

Bash • Carputer • Marco Fotos • Servidor Doméstico • Desarrollo Android

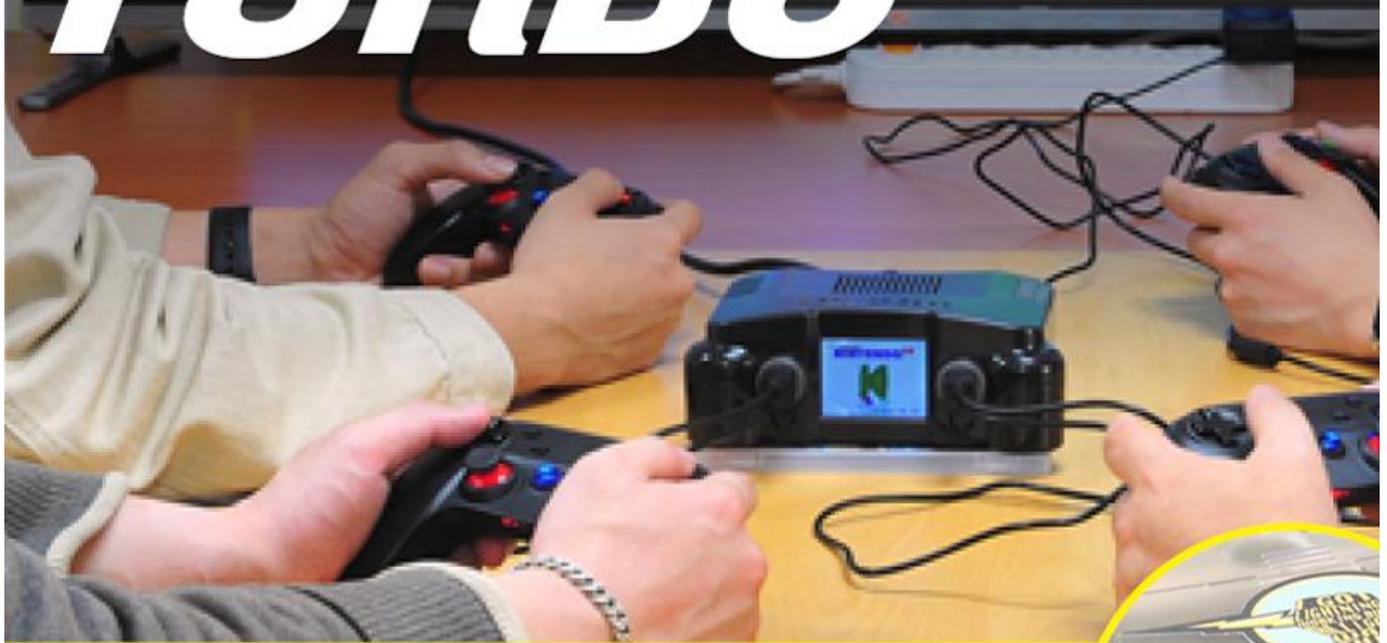
ODROID

Año Cinco
Num. #54
Jun 2018

Magazine

Game Station TURBO

¡DISEÑO DE LA
NOSTÁLGICA
CONSOLA DE
JUEGOS DE LA
VIEJA ESCUELA
EN TU XU4!



MICROSERVIDOR CON
ENERGIA SOLAR :
INFRAESTRUCTURA CUANDO LA NECESITES

THUNDROID:
EL PERFECTO NODO
LIGHTNING BITCOIN





Thundroid: El Perfecto Nodo Bitcoin Lightning

© June 1, 2018

El Bitcoin es una moneda digital nativa de la era de Internet. Podría decirse que otra moneda internacional, pero sin un país de origen, de modo que desafía las fronteras, las políticas comerciales y la inflación arbitraria. En el documento técnico de 2008 (<https://bitcoin.org/bitcoin.pdf>) Satoshi Nakamoto la describe como "... [▶](#)



Juegos Linux: Emulación Nintendo 64 – Parte 1

© June 1, 2018

Me llevó un tiempo lograr que la emulación N64 funcionara en todas las placas ODROID bajo Linux. Sin embargo, ahora que lo he conseguido, es bastante divertido y abre muchas posibilidades a la hora de ejecutar los clásicos juegos de N64. Esperemos que en el futuro veamos más mejoras y [▶](#)



Marco de Fotos Digital: Pantalla 4K de 55 Pulgadas Para Fotos Digitales por unos 400\$

© June 1, 2018

Existen muchos tutoriales sobre cómo hacer un buen marco de fotos digital con una Raspberry Pi.



SO Destacado: ODROID GameStation Turbo

© June 1, 2018

Uno de los proyectos más destacados en los que estoy trabajando para la comunidad ODROID es la imagen ODROID GameStation Turbo, que funciona como interfaz para juegos y reproducción multimedia. Está pensada para ser un sistema de entretenimiento que te permita controlar tu ODROID con solo usar un mando de [▶](#)



Kit Consola de Juegos OGST para ODROID-XU4

© June 1, 2018

El kit consola de juegos OGST para ODROID-XU4 te permite montar tu propia consola de juegos con un potente ODROID-XU4 u ODROID-XU4Q. Su atractivo diseño incluye una elegante pantalla LCD de 2.4" que permite mostrar animaciones del logotipo programables de los juegos, y está específicamente diseñado para funcionar con la [▶](#)



Joystick ODROID GameSir G3w USB

© June 1, 2018

El GameSir G3w es un gamepad de alta calidad que utiliza un chip MCU de 32 bits, con una capacidad informática de hasta 48 millones de operaciones por segundo. Es compatible con los sistemas operativos oficiales de Linux y Android que ofrece

Hardkernel.



Micro Servidor Alimentado con Energía Solar

© June 1, 2018

Los apagones no solo son molestos, sino que causan una serie de problemas que afectan a casi todos los aspectos de la vida moderna, así que, tras meses en la oscuridad, decidí dar el paso y construir un pequeño sistema de energía solar. El

ODROID C2 tiene una CPU ARM ▶



Desarrollo Android: Usando GitHub

© June 1, 2018

¡Bienvenido de nuevo, desarrollador de apps! Si eres como yo, estás más que preparado para mejorar tus habilidades en el desarrollo de aplicaciones. Como he mencionado en otras ocasiones, la empresa que domina prácticamente todo el monopolio de proyectos de código abierto online es GitHub. Existen varias opciones basadas en ▶

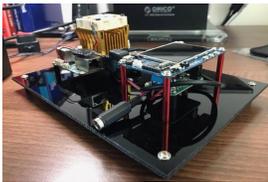


Juegos Linux en ODROID: Juegos Saturn – Parte 4

© June 1, 2018

Volvemos con los juegos de ODROID-XU4/XU4 y Sega Saturn. Esta vez vamos a echar un vistazo a los juegos que empiezan por la letra "S" como Saturn o como "shmup". Existen tantos juegos para Sega Saturn que empiezan por la "S" con los que disfruto

jugando, que he tenido que ▶



Servidor Doméstico ODROID-XU4

© June 1, 2018

En diciembre de 2017, monté mi propio servidor multimedia doméstico Odroid XU4 (<https://goo.gl/6tT6rt>) ya que tuve algunos problemas con la configuración anterior. Lamentablemente, en este montaje no tuve en cuenta ni la estética ni el cableado, solo

me centré en la funcionalidad, porque necesitaba un servidor funcional y no me ▶



Carputer – Pantalla Táctil Android de 7"

© June 1, 2018

Este es un CarPuter Android de pantalla táctil de 7" con GPS USB muy preciso, Bluetooth 4, entrada y salida de audio de 3.5MM, WiFi y una pantalla magnética ajustable. Tienes disponible un archivo sketchup para personalizarlo y cambiar el

tamaño según sea necesario en <https://www.thingiverse.com/thing:2720349>. Como técnico de Servicio Locales ▶



Introducción a los Conceptos Básicos de BASH – Parte 2: Comandos BASH Útiles para Ordenadores de placa Reducida

© June 1, 2018

Last time, we learned about the 'ls' and 'tree' commands. While looking at things is nice, it's more fun to actually do something with our data. This article contains a list of the common commands for manipulating data. The command and its explanation are kept very brief to avoid writing ▶



Juegos Linux: Emulación Nintendo 64 - Parte 2

© June 1, 2018

La Parte 1 de este artículo hemos presentado la última versión del emulador de Nintendo 64 para Linux y hemos comparado su rendimiento en todas las placas ODROID actuales. Esta segunda parte presenta una descripción general de algunos de los juegos más populares de Nintendo 64, entre los que se [▶](#)

Thundroid: El Perfecto Nodo Bitcoin Lightning

© June 1, 2018 By @stadicus ↗ ODRROID-XU4, Tutoriales



El Bitcoin es una moneda digital nativa de la era de Internet. Podría decirse que otra moneda internacional, pero sin un país de origen, de modo que desafía las fronteras, las políticas comerciales y la inflación arbitraria. En el documento técnico de 2008 (<https://bitcoin.org/bitcoin.pdf>) Satoshi Nakamoto la describe como "... una versión electrónica de dinero puramente de igual a igual [que] permitiría realizar pagos on line directamente desde una parte a otra sin pasar por una institución financiera".

Ser de igual a igual significa que Bitcoin no depende de un intermediario, como un banco, y puede transferirse como un activo al portador, como el dinero físico, sin pedirle permiso a nadie. No necesita almacenarse físicamente ya que está protegido por una clave criptográfica, de modo que puede transferirse en minutos a cualquier persona en cualquier parte del mundo. Un componente clave de este sistema financiero de código abierto gratuito es el blockchain, un libro de contabilidad que realiza un

seguimiento de quién posee bitcoin y cuánto posee. Éste es almacenado como una copia idéntica por todos los usuarios que deciden ejecutar un nodo completo Bitcoin. Puedes aprender más en bitcoin.org.

El Bitcoin es un experimento económico de una magnitud épica, y su éxito no es seguro en absoluto. En cualquier caso, Bitcoin como nueva tecnología es un esfuerzo increíblemente interesante, especialmente debido a su naturaleza interdisciplinaria y la baja barrera para su acceso. El Bitcoin como moneda sólida, escasa y no inflacionaria que desafía el dinero como uno de los últimos auténticos monopolios de los estados nacionales, podría tener un gran impacto en los principios económicos y en la sociedad en general.

De momento, Bitcoin es más bien un depósito de valor y realmente no está hecho para hacer pequeñas transacciones diarias. Los verdaderos blockchains

descentralizados son un recurso escaso y no pueden escalarse para dar cabida a todas las transacciones globales. Si lo piensas bien, no puede ser muy adecuado almacenar todas compras de un café de manera redundante en todo el mundo para toda la eternidad. Sería como obligar a que todo el mundo se descargara el correo electrónico de toda la gente. Estas limitaciones podrían motivar la creación de una mejor tecnología en lo que respecta a los blockchains de Bitcoin que permitiera escalarse exponencialmente, en lugar de simplemente hacerse más grandes de forma lineal.

Aquí es donde entra en acción la denominada "Lightning Network". Como una de múltiples y novedosas extensiones del blockchain, promete dar cabida a transacciones casi ilimitadas, con confirmación instantánea, tarifas mínimas y una mayor privacidad. Suena a algo demasiado bueno para ser verdad, pero esta tecnología está muy explotada, comprometida con los valores del código abierto de cypherpunk y aprovecha los sólidos fundamentos del Bitcoin. [Pincha aquí para saber más.](#)

Para preservar la naturaleza descentralizada de este sistema monetario, es importante que todo el mundo tenga al menos la posibilidad de ejecutar su propio nodo Bitcoin, preferiblemente con un hardware de bajo coste como el ODROID.

NOTA: Ten en cuenta que, si bien el Bitcoin ha sido puesto a prueba durante casi una década, Lightning Network aún está en fase beta y su desarrollo continúa siendo muy elevado. Esta guía también te permite configurar tu nodo Bitcoin sin tener en cuenta la parte Lightning. Lee con atención la sección "Mejores prácticas financieras" antes de comprometer tu nodo al Bitcoin real.

Objetivo

Esta guía te permitirá ser tu propio banco. El objetivo es configurar un nodo Bitcoin y Lightning que:

- Permita validar totalmente un Nodo Bitcoin completo y no requiera confiar en un tercero.
- Sea fiable tenerlo activo las 24 horas del día, los 7 días de la semana

- Forme parte de y apoye la descentralización de la red Lightning mediante el enrutamiento de pagos
- Se pueda usar para enviar y recibir pagos personales usando la interfaz de línea de comando.

El servidor está configurado sin interfaz gráfica de usuario y se gestiona de forma remota utilizando la línea de comandos Secure Shell (SSH). También puede funcionar como un backend de Bitcoin personal para la billetera de escritorio Electrum.

Destinatarios

Aunque esta guía se esfuerza por proporcionar instrucciones simples y evidentes, el objetivo también es hacerlo todo por nuestra cuenta, sin atajos que impliquen confiar en un tercero. Esto ha hecho que esta guía se vuelva un tanto técnica, pero he tratado de hacerla lo más sencilla posible para que logres entender lo básico sobre el cómo y el por qué.

Advertencia

Todos los componentes de la red Lightning están aún en desarrollo y aquí manejamos dinero real. Esta guía sigue un enfoque conservador: primero configura y pruebalo todo en TestNet Bitcoin, luego, una vez que estés lo suficientemente seguro como para poner dinero real en línea, cambia a Mainnet Bitcoin con unos simples cambios.

Preparativos

Tras publicar la "Guía para principiantes sobre Lightning en una Raspberry Pi", empecé a explorar otro hardware ya que la Raspberry Pi tiene algunos inconvenientes, principalmente en el área de rendimiento y la dificultad que presenta a la hora de conectar sistemas de almacenamiento externo, algo que es muy importante a la hora de almacenar el amplio blockchain Bitcoin.

El ODROID-HC2 de Hardkernel (http://www.hardkernel.com/main/products/prdt_info.php) o el HC1 en un formato más pequeño como mini PC basada en Linux, se adapta perfectamente. Comparandolo con un Raspberry Pi, tiene las siguientes ventajas:

- Precio equiparable al de la Raspberry Pi

- Más potente (CPU de 8 núcleos, 2 GB de RAM, Ethernet Gigabit)
- Alojamiento interno con disco duro, conexión directa con SATA3
- Solo un adaptador de corriente para todo

No cuenta con determinadas características como son la salida HDMI, Wi-Fi integrado o los pines GPIO, pero éstas no son importantes para este proyecto en concreto. El rendimiento es bastante mayor, de modo que parece estar preparado para el futuro ya que estoy seguro que el Bitcoin y Lightning seguirán evolucionando.

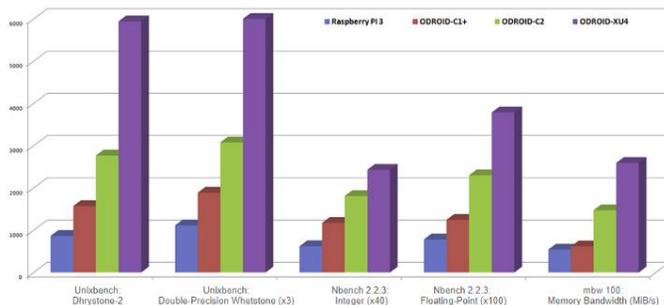


Figura 1 – El rendimiento de ODROID-HC2 es idéntico al XU4 (que es más bien un PC multimedia)

Una vez que opté por este hardware, compre el ODROID-HC2 y, después de configurarlo y ejecutarlo durante dos meses, puedo decir que es lo mejor opción para un nodo Bitcoin/Lightning de bajo coste. Como este proyecto también necesita un nombre ñoño, llamaré a mi nodo Thundroid.

Hice un pedido con los siguientes artículos directamente a Hardkernel en Singapur. Hay [distribuidores](#) disponibles en todo el mundo, pero desafortunadamente no para Suiza.

- ODROID-HC2 – 54\$ (http://www.hardkernel.com/main/products/prdt_info.php?g_code=G151505170472)
- Adaptador de corriente y cable – 7\$ (http://www.hardkernel.com/main/products/prdt_info.php?g_code=G151578376044)
- Carcasa ODROID-HC2 (opcional) – 5\$ (http://www.hardkernel.com/main/products/prdt_info.php?g_code=G151)

También necesitas lo siguiente:

- Tarjeta Micro SD: 16 GB, incluido un adaptador para tu ordenador habitual
- Disco duro interno: 500 GB o más, SATA, HDD 3,5 “o HDD/SSD de 2,5”
- Cable de red RJ45

El montaje es sencillo: simplemente inserta el disco duro y fíjalo con los tornillos incluidos con tu ODROID. Si has comprado la carcasa de plástico, deslízcala sobre la estructura de metal.

Instalando el sistema operativo

Utilizamos Ubuntu 16.04.03 LTE (Sistema operativo mínimo, básico) proporcionado por Hardkernel. Descarga la imagen desde la sección de ODROID-XU4 en wiki.odroid.com

Archivo exacto utilizado: https://odroid.in/ubuntu_16.04lts/ubuntu-16.04.3-4.14-minimal-odroid-xu4-20171213.img.xz

Descarga la imagen, guárdala en tu tarjeta MicroSD, colócala en tu Thundroid, conéctala a tu red a través de un cable y enchufa el adaptador de corriente. El arranque inicial puede durar varios minutos.

Configura tu router de red para que éste asigne una dirección IP estática a tu Thundroid.

Trabajando en tu Thundroid

Anota tus contraseñas

Necesitarás varias contraseñas, me resulta más fácil escribirlas todas al principio, en lugar de buscarlas a lo largo de toda la guía. Deben ser únicas y seguras, con una extensión de al menos de 12 caracteres. No uses espacios en blanco, comillas o caracteres especiales poco comunes

- Contraseña de usuario
- Contraseña de Bitcoin RPC
- Contraseña de la API Lightning
- Contraseña de semilla Lightning

Guarda una copia de tus contraseñas en un lugar seguro (preferiblemente en un administrador de contraseñas como KeePass o LastPass) y mantén tus

notas originales fuera de la vista una vez que el sistema esté funcionando.

La línea de comandos

Todo está configurado desde el prompt de comandos de Linux. A lo largo de toda esta guía utilizaré las siguientes anotaciones:

#: Esto es un comentario, sólo para información \$: Esto es un comando de una sola línea para introducir (sin \$) y confirmar con la tecla Intro Sin prefijo: Esto es el resultado del comando anterior o algo que se puede copiar/pegar en un archivo.

- Comandos de autocompletado: cuando introduces comandos, puedes usar el tabulador para completar automáticamente, ej. para comandos, directorios o nombres de archivo.
- Historial de comandos: presionando las flechas hacia arriba y hacia abajo en el teclado, puede recuperar los comandos introducidos previamente.
- Usar los privilegios de administrador: nuestros usuarios no tienen privilegios de administrador. Si un comando necesita editar la configuración del sistema, necesitamos usar el comando 'sudo' ("superuser do") como prefijo. En lugar de editar un archivo de sistema con "nano /etc/fstab", usaremos "sudo nano /etc/fstab".
- Usando el editor de texto Nano: Utilizaremos el editor Nano para crear nuevos archivos de texto o editar los existentes. No es complicado, pero guardar y salir no es muy intuitivo. Para Guardar: presiona "Ctrl-O" (para Salir), confirma el nombre del archivo y presiona "Intro". Para Salir: presionar "Ctrl-X"

* ** Copiar/Pegar **: si estás utilizando Windows y el cliente SSH PuTTY, puede copiar texto desde el shell seleccionándolo con el ratón (sin necesidad de hacer clic en nada) y pegar cosas en la posición del cursor con un clic derecho en cualquier parte de la ventana SSH.

Conectando a Thundroid

Es hora de conectarse a través de SSH y ponerse a trabajar. Para ello, necesitas un cliente Secure Shell (SSH). Instalar, iniciar y conectar:

- Windows: recomiendo usar el cliente SSH [KiTTY] (<http://kitty.9bis.com>). Puedes copiar texto desde el

shell seleccionándolo con el ratón (sin necesidad de hacer clic en nada) y pegar cosas con un clic derecho.

