Thundroid • Servidor Minecraft • Recuperar eMMC • Juegos Linux • FNA

ODROID Magazine

Año Cinco
Num. #56
Ago 2018

KeePass

GESTIONA DE FORMA SEGURA TUS CONTRASEÑAS ONLINE USANDO SOFTWARE DE CODIGO ABIERTO PARA ODROID

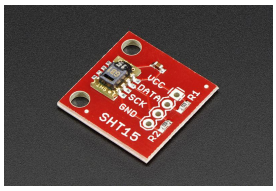
DETECCION DE OBJETOS EN VIDEO EN DIRECTO:
USANDO ODROID-XU4 CON GSTREAMER

PROGRAMACION:
GUIA DE PRINCIPIANTES PARA EL ODROID-GO

Ejecutar Modernos Juegos FNA sobre la Plataforma ODROID

August 1, 2018

FNA es una reimplementación de código abierto de XNA. XNA es un entorno de juegos creado por Microsoft Corp., y es bastante popular. Se han escrito excelentes juegos con XNA (solo para MS Windows) y luego se han exportan a Linux y MacOS usando FNA y/o MonoGames. Hagamos que funcione ▶️



Lectura de Temperatura y Humedad desde un Sensor SHT15: Introducción a la Interfaz GPIO

August 1, 2018

El objetivo de este proyecto es utilizar un ODROID para leer los datos de temperatura y humedad desde un sensor SHT15.

KeePass: Administrador de Contraseñas

August 1, 2018

Existen muchos administradores de contraseñas, pero me centraré en un programa de código abierto muy bien fundamentado llamado KeePass.

Conceptos Básicos de BASH – Parte 4: Variables, Pruebas y Bucles

August 1, 2018

En esta ocasión me voy a centrar en lo más básico de la programación con BASH

Detección de Objetos en Vídeo en Directo Usando el ODROID-XU4 con GStreamer

August 1, 2018

Uno de los campos más utilizados para el aprendizaje profundo ha pasado a ser la detección de objetos

Juegos Linux: Juegos Saturn – Parte 5

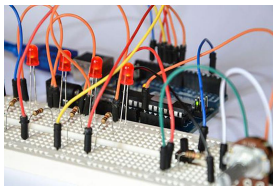
August 1, 2018

Y estamos de vuelta con el ODROID XU3/XU4 para continuar ejecutando juegos de Sega Saturn. En esta publicación cubriremos el resto de juegos (letra en el alfabeto). Los he probado y he disfrutado bastante jugando con mi ODROID. Una vez más, encontré algunas joyas realmente buenas que quiero compartir contigo. ▶️

Transcodificar el Receptor DVB Enigma2 Usando ffmpeg en el ODROID-XU4

August 1, 2018

Usando NAT (asignando la IP externa a la IP interna del dispositivo) y http flux, puedo ver mis canales de TV



Campamento de Programación Parte 1: Primeros Pasos con Arduino

August 1, 2018

En este artículo, aprenderás cómo descargar e instalar Arduino IDE y ejemplos y librerías específicas de ODROID-GO.

Thundroid – Parte 2 Migración desde Bitcoin Testnet a Mainnet

August 1, 2018

¿Recuerdas la Parte 1 de esta guía? Configuramos un completo nodo de Bitcoin con Lightning desde cero, llegamos bastante lejos para asegurar nuestro sistema y empezamos a probar Bitcoin en testnet.

Recuperar eMMC: Reseteando el Módulo eMMC ODROID-XU4 para Solucionar Problemas de Arranque

August 1, 2018

El cargador de arranque de la serie Exynos está ubicado en una partición de arranque oculta dentro de la memoria eMMC para todos los modelos, excepto para el ODROID-C1/C2. Cuando éste se daña, o se quiere utilizar el módulo eMMC en una placa diferente, se debe instalar el cargador de ▶️

Campamento de Programación Parte 2: Cómo Mostrar “Hello, ODROID-GO” en una Pantalla LCD

August 1, 2018

En este artículo, aprenderás cómo mostrar una cadena de texto, cambiar los colores y cambiar el tamaño de la fuente. Si sigues esta guía, serás capaz de escribir el código que te permitirá mostrar “Hello, ODROID-GO” en tu ODROID-GO.

Cómo Configurar un Servidor Minecraft

August 1, 2018

Este artículo detalla cómo instalar un servidor básico de Minecraft en tu ODROID, para que puedas ejecutar juegos online con algunos de tus amigos en un mundo creado por ti. Usar el ODROID como un sandbox – entorno de pruebas – de bajo coste también es una excelente forma de ▶️

Ejecutar Modernos Juegos FNA sobre la Plataforma ODROID

August 1, 2018 By Sebastien Chevalier Juegos, Tutoriales

FNA (http://fna-xna.github.io/) es una reimplementación de código abierto de XNA. XNA es un entorno de juegos creado por Microsoft Corp., y es bastante popular. Se han escrito excelentes juegos con XNA (solo para MS Windows) y luego se han exportan a Linux y MacOS usando FNA y/o MonoGames.

Entre estos juegos, puedes encontrar algunas joyas como Bastion, FEZ, Stardew Valley, Owlboy y Terraria, por enumerar algunos. Para tener una mejor idea de los juegos que usan este entorno, dirígete a https://goo.gl/4MGnys o http://www.monogame.net/showcase/

Figura 01 – Stardew Valley

En la actualidad, este entorno es muy interesante para los SBC ODROID, puesto que no está basado en C o C ++, sino en C#. La gran ventaja de los juegos compilados con C# es que pueden ejecutarse en cualquier variante de Linux (cualquier arquitectura de CPU) siempre que soporte y ejecute Mono. Con un poco de trabajo, la mayoría de estos juegos pueden funcionar sin disponer del código fuente del juego en sí. Basta con usar binarios compilados para Linux x86.

Requisitos previos Por supuesto, para lanzar estos juegos, será necesaria una cierta preparación y compilación. Junto con Mono, necesitaremos algunas librerías que dan soporte a los juegos, como SDL2 y también algunas librerías específicas requeridas por los propios juegos. No podemos usarlas aquí debido a la arquitectura ARM del ODROID. También necesitaremos una nueva versión de GL4ES porque todos estos juegos usan extensiones OpenGL 2.1+ como mínimo. Esta guía debería funcionar con la imagen predeterminada de HardKernel o con ODROID GameStation Turbo (OGST). En primer lugar, nos aseguraremos de que todo esté actualizado (no olvide responder "Yes" si solicita una actualización):

```
$ sudo apt update && sudo apt upgrade
```

Ahora que está actualizado el sistema, instalaremos las librerías estándar que usaremos. Comenzaremos con SDL2 y las librerías relacionadas:

```
$ sudo apt install libsdl2-2.0 libsdl2-net-2.0
libsdl2-mixer-2.0 libsdl2-ttf-2.0
```

Para instalar Mono, ejecuta el siguiente comando:

```
$ sudo apt install mono-complete mono-xbuild
```

Una vez realizadas estas instalaciones, estamos casi listos. El problema es que SDL2 puede no estar compilado para soportar OpenGL, y faltas algunas librerías del repositorio que tienen que compilarse a partir de las fuentes. De modo que, simplemente para estar seguros, vamos a instalar algo de material de desarrollo (es posible que ya tenga la mayoría o todo instalado):

```
$ sudo apt install build-essential git
mercurial cmake libgl1-mesa-dev
$ sudo apt install libtheora-dev libvorbis-dev
```

Si usas OGST y no deseas compilar todas estas librerías, simplemente puedes usar las que compiló @meveric con el siguiente comando:

```
$ sudo apt install monolibs-odroid
```

Para el resto que quiera compilar por si mismos los componentes necesarios, prepararse para una compilación bastante contundente.

Compilar algunas librerías Compilaremos algunas librerías y las colocaremos en una carpeta fácil de encontrar, para que podamos dirigir la búsqueda de esas librerías con LD_LIBRARY_PATH, así que vamos a crear esa carpeta, llamada monolibraries en tu directorio de inicio:

```
$ mkdir ~/monolibs
```

Ahora, compilaremos las librerías que necesitamos, en concreto: gl4es, SDL2 con soporte OpenGL, mojoshaders y libtheroaplay.

GL4ES Esta librería permite utilizar mucho software/juegos OpenGL 1.x y 2.x bajo el hardware GLES. Es la pieza central de software, junto con Mono, permite que todos estos juegos se puedan ejecutar en ODROID. Las fuentes están en mi cuenta github, así que vamos a coger fuentes más recientes:

```
$ cd ~
$ git clone
https://github.com/ptitSeb/gl4es.git
```

Una vez que hayas clonado el repositorio, para obtener las últimas fuentes, simplemente dirígete al repositorio y escribe "git pull". Ahora, configura la compilación para ODROID:

```
$ cd gl4es
$ cmake -DODROID=1  -
DCMAKE_BUILD_TYPE=RelWithDebInfo .
```

y compila las librerías:

```
$ make -j2
```

Luego, simplemente cópialas en la carpeta "monolibraries" para usarlas más adelante:

$ cp lib/libGL.so.1 ~/monolibs/

SDL2 Ya tenemos instalado SDL2, pero puede que sea la versión que solo admite GLES y no OpenGL. Entonces seguro que tenemos que compilar una nueva versión. No es tan complicado de todas formas. Usemos la versión que está en mi cuenta de GitHub. Cualquier otra versión debería funcionar, usaremos la mía simplemente por comodidad:

```
$ cd ~
$ git clone
```

```
https://github.com/ptitSeb/SDL2.git
Ahora configuraremos la compilación para
OpenGL (en esta ocasión haremos una
compilación fuera de árbol):
$ cd SDL2
$ mkdir build
$ cd build
$ cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo.
```

El paso de configuración se ejecutará para un bit. Deberías ver, entre muchas otras cosas: "–VIDEO_OPENGL (Wanted: ON): ON". Así que ahora, vamos a compilar esta librería (tardará algo más que gl4es):

```
$ make -j2
```

Ahora podemos copiar la librería recién compilada al directorio monolibraries:

```
$ cp libSDL2-2.0.so.0 ~/monolibs/
```

mojoshaders Esta librería es una de las librerías secundarias creada por Icculus que nos ayuda a exportar el código de Windows a Linux. Esta librería en particular convierte los sombreadores escritos para DirectX en sombreadores GLSL para OpenGL (o Metal). Esta librería es utilizada por FNA para usar trasparentemente los sombreadores DirectX en OpenGL. Obtendremos el código fuente directamente desde el repositorio de Icculus, usando mercurial esta vez:

```
$ cd ~
$ hg clone
http://hg.icculus.org/icculus/mojoshader
```

Vamos a configurar (para producir una librería compartida, ya que, por defecto, no lo es)

```
$ cd mojoshader
$ cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo -
DBUILD_SHARED=ON
-DDEPTH_CLIPPING=ON -DFLIP_VIEWPORT=ON
```

y creamos la lib:

```
$ make -j2
```

Copiaremos también esta librería y pasaremos a la siguiente:

```
$ cp libmojoshader.so ~/monolibs/
```

XNAFileDialog

Algunos juegos usan esta librería. Vamos a compilar una versión de la parte nativa por si acaso

```
$ cd ~
$ git clone
https://github.com/flibitijibibo/XNAFileDialog
.git
$ cd XNAFileDialog/native
$ make
$ cp libXNAFileDialog.so ~/monolibs/
```

LibTheoraPlay Algunos juegos usan libtheoraplay para los videos ("A Virus Named TOM" por ejemplo). Esta librería es un poco complicada porque viene en 2 partes: la parte C y la parte C#. La parte C# necesita algunos ajustes para poder ejecutarse en ARM, ya que hay una solución para algunos problemas en Mono/x86 que no aplican en nuestro caso (y corrompen las cosas).

```
$ cd ~
$ git clone
https://github.com/flibitijibibo/TheoraPlay-
CS.git
```

El parche es simple: abre TheoraPlay.cs desde ~/TheoraPlay-CS con tu editor de texto favorito y dirígete a la línea 155. Busca la siguiente línea: /* This is only a problem for Mono. Ignore for Win32 */ if (Environment.OSVersion.Platform != PlatformID.Win32NT && IntPtr.Size == 4) y reemplaza el gran "if" que está dividido en 2 líneas con un simple if (false)

Ahora, compilaremos la parte C de la librería y la copiaremos en monolibraries:

```
$ cd TheoraPlay-CS/TheoraPlay
$ cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo .
$ make
$ cp libtheoraplay.so ~/monolibs/
```

Ahora, vamos a compilar la parte C# de la librería (es una DLL de MS Windows) y copiamos el dll en el mismo lugar, para usarlo en el futuro. Podemos ignorar con total seguridad el archivo .dll.config, no lo necesitaremos:

```
$ cd ..
$ xbuild /p:Configuration=Release
$ cp bin/Release/TheoraPlay-CS.dll ~/monolibs/
```

TheoraFile Algunos juegos usan esta librería. Vamos a compilar una versión de la parte nativa por si acaso.

```
$ cd ~
$ git clone https://github.com/FNA-XNA/Theorafile.git
$ cd Theorafile
$ make
$ cp libtheorafile.so ~/monolibs/
```

Otras librerías Algunos juegos pueden solicitar otras librerías que son más difíciles de compilar o no son de código abierto. Por ejemplo, Hacknet pedirá libcef (que es básicamente "Chrome en una lib"), o Bastion preguntará por FMODex (que es de código cerrado). Para esos juegos, estás sólo, aunque si encuentra alguna solución funcional, no dudes en dirigirte a los foros ODROID y añade una publicación al respecto. FMOD se puede descargar para ARMHF, pero no parece existir para ARM64 (así que no pude probarlo en mi N1). Para obtener FMOD, debes registrarte en **http://www.fmod.com** y descarga fmodstudioapi para Linux (obtendrá un archivo similar, fmodstudioapi11006linux.tar.gz). Extrae y copia libfmod en monolibraries con:

```
$ cd ~
$ tar xf fmodstudioapi11006linux.tar.gz
$ cp api/lowlevel/lib/armhf/libfmod.so ~/monolibs/
```

Para FMODex, puedes usar el pequeño contenedor que escribí:

```
$ cd ~
$ git clone https://github.com/ptitSeb/fakemodex.git
$ cd fakemodex
$ cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo .
$ make -j2
$ cp lib/libfmodex.so ~/monolibs/
```

Extraer juegos Ahora, tenemos que instalar algunos juegos en ODROID. Los juegos necesitan usar FNA o MonoGames, los que proceden de GoG, HumbleBundle o incluso Steam en algunos casos.

Todos necesitan ser la versión de Linux del juego. En la mayoría de los casos, la versión de Windows no funcionará. Debes tener en cuenta que la versión Steam de algunos juegos tampoco se ejecutará sin Steam no está activado (DRM) y puesto que no tenemos Steam en el ODROID, simplemente no podremos ejecutar el juego en ODROID. Utiliza la versión GOG o HB (o cualquier otra versión sin DRM) para los juegos que quieras ejecutar en tu ODROID.

Steam u otra versión instalada de Linux Si tiene una versión de Linux instalada del juego, simplemente copia la carpeta completa del juego y estarás listo para empezar a jugar. La versión Steam de FEZ o Owlboy, por ejemplo, se puede utilizar.

Versión Humble Bundle Ahora muchos juegos vienen como un único y gran archivo que termina en "-bin". Estos juegos se pueden extraer fácilmente: hay un archivo zip integrado y todos los juegos están dentro de la carpeta "data". Por ejemplo, "A Virus Named TOM" viene como "avnt-10192013-bin", y "Towerfall: Assension" es "towerfall-07212016-bin". Además, debido a que algunos juegos tienen un "data" que entraría en conflicto con la carpeta "data" extraída, vamos a cambiar el nombre temporalmente a "ODROID". Para preparar "A Virus Named TOM", puede hacer lo siguiente:

```
$ cd ~
$ mkdir AvirusNamedTOM
$ cd AVirusNamedTOM
$ unzip ~/avnt-10192013-bin data/*
$ mv data ODROID
$ mv ODROID/* .
$ rm -r ODROID
```

Versión GOG Los paquetes de juegos GOG son bastante similares a los de extracción. Para bastión, tengo un "gog_bastion_2.0.0.1.sh" que también contiene un archivo zip:

```
$ cd ~
$ mkdir Bastion
$ cd Bastion
$ unzip ~/gog_bastion_2.0.0.1.sh data/noarch/game/*
$ mv data ODROID
$ mv ODROID/noarch/game/* .
$ rm -r ODROID
```

Figura 02 – FEZ01

Lanzar los juegos Finalmente, estamos listos para un poco de acción. Necesitamos eliminar algunas librerías de la instalación en primer lugar. Debido a que utilizaremos la versión de Mono que viene con ODROID, y no la que está incluida en los juegos, primero tenemos que llevar a cabo algunas tareas de limpieza:

```
$ cd ~/AvirusNamedTOM
$ rm mscorlib.dll
$ rm System.*dll
$ rm Mono.*.dll
```

En la actualidad, algunos juegos (como A Virus Named Tom) usan TheoraPlay. Puede aparecer bajo 2 nombres: “TheoraPlay-CS.dll” o “TheoraPlay # .dll”. Si ves algo de esto, asegúrate de reemplazarlo con el que compilamos antes o se bloqueará cuando se inicie el video (solo en 32bits, 64bits son seguros). Para la versión HB de A Virus Named TOM, será:

```
$ cd ~/AvirusNamedTOM
$ cp ~/monolibs/TheoraPlay-CS.dll
TheoraPlay#.dll
```

Figura 03 – TwerFall04

Ahora podemos ejecutar el juego. Necesitamos configurar algunas cosas para que GL4ES emule OpenGL2 y también necesitamos usar todas las librerías que hay dentro de monolibraries. Localiza el archivo “.exe” y simplemente ejecútalo con mono. Para “A Virus Named TOM” los comandos son:

```
$ cd AvirusNamedTOM
$ LC_ALL=”C” LIBGL_ES=2 LIBGL_GL=21
LIBGL_DEFAULTWRAP=2
LIBGL_FBOFORCETEX=1 LD_LIBRARY_PATH=~/monolibs
mono CircuitGame.exe
```

Podemos dar un paso más allá, usando componentes adicionales.

Remuestreo del audio Notarás que algunos juegos tardan un tiempo en inicializarse y usan bastante memoria. La mayoría de las veces, esto se debe al tema del sonido del juego, donde todo se carga en la memoria al inicio. Si tienes problemas de memoria o simplemente quiere experimentar, he desarrollado una pequeña herramienta que se puede utilizar para volver a muestrear los datos. La herramienta se compila fácilmente utilizando los siguientes comandos:

```
$ sudo apt install libsox-dev
$ cd ~
$ git clone
https://github.com/ptitSeb/rexwb.git
$ cd rexwb
$ cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo .
$ make
```

Usar la herramienta es bastante fácil. Tienes que entender que “xwb” WaveBank muestrea a 44 kHz, utilizando compresión MSADPCM. Aunque esto es bastante eficiente en Windows, la mayoría de las versiones en Linux expanden el MSADPCM al formato PCM clásico (por lo que el tamaño se multiplica por 4), lo cual lleva a tener grandes fragmentos del archivo de sonido en memoria. Remuestrear los sonidos en wavebanks a Mono (en lugar de Stereo) y remuestrear a 22kHz (o menos) reduce la demanda de memoria. Los juegos que tienen xwb incluyen “A Virus Named TOM” y “Stardew Valley”. Encontrarás los bancos de ondas dentro de Content/Audio. Ten en cuenta que rexwb siempre trabaja sobre una copia de los bancos

de ondas. Para volver a muestrear el banco de ondas de TOM (este juego tiene 2 bancos de ondas, solo el BGM puede volver a muestrearse, o ambos, dependiendo de la elección individual) a mono/22kHz usa los siguientes comandos:

```
$ cd ~/AvirusNamedTOM
$ cd Content/Audio
$ mv BGMwaves.xwb BGMwaves.xwb.sav
$ ~/rexwb/rexwb BGMwaves.xwb.sav BGMwaves.xwb
22050 -f -m
$ mv SFXwaves.xwb SFXwaves.xwb.sav
$ ~/rexwb/rexwb SFXwaves.xwb.sav SFXwaves.xwb
22050 -f -m
```

FEZ Con FEZ, tuve un problema en mi prototipo N1 con “Hardware Instancing”. Este, sin duda alguna, es un error de GL4ES al que le tengo que seguir la pista (no tuve estos problemas en OpenPandora), así que, si tienes un bloqueo al inicio, simplemente desactiva Instancing en el menú de opciones. Además, este juego usa una enorme lista de más de 400,000 triángulos para trazar esas estrellas en la pantalla de Menú y en algunas pantallas de juego. Aunque algunos ODROID de gran potencial como el N1 pueden manejar esto, otros modelos pueden tener problemas con ese tipo de trazado. Puedes hacer una cosa en GL4ES para evitar este trazado. Con tu editor de texto favorito, dirígete a la carpeta gl4es y edite src/gl/drawing.c. Busca “#if 0” en ese archivo (alrededor de la línea 207) y cámbialo a “#if 1”. Recompila la librería y cópiala en monolibraries para tener una versión que no trace el campo de estrellas y contar así con un menú principal más suave y sin problemas.

Figura 04 – FEZ

Stardew Valley He notado algunos problemas gráficos con Stardew Valley, pero nada serio con la versión de Steam. Una cosa a tener en cuenta es que algunos archivos DLL quieren cargar “oal_soft.dll”. Debería ser redireccionado a “libopenal.so” pero por algún motivo no es así. La manera más fácil es crear un enlace simbólico dentro de la carpeta Stardew valley hacia “libopenal.so” llamado “liboal_soft.so” y funciona. En mi N1, que es ARM64, el comando sería:

```
$ cd ~/StardewValley
$ ln -s /usr/lib/aarch64-linux-
gnu/libopenal.so libsoft_oal.so
```

Es similar en ARM de 32 bits:

```
$ cd ~/StardewValley
$ ln -s /usr/lib/arm-linux-
gnueabihf/libopenal.so libsoft_oal.so
```

Figura 05 – Stardew Valley

A Virus Named TOM He probado este juego en el ODROID-N1. He tenido algunos problemas gráficos con este juego, donde la imagen está limitada a una subparte de la imagen completa. Probablemente sea un error de gl4es, pero también puede ser un error en el controlador GLES del N. No he visto ningún problema con OpenPandora durante mis pruebas.

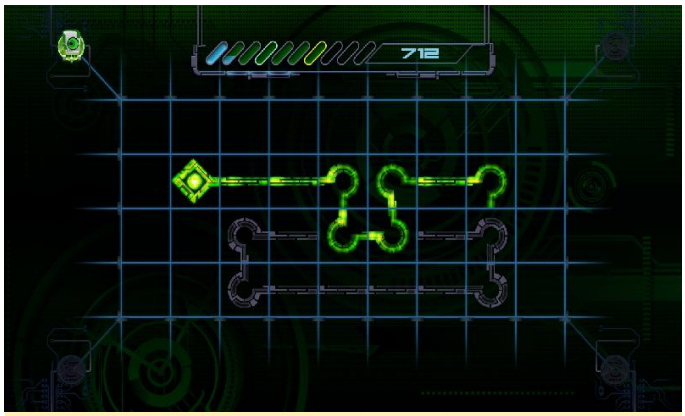


Figura 06 – TOM

Bastion Este básicamente necesita FMODex. Así que probablemente necesitarás descargar FMOD y compilar el contenedor fakemodex para jugar a este juego.

Hammerwatch Aunque no es un juego FNA, Hammerwatch utiliza un motor personalizado y también está hecho en C #. Hammerwatch se puede ejecutar de la misma forma. Sin embargo, ten en cuenta que la última versión (1.32) usa FMOD para la música y el sonido, así que debe obtener la versión nativa del mismo (FMOD, no FMODex). La versión anterior (sin dlc) no usa fmod.

Otros juegos

Figura 07 – Dust: An Elysian Tail

Figura 08 – Towerfall

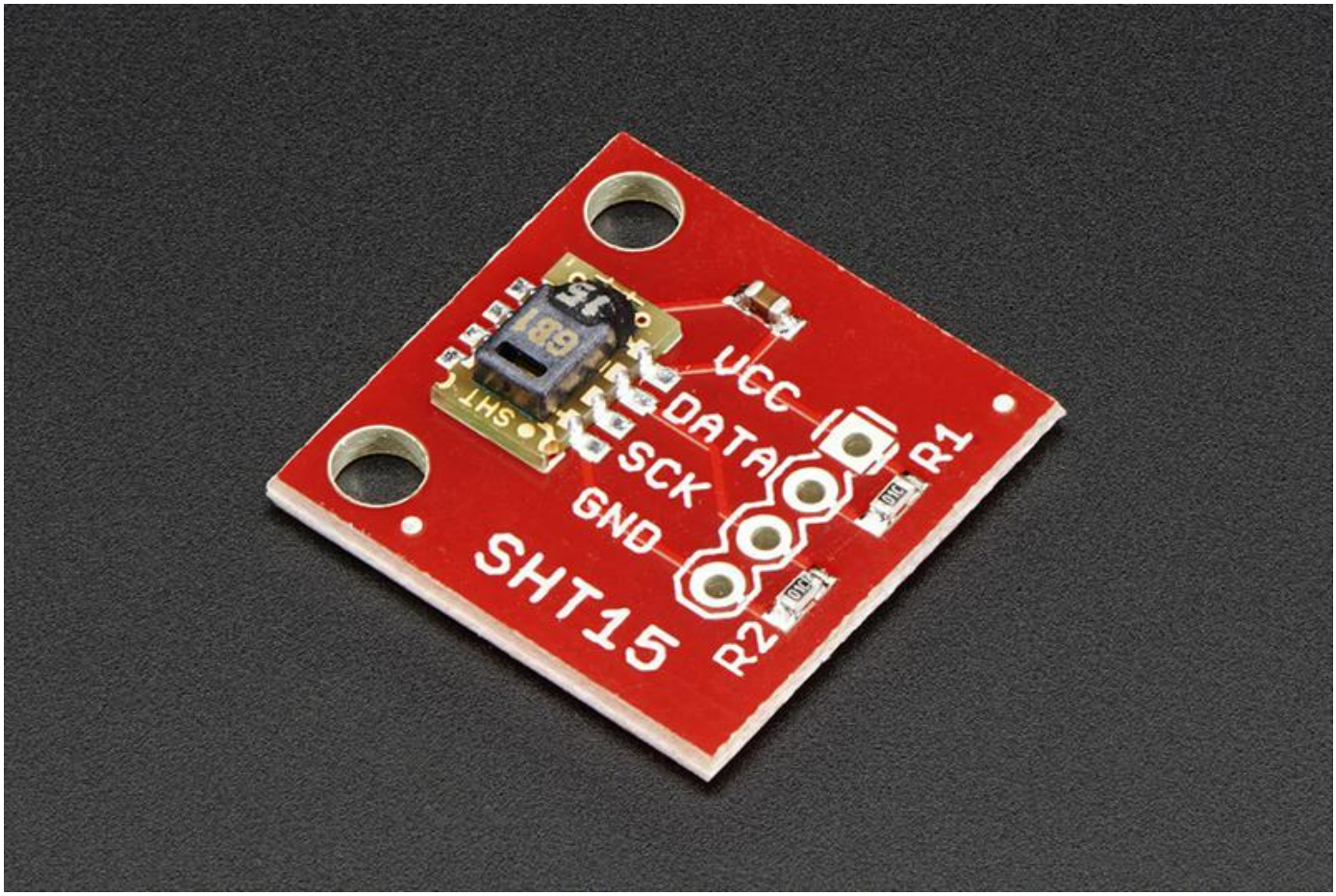
Figura 09 – Owlboy

Figura 10 – Owlboy

No dudes en dirigirte a los foros de ODROID y crear alguna publicación para que comentemos tus éxitos y fracasos con FNA.

