtlasService(context);
        ServiceManager.addService(AssetAtlasService.ASSET_ATLAS_SERVICE, atlas);
    } catch (Throwable e) {
        reportWtf("starting AssetAtlasService", e);
    }
}

if (mPackageManager.hasSystemFeature(PackageManager.FEATURE_PRINTING)) {
    mSystemServiceManager.startService(PRINT_MANAGER_SERVICE_CLASS);
}

mSystemServiceManager.startService(RestrictionsManagerService.class);

mSystemServiceManager.startService(MediaSessionService.class);

try {
    Slog.i(TAG, "Sample Service");
    ServiceManager.addService("SampleService",new SampleService(context));
} catch (Throwable e) {
    reportWtf("starting SampleService", e);
}

if (mPackageManager.hasSystemFeature(PackageManager.FEATURE_HDMI_CEC)) {
    mSystemServiceManager.startService(HdmiControlService.class);
}

if (mPackageManager.hasSystemFeature(PackageManager.FEATURE_LIVE_TV)) {
    mSystemServiceManager.startService(TvInputManagerService.class);
}

if (!disableNonCoreServices) {
```

Fragmento de SystemServer.java

Context

Todas las aplicaciones en Android deben acceder al objeto Context. Este objeto es una especie de clase global, que contiene una API que permitirá a nuestra aplicación interactuar con el entorno de trabajo. En este paso, vamos a presentar nuestro servicio a través de Context cambiando el código en la clase ContextImpl, que contiene la implementación del objeto Context. La siguiente figura muestra el código que debe

ser añadido para nuestro servicio dentro de ContextImpl.java.

```
    registerService(MEDIA_PROJECTION_SERVICE, new ServiceFetcher() {
            public Object createService(ContextImpl ctx) {
                return new MediaProjectionManager(ctx);
            }
    });

    registerService(APPWIDGET_SERVICE, new ServiceFetcher() {
        public Object createService(ContextImpl ctx) {
            IBinder b = ServiceManager.getService(APPWIDGET_SERVICE);
            return new AppWidgetManager(ctx, IAppWidgetService.Stub.asInterface(b));
        }});

    registerService("SampleServiceManager", new ServiceFetcher() {
            public Object createService(ContextImpl ctx) {
                IBinder b = ServiceManager.getService("SampleService");
                return new SampleManager(ISampleService.Stub.asInterface(b));
            }});
    }

    static ContextImpl getImpl(Context context) {
        Context nextContext;
        while ((context instanceof ContextWrapper) &&
```

Fragmento de ContextImpl.java

Política SELinux

En Android Lollipop, cuando definimos un nuevo servicio, tenemos que añadirlo al archivo de contexto SELinux para que éste sea iniciado correctamente y evitar errores. Necesitamos declarar nuestro servicio dentro del archivo service_contexts que reside dentro del directorio external/sepolicy. La siguiente figura muestra la línea adicional que debemos añadir.

```
accessibility                   u:object_r:system_server_service:s0
account                         u:object_r:system_server_service:s0
activity                        u:object_r:system_server_service:s0
alarm                           u:object_r:system_server_service:s0
android.security.keystore       u:object_r:keystore_service:s0
appops                          u:object_r:system_server_service:s0
appwidget                       u:object_r:system_server_service:s0
SampleService                   u:object_r:system_server_service:s0
assetatlas                      u:object_r:system_server_service:s0
audio                           u:object_r:system_server_service:s0
backup                          u:object_r:system_server_service:s0
batteryproperties               u:object_r:healthd_service:s0
```

Fragmento de service_contexts

Compilación

Llegados a este punto, hemos completado la configuración necesaria para nuestro servicio. Ahora, podemos ejecutar la secuencia de compilación que vamos a describir a continuación:

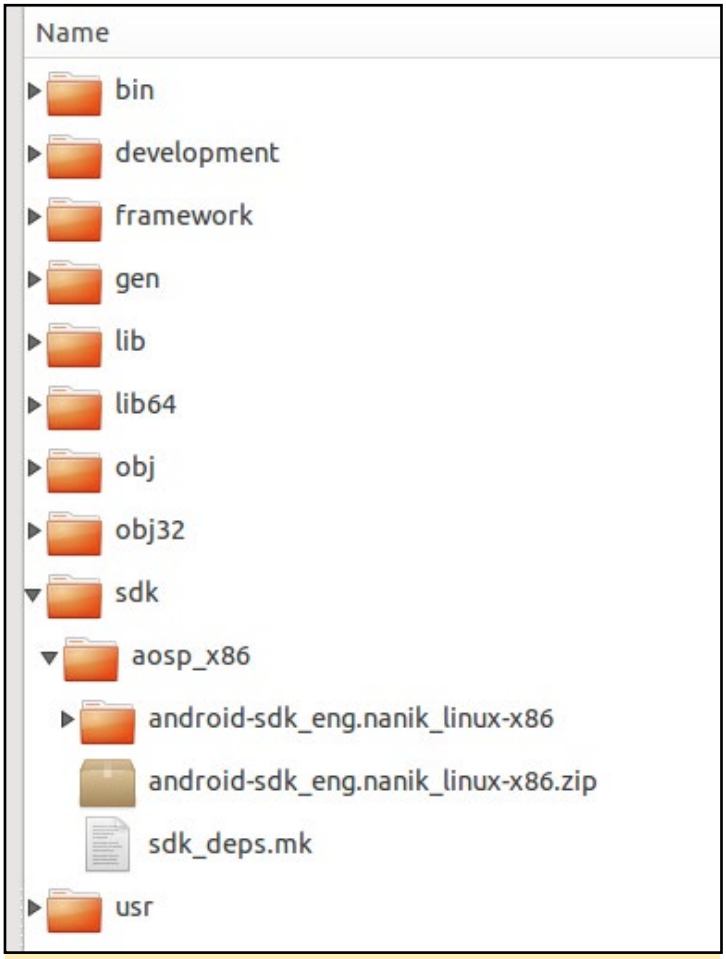
Ejecuta el siguiente comando para actualizar los archivos importantes e informarles de los cambios en el entorno de trabajo y en el SDK. Esto actualizará un archivo llamado current.txt para alojar nuestro nuevo servicio, que luego se utilizará para empaquetar el SDK en el siguiente paso.

```
$ make update-api
```

Asegúrate de que no aparezca ningún error al ejecutar el paso 1. Después, genera el SDK necesario para nuestro IDE:

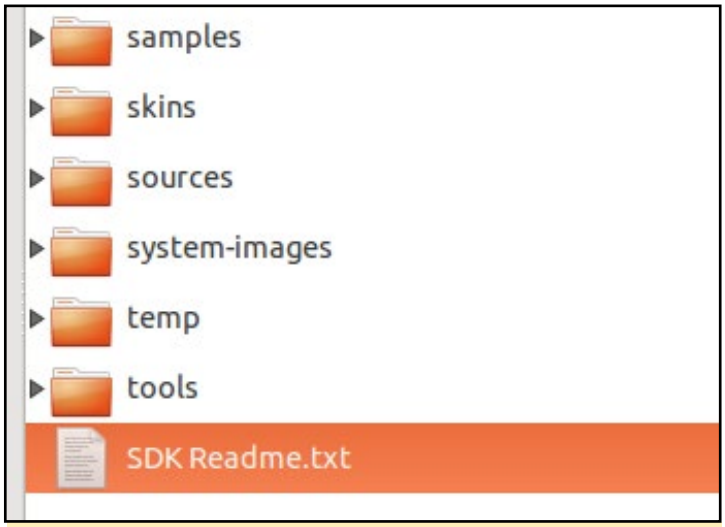
```
$ make sdk -j6
```

En una compilación correcta, verás un archivo en el equipo local que contiene el SDK completo de Android. En mi ejemplo, el archivo se llama android-sdk_eng.nanik_linux-x86.zip, como muestra la siguiente figura. Necesitas copiar los archivos del directorio que tienen el mismo nombre que el archivo .zip.



Archivo SDK .zip generado

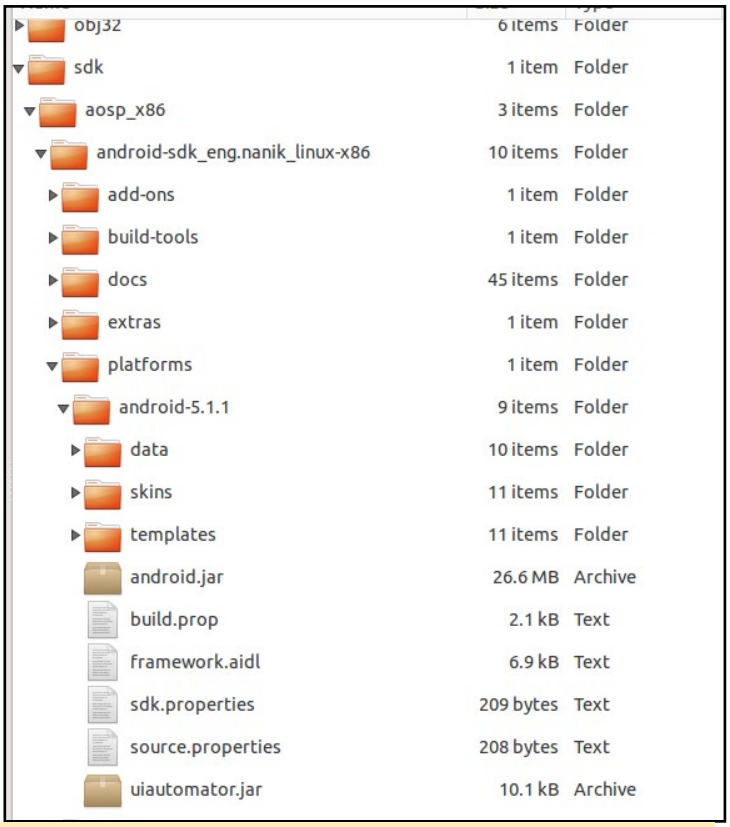
He instalado el SDK de Android en un directorio llamado /home/nanik/Work/android-sdk-linux, como se muestra en la siguiente figura. Puesto que estamos compilando el SDK para Lollipop, necesitamos copiar el nuevo SDK en <tu-carpeta_sdk_android>/platforms/android-22. Elimina todos los archivos que están dentro de la carpeta android-22 pero antes, haz una copia de seguridad de toda carpeta.



SDK completo de Android

En mi caso el nombre de la carpeta que contiene el SDK

se llama android-sdk_eng.nanik_linux-x86 pero el nombre de tu archivo será algo diferente. A continuación, copia todo el directorio android-sdk_eng.nanik_linux-x86/platforms/android-5.1.1 dentro de la carpeta SDK en /home/nanik/Work/android-sdk-linux/platforms/android-22, como se muestra en la siguiente figura. Ahora estamos listos para utilizar nuestro SDK compilado a medida.



Compilación personalizada del SDK

App de muestra

Lanza Android Studio y crea una sencilla aplicación para Android. Asegúrate de que utilizas la versión correcta del SDK dentro de build.gradle para asegurarnos de que el IDE coge los archivos del SDK correctos. La siguiente figura muestra lo que yo tengo en mi Android Studio. La función de autocompletar en el IDE será capaz de reconocer la clase SampleManager, como se muestra en la Figura de arriba.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 22
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId "sample.nanik.com.testsdk"
        minSdkVersion 22
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Fragmento de build.gradle

```
package sample.nanik.com.testsdk;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Sample
    }
}
```

```
SampleManager (android.os)
Sampler (android.renderscript)
ISampleService (com.android.internal.os)
```

Función de autocompletar con SampleManager

El código fuente completo de la aplicación se muestra a continuación:

```
package sample.nanik.com.testsdk;

import android.app.Activity;
import android.os.Bundle;
import android.os.SampleManager;
import android.util.Log;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceS-
tate) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        SampleManager manager = (SampleManager)getSys
temService("SampleServiceManager");
        String s = manager.getHello();
        Log.i("MainActivity",s);
    }
}
```

UNICORN

EMULACION DE INSTRUCCIONES CPU PARA ARM

por @petevine

El entorno de trabajo y emulador de CPU Unicorn, disponible en `http://bit.ly/1VoerJn`, es un nuevo y emocionante proyecto escrito en C que recientemente ha recibido soporte para ARM. Unicorn sólo emula instrucciones de CPU, lo cual permite muchas posibilidades para futuros proyectos de emulación. Unicorn está basado en QEMU, aunque usa técnicas jitting, similar a PyPy, para conseguir una mejora en el rendimiento del tiempo de ejecución. Te recomiendo que leas el documento de `bit.ly/1Vofd9i` para entender mejor las diferencias.

Las instrucciones para instalar Unicorn desde el código fuente las puedes encontrar en `bit.ly/1OgV0wb`, son bastante sencillas. La única dependencia es Capstone (`bit.ly/1U8Pxz3`), que utiliza el mismo sistema de compilación.

Actualmente, el proyecto más desarrollado para mostrar las posibilidades del Unicorn se llama Usercorn (`bit.ly/1mIlh06`). Fue escrito en Go por @lunixbochs, pionero en GLShim, como un emulador del espacio de usuario como qemu-user. Sin embargo, a diferencia de qemu que tiene binarios separados para ARM y x86, es capaz de manejar todas las arquitecturas soportadas por Unicorn al mismo tiempo.

He subido un "kit de inicio usercorn" a `bit.ly/1U8PXoZ` para cualquiera que quiera probarlo sin tener que instalar primero Go. Los binarios busybox en lote deberían servir igualmente. Para comentarios, preguntas y sugerencias, por favor visita el post original en `bit.ly/1Lu9qcJ`.

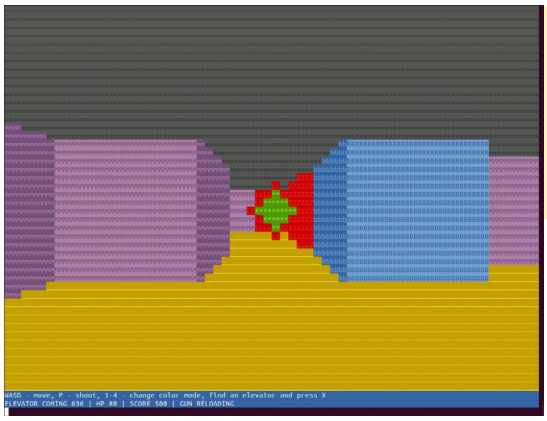
```
odroid@odroid:"/usercorn-kit$  ./usercorn busybox-x86_64 cal
    January 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
odroid@odroid:"/usercorn-kit$  ./usercorn busybox-x86_64 whoami
odroid
odroid@odroid:"/usercorn-kit$  ./usercorn busybox-armv5l df -h
Filesystem                Size      Used Available Use% Mounted on
udev                    467.1M      4.0K    467.1M   0% /dev
tmpfs                    94.8M      2.8M     92.0M   3% /run
/dev/disk/by-uuid/e139ce78-9841-40fe-8823-96a304a09859
                         29.1G     22.2G      6.0G  79% /
none                      4.0K         0      4.0K   0% /sys/fs/cgroup
tmpfs                   474.1M     80.1M    394.0M  17% /tmp
none                      5.0M      8.0K      5.0M   0% /run/lock
none                    474.1M     60.6M    413.5M  13% /run/shm
none                    100.0M     36.0K    100.0M   0% /run/user
/dev/mmcblk0p1          120.3M     10.5M    109.8M  14% /media/boot
/dev/loop0                3.9G    803.9M      3.1G  20% /home/odroid/Downloads/cache
invalid fetch: 0x400001d0, 0x1 = 0x0
[memory map]
  0xffff0000-0x10001000 rwx
  0x0000-0x125000 r-x busybox-armv5l [exe]
  0x12a000-0x132000 rw- busybox-armv5l [brk]
  0x60000000-0x60801000 rwx [stack]
  0x125000-0x127000 rwx
  0x32000-0x134000 rw- [brk]
[registers]
   r0 0x00000000    r4 0xffffffff    r8 0x00000000   r12 0x607febd8
   r1 0x00000001    r5 0x0012a920    r9 0x607fef40       0x00000000
   r2 0xffffffb8    r6 0x0012a8d8   r10 0x00000035       0x00000000
   r3 0x400001d0    r7 0x000003e9   r11 0x00000002       0x00000000
[stacktrace]
  0x400001d0
panic: invalid memory fetch (UC_ERR_FETCH_UNMAPPED)

goroutine 1 [running]:
main.main()
        /home/odroid/go/src/github.com/lunixbochs/usercorn/go/usercorn/cli.go:175 +0x16dc

goroutine 17 [syscall, locked to thread]:
runtime.goexit()
        /home/odroid/go1.5/src/runtime/asm_arm.s:1036 +0x4

goroutine 34 [syscall, locked to thread]:
runtime.goexit()
        /home/odroid/go1.5/src/runtime/asm_arm.s:1036 +0x4
odroid@odroid:"/usercorn-kit$
```

AWKASTER

TERMINAL RETRO CON JUEGOS FPS

por Justin Lee

Te gustaría llevar algo de diversión de la vieja escuela a tu terminal! En este juego seudo-3D inspirado en los clásicos Wolfenstein 3D y Doom, puedes explorar una mazmorra y luchar contra monstruos. Tu objetivo es navegar por el laberinto y activar el ascensor de salida teniendo que eliminar multitud de monstruos por el camino.



Navegar por el laberinto, ¡al estilo terminal!

Awkaster se puede instalar con los siguientes comandos:

```
$ sudo apt-get install gawk
$ git clone\
 https://github.com/TheMozg/awk-raycaster.git
$ cd awk-raycaster/
$ gawk -f awkaster.awk
```

Controles

WASD - movimiento
J/L - giro izquierda/derecha (pulsar shift para girar rápido)
Barra espaciadora - disparar
Números 1 - 4 - cambiar el color
x - Activar ascensor (al que llegarás tras 1000 movimientos)

Puedes encontrar más información y el código fuente en `http://bit.ly/1P1B4V7`. El código es divertido e interesante de leer. ¡Nunca pude imaginar que un juego en 3D se pudiera escribir en tan sólo 600 líneas! Para hacer los muros de las mazmorras, el juego utiliza la intersección rayo-superficie. Se trata de un sencillo algoritmo de representación que no requiere ningún modelado 3D o cálculo complejo, manteniendo el código muy simple. Los monstruos y los proyectiles son añadidos como sprites una vez creados los muros.

Para comentarios, preguntas y sugerencias, por favor visita el post original en `bit.ly/1TrhUcg`.

JUEGOS LINUX

UN ODROID-W DENTRO DE LA CARCASA DE UNA GAMEBOY ADVANCE SP

por Johnny Parks

La creación de sistemas de videojuegos portátiles ha sido para mí un hobby desde hace unos cuatro años. Empecé convirtiendo sistemas de consola como la Nintendo 64 y la GameCube en dispositivos portátiles. Después, oí hablar de los proyectos ODROID. Siempre he querido crear mi propio dispositivo de emulación, pero por aquel entonces la Raspberry Pi era el principal dispositivo para hacerlo. Una vez anunciado el ODROID-W, supe que sería el mejor para crear un pequeño sistema de emulación. Lo que no esperaba es que pudiera coger dentro de la "cáscasa" de una GameBoy Advance SP .

Figura 1 - GameBoy Advance ejecutandose con un ODROID-W en su interior

Aquí tienes las especificaciones de este diminuto ordenador portátil:

- ODROID W (un clone de la Raspberry pi)
- Batería de 1300mAh (batería de teléfono móvil [muy pequeña])
- HUB USB con 3 puertos USB, 1 será usado por el WiFi en el interior y 2 disponibles en el exterior.
- Puerto HDMI, que es fantástico cuando se utiliza como reproductor multimedia en la gran pantalla de tu casa.
- Altavoz mono conectado a un amplificador con volumen digital e interruptores táctiles para cambiar el volumen
- Pantalla de 3,5 pulgadas
- Pad direccional con los botones A, B, X, Y, L, R, Start, y Select

La Game Boy Advance SP (GBA SP) original utiliza una CPU de 16,8 MHz, 96 KB de RAM de vídeo y 32 KB de RAM del sistema. Aunque esto no es nada si lo comparamos con los dispositivos de hoy en día, era realmente asombrosa cuando fue lanzada en 2003. El ODROID-W cuenta con 700 MHz de CPU y 512 MB de memoria RAM, además de una extensa colección de software de código abierto disponible.

Tras conseguir una carcasa de Game Boy Advance SP en internet, utilice un pequeño taladro para dar cabida a todos los componentes. Una de las tareas más complicadas fue cortar la placa PCB que estaba detrás de la pantalla LCD. Mi sistema utiliza una pantalla de 3.5 pulgadas de una cámara de visión trasera que generalmente se utiliza en los coches como dispositivo auxiliar. Puesto que estas pantallas tienen una entrada compuesta y pueden funcionar utilizando 5V con algunas modificaciones, era la pantalla perfecta para este proyecto. El GBA SP original tenía una pantalla de 2,9 pulgadas con una resolución muy baja, así que supuso una mejora considerable, aunque necesité hacerle ciertas modificaciones para que encajara. Incluso tuve que raspar algunas zonas de la placa para poder cablear los condensadores. Tuve que hacer algunos otros retoques menores que precisaron paciencia y destreza.

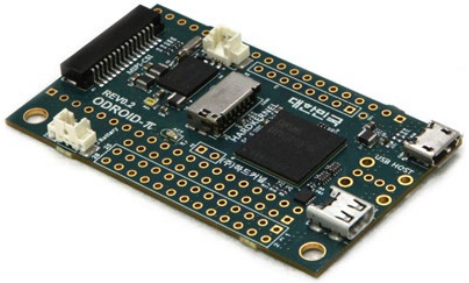


Figura 2 – Primer plano del ODROID-W

Puesto que el ODROID-W ya cuenta con un transformador, puedes utilizar una batería del teléfono para alimentar

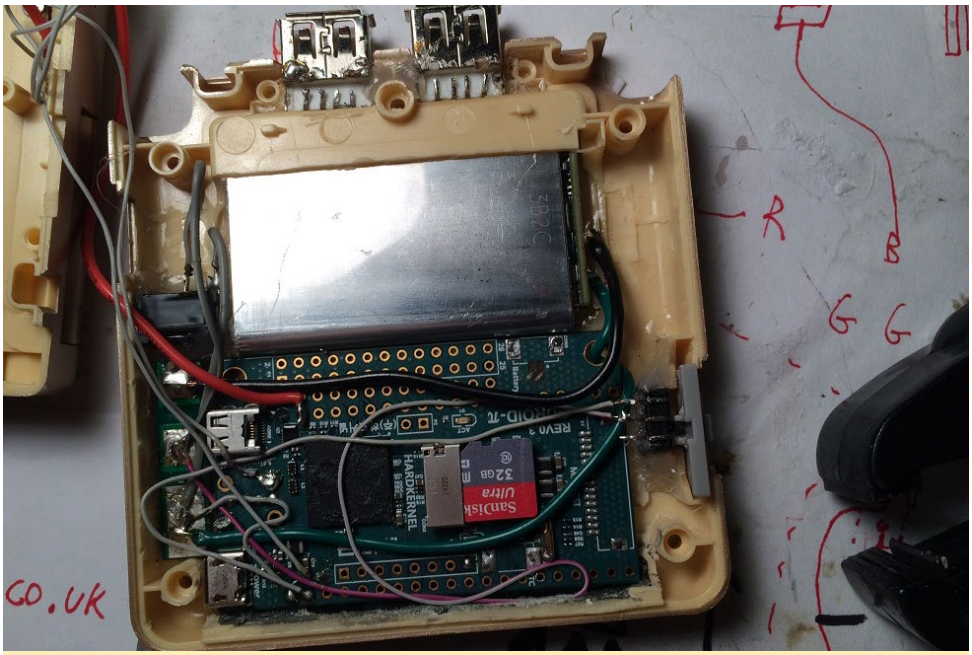


Figura 3 - ODROID-W instalado dentro de la carcasa de la GameBoy Advance

todo el sistema. Sorprendentemente, el ODROID-W funcionaba a muy baja potencia, de modo que con una batería con una capacidad de 1300mAh era capaz de alimentar todo el sistema durante unas tres horas. Incluso llegue a ver por completo The Amazing Spiderman durante un viaje en coche. Una cosa que me preocupaba era saber si el adaptador Wi-Fi funcionaría. Cuando lo conecté a un HUB USB de cuatro puertos, observé que rara vez conectaba. Después me di cuenta que el adaptador Wi-Fi que utilicé absorbía mucho amperaje en los puertos USB. Terminé conectando directamente la fuente de alimentación de la batería a la placa Wi-Fi para suministrarle mayor amperaje.

Para conseguir que los controles funcionasen en el dispositivo, necesitaba algo muy pequeño. Por suerte, tenía a mano un dispositivo llamado Teensy 3.1, que permite emular los movimientos del teclado y ratón usando botones táctiles. Este dispositivo es como un mini ordenador Arduino, que puede ser programado para diferentes cosas. Los botones del dispositivo fueron fáciles de programar, pero el joystick que actuaba como ratón era otra historia. Esta cuestión me supuso un día de investigación y de pruebas para conseguir que el dispositivo Arduino imitara un ratón. El joystick que utilicé procedía de una Nintendo 3DS, que compré online por menos de 10$.

Una vez que el sistema estaba terminado por completo, cargué unos emuladores y ejecuté algunos juegos clásicos que me hicieron sentir nostalgia. Fui capaz de ejecutar el emulador de Nintendo 64 y de jugar a algunos juegos de Super Mario 64 en lo que parecía una GBA SP. La mejor fue ejecutar el Quake III, recordé jugarlo hace años con un amigo. Desafortunadamente, la funciones online no están disponibles, aún así fue divertido practicar con los CPNs. El sistema puede incluso navegar por la web, de modo que tienes donde divertirte sin límites con todo lo que te proporciona este curioso sistema ODROID.

Figura 4 – Parte trasera de la carcasa

En futuros proyectos incluiré un ODROID-C0, aportando un valor añadido a este proyecto. El ODROID-C0 es uno de los dispositivos más recientes de Hardkernel, en el cual se pueden ejecutar aplicaciones de Android. Esta placa también tiene un regulador/transformador incorporado, de modo que puedo utilizar la misma batería del teléfono. En lugar de una carcasa GBA SP, planeo usar una 3DS XL que tiene un montón de espacio para todos los componentes. El ODROID-C0 puede ejecutar juegos que el ODROID-W no puede, los juegos de Nintendo 64 y de Playstation también se ejecutará sin problemas en este sistema. Espero tener terminado este proyecto en un par de meses, así que echa un vistazo de vez en cuando a los foros de ODROID-C0. Y recuerda, ¡No dejes que tus sueños se queden en simplemente sueños!

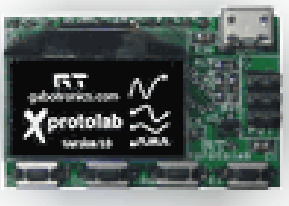
Figura 5 - Game Boy Advance arrancando Linux

Figura 6 - Game Boy con el ODROID-W

MULTISCOPE VERSATIL

UN PROYECTO OSCILOSCOPIO PARA TU CASA

por Venkat Bommakanti



Xprotolab

Hace casi un año, publicamos un artículo sobre cómo conectar un osciloscopio vía USB a un ODROID-C1. En este número, describiremos cómo conectar un MultiScope de bajo coste pero con un gran potencial, llamado Xprotolab Plain a un ODROID-C0/C1/C1+ o a un ODROID-XU4. Una solución para el ODROID-C2 debería estar disponible muy pronto.

La placa fue diseñada y desarrollada originalmente por Gabotronics, pero Hardkernel la ofrece como accesorio para algunos de los SBC ODROID. El software de visualización multi-plataforma basada en QT5 es de código abierto, por lo que la comunidad puede modificarlo según sus necesidades. Los detalles relativos al hardware y software los puedes encontrar en los enlaces que aparecen al final de este artículo.

Figura I - Xprotolab y IO pines

Preprar el sistema

Actualiza el sistema ODROID utilizando los pasos bien documentados que aparecen en la página web de Hardkernel y en números anteriores de ODROID Magazine, asegurándote de reiniciar el sistema tras la actualización.

Si quieres utilizar un ODROID-C0/C1/C1+, tendrás que aumentar el espacio swap de intercambio disponible, ya que la memoria interna de 1 GB de RAM es insuficiente. Los siguientes comandos elevaran el espacio de intercambio al necesario:

```
$ sudo fallocate -l 2G /swapfile
$ sudo mkswap /swapfile
$ sudo swapon /swapfile
```

Una vez que el proceso de compilación del software haya finalizado, puedes liberar el espacio de intercambio con el siguiente comando:

```
$ sudo swapoff -a
```

Instalar el software

El software de código abierto está basado en QT y es compatible con QT5. Los siguientes comandos instalan todo el software previo necesario:

```
$ sudo apt-get install git \
  qt5-default \
  Libqt5serialport5-dev \
  libusb-1.0-0-dev
```

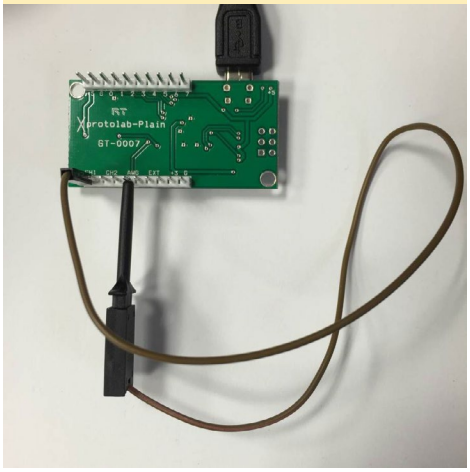
A continuación, crea un directorio para colocar el código fuente del software de visualización, que luego será compilado para crear el ejecutable:

```
$ cd ~ && mkdir xp && cd xp
$ git clone https://github.com/
ganzziani/xscopes-qt
$ cd xscopes-qt
$ qmake
$ make -j8
```

Conectar multiscope

Utilizaremos el ejemplo de la visualización en forma de onda para compro-

Figura 2 - Esquema de conexiones



bar el uso del dispositivo. Utiliza el cable proporcionado como se muestra en la Figura 2 para conectar el pin del canal 1 y el pin AWG. Luego, conecta el cable USB suministrado al MultiScope y a la placa ODROID.

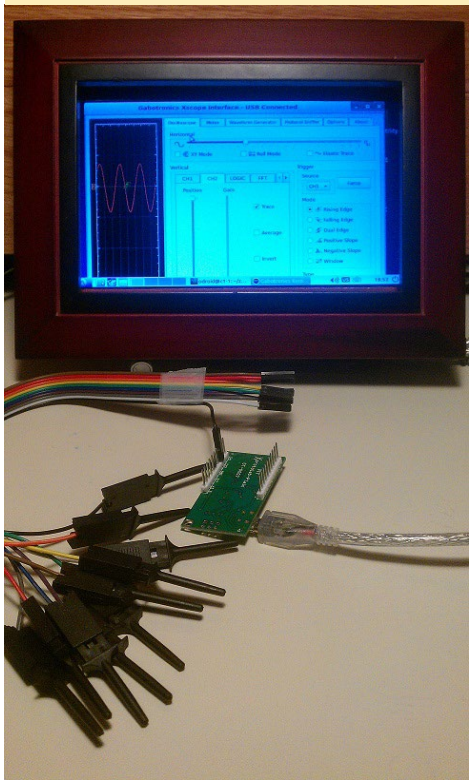
Ejecutar la aplicación

La aplicación tiene que ser compilada en el directorio de trabajo creado anteriormente. Desde allí, puedes ejecutarla usando el siguiente comando:

```
$ sudo ./xscope
```

Aparecerá una pantalla como la que muestra la Figura 3, con un ODROID C1+ y un ODROID-VU7. La Figura 4 muestra las diversas funcionalidades de la interfaz de usuario.

Figura 3 - MultiScope conectado al ODROID-CI +



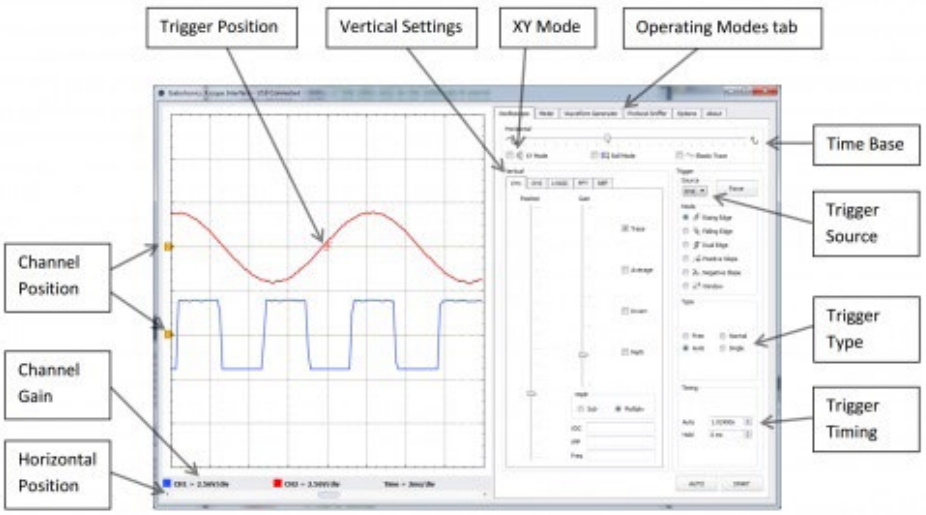
Referencias

Wiki XProtolab
http://bit.ly/22xfjzn

Especificaciones XProtolab
http://bit.ly/25hyVdi

Documentación Qt
http://doc.qt.io/qt-5

Figura 4 - Características de la interfaz de usuario



XMAGE

UN MAGIC ONLINE: THE GATHERING GRATUITO

por David Lima

Te has preguntado alguna vez si es posible jugar a juegos de cartas online o en red, como Magic: The Gathering con amigos repartidos por diferebtes lugares. Sí que puedes, con un ordenador de placa reducida ODROID y el software multi-plataforma XMage (MAGE:Magic, Another Game Engine).

Magic: The Gathering, comúnmente conocido como Magic, fue el primer JCC (Juegos de Cartas Coleccionables) creado por Richard Garfield en 1993. Básicamente, dos o más jugadores construyen su propia baraja de cartas, sobre la base de un conjunto de cartas reales e intentan derrotar a un oponente reduciendo sus niveles de vida a cero, cumpliendo al mismo tiempo con las diferentes reglas del juego. Las cartas y todos los conceptos del juego están basadas en juegos de rol de fantasía, muy parecidos al famoso juego Dungeons & Dragons. Los personajes incluyen enanos, elfos, orcos y otras criaturas mitológicas. Algunos fans del juego simplemente coleccionan las cartas y no las usan en juegos reales. Este popular juego cuenta con más 20 millones de jugadores en todo el mundo, y cada año se añaden nuevas expansiones de cartas.

XMage, una aplicación Java, implementa una método virtual online para ejecutar Magic. Además proporciona un sistema cliente/servidor en el que puedes crear tu propio servidor de juegos privado o unirte a uno público. El moderador hará cumplir las reglas del servidor y los clientes pueden crear un nuevo juego o unirse a uno ya en curso. Los jugadores tienen acceso a la mayoría de las cartas creadas en el servidor para poder así crear sus barajas. Los jugadores pueden llegar a tener cartas increibles o súper raras en su baraja, aun cuando normalmente no pueden permitírselas. Por desgracia, no pueden añadir las cartas que están disponibles en el servidor a su colección permanente.

Requisitos previos

Para instalar XMage, primero debes tener instalada como mínimo la versión 1.7.x de Java Runtime Environment (JRE) en tu sistema ODROID. Si utilizas la imagen de Ubuntu proporcionada por Hardkernel, lo más probable es que ya la tengas. De lo contrario, puedes descargarla desde http://bit.ly/1SxvU43.

Para comprobar tu versión actual de Java escribe lo siguiente:

```
$ java -version
openjdk version "1.8.0_01-internal"
OpenJDK Runtime Environment
(build 1.8.0_01-internal-b15)
OpenJDK 64-Bit Server VM (build
25.01-b15, mixed mode)
```

Instalación

Crea un directorio marcador y después, descarga el paquete de distribución XMage en el mismo:

```
$ cd ~/ && mkdir xmage
$ wget -c http://bit.ly/
xmage_1_4_9v0w
$ unzip xmage_1.4.9v0.zip -d
xmage/
```

Uso

XMage se divide en dos partes: el cliente y el servidor. Si deseas utilizar tu servidor local, puede iniciarlo en modo local con un terminal. Puede utilizar un sistema para ejecutar tanto el cliente como el servidor. Si quieres ejecutar el servidor en un sistema independiente, copia el contenido del directorio ~/xmage/mageserver a la misma ubicación en el nuevo sistema. Otra posibilidad es conectar a un servidor público a través de Internet. Para iniciar el servidor en modo local, escribe lo siguiente en una nueva ventana de terminal:

```
$ cd ~/xmage/mage-server
$ chmod +x startServer.sh
$ ./startServer.sh > /dev/null
2>&1 &
```

A partir de ahora, tú y tus amigos podréis conectaros al servidor utilizando la dirección IP de tu ODROID, incluso si utilizan diferentes plataformas como Windows o Mac OS. Si quieres que gente externa a tu red local pueda conectarse, necesitas configurar un servicio de

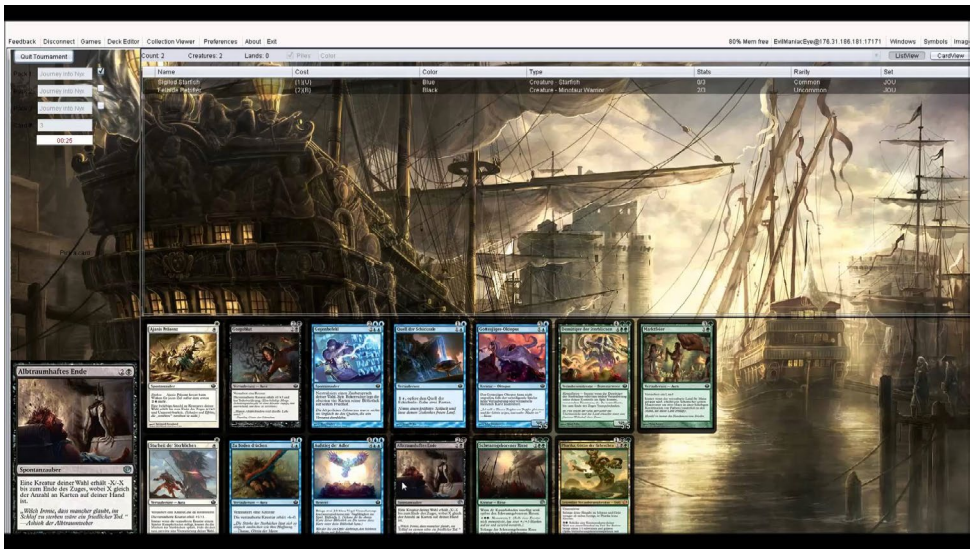
DNS dinámico como DynDNS o No-IP para facilitarle el acceso al servidor. También es necesario redireccionar el puerto 17171 (si se utiliza la configuración por defecto) en tu router a la dirección IP local de tu ODROID. Esta configuración varía para cada modelo de router, de modo que consulta el manual del router para averiguar cómo hacerlo.

Por lo general, se puede utilizar la configuración del servidor por defecto tal como está, pero si deseas cambiarla por cualquier razón, dirígete al directorio xmage/mage-server/config y edita config.xml adecuadamente. Puede modificar el nombre de tu servidor, el puerto de enlace primario y secundario, así como otros ajustes. Encontrarás más detalles de cada parámetro dentro del propio archivo. Ten cuidado al editarlo, puedes provocar que el servidor no se ejecute correctamente, así que asegúrate de hacer una backup de tu configuración antes de hacer cualquier cambio.

A continuación, inicia el software cliente en una nueva ventana de terminal. Para iniciar correctamente el cliente, debes hacerlo sobre la interfaz gráfica del sistema, si utilizas un ODROID secundario o bien exportar el X11 y reenviarlo desde el equipo remoto a tu equipo local. Despues, aparecerá una ventana de conexión al servidor:

```
$ cd ~/xmage/mage-client
$ chmod +x startClient.sh
$ ./startClient.sh > /dev/null
2>&1 &
```

Si estás utilizando un servidor local, tendrás que introducir tu información del servidor, como la dirección IP, el puerto y un nombre de usuario. Si vas a utilizar un servidor público, puedes hacer clic en el botón “Find” para buscar servidores públicos disponibles a través de Internet y unirte a ellos. También hay un campo de contraseña pero no es necesaria, se trata de una posible implementación incompleta del código de autenticación.



Una vez conectado correctamente al servidor, te llevará a la antesala principal, donde puede unirse a cualquiera partida o torneo que esté en curso, o iniciar una nueva utilizando los botones de la parte superior izquierda. Si decides crear un nuevo juego, actualiza la configuración de acuerdo a tus preferencias, como las cartas permitidas, el formato de juego (T2, moderno, o comandante), una contraseña para controlar el acceso y así sucesivamente. Después podrás iniciar tu juego. Invita a tus amigos a que se conecten a la misma partida/torneo ¡y a disfrutar!

La primera vez que inicies tu cliente, aparecerá sin imágenes de cartas y símbolos de acción. Para hacer que tu juego sea más amigable, puedes descargar estos recursos usando los botones superiores a la derecha “Symbols” e “Images”. Haz clic en Symbols y confirma la descarga. Verás un cuadro de diálogo de resultados cuando se complete la descarga. Haz clic en Images y selecciona la fuente desde la que desea descargar las imágenes. Por lo general, las imágenes de los magos o de la información de las cartas son las mejores opciones.

Por favor, ten presente que XMage se encuentra aún en fase de pruebas, por lo que pueden aparecer errores o que falta algunas funcionalidades. Si quieres informar de algo o tienes alguna sugerencia, visita su foro en http://bit.ly/1RdBd3F. Si desea más información sobre XMage, visita su sitio web en http://xmage.de. Para conseguir más información sobre este juego, Magic: The Gathering, visita http://magic.wizards.com. El código fuente también está disponible en GitHub, y puedes ponerte en contacto con el equipo de desarrollo en los foros si deseas contribuir en su desarrollo. Ten en cuenta que estos pasos han sido probados con los modelos ODROID XU4, U3 y C1/C1+ ejecutando Ubuntu de 32 bits.

Referencias

Java SE
http://bit.ly/1SxvU43

Página Web de XMage
http://xmage.de/

Repositorio GitHub de XMage
http://bit.ly/1SbEhzE

Foros de XMage
http://bit.ly/1RdBd3F

SEGUIMIENTO DE OBJETOS USANDO LA OCAM Y EL ODROID-XU4

UNA SENCILLA GUIA PASO A PASO

por DongHyun Yoo

Imagino que muchas personas estarían interesadas en un manual sobre cómo usar oCam con el ODROID XU4 para rastrear objetos usando OpenCV. Esta guía te enseñará los pasos que debes seguir para crear y ejecutar una aplicación de seguimiento de objetos. La habilidad para rastrear un objeto específico en múltiples fotogramas es una tecnología clave en aplicaciones como la robótica o la videovigilancia automática. Con el siguiente código de ejemplo, trataremo de seguir un objeto utilizando las propiedades de los colores que aparece en la imagen.

Configuración

Para empezar, necesitas los siguientes

Figura 1 - Ejemplo de pantalla con seguimiento de objetos

elementos, todos ellos están disponibles en la tienda de Hardkernel:

- ODROID-XU4
- Módulo de memoria, ya sea eMMC o tarjeta microSD con Ubuntu.
- oCam

Además, tendrás que instalar los siguientes paquetes de software, los cuales se pueden instalar utilizando el Gestor de paquetes Synaptic:

- gcc
- wget
- OpenCV

Para preparar tu sistema, abre una ventana de terminal y escribe los siguientes comandos.

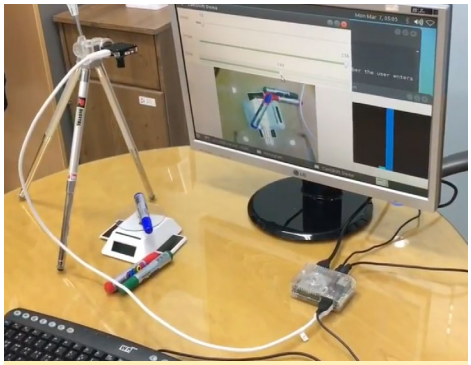


Figura 2 – Configuración del circuito de pruebas de seguimiento de objetos

```
$ sudo apt-get update && apt-get
dist-upgrade
$ sudo reboot
```

El primer comando actualiza la lista de paquetes e instala la actualización de la distribución más reciente, si está disponible. El segundo comando reinicia el ODROID. Tras actualizar la lista de paquetes y la distribución, instala OpenCV introduciendo el siguiente comando

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

A 2 de marzo de 2016, la versión más reciente de OpenCV es la 2.4.9.

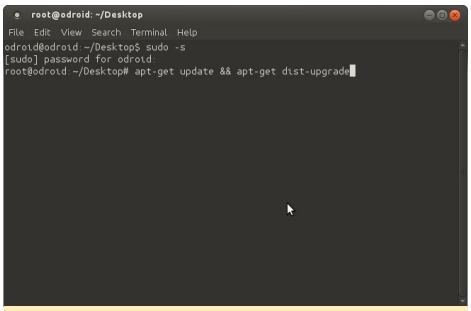


Figura 3 - Pantalla para introducir comandos en ODROID

Compilación

Nuestro ejemplo está basado en el algoritmo Camshift (Continuously Adaptive Mean Shift)), que es un tipo de algoritmo Meanshift que se utiliza para realizar seguimientos de objetos. Puedes encontrar más información sobre estos algoritmos en http://bit.ly/1pPduzS. En nuestro código, usaremos la función cvCamShift () de la librería OpenCV para habilitar el algoritmo camshift. El siguiente texto proporciona más información sobre la función cvCamShift ():

```
RotatedRect CamShift(InputArray
probImage, Rect& window, TermCri-
teria criteria)

Parameters:
probImage - Back projection of
the object histogram.  See calc-
BackProject().
window - Initial search window.
criteria - Stop criteria for the
underlying meanShift().
```

```
Returns:
      Result rectangle
```

Para descargar el archivo fuente camshiftdemo, utiliza el siguiente comando o descargar el archivo desde el navegador web visitando http://bit.ly/21ykrRF:

```
$ wget
 https://raw.githubusercontent.
com/\
  Itseez/opencv/2.4/samples/\
  cpp/camshiftdemo.cpp
```

Ahora estamos listos para compilar camshiftdemo.cpp con el siguiente comando:

```
$ g++ camshiftdemo.cpp -o demo \
  -O2 -lopencv_core -lopencv_img-
proc \
  -lopencv_highgui -lopencv_video
```

Aquí tienes los significados de las opciones del compilador:

-o demo hace un archivo binario ejecutable llamado "demo"

-O2 especifica un nivel de optimización 2. Tienes más información sobre los ajustes de optimización de g++ en http://bit.ly/1OOnopO.

-l vincula librerías externas, hemos utilizado esto para vincular cuatro librerías: OpenCV, opencv_imgproc, opencv_highgui y opencv_video.

Ejecutar la aplicación

Una vez que la oCam esté conectada al ODROID-XU4, ya estamos listos para iniciar una demo de seguimiento de objetos con el siguiente comando:

```
$ ./demo
```

La ventana de la demo CamShift tiene tres apartados: un panel de control con barras, un panel con la imagen de la cámara y un panel con un histograma. Con las 2 barras deslizantes, Vmax y Vmin, puedes controlar el nivel del color. La última barra , Smin, controla el grado de saturación. Estos controles ayudan a limitar el área de la imagen en la cual se va realizar el seguimiento de un objeto específico. Véase la explicación detallada sobre los tonos, la saturación y los valores de color en http://bit.ly/1L6R7zM.

Puedes iniciar el seguimiento de objetos haciendo clic y arrastrando con el ratón sobre la parte de la imagen de la cámara donde deseas realizar un seguimiento. La Figura 4 muestra la aplicación en ejecución al mismo tiempo que visualizamos un área seleccionada sobre una bote de zumo.

La ventana del histograma muestra los componentes de color en el área seleccionada de la imagen sobre el objeto rastreado. Puede activar y desactivar la ventana del histograma pulsando la tecla "h". También puedes cambiar el modo de vista normal a la vista de proyección posterior pulsando la tecla "b". Las figuras 6 y 7 muestran los diferentes modos de vista. Puedes encontrar más detalles

Figura 4 - Arrastrando una parte de la imagen para iniciar el seguimiento

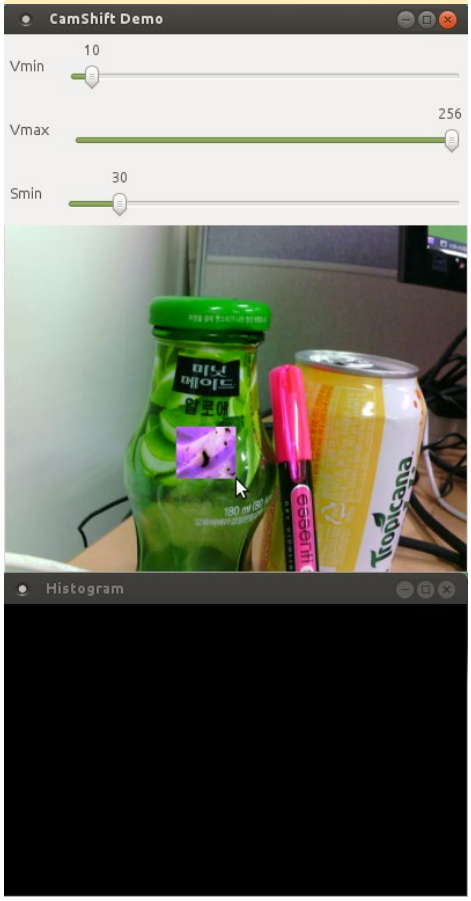


Figura 5 – El área seleccionada rastreada y el histograma de los componentes de color

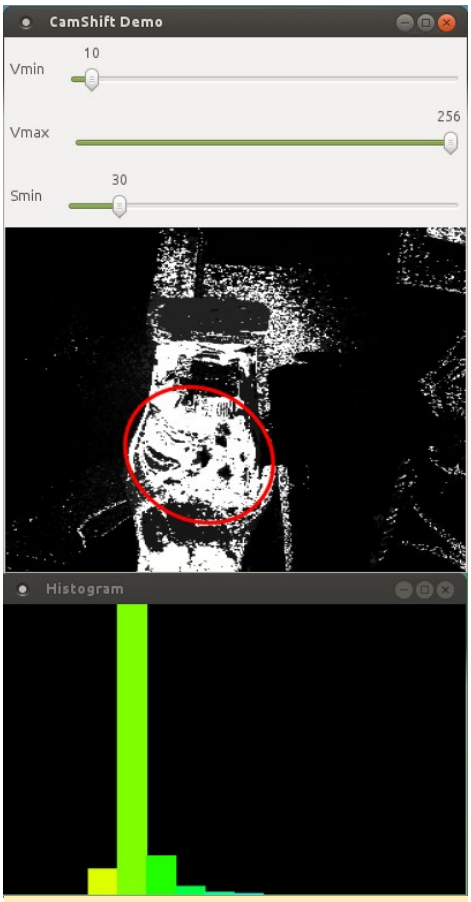


Figura 6 - Vista proyección posterior

sobre la proyección posterior en la página OpenCV en leccionada de la imagen sobre el objeto rastreado. Puede activar y desactivar la ventana del histograma pulsando la tecla “h”. También puedes cambiar el modo de vista normal a la vista de proyección posterior pulsando la tecla “b”. Las figuras 6 y 7 muestran los diferentes modos de vista. Puedes encontrar más detalles sobre la proyección posterior en la página OpenCV en http://bit.ly/1Rqc1MH.

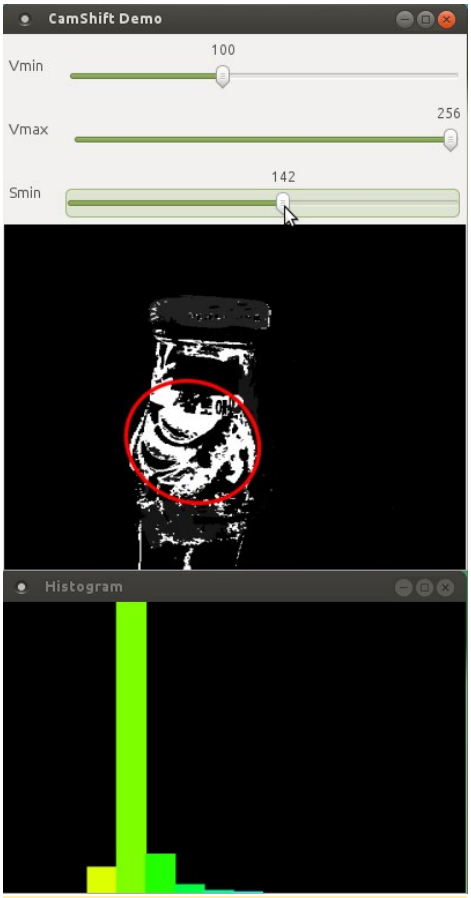


Figura 7 - Control de región interesada en la que el objeto ha sido rastreado.

Para borrar el área seleccionada, pulsa la tecla “c”. Puede iniciar el seguimiento de un nuevo objeto seleccionando otra área de la misma forma. Echa un vistazo al vídeo de http://bit.ly/21zZllS, muestra una panorámica en vivo de la demo de seguimiento de objetos que hemos abordado en esta guía usando una OCAM y el ODROID-XU4.

JUEGOS LINUX

RVGL – RE-VOLT SOBRE OPENGL

por Tobias Schaaf

Hoy quiero hablar del clásico juego de carreras Re-Volt y una versión reciente de éste llamada RVGL. Es uno de mis juegos favoritos y ahora es posible ejecutarlo en el ODROID con excelentes gráficos y una increíble banda sonora.

¿Qué es Re-Volt?

Es un juego de carreras como ningún otro. En lugar de competir con grandes y lujosos coches contra tus amigos en pistas como la famosa Nürnburgring o con Kart de carreras sobre pistas fantasticas, corres con coches de radio control (RC). Esto da al juego un toque muy singular, ya que juegas desde la perspectiva de los diminutos coches de RC en todo tipo de trazados divertidos como supermercados y jardines.

Acerca de Re-Volt

El juego original fue lanzado por Acclaim Entertainment en 1999 para PC, PlayStation, Nintendo 64 y Dreamcast. Aunque honestamente las versiones de Dreamcast y PC son las que tenían mejores gráficos y sonido. El juego era único en su época con impresionantes gráficos en 3D, una gran banda sonora, varios modos multijugador e incluso contaba con un editor de pistas para crear tus propias pistas de carreras sin necesidad de contar con herramientas especiales. A medida que desbloqueabas funcionalidades el juego llegaba a ofrecer hasta 28 coches diferentes, empiezas con pequeños coches alimentados por batería y con el tiempo llegas a conducir coches con motor de gasolina. Hay un total de 13 pistas en las que puedes correr y todas ellas proporcionan los modos opuesto, espejo e invertido. El juego también cuenta con una zona con arena donde puedes hacer trucos y acrobacias con tu coche para atrapar estrellas y desbloquear más coches.

Y como si todo esto no fuera suficiente, en Re-Volt puedes recoger pequeños rayos sobre el terreno que te proporcionan armas o mejoras al azar, tales como manchas de aceite, cohetes o una gran bola de bolos. Es similar a los juegos de Mario Kart en el que corres contra otros mientras luchas al mismo tiempo. Lo singular que hay detrás de esto es que compites desde la perspectiva de un coche de Radio Control. Esto significa que todo lo que te rodea es enorme puesto que correr alrededor de un vecindario. Un balón de baloncesto se convierte en un gran obstáculo y un coche real es 100 veces más grande. Si corres dentro de un supermercado los estantes son enormes y existen caminos y atajos ocultos por descubrir, lo cual hace que sean muy divertidos de explorar los niveles en los que corres. Personalmente me encantaba este juego, realmente era muy divertido por aquel entonces. Me encantaba jugar con los amigos en partidas LAN y me trae muy buenos recuerdos. Incluso hoy en día el juego sigue siendo un clásico atemporal.

Acerca de RVGL

RVGL es una reimplementación de Re-Volt en OpenGL para los PCs modernos. El RV-Team y especialmente Huki y jigebren, han trabajado duramente para conseguir que este viejo juego de hace 17 años pueda funcionar en plataformas modernas, y es lo suficientemente bueno como para permitirnos el lujo de hacer también una versión para los ODROIDs. Gracias a RV-Team, Huki y jigebren por esta increíble versión y la posibilidad de usarlo en los ODROIDs. Asegúrate de revisar la página del proyecto en http://bit.ly/1UEyENI. Puesto que RVGL utiliza OpenGL, gracias al empaquetador GLshim de ptit-Seb ahora podemos ejecutarlo también en los ODROIDs, ¡Tiene un aspecto realmente sorprendente!

Características

Como ya he dicho anteriormente, RVGL cuenta con unos impresionantes gráficos en 3D, incluso en la pantalla del Menú ya se pueden apreciar espectaculares gráficos con sombras y superficies reflectantes, diferente luces, etc. Fíjate en los reflejos de los coches, incluso se puede apreciar que los objetos del fondo también tienen reflejos sobre el terreno.

Las pistas son bastante singulares. Está el vecindario, el supermercado, el museo, el jardín botánico e incluso un pueblo del viejo oeste. Todas las pistas son diferentes y te hacen sentir como si estuvieras en otro mundo; un mundo donde todo lo que te rodea es enorme y simplemente hace falta explorarlo.

Correr dentro de un supermercado, ¿Te imaginas lo que ocurre si entras dentro de la cámara frigorífica?

Como ya he mencionado cuentas con diferentes armas que puedes recoger durante la carrera. Por ejemplo, están los cohetes que te lo encuentras de uno en uno o en una pila de tres. También hay una bola de bolos con la que puede ralentizar al enemigo, así como una mancha de aceite que hace que sea muy difícil o imposible controlar el coche. Las bombas de agua siempre aparecen de tres en tres, pero no cuentan con el seguimiento automático de objetivos como los cohetes. Existe una especie de explosión de energía que lanza a todo el mundo fuera de la pista, o una batería extra que hace aumentar tu velocidad durante un corto período de tiempo. Todas éstas armas son sólo algunos de los extras se puedes coger durante las carreras.

Figura I - Menú principal de RVGL ejecutándose de forma nativa

Figura 2 - Un imagen del juego en el nivel del supermercado

RVGL también ofrece los modos multijugador originales. Esto significa que puede jugar en el modo de pantalla dividida o con hasta cuatro jugadores en tu ODROID, o incluso jugar con otros jugadores a través de LAN o Internet, lo cual es realmente impresionante. Me pregunto qué es lo que hace falta para echar una partida LAN con ODROID justos. Jugar a juegos tan impresionantes como Re-Volt en LAN sobre un ODROID suena bastante divertido.

El juego soporta tanto teclados como joystick. Gracias a SDL2, la mayoría de joystick funcionan de serie o se configuran con facilidad, de modo que es aún más fácil jugar a este tipo de juegos con tus amigos en el mismo ODROID.

Figura 3 – Otro imagen del juego en el nivel del supermercado

Instalar RVGL

Para que este juego funcione necesitas los archivos originales del CD del juego o cualquier otra copia del juego. Como de costumbre, puede instalar este juego desde mi repositorio usando usando mis imágenes con el siguiente comando:

```
$ apt-get install rvgl-odroid
```

También es necesario el empaquetador GLshim de mi repositorio, que debería instalarse de forma automática como una dependencia. A continuación, coloca los archivos de datos de tu CD en una carpeta llamada ".rvgl" (ten en cuenta el "punto" delante de rvgl) en tu carpeta de inicio. Aquí es donde el juego buscará sus archivos de datos. Si no existe la carpeta, créala o inicia el juego una vez. La carpeta será visible tras realizar alguno de estos dos pasos.

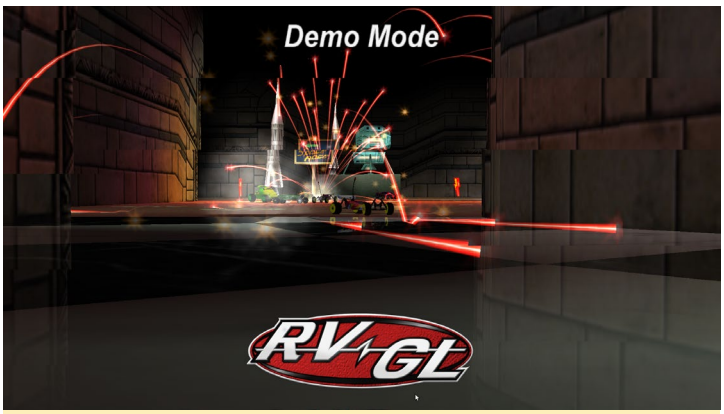


Figura 4 – Se puede usar un cohete para atacar a un coche oponente

Ahora puedes copiar las carpetas del juego original en esa carpeta. Cuando inicies RVGL, puede que tarde un tiempo en cargar, se paciente y esperar un poco. Se están realizando varias tareas en un segundo plano, como comprobar si todos los archivos tienen los nombres correctos, y cambiardolos en los

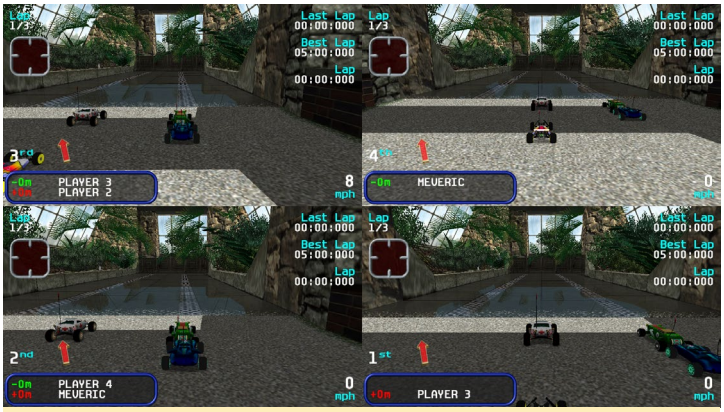


Figura 5 - RVGL en modo multijugador con pantalla dividida

casos que no sea así.

Si quieres banda sonora, copia las pistas de audio del juego original en una carpeta llamada " redbook ". Los archivos deben ser nombrados desde track02 al track15 y estar en formato .ogg, .mp3, .flac, o .wav. El juego es difícil de conseguir hoy en día, ya que existen algunas cuestiones legales al respecto. Algunos consideran que es abandonware, puesto que no está claro quién tiene los derechos en este momento. Tienes más información sobre este tema en http://bit.ly/1RfqRUj.

Una vez copiados todos los archivos del juego en la carpeta del juego, ya está listo para jugar. El juego tarda tiempo en iniciarse, de modo que se paciente si ves que no sucede nada al hacer clic en el icono del juego.

Reflexiones finales

Me gusta mucho este juego, creo que es mejor que Mario Kart o que muchos otros juegos de carreras que he probado. Las pistas son muy divertidas y con los modos inverso y espejo parece que existen más niveles. El modo multijugador es especialmente divertido, y tener la posibilidad de jugar en pantalla dividida con el mismo ODROID hace que sea incluso mejor.

CUT THE ROPE 2

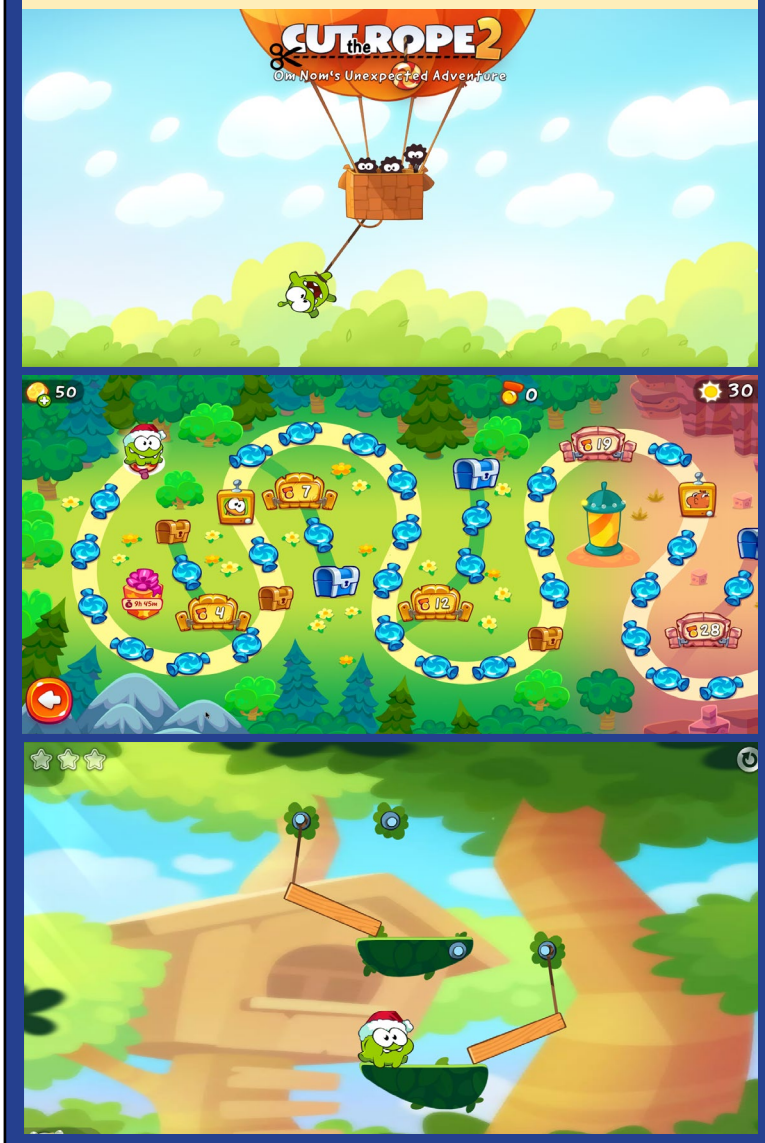
AYUDA A NOM-NOM A CONSEGUIR SUS CARAMELOS

por Rob Roy

Cut the Rope 2 es la continuación del popular juego Cut the Rope, que es un gran juego que desagía la física donde intentas ofrecer caramelos a un pequeño monstruo llamado Nom Nom resolviendo puzzles de gravedad. Es un juego gratuito en el que puedes descubrir premios ocultos y desbloquear niveles secretos a medida que avanzas. La animación es excelente, y seguro que el personaje principal te hará sonreír. Puedes descargarlo desde la tienda Google Play visitando el siguiente enlace:

http://bit.ly/QutCFB

A Nom Nom le encanta los caramelos, y tú puedes ayudarle a conseguir un montón.



KISMET

MONITORIZAR UNA RED INALAMBRICA

por Adrian Popa

La página web WiGLE (http://wigle.net) tiene una gran base de datos de puntos de acceso inalámbricos recopilados por personas que hacen Wardriving. El Wardriving se puede definir como el acto de conducir alrededor de un área en busca de una red inalámbrica sin protección. El número total de puntos de acceso en la base de datos de Wigle es de alrededor de 120 millones. Cada punto de acceso en la base de datos contiene información sobre su SSID, ubicación geográfica, dirección MAC y el tipo de cifrado. Como ya habrás observado, vivimos en un mundo interconectado que se vale del Wi-Fi para conectar todos los dispositivos al mismo tiempo. Es por eso que debes prestar atención a la configuración de tu Wi-Fi personal y asegurarte de que es segura. Esta serie de artículos será tu guía a través del mundo de la seguridad inalámbrica, aprenderás las técnicas que utilizan los malos para penetrar tu red y así poder protegerte mejor. ¡Lo mejor es que todo se hace con tu apreciado ODROID! Aunque Kali Linux es la distro preferida para análisis de seguridad, nosotros vamos a trabajar con Ubuntu 14.04 de HardKernel (o posterior).

Antes de empezar, quiero dejar algo muy claro. Entrar en la red de alguien sin permiso es un delito, y está sancionado por la ley en muchos países. Todas estas pruebas han sido realizadas en condiciones de laboratorio y con el consentimiento del propietario de la red. Si tienes alguna duda o no está seguro de la legislación de tu zona, por favor consulta a un abogado y revisa la normativa.

Conocer Kismet

Una vez conseguido el permiso, lo primero que hay que hacer en cualquier prueba de penetración es recopilar información sobre tu objetivo. En el contexto de las redes WiFi, esto significa que tienes un receptor que te permite escuchar las transmisiones de todos los puntos de acceso que te rodean. Existen varias herramientas que te permiten hacer esto, pero la mejor es Kismet.

Kismet es un detector de redes inalámbricas, un olfateador y un sistema de detección de intrusiones para Linux con una interfaz en modo texto. Algunas de sus características incluyen la monitorización 802.11, conexión PCAP, arquitectura cliente-servidor y salida de datos XML para integración con otras herramientas.

Utilizaremos Kismet para recopilar una lista de puntos de acceso situados dentro del alcance de nuestro receptor. Desde la lista de punto de acceso, podemos comprobar si es posible extraer información de cualquiera de ellos. Para instalar Kismet, simplemente tiene que ejecutar el siguiente comando en un sistema Ubuntu:

```
$ sudo apt-get install kismet
```

Selecciona “Yes” durante la instalación de Kismet cuando se te pregunte, y escribe tu nombre de usuario que se añadirá al grupo kismet. Para utilizar Kismet, necesitas tener una interfaz Wi-Fi disponible con la que poder monitorizar. Yo probé Kismet con varios módulos Wifi ODROID de Hardkernel:

Module 3 Wifi

http://bit.ly/22nyxra

Este módulo está basado en el conjunto de chips Realtek RTL8188. Funciona bien, pero no ofrece los mejores resultados en comparación con los otros módulos que probé.

Module 4 Wifi

http://bit.ly/1FprqF5

El Módulo 4 utiliza el conjunto de chips Ralink RT5572, que es compatible con 5 GHz, pero también funciona a 2,4 GHz..

Module 0 Wifi

http://bit.ly/1M4LdiC

El Módulo 0 está basado en el Ralink RT5370N y funciona sorprendentemente bien para su tamaño.

Ten en cuenta que mientras que la interfaz inalámbrica esté en modo monitor, no podrás utilizar la interfaz para el tráfico normal. Es similar al modo promiscuo del Ethernet. Esto significa que si no cuentas con una conexión de red cableada en tu ODROID, serás aislado de la red cuando la interfaz de conexión entre en modo monitor.

Modo monitor con aircrack-ng

Antes de empezar a utilizar Kismet, es necesario configurar una interfaz de monitorización que estará conectada a tu tarjeta WiFi. Para ello, necesitarás la utilidad airmon-ng, que forma parte del paquete de aircrack-ng. Aircrack-ng se puede instalar escribiendo lo siguiente en el terminal:

```
$ sudo apt-get install aircrack-ng
```

Para comprobar si el controlador inalámbrico es compatible con el modo monitor, puedes ejecutar el siguiente comando, la Figura 1 muestra el resultado de este comando que ha sido capaz de encontrar la interfaz Wi-Fi 'wlan2':

```
$ sudo airmon-ng
```

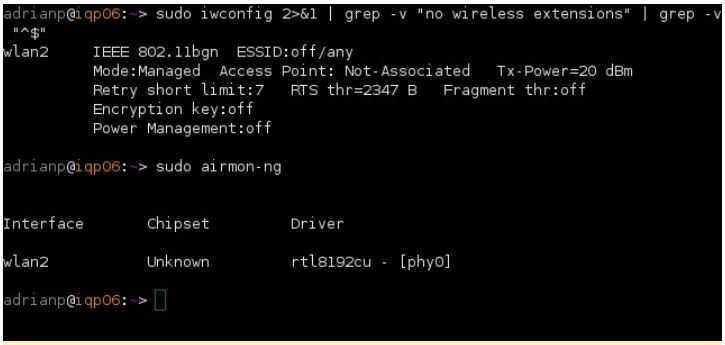


Figura 1 – Hardware compatible

Activar el soporte para modo monitor en el Módulo 3 Wifi

Si no ves tu dispositivo en la lista, el hardware o el controlador no son compatible con el modo monitor. Si tienes el módulo 3 WiFi de ODROID y no aparece en la lista, significa que tienes que actualizar el driver wifi. Por defecto, la versión de Ubuntu de Hardkernel carga el driver 8192cu, que es la variante de código abierto del driver y tiene menos funcionalidades, sólo puede conectarse como cliente. El driver rtl8192cu procedente de Realtek añade soporte para modo monitor y punto de acceso.

Por desgracia, este es el paso en el que necesitas algo de conocimientos técnicos. Otra posibilidad es comprar el módulo 0 Wifi o el Módulo 4 Wifi, que funcionan con los drivers por defecto. Para el Módulo 3, tendrás que volver a compilar el kernel para añadir el módulo rtl8192cu. Puede seguir cualquier guía de compilación de kernel, como la que aparece en ODROID Magazine en http://bit.ly/1RlIm7j o la que tienes en http://bit.ly/1NVRprY. En la configuración del kernel, tienes que editar los elementos que se muestran en la Tabla 1.

Ten en cuenta que algunos de los ajustes anteriores pueden estar ya seleccionados en tu kernel, ya que los ajustes de configuración por defecto difieren de un dispositivo a otro. Una vez compilado el kernel correctamente, utiliza los comandos necesarios para instalarlo. Te recomiendo que hagas una copia de seguridad del kernel y de los módulos existentes. Una vez que todo haya sido compilado e instalado, reinicia tu ODROID. Para tu comodidad, aquí te dejo los pasos necesarios para compilar e instalar un kernel para el ODROID C1. Se supone que ya tienes configurado tu entorno de compilación.

Action	Path	Config Variable
Set your custom kernel name (e.g. -8192cu)	General setup -> Local version	CONFIG_LOCALVERSION
Disable	Networking support -> Wireless -> cfg80211	CONFIG_CFG80211
Enable built-in	Backport Linux -> cfg80211 wireless extensions compatibility	CONFIG_BACKPORT_CFG80211_WEXT
Enable as module	Backport Linux -> Wireless LAN -> Realtek rtlwifi family of devices	CONFIG_BACKPORT_RTL_CARDS
Enable as module	Backport Linux -> Wireless LAN -> Realtek rtlwifi family of devices -> Realtek RTL8192CU/RTL 8188CU	CONFIG_BACKPORT_RTL8192CU
Enable built-in	Backport Linux -> Wireless LAN -> Realtek rtlwifi family of devices -> Debugging output for rtlwifi driver family	CONFIG_BACKPORT_RTLWIFI_DEBUG

Tabla 1 - cambios en la configuración del kernel que son necesarios para utilizar el módulo 3 WiFi