- Mac OS: cliente SSH integrado (<http://osxdaily.com/2017/04/28/howto-ssh-client-mac/>)
- Linux: simplemente usa el comando nativo, ej. ssh root@192.168.0.20

Usa los siguientes parámetros de conexión SSH:

- Host Name: la dirección estática que configuraste en el router, ej. 192.168.0.20
- Port: 22
- Username: root
- Password: odroid

Configuración básica

Ahora estás conectado por línea de comandos a tu propio nodo Bitcoin. En primer lugar, nos ocupamos de la configuración básica. Introduce los siguientes comandos:

```
# change root password to [password A]
$ passwd

# update the operating system
$ apt update
$ apt upgrade
$ apt dist-upgrade
$ apt install linux-image-xu3
# answer [y], then [no] (do not abort)

# install some additional software
$ apt install htop git curl bash-completion jq

# set time zone & localization
$ dpkg-reconfigure tzdata
$ dpkg-reconfigure locales
```

Cuando utilices el editor de texto Nano, puedes usar los mismos atajos de teclado para guardar (Ctrl-O, confirmar o cambiar el nombre del archivo, y presiona Intro) y para salir (Ctrl-X).

```
# change hostname (replace "odroid" with
"thundroid" :) in both files
$ nano /etc/hostname
$ nano /etc/hosts

# create user "admin", set [password A] and
```

```

make it a superuser
$ adduser admin
$ adduser admin sudo

# create user "bitcoin" and set password
[password A]
$ sudo adduser bitcoin

```

Montando el disco duro

El disco duro externo está conectado al sistema de archivos y se puede acceder a él como una carpeta normal. Esto se denomina “montaje”. Como una instalación de servidor, el sistema nativo de archivos Linux Ext4 es la mejor opción para el disco duro externo.

NOTA: ¡Todos los datos de este disco duro se borrarán con los siguientes pasos!

```

# get NAME for hard disk
$ lsblk -o UUID,NAME,FSTYPE,SIZE,LABEL,MODEL

# format hard disk (use [NAME] from above, e.g
/dev/sda1)
$ mkfs.ext4 /dev/[NAME]

# get UUID for hard disk, copy into notepad
$ lsblk -o UUID,NAME,FSTYPE,SIZE,LABEL,MODEL

# edit fstab and enter new line (replace UUID)
at the end, save & exit
$ nano /etc/fstab
UUID=123456 /mnt/hdd ext4 noexec,defaults 0 0

# create mount point, mount, check and set
owner
$ mkdir /mnt/hdd
$ mount -a
$ df /mnt/hdd
Filesystem 1K-blocks Used Available Use%
Mounted on
/dev/sda1 961300808 600388836 312057600 66%
/mnt/hdd

$ chown -R bitcoin:bitcoin /mnt/hdd/

```

Moviendo el archivo Swap

El uso de un archivo Swap puede degradar muy rápido la tarjeta SD. Por lo tanto, lo moveremos al disco duro externo.

```

# install necessary software package
$ apt install dphys-swapfile

# change configuration file to use swapfile on
external hard disk
$ nano /etc/dphys-swapfile
CONF_SWAPFILE=/mnt/hdd/swapfile

# enable new swap configuration
$ sudo dphys-swapfile setup
$ sudo dphys-swapfile swapon

# reboot, login as "admin" and delete old
swapfile
$ restart shutdown -r now
$ sudo rm /var/swap

```

Fortificando tu Thundroid

Tu Thundroid será visible desde Internet y es por ello que debe estar protegido contra posibles ataques. Un firewall controla el tráfico y cierra posibles agujeros de seguridad. Inicia sesión como “admin” (no usaremos “root” nuevamente).

UFW: Uncomplicated Firewall

El firewall deniega por defecto todos los intentos de conexión por parte de otros usuarios y permite que solo sean utilizados determinados puertos específicos.

ADVERTENCIA: La línea “ufw allow from 192.168.0.0/24 ...W que aparece a continuación asume que la dirección IP de tu Thundroid es algo así como ‘192.168.0. ???’’, siendo los signos de interrogación cualquier número de 0 al 255. Si tu dirección IP es ‘12.34.56.78’, debe cambiar esta línea por “ufw allow from 12.34.56.0/24 ...”. De lo contrario, te bloquearás a ti mismo para siempre.

```

# change session to "root"
$ sudo su
$ apt install ufw
$ ufw default deny incoming
$ ufw default allow outgoing

# make sure to use the correct subnet mask
(see warning above)
$ ufw allow from 192.168.0.0/24 to any port 22
comment 'allow SSH from local LAN'

```

```
$ ufw allow 9735 comment 'allow Lightning'
$ ufw deny 8333 comment 'deny Bitcoin mainnet'
$ ufw allow 18333 comment 'allow Bitcoin
testnet'
$ ufw enable
$ systemctl enable ufw
$ ufw status

# exit "root" session back to "admin"
$ exit
```

Fail2ban

Fail2ban monitoriza los intentos de inicio de sesión por SSH y bloquea a un interlocutor remoto durante 10 minutos tras cinco intentos de sesión fallidos. Esto hace que un ataque de fuerza bruta sea inviable, ya que simplemente llevaría demasiado tiempo.

```
$ sudo apt install fail2ban
```

Claves SSH

Una de las mejores opciones para proteger el inicio de sesión por SSH es deshabilitar completamente el inicio de sesión mediante contraseña y requerir un certificado de clave SSH. Solo alguien que posea físicamente la clave privada puede iniciar sesión.

Configura las claves SSH para el usuario "admin" siguiendo este artículo: [Configure "No Password SSH Keys Authentication" with PuTTY on Linux Servers](#)

Ahora debería contar con tres archivos creados. Guárdalos, ya que ahora vamos a desactivar el inicio de sesión por contraseña.

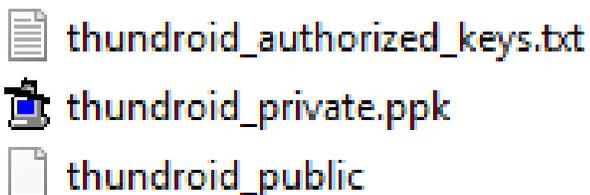


Figura 2 - Listado de archivos con las claves SSH

- Cierra sesión ('Exit') y asegúrate de que puedes iniciar sesión como "admin" con tu clave SSH
- Edita el archivo de configuración SSH

```
$ sudo nano /etc/ssh/sshd_config
```

- Cambia la configuración "ChallengeResponseAuthentication" y "PasswordAuthentication" a "no" (descomenta la línea eliminando el signo # si fuera necesario), guarda los cambios y salte.

```
GNU nano 2.5.3 File: /etc/ssh/sshd_config
# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no

^G Get Help ^C Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^L Go To Line
```

Figura 3 - configuración de SSH

```
# copy the ssh key to user "root", just in
case
$ sudo mkdir /root/.ssh
$ sudo cp /home/admin/.ssh/authorized_keys
/root/.ssh/
$ sudo chown -R root:root /root/.ssh/
$ sudo chmod -R 700 /root/.ssh/
$ sudo systemctl restart ssh

# exit and login again with your private key
$ exit
```

Ahora solo puedes iniciar sesión con "admin" o "root" y tu clave SSH. Como no puedes conectar una pantalla al ODROID, SSH es tu única opción.

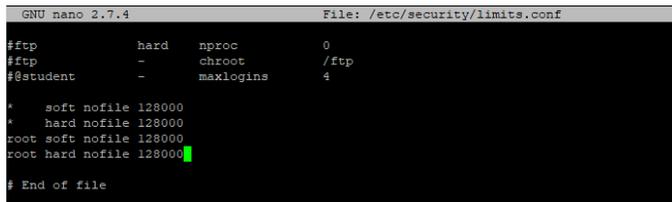
RECUERDA: ¡Haz una copia de seguridad de tu clave SSH! ¡No tienes ninguna otra alternativa para iniciar sesión! En el peor de los casos, necesitarás grabar la tarjeta MicroSD y configurar nuevamente el sistema; las cosas importantes seguirán estando en el disco duro.

Aumenta tu límite de archivos abiertos

En caso de que tu Thundroid se inunde con solicitudes de Internet, honestas o maliciosas debido a un ataque DDoS, te encontrará rápidamente con el error "can't accept connection: too many open files". Esto se debe a un límite con los archivos abiertos (que representan conexiones TCP individuales), el cual tiene un valor demasiado bajo.

Edite los siguientes tres archivos, agrega las líneas adicionales justo antes del comentario final, guarda los cambios y salte.

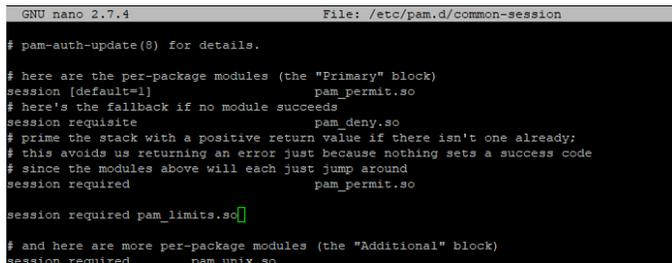
```
$ sudo nano /etc/security/limits.conf
* soft nfile 128000
* hard nfile 128000
root soft nfile 128000
root hard nfile 128000
```



```
GNU nano 2.7.4 File: /etc/security/limits.conf
#ftp      hard   nproc    0
#ftp      -      chroot   /ftp
#@student -      maxlogins 4
* soft nfile 128000
* hard nfile 128000
root soft nfile 128000
root hard nfile 128000
# End of file
```

Figura 4 - Editando pam.d/limits.conf.png

```
$ sudo nano /etc/pam.d/common-session
session required pam_limits.so
```



```
GNU nano 2.7.4 File: /etc/pam.d/common-session
# pam-auth-update(8) for details.
# here are the per-package modules (the "Primary" block)
session [default=1]      pam_permit.so
# here's the fallback if no module succeeds
session requisite       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required        pam_permit.so
session required pam_limits.so
# and here are more per-package modules (the "Additional" block)
session required        pam_unix.so
```

Figura 5

```
$ sudo nano /etc/pam.d/common-session-
noninteractive
session required pam_limits.so
```



```
GNU nano 2.7.4 File: /etc/pam.d/common-session-noninteractive
session requisite       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required        pam_permit.so
# and here are more per-package modules (the "Additional" block)
session required        pam_unix.so
session required pam_limits.so
# end of pam-auth-update config
```

Figura 6 - Editando pam.d/common-session-noninteractive

Bitcoin

La base del nodo Lightning es un nodo de Bitcoin completamente seguro (<https://bitcoin.org/en/bitcoin-core/>). Mantiene una copia completa del blockchain y valida todas las transacciones y bloques. Al hacer todo este trabajo nosotros mismos, no necesitamos que nadie más nos aporte confianza.

Al principio, usaremos TestNet Bitcoin para familiarizarnos con sus operaciones. Esta sincronización es gestionada directamente por Thundroid y no debería tardar más de unas cuantas horas. Simplemente déjalo sincronizar por la noche.

Instalación

Descargaremos el software directamente desde bitcoin.org, verificaremos su firma para asegurarnos de que usaremos una versión oficial y la instalaremos.

Inicia sesión como "admin" y crea una carpeta para la descarga:

```
$ mkdir /home/admin/download
$ cd /home/admin/download
```

Descargaremos los últimos binarios de Bitcoin Core y comparamos el archivo con la suma de verificación firmada. Esto lo hacemos simplemente por precaución para asegurarnos de que se trata de una versión oficial y no de una maliciosa que intenta robar nuestro dinero.

Hazte con los últimos enlaces de descarga en bitcoin.org/en/download. Suelen cambiar con cada actualización. Ejecuta los siguientes comandos con los nombres de archivo modificados y verifica el resultado donde se te indique.

```
# download Bitcoin Core binary
$ wget https://bitcoin.org/bin/bitcoin-core-0.16.0/bitcoin-0.16.0-arm-linux-gnueabi.tar.gz
$ wget https://bitcoin.org/bin/bitcoin-core-0.16.0/SHA256SUMS.asc
$ wget https://bitcoin.org/laanwj-releases.asc

# check that the reference checksum matches the real checksum
# (ignore the "lines are improperly formatted" warning)
$ sha256sum --check SHA256SUMS.asc --ignore-missing
> bitcoin-0.16.0-arm-linux-gnueabi.tar.gz:
OK

# manually check the fingerprint of the public key
$ gpg --with-fingerprint ./laanwj-releases.asc
> 01EA 5486 DE18 A882 D4C2 6845 90C8 019E 36C2
```

E964

```
# import the public key of Vladimir van der
Laan, verify the signed checksum file
# and check the fingerprint again in case of
malicious keys
$ gpg --import ./laanwj-releases.asc
$ gpg --verify SHA256SUMS.asc
> gpg: Good signature from Vladimir ...
> Primary key fingerprint: 01EA 5486 DE18 A882
D4C2 6845 90C8 019E 36C2 E964
```

```
admin@thundroid:~/download$ sha256sum --check SHA256SUMS.asc --ignore-missing
bitcoin-0.16.0-arm-linux-gnueabi.tar.gz: OK
sha256sum WARNING: 30 lines are improperly formatted
admin@thundroid:~/download$ gpg --with-fingerprint ./laanwj-releases.asc
gpg: directory '/home/admin/.gnupg' created
gpg: new configuration file '/home/admin/.gnupg/gpg.conf' created
gpg: WARNING: options in '/home/admin/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring '/home/admin/.gnupg/secring.gpg' created
gpg: keyring '/home/admin/.gnupg/pubring.gpg' created
pub 4096R/36C2E964 2015-06-24 Vladimir J. van der Laan (Bitcoin Core binary release signing key) <laanwj@gmail.com>
Key fingerprint = 01EA 5486 DE18 A882 D4C2 6845 90C8 019E 36C2 E964
admin@thundroid:~/download$ gpg --import ./laanwj-releases.asc
gpg: /home/admin/.gnupg/trustdb.gpg: trustdb created
gpg: key 36C2E964: public key 'Vladimir J. van der Laan (Bitcoin Core binary release signing key) <laanwj@gmail.com>' imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
gpg: no ultimately trusted keys found
admin@thundroid:~/download$ gpg --verify SHA256SUMS.asc
gpg: Signature made Mon 26 Feb 2018 08:00:19 CET using RSA key ID 36C2E964
gpg: Good signature from 'Vladimir J. van der Laan (Bitcoin Core binary release signing key) <laanwj@gmail.com>'
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
Primary key fingerprint: 01EA 5486 DE18 A882 D4C2 6845 90C8 019E 36C2 E964
admin@thundroid:~/download$
```

Figura 7 - Comandos para verificar la firma bitcoind

Extrae los binarios Bitcoin Core, instálalos y verifica la versión.

```
$ tar -xvf bitcoin-0.16.0-arm-linux-
gnueabi.tar.gz
$ sudo install -m 0755 -o root -g root -t
/usr/local/bin bitcoin-0.16.0/bin/*
$ bitcoind --version
> Bitcoin Core Daemon version v0.16.0
```

Prepara el directorio Bitcoin Core

Usamos el demonio Bitcoin, llamado "bitcoind", que se ejecuta en segundo plano sin interfaz de usuario y que almacena todos los datos en el directorio '/home/bitcoin/.bitcoin'. En lugar de crear un directorio como tal, creamos un enlace que apunte a un directorio del disco duro externo.

```
# change to user "bitcoin"
$ sudo su bitcoin

# add symbolic link that points to the
external hard drive
$ mkdir /mnt/hdd/bitcoin
$ ln -s /mnt/hdd/bitcoin
/home/bitcoin/.bitcoin

# Navigate to home directory and check the
symbolic link (the target must not be red).
$ cd
$ ls -la
```

El contenido de este directorio estará en realidad en el disco duro externo.

```
bitcoin@thundroid:~$ ls -la
total 28
drwxr-xr-x 3 bitcoin bitcoin 4096 Mai 14 23:54 .
drwxr-xr-x 4 root root 4096 Mai 5 12:37 ..
-rw-r----- 1 bitcoin bitcoin 539 Mai 14 23:53 .bash_history
-rw-r--r-- 1 bitcoin bitcoin 220 Mai 5 12:32 .bash_logout
-rw-r--r-- 1 bitcoin bitcoin 3771 Mai 5 12:32 .bashrc
lrwxrwxrwx 1 bitcoin bitcoin 16 Mai 5 15:46 .bitcoin -> /mnt/hdd/bitcoin
drwxrwxr-x 2 bitcoin bitcoin 4096 Mai 5 15:53 .nano
-rw-r--r-- 1 bitcoin bitcoin 655 Mai 5 12:32 .profile
bitcoin@thundroid:~$
```

Figura 8 - Verificando el enlace simbólico de .bitcoin

Configuración

Necesitamos crear el archivo de configuración para bitcoind. Ábrelo con Nano y pega la siguiente configuración. Guarda y Salte.

```
$ nano /home/bitcoin/.bitcoin/bitcoin.conf

# Thundroid: bitcoind configuration
# /home/bitcoin/.bitcoin/bitcoin.conf

# remove the following line to enable Bitcoin
mainnet
testnet=1

# Bitcoind options
server=1
daemon=1
txindex=1
disablewallet=1

# Connection settings
rpcuser=bitcoin
rpcpassword=PASSWORD_[B]
zmqpubrawblock=tcp://127.0.0.1:29000
zmqpubrawtx=tcp://127.0.0.1:29000

# Optimizations for limited hardware
dbcache=100
maxorphantx=10
maxmempool=50
maxconnections=40
maxuploadtarget=5000
```

NOTA: Cambia rpcpassword por una contraseña segura, de lo contrario tus fondos podrían ser robados.

Inicio automático de bitcoind

El sistema necesita ejecutar el demonio bitcoin automáticamente en segundo plano, incluso cuando nadie esté conectado. Usamos "systemd", un

demonio que controla el proceso de inicio usando los archivos de configuración.

Salte de la sesión de usuario "bitcoin" para volver al usuario "admin"

```
$ exit
```

Crea el archivo de configuración con el editor de texto Nano y copia el siguiente párrafo. Guardalo y Salte.

```
$ sudo nano
/etc/systemd/system/bitcoind.service

# Thundroid: systemd unit for bitcoind
# /etc/systemd/system/bitcoind.service

[Unit]
Description=Bitcoin daemon
After=network.target

[Service]
ExecStart=/usr/local/bin/bitcoind -daemon -
conf=/home/bitcoin/.bitcoin/bitcoin.conf -
pid=/run/bitcoind/bitcoind.pid
# Creates /run/bitcoind owned by bitcoin
RuntimeDirectory=bitcoind
User=bitcoin
Group=bitcoin
Type=forking
PIDFile=/run/bitcoind/bitcoind.pid
Restart=on-failure

# Hardening measures
#####

# Provide a private /tmp and /var/tmp.
PrivateTmp=true

# Mount /usr, /boot/ and /etc read-only for
the process.
ProtectSystem=full

# Disallow the process and all of its children
to gain
# new privileges through execve().
NoNewPrivileges=true

# Use a new /dev namespace only populated with
API pseudo devices
# such as /dev/null, /dev/zero and
/dev/random.
PrivateDevices=true
```

```
# Deny the creation of writable and executable
memory mappings.
MemoryDenyWriteExecute=true

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

Habilita el archivo de configuración:

```
$ sudo systemctl enable bitcoind.service
```

Copia 'bitcoin.conf' en el directorio de inicio del usuario "admin" para las credenciales RPC:

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

Reinicia Thundroid

```
$ sudo shutdown -r now
```

Verificación de las operaciones de bitcoind

Después de reiniciar, el bitcoind debería empezar a sincronizar y validar el blockchain de Bitcoin. Espera un poco, vuelve a conectarte a través de SSH e inicie sesión con el usuario "admin". Verifica el estado del demonio Bitcoin, el cual fue iniciado por systemd (salte con 'Ctrl-C').