Lectura de Temperatura y Humedad desde un Sensor SHT15: Introducción a la Interfaz GPIO

August 1, 2018 By Jon Petty ODROID-C0, ODROID-C1+, ODROID-C2, Mecaniqueo



El objetivo de este proyecto es utilizar un ODROID para leer los datos de temperatura y humedad desde un sensor SHT15, y explicar cómo un ODROID se comunica con un SHT15 a través de los pines GPIO. Los sensores SHT15 son fabricados por Sensirion y miden tanto la temperatura como la humedad de un entorno. La comunicación con un sensor tiene lugar a través de los pines GPIO del ODROID. Un pin GPIO se conecta al pin SCK del sensor, que controla la rapidez con la que se produce la comunicación. El segundo pin GPIO se conecta al pin DATA del sensor, que se usa para enviar comandos y leer resultados. Una vez configurado todo, el ODROID enviará una solicitud para medir la temperatura o la humedad a través del pin de DATOS, espera a que el sensor complete su medición y luego lee el resultado sobre el pin de DATOS.

Conectar del sensor SHT15

El siguiente diagrama describe cómo conectar un sensor SHT15 a un ODROID.

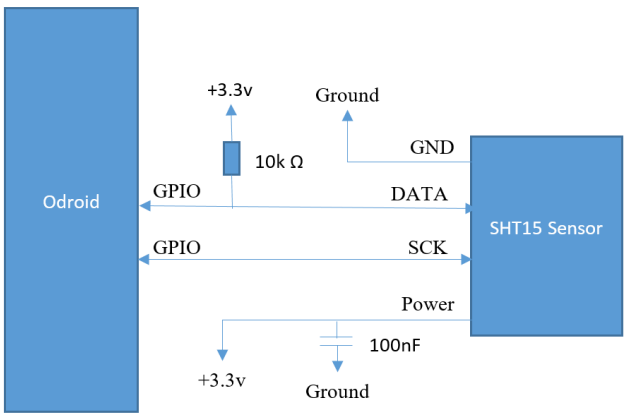


Figura 1 – Diagrama SHT15

Hay dos cosas que debemos tener presente. En primer lugar, las hojas de especificaciones técnicas son un excelente lugar para obtener información sobre cómo usar los componentes electrónicos. El circuito de la Figura 1 ha sido copiado de la hoja de

datos del sensor. Se trata de una recomendación del fabricante a modo de configuración que produce buenas mediciones. En segundo lugar, soldar un SHT15 es complicado. Para facilitar las cosas, este tutorial utiliza una placa sensor SHT15 prefabricada.

Elementos necesarios

Para empezar, se necesitan las siguientes componentes y herramientas:

- ODROID (**http://bit.ly/1QPVZa9**)
- Kit de retoque ODROID (**http://bit.ly/1LmFcdf**)
- Placa sensor SHT15 (**http://bit.ly/1qd22ZL**)
- Cables
- Soldador y estaño

Una vez que tengas la placa del sensor SHT15, realiza las siguientes conexiones tras soldar los cables a la misma:

- Conecta VCC a la fuente de alimentación +3.3V del ODROID
- Conecta DATA al pin #100 GPIO del ODROID
- Conecta SCK al pin #97 GPIO de ODROID
- Conecta GND a la GND del ODROID

Deberías terminar con algo que se parezca a la Figura 2.

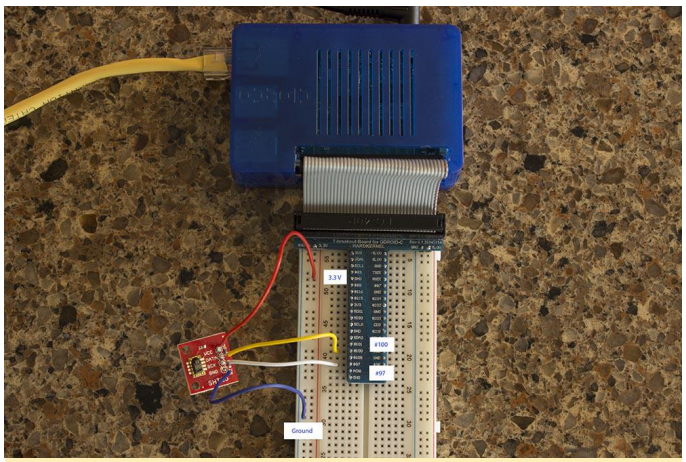


Figura 2 – Conexiones SHT15

Lectura y escritura de los valores GPIO

Pin GPIO quiere decir pin de entrada/salida de propósito general. Cuantos tiene tu ODROID depende del modelo en concreto, pero en todos los casos, se utilizan para leer y escribir datos binarios. Los datos binarios son datos con solo dos estados, comúnmente denominados HIGH y LOW, o 1 y 0. Físicamente, un valor HIGH significa que el voltaje del pin es +3.3 voltios, y un valor LOW significa que el voltaje del pin es de +0.0 voltios. Ten en cuenta que el nivel de voltaje depende del dispositivo. Por ejemplo, un Arduino funciona desde los +5.0 voltios a +0.0 voltios. Si el ODROID está escribiendo datos en un pin GPIO, cambiará el voltaje entre +3.3 voltios y +0.0 voltios dependiendo de si se ha escrito HIGH o LOW. Si el ODROID está leyendo datos, registrara HIGH cuando se aplica +3.3 voltios al pin, y LOW cuando se aplica +0.0 voltios al pin. Para este proyecto, vamos a leer y escribir datos desde y hacia dos pines GPIO. Esto implica los siguientes pasos:

- Conectar los pines GPIO ODROID al sensor
- Iniciar sesión en Linux en el ODROID y navegar hasta el directorio GPIO
- Iniciar una conexión con los dos pines GPIO conectados (uno para DATA y otro para SCK)
- Cuando sea necesario, fijar los pines en modo escritura y escribir datos
- Cuando sea necesario, fijar los pines en modo lectura y leer datos

Para empezar, inicia sesión en tu ODROID y abre un terminal de línea de comandos. Algunos de los siguientes comandos se deben ejecutar como root, lo cual se puede hacer con el siguiente comando:

```
$ sudo su -
```

Los pines GPIO se encuentran en el directorio /sys/class/gpio:

```
$ cd /sys/class/gpio
```

Un programa llamado “export” se encuentra en este directorio, que activa las conexiones con los pines GPIO. Es necesario activar un pin antes de que los datos puedan leerse o escribirse en él. Para activar una conexión, transmite el número de identificación del pin. En este tutorial, conectaremos el pin DATA del sensor SHT15 al pin 100 GPIO, y el pin SCK del sensor al pin 97 GPIO. Estas dos conexiones se activan con los comandos siguientes.

```
$ echo 100 > /sys/class/gpio/export
$ echo 97 > /sys/class/gpio/export
```

Tras completarse estos comandos, deberías observar que se han creado los siguientes directorios:

```
/sys/class/gpio/gpio100
/sys/class/gpio/gpio97
```

Estos directorios contienen todo lo necesario para leer y escribir datos desde los correspondientes pines GPIO. El primer archivo importante a tener en cuenta es “direction”. Para el pin 100 GPIO, se encuentra en /sys/class/gpio/gpio100/direction. El archivo “direction” cambia un pin entre el modo lectura y el modo escritura. No puedes leer y escribir datos simultáneamente al mismo tiempo en un solo pin. Sin embargo, puede tener varios pines donde algunos están leyendo datos y otros están escribiendo datos. Un pin se puede cambiar al modo escritura escribiendo un valor “out” en el archivo de “direction”. Del mismo modo, un pin se puede cambiar al modo lectura escribiendo un valor “in” en el archivo “direction”. Por ejemplo, el siguiente comando cambia el pin 100 GPIO a modo escritura:

```
$ echo out > /sys/class/gpio/gpio100/direction
```

El siguiente comando cambia el pin 100 GPIO a modo lectura.

```
$ echo in > /sys/class/gpio/gpio100/direction
```

Para identificar en qué modo está un pin GPIO, puede leer el valor de “direction”. Por ejemplo, el siguiente comando identifica si el pin GPIO 100 está en modo lectura o modo escritura.

```
$ cat /sys/class/gpio/gpio100/direction
```

El segundo archivo importante a tener en cuenta es “value”. Para el pin 100 GPIO, se encuentra en /sys/class/gpio/gpio100/value. La lectura y escritura de datos binarios se realiza utilizando el archivo “value”. Si el pin está en modo escritura, el archivo “value” se utiliza para generar datos binarios. Si el pin está en modo lectura, el archivo “value” se usa nuevamente, pero en este caso lee datos binarios del pin. Para evidenciar esto, podemos ejecutar una pequeña prueba y ver si la placa de circuitos está conectada correctamente. Cuando se conecta inicialmente, el pin DATA debe estar HIGH y el pin SCK debe estar LOW. Para determinar si este es el caso, primero cambia ambos pines al modo lectura.

```
$ echo in > /sys/class/gpio/gpio100/direction
$ echo in > /sys/class/gpio/gpio97/direction
```

Segundo, lee el valor de GPIO para cada pin.

```
$ cat /sys/class/gpio/gpio100/value
$ cat /sys/class/gpio/gpio97/value
```

El pin 100 (DATA) debe imprimir un valor “1”, y el pin 97 (SCK) debe imprimir un valor “0”. Si este no es el caso, la solución del problema pasa por verificar tus conexiones usando el diagrama de cables anterior como referencia, y comprobar que los pines GPIO están configurados en modo lectura verificando los valores del archivo “direction” :

```
$ cat /sys/class/gpio/gpio100/direction
$ cat /sys/class/gpio/gpio97/direction
```

Comunícanse con el sensor SHT15

Los siguientes pasos hacen que los datos de humedad o temperatura puedan leerse desde un sensor:

1. El ODROID envía una solicitud al sensor para registrar la temperatura o la humedad. Ten en cuenta que el sensor no puede leer la temperatura y la humedad de forma simultánea. Si se necesitan tomar ambas mediciones, éstas se deben realizar de forma secuencial.
2. El sensor empieza a tomar una medición, el ODROID espera.
3. Una vez que se complete la medición, el ODROID lee el resultado desde sensor.
4. El ODROID convierte la medición en algo legible por los seres humanos.

Para solicitar que se tome una medida, el ODROID envía un número binario al sensor. Por ejemplo, el número 00000011 solicita que se mida la temperatura, y el número 00000101 solicita que se mida la humedad. Los números en sí se envían bit por bit sobre el pin de DATOS. El pin SCK controla la

rapidez con la que se envían los valores. Echa un vistazo a la Figura 3, muestra los valores de los pines GPIO al transmitir el número 00000101 (solicitud de medición de humedad).

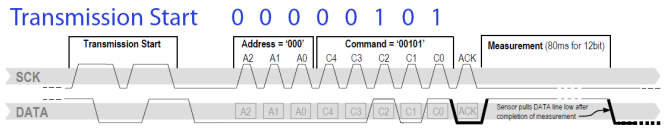


Figura 3 – Solicitud de medición de humedad

En la figura 3 podemos diferencias tres secciones. La primera es la secuencia de inicio de la transmisión. Se trata de una combinación de valores HIGH y LOW transmitidos a través de DATA y SCK que indican que el sensor está a punto de enviar un comando. La segunda sección es el número de solicitud. En él, el pin DATA transmite cada bit 0-0-0-0-0-1-0-1 y el pin SCK varía entre 1 y 0. El pin SCK controla el tiempo de transmisión de los datos. Cuando SCK es 0, indica que no hay nada para ser leído. Cuando SCK es 1, indica que hay algo que está listo para ser leído. Alternar SCK entre 1 y 0 mientras se transmite cada bit sobre DATA permite que ODROID envíe solicitudes de medición al sensor.

La última sección del diagrama anterior es la sección ACK, también conocida como la sección de confirmación. En esta sección, el ODROID cambia el pin DATA a modo lectura. Esto hace que lea valores escritos por el sensor. Si el sensor SHT15 recibió correctamente el comando, escribirá un valor 0 en DATA durante la sección ACK, luego cambiará DATA a 1. El ODROID continúa controlando el valor de SCK en modo escritura y deja un momento para que registre una medición. Una vez completada la medición, el sensor cambia el pin DATA a 1. Esto indica que el ODROID puede leer libremente el resultado del sensor. Los resultados se componen de dos bytes, con un total de 16 bits. La Figura 4 muestra el ODROID leyendo el resultado de una medición de ejemplo.

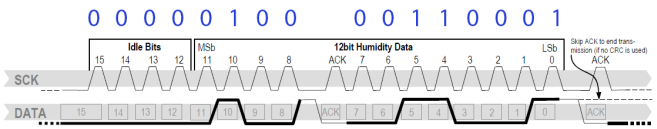


Figura 4 – Lectura de mediciones

Tal y como se puede apreciar en la Figura 4, el ODROID lee el número en dos trozos, 00000100 y 00110001. Cada uno de estos trozos se denomina byte. Esto sucede en tres secciones. Las secciones primera y tercera transmiten los bytes reales. Estas transmisiones se producen poco a poco ya que el ODROID alterna SCK entre 0 y 1 mientras lee DATA. La segunda sección es otra señal ACK. Una vez enviado el primer byte, el sensor cambia DATA a 1. Para enviar una señal ACK, ODROID necesita cambiar DATA a 0 y alternar SCK entre 0 y 1. Esto le dice al sensor que el ODROID está listo para leer el segundo byte. El número leído desde sensor está en formato binario y debe convertirse a un sistema numérico de base 10. Más adelante, usaremos un software para hacer esto. Pero por ahora, simplemente ten en cuenta que 00000100 00110001 equivale a 1073. Tras registrar una medición y ser convertida a un sistema numérico de base 10, debe ser añadida a una ecuación para obtener el resultado final. Si se ha tomado una medición de temperatura, se usa la siguiente ecuación:

```
T = -39.7 + 0.04x
```

En estas ecuaciones, x es el número de base 10 registrado desde el sensor SHT15 y T es el resultado final. Por ejemplo, un valor de 1617.5 registrado desde el sensor tras una medición de temperatura indica una temperatura de 25 °C. Si se toma una medición de humedad, se usa la siguiente ecuación.

```
H = -2.0468 + 0.0367x - 0.0000015955x2
```

En esta ecuación, x es el número de base 10 registrado desde el sensor SHT15 y H es el resultado final. Por ejemplo, un valor de 1073 registrado desde el sensor tras una medición de humedad indica una humedad del 35.5%

Usar PHP para leer datos de humedad y temperatura

Después de echar un vistazo a la sección anterior, la idea de controlar los pines SCK y DATA a través de la línea de comandos de Linux para solicitar y leer las mediciones podría no parecer muy atractiva. Si ese es el caso, ¡Estoy totalmente de acuerdo contigo! Para hacer todo esto más manejable, he escrito dos scripts

PHP para que hagan el trabajo duro. Para descargar estos scripts, dirígete al directorio donde quieres que se guarden y ejecuta los siguientes comandos:

```
$ sudo apt-get install git php5
$ git clone git@github.com:jon-
petty/shtx_php_example.git
```

El primer comando instala PHP, que es necesario para ejecutar los scripts. El comando también instala un programa llamado git, que se suele usar para descargar repositorios de código. El segundo comando usa git para descargar realmente los scripts. Si quieres echar un vistazo a los scripts antes de descargarlos, los pueden ver en **http://bit.ly/1OGGK5Q**. Para ejecutar estos scripts, primero cambia a los directorios, luego sigue las instrucciones del archivo README.md. Contiene las instrucciones más actualizadas sobre cómo ejecutar los scripts:

```
$ cd shtx_php_example
$ less README.md
```

Proyectos futuros

Llegados a este punto, has conectado un sensor SHT15 a tu ODROID y puedes registrar la humedad y la temperatura. También has aprendido como se controlan los pines GPIO en Linux y qué protocolo de comunicación se usa con un sensor SHT15. Si tienes curiosidad y deseas obtener más información, te recomiendo que eche un vistazo a los scripts PHP y compares el código con el protocolo de comunicación y las ecuaciones. También puedes echar un vistazo a las especificaciones técnicas para aprender cosas que están fuera del alcance de este artículo. Por ejemplo, si las temperaturas varían mucho de los 25 °C, la humedad registrada se debe ejecutar a través de una ecuación de compensación para que los resultados sean más precisos.

Referencias

Datasheet SHT1x. Sensirion, Dec. 2011. **http://bit.ly/1x0FfqK**

KeePass: Administrador de Contraseñas

August 1, 2018 By Adrian Popa Linux, Tutoriales

Si eres como yo, que llevan navegando por Internet más de 20 años, y en todos estos años has seguido cometiendo el mismo pecado capital: usar las mismas contraseñas en diferentes sitios por comodidad, tal y como refleja esta viñeta **https://xkcd.com/792/**. Por otro lado, no hay forma de saber cómo protegen tus contraseñas estos sitios, tal vez las almacenen tal cual o codificadas con una baja encriptación, lo cual hace que sean fáciles de descifrar con tablas rainbox, o simplemente la depuración de mensajes revela las contraseñas dentro de los registros logs del servidor.

Recientes Revelaciones han puesto de manifiesto que incluso las grandes compañías como Yahoo, Apple y LinkedIn han sufrido violaciones de datos y les han robado contraseñas. La lista notoriamente larga la puedes encontrar aquí, **https://en.m.wikipedia.org/wiki/List_of_data_breaches**. La única protección que tienes es la de recurrir a los frecuentes cambios de contraseña y evitar la reutilización de las mismas, de modo que una cuenta comprometida no se convierta en una identidad comprometida.

Como ya sabes, los seres humanos somos notoriamente malos a la hora de elegir y recordar muchas y diferentes contraseñas; las más frecuentes las puedes encontrar aquí, **https://en.m.wikipedia.org/wiki/List_of_the_most_common_passwords**. De modo que, necesitamos la ayuda del ordenador para recordar y generar todas las contraseñas y nos hace falta una contraseña maestra muy robusta para proteger el resto. En resumen, necesitamos un administrador de contraseñas. Existen muchos administradores de contraseñas, pero yo me voy a centrar en un programa de código abierto bien fundamentado llamado, **https://keepass.info**, que tiene un back-end para muchos sistemas operativos.

Cliente (GUI) Linux

KeePass es originalmente una aplicación Mono, escrita en .NET, pero debido al soporte inesperado de Microsoft, Mono puede funcionar bastante bien en sistemas Linux, incluso en armhf/arm64. Puede instalar KeePass2 directamente desde apt en tu dispositivo ODROID:

```
$ sudo apt-get install keepass2
```

Una vez iniciado, deberás crear una nueva base de datos para almacenar tus contraseñas (Archivo -> Nuevo ...). Selecciona un nombre adecuado, yo usé 'NewDatabase.kdbx', se te pedirá que definas una Contraseña maestra segura y, opcionalmente, una clave para desbloquear la base de datos. Si usas una clave, puede crearla desde el cuadro de diálogo moviendo el ratón, o con dd, desde /dev/random. En nuestro caso, no vamos a usar una clave, solo una contraseña maestra, así que asegúrate de poner una que sea difícil de adivinar, aquí tienes una sugerencia: **https://security.stackexchange.com/questions/62832/is-the-oft-cited-xkcd-scheme-no-longer-good-advice**. Yo voy a usar "odroid", simplemente como contraseña de ejemplo 🙂

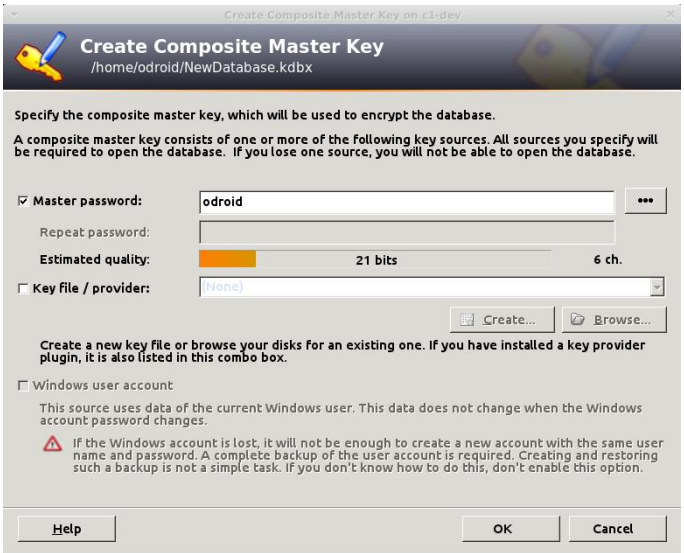


Figura 1 – Creación de la base de datos: usa una contraseña segura

En el segundo paso, se te pedirá un nombre para la base de datos (por ejemplo, "cosas personales"), un nombre de usuario por defecto, algoritmo de cifrado, AES es el utilizado por defecto y el número de ciclos de transformación. Esto significa cuántas veces se debe volver a cifrar la contraseña maestra para generar la clave real. Cuanto mayor sea el número, más difícil lo tendrá los ataques de fuerza bruta, pero también te llevará más tiempo abrir o guardar tu base de datos. La GUI ofrece una opción de "demora de 1 segundo" que calcula el número de ciclos basándose en el PC actual, para un ODROID-C1 es de 78,000, mientras que para un Intel alcanza las decenas de millones. Si seleccionas un número demasiado alto, te llevará más tiempo abrir la base de datos en dispositivos más lentos, 100.000 debería estar bien. Siempre puedes ajustar este número más tarde o cambiar la contraseña maestra. Asegúrate de guardar la base de datos una vez abierta.

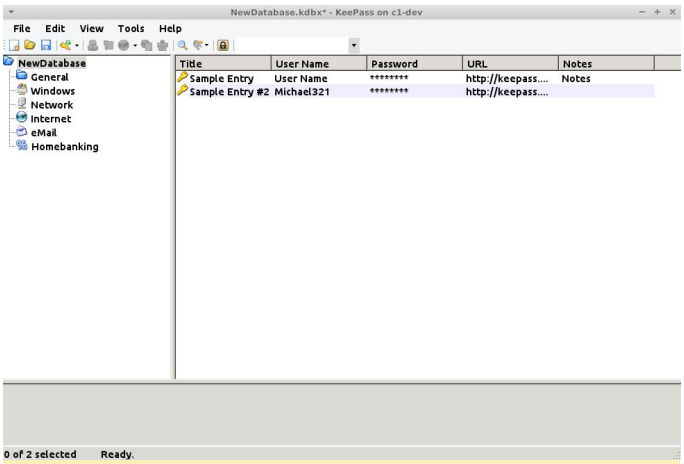


Figura 2 – Ventana principal

Desde la ventana principal puedes usar la barra de herramientas para crear una nueva entrada, buscar entradas existentes y abrir las entradas. También puede agrupar entradas en carpetas. La típica entrada suele tener un título, un nombre de usuario, una contraseña, la URL y algunas notas. También puede añadir archivos y KeePass mantiene un historial de todas las modificaciones realizadas a una entrada, por ejemplo, contraseñas antiguas, que puede ser útil.

Para usar una contraseña guardada, tiene varias opciones. Utilizar CTRL + B para copiar el nombre de usuario y CTRL + C para copiar la contraseña (o hacer doble clic en la entrada de usuario/contraseña) y pégarlos en la aplicación/formulario deseado, o usar la opción de autocompletar (CTRL + V) como esta:

- Navega hasta el recurso deseado, por ejemplo **https://forum.odroid.com/ucp.php?mode=login**

- Cambia el objetivo a KeePass2 y seleccione la entrada deseada y pulsa CTRL + V
- KeePass volverá a la aplicación anterior y pegará el nombre de usuario, usa el tabulador para navegar al siguiente campo y pega la contraseña, presiona Intro. La secuencia se puede cambiar por entrada o por grupo en el caso de que necesites usar otra tecla para la secuencia de inicio de sesión.