```
$ git clone --depth 1 --single-branch -b odroidc-
3.10.y https://github.com/hardkernel/linux
$ cd linux
$ make odroidc_defconfig
$ make menuconfig
$ make -j 4 uImage dtbs modules
$ sudo cp arch/arm/boot/uImage \
 arch/arm/boot/dts/*.dtb /media/boot
$ sudo make modules_install
$ sudo make firmware_install
$ kver=`make kernelrelease`
$ sudo cp .config /boot/config-${kver}
$ cd /boot
$ sudo update-initramfs -c -k ${kver}
$ sudo mkimage -A arm -O linux \
 -T ramdisk -a 0x0 -e 0x0 -n initrd.img-${kver} -d
initrd.img-${kver} uInitrd-${kver}
$ sudo cp uInitrd-${kver} /media/boot/uInitrd
```

Ten en cuenta que si utilizas GCC 5.x, el kernel fallará cuando intentes arrancar el ODROID-C1. Necesitas añadir el parche de http://bit.ly/1QZlWWi, o compilar con GCC 4.8. Si tienes cualquier problema con la compilación del kernel, pue-des pedir ayuda en http://bit.ly/1RlJqbn.

Modo Monitor - Práctica

En esta sección, vamos a utilizar "wlan0" como ejemplo de interfaz Wifi que monitorizaremos, aunque el nombre de tu interfaz seguramente será diferente. Ahora que tu tarjeta de red

es es compatible con el modo monitor, puedes crear una interfaz monitor. Recuerda que el módulo Wifi no puede funcionar en modo monitor y en modo cliente al mismo tiempo. También puede especificar un número de canal o una frecuencia, que la antena ajustará. Puedes obtener una lista de canales y frecuencias ejecutando el siguiente comando:compatible con el modo monitor, puedes crear una interfaz monitor. Recuerda que el módulo Wifi no puede funcionar en modo monitor y en modo cliente al mismo tiempo. También puede especificar un número de canal o una frecuencia, que la antena ajustará. Puedes obtener una lista de canales y frecuencias ejecutando el siguiente comando:

```
$ iw phy0 info
```

Para iniciar la monitorización con airmon-ng, utiliza el siguiente comando, teniendo en cuenta que debes cambiar "wlan0" por la interfaz de tu sistema:

```
$ sudo airmon-ng start wlan0
```

```
adrianp@iqp06:~> sudo airmon-ng start wlan2


Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
645     avahi-daemon
646     avahi-daemon
1070    dhclient
1263    NetworkManager
1479    wpa_supplicant


Interface       Chipset         Driver

wlan2           Unknown         rtl8192cu - [phy0]
                                (monitor mode enabled on mon0)
```

Figura 2 - Modo monitor activado para wlan2

Deberías ver un mensaje indicando que se ha creado una interfaz de monitor llamada "mon0". Las versiones posteriores de airmon-ng usan una convención de nombres algo diferente, en este caso la interfaz pasaría a llamarse wlan0mon. Ahora puedes utilizar la interfaz monitor para hacer cosas interesantes, como capturar paquetes, utilizar programas como Wireshark, tcpdump o Kismet. Si aparecen errores mientras esté activado el modo monitor, debes detener los procesos registrados por airmon-ng e intentar activar de nuevo el modo monitor.

Si te gustan los retos, podrías llegar a hacer lo mismo bajo Android. Necesitarás compilar el driver correcto para tu kernel Android y configurar aircrack-ng directamente en Android. No funciona pero si estás interesado, tienes más detalles en http://bit.ly/1S6XAtR.

Los protocolos 802.11 utilizan datagramas de capa 2 para encapsular los datos que se transmiten. Existen 3 tipos de secciones en función de su finalidad:

- Gestión, que gestiona la autenticación, asociación, análisis, desautentificación, desvinculación y los avisos.
- Control, que facilita el intercambio de datos entre estaciones. Incluye reconocimiento, autorización para enviar y la petición de envío.
- Datos, que encapsula la fecha real en la que el usuario final transmite y recibe.

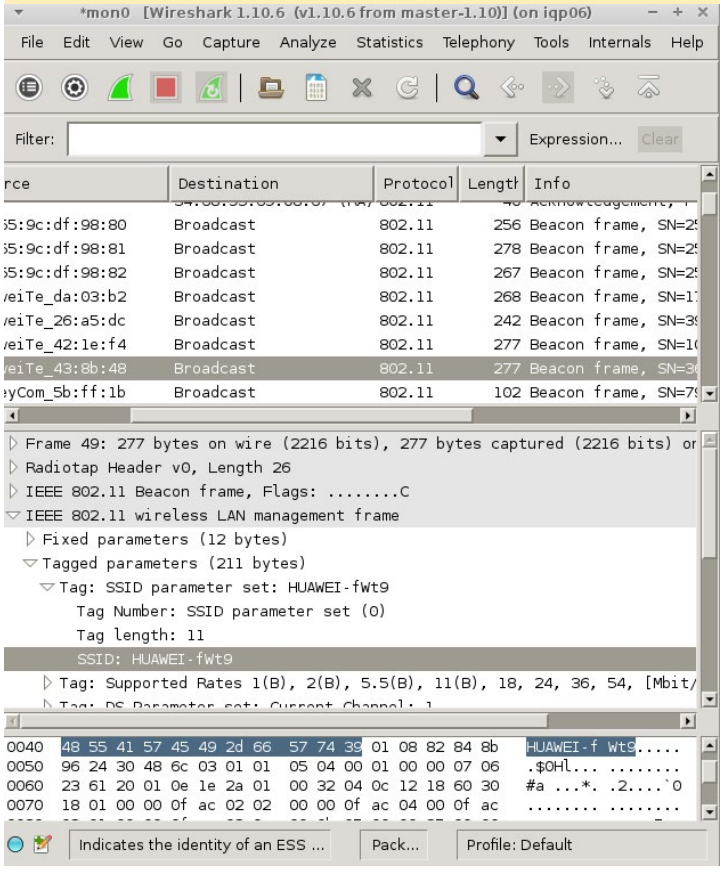
Si haces capturas de paquetes en una red inalámbrica sin uso, es probable que vea una gran cantidad de tráfico de gestión, como son avisos y asociaciones, y muy poco tráfico de datos o de control. El tráfico de datos es intermitente por naturaleza, especialmente durante la navegación web. Puedes hacerte una idea de lo que están activadas las estaciones base y los clientes basándote en el tráfico de gestión. Una captura de paquetes de prueba con los tres tipos de secciones para analizar en Wireshark la tienes disponible de http://bit.ly/22ouM8k.

Analizando los paquetes capturados, te darás cuenta de que el control y la gestión del tráfico se envía sin encriptar mientras que el tráfico de datos se codifica en función del tipo de cifrado (WEP o WPA).

Volver a Kismet

Ahora que tiene una interfaz de monitor, puedes finalmente poner en marcha Kismet:

Figura 3 - Wireshark muestra una captura de paquetes del tráfico de gestión 802.11



```
$ kismet
```

La primera vez que ejecutes Kismet, tendrá que responder a algunas preguntas, como por ejemplo si determinados colores son visibles en pantalla. Si deseas que el servidor Kismet se ini-

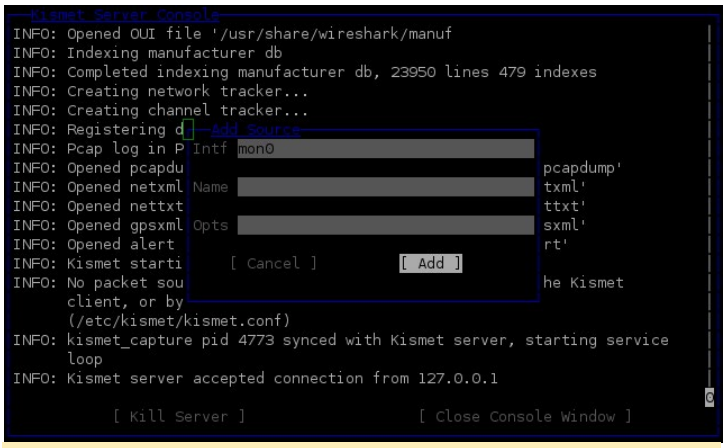


Figura 4 – Añadiendo la interfaz mon0 en Kismet

cie automáticamente, debes responder “yes”. inicie automáticamente, debes responder “yes”. Una vez que se inicie el servidor Kismet, te indicará que no dispone de una fuente de paquetes definida y te preguntará si desea agregar una interfaz, a lo que debe responder “ Yes”. En la siguiente ventana, escribe “mon0” en el campo “Intf”.

Ahora puedes cerrar la ventana de registro log y deberías tener recopilados los puntos de acceso. Kismet funciona realizando “saltos” entre todos los canales y escuchando la información de gestión. Kismet creará una lista de los puntos de acceso y de los clientes que detecta el monitor wifi. Al mismo tiempo que kismet crea esta lista, captura tráfico. Si dejas que Kismet se ejecute durante mucho tiempo, es muy posible que detecte una gran cantidad de redes, ¡incluso si lo mantienes fijo en un sitio!

Si ves un error en la ventana de registro que dice “ Packet source ‘mon0’ failed to set channel 1: mac80211_setchannel() could not set channel 1/2412 on interface ‘mon0’ err -25”, es posible que necesites eliminar la interfaz de red con la que trabajas (la misma que utilizaste para crear la interfaz de monitor). La interfaz se puede borrar con el siguiente comando:

```
$ sudo iw dev wlan0 del
```

Para navegar por la interfaz de Kismet, utiliza ALT + K para abrir el menú. Haz uso de las flechas, TAB y la barra espaciadora para desplazarse y seleccionar los elementos. Si deseas mostrar más columnas, puedes activarlas desde Kismet -> Preferences -> Client Columns and Kismet -> Preferences -> Network Columns options. También puede usar el menú Sort para cambiar el orden de acuerdo a tus preferencias.

La ventana principal de Kismet se divide en varias secciones, como puedes ver en la Figura 5. En la parte superior, tienes la lista de redes (1). Si seleccionas una red de la lista, Kismet mostrará a continuación la lista de clientes asociados a dicha red (2). Si tienes un GPS conectado a tu sistema, Kismet también mostrará los datos de localización, velocidad y altitud (3). La siguiente sección es un gráfico que muestra la relación entre los paquetes y los datos (4). Y por último, la parte inferior muestra información de registro, como son errores, análisis, etc (5). Los Colores identifican el tipo de cifrado. Naranja es

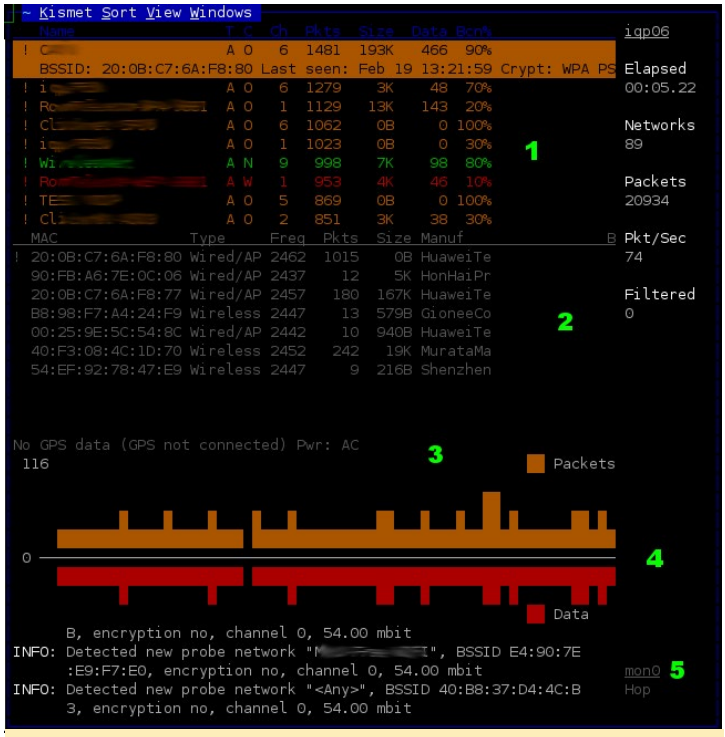


Figura 5 - Escaneo de Kismet

WPA-PSK, WEP se identifica con el color rojo y verde indica que se trata de una red abierta.

Recopilar los datos es una cosa, pero darles sentido es otra cuestión muy distinta. Por defecto, Kismet genera registros log en el directorio de trabajo actual. Si dejas Kismet ejecutándose por mucho tiempo, es posible que te quedes sin espacio en disco. Una hora de escaneo me llego a generar alrededor de 150MB de registros log cuando escaneé mi zona. Es mejor si se utiliza un dispositivo independiente, como una memoria USB o recurso compartido en red para guardar los registros log, lo cual te evitará accidentes causados por la reducción de espacio en disco. Además, las escrituras frecuentes en las tarjetas SD no son muy buenas para la salud de las mismas. Los registros log que se generan contienen diversa información, como se muestra en la Tabla de la siguiente página

Existes muchos script analizadores disponibles en Internet que puede analizar estos registros y generar informes en diferentes formatos. Por ejemplo, Kismet Log Viewer (http://bit.ly/1pP2pyF) puede generar una página resumen HTML de tus datos.

Log name	Information collected
Kismet-yyyymmdd-hh-mm-ss-1.alert	Fingerprints for intrusion detection systems
Kismet-yyyymmdd-hh-mm-ss-1.gpsxml	Geographical data associated to the networks
Kismet-yyyymmdd-hh-mm-ss-1.nettxt	A summary in text format of all networks and clients seen. Details include SSIDs, packets sent and received and encryption types.
Kismet-yyyymmdd-hh-mm-ss-1.netxml	Same information as above in easier to parse XML format
Kismet-yyyymmdd-hh-mm-ss-1.pcapdump	All captured packets in pcap format (you can open them with Wireshark)

Tabla 2 - Información del registro log

Aunque logras recoger muchos megabytes de datos, todavía hay una gran cantidad de datos que se pierden. Esto se debe a que Kismet realiza saltos de canal y escucha en un único canal al mismo tiempo. Si lo que quieres es capturar un tráfico específico, tienes dos opciones:

1. Reajustar tu interfaz monitor al canal deseado e impedir que Kismet realice saltos de canal(Alt+K -> Config Channel… -> mon0 -> Lock -> Número de canal)
2. Si dispones de más adaptadores, puedes configurar hasta tres adaptadores inalámbricos en modo monitor con canales que no se solapen (por ejemplo, 1, 6 y 11) y Kismet registrará datos de todas las interfaces de monitor al mismo tiempo:

Figura 6 – El Módulo 3 Wifi es más grueso que el puerto USB y requiere de un cable para poder conectarlo