```
$ systemctl status bitcoind
```

```
admin@thundroid:~$ systemctl status bitcoind
● bitcoind.service - Bitcoin daemon
   Loaded: loaded (/etc/systemd/system/bitcoind.service; enabled; vendor preset: enable
   Active: active (running) since Sun 2018-05-13 06:24:45 CEST; 1 day 17h ago
   Process: 663 ExecStart=/usr/local/bin/bitcoind -daemon -conf=/home/bitcoin/.bitcoin/b
   Main PID: 672 (bitcoind)
   CGroup: /system.slice/bitcoind.service
           └─672 /usr/local/bin/bitcoind -daemon -conf=/home/bitcoin/.bitcoin/bitcoin.
admin@thundroid:~$
```

Figura 9 - Estado de Bitcoind

Pueder ver el bitcoind en acción monitorizando su registro log (salte con 'Ctrl-C'):

```
$ sudo tail -f
/home/bitcoin/.bitcoin/testnet3/debug.log
```

Utiliza el cliente Bitcoin Core 'bitcoin-cli' para obtener información sobre el blockchain actual:

```
$ bitcoin-cli getblockchaininfo
```

NOTA: Cuando "Bitcoind" se esté iniciando, puede que recibas un mensaje de error "verifying blocks". Es normal, simplemente espere unos minutos.

Entre mucha otra información, aparecerá el “verificationprogress”. Una vez que este valor alcance casi 1 (0,999 ...), el blockchain está actualizado y completamente validado.

Lightning Network

Descargaremos e instalaremos el LND (Lightning Network Daemon) desde Lightning Labs <http://lightning.engineering/>. Visita su repositorio Github

(<https://github.com/lightningnetwork/lnd/blob/master/README.md>) para obtener abundante información sobre su proyecto de código abierto y Lightning en general.

Instalar LND

Ahora viene lo bueno: descarga, verifica e instala los archivos binarios LND.

```
$ cd /home/admin/download
$ wget
https://github.com/lightningnetwork/lnd/releases/download/v0.4.1-beta/lnd-linux-arm-v0.4.1-beta.tar.gz
$ wget
https://github.com/lightningnetwork/lnd/releases/download/v0.4.1-beta/manifest-v0.4.1-beta.txt
$ wget
https://github.com/lightningnetwork/lnd/releases/download/v0.4.1-beta/manifest-v0.4.1-beta.txt.sig
$ wget
https://keybase.io/roasbeef/pgp_keys.asc

$ sha256sum --check manifest-v0.4.1-beta.txt --ignore-missing
> lnd-linux-arm-v0.4-beta.tar.gz: OK

$ gpg ./pgp_keys.asc
> pub 4096R/DD637C21 2017-09-12 Olaoluwa Osuntokun <laolu32@gmail.com>
> sub 4096R/5FA079A1 2017-09-12 [expires: 2021-09-12]
> 65317176B6857F98834EDBE8964EA263DD637C21

$ gpg --import ./pgp_keys.asc
$ gpg --verify manifest-v0.4.1-beta.txt.sig
> gpg: Good signature from "Olaoluwa Osuntokun <laolu32@gmail.com>" [unknown]
```

```
> Primary key fingerprint: 6531 7176 B685 7F98
834E DBE8 964E A263 DD63 7C21
```

```
admin@thundroid:~/download$ sha256sum --check manifest-v0.4.1-beta.txt --ignore-missing
lnd-linux-arm-v0.4.1-beta.tar.gz: OK
admin@thundroid:~/download$ gpg ./pgp_keys.asc
pub 4096R/DD637C21 2017-09-12 Olaoluwa Osuntokun <laolu32@gmail.com>
sub 4096R/5FA079A1 2017-09-12 [expires: 2021-09-12]
admin@thundroid:~/download$ gpg --import ./pgp_keys.asc
gpg: key DD637C21: "Olaoluwa Osuntokun <laolu32@gmail.com>" not changed
gpg: Total number processed: 1
gpg:      unchanged: 1
admin@thundroid:~/download$ gpg --verify manifest-v0.4.1-beta.txt.sig
gpg: assuming signed data in 'manifest-v0.4.1-beta.txt'
gpg: Signature made Die 03 Apr 2018 02:27:09 CEST using RSA key ID DD637C21
gpg: Good signature from "Olaoluwa Osuntokun <laolu32@gmail.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:       There is no indication that the signature belongs to the owner.
Primary Key fingerprint: 6531 7176 B685 7F98 834E DBE8 964E A263 DD63 7C21
admin@thundroid:~/download$
```

Figura 10 – Suma de verificación LND

```
$ tar -xzf lnd-linux-arm-v0.4.1-beta.tar.gz
$ ls -la
$ sudo install -m 0755 -o root -g root -t
/usr/local/bin lnd-linux-arm-v0.4.1-beta/*
$ lnd --version
> lnd version 0.4.1-beta commit=
```

Configurar LND

Ahora que LND esta instalado lo configuraremos, debemos configurarlo para que funcione con Bitcoin Core y se ejecute automáticamente al inicio.

Abre una sesión de usuario “bitcoin”:

```
$ sudo su bitcoin
```

Crea el directorio de trabajo LND y el enlace simbólico correspondiente:

```
$ mkdir /mnt/hdd/lnd
$ ln -s /mnt/hdd/lnd /home/bitcoin/.lnd
$ cd
$ ls -la
```

```
bitcoin@thundroid:~$ ls -la
total 28
drwxr-xr-x 3 bitcoin bitcoin 4096 Mai  5 22:00 .
drwxr-xr-x 4 root    root    4096 Mai  5 12:37 ..
-rw-r----- 1 bitcoin bitcoin 518 Mai  9 20:31 .bash_history
-rw-r----- 1 bitcoin bitcoin 220 Mai  5 12:32 .bash_logout
-rw-r----- 1 bitcoin bitcoin 3771 Mai  5 12:32 .bashrc
lrwxrwxrwx 1 bitcoin bitcoin  16 Mai  5 15:46 .bitcoin -> /mnt/hdd/bitcoin
lrwxrwxrwx 1 bitcoin bitcoin  12 Mai  5 22:00 .lnd -> /mnt/hdd/lnd
drwxrwxr-x 2 bitcoin bitcoin 4096 Mai  5 15:53 .nano
-rw-r--r-- 1 bitcoin bitcoin 685 Mai  5 12:32 .profile
bitcoin@thundroid:~$
```

Figura 11 – Comprobando enlace simbólico LND

Crea el archivo de configuración LND y pega el siguiente contenido (ajústalo a tu alias). Guardalo y Salte.

```
'$ nano /home/bitcoin/.lnd/lnd.conf'

# Thundroid: lnd configuration
# /home/bitcoin/.lnd/lnd.conf

[Application Options]
```

```

debuglevel=info
debugtlc=true
maxpendingchannels=5
alias=YOUR_NAME [LND]
color=#68F442

[Bitcoin]
bitcoin.active=1

# enable either testnet or mainnet
bitcoin.testnet=1
#bitcoin.mainnet=1

bitcoin.node=bitcoind

[autopilot]
autopilot.active=1
autopilot.maxchannels=5
autopilot.allocation=0.6

```

```

Type=simple
KillMode=process
TimeoutSec=180
Restart=always
RestartSec=60

[Install]
WantedBy=multi-user.target

```

Activa e inicia LND

```

$ sudo systemctl enable lnd
$ sudo systemctl start lnd
$ systemctl status lnd

```

Monitoriza el registro log de LND en tiempo real (salte con 'Ctrl-C')

```

$ sudo journalctl -f -u lnd

```

Información Adicional

```

admin@thundroid:~$ sudo journalctl -u lnd -f
-- Logs begin at Tue 2018-05-15 00:10:07 CEST. --
Mai 15 20:29:49 thundroid systemd[1]: Stopped LND Lightning Daemon.
Mai 15 20:29:09 thundroid systemd[1]: Started LND Lightning Daemon.
Mai 15 20:29:09 thundroid lnd[15557]: Attempting automatic RPC configuration to bitcoind
Mai 15 20:29:09 thundroid lnd[15557]: Automatically obtained bitcoind's RPC credentials
Mai 15 20:29:09 thundroid lnd[15557]: 2018-05-15 20:29:09.242 [INF] LNDR: Version 0.4.1-beta commit=
Mai 15 20:29:09 thundroid lnd[15557]: 2018-05-15 20:29:09.242 [INF] LNDR: Active chain: Bitcoin (network=testnet)
Mai 15 20:29:09 thundroid lnd[15557]: 2018-05-15 20:29:09.243 [INF] CHDR: Checking for schema updates: latest_version=0, db_version=0
Mai 15 20:29:09 thundroid lnd[15557]: 2018-05-15 20:29:09.983 [INF] RPCS: password RPC server listening on 127.0.0.1:111009
Mai 15 20:29:09 thundroid lnd[15557]: 2018-05-15 20:29:09.983 [INF] RPCS: password gRPC proxy started at 127.0.0.1:10880
Mai 15 20:29:09 thundroid lnd[15557]: 2018-05-15 20:29:09.984 [INF] LNDR: Waiting for wallet encryption password. Use 'lncli create
to create wallet, or 'lncli unlock' to unlock already created wallet.

```

Figura 12 - sample-lnd.conf

Dentro del repositorio del proyecto LND:

Salte de la sesión de usuario "bitcoin" para volver a "admin"

```

$ exit

```

Crea una unidad systemd LND con el siguiente contenido. Guarda y Salte.

```

$ sudo nano /etc/systemd/system/lnd.service

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

[Unit]
Description=LND Lightning Daemon
Wants=bitcoind.service
After=bitcoind.service

[Service]
ExecStart=/usr/local/bin/lnd
PIDFile=/home/bitcoin/.lnd/lnd.pid
User=bitcoin
Group=bitcoin
LimitNOFILE=128000

```

Configurar la billetera LND

Una vez que LND se inicie, el proceso nos espera para que creamos la billetera Bitcoin integrada. No usa la billetera bitcoind.

Inicia una sesión con el usuario "bitcoin"

```

$ sudo su bitcoin

```

Crear la billetera LND

```

$ lncli create

```

Si quieres crear una nueva billetera, introduce tu 'password [C]' como contraseña de la billetera, selecciona 'n' en relación a una semilla existente e introduce el 'password [D]' opcional como contraseña de la semilla. Se creará una nueva semilla de cifrado que consta de 24 palabras.

```

Input wallet password:
Confirm wallet password:

Do you have an existing cipher seed mnemonic you want to use? (Enter y/n): n
Your cipher seed can optionally be encrypted.
Input your passphrase you wish to encrypt it (or press enter to proceed without a cipher seed passphrase):
Generating fresh cipher seed...

!!!YOU MUST WRITE DOWN THIS SEED TO BE ABLE TO RESTORE THE WALLET!!!

-----BEGIN LND CIPHER SEED-----
1. about 2. jungle 3. vocal 4.
5. nuclear 6. flavor 7. matter 8.
9. situate 10. decade 11. cart 12.
13. track 14. puzzle 15. lounge 16.
17. begin 18. wallet 19. volume 20.
21. scatter 22. consider 23. rubber 24.
-----END LND CIPHER SEED-----

!!!YOU MUST WRITE DOWN THIS SEED TO BE ABLE TO RESTORE THE WALLET!!!

lnd successfully initialized!

```

Figura 13 - Nueva semilla de cifrado LND

Estas 24 palabras, combinadas con tu contraseña ('password [D]' opcional) es todo lo que necesitas para restaurar tu billetera Bitcoin y todos los canales

Lightning. El estado actual de tus canales, sin embargo, no pueden crearse de nuevo desde esta semilla, ya que esto aún se encuentra en fase de desarrollo para LND.

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

Salte de la sesión de usuario "bitcoin"

```
$ exit
```

Asignar permisos LND a "admin"

Comprueba si se han creado los archivos de permiso "admin.macaroon" y "readonly.macaroon". De lo contrario, consulta el problema nº 890 abierto de LND (<https://github.com/lightningnetwork/lnd/issues/890>).

```
$ ls -la /home/bitcoin/.lnd/
```

```
admin@thundroid:~$ ls -la /home/bitcoin/.lnd/
total 40
drwxrwxr-x 4 bitcoin bitcoin 4096 Mai  9 20:31 .
drwxr-xr-x 13 bitcoin bitcoin 4096 Mai  5 15:58 ..
-rw-r----- 1 bitcoin bitcoin 232 Mai  9 20:31 admin.macaroon
drwx----- 4 bitcoin bitcoin 4096 Mai  9 20:31 data
-rw-r----- 1 bitcoin bitcoin 115 Mai  9 20:31 invoice.macaroon
-rw-rw-r-- 1 bitcoin bitcoin 379 Mai  5 22:03 lnd.conf
drwx----- 3 bitcoin bitcoin 4096 Mai  9 20:29 logs
-rw-r----- 1 bitcoin bitcoin 183 Mai  9 20:31 readonly.macaroon
-rw-r----- 1 bitcoin bitcoin 704 Mai  9 20:29 tls.cert
-rw----- 1 bitcoin bitcoin 227 Mai  9 20:29 tls.key
admin@thundroid:~$
```

Figura 14 - verificación

Copia los archivos de permiso y el certificado TLS al usuario "admin" para usar "lncli".

```
$ mkdir /home/admin/.lnd
$ sudo cp /home/bitcoin/.lnd/tls.cert
/home/admin/.lnd
$ sudo cp /home/bitcoin/.lnd/admin.macaroon
/home/admin/.lnd
$ sudo chown -R admin:admin /home/admin/.lnd/
```

Asegúrate de que 'lncli' funciona desbloqueando tu billetera (introduce el 'password [C]') y aparecerá cierta información del nodo.

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

Monitoriza el progreso de arranque de LND hasta que alcance el blockchain testnet (aproximadamente 1,3 millones de bloques en este momento). Esto puede tardar hasta 2 horas. Después, verás pasar una parrafada muy rápido. Salte con 'Ctrl-C'.

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

Conseguir algo de bitcoin testnet

Ahora tu nodo Lightning está listo. Para usarlo en testnet, puedes conseguir un bitcoin testnet gratis desde un faucet.

Genera una nueva dirección Bitcoin para recibir fondos en cadena

```
$ lncli newaddress np2wkh
```

```
'> "address":
"2NCoq9q7.....dkuca5LzPXnJ9NQ"
```

Conseguir bitcoin testnet:

```
<https://testnet.manu.backend.hamburg/faucet>
```

Comprueba el saldo de tu billetera LND.

```
$ lncli walletbalance
```

Monitoriza tu transacción (el faucet muestra el ID de TX) en un explorador de Blockchain:

```
<https://testnet.smartbit.com.au>
```

LND en acción

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

Recíbete a ti mismo una solicitud de pago en StarBlocks (<https://starblocks.acinq.co/#/>) o Y'all's (<https://yalls.org/>) y mueve algunas monedas.

Algunos comandos que puedes intentar:

Listar todos los argumentos de la interfaz de línea de comando (cli)

```
$ lncli
```

Obtener ayuda de un argumento específico

```
$ lncli help [ARGUMENT]
```

Conocer algunas estadísticas generales sobre tu nodo:

```
$ lncli getinfo
```

Conéctase a un nodo compañero (puede encontrar algunos nodos para conectarte aquí:

<https://1ml.com/>)

```
$ lncli connect [NODE_URI]
```

Verifica los nodos a los que está conectado actualmente:

```
$ lncli listpeers
```

Abrir un canal con un nodo/compañero:

```
$ lncli openchannel [NODE_PUBKEY]
[AMOUNT_IN_SATOSHIS] 0
```

Ten en cuenta que [NODE_URI] incluye @IP:PORT al final, mientras que [NODE_PUBKEY] no.

Verifica el estado de tus canales pendientes:

```
$ lncli pendingchannels
```

Verifica el estado de tus canales activos:

```
$ lncli listchannels
```

Antes de pagar una factura, debes decodificarla para verificar si la cantidad y la información que contiene son correctos:

```
$ lncli decodepayreq [INVOICE]
```

Paga una factura:

```
$ lncli payinvoice [INVOICE]
```

Verifica los pagos que has enviado:

```
$ lncli listpayments
```

Crea una factura:

```
$ lncli addinvoice [AMOUNT_IN_SATOSHIS]
```

Listar todas las facturas:

```
$ lncli listinvoices
```

Para cerrar un canal, necesitas los dos argumentos siguientes que se pueden determinar con 'listchannels' y que aparecen listados como "channelpoint": 'FUNDING_TXID': 'OUTPUT_INDEX'.

```
$ lncli listchannels
$ lncli closechannel [FUNDING_TXID]
[OUTPUT_INDEX]
```

Para forzar el cierre de un canal (si tu compañero no está conectado o no coopera), usa

```
$ lncli closechannel --force [FUNDING_TXID]
[OUTPUT_INDEX]
```

Consulta la referencia de la API Lightning (<http://api.lightning.community/>) para obtener información adicional

Preparar el mainnet Bitcoin

En la segunda parte de esta guía moveremos el nodo Thundroid Bitcoin & Lightning al mainnet Bitcoin, que usa un blockchain diferente. Al igual que el pequeño blockchain testnet, la blockchain mainnet registra todas las transacciones de Bitcoin y básicamente define quién tiene bitcoins y cuántos tiene. Esta es la información más importante y no debemos confiar en que otra persona nos proporcione esta información. Para configurar nuestro Nodo completo Bitcoin en mainnet, necesitamos:

- Descargar todo el blockchain (~ 200 GB)
- Verificar cada transacción de Bitcoin que haya tenido lugar y cada bloque que se haya minado
- Crear una base de datos índice para todas las transacciones, de modo que podamos consultarla más adelante
- Calcular todos los saldos de direcciones de bitcoin (llamado el conjunto UTXO)

Consulta "Running a Full Node" (<https://bitcoin.org/en/full-node>) para obtener información adicional.

Como te puedes imaginar Thundroid no está a la altura de esta gran tarea. La descarga no es el problema, la cuestión está en que procesar

inicialmente todo el blockchain llevaría varias semanas debido a las restricciones de recursos. Necesitamos descargar y verificar el blockchain con Bitcoin Core en un ordenador normal y luego transferir los datos al Thundroid. Esto solo se hace una vez. Después, Thundroid puede mantenerse al día con los nuevos bloques.

Para cambiar a mainnet, el blockchain mainnet debería estar listo, de esta manera empezáramos con esta tarea.

Usando un ordenador normal

Suponemos que utilizaras una ordeandor con Windows para esta tarea, aunque puedes realizarla con la mayoría de sistemas operativos. Necesitas tener aproximadamente 250 GB de espacio libre en disco, internamente o en un disco duro externo. Como la indexación crea un tráfico pesado de lectura/escritura, cuanto más rápido sea tu disco duro, mejor. Una unidad interna o un disco duro externo USB3 será significativamente más rápido que uno disco con conexión USB2.

Descargar y verificar Bitcoin Core

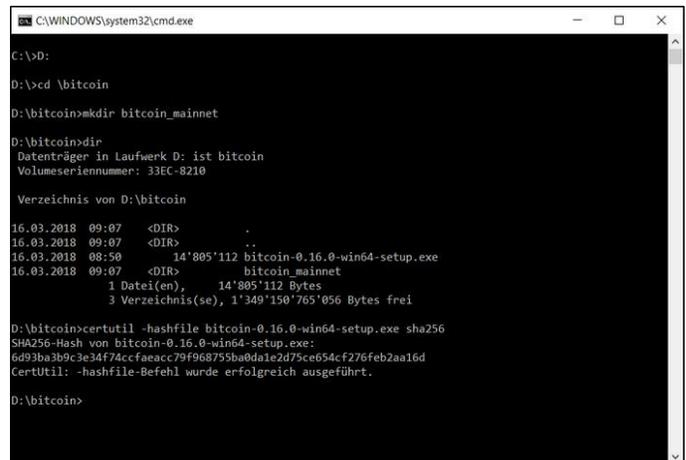
Descarga el instalador Bitcoin Core desde bitcoin.org/download y guárdalo en el directorio que deseas utilizar para descargar el blockchain. Para verificar la autenticidad del programa, calcula su suma de verificación y compárala con las sumas de verificación proporcionadas.

En Windows, antepondré a todos los comandos que necesitamos introducir el signo ">", de modo que con el comando "> cd bitcoin", simplemente introduce "cd bitcoin" y presiona Intro.

Abre el prompt de comandos de Windows ('Win + R', introduce 'cmd', pulsa 'Intro'), navega hasta el directorio bitcoin (en mi caso, está en la unidad 'D:', comprobada con el Explorador de Windows) y crea el nuevo directorio 'bitcoin_mainnet'. Luego calcula la suma de verificación del programa descargado.