Ten en cuenta que, por razones de seguridad, tus datos copiados se guardan en el portapapeles solo durante 12 segundos y luego serán reemplazados por "–" con el fin de mantener tus contraseñas en secreto. Puede cambiar esto en Herramientas -> Opciones -> Tiempo para la limpieza automática del portapapeles. Si la aplicación no es de tu gusto y te recuerda demasiado a Windows, también puedes usar otros clientes GUI para Linux, como KeePassX, pero no tendrás tantos complementos/opciones de importación como en el caso de KeePass2:

```
$ sudo apt-get install keepassx
```

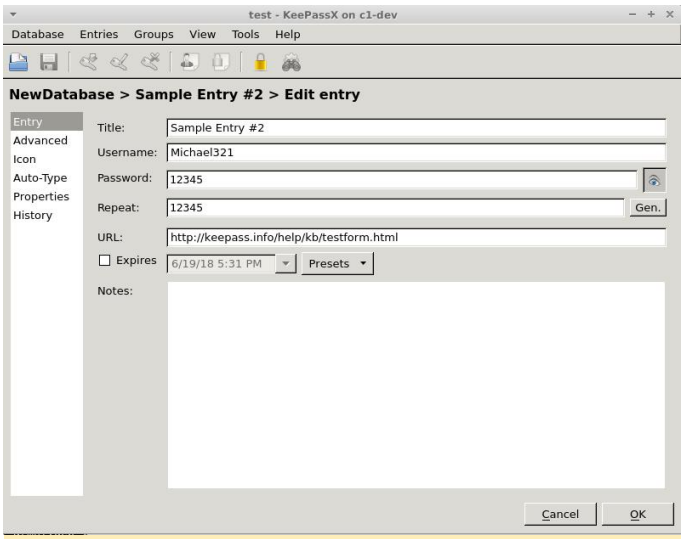


Figura 3 – KeePassX en ODROID-C1

Cliente Web

KeeWeb es una aplicación web escrita en JavaScript con la que puedes administrar tus contraseñas en un navegador, además de poder ejecutarse sin conexión. La idea es que puedas alojarlo en tu ODROID, almacenar el archivo de contraseñas en ODROID y conectarte a la aplicación cada vez que necesites administrar sus contraseñas, sin tener que utilizar otros clientes, https://github.com/keeweb/keeweb/wiki/FAQ.

Puedes obtener la última versión desde Github: https://github.com/keeweb/keeweb/releases. Necesitaras ejecutar Apache o NGINX, para trabajar con archivos estáticos:

```
$ sudo apt-get install apache
$ cd /var/www/html
$ sudo wget
https://github.com/keeweb/keeweb/archive/gh-
pages.zip
$ sudo unzip gh-pages.zip
$ sudo mv keeweb-gh-pages/ keeweb/
$ sudo service apache2 start
$ sudo systemctl enable apache2
```

Llegado a este punto, puedes navegar a https://odroid-ip/keeweb y, tras aceptar el certificado autofirmado, aparecerá en pantalla la página de la figura 4. Aquí puede cargar, solo se carga en el navegador, un archivo KeePass local y puedes verlo.

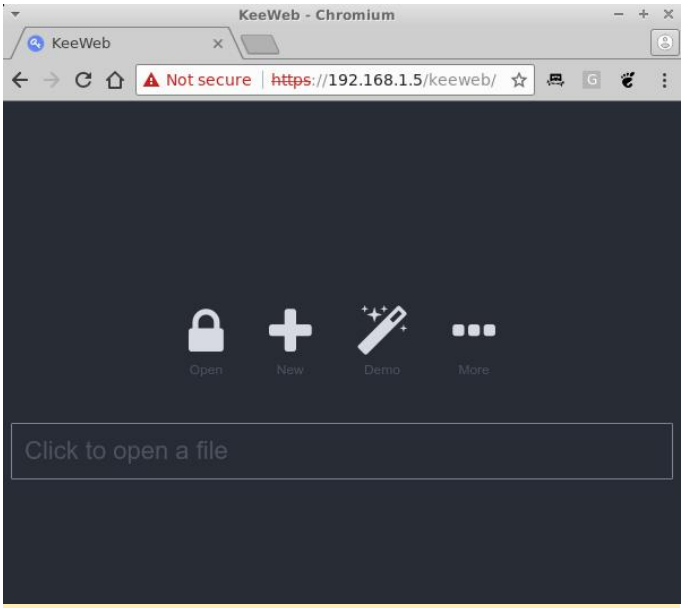


Figura 4 – Vista inicial de KeeWeb

Si quieremos realizar cambios, necesitamos alojar el archivo en el servidor y habilitar WebDAV. WebDAV es un estándar para administrar archivos en un servidor web. Primero crearemos un directorio para almacenar tu base de datos de contraseñas en el servidor y luego la copiaremos allí:

```
$ cd /var/www/html
$ sudo mkdir kppassword
$ sudo cp /path/to/NewDatabase.kdbx
/var/www/html/kppassword
```

```
$ sudo chown -R www-data
/var/www/html/kppassword
```

Asegúrate de introducir la ruta correcta de tu archivo de contraseñas. Además, si añades archivos nuevos despues, asegúrate de que el directorio kppassword y todos los archivos kdbx que contenga sean propiedad del usuario www-data, para que apache pueda guardar los cambios. A continuación, crearemos una cuenta de autentificación HTTP a la que se le permitirá acceder a este archivo mientras esté cifrado. Asegúrate de proporcionar el nombre de usuario y la contraseña requeridos, y que ésta ultima no coincidan con la contraseña maestra, ya que estas credenciales suelen almacenarse en caché de un modo poco seguro:

```
$ sudo htpasswd -c
/etc/apache2/kppassword.access adrianp
```

Ahora podemos habilitar WebDAV para el directorio web de kppassword creando /etc/apache2/conf-available/keeweb.conf y activando algunos módulos de apache. Tiene más detalles aquí: https://github.com/keeweb/keeweb/wiki/WebDAV-Config:

```
$ sudo vi /etc/apache2/conf-
available/keeweb.conf
RewriteEngine on
RewriteCond %{REQUEST_METHOD} OPTIONS
RewriteRule ^(.*)$ blank.html
[R=200,L,E=HTTP_ORIGIN:%{HTTP:ORIGIN}]

< Directory "/var/www/html/kppassword" >
AuthType "Basic"
AuthName "Password Manager"
AuthBasicProvider file
AuthUserFile "/etc/apache2/kppassword.access"
Require valid-user

DAV On
Options Indexes
Header always set Access-Control-Allow-Origin
"*"
Header always set Access-Control-Allow-Headers
"origin, content-type, cache-control, accept,
authorization, if-match, destination,
overwrite"
Header always set Access-Control-Expose-
Headers "ETag"
Header always set Access-Control-Allow-Methods
"GET, HEAD, POST, PUT, OPTIONS, MOVE, DELETE,
COPY, LOCK, UNLOCK"
Header always set Access-Control-Allow-
Credentials "true"
< /Directory >

$ sudo a2enconf keeweb
$ sudo a2enmod dav
$ sudo a2enmod dav_fs
$ sudo service apache2 restart
```

Ahora, si tuvieras que volver a cargar KeeWeb en tu navegador, puedes presionar el botón “More” y seleccionar WebDAV. Debes introducir lo siguiente:

- URL: https://odroid-ip/kppassword/NewDatabase.kdbx
- User: el nombre de usuario que has creado con htpasswd
- Password: el nombre de usuario que has creado con htpasswd

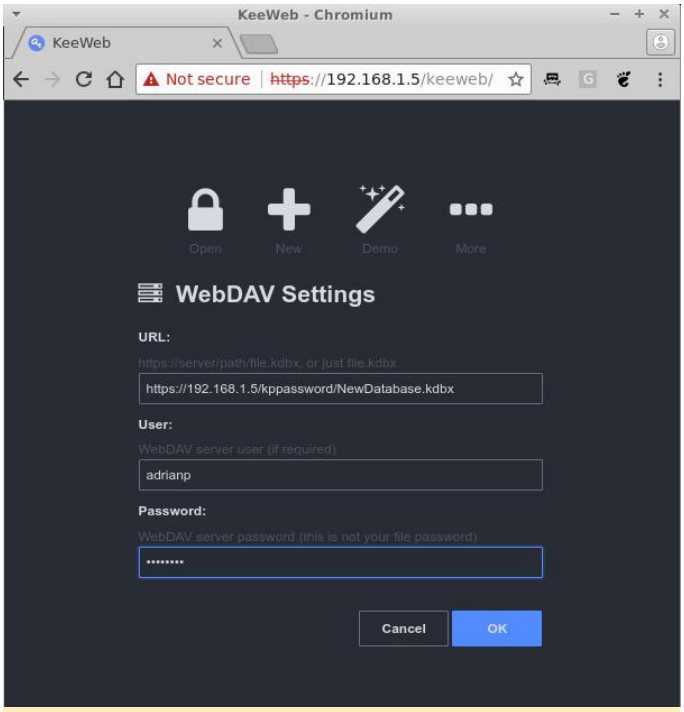


Figura 5 – Cargando la base de datos desde el servidor

Ahora deberías poder hacer cambios y guardarlos, cuando haya cambios sin guardar, aparece un punto azul al lado del nombre de la base de datos. Haz clic en él y se te pedirá que guardes los cambios, el estado cambiará a guardado. Si realizas cambios simultáneos en la misma base de datos desde varios

clientes, debes tener en cuenta que el archivo se sobrescribirá, de modo que es posible que se pierdan los cambios realizados desde los diferentes clientes. Además, es posible que quieras cerrar y reiniciar KeeWeb de vez en cuando para asegurarte de que se carga la última versión del archivo y no una versión anterior de caché. También es obligatorio hacer copias de seguridad periódicas en un lugar distinto al del sitio en el que almacenas tu archivo de contraseñas.

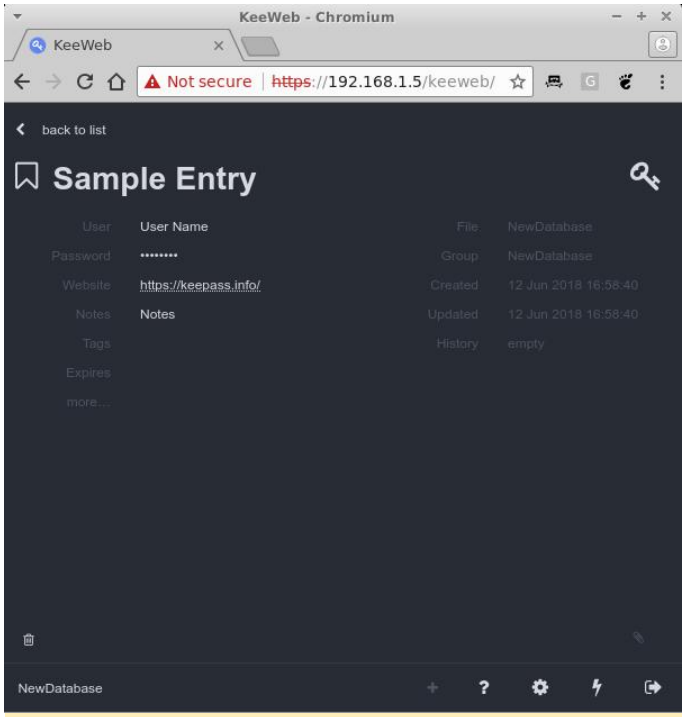


Figura 6 – Accediendo a una entrada

Eso es todo: ahora puede acceder y administrar tus contraseñas desde cualquier navegador con Javascript habilitado. Aunque tal vez estés buscando algo con base terminal...

Cliente CLI

En ocasiones, es posible que sólo tengas acceso por el interprete de comandos – shell y necesites acceder a tus contraseñas. En este caso puedes usar kpcli:

```
$ sudo apt-get install kpcli
```

Puedes conectarte a tu base de datos de contraseñas, como si fura un archivo, y navegar dentro del mismo usando comandos similares a los de Linux, como ls, cd, show.

```
$ kpcli --kdb myPasswordDB.kdbx
Please provide the master password:
**********************
kpcli:/> ls
=== Groups ===
NewDatabase/
kpcli:/> cd NewDatabase/
kpcli:/NewDatabase> ls
=== Groups ===
eMail/
General/
Homebanking/
Internet/
Network/
Windows/
=== Entries ===
0. Sample Entry keepass.info
1. Sample Entry #2
keepass.info/help/kb/testform.
kpcli:/NewDatabase> show -f 0

Title: Sample Entry
Uname: User Name
Pass: Password
URL: https://keepass.info/
Notes: Notes
```

Puedes encontrás mas ejemplos aquí: **https://www.digitalocean.com/community/tutorials/how-to-use-kpcli-to-manage-keepass2-password-files-on-an-ubuntu-14-04-server**

Clientes KeePass de Android

También tiene una amplia selección de clientes Android/iO, **https://keepass.info/download.html**l, y aunque no he probado muchos, me ha gustado bastante KeePass2Android, disponible en **https://play.google.com/store/apps/details?id=keepass2android.keepass2android**, por las siguientes características:

- Código abierto
- Permite desbloquear rápidamente una base de datos previamente desbloqueada con tan sólo los 3 últimos caracteres de la contraseña maestra
- Soporte WebDAV
- Base de datos offline con sincronización automática cuando esté online
- Inicio de sesión automático en sitios a través de la función “compartir”

Puedes instalar el cliente desde Play Store y seleccionar Open file -> HTTPS (WebDAV). Introduce la URL y el usuario/contraseña que definiste al instalar keeweb y se te solicitará la contraseña maestra.

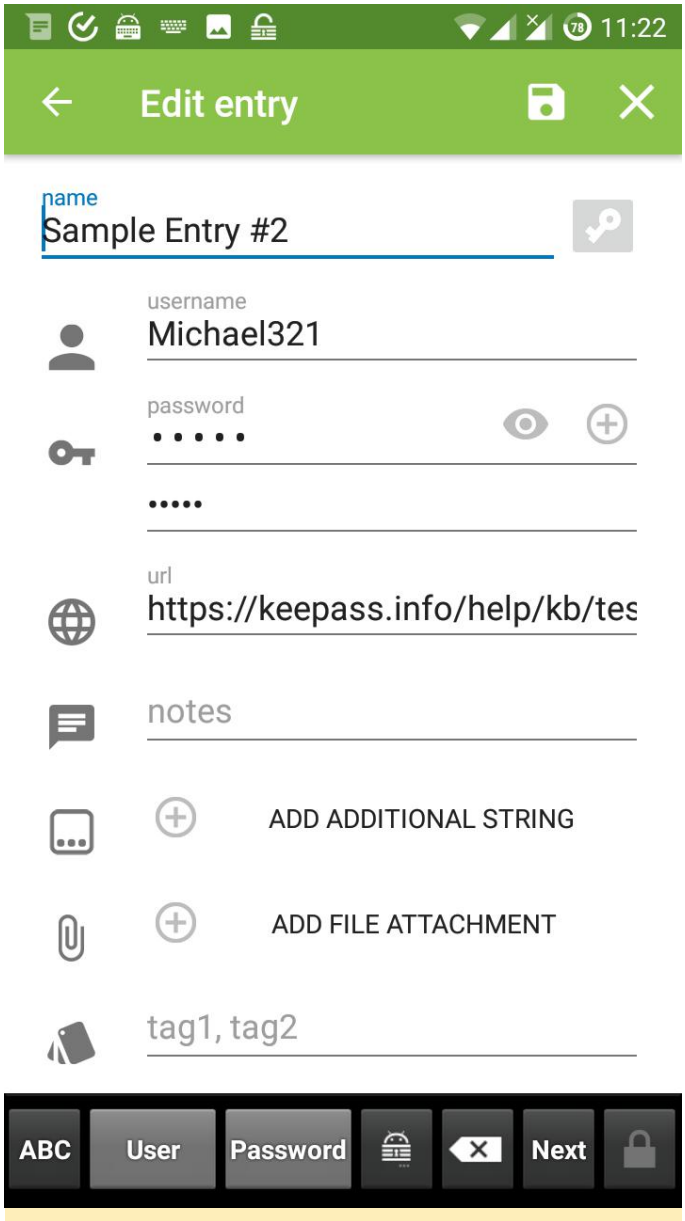


Figura 7 – KeePass2Android

Una vez desbloqueado, puede buscar o administrar tus contraseñas con normalidad. Cuando naveges por una página donde necesites iniciar sesión, puedes seguir estos pasos:

- Comparte la página desde el navegador a KeePass2Android: Buscar. A continuación, aparecerá una lista de todas las cuentas guardadas para ese sitio
- Selecciona la cuenta deseada y vuelve al navegador. El nombre de usuario/contraseña estará disponible por un tiempo, ya sea a modo de notificación o como botones en el teclado KP2A. Puede usar estos botones para copiar las credenciales y pegarlas en el navegador.

Importar contraseñas desde Firefox

Si está usando Firefox, posiblemente en muchos sistemas, sin sincronización, puedes exportar las contraseñas guardadas en tu administrador de contraseñas usando este script: https://github.com/unode/firefox_decrypt. Solía haber varias extensiones que te permitian hacerlo desde Firefox, pero desde que se mudaron a Quantum, las extensiones perdieron la capacidad de leer las contraseñas, lo cual debería ser algo bueno. Firefox también se está moviendo a un nuevo administrador de contraseñas llamado Lockbox, https://www.bleepingcomputer.com/news/software/firefox-to-get-a-better-password-manager/. Así que este método podría no funcionar cuando se produzca este cambio.

```
$ git clone
https://github.com/unode/firefox_decrypt.git
$ cd firefox_decrypt
$ ./firefox_decrypt.py -f csv -d , >
/dev/shm/firefox-passwords.csv
```

Se te preguntará qué perfil desea exportar y también cuál es la contraseña maestra (presiona Intro si no). También puede seleccionar una ruta personalizada para tu perfil, por ejemplo, si está importando contraseñas guardadas en un sistema Windows o desde un recurso remoto. Tus contraseñas se escribirán en un disco RAM, /dev/shm, para evitar dejar rastros en el sistema de archivos.

Antes de importar nada, necesitamos mejorar un poco los datos que vamos a importar. El problema es que los datos exportados solo tienen la URL, el usuario y la contraseña, y carece de un título. Además, la URL no contiene subdirectorios, por ejemplo www.domain.com/example, por lo que nos faltará información que debe ser solventada manualmente más adelante. Vamos a intentar visitar todos los sitios de la lista y coger el título de la página y agregarlos a una nueva lista. He escrito un pequeño script para hacer todo eto:

```
$ sudo apt-get install curl libtext-csv-perl
$ wget -O /usr/local/bin/enrichFirefox.pl
https://raw.githubusercontent.com/mad-
ady/enrichFirefoxPasswords/master/enrichFirefo
x.pl
$ sudo chmod a+x
/usr/local/bin/enrichFirefox.pl
$ /usr/local/bin/enrichFirefox.pl
/dev/shm/firefox-passwords.csv | tee
/dev/shm/firefox-passwords-enriched.csv
```

A continuación, puedes usar keepass2 e importar el archivo de contraseña con File -> Import -> Generic CSV Importer. Navega a /dev/shm y selecciona el archivo y presiona ok. Aparecerá una vista previa del archivo. Selecciona la pestaña "Structure" y marca la opción "Ignore first row" ya que es un encabezado. Ahora, necesitamos asignar los campos del CSV a los campos de KeePass en la siguiente sección. Elimina la primera entrada "URL" y la entrada "(Ignore)" y agrega "Title" a la parte superior y "Group" a la parte inferior y seleccione Next. Ahora debería ver tus contraseñas desglosadas correctamente y puede presionar Finish para importarlas.

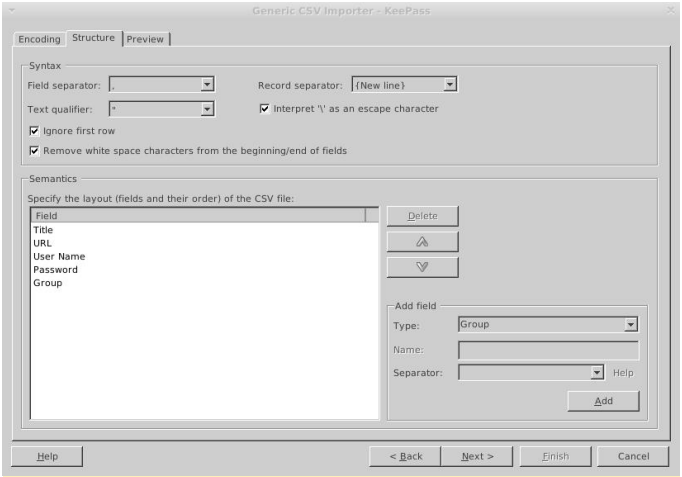


Figura 8 – Configurando la importación

Una vez importado, asegúrate de eliminar el archivo de contraseñas exportado y guarda tu base de datos:

```
$ rm -f /dev/shm/firefox-passwords*.csv
```

Puede volver a realizar estos pasos para importar datos desde otros perfiles de Firefox almacenados en múltiples sistemas.

Importar contraseñas desde Chrome

Para exportar contraseñas desde Chrome, debes navegar a chrome://flags y buscar "exportar contraseña". Deberás activar esta opción y reiniciar Chrome. Una vez reiniciado, dirígete a Menú -> Configuración -> Avanzado -> Administrar contraseñas. Debajo de la lista de contraseñas guardadas, haz clic en el menú de tres puntos y seleccione Exportar contraseñas y guárdalo también en /dev/shm.

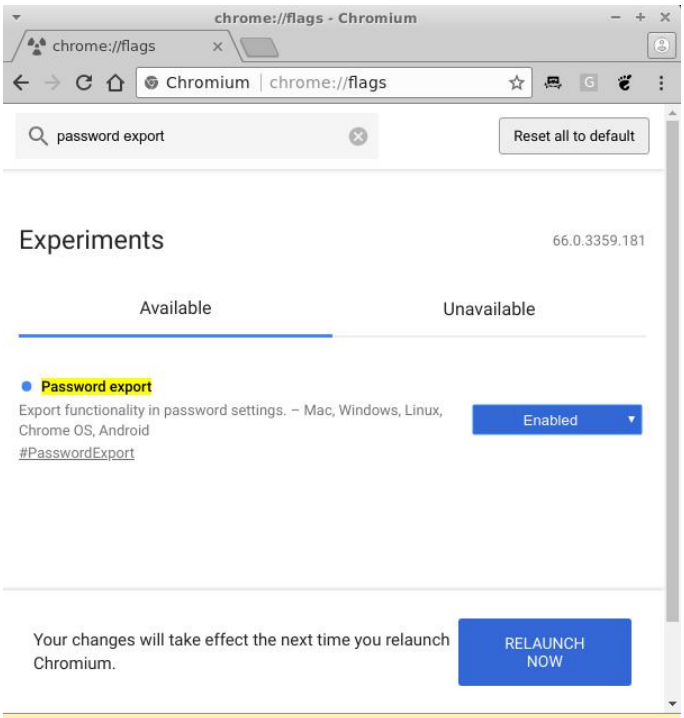


Figura 9 – Exportando contraseñas desde Chrome

Si no ves ninguna contraseña guardada localmente, tus contraseñas están disponibles en https://passwords.google.com y están sincronizadas entre varias instancias del navegador. A continuación, puede importarlas de la misma forma que las contraseñas de Firefox, pero la estructura CSV ahora es Nombre, URL, Nombre de usuario, Contraseña. Una vez que haya terminado, recuerda eliminar el archivo exportado y deshabilitar la opción "Exportar contraseñas".

Reemplazar el administrador de contraseñas de Firefox/Chrome – Tusk

Copiar/pegar contraseñas puede estar bien para los accesos ocasionales, pero usar un administrador de contraseñas integrado con KeePass se vuelve necesario para un uso más regular del navegador. Para esto, necesitamos usar una extensión del navegador para obtener contraseñas del servidor

web. Para Firefox y Chrome, una de esas extensiones es KeePass Tusk, https://addons.mozilla.org/en-US/firefox/addon/keepass-tusk/?src=search, https://chrome.google.com/webstore/detail/keepass-tusk-password-acc/fmhmiaejopepamlcjkncpgpdjichnecm. Se puede conectar a una base de datos KeePass almacenada a través de WebDAV y no necesita un cliente keepass local. Puedes instalar la extensión dirigiéndote a: addons -> Extensions y buscando por "tusk". Selecciona el complemento y haz clic en "+ Add a Firefox". Para Chrome, se puede instalar desde el propio enlace de arriba. Una vez que esté instalado, aparecerá un icono al lado de la barra de búsqueda donde puedes configurarlo y conectarlo a una base de datos de contraseñas. Deberás seleccionar "Cloud storage set" y activar "WebDAV". Puede ignorar la advertencia sobre el almacenamiento del nombre de usuario y la contraseña de WebDAV en disco.

Necesitarás completar la ruta al directorio kppassword con http, lamentablemente no le gusta mi certificado autofirmado, aunque no la ruta a la base de datos porque se descubrirán todas las bases de datos de ese directorio. Además, añade tu nombre de usuario y contraseña de WebDAV.

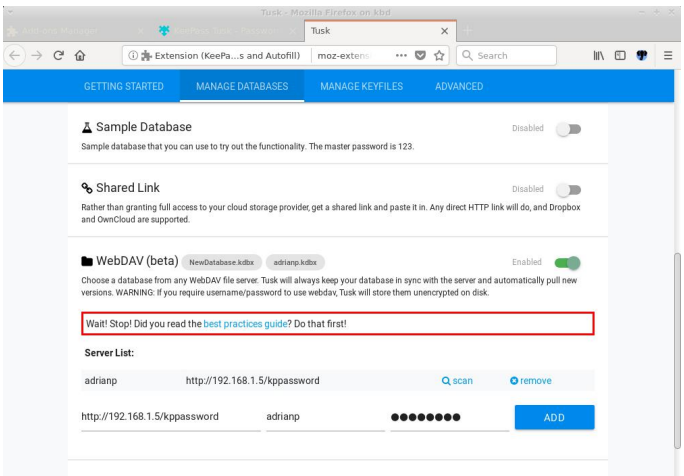


Figura 10 – Configuración de Tusk

Ahora puedes cerrar la pestaña Tusk y cuando hagas clic en su icono, te preguntará qué base de datos desea cargar y te pedirá la contraseña maestra y el tiempo que quieres mantenerla abierta. Una vez que desbloquees tu base de datos y te dirijas a un sitio que tengas registrado en tu base de datos, Tusk lo buscará automáticamente en tu base de datos, pero no la autocompletará. Puedes hacer clic en el ícono Tusk y en el ícono "Autocompletar" que se encuentra junto a la entrada que deseas completar completar. Si activas la navegación mediante atajos de teclado en la configuración de Tusk, puedes usar la siguiente combinación para hacer lo mismo: CTRL+ALT+Space Tab Enter.

Para beneficiarse completamente de esto, deberás desactivar el administrador de contraseñas integrado en Firefox dirigiendote a Menú -> Preferencias -> Privacidad y seguridad -> Desmarca Recordar inicios de sesión y contraseñas para sitios web. También debe eliminar logins.json de tu carpeta de perfil. Deberías hacer lo mismo para Chrome.

Una limitación de Tusk es que es de solo lectura. Si necesitas actualizar una contraseña, necesitará usar un cliente diferente para llevar a cabo la actualización.