```
$ sudo airmon-ng start wlan0 1
$ sudo airmon-ng start wlan1 6
$ sudo airmon-ng start wlan2 11
```

Ahora debería tener 3 interfaces de monitorización: mon0, mon1 y mon2. Dentro de Kismet, pulsa Alt + K -> Add source… para añadir todas las interfaces, así como Alt + K -> Config Channe... para bloquear cada uno de sus respectivos canales.

Conclusión

Kismet es una potente herramienta de reconocimiento de redes pasiva usada para capturar datos de puntos de acceso WiFi. Analizando estos datos, puedes aprender mucho sobre tus vecinos y sus patrones de uso de la red. Por ejemplo, yo he descubierto un par de puntos de acceso WEP alrededor de mi casa. Además, identifique algunos puntos de acceso móviles de varias compañías de taxis que tenían coches en el barrio. Aunque no lo parezca, estoy recogiendo una SSID "WirelessNet" abierta tanto en lugares de trabajos como en hogares que se parecen a un honeypot, que probablemente sean utilizados para atraer piratas informáticos locales. El tráfico de red registrado también puede ser potencialmente descifrado cuando obtengas la clave de red. Tener un ODROID alimentado por una batería te proporciona una gran flexibilidad, ya que es posible ocultarlo cerca de tu objetivo, como en un buzón de correos y recuperarlo después de un día de recopilación de datos. Sin embargo, el mejor uso que puedes dar a Kismet es el de tu propia protección, husmeando en tus dispositivos personales para ver qué tipo de datos te puede robar. Veremos esto y mucho más en artículos posteriores. Por favor, dirije sus preguntas, comentarios o sugerencias al hilo del foro ODROID en http://bit.ly/1o2Vr7I.

CONOCIENDO A ODROIDIAN

ADRIAN POPA (@MAD_ADY), AFICIONADO A LOS SBC

editado por Rob Roy

O bien hace un poco de turismo en Lanois, Francia o bien ayuda a sus niños a explorar el universo, Adrian es un super ingeniero que ayuda a nuestras comunidades ODROID con sus amplios conocimientos.

Háblanos un poco sobre ti

Soy Adrian Popa y utilizo el apodo de "mad_ady " en diversas comunidades y foros de Internet. Tengo casi 34 años y vivo en Bucarest, Rumania con mi maravillosa esposa Irina y nuestros dos hijos: Teo (4 años) y Matei (9 meses). Trabajo como ingeniero de telecomunicaciones. Durante los últimos 10 años, he estado contratado por un gran proveedor de servicios de Internet rumano desarrollando aplicaciones personalizadas y automatizando redes de datos. También estoy interesado en la seguridad, y me encanta trapichear con las cosas para ver si y cómo se rompen.

Soy curioso y me encanta descubrir y aprender cosas nuevas. Durante los últimos 5 años, he sido miembro de la comunidad WDLXTV (http://bit.ly/1Uh4TD4), contribuyendo a la creación de un firmware personalizado para un reproductor multimedia MIPS de código cerrado desarrollado por Western Digital. También tengo pasión por el rediseño de viejos aparatos para que hagan cosas para las cuales no fueros creados. Estoy orgulloso de decir que he rooteado todos los dispositivos que poseo, como mi reproductor de medios, Smart TV, cámaras IP y NAS.

¿Cómo fueron tus inicios con los ordenadores?

Mi primer "ordenador" fue el clone ZXSpectrum de Europa del este de mi hermana (http://bit.ly/1RwAg5J). Lo utilice sólo para jugar con la única cinta de casete que tenía. Más tarde, a los 12 años, mis padres nos compraron un 486-SX con 4 MB de RAM y un disco 420MB. Tuve la suerte de tener un único juego instalado llamado Prehistorik-2 (http://bit.ly/1T4KbF4) y me quede en el nivel 2. Esto trunco el jugador que llevaba dentro y me permitió empezar a explorar el equipo y aprender a utilizar MS-DOS y Windows 3.1. Me convertí en un adicto a la línea de comandos cuando supe cómo utilizar arj:arj -a -v1440 -r. Después de aprender todo lo que pude de Windows 95/98, que no fue difícil para mi gusto, me mudé a Linux RedHat 6 en 1999 y comencé un lento pero emocionante aprendizaje. He jugado con la mayoría de las distribuciones y me irritaba bastante el infierno de dependencias de RPM, lo cual me llevó a Mandrake. Luego probé Debian y estuve un corto período experimentando con Gentoo, que abandoné tras compilar X11, QT y StarOffice durante 3 días seguidos, mientras que simplemente trataba de actualizar Pidgin. Actualmente estoy usando Ubuntu para trabajar y jugar. Aunque tarde bastante en utilizar Android, me interesé por las placas de Linux embebidas en 2010 cuando conseguí un reproductor multimedia WDTV Live. Formar parte de una pequeña pero entregada comunidad me ayudó a aprender y entender muchas cosas.

¿Qué te atrajo de la plataforma ODROID?

Me enteré del ODROID por un compañero desarrollador, que tenía el modelo U3. Plantee a mi jefe comprar un par de placas ODROID para desarrollar prototipos de algunos experimentos de calidad de redes en los que estábamos trabajando. Me gustaba el hecho de que las placas fueran fáciles de usar y pudieran servir como servidor/ordenador de aplicación general ocupando muy poco espacio. Viniendo de plataformas embebidas de código cerrado, tenia una cierta preocupación por no encontrar desafíos a los que enfrentarme, siendo todo muy aburrido ya que las placas venían preconfiguradas y listas para funcionar. Estoy muy contento de ver que aún existen peculiaridades en las que se están trabajando, y que se pueden hacer grandes cosas si el proveedor y el fabricante del los chip trabajan junto con la comunidad.

¿Cómo utilizas tus ODROIDs?

Los ODROIDs que utilizo para proyectos relacionados con

el trabajo tienen múltiples usos, generalmente para proyectos de red y software, como evaluaciones iperf y mediciones de latencia. Tengo un C1+ que utilizo como banco de pruebas para prototipos, que normalmente es muy inestable. Tengo el nuevo C2 que lo estoy convirtiendo en reproductor multimedia ya que el H265 parece estar ganando terreno, pero también será mi ordenador de trabajo principal, ya que habitualmente trabajo de forma remota a través de SSH. Mi plan es empezar a aprender electrónica a un nivel práctico, por lo que el kit Tinkering me será muy útil. ¡Sólo espero no llegar a producir una fumata negra! En el futuro, cuando los niños sean mayores, tengo la intención de usar los ODROIDs como plataformas de aprendizaje para enseñarles programación y a hacer modificaciones de hardware.

¿Cuál es tu ODROID favorito?

Puesto que todavía soy nuevo en esto de los ODROIDs, no he podido probar los modelos más antiguos, pero por ahora el que más me gusta es el C1+ porque es silencioso, potente y cuenta con un excelente soporte de software. He utilizado también el XU3, pero tuve que limitar la frecuencia máxima a 600 MHz para evitar que el ventilador arrancara. Me hace de servidor de 8 núcleos, aunque me duele tenerlo limitado de esta forma. He recibido recientemente el C2, necesita un poco de tiempo hasta que el software de 64 bits madure lo suficiente.

Tu experiencia en hardware y software es muy bien recibida en tus artículos de ODROID Magazine y en tu blog personal. ¿Cómo llegaste a convertirte en un experto?

Lo más importante que he aprendido durante mi carrera de ingeniería es cómo descomponer un gran problema en problemas más pequeños y más manejables. Además, si quieres aprender o resolver algo, por lo general tienes que documentarte y tratar de simplificar las cosas tanto como puedas. También ayuda si documentas tus progresos. Puesto que tiendo a ser una persona olvidadiza, normalmente escribo los pasos que he seguido para llegar a determinadas soluciones. Me he encontrado que años más tarde, cuando me he enfrentado a un problema similar, me he dado las gracias a mi mismo por haberlo documentado en aquel momento. El haber trabajado con Linux durante más de 16 años me ha llevado a enfrentarme a un montón de cuestiones y problemas. La persistencia sólo conduce a la experiencia.

¿Qué innovaciones te gustaría ver en futuros productos de Hardkernel?

El diseño y el desarrollo de una plataforma embebida que sea rentable y cubra las necesidades de la mayoría de los usuarios debe ser una tarea repleta de dificultades. Es normal que los dispositivos ARM sean menos flexibles y estén más limitados que los PC convencionales debido al modo en el que son desarrollados (dispositivos de aplicación general frene a placas embebidas especializadas). De modo que no voy a pedir conexiones SATA y PCI, aunque me gustaría ver un ODROID con refrigeración pasiva que tuviese puertos USB3 y Ethernet Gigabit, lo cual lo convertiría en un espectacular NAS. Sé que Hardkernel está buscando formas para que el XU4 pueda refrigerarse de un modo pasivo.

¿Qué aficiones e intereses tienes aparte de los ordenadores?

Me encanta explorar nuevos lugares, pasear en bicicleta, leer y ver películas con mi esposa. Me gusta aprender cosas nuevas sobre la ciencia, el espacio y la tecnología. A los niños les encanta mantenerme ocupado en casa, por lo que se puede decir que también soy un experto en Lego.

¿Qué consejo le daría a alguien que quieren aprender más acerca de la programación?

Aprender programación hoy en día es mucho más fácil de lo que era hace 20 años. Un nuevo programador tiene varios lenguajes de alto nivel donde elegir, con un montón de librerías y tutoriales. Sin embargo, tener tantas opciones puede llegar a ser desalentador: ¿cuál es el mejor lenguaje para aprender y qué se puede hacer con él? Yo recomiendo empezar con Python, es versátil, fácil de aprender y tiene un montón de librerías útiles. Soy un fan de Perl, porque lo suelo usar para hacer cosas en mi día a día.

No tengas miedo de escribir código malo al principio, es normal y te ayuda a aprender de tus errores. Es fundamental tener un proyecto o meta en el que estés interesado para mantenerte motivado. También te ayudará el hecho de escribir el código que has aprendido, para que no se te olvide. En lugar de copiar y pegar ejemplos, deberías escribirlo, te ayudará a aprender la sintaxis. Elije las funciones y variables más importantes y asegúrate escribir comentarios en el código. No deberías escribir lo que hace el código, puesto que un programador puede simplemente leer el código, en su lugar describe por qué el código hace lo que hace. Esta es una información muy valiosa cuando dentro de dos meses se te haya olvidado todo. También deberías buscar librerías que ya hagan lo que tú quiere hacer, en la mayoría de los casos, los problemas ya han sido resueltos.

Una vez que hayas aprendido los conceptos básicos de la programación, puedes aprender fácilmente la sintaxis de un lenguaje diferente, te debería llevar 1-2 semanas, lo cual te permite explorar áun más. Los conceptos de la programación orientada a objetos, modelo-vista-controlador, y otros paradigmas de la programación suelen ser los mismos en todos los lenguajes. Hay una diferencia entre los lenguajes de programación de bajo nivel y los de alto nivel. El primero requiere que tengan un mayor conocimiento de cómo los microprocesadores funcionan y como la memoria se alinea y se asigna, que se debe utilizar al escribir programas eficientes. Los últimos esconden muchas complejidades de un micro-ordenador y te permite sacar el trabajo de manera más rápida, con menos dolores de cabeza. Hay grandes recursos online que te pueden a ayudar a empezar, como Coursera y Codecademy, ¿A qué estás esperando?