```
> G:
> cd itcoin
> mkdir bitcoin_mainnet
> dir
> certutil -hashfile bitcoin-0.16.0-win64-
```

```
setup.exe sha256
6d93ba3b9c3e34f74ccfaeacc79f968755ba0da1e2d75c
e654cf276feb2aa16d
```



(Figura 15 – Prompt de comandos de Windows: comprobando la suma de verificación)

Puedes comparar estas sumas de verificación con las sumas de verificación de referencia en tu Thundroid desde el archivo que hemos descargado anteriormente y que ya ha verificado tu autenticidad. Compara el siguiente resultado con la suma de verificación de tu descarga de Bitcoin Core para Windows.

```
# on Thundroid, with user "admin"
$ cat /home/admin/download/SHA256SUMS.asc |
grep win

7558249b04527d7d0bf2663f9cfe76d6c5f83ae90e5132
41f94fda6151396a29 bitcoin-0.16.0-win32-
setup.exe
60d65d6e57f42164e1c04bb5bb65156d87f0433825a1c1
f1f5f6aebf5c8df424 bitcoin-0.16.0-win32.zip
6d93ba3b9c3e34f74ccfaeacc79f968755ba0da1e2d75c
e654cf276feb2aa16d bitcoin-0.16.0-win64-
setup.exe
42706da1a95b2db8c5808529f73c2063a0dd770f71e0c8
506bfa86dc0f3403ef bitcoin-0.16.0-win64.zip
```

Instalando de Bitcoin Core

Ejecuta el archivo de instalación de Bitcoin Core. Es posible que tengas que hacer clic derecho y elegir "Ejecutar como administrador". Instálalo usando la configuración por defecto. Inicia el programa 'bitcoin-qt.exe' localizado en el directorio "C:\Program Files\Bitcoin". Elige tu nueva carpeta "bitcoin_mainnet" como directorio de datos personalizado.

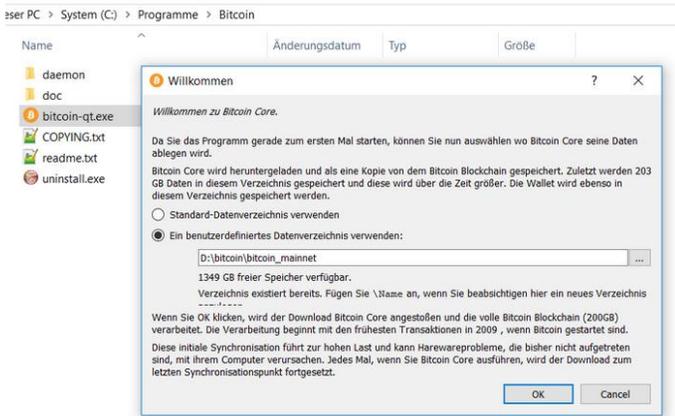


Figura 16 – Seleccionando el directorio de Bitcoin Core

Bitcoin Core abre y empieza a sincronizar inmediatamente el blockchain. Ahora, necesitamos aplicar una configuración adicional muy importante en el archivo “bitcoin.conf”. Si no se aplica, todo el blockchain será inútil y necesitará ser revalidado. Usando el menú, abre “Settings”/“Options” y haz clic en el botón “Open Configuration File”. Introduce la siguiente línea:

```
$ txindex=1
```

Si tu ordenador tiene mucha memoria, puedes aumentar la caché en memoria de la base de datos añadiendo la siguiente línea (con los megabytes de memoria que se van a usar, ajustándolo a tu ordenador):

```
$ dbcache=6000
```

Guarda y cierra el archivo de texto, salte de Bitcoin Core usando “File”/“Exit” y reinicia el programa. El programa empezará a sincronizar nuevamente.

Deja que el blockchain se sincronice, esto le llevará un día o dos.

Antes de pasar a mainnet

En la parte 2 de esta guía, haremos la transición al mainnet Bitcoin. Este será un punto sin retorno. Hasta aquí, puedes empezar de nuevo. Experimenta con testnet bitcoin. Abre y cierra canales en el testnet. Es importante que te sientas seguro a la hora de realizar operaciones con Thundroid, antes de poner dinero real en línea.

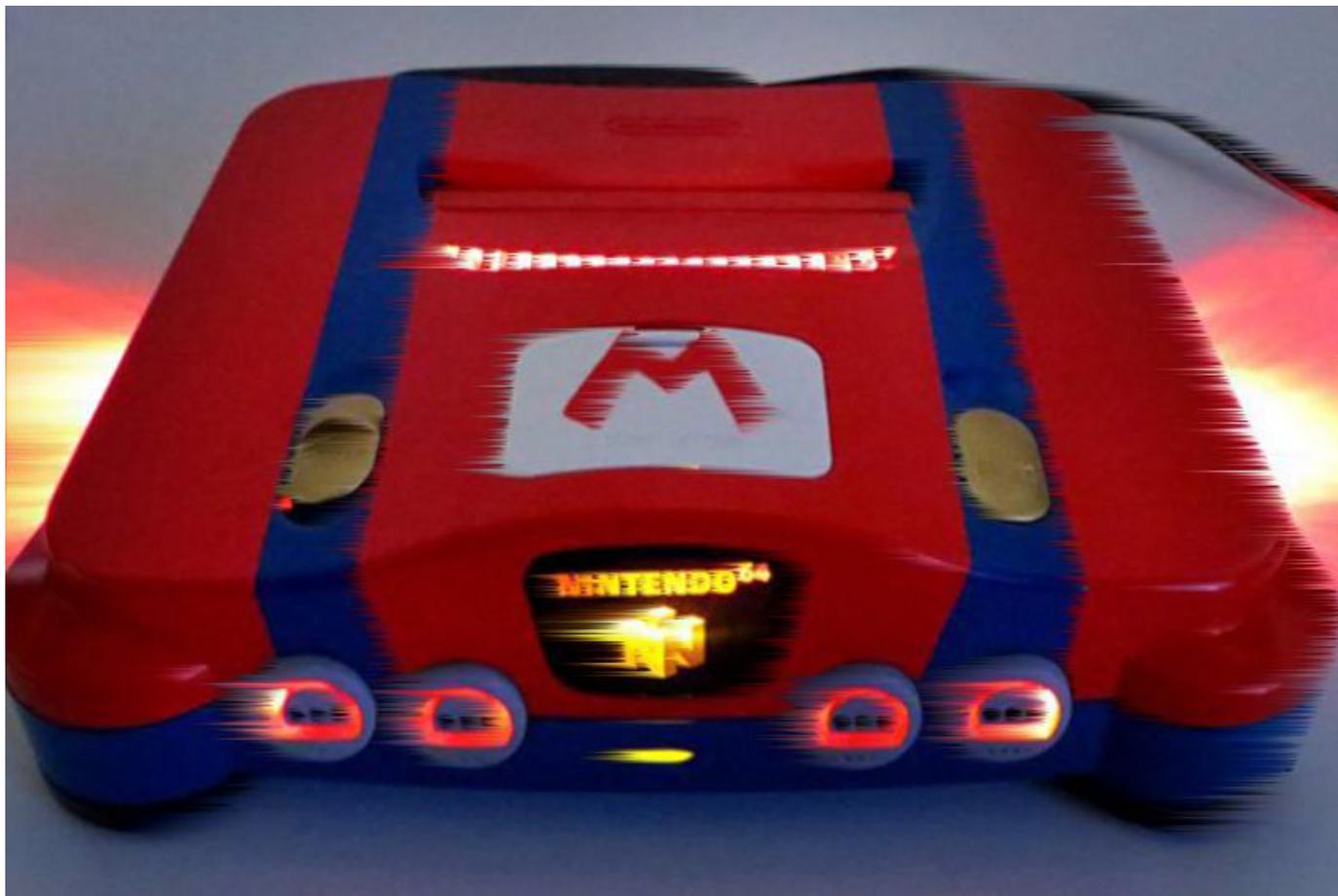
Una vez que cambies a mainnet y envíes Bitcoin real a tu Thundroid, tienes “entras en el juego”.

- Asegúrate de que tu RaspiBolt está funcionando según lo previsto. Práctica un poco con “bitcoin-cli” y sus opciones. Consulta la documentación de Bitcoin Core RPC (<https://bitcoin-rpc.github.io/>)
- Realiza ejecuciones simuladas con “Incli” y sus muchas opciones. Consulta la referencia de la API de Lightning (<http://api.lightning.community/>)
- Prueba algunos reinicios (“sudo shutdown -r now”). ¿Todo se inicia bien?

Nos vemos pronto en la parte 2, “El perfecto nodo Bitcoin Lightning”.

Juegos Linux: Emulación Nintendo 64 – Parte 1

© June 1, 2018 By Tobias Schaaf ↗ Juegos, Linux



La emulación de la Nintendo 64 ha evolucionado recientemente permitiendo que se pueda ejecutar en todos los dispositivos ODROID, utilizando el emulador independiente Mupen64plus o el núcleo Libretro para Retroarch. Ahora que está mucho más accesible, he decidido hacer una comparativa no solo de la versión independiente y el núcleo Libretro, sino también de las diferentes plataformas ODROID, para evaluar su potencial a la hora de emular la consola Nintendo 64 (N64). Ten en cuenta que este artículo sólo se centra en la emulación con Linux, y no tiene en cuenta para nada Android, aunque existen varios emuladores de Nintendo 64 disponibles para Android, como Mupen64plus y N64oid.

Información general

Me llevó un tiempo lograr que la emulación N64 funcionara en todas las placas ODROID bajo Linux. Sin embargo, ahora que lo he conseguido, es bastante divertido y abre muchas posibilidades a la hora de

ejecutar los clásicos juegos de N64. Esperemos que en el futuro veamos más mejoras y dispongamos de un mejor soporte para los emuladores N64 en los dispositivos ODROID bajo Linux. Por ahora, tenemos algunas restricciones. Solo el XU3 puede usar el núcleo Libretro bajo Linux, que tiene mejores gráficos y es más fácil de controlar que el emulador independiente Mupen64plus. Mupen64plus se ejecuta en el resto de dispositivos ODROID, como es en la serie Exynos 4 (X, X2, U2 y U3), así como en el más reciente, aunque menos potente ODROID-C1. Ambas versiones ofrecen diferentes plugins y métodos para ejecutar los juegos.

Plugins gráficos

Ya sea usando Mupen64plus o el núcleo Libretro, se utilizan diferentes plugins para mostrar los gráficos de los juegos. Mupen64plus puede usar un plugin de video llamado rice, y otro llamado glide64mk2. El núcleo Libretro ofrece rice, glide64 y otro llamado

gln64. Mientras los probaba, descubrí que el mejor videocore depende del propio juego. Sin embargo, parece que glide ofrece mejores funciones gráficas que el plugin rice, pero presenta algunos fallos menores que no están presentes en el plugin de video rice. Usando el independiente Mupen64plus, rice es incapaz de escalar la relación de aspecto y siempre ajusta el juego al tamaño máximo de tu resolución de video. Esto distorsiona la imagen, haciendo que los personajes y objetos se muestren más grandes de lo normal. El ODROID-C1 funciona mejor cuando se utiliza el plugin de video Rice, ya que glide64mk2 no funciona a menos que la profundidad de color se reduzca a 16 bits, lo cual hace que los efectos de transparencia queden desactivados. Esto también causa problemas si intentas ver películas o quiere iniciar otras aplicaciones que requieran más de 16 bits. Puesto que las pruebas iniciales con el C1 no tuvieron mucho éxito, decidí volver a probar cada juego en 16 bits utilizando el núcleo de video glide64mk2. Parece ser que hay una solución usando controladores framebuffer en lugar de controladores X11 y recurriendo a algunos scripts que permiten cambiar la resolución y la profundidad de color, pero como mi ODROID GameStation Turbo usa drivers X11 por defecto, no dedique tiempo a realizar pruebas con framebuffer. El plugin glide en los dispositivos de la serie Exynos 4 (X, X2, U2 y U3) funciona bien y respeta la relación de aspecto con una buena calidad en general, aunque puede llegar a ser un poco más lento que rice en determinados juegos. Glide, además, parece reproducir la imagen un tanto más oscura de lo que lo hace rice, lo cual probablemente se deba a que faltan algunas opciones de sombreado. El plugin glide64mk2 en los dispositivos Exynos 4 es el plugin de gráficos preferido para el independiente Mupen64plus. El XU3 puede usar rice, glide64 y gln64, aunque glide64 parece ser la mejor opción para ODROID en este momento.

Mandos

Afortunadamente los Joysticks funcionan muy bien en todos los dispositivos ODROID, lo que significa que todos los emuladores (mupen64plus y libretro) son totalmente compatibles con cualquier mando para

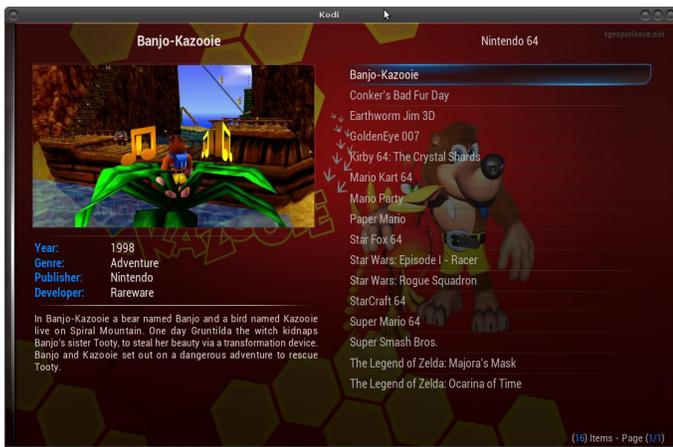
jugar. El emulador Mupen64plus configura los mandos automáticamente, pero no todos los mandos funcionan a la perfección con la configuración por defecto. Gracias a Retroarch en el XU3, puede configurar cualquier mando asignando manualmente los botones, de esta forma cualquier mando puede ser 100% compatible. Normalmente, deberías poder activar el soporte para la vibración en los mandos, aunque yo tuve problemas para habilitarlo con todos los emuladores y mandos. Pude usarla con algunos mandos del estilo de la PS3 en el emulador independiente Mupen64plus, pero no pude activar la vibración con el núcleo Libretro.

Sonido

El sonido funciona bastante bien en prácticamente todos los emuladores, no he encontrado ningún problema importante con él. Aunque si es verdad que un juego presentaba retardos en el sonido, lo que provocaba que los efectos no estuvieran sincronizados con la acción mostrada en pantalla. Ésta fue una excepción y no lo he vuelto a ver en ningún otro juego.

Selección de juegos

¿Estás listo para jugar a tus juegos favoritos de Nintendo 64 en ODROID? Bueno, eso es exactamente lo que queremos probar y descubrir: ¿Funcionan tus juegos favoritos? Para dar respuesta a esa pregunta, he realizado una búsqueda sobre los que, en líneas generales se consideran los mejores juegos de N64. Luego he seleccionado algunos de ellos para probarlos, tal y como se muestra en las Figuras 1 y 2. Espero que encuentres algunos de tus juegos favoritos en esta lista.



Figuras 1 - Lista de algunos de los juegos de N64 de diferentes géneros más preferidos

Banjo-Kazooie

Banjo-Kazooie es un juego de plataformas que mezcla las carreras y los saltos con la aventura y la acción. Juegas como el oso Banjo tratando de salvar a tu hermana que ha sido secuestrada por una bruja. Tiene un amigo Kazooie the Pajaro, con quien necesitas resolver un par de acertijos. Como la mayoría de los juegos de Rareware, es bastante divertido y tiene un estilo un tanto cómico.



Figura 2 - Banjo Kazooie)

U3

En general, el juego se ejecuta de forma aceptable en el U3. A veces es un poco lento, especialmente durante la intro. Las intro de los juegos Rare son normalmente bastante largas y no se pueden omitir. En el XU3 y el núcleo Libretto, es posible aumentar la velocidad del emulador, de modo que puedes hacer que la introducción avance más rápido. Aún no he visto esta opción en el emulador independiente Mupen64plus, lo cual significa que tienes que esperar un tiempo. Además, las fuentes no son las correctas

en el emulador independiente, lo cual es un poco molesto. El juego se vuelve algo lento después de estar jugándolo un tiempo. Utilicé las opciones de configuración del emulador para activar el frameskipping con un máximo de tres fotogramas, lo cual aumentó considerablemente la velocidad del mismo. Con esta configuración, nos encontramos ante un juego realmente bueno, con el único problema de las fuentes que he mencionado. Decidí dejar la opción frameskipping configurada en tres fotogramas para el resto de juegos.

C1 - plugin rice

Para el C1, utilicé el emulador independiente Mupen64plus, usando el plugin rice, ya que no quería cambiar los colores a 16 bits. Además, rice es algo más rápido que glide64mk2, y se adapta mejor al C1. También tuve que activar frameskip con el plugin rice para que el juego funcionase sin problemas. Sin frameskipping, el sonido presenta retardo y la experiencia no es muy buena. Aunque rice no respeta la relación de aspecto, no se ve del todo mal. Los problemas con las fuentes que aparecen con glide64mk2 no existen en rice, de modo que las fuentes parecen ser las normales. Con el frameskip activado, el juego se ejecuta sorprendentemente rápido en el C1, lo cual me sorprendió. Si el juego fuera capaz de soportar una adecuada relación de aspecto, estaríamos ante un juego que se ejecutaría a la perfección sobre el C1

C1 - glide64mk2

Banjo-Kazooie se ejecuta en el ODRROID-C1 usando glide64mk2, pero es extremadamente lento y no resulta divertido. El plugin Rice con la profundidad del color configurada a 32 bits funciona mucho mejor que glide64mk2 a 16 bits.

XU3

XU3 usa el núcleo Libretto de Mupen64plus y Retroarch para emular el juego, y la experiencia en el XU3 es la mejor de las tres plataformas. El emulador ejecuta glide2gl como plugin de video, que parece ser mucho mejor que el anterior glide64mk2, y no reproduce los colores tan oscuros. El núcleo Libretto no tiene la característica del frameskip que ofrece el

emulador Mupen64plus, lo cual significa que únicamente puede funcionar tan rápido como lo sea la propia placa en la que se ejecuta, pudiéndose observar algunas ralentizaciones, dependiendo de las escenas. En Banjo-Kazooie, esto sucede en la introducción, pero no es de lo peor. Los gráficos se ven mucho mejor con Libretro, siendo el juego totalmente jugable.

Conker's Bad Fur Day

Conker's Bad Fur Day es otro juego de Rareware similar a Banjo-Kazooie. Sin embargo, no es adecuado para niños pequeños debido a sus continuas referencias a las drogas y al alcohol junto con un lenguaje un tanto rudo, a pesar de su estilo cómico. También encontrarás que algunos personajes son iguales en ambos juegos. Este juego es una mezcla de muchos géneros, en su mayor parte saltar-correr y aventura de acción, pero da la sensación de ser más bien un shooter en primera persona con mezcla de otros géneros. El juego es, de hecho, uno de mis favoritos para Nintendo 64, y algunas personas le otorgan muy buenas críticas: <http://bit.ly/1bo6odW>. Muy recomendable para adultos y adolescentes.



Figura 3 - Conker's Bad Fur Day

U3

El U3 con el emulador independiente Mupen64plus también es un poco lento para Conker's Bad Fur Day, hay escenas en las que se percibe un cierto retardo, lo cual afecta a los controles. A veces éstos reaccionan muy lento, lo cual es bastante molesto durante los puzzles de salto. El plugin glide64mk2 hace que los

gráficos sean muy oscuros, especialmente durante las escenas del castillo. Cuando te encuentran en habitaciones oscuras, casi todo se ve negro en determinados lugares.

U3 - plugin rice

Mientras que el U3 tiene problemas de velocidad, el C1 falla casi por completo porque éste simplemente no es lo suficientemente potente como para ejecutar un juego tan exigente como Conker's Bad Fur Day. El plugin de gráficos rice también tiene muchos problemas con este juego, bordes negros y gráficos distorsionados, no resultan muy agradables. Aunque el juego en líneas generales funciona, es bastante lento. Algunas escenas son de hecho lo suficientemente rápidas para ser consideradas como jugables, pero están lejos de alcanzar la máxima velocidad. Por todo ello, considero que este juego es impracticable sobre el ODROID-C1.

U3 - glide64mk2

Conker funciona mejor con el plugin glide64mk2 que con el plugin rice. No llega a la velocidad máxima, pero si puedes tolerar un poco de retardo, es jugable.

XU3

El XU3 es el que ofrece mejor experiencia de juego cuando ejecutamos Conker's Bad Fur Day. El juego, aunque no se ejecuta a toda velocidad, es más uniforme en el XU3. Con el plugin glide2gl se ve muy bien y solo presenta unos cuantos problemas. No he llegado muy lejos en el juego ahora mismo, de modo que no puedo comparar el rendimiento de los niveles posteriores, especialmente a la hora de conducir tanques y hacer uso de los modos de francotirador.

Earthworm Jim 3D

Earthworm Jim es un buen shooter de acción de plataformas sobre un gusano llamado Jim, que ha sido alcanzado por un traje espacial avanzado, transformándolo en un héroe. Mientras que el juego fue en primer lugar un gran éxito en la SNES, SEGA Genesis e incluso en la Playstation 1, con la N64, se dio un paso más transformando el juego desde una plataforma 2D en un juego de acción en 3D.



Figura 4 - Earthworm Jim 3D

U3

La experiencia de juego del Earthworm Jim 3D en todos los ODROID es muy agradable. El U3 ejecuta el emulador muy rápido y con fluidez, con algunos fallos gráficos menores ocasionales. Puesto que el juego es bastante colorido y brillante con recintos iluminados sin sombras ni rincones oscuros, la oscuridad gráfica del glide64mk2 que afecta a otros juegos no se aprecia al jugar Earthworm Jim 3D, lo cual mejora realmente la experiencia del juego.

C1 - plugin rice

Aunque la intro y la demo del juego son rápidos, no pude empezar a jugar. La primera escena en la que hablas con uno de tus amigos no es solo lenta, sino que la ventana que se supone que muestra el texto permanece vacía y no reproduce nada. Los botones tampoco responden, de modo que te encuentras atascado incluso antes de empezar el juego. Es probable que esto sea solo un error con rice, pero como no he probado glide64mk2 en el C1, no puedo afirmar que funcione bajo este plugin. Por lo tanto, solo puedo decir que Earthworm Jim no funciona en el C1.

C1 - glide64mk2

Este juego funciona muy bien con el plugin glide64mk2, sin problemas gráficos ni ralentizaciones. Todo funciona como era de esperar.

XU3

Como el C1 con glide64mk2 y el U3 son lo suficientemente potentes como para jugar con fluidez, no nos sorprenderá que la experiencia de juego en el XU3 también sea perfecta. Si te gustan los buenos juegos de plataforma de acción, definitivamente te tienes que hacer con éste, aunque presenta algunos fallos menores. Algunos de los objetos que puedes recoger no se muestran correctamente. Parecen estar demasiado altos y a menudo solo ves una sombra donde se supone que deberían estar los objetos. Aun así, puedes recogerlos, simplemente son invisibles, aparte de eso el juego funciona a la perfección.

GoldenEye 007

GoldenEye 007 se encuentra en la lista de los 10 mejores juegos del N64, ya que el juego supuso una auténtica revolución por aquel entonces. No solo ofrece buenos gráficos, sino que también era conocido por su impresionante modo multijugador. La historia y las misiones de un jugador también son muy emocionantes y también muy divertidas. GoldenEye es un shooter en primera persona bastante formal con el toque adecuado de agente secreto. Aunque no es tan espectacular como Cate Archer, James Bond lucha o se cuela a través de diferentes niveles y tiene que defenderse de los guardias y espías enemigos. Sin embargo, Cate Archer siempre será mi espía favorito en situaciones de riesgo. Aunque este juego tiene muy buenas críticas para el N64, realmente no me gustan demasiado los shooters en primera persona para las consolas. Por consiguiente, no es uno de mis juegos favoritos, aunque he de reconocer que es bastante divertido.



Figura 5 - GoldenEye 007

U3

El juego funciona muy bien en el U3. A excepción de la breve escena de la introducción, no he observado ralentizaciones ni dentro ni fuera de los edificios. Tuve algunos problemas a la hora de usar el mando inalámbrico Xbox 360 con el stick analógico correcto, lo cual dificultaba el movimiento del juego. Sin embargo, si usamos únicamente el stick analógico izquierdo parece ser que es la mejor opción a la hora de jugar.

C1 - plugin rice

El C1 también tiene problemas gráficos con el complemento rice en este juego. Ni el logotipo, ni la intro son visibles, ambos se muestran ocultos tras un marco negro. La escena que causaba una ralentización masiva a 8 fps en el U3 es demasiado para el C1, y el emulador se detiene por completo y finalmente, se bloquea. Observando el ODR0ID mientras ejecutaba el juego, me di cuenta de que cuando aparece la ralentización, el uso de la RAM se dispara hasta el punto de agotarse toda la RAM disponible. Después, usa la partición swap y a continuación, el memory killer finaliza el emulador, lo cual no sucede en el U3. Entonces decidí cambiar al plugin rice en el U3, y aunque el emulador era mucho más lento que con glide64mk2, funcionaba correctamente sin que aparecieran pantallas negras o problemas de memoria, no llegando a bloquearse. Por lo tanto, llegué a la conclusión de que simplemente se trataba de un problema con el propio C1.

C1 - glide64mk2

El juego funciona con glide64mk2, pero la velocidad varía siendo muy lento en algunos casos y alcanzado máxima velocidad en otros. Se puede jugar, pero no tan bien como en el U3 o XU3.

XU3

El juego se ejecuta una velocidad decente en el XU3 usando el núcleo Libretto. Los gráficos se ven muy bien, pero presentan recesos en algunas ocasionales, aunque no de una forma que impida jugar. También pude usar el mando Xbox 360 sin ningún problema.

Kirby 64: The Crystal Shards

¿Quién no reconoce al pequeño Kirby? Este juego es muy adecuado para niños y presenta gráficos muy graciosos. El amiguito de color rosa tipo marshmallow puede succionar sus enemigos y tragárselos para absorber sus poderes. La versión N64 tiene magníficos gráficos en 3D y es bastante fácil de jugar, lo cual lo hace perfecto para niños. Aunque el juego se muestra en 3D, los niveles son muy lineales. Puedes ir a la izquierda, derecha, arriba y abajo, pero no puede caminar libremente por el mapa, lo que probablemente redujo en gran medida el tamaño del mapa y permitió asignar un mejor rendimiento a los efectos. El juego es similar a Mario 64, pero sin movimiento libre en todas las direcciones.



Figura 6 - Kirby 64: The Crystal Shards

U3

La experiencia en general es bastante buena, y el juego funciona perfectamente a máxima velocidad. Sin embargo, presenta algunos fallos gráficos con el suelo y las sombras que lo hacen parpadear en algunas situaciones. También tuve ciertos problemas con los controles. Tuve que usar un mando diferente, ya que mi mando Xbox 360 no funcionaría con este juego. Parece que el movimiento solo funciona con el D-Pad, que no está disponible en el joystick Xbox 360. Tras cambiar a un mando "Thrustmaster Dual Trigger 3 en 1", que es similar a un mando PS3, los controles de movimiento empezaron a funcionar y se podía jugar sin problemas al margen de los fallos técnicos que he mencionado anteriormente. La experiencia del juego es realmente suave y divertida. Los gráficos son adorables y las escenas realmente encajan perfectamente en este tipo de juegos.

C1 - plugin rice

La experiencia con el C1 fue diferente a la del U3, era inesperadamente lenta. Las escenas eran tan lentas que decidí omitirlas, en lugar de esperar a que se reprodujeran. Sin embargo, los fallos gráficos habían desaparecido. La jugabilidad era más lenta de lo esperado, con un cierto desfase en algunas escenas, mientras que, en otras, apenas se podía percibir. Esperaba que funcionase mejor en el C1.

C1 - glide64mk2

Kirby funciona bien con glide64mk2, con algunos problemas menores con las sombras y las texturas del suelo.

Marco de Fotos Digital: Pantalla 4K de 55 Pulgadas Para Fotos Digitales por unos 400\$

© June 1, 2018 By @tung256 ➤ ODROID-C2, Mecaniquero



Existen muchos tutoriales sobre cómo hacer un buen marco de fotos digital con una Raspberry Pi. Lamentablemente, la Raspberry Pi no es compatible con resoluciones 4K. El ODROID-C2 puede manejar fácilmente resoluciones 4K, pero ninguno de estos tutoriales de Raspberry Pi funcionan para la unidad C2. Me llevó más de 30 horas llegar a donde estoy hoy. Si quieres puedes comprar uno en Memento (mementosmartframe.com) por 900\$ de 35 pulgadas 4K, o un televisor Samsung por alrededor de 1300\$.



Figura 1 - Esta pared ha estado vacía durante tres años

Deseaba tener un bonito póster o una pantalla retroiluminada, pero éstas suelen ser muy costosas y solo pueden mostrar una foto al mismo tiempo. Tras una corta espera, observé que había disponible un televisor 4K de 55 pulgadas en Walmart por 260\$. Añadí 26\$ para la garantía de 3 años y los impuestos, el coste final ascendió a 306\$.



Figura 2 - Añadir una toma de corriente cerca del televisor para que diera un aspecto más limpio

Logre pescar un cable del calibre 12 Romex de una toma de corriente cercana. ¡Asegúrate de apagar la electricidad antes de realizar esta tarea! Hay tutoriales muy detallados disponibles en YouTube para instalar una toma de luz. De hecho, olvidé tomar fotos y videos mientras instalaba la mía. ¡Lo siento!

Reúne todos los materiales que necesitarás

- TV 4K de 55 pulgadas de Walmart, 260\$ a 30 de abril de 2018.
- ODROID-C2 con adaptador de corriente en forma de cilindro, 65\$ con envío de 4 días en EE. UU. No use el cable micro USB para la alimentación, no es lo suficientemente bueno. Compré mi unidad y cable de alimentación en Ameridroid.com.
- Cable HDMI de alta velocidad de Monoprice.com, 5\$. Certificado para trabajar con 4K 60hz y con escala cromática 4:4:4.
- Unidad flash de 32GB para guardar fotos, 15\$.
- Tarjeta microSD de 8GB, 4\$.
- Combo teclado y USB Wireless, 30\$.
- Adaptador WiFi USB opcional, 10\$.

Todos los precios están sujetos a cambios. Mis costes fueron en realidad más bajos porque disponía de algunos elementos a mano procedentes de otros proyectos.

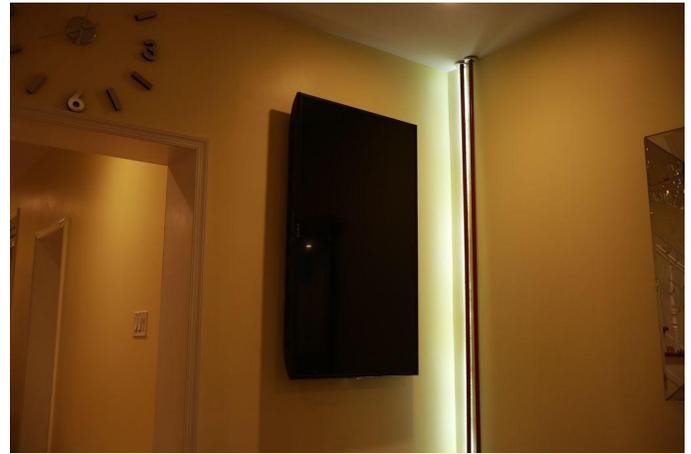


Figura 3 - El marco de fotos digital ODROID-C2 montado en la pared

Instala Ubuntu mate de https://wiki.odroid.com/odroid-c2/os_images/ubuntu/v2.4. Graba la imagen ISO descargada en la tarjeta microSD usando la herramienta Win32disk. Inserta tu tarjeta microSD en el ODROID-C2. Conecta el cable HDMI desde C2 al televisor. Enciéndelo y asegúrate de que todos los elementos estén conectados al ODROID-C2. Puedes omitir la conexión de la unidad USB y del adaptador Wifi USB por ahora. Deja que Ubuntu se inicie y finalice el proceso. Una vez hecho, te pedirá que inicie sesión.

User ID: odroid Password: odroid

Activa el inicio de sesión automático para que no tenga que iniciar sesión manualmente cada vez que enciendas el C2. Dirígete al menú Applications/System Tools/Mate Terminal y escribe:

```
$ sudo nano
/usr/share/lightdm/lightdm.conf.d/60-lightdm-gtk-greeter.conf
```

Si te solicita una contraseña, escriba "odroid". Escribe la línea de inicio de sesión automático. El archivo final debería verse así:

```
[Seat:*] greeter-session=lightdm-gtk-greeter
autologin-user=odroid
```

Presiona CTRL + X para salir, o "Y" para guardar el archivo. Instala FEH para ver fotos. En la misma ventana de terminal, escribe los siguientes comandos, presionando "Y" para confirmar, si te pregunta:

```
$ sudo apt-get install feh
```

Mis fotos están editadas con una resolución 4K a 3840 x 2160 píxeles. Copia tus fotos en una unidad USB y conéctala al ODROID-C2. Usa tu ratón para navegar al directorio /home/odroid y haz clic derecho para crear un documento vacío. Yo lo llamé pixx.sh, pero puedes ponerle el nombre que quieras. Abre pixx.sh y añade lo siguiente dentro:

```
$ sleep 15

$ feh --quiet --fullscreen --borderless --
hide-pointer --randomize --slideshow-delay 30
/media/odroid/38C1-602E/*
```

El nombre de tu unidad USB será diferente al mío. En mi caso, la unidad se llama "38C1-602E". Para localizar el nombre de tu USB, simplemente navega a media/odroid y lo verás. Cambia el valor de retardo de la presentación de diapositivas de 30 segundos a lo que quieras. Guarda el archivo y ciérralo. Haz clic derecho en pixx.sh para ver sus propiedades. Hazlo "executable" con una de las opciones.

Añade pixx.sh al menú de inicio automático. Dirígete al menú System/Preferences/Personal/Startup Applications y haz clic en Add.

- Name=slideshow
- Command=(Selecciona el archivo pixx.sh desde donde lo tengas guardado)
- Comment= slideshow autostart, haz clic en Add y salte

Desactiva el protector de pantalla dirigiéndote a System/Control Center y eligiendo Look and Feel, luego ScreenSaver. Desactiva "Activate Screensaver" y cualquier otra cosa que pueda activar el modo reposo. He olvidado todos los parámetros, aunque todos están aquí. Vuelve a la ventana del terminal y escribe:

```
$ sudo reboot
```

Esto reiniciará el C2. Una vez reiniciado, deberías iniciar sesión automáticamente, espera 15 segundos y luego empezará a mostrar las fotos almacenadas en la unidad USB. Presione ESC en tu teclado para salir

de FEH si lo necesitas. Yo dejo el mío ejecutándose todo el tiempo. Solo apago el televisor si lo necesito.

Pasos opcionales

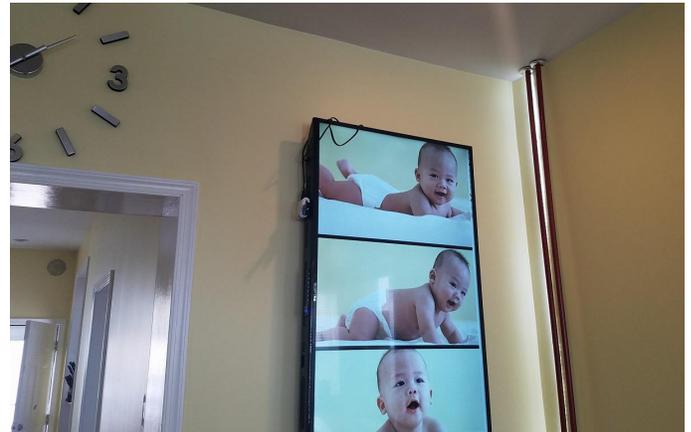


Figura 4 – El marco de fotos completamente montado mostrando fotos familiares

Añade más fotos a la unidad flash USB copiando y pegando en la unidad USB manualmente o a través de FTP. Para cargar via FTP y así no tener que desconectar físicamente la unidad USB, asegúrate de que el adaptador USB WiFi esté enchufado. Ve a la esquina superior derecha del menú de Ubuntu y conéctate a tu red WiFi.

Descargue FileZilla y conéctate al ODROID a través del protocolo SFTP, no por el protocolo FTP. Introduce la dirección IP de tu C2 en el campo host:

user: odroid password: odroid login type: normal

Sube los nuevos archivos a tu dispositivo media/odroid/usb y reinicia C2 para que FEH pueda cargar las nuevas fotos en la memoria. Para girar la pantalla de visualización a modo vertical, ve al terminal y escribe:

```
$ sudo nano /etc/X11/xorg.conf
```

Añade dentro esta línea:

```
Option "Rotate" "CCW"
```

De modo que el resultado final se verá así:

```
Section "Device"
Identifier "Mali FBDEV"
Driver "fbturbo"
Option "fbdev" "/dev/fb0"
Option "Rotate" "CCW"
```

```
Option "SwapbuffersWait" "true"  
EndSection
```

Salte y guarda, luego reinicia el ordenador:

```
$ sudo reboot
```

Una vez que el televisor se haya iniciado nuevamente, la pantalla debería girarse al modo vertical.

Solución de problemas

- Ubuntu se reinicia solo No hay suficiente potencia. No enciendas la unidad C2 a través del puerto micro USB. Enciende la unidad C2 a través del enchufe en forma

SO Destacado: ODROID GameStation Turbo

© June 1, 2018 By Tobias Schaaf ↗ Juegos, ODROID-C2, ODROID-XU4

OS Spotlight: GameStation



Uno de los proyectos más destacados en los que estoy trabajando para la comunidad ODROID es la imagen ODROID GameStation Turbo, que funciona como interfaz para juegos y reproducción multimedia. Está pensada para ser un sistema de entretenimiento que te permita controlar tu ODROID con solo usar un mando de juegos sin tener que tocar el teclado para ver películas, escuchar música o jugar a tus juegos favoritos. Para comprender mejor la posible utilidad de la imagen, quisiera proporcionarte una visión interna de cómo llegue a crear la imagen, qué me motivó a hacerla y cómo puedes adaptarla para satisfacer tus propias necesidades.

Motivación

El primer dispositivo basado en tecnología ARM que tuve en cuenta fue en realidad Open Pandora, pero cuando estuve listo para comprar uno, no estaba disponible. No obstante, incluso cuando volvió a estar disponible, era tan caro que no me lo podía permitir.

Finalmente, cuando logré reunir el suficiente dinero, me volví un poco escéptico y decidí buscar otras opciones. La placa Pandora es un dispositivo con un único núcleo ARM de solo 1GHz y con tan solo 512MB de RAM por 700\$, ¿Realmente merece la pena? Bueno, aunque la comunidad era y sigue siendo tremenda, y es un dispositivo totalmente portátil (como una Nintendo DS), en mi opinión, era demasiado caro para lo que podía hacer. En ese momento, había mejores dispositivos disponibles, incluido el ODROID. Después de ver el ODROID-X2 en un artículo en una página alemana de noticias TI, realmente me enganchó. Cuando se anunció Ubuntu para ODROID, me compré un X2. Sin embargo, para lo que quería Pandora que era para ejecutar juegos, ODROID no tenía demasiados juegos por aquel entonces (2012). A medida que mis sobrinos se hacían mayores, pensé que podría hacer algo realmente bueno para ellos que evolucionase con ellos a medida que se hacían mayores. En primer lugar, el ODROID

podría ser una consola que ejecutase juegos y más tarde funcionar como un PC para hacer tareas escolares y aprender Linux. Básicamente este fue mi objetivo y motivación para crear la imagen.

Pasos hacia el éxito

El primer paso para lograr mi objetivo era generar contenido, de modo que tuve que exportar muchos juegos y emuladores a ODROID. Si echas un vistazo a la sección de Ubuntu (All Linuxes) de ODROID Forum, encontrarás allí muchos juegos y programas que he exportado por mi cuenta. Me supuso bastante esfuerzo, ya que pasé de no saber prácticamente nada sobre como exportar juegos y compilar software en Linux a tener amplios conocimientos de ello. ODROID fue de gran ayuda a la hora de aprender nuevas habilidades y mejorar aprendiendo con este tipo de cosas. Ahora, sé cómo optimizar ciertos programas, cómo aplicar diferentes indicadores de optimización y cuándo son necesarios aplicar esos indicadores. Aprendí más sobre cómo funcionan las CPU ARM y, especialmente, sobre los diferentes hardware de las placas de Hardkernel. Mi primer proyecto fue exportar muchos juegos y compilar también algunos emuladores. Si ha leído mis artículos publicados en anteriores números de ODROID Magazine, encontrarás mucha información sobre qué juegos pueden ejecutarse realmente en ODROID, la lista sigue creciendo. El siguiente gran paso fue hacer que fuese sencillo, incluso para niños, usar y trabajar con Linux y jugar a juegos. Empecé a montar Gamestation Turbo partiendo de la imagen Linaro Ubuntu 12.04. La escogí sobre el resto de imágenes de sistemas operativos porque incorporaba Unity Desktop. Unity es fácil de usar y entender incluso para personas que nunca antes hayan usado Linux. Puede que no sea el mejor entorno de escritorio para todas las aplicaciones, pero es colorido y fácil de manejar. Para alguien que nunca haya usado Linux antes, es una muy buena forma de empezar con Linux. Lo primero que aborde fue intentar hacer que los programas fueran lo suficientemente fáciles de ejecutar en todo el sistema. Proporcioné a todas las aplicaciones y juegos que creé un archivo de icono .desktop, para que puedas localizarlo en Unity o simplemente colocar un acceso directo en el

escritorio. Esto funcionaba muy bien con los juegos, pero no para los emuladores, ya que los emuladores normalmente usan su propia interfaz de administrador de archivos para cargar los ROM. Aunque los adultos podrían ser capaces de iniciar manualmente todos los juegos, los niños no tendrán ni idea de lo que significaría determinadas expresiones, y resultaba un tanto complicado ver qué juegos están disponibles o qué buscar en Unity. Inmediatamente me quedó claro que necesitaba algún tipo de interfaz para iniciar los juegos. Ya había usado XBMC en una PC antiguo que funcionaba como PC de cine en Casa (HTPC), y posteriormente descubrí un buen complemento XBMC llamado Rom Collection Browser (RCB). RCB te permite organizar las ROM del emulador de la misma forma que organizas tu colección de videos. Incluso puede descargar imágenes de vista previa y portadas, y proporcionar una breve descripción de los juegos, algo similar a los servicios de video para las películas.

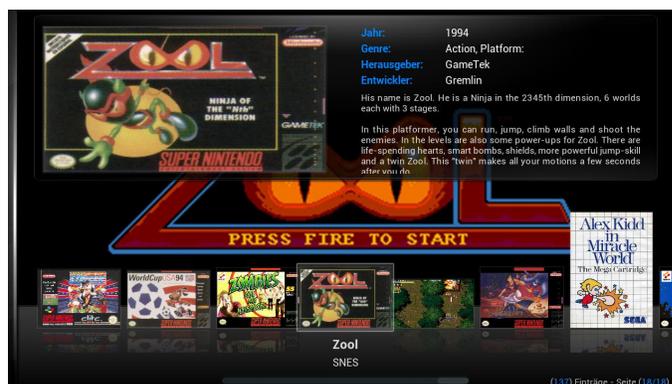


Figura 1 – Rom Collection Browser en ODROID GameStation Turbo

Sabiendo esto, me surgió la idea de utilizar XBMC como frontend y configurarlo para que los niños jugaran y se divirtieran o, mejor dicho, configurarlo de forma que incluso un niño pudiera jugar con él. Por aquel entonces, la reproducción de video y XBMC acelerados por hardware estaban fuera de onda ya que su desarrollo aún no se había completado. La versión XBMC que venía con Ubuntu 12.04 era XBMC 11 (Eden), que funcionaba, pero no muy rápido debido a la decodificación por software. Aunque el menú funcionaba sin problemas, la reproducción de video no era muy fluida. Aun así, funcionaba lo suficientemente bien como para poder probar Rom Collection Browser, y comencé experimentar como

configurarlo todo. Cuando salió la primera imagen de XBMC 12 (Frodo) para ODROID, todavía no era compatible con la reproducción de películas acelerada por hardware, pero sí que soportaba OpenGL ES 2. Las cosas se volvieron un tanto difíciles llegado a este punto, ya que la compilación del código fuente de XBMC de Hardkernel no me funcionaba en mi caso, la versión que ofrecían no tenía soporte para joystick, lo cual era algo que consideraba crucial en mis planes. Decidí que, ya que estaba pensada para ser una plataforma de juegos, la reproducción de video no iba ser una característica tan importante, aun así, podía reproducir sin problemas todo lo que no era HD, siempre y cuando fuese 720p o menos. A los niños, normalmente no les importa demasiado si su anime o historietita favorita está en HD o solo en SD. Pues bien, fue en ese momento cuando se lanzó una imagen de XBMC acelerada por hardware que funcionaba, y fui capaz de reconstruir la imagen con el soporte de joystick necesario. Poco antes de lanzar la primera versión de GameStation Turbo, me pase a una versión XBMC completamente funcional.

Componentes que fueron incluidos

Después de decidir cómo debería funcionar la imagen, era importante unir todas las piezas pequeñas en una imagen muy bien empaquetada, y para ello, se necesitaba realizar algunas tareas y contar con diferentes tipos de programas para lograr lo que realmente quería hacer. La máxima prioridad la tenía el sistema operativo, tenía que ser muy estable, fácil de mantener y disponer de una interfaz que le fuese familiar a muchas personas. Las únicas alternativas eran Ubuntu 12.04 y Debian Wheezy. Cualquiera otra imagen o era inestable (Debian Jessie / Sid) o tenía soporte por poco tiempo (Ubuntu 13.04 o la recién lanzado 13.10). Ubuntu 12.04 es una versión LTS que dispone de soporte hasta 2017, lo cual siempre es bueno, sin embargo, Debian Wheezy tenía mejor rendimiento que Ubuntu 12.04. También descubrí mientras desarrollaba Debian Wheezy que los programas tenían más posibilidades de ejecutarse en Ubuntu 12.04 y versiones posteriores sin ningún problema, pero no al revés. Así que, decidí usar Debian Wheezy y LXDE, que usa menos de 150 MB de RAM incluso con XBMC y un par de programas

ejecutándose. Después de esto, era cuestión de reunir el tipo de software adecuado para convertir el ODROID en una máquina de juegos.

Rom Collection Browser

Utilicé Rom Collection Browser como base para instalar diferentes tipos de emuladores como Retroarch, Mednafen, PPSSPP y ScummVM. Una vez completada la configuración básica, resultó que no todo funcionaba con un gamepad con simplemente conectarlo, así que decidí incluir antimicro, que es capaz de asignar determinadas teclas a los botones del joystick con el fin de cubrir las lagunas donde los drivers del joystick no funcionaban. También mantengo mis propias compilaciones del kernel e incluyo los archivos cabecera, ya que algunas partes del núcleo proporcionadas por Hardkernel no satisfacían mis necesidades, no estando incluidos los archivos cabecera. Además de esto, había una enorme diferencia de espacio entre los módulos del kernel de hardkernel y el que yo produje. El tamaño de la compilación de Hardkernel era de más de 300 MB, pero la mía era tan solo 16 MB, que simplemente se lograba depurando los módulos. Mis scripts también permiten a los usuarios instalar o desinstalar paquetes del kernel, en lugar de simplemente copiar el kernel directamente sobre los archivos existentes.

Complicaciones

El mayor problema con el que me encontré fue el de lograr que todas las piezas funcionasen en conjunto, y facilitar a la gente el uso de la imagen, incluso si no tienen conocimientos de cómo configura las cosas. El Rom Collection Browser era un tanto difícil de usar para un principiante, ya que tenías que elegir el emulador, poner en marcha los parámetros y proporcionar la extensión estándar de los archivos ROM para configurarlo y ejecutarlo. Así que tuve que idear un sistema que facilitase al usuario esta cuestión. Había otro problema. Quería tener el control total de Joystick (GamePad), pero algunos emuladores necesitaban además otras teclas, como Retroarch y Mednafen que requerían la tecla ESC para finalizar el juego actual y volver a XBMC, y también los juegos MAME que necesitaban introducir un "OK" para continuar.

Configuración

Un problema con la pre-configuración del Rom Collection Browser era que necesita la ruta completa del lugar donde se encuentran el emulador y las ROM, y qué extensión de archivo se utiliza para buscar las ROMs. Esto puede ser bastante complicado para alguien que nunca ha trabajado con Rom Collection Browser. Es por eso que pre-seleccioné el emulador y los juegos, y creé una estructura de carpetas donde deberían colocarse las ROM, para que Rom Collection Browser pudiera localizar los juegos. Se pueden agregar emuladores adicionales presionando la tecla C dentro de Rom Collection Browser y seleccionando "Add a new ROM collection". Se debe indicar la ruta del emulador, la ruta de las ROMs, la ruta donde se debe almacenar la información, las imágenes y la extensión de las ROMs para poder localizarlas. El archivo de configuración de Rom Collection Browser se almacena en

```
/home/odroid/.xbmc/userdata/addon_data/script.  
game.rom.collection.browser/config.xml
```

Al editar este archivo, también puedes cambiar otras opciones, como si se debe extraer un archivo .zip en una carpeta temporal y si se debe buscar una ROM dentro de un archivo .zip (que, por ejemplo, debe desactivarse para los juegos MAME). Si tienes suficiente experiencia, incluso puedes añadir nuevas colecciones directamente en este archivo.

Iniciando un emulador

Aunque iniciar una ROM directamente a través del emulador sin duda alguna funciona, tiene un par de desventajas. En primer lugar, XBMC seguirá ejecutándose en segundo plano utilizando algunos recursos que podrían ser necesarios para mejorar la experiencia de juego. En segundo lugar, como he mencionado anteriormente, algunos emuladores necesitan teclas adicionales que no están asignadas a los botones. Si usa un joystick que no es compatible, necesitas antimicro para mapear los botones. Si es así, debes asegurarte de que el antimicro se inicia cuando realmente lo necesites, lo que podría no ser tan siempre como parece. Iniciar directamente antimicro junto con el emulador tampoco funciona.

Para resolver estos y otros problemas, dejé que XBMC ejecutara un pequeño script en lugar de iniciar directamente el emulador. En ese script, se inicia el emulador, que luego ejecuta el archivo ROM que es ofrecido al script como un parámetro de línea de comandos desde XBMC. De esta forma, puedo definir los diferentes pasos que me aseguren que el emulador se ejecutará de la mejor forma posible.

```
Example: running an SNES game with Retroarch:  
#!/bin/sh  
/usr/bin/killall -STOP xbmc.bin  
if [ `ps aux | grep antimicro | grep -v grep |  
wc -l` -lt 1 ]; then  
antimicro --tray --profile  
/home/odroid/joydev.xml &  
else  
/usr/bin/killall -CONT antimicro  
fi  
/usr/local/bin/retroarch -L  
/usr/local/share/retroarch/cores/working/snes9  
x_next_libretro.so "$1"  
/usr/bin/killall -CONT xbmc.bin  
/usr/bin/killall -STOP antimicro
```

Si analizas el código anterior, podrás ver que XBMC está configurado para que entre en modo suspensión, lo cual significa que no usará capacidad de procesamiento alguna mientras ejecutamos nuestro emulador. Después de esto, realizo una comprobación para ver si se está ejecutando antimicro y, o bien lo cargo con el archivo de perfil requerido, o lo reanudo en caso de que todavía esté en ejecución. A continuación, llamo al emulador en sí. Aquí puedo pasar parámetros de comando que me permiten configurar el emulador. Una vez que finalice el emulador al salir del mismo, se reanuda XBMC y se suspende antimicro. Justo después de que el script complete todo el proceso, el control vuelve a XBMC. Esto nos permite realizar algunos trabajos de limpieza que quizás sean necesarios. Escribí varios scripts para poder adaptarme a las diferentes circunstancias. Por ejemplo, el script ScummVM y Amiga es un poco más complicado; pero en general es más o menos lo mismo.

- Suspende los procesos que no necesitas (por ejemplo, XBMC)

- Configura tu entorno preparando el sistema con las cosas que necesitas (por ejemplo, cargar antimicro con el perfil correcto)
- Llama al emulador y le proporcionamos los parámetros que crees que vas a necesitar. El "\$1" representa el archivo ROM que XBMC pasa como parámetro.
- Realiza un trabajo de limpieza y reanuda los procesos que suspendiste anteriormente

Todos los scripts que utilicé para iniciar los emuladores se encuentran en `/usr/local/bin/`, donde puedes revisar, mejorar o añadir tus propios scripts.

FAQ

De vez en cuando, recibo algunas preguntas sobre mi imagen que me gustaría responder a continuación a modo de Preguntas frecuentes.

¿Dónde tengo que poner los archivos ROM para mis juegos? Dirígete a `/home/odroid/ROMS`, donde encontrarás una estructura de carpetas ya creada para cada tipo de ROM que quieras ejecutar, como GBA y SNES. Consulta el post del foro <http://bit.ly/1nVvQqz> para obtener más detalles sobre qué extensiones de archivos son compatibles.

¿Hay alguna forma de cargar ROM desde un almacenamiento externo? Copia los contenidos de `/home/odroid/ROMS` al dispositivo de almacenamiento externo y luego monta automáticamente el dispositivo externo en `/home/odroid/ROMS` agregándolo a `/etc/fstab` o usando `/etc/rc.local` para hacerlo permanente.

¿Qué joystick/gamepad son compatibles? Creé la imagen para utilizarla con un mando inalámbrico Xbox 360 y un receptor USB inalámbrico Xbox 360. De modo que, si tiene este hardware, la imagen debería funcionar de tirón sin tener que hacer ninguna modificación, a menos que haya olvidado algo. Además, cualquier joystick/gamepad compatible con Linux debería funcionar también, pero tendrás que adaptar la configuración a tu dispositivo. Por lo tanto, deberás cambiar la configuración del joystick en los emuladores de forma individual. Al ejecutar Mednafen, puedes simplemente presionar ALT + MAYÚS + 1 para configurar los controles de tu dispositivo. El programa de configuración es fácil de

entender. El segundo jugador, si lo soporta el emulador, se puede configurar con ALT + MAYÚS + 2, y así sucesivamente. Para Retroarch, es un poco más complicado. Salte de XBMC, abre un terminal y escribe `retroarch-joyconfig`, luego sigue las instrucciones de la pantalla. Al final, obtendrás una larga lista de parámetros de configuración en la ventana de Terminal. Copia esta lista, luego abre el archivo `/home/odroid/.config/retroarch/retroarch.cfg`, donde encontrarás los mismos parámetros de la lista. Reemplaza los parámetros existentes con los que obtuviste con `retroach-joyconfig`, y tu dispositivo debería funcionar en retroarch.

Desafortunadamente, XBMC no admite muchos dispositivos con soporte de joystick. Aunque los mandos de la Xbox 360 funcionan muy bien, otros simplemente no hacen nada. Con PPSSPP, puedes cambiar la configuración del mando dentro del emulador simplemente dirigiéndote al menú. Sin embargo, en PPSSPP, la forma en que se implementan los mandos es bastante lenta, de modo que sólo unos cuantos llegan a funcionar realmente bien. En el peor de los casos, ya no podrás usar el teclado puesto que la configuración del mando no te permitirá presionar ciertas teclas de dirección. Si esto te sucede, elimina el archivo `/home/odroid/.config/ppsspp/PSP/SYSTEM/controls.in` i y empieza de nuevo. Si todo lo demás falla, elimina todas las asignaciones para el mando y mantén la configuración solo para el Teclado, que siempre debería funcionar. A continuación, le toca el turno a tu mejor amigo, ¡antimicro! Si usas un dispositivo de joystick diferente y realmente tiene problemas para hacerlo funcionar con PPSSPP o XBMC, Antimicro te servirá de mucha ayuda. Usando antimicro, puedes asignar los comandos del teclado a un botón de tu gamepad/joystick de la misma forma que simplemente podías asignar las teclas de dirección del teclado a tus gamepads.

¿La imagen es compatible con CEC? En principio No. La imagen tiene libcec instalado, que funciona en HDMI 1, pero eliminé CEC de la imagen XBMC porque causaba problemas. Sin embargo, puedes instalar XBMC con soporte CEC si lo deseas desde mi repositorio <http://oph.mdrjr.net/meveric/>.

Cuando salgo de un emulador, la ventana del XBMC es muy pequeña, ¿cómo puedo solucionar esto? Si sales del XBMC y lo reinicia nuevamente, volverá a la pantalla completa. Te aconsejaría "maximizar" la ventana del XBMC de todos modos para que te sea más fácil seleccionar las opciones.

¿Hay algún entorno aparte del XBMC en la imagen? Sí, detrás de XBMC existe una completa distribución de Debian con LXDE. Esto significa que puede instalar todo lo que quieras en la imagen que esté disponible desde el repositorio de Debian, y mucho más. También puede hacer todo lo que sueles hacer en las imágenes de Ubuntu, como navegar por internet, editar documentos y diseñar elementos gráficos.

Kit Consola de Juegos OGST para ODROID-XU4

© June 1, 2018 By Justin Lee Juegos, ODROID-XU4, Tutoriales



El kit consola de juegos OGST para ODROID-XU4 te permite montar tu propia consola de juegos con un potente **ODROID-XU4** o **ODROID-XU4Q**. Su atractivo diseño incluye una elegante pantalla LCD de 2.4" que permite mostrar animaciones del logotipo programables de los juegos, y está específicamente diseñado para funcionar con la popular imagen **ODROID GameStation Turbo**. Este **kit de consola está disponible en la tienda Hardkernel por 24\$**, y funciona muy bien con el nuevo joystick **GameSir G3w USB Controller**. El ODROID-XU4 o XU4Q, la tarjeta microSD o el módulo eMMC, la fuente de alimentación 5V/4A y el mando/joystick USB no están incluidos.

Contenido del kit

- A. Carcasa superior
- B. Carcasa inferior
- C. Placa LCD OGST-XU4
- D. Cable de extensión USB2.0

- E. Cable plano de 30 pines
- F. Tapa del puerto USB
- G. Conjunto de bases de goma
- H. Tornillos
- I. Cable de extensión USB3.0 (para almacenamiento opcional HDD USB de 2.5")



OGST Gaming Console Kit for ODROID-XU4

Figura 1 - Kit de GameStation con anotaciones

Para obtener más información, visita nuestra [página de la Wiki](#).

Montaje



Figura 2 - Utiliza un cable plano de 30 pines (E) para conectar el XU4 y la placa LCD (C). A continuación, retira la película protectora de la superficie de la LCD. El ODROID-XU4 se vende por separado



Figura 3 - Monta la placa LCD + el XU4 en la carcasa superior (A) y fija los tornillos (H)

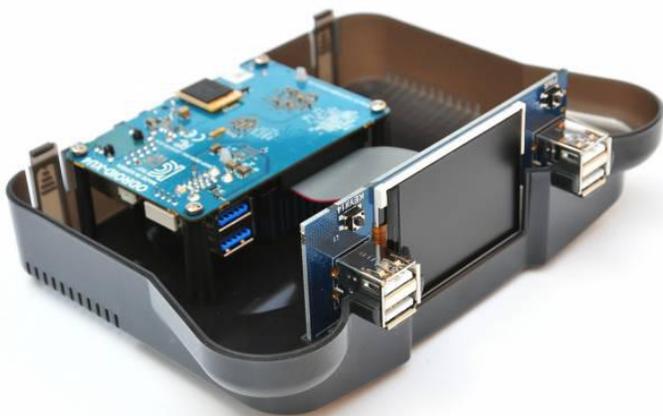


Figura 4 - Conecta la placa XU4 y la placa LCD con el cable USB2.0 (D)



Figura 5 - Conecta la placa XU4 y la placa LCD con el cable USB2.0 (D)



Figura 5 - Conecta la placa XU4 y la placa LCD con el cable USB2.0 (D)



Figura 7 - Puedes conectar un HDD de 2.5" usando el cable USB3.0 (I) incluido en el pack y asegurar la unidad de disco duro a la carcasa inferior con cinta velcro. El pack no incluye una unidad de disco duro. Si tu almacenamiento HDD consume más de 500 mA, es posible que necesites utilizar una fuente de alimentación de 5V/6A en lugar de una de 5V/4A



Figura 8: Monta la carcasa inferior (B) y coloca la cubierta USB (F). A continuación, coloca las bases de goma (G) en la carcasa inferior

Software

ODROID GameStation Turbo es una imagen de sistema operativo ODROID basada en el entretenimiento que te permite ver películas en Kodi, escuchar música, jugar desde tu escritorio, además de poder ejecutar juegos retro con diferentes emuladores desde Kodi, EmulationStation o Attract-Mode como interfaz.

La imagen se puede descargar desde los foros ODROID desde el hilo [ODROID GameStation Turbo for XU3/XU4](#). Desplácese hacia abajo hasta que encuentre el texto que se muestra en la Figura 9.

Where can you get it
Debian Jessie Version (hosted by @mdrjr): <http://oph.mdrjr.net/neveric/images/OGS...sie.img.xz - md5 - sha512 - signature>
Debian Jessie Version (hosted by Fourdee): <http://fuzon.co.uk/neveric/images/ODROI...sie.img.xz - md5 - sha512 - signature>

Figura 9 - Ubicaciones de descarga de ODROID GameStation Turbo

Haz clic para seleccionar uno de los enlaces. Los dos enlaces deberían ser iguales, pero si uno te va muy lento, puedes probar el otro. Los archivos md5, sha512 y sig son para aquellos usuarios que desean asegurarse de que el archivo descargado no está dañado o no ha sido alterado, pero puede ignorarlos si no sabe qué hacer con ellos.

Una vez que hayas descargado la imagen, sigue la guía por el [camino que más facil te resulte para instalar el nuevo SO](#) y grabar la imagen en tu tarjeta SD o módulo eMMC. Necesitará al menos una tarjeta SD/eMMC de 8GB para usar esta imagen, pero debido a la naturaleza de la imagen (juegos y videos), sugiero utilizar tarjetas SD/eMMC más grandes, como 32 GB o 64 GB (o usar una unidad de almacenamiento externo adicional).

Si usa una tarjeta SD, sugiero usar una SanDisk Ultra (no Ultra Plus o Extreme), ya que parece ser la marca/modelo más compatible que hay disponible (también es bastante barata), otros modelos pueden tener problemas (limitaciones del hardware XU4). Una vez que haya grabado la imagen en tu tarjeta SD o eMMC, insértala en el ODROID y arranca el sistema.

Cuando inicias la imagen por primera vez, aparecerá el escritorio e se iniciará Kodi, luego se reiniciará para terminar con las primeras tareas de mantenimiento. No tienes que hacer nada, todo sucederá automáticamente y simplemente debe esperar a que el sistema se reinicie. Esto puede tardar hasta 5 minutos, pero por lo general debería ser rápido.

Después de reiniciarse el sistema, volverá a cargarse el escritorio y Kodi. Salte de Kodi presionando "S" en el teclado y selecciona Exist/Quit Kodi. Ahora te encuentras en el escritorio de tu sistema operativo y puedes acceder a todos los programas que quieras, como el navegador web, reproductor de música, instalar nuevos programas o puedes volver a Kodi para ver películas. Pero empecemos con algunos conceptos básicos y luego instalemos algunos juegos en el sistema para poder jugar un rato.

Llegado a este punto, querrás estar conectado a Internet. Conecta un Módulo WiFi o el cable Ethernet a tu ODROID. En el escritorio de tu sistema operativo, asegúrese de estar conectado a internet. Para hacerlo, haz clic en el menú superior derecho para verificar/conectarte a internet. Selecciona la red WiFi deseada e introduce su contraseña.

A continuación, abre una nueva ventana de terminal, pinchando el símbolo Terminal MATE en la lista de Inicio rápido de la parte superior de la pantalla o en el Escritorio, o presionando la combinación de teclas CTRL + ALT + T. En la ventana de terminal recién abierta escribe los siguientes comandos para actualizar el sistema.