Reemplazar el administrador de contraseñas de Firefox/Chrome – chromeIPass/KeePassHelper

Si no quieres que tu plugin se conecte directamente a tu base de datos de contraseñas y revele potencialmente tu contraseña maestra en el navegador, puede usar una técnica diferente y realizar solicitudes proxy a través de KeePass2. Plugins que hacen esto: chromeIPass https://chrome.google.com/webstore/detail/chromeipass/ompiailgknfdndiefoaoiligalphfdae/, KeePassHelper para Firefox https://addons.mozilla.org/en-US/firefox/addon/keepasshelper/) se conneca vía HTTP localmente a la instancia KeePass2 para recuperar contraseñas. Para hacer esto, debe instalar el plugin KeePassHTTP dentro de KeePass2, https://keepass.info/plugins.html#keepasshttp.

Puedes hacer esto en el sistema donde ejecutas KeePass2 de la siguiente manera:

```
$ cd /usr/lib/keepass2/Plugins
$ sudo wget
https://raw.github.com/pfn/keepasshttp/master/
KeePassHttp.plgx
$ sudo apt-get install mono-complete
```

A continuación, deberás reiniciar KeePass2 y abrir tu base de datos de contraseñas, y podrás instalar el

plugins en el navegador. Una vez que el plugins esté instalado, se conectará automáticamente a KeePass2 y generará una clave de cifrado que debes aprobar. Luego, al conectarte a un sitio conocido con el navegador, se completará automáticamente el nombre de usuario/contraseña en los campos de inicio de sesión (para Chrome).

Figura 11 – Inicio de sesión de Chrome con chromeIPass

Mantenimiento

Ahora que tiene todas tus contraseñas en un solo lugar, puede ejecutar diversos informes desde KeePass2, como Edit -> Show entries -> Find duplicate passwords y verás un informe de tu adicción a la reutilización de contraseñas. Necesitarás buscar algo de tiempo y tomar medidas, restablecer esas contraseñas y reemplazarlas por algunas otras para que sean más seguras.

Una última cosa que debe tener presente, que es muy importante, es la copia de seguridad. Tu base de datos de contraseñas contiene todas tus identidades digitales y si se pierde o se corrompe te dará un mal día. Es por eso que necesita configurar una estrategia de copias de seguridad y hacer copias de tu base de datos en diferentes discos físicos y también en diferentes ubicaciones físicas, con el objeto de disponer de redundancia geográfica, de modo que, en el pero de los casos, si un asteroide destruye tu ciudad, tus contraseñas seguirán estando seguras.

Para hacer esto, he creado un script que pone atención a los cambios de archivos, compara el archivo actual con su versión anterior, para ver si habido algun cambio y si es necesario lo copia a otras unidades y a otros sistemas a través de los recursos de la red.

Necesitarás inotify para activar las sincronizaciones:

```
$ sudo apt-get install inotify-tools
$ sudo wget -O /etc/systemd/system/password-backup.service https://raw.githubusercontent.com/mad-ady/password-backup/master/password-backup.service
$ sudo wget -O /usr/local/bin/password-backup-detect.sh https://raw.githubusercontent.com/mad-ady/password-backup/master/password-backup-detect.sh
$ sudo wget -O /usr/local/bin/password-backup-execute.sh https://raw.githubusercontent.com/mad-ady/password-backup/master/password-backup-execute.sh
$ sudo chmod a+x /usr/local/bin/*.sh
$ sudo systemctl enable password-backup
$ sudo systemctl start password-backup
```

El código consta de dos scripts bash: uno para detectar cambios y otro para realizar la copia de seguridad, y un archivo de servicio que ayuda con el inicio al arrancar.

Vamos a detenernos un segundo para analizar el script password-backup-detect. En la línea 2 fijamos la ruta de la ubicación de la base de datos de contraseñas, luego iniciamos un bucle infinito. Usamos inotifywait para prestar atención a los cambios en los archivos existentes y cuando un archivo se cierra usamos find para obtener una lista de los archivos modificados en los últimos 4 segundos y para cada archivo activaremos el script de copia de seguridad con el nombre del archivo como argumento. Una vez que finaliza el script password-backup-execute.sh, volveremos a escuchar los cambios.

password-backup-detect.sh × password-backup-execute.sh ×

```
#!/bin/bash
DIR=/var/www/html/kppassword

while [ : ];
do
    #wait for a file to change
    echo "Waiting for a change in $DIR"
    inotifywait -e close_write,move,delete "$DIR/"*.kdbx
    #wait for things to stabilize
    sleep 2
    #find which file changed a few seconds ago
    find "$DIR" -name '*.kdbx' -type f -newermt '-4 seconds' -execdir /usr/local/bin/password-backup-execute.sh {} \;
done
```

Figura 12 – password-backup-detect

El script password-backup-execute coge un archivo del que se realiza una copia de seguridad como argumento, comprueba si es necesario hacer una copia de seguridad y la realiza en múltiples ubicaciones potenciales. Al principio, fijamos la cantidad de versiones de la copia de seguridad que queremos conservar y configurarmos las rutas para los archivos de copia de seguridad; puede usar autofs para montar un recurso compartido remoto cuando lo necesites, por ejemplo. A continuación, definimos una función doBacku() que itera a través de la lista de carpetas de destino y realiza la copia real y una función cleanup() que usa una combinación de ls, tail y rm para eliminar copias de seguridad anteriores y preservar algunas versiones más recientes. Los archivos respaldados contienen una marca de tiempo en su nombre, aunque se ordenan en función de la última hora de modificación.

password-backup-detect.sh × password-backup-execute.sh ×

```
#!/bin/bash

echo "Backing up $1"
backupsToKeep=20
# Change the destinations array to point to where you want to save the files
destinations=()
destinations[0]=/backup/kppassword
destinations[1]=/media/ssd/ssd250/backup/kppassword

fullfile=$1
file=`basename "$1"`;
let "backupsToKeep++";

doBackup() {
    date=`date +%Y%m%d_%H%M%S`
    #copy to each destination
    for dst in ${destinations[@]}; do
        cp -av "$fullfile" "$dst/$date-$file"
    done
}

cleanup() {
    # find all backed up files with $file name and sort them by date. Keep last $backupsToKeep
    for dst in ${destinations[@]}; do
        cd "$dst"
        ls -1pt "$dst/"*-$file | tail -n +$backupsToKeep | xargs rm -fv
    done
}

#see if the file is the same as the last one
if [ -f "/tmp/$file" ]; then
    # check the checksum of the old backup
    md5sum "$fullfile" > "/tmp/.$file"
    cmp "/tmp/.$file" "/tmp/$file"
    if [ "$?" -ne 0 ]; then
       # the file differs from the last backup
       doBackup
       cleanup
       mv "/tmp/.$file" "/tmp/$file"
    else
       # it's unchanged, skip backup
       echo "File is unchanged, skipping backup"
    fi
else
   # I don't know what the last backup was, do a backup now
   doBackup
   cleanup
   md5sum "$fullfile" > "/tmp/$file"
fi
```

Figura 13 – password-backup-execute

El código principal comprueba si hay un hash md5sum de la copia de seguridad anterior, y si lo hay, se compara con el hash actual. Si los valores hash difieren, o no hay hash previo, la copia de seguridad y la limpieza se llevan a cabo. La comprobación se realiza para evitar que se haga copias de seguridad del mismo archivo varias veces, por ejemplo. haces clic en guardar en KeePass2 sin haber hecho cambios.

Tendrás que modificar este código y agregar tus ubicaciones para las copias de seguridad

El código puede parecer perfecto, pero tiene dos deficiencias, “bug” es una palabra demasiado desagradable. En primer lugar, inotifywait no reacciona ante los archivos nuevos hasta que se reinicie o hasta que se modifique o se toque un archivo anterior. En segundo lugar, durante la fase de respaldo, los cambios de archivo no se detectan. Si un archivo tarda un minuto en guardarse en todos los destinos remotos (debido a la latencia de la red o al tiempo de activación del disco duro), los cambios de archivos que tengan lugar durante ese tiempo no se recogerán y no se guardarán copias de seguridad. De modo que, el script es adecuado para pocos usuarios simultáneos y con cambios poco frecuentes.

Espero que este artículo haya sido lo suficientemente interesante como para convencerte de hacerte cargo de tus contraseñas y tener mejores prácticas de seguridad. Con respecto a la seguridad, keepass te protege de tener tu contraseña expuesta cuando se piratea un servicio online, pero no necesariamente te protege de un ordenador infectado con malware o un registrador de teclas. El malware o los atacantes que pueden leer arbitrariamente la memoria de tu ordenador pueden acceder a las contraseñas una vez descifradas o pueden hacerse con tu contraseña maestra. La razón por la que recomendé usar un ODROID con WebDAV en lugar de usar Dropbox (que también es compatible) fue para minimizar su exposición. Mantener tu archivo encriptado en el ordenador de otra persona (por ejemplo, “la nube”), puede estar bien ahora, pero tus contraseñas pueden llegar a manos equivocadas y la contraseña maestra se puede descifrar más fácilmente en el futuro, dejándote nuevamente vulnerable.

Conceptos Básicos de BASH – Parte 4: Variables, Pruebas y Bucles

August 1, 2018 By Erik Koennecke Linux, ODROID-C0, ODROID-C1+, ODROID-C2, ODROID-HC1, ODROID-MC1, ODROID-XU4, Tutoriales

En esta ocasión me voy a centrar en lo más básico a la hora de crear scripts: las variables, las pruebas y los bucles. La frase para crear archivos de diferentes carpetas también nos hará compañía en otras ingeniosas líneas de comandos. Pero primero, vamos a aprender algunos atajos rápidos que harán que el trabajo con BASH sea más ameno.

Atajos de BASH

Si quieres ejecutar un único comando como un usuario diferente, usa el siguiente comando:

```
$ su - otheruser -c "command argument"
```

Para que dos comandos se ejecuten uno tras otro, únelos con ';', por ejemplo:

```
$ ls /home/odroid/Music; ls
/home/odroid/Videos
```

Si quieres que el segundo comando se ejecute únicamente cuando el primer comando tenga exitoso, utiliza '&&':

```
$ apt update && apt full-upgrade
```

Rara vez se utiliza '||', donde el segundo comando sólo se ejecuta cuando el primer comando NO tiene éxito.

En el día a dia, hay determinados pasos que solemos hacer con bastante frecuencia en diferentes contextos. Un ejemplo es ejecutar un comando como root tras haberlo probado con un usuario normal y éste falla por no disponer de los privilegios suficientes.

Después de ejecutar el comando cp somefilewithverylongname.conf /etc para copiar una configuración en el directorio /etc y éste falla, podemos repetir este proceso como root con solo

'sudo !!' – la mayoría de vosotros ya lo sabréis. Los !! significa repetir el último comando y se puede usar con sudo, sin o con otras combinaciones. Alterado con el modificador '!!:p' recupera el último argumento de la línea de comandos. Por lo general, basta con usas la tecla de flecha hacia arriba. ¡Pero en los casos en los que las teclas de flecha no funcionen, por ejemplo en ssh, puede serte de gran ayuda! ! -1 usa el penúltimo comando, que en determinadas ocasiones también es muy útil. Sin embargo, ¿sabías que también puedes reutilizar únicamente el último argumento de un comando?

```
$ ls /very/long/path/to/a/directory
$ cd !$
```

se amplía a:

```
$ cd /very/long/path/to/a/directory
```

Lo cual te ahorra mucha escritura.

```
$ ls /very/long/path/to/a/directory
```

Toneladas de comandos, ninguno de los cuales empiezan con ls

```
$ !ls
```

también se amplía a

```
$ ls /very/long/path/to/a/directory
```

En el caso de que quieras repetir un comando anterior. Usa '!Rm:p' para examinar el último comando rm antes de ejecutarlo, al igual que ocurre con otros comandos peligrosos. Si solo quieres cambiar algunos detalles del último comando, puede hacer un buscar y reemplazar de los argumentos del siguiente modo:

```
$ ls /very/long/path/to/a/directory
$ ^very^veryvery
```

cambia y ejecuta el último comando a

```
$ls /veryvery/long/path/to/a/folder
```

Para resumir lo que hemos visto hasta ahora:

- !! último comando
- !-1 penúltimo comando
- !$ último argumento del último comando
- !command1 última línea con command1
- ^searchterm^replaceterm reemplaza la primera vez que aparece searchterm por replaceterm

Para moverse por la línea de comando, alt-f mueve el cursor hacia adelante una palabra, alt-b hacia atrás una palabra, ctrl-w borra palabras sueltas hacia atrás, mientras que ctrl-u borra desde la posición del cursor hasta el comienzo de la línea. Puedes moverte al inicio de la línea con ctrl-a y borrar la línea tras el cursor con ctrl-k. Ctrl-t intercambia los dos últimos caracteres antes del cursor para los casos en lo que se suele cometer este error al escribir.

Si quieres algo que lo aligere y sueles escribir erroneamente sl en lugar de ls con bastante frecuencia, también puede instalar sl con apt install sl. No te diré lo que hace, simplemetente pruébalo. Lo más importante, si tienes un nombre de archivo o comando único después de escribir las primeras letras, BASH lo expenderá tras presionar el tabulador. Al presionarlo dos veces, aparecerá una lista de opciones si las primeras letras no son únicas.

Una última cosa, por ahora, el uso de las llaves. Si tienes un archivo llamado abcde.conf y quieres hacer una copia de seguridad del éste con el nombre abcde.conf.bak, todo lo que tiene que hacer es cp abcde.conf{,.Bak} que se amplía a cp abcde.conf abcde.conf.bak al utilizar las llaves. Si deseas listar el contenido del directorio de Videos de los usuarios archie, bert y claude, tienes que colocar éstos dentro de las llaves. El filtro se encarga de hacer el resto.

```
$ ls /home/{archie,bert,claude}/Videos
```

Conceptos básicos de programación

Un script BASH es simplemente un archivo de texto con la primera línea #!/Bin/bash y se hace ejecutable con:

```
$ chmod a+x scriptname.sh
```

Si creas un directorio llamado bin en tu directorio de inicio, los scripts que estén dentro se podrán ejecutar desde cualquier lugar en Ubuntu. Una entrada especial en ~/.profile se encarga de esto.

Si quieres, incluso puedes programar el juego “Tetris” en BASH en un poco más de 500 líneas, tal y como se muestra en la Figura 1.

Figura 1 – Tetris en BASH sobre líneas de comandos

Un típico ejemplo de un script sería hello-world.sh, no olvides ejecutar chmod a+x hello-world.sh después de haberlo guardado desde tu editor de pruebas.

```
#!/bin/bash
# This script just puts out "Hello World".
echo "Hello World"
```

Excepto en casos especiales, todo el texto que aparece tras el signo # son comentarios y no se ejecuta.

Si sólo tienes “Hello World” en pantalla, o cualquier otro texto fijo, esto se vuelve aburrido muy rápido. Es el momento de introducir variables BASH para hacer que el script haga algo diferente dependiendo de la entrada. La forma más simple de verlo es de la siguiente manera en nuestro archivo hello-user.sh.

```
#!/usr/bin/bash
# Greets currently logged-in user
echo "Hello," "$USER"
```

“Hola, odroid” es el resultado. También podemos definir la variable en el script en lugar de usar una variable de entorno como USER. Aquí tienes nuestro próximo archivo script, hello-user2.sh

```
#!/usr/bin/bash
# Greets currently logged-in user
user=$(whoami)
echo "Hello," "$user"
```

“Hola, odroid” es el mismo resultado, pero una variable user es definida por el resultado de la función whoami y luego se imprime con la función echo. También puedes simular esto paso a paso en la línea de comandos sin escribir el script con el editor de texto. Si defines una variable $user, no olvides usar después unset user para dejar limpio el sistema. Hablaremos más sobre las variables en la siguiente parte. Por ahora, vamos a ver un ejemplo de cada componente básico creado en primer lugar para tener una mejor visión de conjunto del típico uso de un script. El siguiente elemento fundamental son las pruebas dentro del script.

En un ejemplo real, vamos a echar un vistazo a un pequeño script para probar la conectividad a Internet, outside-connected.sh:

```
#!/bin/bash
test=google.com
if
nc -zw1 $test 443
then
echo "we have connectivity"
else
echo "no outside connectivity"
fi
```

El script define la variable $test como el servidor de google.com, luego usa netcat (nc) en modo escaneo de puertos para un poke rápido, -z es el modo E/S cero, con un tiempo de espera pequeño –w 1 espera como máximo uno segundo. Comprueba Google en el puerto 443 (HTTPS). El resultado depende de si puedes acceder a los servidores de Google o no.

Ahora, veamos los bucles. Con variables, pruebas y bucles, ya tiene cubierto el 95% del uso normal de los scripts. Un simple bucle en un script real sería convertir todos los archivos flac en mp3 de un determinado directorio:

```
flac2mp3.sh
#!/bin/bash
for i in *.flac
do
ffmpeg -i "$i" -acodec libmp3lame "$(basename
"${i/.flac}").mp3"
done
```

Este script realiza un bucle, convierte y cambia el nombre de cada archivo flac que está dentro del

directorio actual. Echa un vistazo a la función basename junto con la variable a la hora de cambiar la extensión .flac a .mp3 en este ejemplo. Estos son los ejemplos más básicos de variables, bucles y pruebas; seguiremos más tarde. En la siguiente parte, continuamos con la programación y también echamos un vistazo a la historia de BASH.

Referencias

https://raw.githubusercontent.com/kt97679/tetris/master/tetris.sh
https://www.tldp.org/LDP/abs/html/

Detección de Objetos en Vídeo en Directo Usando el ODROID-XU4 con GStreamer

August 1, 2018 By Marian Mihailescu ODROID-XU4, Tutoriales

El aprendizaje profundo se ha vuelto un tema bastante importante en los últimos años, y muchas empresas han invertido en redes neuronales de aprendizaje profundo, ya sea en términos de software o hardware. Uno de los campos más utilizados en el aprendizaje profundo ha pasado a ser la detección de objetos; por ejemplo, las fotografías tomadas con nuestros teléfonos no se clasifican automáticamente en categorías usando la detección de objetos de aprendizaje profundo.

En este artículo, investigamos un nuevo uso del ODROID-XU4: crear una cámara de seguridad inteligente que pueda detectar objetos de interés en la fuente sobre la marcha y actuar en consecuencia. Utilizaremos el módulo dnn de OpenCV para cargar una red de detección de objetos previamente adiestrada basada en el detector de disparo único MobileNets. El artículo está inspirado en la excelente serie introductoria sobre detección de objetos de Adrian Rosebrock publicada en su blog, PyImageSearch. En las pruebas de Adrian, un SBC de baja potencia como la Raspberry Pi ni siquiera es capaz de alcanzar 1 fps cuando realiza la detección en tiempo real. Por lo tanto, no me voy a detener en los conceptos básicos de la detección de objetos y OpenCV, sobre los cuales puedes leer en sus publicaciones, sino que me voy a centrar en optimizar el SBC ODROID-XU4 para lograr la máxima tasa de fotogramas por segundo en tiempo real en una transmisión en vivo.

CPU vs GPU

Lo primero que debemos determinar es si la GPU ODROID puede ayudarnos a acelerar la detección mediante el uso de OpenCL. ARM mantiene la ARM Compute Library, una librería de aprendizaje automático y visión optimizada para las GPUs Malí.

Sin embargo, mis descubrimientos me han llevado a pensar que los núcleos quad-core de 2Ghz A15 proporcionan un rendimiento mucho mejor que los 6 núcleos 700Mhz de la GPU Mali en el ODROID-XU4. Puedes leer más sobre estos resultados en el foro https://forum.odroid.com/viewtopic.php?f=95&t=28177.

En mis pruebas, usar los 8 núcleos también es perjudicial, ya que los pequeños núcleos ARM por lo general ralentizan en tiempo de detección. Para asegurarnos de que estamos utilizando únicamente los potentes núcleos A15, debemos ejecutar nuestro programa de detección utilizando el conjunto de tareas 0xF0. También es recomendable utilizar una adecuada refrigeración para mantener la frecuencia máxima de los núcleos A15.

Optimizaciones de OpenCV

A continuación, compilaremos la última versión de OpenCV, que proporciona un módulo de aprendizaje profundo y que está optimizarlo para ODROID-XU4. Para esto, actualizamos CPU_NEON_FLAGS_ON en cmake/OpenCVCompilerOptimizations.cmake y así poder usar -mfpu=neon-vfpv4 en lugar de -mfpu=neon, habilita Threading Building Blocks (TBB) con los delimitadores -DWITH_TBB=ON -DCMAKE_CXX_FLAGS="- DTBB_USE_GCC_BUILTINS=1" y asegúrate de que se utilizan los siguientes delimitadores de compilación: -mcpu=cortex-a15.cortex-a7 -mfpu=neon-vfpv4 -ftree-vectorize -mfloat-abi=hard fijando C_FLAGS, CXX_FLAGS, -DOPENCV_EXTRA_C_FLAGS y -DOPENCV_EXTRA_CXX_FLAGS. También debemos asegurarnos de que la libería GStreamer esté disponible para OpenCV utilizando el delimitador -DWITH_GSTREAMER=ON. Los paquetes precompilados de Ubuntu 18.04 para OpenCV y GStreamer están disponibles en mi repositorio en https://oph.mdrjr.net/memeka/bionic/.

Con solo optimizar la CPU y OpenCV, ya podemos alcanzar los 3fps usando el mismo código que se ejecuta en la Raspberry Pi y que solo obtiene ~ 0.9fps. Vamos a ponerlo a prueba y a mejorarlo.

GStreamer

En lugar de usar OpenCV para conectar la cámara, podemos usar GStreamer. Esto nos permitirá varias cosas: conectarnos a cámaras inalámbricas de la red, usar el decodificador de hardware, el codificador de hardware y el escalador de hardware del ODROID. Podemos usar el decodificador de hardware para procesar H264 desde una transmisión en vivo o desde una cámara H264, el escalador de hardware para cambiar la resolución de la imagen y el formato de píxel y el codificador para producir una secuencia codificada H264, ya sea para guardarla en un archivo o para transmitirla. Esto también nos aportará una pequeña mejora en el rendimiento general. Algunos ejemplos de tuberías GStreamer son:

Conectarse a la transmisión H264 desde la cámara:

```
$ v4l2src device=/dev/video1 do-timestamp=true
! video/x-h264, width=1280, height=720,
framerate=15/1 ! v4l2h264dec !
v4l2video20convert ! appsink
```

Conectarse al flujo MJPEG/YUV desde la cámara:

```
$ v4l2src device=/dev/video0 do-timestamp=true
! video/x-raw, width=1280, height=720,
framerate=15/1 ! v4l2video20convert ! appsink
```

Guardar el resultado en un archivo mp4:

```
$ appsrc ! videoconvert ! v4l2h264enc extra-
controls="encode,frame_level_rate_control_enab
le=1,video_bitrate=8380416" ! h264parse !
mp4mux ! filesink location=detected.mp4
```

Emitir la transmisión por la web con HLS:

```
$ appsrc ! videoconvert ! v4l2h264enc extra-
controls="encode,frame_level_rate_control_enab
le=1,video_bitrate=8380416" ! h264parse !
mpegtsmux ! hlssink max-files=8 playlist-
root="http://0.0.0.0/hls" playlist-
location="/var/www/html/hls/stream0.m3u8"
location="/var/www/html/hls/fragment%06d.ts"
target-duration=30
```

Procesamiento por lotes multiproceso

Con estas mejoras y un modelo de subprocesos múltiples en el que buscar el siguiente fotograma se ejecuta de forma independiente en un subproceso diferente a la detección de objetos, el ODROID-XU4

puede llegar hasta los 4 fps: en un segundo, puede detectar objetos en 4 imágenes. Puesto que la detección es el objetivo principal, 4 fps en realidad es suficiente para avisarnos de la presencia de los objetos que son de nuestro interés. De modo que podemos tener un flujo de entrada con una tasa de fotogramas mayor y selectivamente seleccionar fotogramas para la detección de objetos.

Para mantener la ilusión de que cada fotograma se procesa, hacemos un simple truco: cuando se detecta un objeto, resaltamos su posición tanto en el fotograma procesado como en los fotogramas siguientes hasta la próxima detección. La posición perderá precisión cuando el objeto se mueva, pero dado que somos capaces o procesamos hasta 4 fps, el error será bastante pequeño. Usamos una cola de espera para leer los fotogramas de la secuencia de entrada, y procesamos n fotogramas a la vez: el primer fotograma se usa para la detección, y el procesamiento posterior se realiza para todos los n fotogramas en función de los objetos detectados en el primer fotograma. Elegimos n, el tamaño del lote, como la función de la velocidad de los fotogramas del flujo de entrada, y las capacidades de procesamiento del ODROID-XU4.

Por ejemplo, para una entrada con 15fps, podemos usar n=4 (detección de ejecución para 1 en 4 fotogramas) para maximizar la utilización. El código en Python para esto es bastante simple:

```
# function to read frames and put them in
queue
def read_frames(stream, queue):
global detect
while detect is True:
(err, frame) = stream.read()
queue.appendleft(frame)

# start reader thread
detect = True
reader = threading.Thread(name='reader',
target=read_frames, args=(vin, queue,))
reader.start()

# grab a batch of frames from the threaded
video stream
frames = []
for f in range(n):
while not queue:
# wait for n frames to arrive
time.sleep(0.01)
frames.append(queue.pop())
frame_detect = frames[0]
```

Objetos de interés

Definimos los objetos de interés desde las clases que MobileNets SSD puede detectar. Estas clases incluyen “persona”, “pájaro”, “gato”, “perro”, “bicicleta”, “automóvil”, etc. Queremos ser capaces de asignar diferentes niveles de confianza de detección para cada objeto, y también un tiempo de espera para la detección: p.ej. si se detecta el mismo objeto de interés en el siguiente fotograma procesado, no queremos recibir una nueva notificación (es decir, no queremos recibir 4 correos electrónicos cada segundo); Para ello, usamos un valor de tiempo de espera y obtendremos una nueva notificación cuando expire el tiempo de espera. El código en Python sería:

```
# check if it's an object we are interested in
# and if confidence is within the desired
levels
timestamp = datetime.datetime.now()
detection_event = False
if prediction in config['detect_classes']:
if not confidence > (float)
(config['detect_classes'][prediction]):
# confidence too low for desired object
continue
else:
# we detected something we are interested in
# so we execute action associated with event
# but only if the object class was not already
detected recently
if prediction in DETECTIONS:
prev_timestamp = DETECTIONS[prediction]
duration = (timestamp -
prev_timestamp).total_seconds()
if duration > (float)
(config['detect_timeout']):
# detection event (elapsed timestamp)
detection_event = True
else:
# detection event (first occurence)
detection_event = True
else:
if not confidence > (float)
```

```
(config['base_confidence']):
# confidence too low for object
continue
```

Detección de eventos y resultados

Por último, queremos tener dos acciones separadas que se lleven a cabo tras procesar un fotograma: la primera acción es independiente de los resultados de la detección, mientras que la segunda acción solo se aborda cuando se detectan los objetos de interés. En mi código de ejemplo, todos los fotogramas son modificados con un recuadro y una etiqueta alrededor de todos los objetos detectados (de interés o no). Estos fotogramas se guardan en la secuencia de salida, que puede ser un video via streaming. Por lo tanto, cuando se conecta remotamente a la fuente de seguridad, puedes ver el video procesado, que incluye cuadrados codificados por colores alrededor de los objetos en movimiento.

Cuando son detectados los objetos de interés, el fotograma también se guarda como un archivo jpeg y se pone a disposición de un script definido por el usuario que es responsable de notificar al usuario. Por ejemplo, la imagen puede enviarse por correo electrónico, usarse con IFTTT o enviarse directamente al teléfono móvil del usuario.

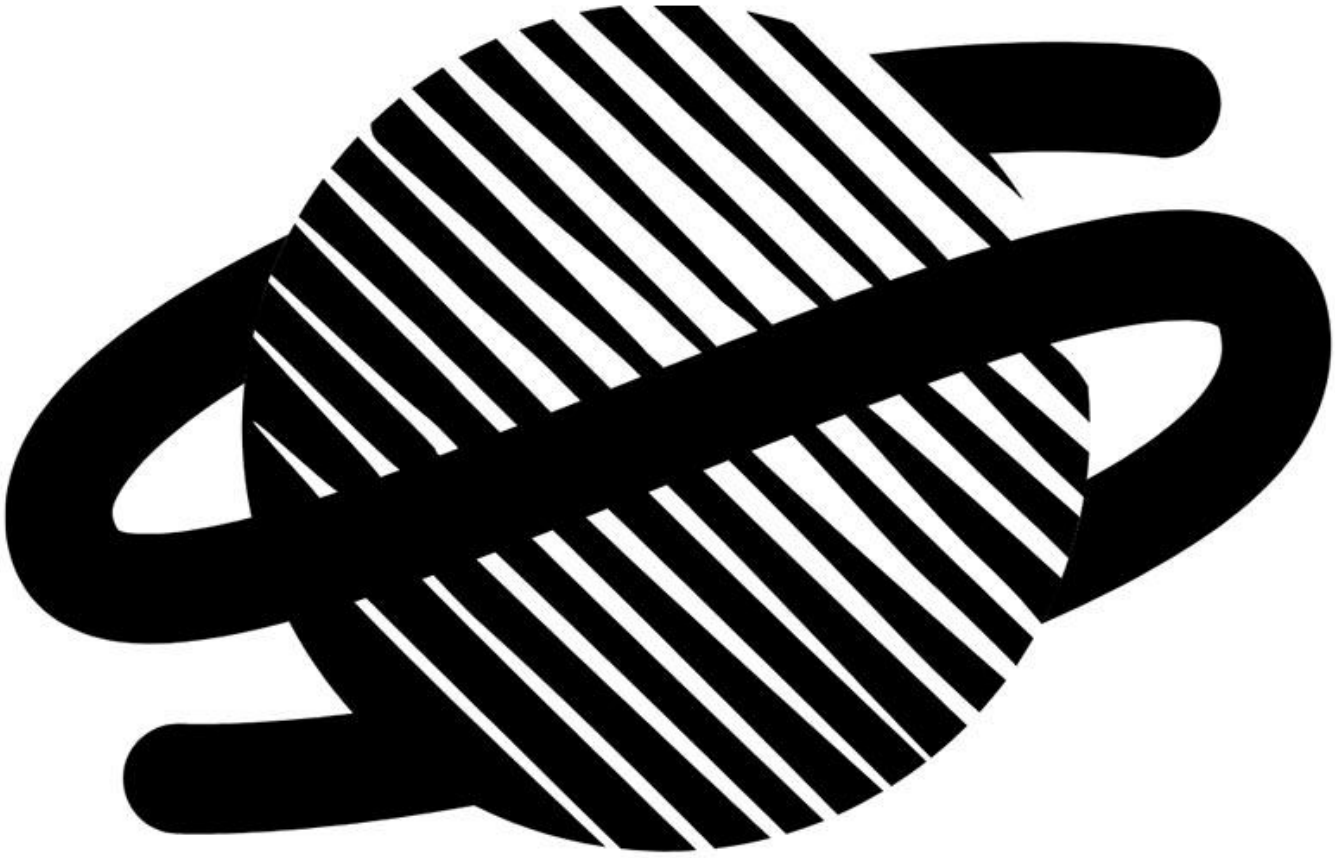
El código de ejemplo completo está disponible en **https://gist.github.com/mihailescu2m/d984d9fe3e3937573456c2b0423b4be9** y el archivo de configuración en formato json está en **https://gist.github.com/mihailescu2m/42fdccd624dc91bb9e04b3adc39bc50f**

Recursos

https://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/
https://www.pyimagesearch.com/2017/09/18/real-time-object-detection-with-deep-learning-and-opencv/
https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/

Juegos Linux: Juegos Saturn – Parte 5

August 1, 2018 By Tobias Schaaf



Y estamos de vuelta con el ODROID XU3/XU4 para continuar ejecutando juegos de Sega Saturn. En esta publicación cubriremos el resto de juegos (letra en el alfabeto). Los he probado y he disfrutado bastante jugando con mi ODROID. Una vez más, encontré algunas joyas realmente buenas que quiero compartir contigo. Este será probablemente el último número de la serie, aunque podría retomarla cuando los emuladores mejoren y haya más juegos disponibles o que se ejecuten en ODROID para la Sega Saturn. Por ejemplo, la versión más reciente del núcleo Yabause libretro en determinadas ocasiones funciona bastante bien y es posible ejecutar algunos juegos en este emulador, lo he tenido en cuenta.

Tenchi Wo Kurao 2 / Warrior of Fate

Se trata de un arcade muy bueno. En mi opinión, es un arcade perfecto. Puedes encontrarlo también en MAME, CPS1 y PlayStation. Si bien todas las versiones son bastante buenas, la versión de Saturn lo tiene todo en mi opinión. Los gráficos y los sonidos son como la contraparte del arcade y la música con calidad de CD es simplemente una mejora de un título que ya de por sí es increíble. También funciona muy bien en ODROID, así que lo recomiendo.

Figura 1: Elige uno de los cinco luchadores, todos con diferentes armas y ataques

Figura 2 – Sal y da caza al enemigo

El juego es el típico camorrista arcade, con toneladas de enemigos que te atacan desde la izquierda y la derecha. Los golpeas las suficientes veces y los tiras al suelo, derribas los suficientes y pasas a la siguiente sección. Al final de cada sección, te encuentras con un jefe que necesita muchos más golpes y te llevara bastante tiempo, además de recurrir a determinadas tácticas y habilidades para abatirlo.

Me gusta y se ejecuta increíblemente bien en el XU4, incluso si necesitas usar frame skipping. Aún así, el juego no se vuelve lento incluso con 10 o más enemigos en la misma pantalla. Los sprites son grandes y los fondos coloridos están muy bien trazados; realmente no hay nada de qué quejarse. Esta versión de saturn está muy conseguida, si te gustan los juegos de este tipo, te sugiero que lo pruebes.

Figura 3 – Combate del jefe en el segundo nivel, las barras de salud se hacen cada vez más largas

The Legend of Oasis / The Story of Thor 2

Si has jugado a Beyond Oasis en el Sega Genesis, sabes exactamente a lo que te enfrentas. Se supone que este juego es la “precuela”, pero se parece mucho a la versión de Génesis. Pero con gráficos mejorados. De hecho, es uno de los mejores juegos 2D que suelo mencionar.

Figura 4 – Magníficos gráficos en Legend of Oasis

Figura 5 – Muchas animaciones, cascadas, peces nadando, ondas de agua

Tu primera misión es bajar al calabozo y hacerte amigo del espíritu del agua. Similar a Beyond Oasis, encuentras espíritus elementales y los utilizas para resolver puzzles y luchar contra enemigos. Puedes recoger diferentes armas que tienen usos limitados, aunque siempre tienes tu daga que no expira. También puedes recolectar otros artículos, especialmente alimentos que puede usar para rellenar tu barra de salud.

Sin embargo, una gran parte del juego cambia con respecto a Beyond Oasis. Ya no encuentras elementos para aumentar tu nivel o tu salud, sino que "adiestras" a tu personaje: recibes los suficientes golpes, te sanas y obtienes más salud, cosas como esta. Es un concepto interesante, pero también un tanto extraño. El diseño del nivel puede ser difícil en determinadas ocasiones ya que a menudo te tienes que esforzar por descubrir a dónde tiene que ir y cómo subir o bajar de una plataforma. A menudo parece que puedes avanzar de una cierta manera, pero luego te das cuenta que tienes que subir más alto no pudiendo alcanzar las cosas. Puede que no sea el mejor juego de su tiempo, pero tiene buen aspecto y funciona bastante bien en el ODROID XU4.

Figura 6 – Puedes continuar usando tus movimientos de ataque de Beyond Oasis y moverte alrededor de tu personaje

Theme Park

Theme Park es uno de esos juegos a los que jugué bastante tiempo de niño. De hecho, recuerdo haber jugado a este juego en mi fiel Amiga y me ha sorprendido bastante la versión de Sega Saturn. Como en Amiga este juego lo controlas con el ratón, tenía mis dudas sobre cómo funcionaría en una consola con joystick/gamepad. Me sorprendió gratamente descubrir que en realidad funciona bastante bien. Me llevo algo de tiempo descubrir todos los controles, pero funcionan adecuadamente. Todo funciona como lo hacía en la versión de Amiga que yo recuerdo.

Figura 7 – Diferentes atracciones que construyo en mi parque temático

Figura 8 – El juego tiene una gran variedad de atracciones y tiendas donde elegir

Lo que también me gusta mucho son los videos de renderizado que puedes ver para cada atracción, lo cual te permite recrear el típico paseo de cuando eras niño y percibir precisamente cómo se ve la atracción desde el punto de vista de un niño.

Tienes que hacer un montón de micro gestiones en este juego. Contratar el personal que distribuyes por todo el parque, seleccionar donde quieres colocar las atracciones y qué tiendas quieres tener. Puede administrar los precios de las tiendas, la cantidad de hielo en la Coca-Cola o la sal de las patatas fritas. Debes fijar el precio correcto de la entrada y así sucesivamente.

Figura 9 – Gente entrando y saliendo en la parada de autobús

Es un juego interesante y divertido, y sí, todos solemos aumentar la velocidad de las montañas rusas hasta el punto de que la gente empiece a vomitar. Una desventaja de este juego es que solo puedes hacerlo funcionara con el nucleo libretro. No obstante, se ejecuta a muy buena velocidad, así que supongo que con Yabause también debería funcionar bien, pero lamentablemente el emulador independiente se cuelga. Tal vez con una versión más reciente se solucione este problema.

Time Bokan Series – Bokan to Ippatsu! Doronboo Kanpekiban

Para serte sincero, no tengo ni idea de qué va este juego, ya que está completamente en japonés. Aún así, este juego es bastante divertido y ofrece muchas escenas anime que parecen ser de una serie de anime real de los 80. Básicamente es un shooter vertical bastante rápido con colores brillantes y toneladas de cosas que van y viene por toda la pantalla.

Figura 10 – Selecciona uno de los 6 personajes con diferentes armas

Figura 11 – Este juego ofrece jefes a mitad de nivel y no solo jefes de final de nivel

Este juego no es fácil, pero es muy divertido, nada que te lo tengas que tomar demasiado en serio, pero los colores brillantes y las toneladas de enemigos en la pantalla son simplemente divertidos.

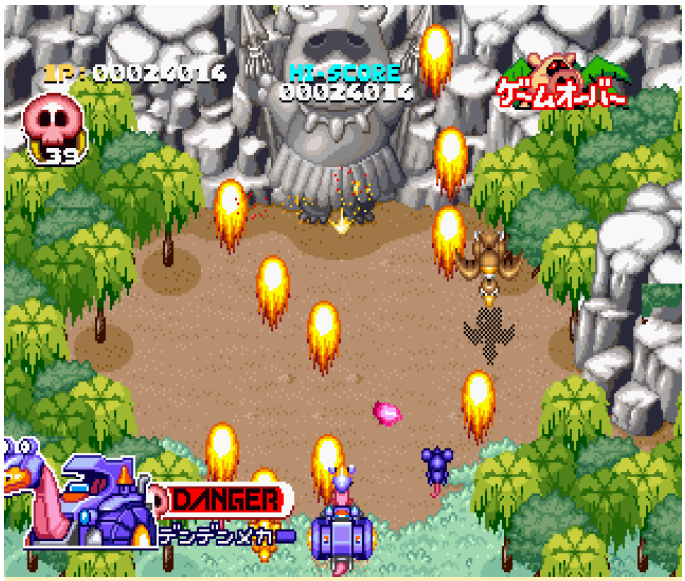


Figura 12 – Aumenta tu potencia de fuego y continúa

Este es uno de esos shooters casuales que retomas durante media hora y te sientes satisfecho. También existe para la PlayStation, así que depende de ti desde que plataforma quiere jugar.

Tryrush Deppy

Ni siquiera sé por dónde empezar a describir este juego. Tiene un estilo muy caricaturesco tanto en los gráficos como en la introducción del juego. Eres un cab (taxi) y participas en una carrera de coches, pero no es lo que podrías esperar. En este juego, los coches actúan como humanos y en realidad CAMINAN con sus neumáticos traseros (no bromeo).

Figura 13 – Tryrush Deppy, uno de los juegos más extraños a los que he jugado hasta ahora

El juego en sí resulta ser un juego de plataformas acelerado similar a Sonic, pero sin los anillos. Es muy rápido y puedes correr, saltar o esprintar por el mundo. Tienes que recoger “aceite” para continuar, pero hay más elementos que puedes recolectar y que te dan poderes temporales, como un escudo para que puedas atravesarlo todo o te vuelvas mucho más rápido durante un corto periodo de tiempo. El objetivo siempre es correr hasta el final del nivel y golpear el símbolo de meta (¿te suena familiar?).

Figura 14 – En la introducción aparece tú y otros coches “de pie” en línea esperando a que empiece la carrera

Una vez más, me gustan bastante los colores brillantes, así como el diseño de los dibujos

animados, ya que realmente se ajustan al juego. También puedes seleccionar tu propia placa de matrícula al inicio del juego para poder guardar tus progresos dentro del juego.

Figura 15 – Crea tu propia matrícula individual

Figura 16 – Sí, en este juego caminas sobre tus neumáticos traseros

Aún no he jugado demasiado, pero se pueden hacer muchas cosas además de correr y saltar. Hay policías que te persiguen por "exceso de velocidad". Hay villanos que son perseguidos por la policía y pueden arrollarte. Tu arrancada puede actuar como un ataque y puedes cargarlo para que corras más rápido y te estrelles más. Algunos enemigos pueden ser eliminados saltando sobre su cabeza o estrellándote contra ellos. Hay objetos difíciles de alcanzar, caminos ocultos y mucho más.

En resumidas cuentas, se trata de un juego muy divertido y realmente disfruto jugándolo. Encontré un pequeño problema, el sprite del personaje a veces falla en un fotograma o algo así y luego vuelve a ser normal. No estoy seguro si se debe al emulador o a una ROM defectuosa. Si te gustan las plataformas (juegos similares a Sonic o Mario) deberías probar este juego. Además, ¿He mencionado que también hay combates con jefes en este juego? En algunos puntos tienes que chocar contra monstruos enormes que tienes que golpear con tu ataque de arrancada en determinados puntos.

Twinkle Star Sprites

Twinkle Star Sprites es un pequeño y divertido juego de Saturn. También tengo el juego para Dreamcast, pero lamentablemente es extremadamente lento en el emulador dreamcast reicast. Afortunadamente para nosotros, la versión de Saturn funciona bastante bien.

Figura 17 – Twinkle Star Sprites en Sega Saturn

En este juego de disparos al estilo anime, puedes seleccionar entre muchos personajes diferentes y luchas contra tu oponente no disparándole directamente, sino disparando a los monstruos y objetos que vienen hacia ti evitando que te golpeen. Si destruyes suficientes objetos en la pantalla, provocas más cosas en la pantalla de tus oponentes, lo cual hace que le sea más difícil evitarlos. Si tú o tu oponente sois golpeados con frecuencia, pierdes o ganas.

Figura 18 – Twinkle Star Sprites es un juego muy al estilo anime que se puede apreciar en las pantallas de carga

Figura 19 – Twinkle Star Sprites tiene muchos personajes diferentes donde elegir

El sistema de juego es bastante fácil: dispara a todo y evita lo que no puedes destruir. Es muy divertido y se ve muy bien. Me encanta el color brillante y sí, incluso los destellos y las explosiones que llegan a cubrir casi toda la pantalla están muy conseguidas. No estoy seguro de por qué la versión de Dreamcast se ejecuta tan lenta, aunque estoy muy contento de poder jugar a la versión de Sega Saturn, que el ODROID-XU4 puede manejar muy bien. Tienes un medidor de potencia que te permite disparar un ataque cargado que hace más daño y puede golpear a más enemigos a la vez. También puede tener un nivel diferente dependiendo de tu carga. Además, tienes un número limitado de bombas que puedes usar para destruir una gran cantidad de enemigos en pantalla.

Figura 20 – Para destruir un jefe de tu lado lanza un gran ataque sobre tu oponente

Vampire Hunter Darstalkers' Revenge

Aunque por lo general no soy un gran fan de los juegos de lucha como Street Fighter, este es un juego con el que realmente disfruto jugando y lo hago precisamente con Sega Saturn.

Figura 21 – Vampire Hunter tiene una buena cantidad de luchadores para elegir

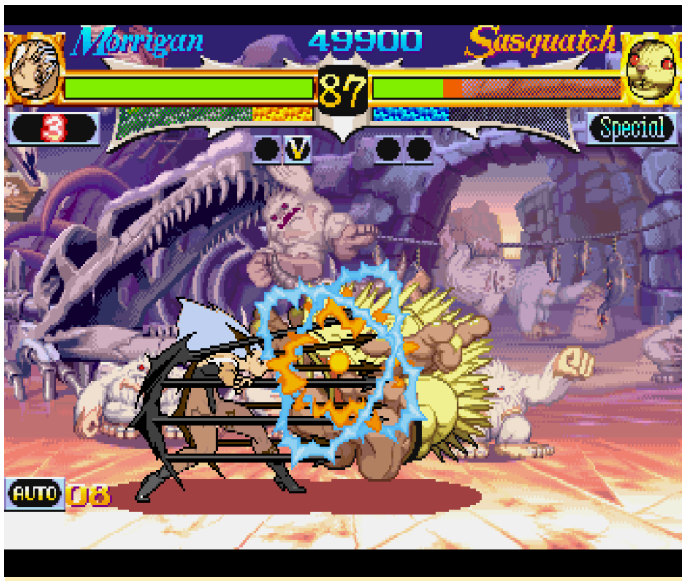


Figura 22 – Vampire Hunter tiene gráficos muy buenos y una acción de lucha bastante rápida

Figura 23 – Restaurando tu salud con pentagramas

Figura 24 – Los diferentes hechizos cuestan diferentes cantidades de MP por ataque

Tengo que decir que disfruto mucho de los colores brillantes de la Sega Saturn. Comparada con la Génesis, donde todo parece más oscuro y turbio, los juegos de Saturn son mucho más vibrantes en lo que respecta a los colores y yo especialmente disfruto mucho con eso. Lo mismo ocurre con este juego de lucha que, junto con las numerosas animaciones para cada personaje y la gran variedad de ataques hacen que la experiencia de juego sea realmente muy buena. Al igual que ocurre con la mayoría de los buenos juegos de lucha, éste usa 6 botones que, además, es mejor para la Saturn ya que otras consolas solo usan 4 botones de acción la mayor parte del tiempo. Esto significa también que puedes realizar una gran cantidad de movimientos diferentes y ataques especiales con bastante facilidad. Posiblemente sea uno de mis juegos de lucha favoritos (quizás también porque puedo vencer a los enemigos). Si te gustan los juegos de lucha, te recomiendo que pruebes este en ODROID.

Waku Waku Puyo Puyo Dungeon

Desafortunadamente, no he tenido tiempo para jugar tanto como me gustaría, también es un poco difícil de entender, ya que todo el juego está en japonés. Aun así, logré entrar en una mazmorra y descubrí cómo atacar, lanzar hechizos y cambiar los diferentes hechizos que tenía, y por lo que a mí respecta me divertí bastante.

Aunque el juego parece un poco lento, realmente no importa ya que no necesitas un tiempo de reacción rápido. Tú y los enemigos cogen “rondas”, incluso si no tiene ganas. Caminas un paso cuando lo hace el enemigo, golpeas al enemigo o te ataca igualmente. Simplemente es de ida y vuelta. Tu personaje se vuelve automáticamente hacia el enemigo cuando te atacan para que no tengas que averiguar cómo hacer las diagonales. Encuentras oro y otros objetos en la mazmorra, así que supongo que puedes comprar nuevas armas y armaduras o simplemente artículos de salud fuera de la mazmorra, pero todavía no he descubierto como se hace. También encontré dos tipos de pentagramas en las mazmorras. Uno restaura tu salud, y el otro restaura MP. El de MP

desaparece rápidamente tras usarse una o dos veces, mientras que el HP parece mantenerse.

Desde el principio cuentas con ataques de fuego y hielo como habilidades y en un nivel más profundo de la mazmorra encuentras enemigos que reciben más daño dependiendo del elemento que estés usando. Matar a un enemigo te da Exp y Gold de vez en cuando, también dejan caer objetos como manzanas y cosas por el estilo. Los primeros dos niveles subirán rápido y tu HP y MP aumentará automáticamente, además tu ataque y la magia se volverán más fuertes. Encontrarás una escalera en algún punto de la mazmorra que conduce al siguiente nivel, profundiza lo suficiente y te encontrarás con un jefe.

Figura 25 – En el siguiente nivel de la mazmorra

Warcraft II – The Dark Saga

Me sorprendió bastante ver este juego en Saturn, pero una vez más, también he visto otras versiones para PC del sistema. Me asombró ver que el juego funcionaba bastante bien, aunque descubrí que tiene algunos problemas. De modo que lo malo primero. En Yabause independiente, los videos están dañados y no puedes verlos (pero si escucharlos). También tiene algunos problemas con los gráficos transparentes, ya que el juego pierde algunos elementos. Por ejemplo, cuando haces clic en un personaje, no ves que lo tienes seleccionado y si abres la pantalla de información, el fondo es blanco en lugar de transparente/mallado. Esto resulta bastante molesto, ya que ves cuántos personajes has seleccionado o qué salud les queda, pero todo lo demás funciona bien una vez que descubres el diseño del juego.

Cuando descubriste el esquema, el juego en realidad se puede disfrutar bastante en Yabause y es muy divertido, incluso con las deficiencias que hemos comentado. El núcleo libretro no tiene problemas gráficos y se ve bastante bien, los vídeos funcionan, las transparencias y las mallas parecen funcionar bien, PERO la velocidad es demasiado lenta para poder disfrutarlo.

Figura 26 – Esquema general de la misión con contraseña de nivel y una descripción de la misión con sonido

Figura 27 – Un rápido vistazo a los gráficos del juego

Todo, desde la versión DOS del juego, parece estar en su sitio. El juego incluso incluye un paquete de expansión. Todas las descripciones de la Misión tienen sonido. Los videos están todos (incluso si no

funcionan en Yabause) y la música es la que cabría esperar.

Aunque el juego usa contraseñas para los niveles, también hay una opción para guardar el juego. Aunque probablemente no te cojan más de 2 juegos guardados en la memoria interna de Saturn, ya que las partidas guardadas ocupan bastante. Aun así, esto te permite guardar y cargar en cualquier punto del juego. No obstante, también puedes jugar con facilidad a la versión DOS del juego en ODROID, así que depende de ti elegir desde que plataforma quieres jugar a esta obra maestra.

Willy Wombat

Este juego me llego a sorprender un poco. En este juego de plataformas 3D, juegas como Willy Wombat y saltas y peleas a través de los diferentes niveles. Lo que resulta muy interesante es que puede girar libremente la cámara con los botones R y L del mando de Saturn (para mí asignados como L2 / R2). Realmente es muy interesante ya que muchos objetos solo aparecen si giras la cámara, están colocados estratégicamente para que solo los vea desde un cierto ángulo. Aunque el potencial 3D de Satur es muy limitado, funciona bastante bien en este juego, ya que usa sprites 2D renderizados previamente (similares a Donkey Kong) para el personaje y los objetos en un entorno 3D.

Figura 28 – Gráficos 3D con mínimas texturas en Saturn

Figura 29 – Recoge cinco de estas burbujas para aumentar tu nivel de vida

Todavía no he profundizado demasiado en el juego, pero realmente funciona bastante bien. Los controles funcionan bien, aunque a veces es un poco difícil ubicar el salto correctamente debido al ángulo de visión isométrico, pero el hecho de poder girar la cámara a menudo ayuda bastante. Tienes dos tipos de ataques: una barra de corto alcance y un ataque bumerán que tiene un alcance limitado. También se puede usar para recoger artículos. Más adelante también puedes encontrar algunos ataques especiales llamados fuerzas que puedes darte apoyo en tu misión. Estos hacen un daño masivo a muchos enemigos de la pantalla.

Figura 30 – Atacando al enemigo con tu fiel bumerán

Aunque el juego está completamente en japonés, todas las interpretaciones de voz se realizan en inglés,

lo cual facilita el seguimiento de la historia y la comprensión del juego. Además, el juego es exclusivo de Sega Saturn, así que puedes comprobarlo, no lo encontrarás en ningún otro sitio.