```
$ su
$ apt-get update && apt-get upgrade && apt-get
dist-upgrade
```

Esto te conectará como root y te solicitará tu contraseña de root que es "odroid" (sin comillas) por

abre EmulationStation. Dirigete a Applications -> Games -> EmulationStation y tus juegos ya deberían aparecer en el listado.



Figura 12 - Iniciando EmulationStation desde el escritorio de ODROID GameStation Turbo

Tenga en cuenta que esto no afecta al emulador ni a los juegos que quieres ejecutar, simplemente se usa para navegar dentro de EmulationStation. Necesitarás

Joystick ODROID GameSir G3w USB

© June 1, 2018 By Justin Lee Android, Juegos, Tutoriales



El GameSir G3w es un gamepad de alta calidad que utiliza un chip MCU de 32 bits, con una capacidad informática de hasta 48 millones de operaciones por segundo. Permite que el dispositivo sea altamente sensible y preciso en cuanto a rendimiento se refiere. El D-pad flotante hace que sea más fácil distinguir los ocho puntos de la brújula. Los joysticks analógicos tienen un posicionamiento de 360 grados de ultra precisión.



Figura 1 - El gamepad GameSir es el último accesorio de alta calidad para juegos de Hardkernel

También cuenta con botones ergonómicos muy bien diseñados, sólidos y sensibles, de acción firme y pueden presionarse sin esfuerzo. La sensibilidad al presionar R2 / L2 permite a los jugadores percibir la profundidad de la presión ejercida con bastante exactitud, lo que hace que sea mucho más fácil jugar a los juegos de carreras de coches. La longitud del cable USB es de 2 metros (6,6 pies) para un uso más cómodo.

High Sensibility & Accuracy Buttons

-  **D-Pad**
Pressure sensitive D-pad
-  **Joystick**
360 degree without-dead zone
-  **A/B/X/Y**
Color LED lights



Press Test

1000000+

A/B/X/Y, D-PAD, L1/R1/L2/R2 silicone pads are made of anti-static, highly sensitive and conductive silicone material, all pads have been pre-tested by strict pressing of one million times that ensures the lifecycle of buttons to be 5 years or more.

Figura 2

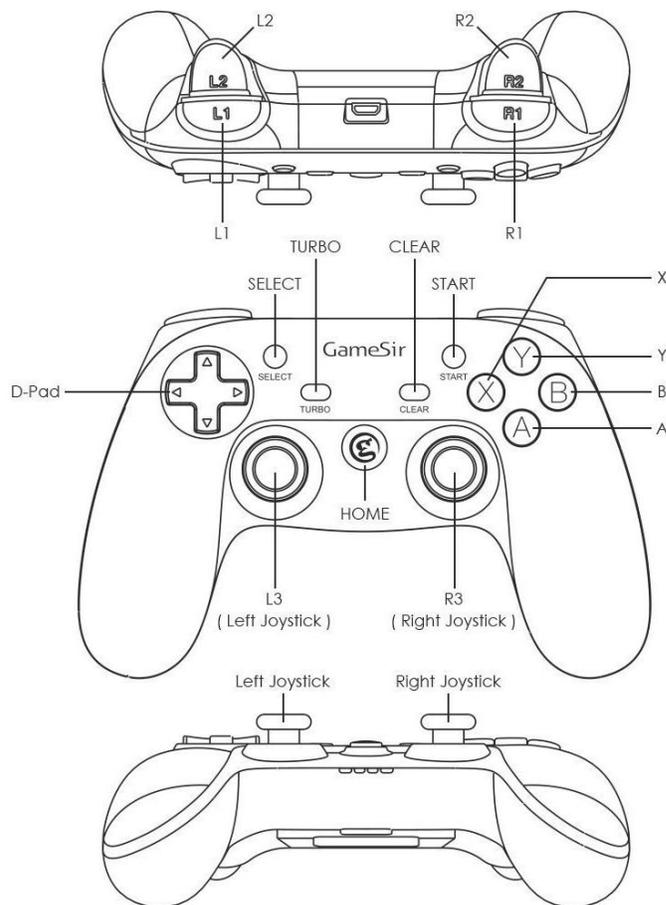


Figura 3



Figura 4

Este gamepad funciona con el kit Consola OGST de Hardkernel (<https://goo.gl/5fSR48>). Es compatible con los sistemas operativos oficiales de **Linux** y Android que ofrece Hardkernel.

Configuración

A continuación, se muestra un archivo de configuración udev a modo de referencia para

(/usr/share/libretro/autoconfig/udev/xiaoji_GameSir_G3w.cfg)

```
input_driver = "udev"
input_device = "Gamesir-G3w"
input_device_display_name = "xiaoji GameSir
G3w"
# Hex vid:pid is found using "dmesg -w" or
"tail -f /var/log/syslog"
# and converted to Decimal using
# http://www.binaryhexconverter.com/hex-to-
decimal-converter
# Hex vid:pid = 20BC:5500 -> Decimal vid:pid =
8380:21760
input_vendor_id = "8380"
input_product_id = "21760"
input_b_btn = "0"
input_y_btn = "3"
input_select_btn = "10"
input_start_btn = "11"
input_up_btn = "h0up"
input_down_btn = "h0down"
input_left_btn = "h0left"
input_right_btn = "h0right"
input_a_btn = "1"
input_x_btn = "4"
input_l_btn = "6"
input_r_btn = "7"
input_l2_btn = "8"
input_r2_btn = "9"
input_l3_btn = "13"
input_r3_btn = "14"
input_l_x_plus_axis = "+0"
input_l_x_minus_axis = "-0"
input_l_y_plus_axis = "+1"
input_l_y_minus_axis = "-1"
```