Wonder 3 Arcade Gears / Three Wonders

Wonder 3 Arcade Gears es una colección de 3 juegos arcade en un único disco. Gallos, Chariot y Donburu son los tres juegos de este disco. Roosters es un juego de plataformas orientado a la acción, Chariot, un juego de disparos arcade de desplazamiento lateral con los mismos personajes que en Roosters y Donburu es un juego de rompecabezas sin relación. Todos los juegos funcionan bastante bien, sin ralentizaciones ni nada.

Figura 31 – La pantalla de selección de juegos de Wonder 3 Arcade Gear

Cada juego es divertido a su manera, es una muy buena compilación de juegos en un único disco. Conocía Roosters antes de conseguir esta compilación al que he jugado en las maquinas arcade o en la PS1. Sin embargo, los otros dos juegos eran nuevos para mí. Chariot me recordó al instante a Roosters, tiene los mismos personajes y enemigos, y los cofres del tesoro son los mismos que en Roosters, aunque ahora todos vuelan.

Tienes tu ataque principal, que se puede actualizar y cambiar recopilando potenciadores, y tienes tu ataque secundario más poderoso que depende de la longitud de tu “cola”. Cuanto más larga es la cola, más fuerte es el ataque, pero también lleva más tiempo recargar.

Figura 32 – Plataformas de acción Roosters

Figura 33 – Shooter arcade Chariot

Figura 34 – Puzzle de "acción" Donburu

Donburu es un juego totalmente diferente, y no guarda relación con los otros dos anteriores. Tu objetivo aquí es empujar bloques y con ello, matar a todos los enemigos de la pantalla. Los diferentes bloques tienen diferentes propiedades. Por ejemplo, algunos explotan y dañan a los enemigos que están cerca. Obtienes puntos dependiendo de lo rápido que hayas completado el nivel. Los tres juegos son muy divertidos, tienen buenos gráficos y se ejecutan con fluidez en el ODROID-XU4. Vale la pena tener este disco en tu Saturn o tu emulador Saturn favorito.

Worms

Recuerdo el original Worms de mi fiel Amiga. Este juego fue el mejor de los mejores por aquel entonces, y dio lugar a muchas secuelas y clones como Hedgewars. La versión de Saturn de este juego parece ser similar a la versión de Amiga CD32, lo cual significa que viene con toda la música, voces y escenas de video que lo hacen tan divertido y autentico. Y aunque este juego tiene un montón de cosas ejecutándose en segundo planto, con todos los personajes en pantalla funciona perfectamente en el ODROID XU4.

Figura 35 – Pantalla de títulos de Worms en la Sega Saturn

Figura 36 – Como de costumbre, el ordenador deja caer la bomba … umm granada sobre tu cabeza.

En este juego, tienes equipos de gusanos que luchan entre sí con diferentes armas que van desde Bazooka, Shotgut y Air Strikes hasta las famosas ovejas que explotan. Hay toneladas de armas para elegir. ¡Es brutal, es guerra, es divertido! Las armas se ven afectadas por la dirección y la fuerza del viento. Con la suficiente fuerza de viento puedes disparar hacia la derecha para esquivar un objeto y aún así matar a un enemigo que se encuentra a tu izquierda.

Figura 37 – Caída de suministros de el cielo, cógelos antes de que lo haga alguien

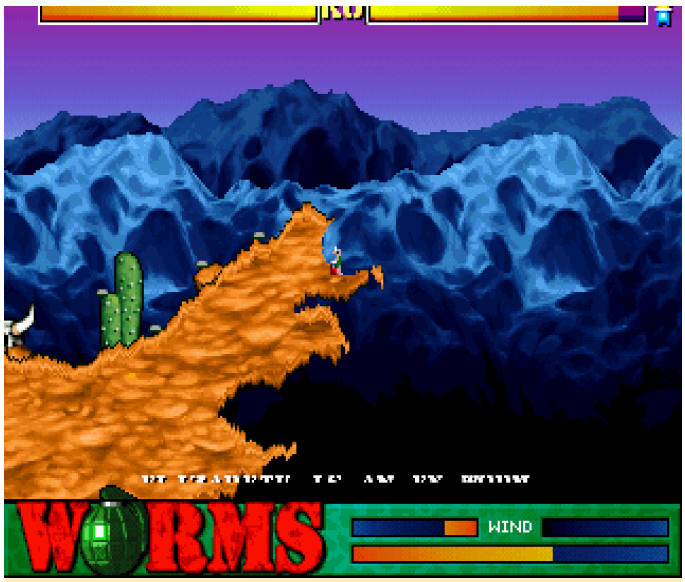


Figura 38 – Si tu salud llega a cero tu gusano explotará solo

Este juego tiene unos gráficos muy buenos, especialmente teniendo en cuenta que fue creado para Amiga. Es un juego muy competitivo y divertido para jugar con amigos o contra el ordenador. Este juego en particular es el que inició la serie, que continúa hoy día con muchas y muy diferentes versiones del juego. Si has jugado a Worms Armageddon o World Party, ya sabes lo divertido que es este juego, aquí tienes la oportunidad de visitar las raíces de estos increíbles juegos.

Menciones honoríficas

Terradiver

Este es otro shoot 'em up vertical bastante genérico. Tiene buenos efectos visuales y en realidad usa diferentes planos donde los enemigos están activos, pero no todas tus aeronaves pueden atacar todos los planos, lo cual puede llegar a ser bastante molesto. Aun así, no he encontrado nada especial y es un poco lento para mi gusto a la hora de poder disfrutarlo completamente, pero vale la pena si te gustan los shmups.

Tetris Plus

Qué puedo decir: es el Tetris y no el único, aunque en mi opinión es la mejor versión. No hay nada malo en el juego. Tiene diferentes modos de juego, algunos personajes de anime, buena música y un sistema de niveles. Es un juego de Tetris muy conseguido y agradable.

The Lost World Jurassic Park

Se colgó cuando inicie el juego en Yabause, pero funciona con libretro, pero lamentablemente un poco lento. En esta ocasión juegas con un dino y luchas contra otros dinosaurios y tienes que resolver puzles de saltos y cosas por el estilo. Utiliza un mundo y personajes en 3D y la verdad que su aspecto es bastante sorprendente. Desearía que funcionase con el emulador independiente Yabause ya que probablemente se ejecutaría a toda velocidad.

Three Dirty Dwarves

Desafortunadamente, este juego presenta muchos problemas gráficos al intentar ejectuarlo en mi versión actual del emulador independiente Yabause, pero funciona sin problemas gráficos en el nucleo libretro, pero como de costumbre, se ejecuta demasiado lento. Este juego en realidad es bastante bueno: juegas con 3 enanos que luchan contra todo tipo de monstruos. Uno usa una pelota de béisbol y el bate como arma principal, otro usa una escopeta, y el último usa una bola de boliche para luchar contra los enemigos. Si te golpean, el tipo cae y el siguiente en la línea se hace cargo, aunque puedes revivir a tus compañeros caídos pateándoles el trasero, por así decirlo. Es un divertido juego de acción con buenos gráficos. Espero poder ejecutarlo con una versión más reciente de Yabause en algún momento, y con

ello solucionar el problema de la lentitud, ya que éste es un juego realmente bueno.

Thunder Force Gold Pack 1, 2 and V

Thunder Force Gold Pack 1 se compone de Thunder Force II MD y Thunder Force III. El primero es de arriba hacia abajo, y el segundo es de desplazamiento lateral. Ambos son agradables y divertidos con un poco de desplazamiento paralaje. Los gráficos están bien, pero no son nada del otro mundo. Puedes cambiar entre diferentes tipos de armas para optimizar el efecto sobre tu enemigo. Especialmente en Thunder Force III se hace esto bastante.

Thunder Force Gold Pack 2 contiene Thunder Force AC y Thunder Force IV. El primero es más o menos lo mismo que Thunder Force III, pero con mucho más retardo. Supongo que intensificaron los gráficos con una capa de fondo adicional, aunque el rendimiento es bastante notable. Sin embargo, es más fácil que Force III, a pesar de que eliminaron la opción de selección de nivel.

Thunder Force IV tiene mejores gráficos de desplazamiento paralaje y es mucho mejor en cuanto a rendimiento, e incluye de nuevo la selección de nivel. El mapa es mucho más grande en esta ocasión, y puedes subir y bajar bastante en la pantalla, lo cual significa que no ves lo que hay encima o debajo de ti si no subes y bajas. Me gusta más, aunque es bastante difícil.

Thunder Force V esta en 3D. Se ve bien, pero vuelve haber retardo y los gráficos tienen bastentes problemas. Nunca sé si golpeo o no al jefe, ya que no veo ningún indicador de ello, y el jefe presenta fallos por todos lados. Si Thunder Force IV fuera un título independiente, lo recomendaría, pero junto con los otros, no vale la pena.

Ultimate Mortal Kombat 3

Me gusta este juego: es muy rápido y tienes muchos personajes para elegir. No es muy fácil, pero no creo que ninguno de los juegos de Mortal Kombat lo sean. Es una versión arcade realmente buena y se ejectua bastante bien en el ODROID-XU4. Si te gusta la serie Mortal Kombat, deberías probar esta versión y ver si es de tu agrado.

Virtua Fighter 2

Funciona y los gráficos en 3D son bastante buenos. Las texturas y los fondos están bien. Simplemente no soy fan de este juego. La velocidad está bien si usas frame skipping.

Wakumon / Waku Waku Monster

Este juego es realmente divertido para pasar un rato. Es uno de essos juegos en el que sueltas elementos y tienes que hacer coincidir el mismo color, si juntas 3 o más, estallan y obtienes puntos. Consigue suficientes puntos y podras atacar a tu enemigo, si el enemigo ataca, presionas un botón para contrarrestar el ataque y minimizar el daño. Sé más rápido que tu oponente y gana. Cada vez que ganas, tu “Monstruo” evoluciona. Empiezas con un huevo y cada vez que ganas, tu mascota crece y cambia de forma, lo cual es muy divertido. Es genial en pequeñas dosis y realmente me gusta bastante.

Whizz

Whizz es un interesante juego de plataformas donde juegas con un conejo que tiene un sombrero que parece directamente sacado de Alicia en el país de las maravillas. Los colores son suaves, la música es muy agradable y los controles del juego son bastante buenos. Es lo suficientemente rápido en el XU4 como para disfrutarlo. Solo hay un problema: el juego se bloquea tras un par de segundos.

Usar el núcleo libretro ayuda a que el juego no se congele, pero es demasiado lento para jugar. Aunque actualmente el juego no funciona como debería, me pregunto si las nuevas versiones de Yabause solucionarán este problema, y de ser así, es muy posible que podamos jugar a este juego en el futuro.

Wipeout 2097

Actualmente, este juego solo funciona con el nuleo libretro, aunque es demasiado lento para que resulte divertido. Sin embargo, te permite apreciar cuál es su potencial, creo que probablemente disfrutaría más que con la versión de PS1. Desafortunadamente, no es jugable en este momento.

WolfFang

Este juego es otro de desplazamiento lateral, que no está nada mal, pero tampoco es nada del otro mundo. Puedes construir tu propio “mech” seleccionando armas secundarias, armas de corto alcance, tipos de piernas y demás, o puedes elegir una de las opciones preconfiguradas. Los controles no son los mejores en mi opinión, si mantienes presionado el botón de disparo, seguirás mirando en una única dirección, y tendrás que apuntar hacia arriba y hacia abajo, si no mantienes el botón puedes darte la vuelta y matar los enemigos que hay detrás de ti. Aunque tu arma está muy extensiva, no existe un ángulo desde el que puedas golpear a todos los enemigos, incluso cambiando las dos alturas que puedes seleccionar de atrás para adelante no te permitirá golpear a todos los enemigos. Debes saltar con la esperanza de golpear al resto de enemigos antes de que te golpeen a ti.

Cuando tu salud llega a cero, tu mech se destruye y saltas de él. El personaje resultante solo tiene una pequeña pistola y el más mínimo golpe te matará, lo cual no es nada divertido en absoluto. En el independiente Yabause, aparecen algunos problemas gráficos, y libretro presenta algunos problemas de velocidad, pero probablemente sea lo suficientemente rápido como para jugar.

Z

Z fue una sorpresa para mí. Adoro Z, y esta versión parece tener todo lo que tiene la versión para PC, incluidos todos los videos, música y niveles, y eso es muy positivo. Los gráficos también están bien, pero este juego requiere velocidad y mucha. Si no sabes qué y cuándo hacer las cosas, perderás. Este juego es muy difícil y para serte sincero, sin un ratón, jugar a este juego no es nada divertido. TCuentas con un par de atajos para saltar a la próxima unidad, a la siguiente bandera y al siguiente enemigo, pero esto no ayuda demasido. Aún así, el juego funciona perfectamente.

Resumen final de Sega Saturn en ODROID

Ha sido un largo viaje. He probado cientos de juegos de Saturn y he seleccionado los juegos que más me gustaron. No todos los juegos han llegado a funcionar: algunos eran insoportablemente lentos y otros se bloqueaban antes de que pudiera probarlos. Ha sido una montaña rusa de emociones. Hay algunos juegos que me hubiera gustado mucho jugar en el ODROID, pero simplemente no funcionan. La emulación de Sega Saturn es muy difícil, y a menudo falla. El desarrollo también es muy lento, y ha cambiado muy poco en mucho tiempo. Todavía estoy usando una versión muy antigua del emulador independiente Yabause, aunque las versiones más recientes simplemente no llegaron a funcionar.

Sin embargo, ¡hay esperanza! Las nuevas versiones de los emuladores de Saturn todavía están en desarrollo, e incluso hay una versión de Android que se ejecuta en OpenGL ES. De hecho, he visto vídeos que muestran como este emulador también llega a funcionar en el ODROID-XU4.

Espero que, en el futuro, podamos ver más juegos de Saturn ejecutandose en ODROID, y podría retomar el tema una vez más cuando esto suceda. Por ahora estoy bastante satisfecho con Saturn en el ODROID-XU4. Tengo una colección de más de 50 juegos para Saturn con lo que realmente disfruto jugando, y que me llevará bastante tiempo terminar. Espero que tú también te diviertas con la Sega Saturn en tu ODROID-XU4 y que esta serie te haya dado algunas ideas sobre a lo que poder jugar.

Transcodificar el Receptor DVB Enigma2 Usando ffmpeg en el ODROID-XU4

August 1, 2018 By @martos ODROID-HC1, ODROID-XU4, Tutoriales

Cuando me alojo en hoteles durante mis viajes, observo que algunos canales no están disponibles en la televisión. Usando NAT (asignando la IP externa a la IP interna del dispositivo) y http flux, puedo ver algunos de esos canales de TV en mi teléfono móvil o en mi ordenador portátil. Si el ancho de banda es bajo, puedes usar un acceso 3G o superior (350 Kb de velocidad).

En mis primeras pruebas, con una Raspberry Pi 3, usando el decodificador de hardware, sólo podemos transcodificar a 320*240. Descubrí el ODROID-MC1 en un sitio web de reventa, así que me hice con uno. Éste utiliza el Exynos 5520, que al ser más potente, podemos llegar a transcodificar a 512*384.

Comparaciones de precios

- Raspberry Pi 3 Modelo B + Desktop Starter Kit (16Gb) cuesta 60 € (70$)
- ODROID-HC1 (SD 16 Gb + PSU) = 70€ (82$)
- ODROID-XU4 (sd 16 Gb + PSU) = 85€ (100$)

Los SBCs ODROID podrían ser una buena opción, aunque el decodificador de hardware puede tener algunos errores en el procesamiento de MPEG4, así que, usamos solo el formato de video H.264 de momento. La decodificación de video MPEG4/MPEG2 es muy inestable, de modo que estamos obligados a utilizar el codificador/decodificador por software.

La Raspberry Pi 3 Modelo B+ está bien, pero la CPU es menos potente. No obstante, con el decodificador/codificador de hardware, su rendimiento es aceptable. En cualquier caso, no use la red WiFi, sino la red cableada (conector RJ45 ethernet). Con cualquiera de las dos placas, no podemos decodificar video 4K, el cual solo está disponible en ODROID-C1.

Instalación en ODROID-XU4 o en ODROID-HC1

Utilizaremos Ubuntu 18.04 (Versión: 20180531 – https://goo.gl/LKPL9F). Instala el software de https://goo.gl/A9gbkD. Lee las Notas de la versión en ambos enlaces que tienen información muy útil. Después de esto, instalamos el software e2transcoder (la GUI web) y ejecutamos los pasos de https://goo.gl/p9c4Pi. Los pasos incluyen:

```
$ su
# apt-get install mali-fbdev
# apt-get install ffmpeg
# apt-get install apache2
# apt-get install libav-tools
# apt-get install zip
# apt-get install mc
# apt-get install zip
# apt-get install php
# apt-get install libapache2-mod-php
# apt-get install sqlite
# apt-get install php-sqlite3
# apt-get install php-xml
```

Coge el archivo zip de e2transcoder usando:

```
$ wget http://e2transcoder.sharetext.net/wp-content/uploads/files/enigma2_transcoder_072.zip
```

Expande el archivo zip y copia el contenido de

```
repertories/DB/*.*
```

en

```
/var/www/html/
```

Ahora deberías ver los siguientes archivos y directorios dentro de la carpeta /var/www/html/:

```
/admin
/stream
index.html (original )
Index.php
```

Ejecuta los siguientes comandos:

```
$ rm /var/www/html/admin/config.php
$ cp /var/www/html/admin/config_linux.php /var/www/html/admin/config.php
```

Edita el archivo /var/www/html/admin/config.php para que contenga la siguiente información:

```
// if enigma2 receiver is not used must be 0, but it is not mandatory if receiver is used
$conf["callreceiver"] = 1;
// full path of stream dir
$conf["stream_dir"]="/var/www/html/stream/";
// path of avconv or ffmpeg executable, if avconv of ffmpeg installed $conf["command"] = "/usr/bin/ffmpeg";
from package only need executable name
// web url folder of stream enigma2 receiver configuration
$conf["stream_web_dir"] = "/stream/";
// enigma2 user name
$conf["db_username"] = "root";
// enigma2 password
$conf["db_password"] = "YYYYYY";
// enigma2 IP
$conf["db_ip"] ="192.168.ZZZ.ZZZ";
$conf["parameters"] = "-threads 16 -vcodec h264 -i {stream_url} -s 512x384 -vf fps=21 -maxrate:v 400k -bufsize:v 60000k -ac 1 -ar 22050 -vbr 1 -sn {stream_dir}ystream.m3u8";
// full path of avconv or ffmpeg log
$conf["stream_log"] = "/var/log/stream.log";
```

Las carpetas /admin/db y /stream/ deberían ser editables por el servidor web. En el caso de Apache, usamos el usuario “apache”, para nginx www-data.

```
$ chown -R www-data /var/www/html/
$ chmod -R 755 /var/www/html/
$ touch /var/log/stream.log
$ chown www-data /var/log/stream.log
```

Existe un pequeño problema con slqlite. Localiza la libreria sqlite3.so. En Raspberry Pi, está en /usr/lib/php/20151012/sqlite3.so. Busca php.ini normalmente en /etc/php/7.2/apache2/php.ini. En este archivo, encuentra la sección [sqlite3] y cámbiala por:

```
[sqlite3]
sqlite3.extension_dir =/usr/lib/php/20170718/sqlite3.so
```

Ahora, en tu navegador web, dirígete a http://ip_de_tu_receptor/index.php. Introduce la información de inicio de sesión (también en config.php). Ve a la sección “Settings” y haz clic en “Reload E2 Playlist”. Regresa a “Channels”. Selecciona tu canal haciendo clic en él, y en la parte superior

debería ver “TV: nombre de canal”. Ten cuidado, no puedes mostrar TV y transcodificar otro canal si no está en el mismo transpondedor. Debería ver el cambio de estado de “Preparing” a “Running” (también puede verificarlo usando: $ tail -f /var /log/stream.log). Dirígete a la sección “Live” y haz clic en el icono de reproducción o introduce el enlace en el reproductor vlc.

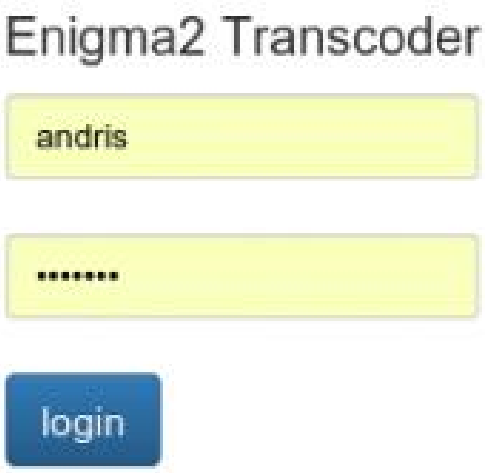


Figura 01 – Inicio de sesión

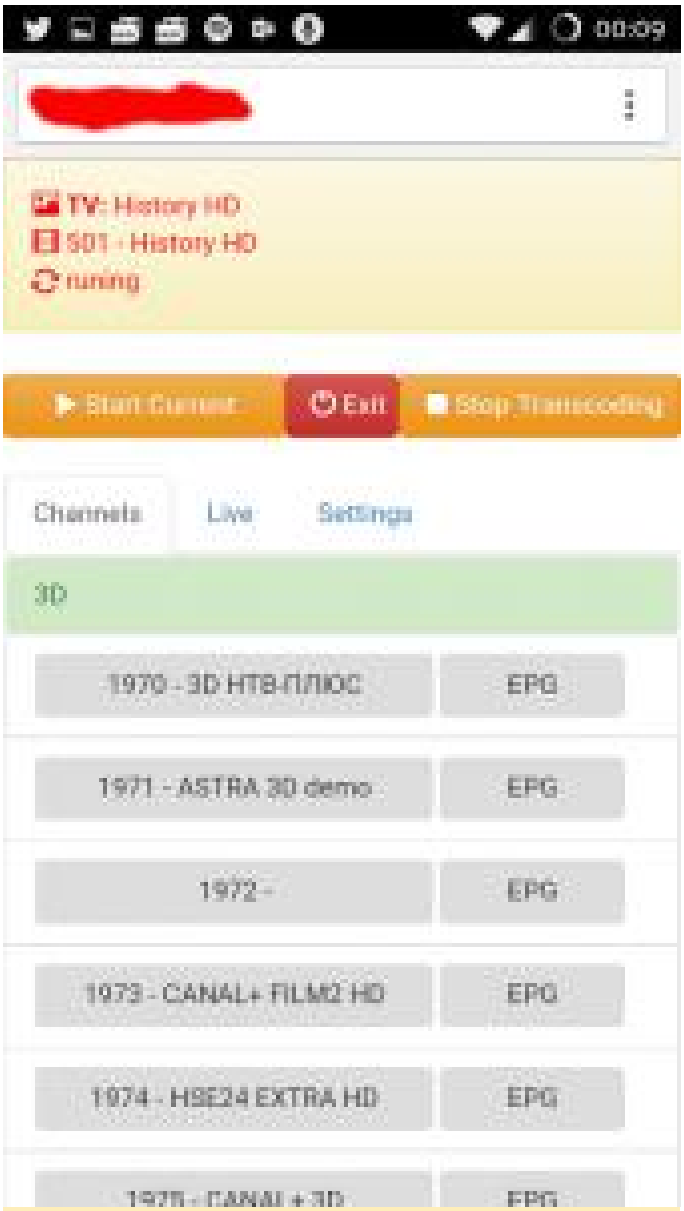


Figura 02 – Página principal

Figura 03 – Reproductor HTML5 en Chrome en un teléfono

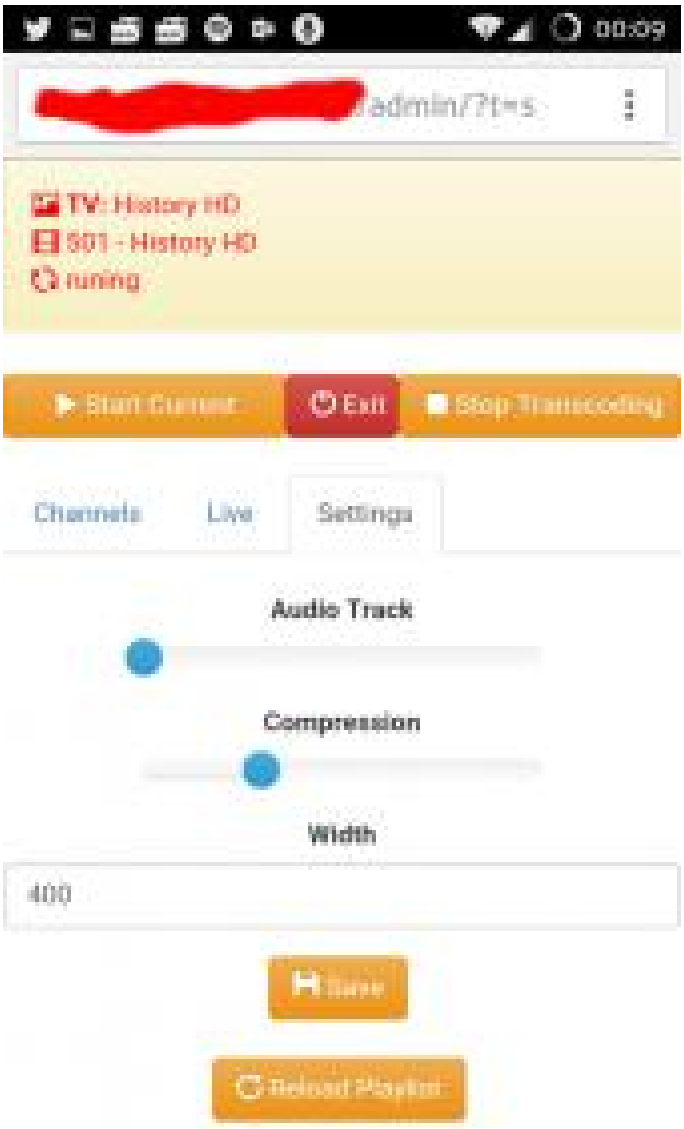


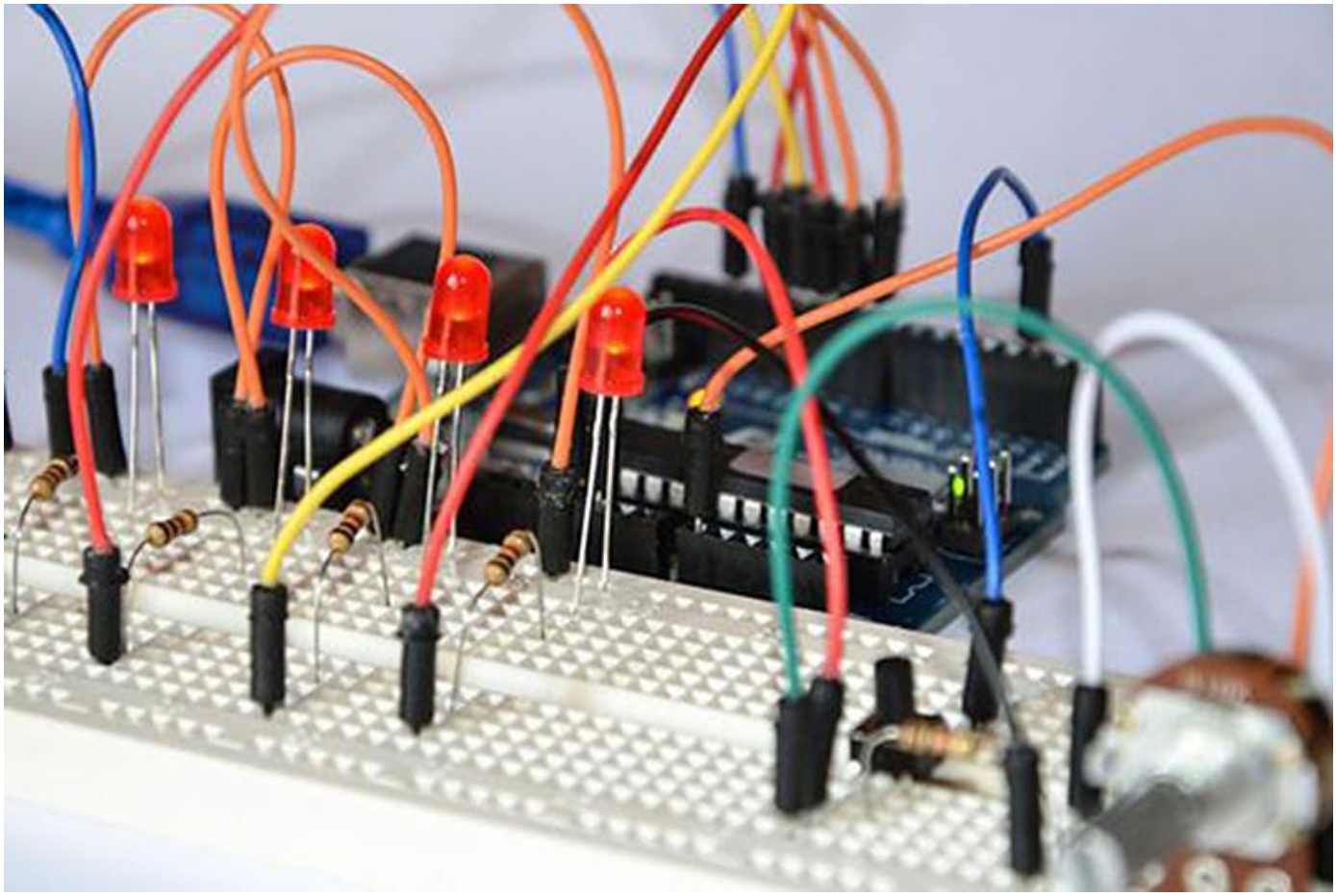
Figura 04 – Configuraciones

Figura 05 – EPG

Para comentarios, preguntas y sugerencias, visita el artículo original en **https://forum.odroid.com/viewtopic.php?f=95&t=31358**.

Campamento de Programación Parte 1: Primeros Pasos con Arduino

August 1, 2018 By Justin Lee Mecaniqueo, Tutoriales, ODROID-GO



En este artículo, aprenderás cómo descargar e instalar Arduino IDE y ejemplos y librerías específicas de ODROID-GO. Existen guías oficiales paso a paso para plataformas compatibles que están mantenidas por miembros de la comunidad.

- Windows
- Debian / Ubuntu
- Fedora
- openSUSE
- macOS

Instalar librerías ODROID-GO

Windows Ejecuta un programa Git Bash desde el Menú de Inicio e introduce los siguientes comandos:

```
$ git clone
https://github.com/hardkernel/ODROID-GO.git
$
```

```
USERPROFILE/Documents/Arduino/libraries/ODROID
-GO
```

Linux Abre un Terminal presionando CTRL-ALT-T e introduce los siguientes comandos:

```
$ git clone
https://github.com/hardkernel/ODROID-GO.git
~/Arduino/libraries/ODROID-GO
```

Seleccionar un dispositivo destino Arduino IDE tiene que saber qué placa se usará para compilar y enviar datos.

Select Tools → Board → ODROID-ESP32.

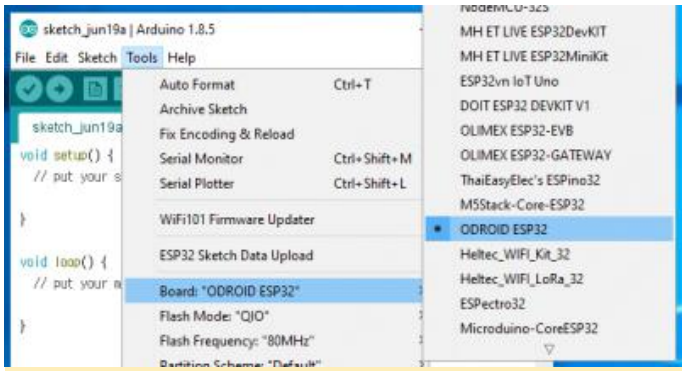


Figura 1 – Seleccionando un dispositivo destino

Seleccionar el puerto serie adecuado Arduino IDE tiene que saber a qué puerto está conectado el dispositivo. El número del puerto depende de tu sistema. Es posible que necesite instalar los drivers CP2104 VCP en tu ordenador si no logras abrir el puerto serie.

Windows

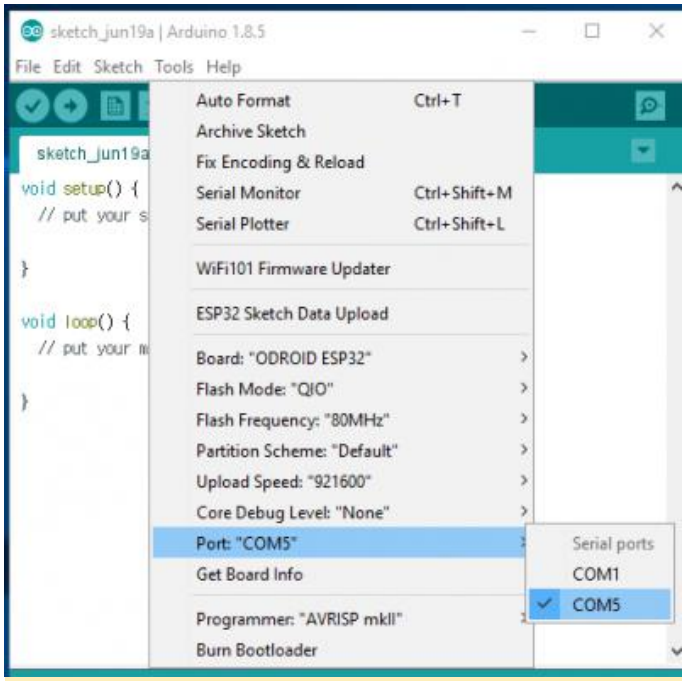


Figura 2 – Seleccionando un puerto serie adecuado en Windows

Select Tools → Port: "COM#" → COM#.

Linux

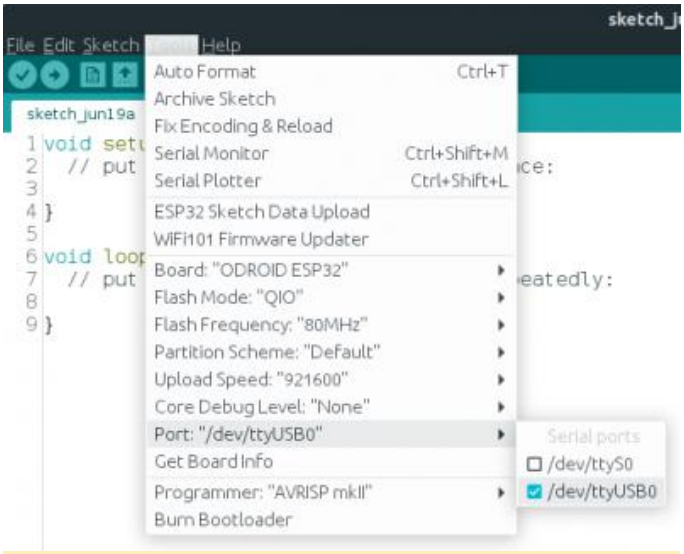


Figura 3 – Seleccionando un puerto serie adecuado en Linux

Puesto que ODROID-GO siempre se conecta al PC a través de un cable USB, selecciona un archivo de dispositivo USB.

Select Tools → Port → /dev/ttyUSB#.

Vamos a codificar con ODROID-GO Ahora estás listo para escribir tu código fuente. Para aprender a escribir el código fuente, consulte la Parte 2 de la serie en este número de ODROID Magazine. Para comentarios, preguntas y sugerencias, visite el artículo original en https://wiki.odroid.com/odroid_go/arduino/01_arduino_setup.

Thundroid – Parte 2 Migración desde Bitcoin Testnet a Mainnet

August 1, 2018 By @stadicus Linux, Tutoriales

¿Recuerdas la Parte 1 de esta guía? Configuramos un completo nodo de Bitcoin con Lightning desde cero, llegamos bastante lejos para asegurar nuestro sistema y empezamos a probar Bitcoin en testnet. Si no te has hecho con la Parte 1, léala primero ya que esta parte no tiene mucho sentido sin ella. El objetivo de esta guía es cambiar nuestro Thundroid de Bitcoin testnet a mainnet y realizar transacciones con dinero real.

Buenas prácticas financieras El Bitcoin es un activo al portador como dinero en efectivo físico, de modo que las transacciones no se pueden revertir. Controlar tu bitcoin en realidad significa controlar las claves privadas que te permiten usarlo. Esto significa que, si alguien tiene acceso a tus claves privadas, esta persona tiene control total sobre tu bitcoins. Una vez que se envían a una dirección diferente, no hay nada que puedas hacer para recuperarlos.

Para administrar tu bitcoin, necesitas una "Wallet" o billetera. Esta es una aplicación que administra las claves privadas. Tenemos que hacer una distinción muy importante:

- Hot wallet: una aplicación que administra su clave privada y está expuesta a Internet. Es una aplicación móvil muy cómoda, pero podría ser potencialmente pirateada. Esta billetera suele usarse para cantidades pequeñas y para un uso diario.
- Cold storage: tu clave privada nunca ha estado expuesta a ninguna red. Algunos ejemplos son las billeteras en papel que se crean y/o imprimen utilizando un ordenador sin conexión, o billeteras de hardware como Ledger o Trezor. Así es como aseguras tus ahorros en bitcoin.

Por definición, este proyecto es un hot wallet ya que está conectado a internet. Dicho esto, ¡No almacene grandes cantidades de dinero en su Thundroid!

- Bitcoin: no use la billetera integrada en Bitcoin Core para nada. El camino a seguir es utilizar una pequeña billetera de hardware para proteger tus claves privadas con Thundroid como tu back-end de confianza para enviar/verificar las transacciones. Más sobre este tema más adelante.
- Lightning: como toda la red aún está en fase beta, es evidente que no debes poner los ahorros de tu vida en ella. Experimentar con pequeñas cantidades está bien, pero hazlo bajo tu propia responsabilidad.

Tenga en cuenta que, aunque Bitcoin ha sido puesto a prueba durante casi una década y se usa para mover miles de millones de dólares estadounidenses todos los días, Lightning Network aún está en fase beta y está bajo desarrollado. Esta guía también te permite configurar tu nodo Bitcoin ignorando la parte de Lightning.

Cambiar a Mainnet La configuración actual de tu Thundroid se ejecuta en Bitcoin testnet. Asegúrate de que tu sistema funciona sin problemas para que podamos pasar a copiar la cadena de bloques de mainnet que ya descargaste con un ordenador corriente (ver Parte 1).

En tu ordenador habitual, comprueba el progreso de la verificación en Bitcoin Core. Para continuar, debe estar completamente sincronizado (ver barra de estado). Cierra Bitcoin Core en Windows para que podamos copiar toda la estructura de datos a Thundroid. Esto nos llevará alrededor de 6 horas.

NOTA: Si te quedas atascado, echa un vistazo a mi repositorio GitHub. Puedes buscar respuestas entre las cuestiones ya resueltos o abrir un nuevo tema si lo consideras necesario.

Activar el acceso por contraseña de forma temporal Para copiar los datos con el usuario "bitcoin", necesitamos habilitar temporalmente el inicio de sesión con contraseña. Como usuario "admin", edita el archivo de configuración de SSH y coloca un # delante de "PasswordAuthentication no" para deshabilitar toda la línea. Guarda y Salte.

```
$ sudo nano /etc/ssh/sshd_config
# PasswordAuthentication no
```

Reinicia el demonio SSH.

```
$ sudo systemctl restart ssh
```

Copiar la cadena de bloques de mainnet usando SCP Estamos utilizando "Secure Copy" (SCP), así que, descarga e instala WinSCP, un programa gratuito de código abierto. Existen otros programas de SCP disponibles para Mac o Linux que funcionan de manera similar. No uses rsync ya que este te puede ocasionar problemas más adelante.

Con WinSCP, ahora puedes conectarte a tu Pi con el usuario "bitcoin". Ambos protocolos SCP y SFTP funcionan, bajo mi experiencia SCP es un poco más rápido.

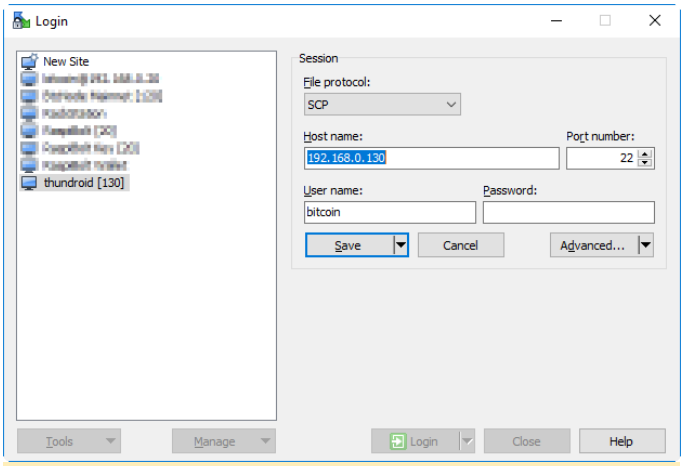


Figura 1 – Iniciar sesión con el nombre de usuario "Bitcoin"

Acepta el certificado del servidor y navega hasta los directorios locales y remotos de bitcoin:

- Local: d:itcoinitcoin_mainnet\
- Remoto: mnthdditcoin\

Ahora puedes copiar los dos subdirectorios blocks y chainstate de Local a Remoto. Esto nos llevará unas 6 horas.

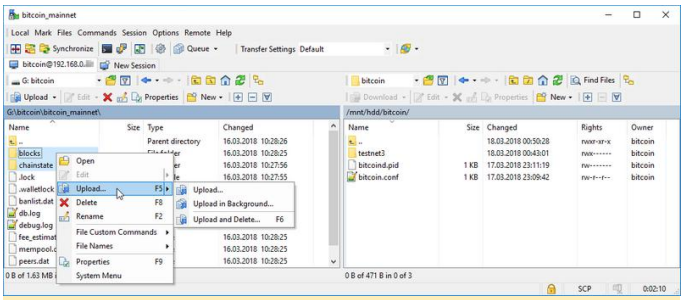


Figura 2: Copiando los dos subdirectorios blocks y chainstate de Local a Remoto

NOTA: La transferencia no debe interrumpirse. Asegúrate de que tu ordenador no entre en modo reposo.

Desactivar de nuevo inicio de sesión por contraseña Como usuario "admin", elimina el signo # que hay delante de "PasswordAuthentication no" para habilitar la línea. Guarda, salte del archivo de configuración y reinicia el daemon ssh.

```
$ sudo nano /etc/ssh/sshd_config
PasswordAuthentication no

# Restart the SSH daemon.
$ sudo systemctl restart ssh
```

Enviar de vuelta tu Bitcoin testnet Para evitar que se consuma nuestro Bitcoin testNet, y a modo de cortesía para los siguientes analistas, cerramos todos nuestros canales y retiramos los fondos de la dirección indicada en el sitio web de Bitcoin Testnet Faucet.

```
$ lncli closeallchannels
```

Espere hasta que el saldo del canal sea cero y los fondos hayan sido devueltos a nuestra billetera en cadena.

```
$ lncli channelbalance
$ lncli walletbalance
```

Envía la cantidad contemplada por walletbalance de menos 500 satoshis para justificar las tarifas. Si recibes un error de "insufficient funds", reduce un poco más hasta que la transacción se emita.

```
$ lncli sendcoins
2N8hwP1WmJrFF5QWABn38y63uYLhnJYJYTF [amount]
```

Ajustar la configuración Detener los servicios de Bitcoin y Lightning:

```
$ sudo systemctl stop lnd
$ sudo systemctl stop bitcoind
```

Elimina la billetera LND. Edita el archivo "bitcoin.conf" comentando 'testnet = 1', luego guarda y salte.

```
$ sudo nano
/home/bitcoin/.bitcoin/bitcoin.conf
# remove the following line to enable Bitcoin
mainnet
#testnet=1
```

Copia "bitcoin.conf" actualizado al usuario "admin" para las credenciales (el comando bitcoin-cli busca "rpcpassword")

```
$ sudo cp /home/bitcoin/.bitcoin/bitcoin.conf
/home/admin/.bitcoin/
```

Edita el archivo "lnd.conf" cambiando de bitcoin.testnet=1 a bitcoin.mainnet=1, luego guarda y salte.

```
$ sudo nano /home/bitcoin/.lnd/lnd.conf
# enable either testnet or mainnet
#bitcoin.testnet=1
bitcoin.mainnet=1
```

Elimina los archivos de autorización LND (*.macaroon). Están vinculados a la billetera actualmente activa y deben crearse cuando creamos una nueva billetera para mainnet.

```
$ sudo rm /home/bitcoin/.lnd/*.macaroon
$ sudo rm /home/bitcoin/.lnd/data/macaroons.db
```

Reiniciar bitcoind & lnd para mainnet NOTA: No continúes hasta que la tarea de copia de la cadena de bloques de mainnet haya finalizado por completo. Inicia Bitcoind y comprueba si está funcionando en mainnet:

```
$ sudo systemctl start bitcoind
$ systemctl status bitcoind.service
$ sudo tail -f
/home/bitcoin/.bitcoin/debug.log  (exit with
Ctrl-C)
$ bitcoin-cli getblockchaininfo
```

Espera hasta que la cadena de bloques esté completamente sincronizada. "blocks" = "headers", de lo contrario podrías encontrarse con problemas de rendimiento/memoria al crear una nueva billetera mainnet. Inicia LND y revisa su funcionamiento. Esperará a que se cree la billetera.

```
$ sudo systemctl start lnd
$ systemctl status lnd
```

Crear una billetera mainnet Una vez iniciado LND, debemos crear una nueva billetera Bitcoin integrada

para mainnet. Inicia una sesión de usuario “bitcoin” y crea una nueva billetera

```
$ sudo su - bitcoin
$ lncli create
```

Si quieres crear una nueva billetera, introduce tu contraseña [C] como contraseña de la billetera, selecciona n en relación a una semilla existente e introduce la contraseña opcional [D] como contraseña de la semilla

Figura 3 – Creando una nueva billetera Bitcoin integrada para mainnet

Las 24 palabras iniciales que se muestran, combinadas con tu contraseña opcional, es la copia de seguridad para su Bitcoin en cadena. El estado actual de tus canales, sin embargo, no se pueden volver a crear a partir de esta semilla, esto todavía está bajo desarrollo en LND.

NOTA: esta información debe mantenerse en secreto en todo momento. Escribe estas 24 palabras manualmente en vertical sobre un pedazo de papel y guárdalo en un lugar seguro. ¡Este pedazo de papel es todo lo que necesita un atacante para vaciar completamente tu billetera! No lo guardes en un ordenador. No tomes una foto con tu teléfono móvil. Esta información nunca debe almacenarse en ningún sitio en formato digital.

Salte de la sesión de usuario “bitcoin”. Para usar lncli con el usuario “admin”, copia los archivos de permiso y el certificado TLS. Verifica si está operativo.

```
$ exit
$ sudo cp /home/bitcoin/.lnd/tls.cert
/home/admin/.lnd
$ sudo cp /home/bitcoin/.lnd/admin.macaroon
/home/admin/.lnd
```

Comprueba si funciona al obtener alguna información del nodo

```
$ lncli getinfo
```

Reinicia lnd y desbloquea tu billetera (introduce la contraseña [C])

```
$ sudo systemctl restart lnd
$ lncli unlock
```

Monitoriza el progreso de arranque de LND hasta que se ponga al día de la cadena de bloques de mainnet (aproximadamente 515k bloques en este momento). Esto puede tardar hasta 2 horas, después de lo cual verás una larga parrafada pasar muy rápido. Salte con Ctrl-C.

```
$ sudo journalctl -f -u lnd
```

Este comando devolverá “synced_to_chain: true” si LND está listo.

```
$ lncli getinfo
```

Mejorar el proceso de inicio Lleva un poco de tiempo acostumbrarse al hecho de que la billetera LND debe desbloquearse manualmente cada vez que se reinicia el demonio LND. Esto tiene sentido desde el punto de vista de la seguridad, ya que la billetera está encriptada y la clave no se almacena en la misma máquina. Sin embargo, para operaciones seguras, esto no es óptimo para nada, ya que puedes recuperar LND fácilmente si tiene que reiniciar por alguna razón (por un bloqueo o un corte de energía), pero luego se queda bloqueada y no puede operar en absoluto.

Es por eso que un script que desbloque automáticamente la billetera resulta muy útil. La contraseña se almacena como texto plano en un directorio al que sólo tiene acceso root, claramente no es muy seguro, pero es razonablemente aceptable en este caso. Siempre puede optar por realizar un desbloqueo manual o implementar una solución que desbloquee la billetera desde una máquina remota.

Como usuario “admin”, crea un nuevo directorio y guarda tu contraseña de la billetera LND [C] en un archivo de texto:

```
$ sudo mkdir /etc/lnd
$ sudo nano /etc/lnd/pwd
```

El siguiente script desbloquea la billetera LND a través de su servicio web (interfaz REST). Cópialo en un nuevo archivo.

```
$ sudo nano /etc/lnd/unlock
#!/bin/sh
# LND wallet auto-unlock script
# 2018 by meeDamian, robclark56

# Delay is needed to make sure bitcoind and
lnd are ready.  You can still
# unlock the wallet manually if you like.
Adjust to your needs:
/bin/sleep 300s

LN_ROOT=/home/bitcoin/.lnd

curl -s
        -H "Grpc-Metadata-macaroon: $(xxd -ps
-u -c 1000 ${LN_ROOT}/admin.macaroon)"
        --cacert ${LN_ROOT}/tls.cert
        -d "{"wallet_password": "$(cat
/etc/lnd/pwd | tr -d '
' | base64 -w0)"}"
        https://localhost:8080/v1/unlockwallet
> /dev/null 2>&1

echo "$? $(date)" >> /etc/lnd/unlocks.log
exit 0
```

Haz que el directorio y todo el contenido sólo sea accesible por “root”

```
$ sudo chmod 400 /etc/lnd/pwd
$ sudo chmod 100 /etc/lnd/unlock
$ sudo chown root:root /etc/lnd/*
```

Nota: Me Encontré con el problema de que curl no funcionaba correctamente en mi máquina y tuve que volver a instalar una librería antes de que su funcionamiento fuera el adecuado:

```
$ sudo apt install --reinstall libroken18-
heimdal.
```

Crea una nueva unidad systemd que se inicie inmediatamente después de LND.

```
$ sudo nano /etc/systemd/system/lnd-
unlock.service

# Thundroid: system unit for lnd unlock script
# /etc/systemd/system/lnd-unlock.service

[Unit]
Description=LND wallet unlock
After=lnd.service
Wants=lnd.service

[Service]
ExecStart=/etc/lnd/unlock
Type=simple

[Install]
WantedBy=multi-user.target
```

Edita el archivo de configuración de LND para activar la interfaz REST en el puerto 8080:

```
$ sudo nano /home/bitcoin/.lnd/lnd.conf
# add the following line in the [Application
Options] section
restlisten=localhost:8080
```

Recarga systemd y habilita la nueva unidad. Reinicia tu Thundroid y mira el proceso de inicio para ver si la billetera se desbloquea automáticamente.

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable lnd-unlock.service
$ sudo shutdown -r now
---- reconnect ----
# Unlocking the wallet will take several
minutes due to the build in delay
$ sudo journalctl -u lnd -f
```

Nota: una forma más elegante sería ejecutar el script con ExecStartPost=+/etc/lnd/unlock en la unidad lnd.service. Esto te permitiría desbloquear la billetera si el servicio LND se reinicia fuera del proceso de inicio. El =+ es necesario para ejecutar LND con el usuario “bitcoin” y el script de desbloqueo con privilegios root. Desafortunadamente, esto solo está disponible a partir de la versión de sistema 331, pero nosotros estamos usando la versión 229.

Empezar a usar Lightning Network Financiar tu nodo ¡Felicidades, tu Thundroid ahora está vivo en el Bitcoin mainnet! Para abrir canales y empezar a usarlo, debe financiarlo con un bitcoin. Para empezar, pon solo en tu nodo lo que estás dispuesto a perder, y trátalo como dinero monopolio. Genera una nueva dirección Bitcoin para recibir fondos en cadena:

```
$ lncli newaddress np2wkh
> "address": "3........................."
```

Desde tu billetera Bitcoin habitual, envía una pequeña cantidad de bitcoin a esta dirección, o pídele a su amigo frustrado con los Bitcoin que te envíe algunos dólares. A continuación, comprueba el saldo de tu billetera LND:

```
$ lncli walletbalance
```

Monitoriza tu transacción en un explorador Blockchain tal y como se describe en https://smartbit.com.au.

LND en acción

Tan pronto como tu operación de financiación sea minada y confirmada, LND empezará a abrir y a mantener canales. Esta función se denomina "Autopilot" y está configurada en el archivo "lnd.conf". Si desea mantener sus canales manualmente, puede desactivar el piloto automático.

Puedes usar los mismos comandos que aparecen en la Parte **Parte 1** de esta guía o usa, dirígete a la referencia **LND API** o simplemente escribe lncli –help.

Pruébalo y explora el mainnet de Lightning Existen muchos y excelentes recursos para explorar el mainnet de Lightning con respecto a tu propio nodo.

- **Lightning Spin**: Un simple juego Wheel of Fortune
- **Lightning Network Stores**: Tiendas y servicios que aceptan pagos Lightning
- **Recksplorer**: Lightning Network Map
- **1ML**: Motor de búsqueda y análisis de la red Lightning
- **lnroute.com**: Lista completa de recursos de la red Lightning

¿Qué será lo próximo? Ahora tiene tu propio nodo completo Bitcoin/Lightning. Los objetivos inicialmente definidos fueron los siguientes y los hemos conseguidos todos:

- Una validación completa del nodo completo Bitcoin que no requiere confiar en un tercero
- Funciona de un modo fiable 24/7
- Soporta la descentralización de la red Lightning mediante el enrutamiento de pagos
- Se puede usar para enviar y recibir pagos personales usando la interfaz de línea de comando.
- ¿funcionalidad? No tanta...

¿El nodo Bitcoin Lightning ya es perfecto? Es torpe y la línea de comando simplemente no da la talla. En la Parte 3 de esta guía, continuaremos ampliando Thundroid con aplicaciones adicionales que lo utilizan como nuestro propio backend privado.

- La billetera de escritorio Electrum es la billetera perfecta para usuarios avanzados que manejan transacciones regulares de Bitcoin en cadena. Debido a que es compatible con una amplia variedad de billeteras de hardware, tus claves privadas nunca deben exponerse a ningún ordenador on-line (posiblemente comprometido). Con Electrum Personal Server ejecutándose en Thundroid, usted tiene control total para enviar, recibir y verificar transacciones de Bitcoin con una gran seguridad y privacidad.

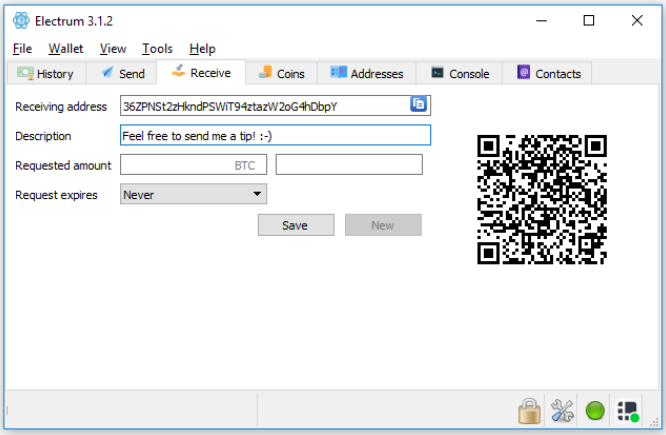


Figura 4 – La billetera de escritorio Electrum

- La billetera móvil Shango lightning es perfecta para pagos que sean pequeños e instantáneos sobre la marcha. Se conecta a tu Thundroid y proporciona una interfaz de usuario ordenada en tu teléfono iOS/Android para enviar y recibir pagos, y administrar pares y canales. Aún está en beta cerrado, espero que se haga público justo a tiempo.

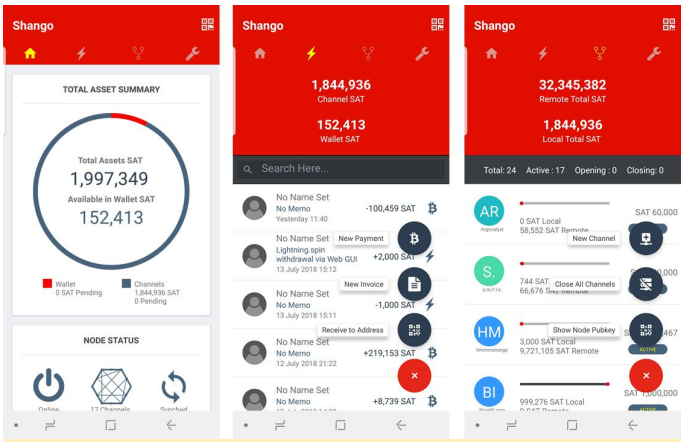


Figura 5 – La billetera móvil Shango Lightning

¡Acompáñame en la parte 3 de la guía “El nodo de Bitcoin Lightning perfecto” y descubre algunas aplicaciones punteras que funcionan muy bien sobre nuestro completo nodo Bitcoin!

Recuperar eMMC: Reseteando el Módulo eMMC ODROID-XU4 para Solucionar Problemas de Arranque

August 1, 2018 By Justin Lee ODROID-XU4, Mecaniqueo, Tutoriales

El cargador de arranque de la serie Exynos está ubicado en una partición de arranque oculta dentro de la memoria eMMC para todos los modelos, excepto para el ODROID-C1/C2. Cuando éste se daña, o se quiere utilizar el módulo eMMC en una placa diferente, se debe instalar el cargador de arranque correcto en el módulo eMMC. Ten en cuenta que necesitas tener una tarjeta micro SD en blanco para ejecutar el proceso de recuperación.

Recuperar con una imagen de recuperación

- Descargar el archivo de imagen de recuperación.
- Preparar una tarjeta micro SD y grabar la imagen descargada
- Conectar tanto el módulo eMMC como la tarjeta micro SD en el ODROID-XU3 / XU4
- Configurar los interruptores DIP o el interruptor deslizante del XU3/XU4 en el “Modo de arranque SD”
- Conectar la fuente de alimentación y observar el estado del LED
- Los LED azules y rojos deben permanecer encendidos, el proceso puede durar entre 40 segundos y 3 minutos
- Finalizado el proceso de recuperación, el LED azul parpadeará como el latido de un corazón, llegados a este punto puede retirar la fuente de alimentación.
- Configurar los interruptores DIP o desliza nuevamente el interruptor al modo de arranque eMMC
- Retirar la tarjeta micro SD
- Proceder con un encendido normal con tus periféricos conectados

Tras verificar que Android arranca con el módulo eMMC, puede grabar otra imagen de sistema operativo en el eMMC, y el gestor de arranque cargará el nuevo sistema operativo.

Recuperar eMMC con una tarjeta micro SD y u kit USB-UART

Estas instrucciones requieren un kit USB-UART y una aplicación de terminal

- Ajustar los interruptores DIP o deslizar el interruptor en el XU3/XU4 al “modo de arranque SD”
- Insertar la micro SD con la imagen de inicio en su ranura y encenderlo, detenerse en U-boot una vez que la placa se haya encendido
- Introducir el comando “run copy_uboot_sd2emmc” para copiar la imagen del cargador de arranque desde la micro SD al módulo eMMC
- Una vez finalizada la copia, configurar los interruptores DIP o deslizar el interruptor nuevamente al modo de arranque eMMC
- Proceder con el encendido normal con tus periféricos conectados
- Tras verificar que Android arranca en el módulo eMMC, puedes grabar otra imagen del sistema operativo en el eMMC, y el gestor de arranque cargará el nuevo sistema operativo

El ODROID-XU4 tiene un interruptor deslizante para elegir el soporte de arranque, tal y como se muestra en la Figura 1. El interruptor de configuración de arranque debe configurarse para iniciarse desde la tarjeta SD si no tienes un módulo eMMC conectado. El dispositivo no vuelve automáticamente a la tarjeta SD si no hay conectado un módulo eMMC

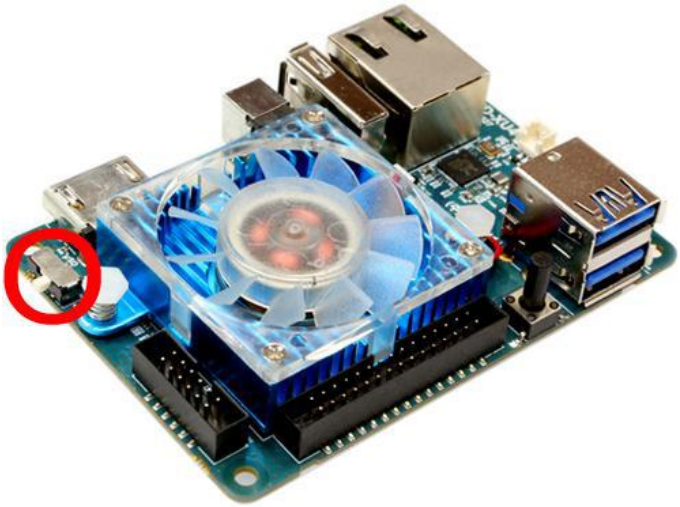


Figura 1 – Ubicación del selector del modo de arranque en el ODROID-XU4

Campamento de Programación Parte 2: Cómo Mostrar “Hello, ODROID-GO” en una Pantalla LCD

August 1, 2018 By Justin Lee Mecaniqueo, Tutoriales, ODROID-GO

En este artículo, aprenderás cómo mostrar una cadena de texto, cambiar los colores y cambiar el tamaño de la fuente. Si sigues esta guía, serás capaz de escribir el código que te permitirá mostrar “Hello, ODROID-GO” en tu ODROID-GO.

Estructura básica de código para Arduino Cuando ejecutas por primera vez Arduino IDE, verás una pantalla similar a la que se muestra en la Figura 1.

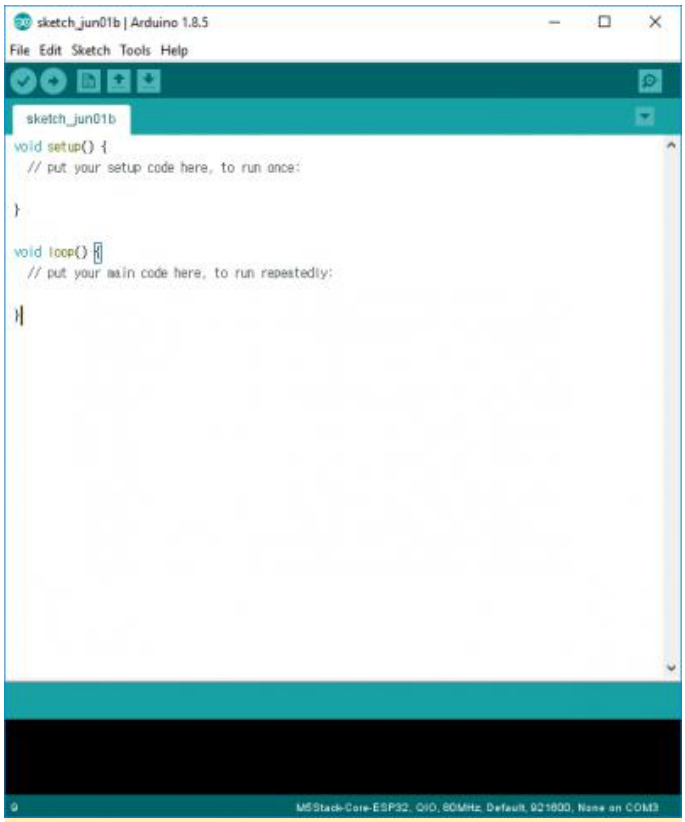


Figura 1 – Sketch para Arduino

Este editor se llama Sketch, y será tu patio de recreo. El código fuente por defecto es:

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run
repeatedly:

}
```

Hay dos funciones con algunos comentarios que te permiten conocer qué hace exactamente cada función en el código. Utilizaremos esta simple estructura.

Arduino para ODROID-GO Usaremos la librería de desarrollo de Arduino: odroid_go.h. Esta librería te ayuda a controlar los componentes de la placa, como son la LCD, los botones, el altavoz, etc. La librería debe estar incluida desde el principio.

Para preparar la placa para su uso, debes activarla. Para inicializar la placa, usa la función GO.begin(). Si desea controlar los botones o el altavoz de la placa, debes usar la función GO.update() para aplicar los cambios desde el código.

La función GO.update() no se utiliza en esta guía, ya que solo necesitaremos utilizar la pantalla LCD para mostrar una simple cadena.

```
#include

void setup() {
  // put your setup code here, to run once:
  GO.begin();
}

void loop() {
  // put your main code here, to run
repeatedly:

}
```

La función GO.begin() debe incluirse dentro de la función setup() ya que solo es llamada una vez. La instancia GO no solo tiene las dos funciones principales, sino también muchas funciones auxiliares que te permiten controlar los componentes de la placa. Ahora, usemos las funciones GO.lcd para mostrar “Hello, ODROID-GO”.

Hello World Utilizaremos la función GO.lcd.print para mostrar una cadena:

```
#include

void setup() {
  // put your setup code here, to run once:
  GO.begin();

  GO.lcd.print("Hello, ODROID-GO");
}

void loop() {
  // put your main code here, to run
repeatedly:

}
```

En sketch se ve bien, pero el texto en la pantalla LCD será demasiado pequeño para verlo. Aumentemos el tamaño de fuente a 2 usando la función GO.lcd.setTextSize().

```
#include

void setup() {
  // put your setup code here, to run once:
  GO.begin();

  GO.lcd.setTextSize(2);
  GO.lcd.print("Hello, ODROID-GO");
}

void loop() {
  // put your main code here, to run
repeatedly:

}
```

También puedes cambiar el color del texto con GO.lcd.setTextColor(). Cambia el texto a verde.

```
#include

void setup() {
  // put your setup code here, to run once:
  GO.begin();

  GO.lcd.setTextSize(2);
  GO.lcd.setTextColor(GREEN);
  GO.lcd.print("Hello, ODROID-GO");
}

void loop() {
  // put your main code here, to run
repeatedly:

}
```

Como característica avanzada, también hemos añadido una función llamada displayGO() que incluye varios efectos. Las nuevas funciones introducidas incluyen:

- GO.lcd.setRotation(): gira la pantalla de salida. El parámetro de rotación puede ser de 0 a 7.
- GO.lcd.clearDisplay(): resetea todos los textos de la pantalla.
- GO.lcd.setTextFont(): fija el estilo de la fuente después de llamar a ésta. El nombre de una determinada fuente se especifica mediante un número.

```
#include

uint8_t idx;
uint8_t rotate;

void setup() {
  // put your setup code here, to run once:
  GO.begin();

  GO.lcd.println("Hello, ODROID-GO");
  delay(1000);
}

void displayGO() {
  GO.lcd.clearDisplay();
  GO.lcd.setRotation(rotate + 4);
  GO.lcd.setCursor(30, 40);

  if (idx) {
    GO.lcd.setTextSize(1);
    GO.lcd.setTextFont(4);
    GO.lcd.setTextColor(MAGENTA);
  } else {
    GO.lcd.setTextSize(2);
    GO.lcd.setTextFont(1);
    GO.lcd.setTextColor(GREEN);
  }
  GO.lcd.print("Hello, ODROID-GO");

  idx = !idx;
  rotate++;
  rotate %= 4;

  delay(1000);
}

void loop() {
  // put your main code here, to run
repeatedly:
  displayGO();
}
```

Puedes verificar, compilar o cargar cualquier apunte desde la barra de herramientas o desde el menú Sketch. Aquí tienes algunos atajos útiles que puedes usar:

- CTRL-R: Verificar y compilar.
- CTRL-U: Cargar

Antes de cargar el archivo binario, debes seleccionar el puerto correcto en el menú Tools – Port. Si el

proceso ha ido bien, podrás ver “Hello, ODROID-GO” en tu dispositivo.

Figura 2 – ODROID-GO

Un ejemplo completo El ejemplo completo está disponible haciendo clic en el menú Files → Examples → ODROID-GO → Hello_World para importarlo y presionar CTRL-U para compilarlo/cargarlo.

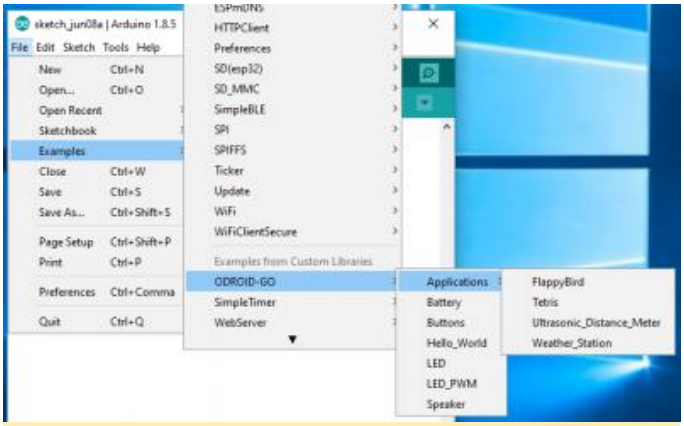


Figura 3 – Un ejemplo completo

Para comentarios, preguntas y sugerencias, visita el artículo original en **https://wiki.odroid.com/odroid_go/arduino/02_hello_world**.

Cómo Configurar un Servidor Minecraft

August 1, 2018 By @qkpham Juegos, Linux, Tutoriales

¡Casi todo el mundo adora los juegos, especialmente Minecraft! Lo han disfrutado más de 14 millones de personas en todo el mundo por su sistema de juego adictivo y mapas personalizables. Aunque el paquete oficial de Mojang Software es de código cerrado, hay disponible varias versiones Java de código abierto de Minecraft Server para la plataforma ODROID. ¡Programar un mundo virtual usando un paquete gratuito de Minecraft Server como Spigot, Bukkit o BungeeCord es también una excelente forma de aprender Java mientras te diviertes!

Este artículo detalla cómo instalar un servidor básico de Minecraft en tu ODROID, para que puedas ejecutar juegos online con algunos de tus amigos en un mundo creado por ti. Usar el ODROID como un sandbox – entorno de pruebas – de bajo coste también es una excelente forma de probar mapas, actualizaciones y modificaciones antes de subirlos a un servidor público.

Requisitos

- 1. Un ODROID de la serie X, U o XU
- 2. Un módulo eMMC o tarjeta MicroSD Clase 10+ de al menos 8GB
- 3. Una imagen personalizada de Ubuntu, Debian o similar (13.04 o superior), disponible en los foros de ODROID (**http://forum.odroid.com/**)
- 4. Java versión 1.8 (OpenJDK8 u Oracle JDK8)
- 5. Conexión de red de área local (LAN), que incluya un router con la función de reenvío de puertos

Instalar Java

Si la versión 1.8 de Java todavía no la tienes instalada en tu sistema, consulta el artículo de ODROID Magazine sobre la Instalación de Oracle JDK8. Mojang publica una versión Java del software Minecraft compatible con otros sistemas operativos como ARM Linux.

Instalar Minecraft

Primero, descarga el último software de Minecraft Server desde el sitio oficial en **https://minecraft.net/download**, asegurándote de coger la versión .tar basada en Java.

Crea un directorio de Minecraft en tu directorio de inicio para almacenar el minecraft_server.jar descargado. Una vez descargado el tarball, escribe los siguientes comandos para iniciar el servidor:

```
$ cd ~/minecraft
$ java -Xms1536M -Xmx1536M -jar
minecraft_server.jar nogui
```

¡El servidor de Minecraft debería estar ejecutándose ahora! El último paso es conseguir la dirección IP del servidor para que nuestros jugadores puedan conectarse a través de sus clientes Minecraft.

Obtener la dirección IP interna

Averigua la dirección IP interna (local) de tu servidor escribiendo ifconfig en la ventana del Terminal y localizando la etiqueta inet addr. En mi ODROID, la dirección IP aparecía como 192.168.1.10. Asegúrate de que esta dirección tenga una alta asignación emitida por el servidor DHCP local o el router para evitar actualizaciones frecuentes de la configuración.

Configurar el reenvío de puertos

Minecraft utiliza el puerto TCP 25565, que tu router local debe reenviar a la dirección IP del servidor mediante el reenvío de puertos. Consulta el manual del usuario para obtener ayuda sobre cómo configurar el router y poder redireccionar el puerto 25565 a la dirección IP obtenida en el paso anterior.

Obtener la dirección IP externa

La dirección IP pública que identifica tu LAN en el mundo exterior puedes obtenerla visitando http://www.whatismyip.com. La dirección tendrá el formato aaa.bbb.ccc.ddd, lo que significa que la URL completa para conectarse al Servidor Minecraft en tu LAN será http: //aaa.bbb.ccc.ddd: 25565. Hay que tener en cuenta el puerto TCP y la URL.

Si tu IP externa es dinámica (generalmente cambiada por tu ISP), puede usar servicios como No-IP. Puedes crear una cuenta en su sitio web, luego descargar e instalar el Cliente de actualización de DNS dinámico (DUC) desde **http://www.noip.com/download**. Las instrucciones detalladas sobre la configuración de DNS dinámico las puedes encontrar en **http://bit.ly/1ggmo2n**. En este caso, la dirección completa del servidor Minecraft será http://youracctusername.no-ip.com:25565.

Para asegurarte de que todo funciona, puede probar que tu servidor esté visible online en **http://www.canyouseeme.org**. También puede verificar rápidamente su estado desde **http://dinnerbone.com/minecraft/tools/status/**.

El rendimiento del sistema será aceptable en condiciones normales con una conexión inalámbrica, pero una conexión por cable disminuirá la latencia y aumentará la capacidad de respuesta del juego.

Unirse al juego

Inicia tu cliente de Minecraft en una máquina con Windows o OSX introduciendo la dirección IP pública del paso anterior (http: //aaa.bbb.ccc.ddd:25565) añadiendo un nuevo servidor a la lista de servidores del cliente. Cuando escribí esta guía, el software Minecraft Client desafortunadamente aún no se podía ejecutar en la plataforma ODROID. Hay una Edición de Minecraft Pocket disponible para Android, pero no es compatible con la versión completa de Minecraft Server.

Una conexión exitosa con el servidor Minecraft ODROID llevará al usuario a nuestro mundo virtual tal y como se ve en la Figura 1.

Configuración adicional del servidor

Las opciones del servidor en Minecraft se configuran editando el archivo server.properties ubicado en

/home/yourusername/minecraft/server.properties:

```
#Minecraft server properties
#Mon Dec 24 09:23:18 EST 2012
#
generator-settings=
level-name=world
enable-query=false
allow-flight=false
server-port=25565
level-type=DEFAULT
enable-rcon=false
level-seed=
server-ip=
max-build-height=256
spawn-npcs=true
white-list=false
spawn-animals=true
hardcore=false
texture-pack=
online-mode=true
pvp=true
difficulty=1
gamemode=0
max-players=20
spawn-monsters=true
generate-structures=true
view-distance=10
motd=A Minecraft Server
```

Los tres parámetros útiles para cambiar mapas y mejorar el rendimiento son:

- level-name: si quieres añadir otro mapa o mundo a tu servidor, simplemente descomprime el archivo del mundo dentro de tu carpeta Minecraft y luego cambia la configuración del nombre del nivel al nombre de esa carpeta. Por ejemplo, si la carpeta de mundo que has extraído es odroid, cambia el valor de level-name a odroid en lugar del valor world por defecto
- view-distance: se puede reducir a 7 para mejorar la capacidad de respuesta del servidor
- max-players: funciona mejor cuando se fija entre 2 y 5

Ten en cuenta que Minecraft depende en gran medida de las operaciones de punto flotante. A diferencia de las CPU basadas en arquitectura x86, los SOC basados en ARM no están optimizados para operaciones de punto flotante, así que las opciones del servidor deben ajustarse para compensar la carga de trabajo más pesada.

Si deseas mejorar aún más el rendimiento, existen varias versiones de código abierto de Minecraft Server que disminuyen significativamente los cálculos del servidor, proporcionando una experiencia más fluida y permitiendo que se unan al juego más jugadores.

Craftbukkit

Crea una carpeta para Craftbukkit escribiendo mkdir ~/craftbukkit en una ventana de Terminal, luego visita **https://dl.bukkit.org/downloads/craftbukkit/** para descargar la última versión de Craftbukkit al directorio recién creado. Una vez completada la descarga, ejecuta el servidor para construir tu mundo.

```
java -Xms1536M -Xmx1536M -jar craftbukkit.jar
cd ~/craftbukkit/plugins
wget
http://dev.bukkit.org/media/files/674/323/NoLa
gg.jar
wget
http://dev.bukkit.org/media/files/665/783/PTwe
aks.jar
wget
http://dev.bukkit.org/media/files/586/974/NoSp
awnChunks.jar
```

Spigot

Una alternativa a Craftbukkit es Spigot, que ofrece más opciones de configuración y está optimizado para un mayor rendimiento y velocidad. Siguiendo el mismo procedimiento que hemos descrito anteriormente, descarga el paquete Spigot, que se encuentra en http://www.spigotmc.org/.

```
mkdir ~/spigot
cd spigot
wget http://ci.md-
5.net/job/Spigot/lastSuccessfulBuild/artifact/
Spigot/target/spigot.jar
java -Xms1536M -Xmx1536M -jar spigot.jar
```

Spigot es muy estable y, puesto que está basado en Craftbukkit, los plugins Bukkit de NoLagg, PTweaks y NoSpawnChunks anteriores también funcionarán con Spigot.

MineOS

MineOS es un panel administrativo basado en web que permite una fácil administración de los servidores de Minecraft. Te permite manejar Vanilla, Bukkit, Tekkit y Canary por defecto, pero puedes instalar cualquier otro sistema de servidor y configurarlo para que descargues automáticamente una nueva versión siempre que esté disponible.

Copiar tu servidor a un servicio de alojamiento externo

Utilizar una versión de código abierto de Minecraft te permite cambiar cualquier aspecto del servidor, incluyendo la corrección de errores y la instalación de plugins. Dado que Minecraft para ODROID está escrito en Java, es fácil para principiantes y expertos mejorar el software y personalizarlo en base a sus propias necesidades.

Una vez que tengas tu mundo listo, puedes migrar tu creación de Minecraft a un servidor de alto tráfico para que pueda incorporar más jugadores. Simplemente carga todos los archivos del servidor desde directorio minecraft, spigot o craftbukkit del ODROID a través del panel de administración del hosting web.

¡Disfruta de tu nuevo servidor ODROID Minecraft y recuerda mantenerte alejado de la lava! Para obtener más información o hacer preguntas, visita el hilo del foro original en **http://forum.odroid.com/viewtopic.php?f=52&t=84**.