```
input_r_x_plus_axis = "+2"
input_r_x_minus_axis = "-2"
input_r_y_plus_axis = "+3"
input_r_y_minus_axis = "-3"
input_b_btn_label = "A"
input_y_btn_label = "X"
input_select_btn_label = "Select"
input_start_btn_label = "Start"
input_up_btn_label = "D-Pad Up"
input_down_btn_label = "D-Pad Down"
input_left_btn_label = "D-Pad Left"
input_right_btn_label = "D-Pad Right"
input_a_btn_label = "B"
input_x_btn_label = "Y"
input_l_btn_label = "L1"
input_r_btn_label = "R1"
input_l2_btn_label = "L2"
input_r2_btn_label = "R2"
input_l3_btn_label = "Left Stick"
input_r3_btn_label = "Right Stick"
input_l_x_plus_axis_label = "Left Stick Right"
input_l_x_minus_axis_label = "Left Stick Left"
input_l_y_plus_axis_label = "Left Stick Down"
input_l_y_minus_axis_label = "Left Stick Up"
input_r_x_plus_axis_label = "Right Stick
Right"
input_r_x_minus_axis_label = "Right Stick
Left"
input_r_y_plus_axis_label = "Right Stick Down"
input_r_y_minus_axis_label = "Right Stick Up"
```

Para más información y comprar el Gamepad
GameSir G3w, visita

http://www.hardkernel.com/main/products/prdt_info.php?g_code=G152605577891.

Micro Servidor Alimentado con Energía Solar

© June 1, 2018 By Roberto Rosario Mecnico, Tutoriales



Como probablemente ya habrás leído, hace menos de un año Puerto Rico experimentó uno de los peores desastres naturales de la historia escrita (<https://www.mayan-edms.com/post/hurricane-maria>). La situación tras el paso del huracán María se ha visto agravada por una infraestructura energética deficiente de por sí. Los frecuentes apagones son bastante comunes en la vida diaria.

Los apagones no solo son molestos, sino que causan una serie de problemas que afectan a casi todos los aspectos de la vida moderna. Al trabajar con tecnología, los apagones afectan directamente a mi resultado final y a mi principal fuente de ingresos.

Tras meses en la oscuridad, decidí dar el paso y construir un pequeño sistema de energía solar. Para hacer uso de la energía solar, es muy común recurrir a un convertidor de voltaje con el fin de potenciar los 12 voltios de corriente continua (CC) producidos por los paneles solares y almacenados en baterías de 120

voltios de corriente alterna (CA), que es la que utilizan la mayoría de los dispositivos electrónicos. El uso de un convertidor provoca algunas pérdidas de conversión en el sistema. Para minimizar estas pérdidas, empecé a convertir tantos dispositivos como pude para que trabajasen con 12 voltios de CC. De esta forma podrían ser alimentados directamente desde las baterías del sistema solar. Era hora de empezar a convertir también mis herramientas de trabajo.

Algunos de los sistemas que necesito para mi trabajo diario son los ordenadores y los servidores. Tengo algunos alojados en la nube y otros en local. Los constantes cortes de energía significan que una gran parte de la carga de las baterías de mi sistema solar se está desperdiciando, manteniendo los servidores y los ordenadores en funcionamiento. Tenía que haber una mejor forma de mantener encendido los equipos que necesitaba y optimizar su uso de la energía eléctrica.

En lugar de tener ordenadores más grandes, decidí empezar a reducir los dispositivos que usaba. Más dispositivos implicaba tener que repartir los servicios entre todos ellos.



Servidor con energía solar montado

Los ordenadores de placa reducida o SBC son muy comunes hoy día con una gran variedad donde poder elegir. Estos SBC suelen funcionar con USB, de modo que ya funcionan con corriente continua. Uno de mis ordenadores favoritos de placa reducida es el ODROID C2. El ODROID C2 es producido por una empresa bastante responsable, está bien documentado y cuenta con un gran soporte. ODROID C2 tiene una CPU ARM de cuatro núcleos de 64 bits, 2 GB de RAM y soporta módulos eMMC para el almacenamiento. Es uno de los SBC más potentes del mercado dentro de su gama de precios. Con esto en mente, me propuse transformar algunos de mis ODROID y SBC existentes para reemplazar los típicos servidores y ordenadores que normalmente uso.



Figura 2: Las carcasas impresas en 3D de plástico muy comunes para estos SBC no son suficientes para protegerlos y no dejan espacio para añadir componentes adicionales. Las carcasas de aluminio representan una opción bastante mejor.

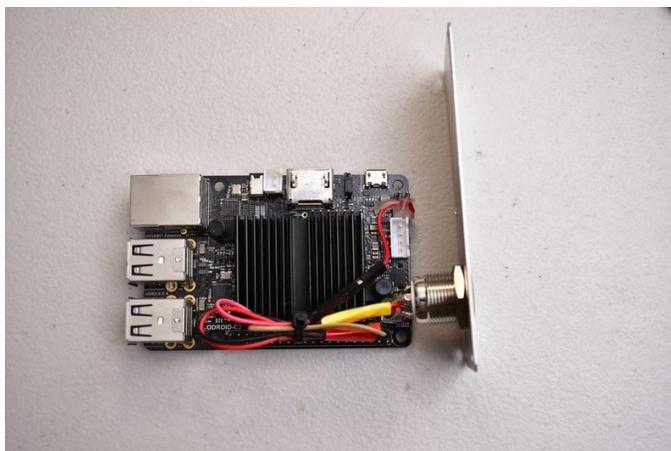


Figura 3 - Este es el ODROID C2 con botón e indicador de encendido instalados. El interruptor no está conectado a nada y se deja para un futuro uso (tal vez a través de GPIO). El indicador de alimentación está soldado al LED de actividad de la placa

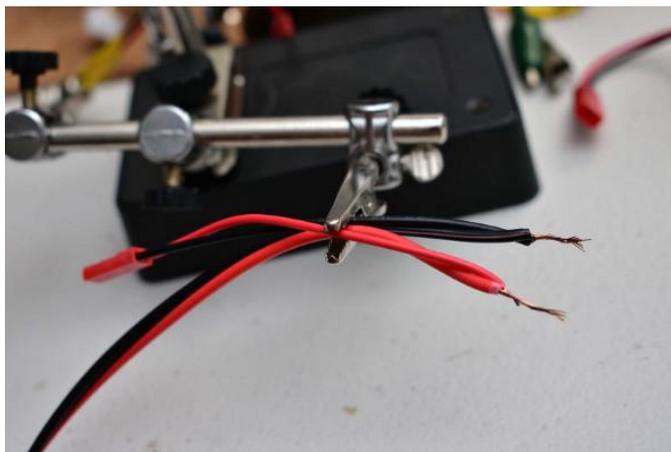


Figura 4 - El cable de alimentación

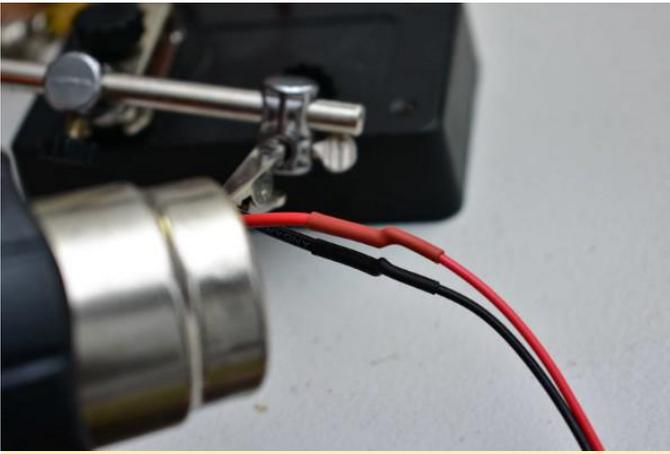


Figura 5 - El conector JST va al convertidor de voltaje que está dentro de la carcasa de aluminio. Todas las conexiones de cables están protegidas con tubos termo retráctiles



Figura 8 - Aquí estoy probando la continuidad

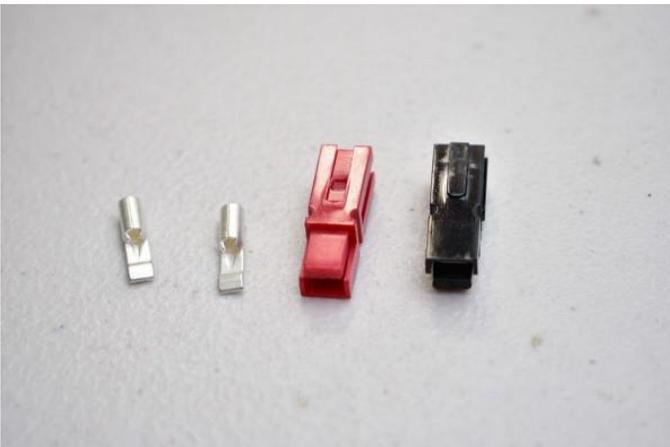


Figura 6 - El conector Anderson va por la parte exterior del cable



Figura 9: Añadí un poco de tubo termo retráctil para que trabajase a modo de presión con el cable

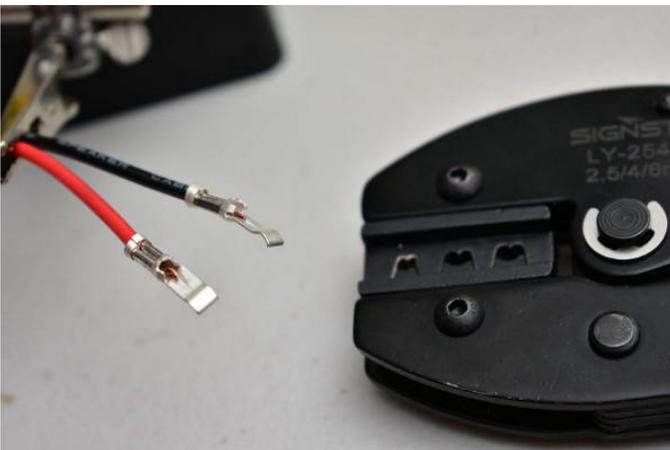


Figura 7: Añadí un poco de soldadura para una mejor conductividad



Figura 10 - El ODR0ID montado usando separadores de latón. El residuo procede de una cinta de doble cara de un intento anterior

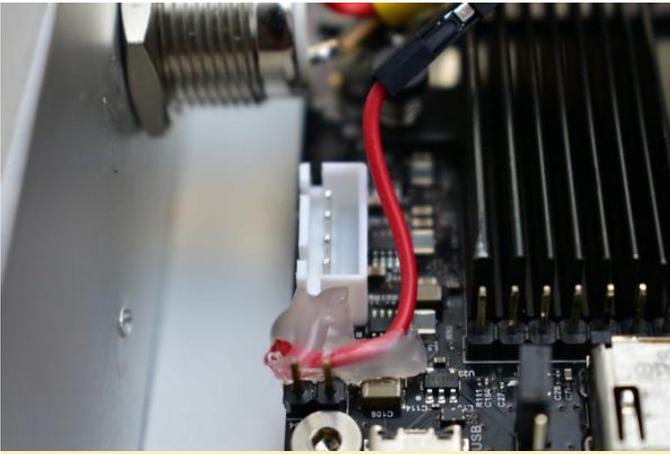


Figura 11 - La consola serial y las clavijas de alimentación fueron reemplazados tal y como se muestra en la imagen.

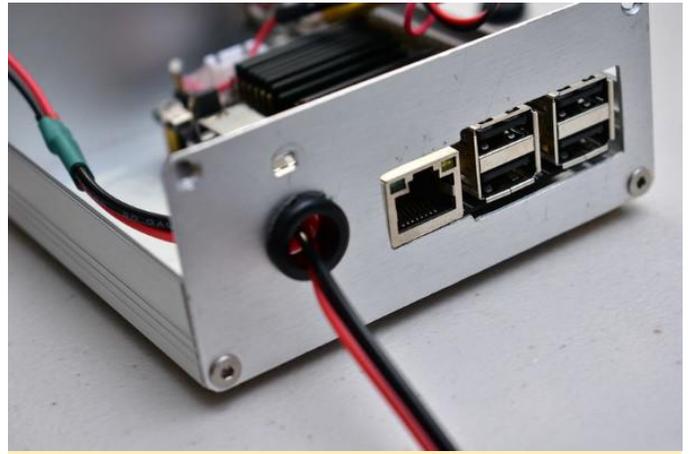


Figura 14 - Este convertidor en particular puede suministrar 3 amperios de forma continuada

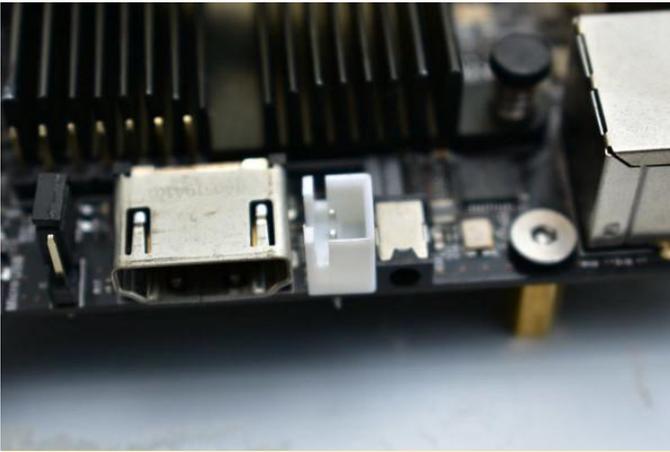


Figura 12 - La consola serial y las clavijas de alimentación fueron reemplazados tal y como se muestra en la imagen



Figura 15 - Todo montado y todos los cables atados correctamente

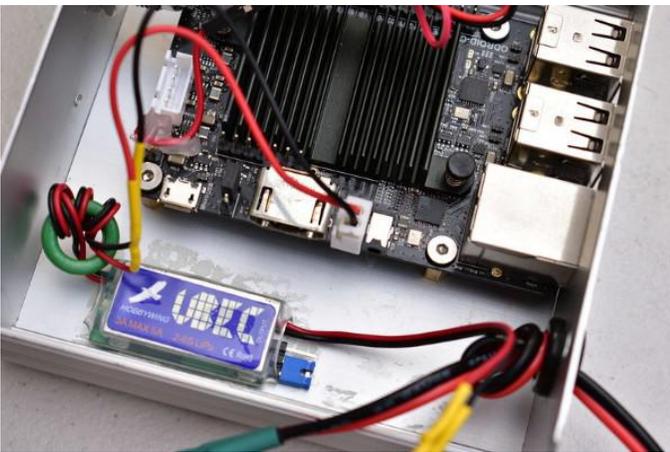


Figura 13 - Aquí puedes ver el convertidor de voltaje montado con cinta de doble cara. El convertidor coge los 12 voltios del sistema solar y los reduce a 5 voltios

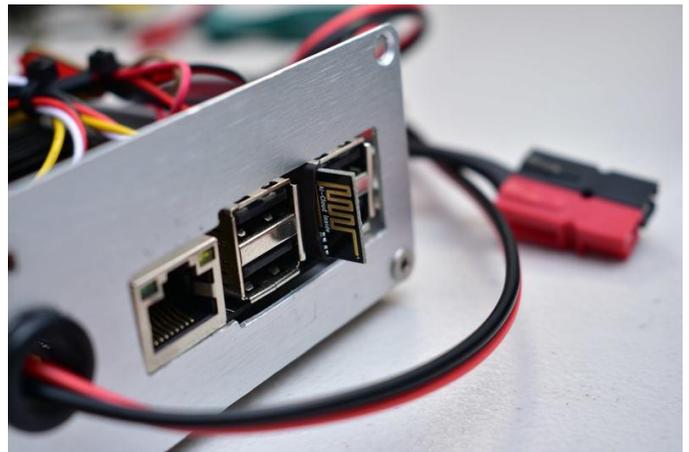


Figura 16 - El ESP8266 (ESP-01) programado para funcionar como enlace entre la consola serie y el WiFi. Esto permitirá acceder a la consola del ODROID incluso si las comunicaciones fallan. Esto es perfecto para tareas de depuración y gestión



Figura 17 - Aquí tiene la carcasa completamente montada. Los agujeros para el USB, el conector de red y el botón de encendido fueron hechos a mano y hubo algunos rasguños. Esto puede evitarse usando cinta de pintores o usando mejores herramientas como un CNC o un taladro de banco

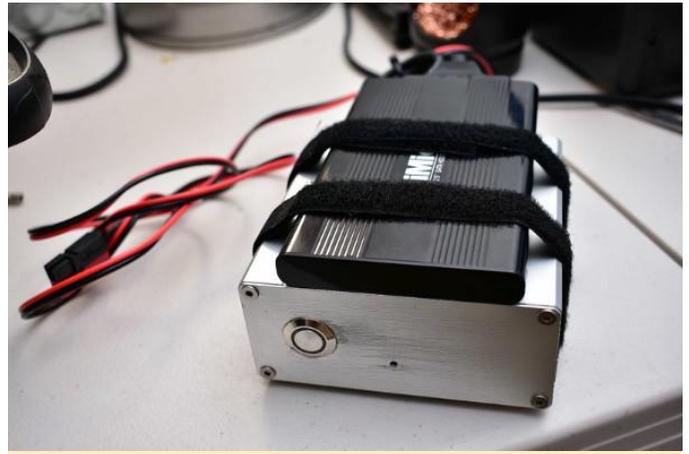


Figura 19 - Los puertos USB de ODROID son USB 2.0. Esto significa que el acceso al almacenamiento no será muy rápido, pero es adecuado para el trabajo que realizará este sistema



Figura 18 - El ODROID-C2 es compatible con almacenamiento eMMC y éste tiene una tarjeta de 16 GB que se utiliza para los archivos del sistema. Para el almacenamiento, se usa un disco duro SSD SATA externo con un adaptador SATA a USB

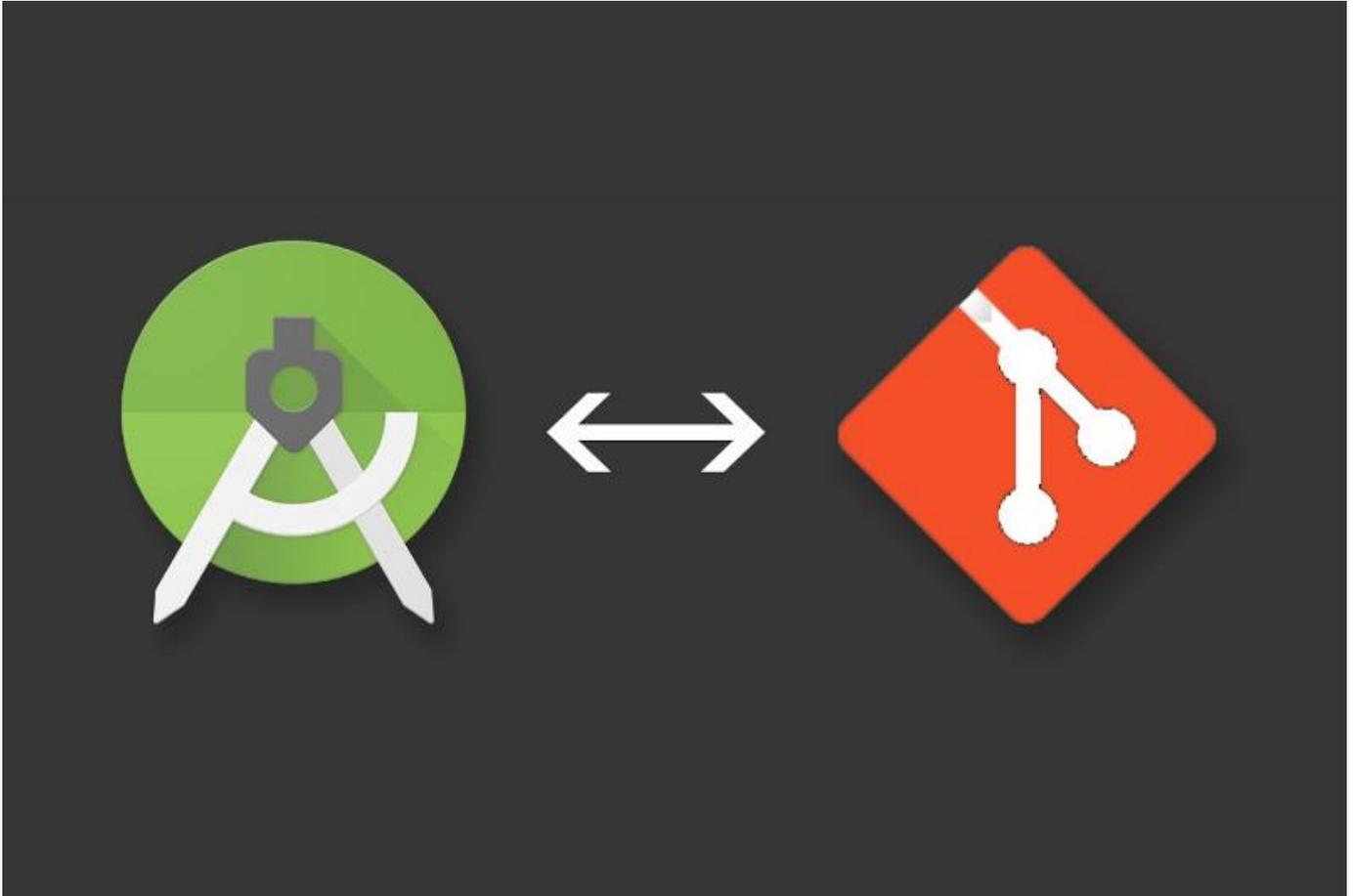


Figura 20 - Aquí tienes el microservidor colocado al lado del controlador de carga del sistema solar y conectado a la caja de distribución de energía Anderson

Para comentarios, preguntas y sugerencias, visita el artículo <https://medium.com/@siloraptor/solar-powered-microservers-for-a-post-hurricane-maria-puerto-rico-ca83027d20ac> original en

Desarrollo Android: Usando GitHub

© June 1, 2018 By Randy Hall Android, Tutoriales



¡Bienvenido de nuevo, desarrollador de apps! Si eres como yo, estás más que preparado para mejorar tus habilidades en el desarrollo de aplicaciones. Como he mencionado en otras ocasiones, la empresa que domina prácticamente todo el monopolio de proyectos de código abierto online es GitHub. Existen varias opciones basadas en Git en el mercado, pero teniendo en cuenta nuestros objetivos, utilizaremos GitHub, con la ventaja de que todos los proyectos y errores estarán disponibles para que la comunidad los vea, aporte sus comentarios y mejoras.

La buena noticia es que Android Studio trata a GitHub como un Repositorio de código fuente de primera clase, es decir, puedes hacer que funcionen otros proveedores Git, pero no están tan pulidos. Esto simplemente significa que si decides seguir tu propio camino con algo diferente a GitHub como herramienta de control para el código fuente, entonces lo dejaremos como un simple ejercicio, para que lo completes y nos dejes tus comentarios en los

foros ODROID. Así pues, conectemos el IDE a nuestro proveedor de control de código fuente.

Subir el código a GitHub

Si nos fijamos en el código de nuestra última entrega, era bastante básico. Android Studio sólo tardó unos cuantos minutos en crearlo, y luego tuvimos una completa carpeta de proyecto con scripts de compilación, código de la aplicación y archivos de diseño. El inconveniente de toda esta automatización con las herramientas del entorno de desarrollo integrado es que crea una gran cantidad de archivos, y hacer un seguimiento de ellos (y los cambios que se realizan en los mismos) puede darte un gran dolor de cabeza. Aquí es donde los sitios de alojamiento como GitHub realmente nos ayudan, ya que nos permiten mantener una copia de seguridad de todos nuestros valiosos archivos de código, listos para ser recuperados en caso de fallo en el disco, pérdida del equipo o cualquier otro desastre. De modo que,

partiremos del proyecto que creamos la última vez con Android Studio, nos dirigiremos al menú “VCS”, seleccionaremos “Import into Version Control” y seleccionaremos “Share Project on GitHub”. Si aún no ha pasado por el proceso de crear una cuenta GitHub, al que hicimos referencia la última vez, ahora es tu oportunidad.

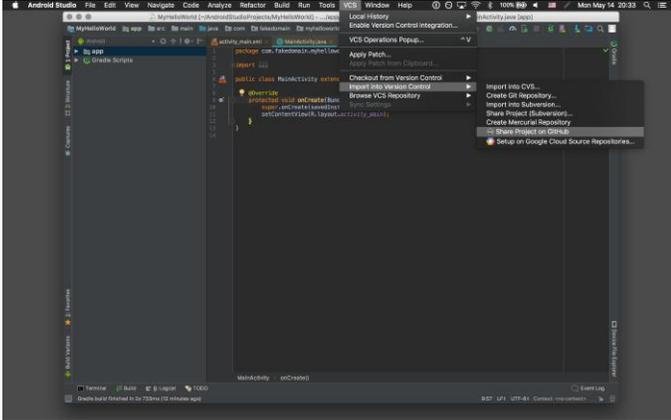


Figura 01 – Definir el control de versiones en GitHub

Una vez que hayas confirmado y configurado la cuenta, puedes volver a Android Studio y dentro del cuadro de diálogo “Login to GitHub”, seleccionar “Password” en el Tipo de autenticación e introduce tu nombre de usuario y contraseña. A continuación, se mostrará el cuadro de diálogo “Share Project on GitHub”, que te permite cambiar el repositorio, o “repo”, el nombre, el nombre remoto y la descripción. Por ahora, deja tal cual el nombre del repositorio y el nombre remoto, y añade una simple descripción para el proyecto. Yo escribí algo simple, como “My first Android project”. Después, podrás confirmar qué archivos se enviarán inicialmente a GitHub, es adecuado aceptar el todo conjunto por defecto y hacer clic en “OK”. En pocos segundos, aparecerá una notificación en la cual se indicará que el proyecto está subido a GitHub. Puede hacer clic en el enlace de la notificación y verificar tu código fuente online.

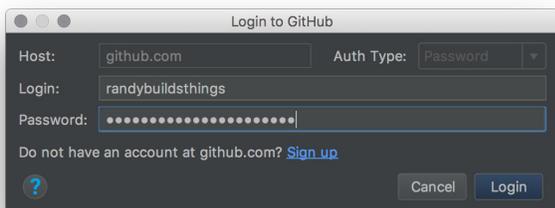


Figura 02 – Conectarse a GitHub

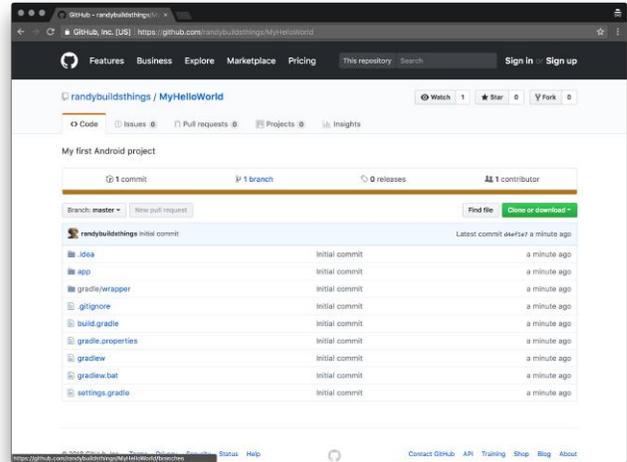
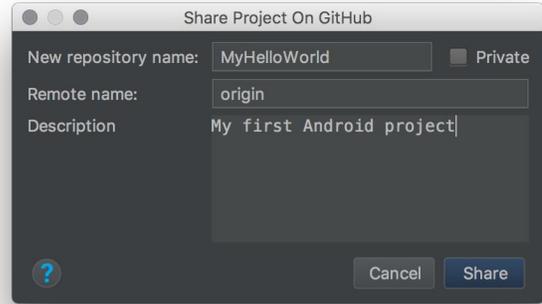


Figura 04 – Página del proyecto GitHub

La app más básica jamás creada

Con todo el tema sobre cómo poner nuestro código para el control de versiones, realmente no hemos dedicado mucho tiempo a ver lo que Android Studio había generado. Vamos a hacerlo ahora. Al crear esta aplicación, Android Studio genera una gran cantidad de archivos, aunque solo unos cuantos son realmente importantes para comprender lo que sucede. Estos son los archivos que nos interesan por el momento:

- app/manifests/AndroidManifest.xml
- java/com/fakedomain/myhelloworld/MainActivity.java
- res/layout/activity_main.xml
- res/values/strings.xml

AndroidManifest.xml

Este archivo es útil desde el punto de vista de que realmente le dice al dispositivo Android qué código se ejecutará cuando la aplicación se inicie por primera vez. Si haces doble clic en el archivo y lo examinas en

el IDE, no hay mucho que ver, pero todo lo que contiene es importante para ejecutar correctamente tu aplicación. Dedicaremos tiempo en otra entrega para analizar a fondo lo que realmente hace el archivo por tu aplicación, pero por ahora es suficiente saber que Android Studio se encargará de incluir nuevas actividades y otras funciones de la aplicación en el archivo, al menos hasta que te se sientas más cómodo haciéndolo tú mismo.

MainActivity.java

Éste solo contiene unas cuantas líneas de código, pero dirige al lector a un recurso importante al que sin duda alguna dedicarás tiempo: el archivo de diseño `activity_main.xml`. Aquí es donde se define el “aspecto general” de la aplicación y es donde se puede cambiar utilizando las herramientas de diseño visual de Android Studio. Si haces doble clic en el archivo `activity_main.xml`, el IDE debería mostrarte su editor de diseño visual. Si ves un diseño con un panel de archivo XML al lado, puede que hayas elegido la vista “Text” del diseño. Si haces clic en la pestaña “Design” que se encuentra en la parte inferior del panel, volverá a cambiar. Aunque el diseño es algo escaso, todavía podemos hacer bastante desde aquí. De hecho, si te fijas bien, nuestro texto “Hello World” ya aparece en su lugar en medio de la vista. Aquí entra en juego la denominada Empty Activity: no está completamente vacía, aunque tiene un elemento `TextView` muy simple en el diseño para que puedas iniciarla. ¿Y qué mejor texto de muestra para poner en un widget de texto que nuestra favorita frase de dos palabras?

strings.xml

Strings.xml es el último archivo que no debemos perder de vista. La razón de ello es que, como desarrollador de aplicaciones, las buenas prácticas son esenciales y cuanto antes las adoptes, mejor. Para los proyectos puntuales desechables, está bien andar escribiendo valores fijos de texto para los diferentes botones, vistas de texto y controles de navegación, pero en el mundo real hay muchos idiomas, y seríamos realmente cerrados, si no arrogantes, al asumir que el inglés es el único idioma que debería incluir nuestra aplicación. De modo que, modificaremos una configuración para hacer uso de este archivo. Si haces doble clic en el texto “Hello World” en el diseño, verás que aparece el panel de Atributos. Como resultado, el atributo de texto `TextView` muy probablemente aparecerá resaltado, puesto que has hecho doble clic en el editor de diseño. A la derecha de ese recuadro de atributo, debería ver unos puntos suspensivos (...), sobre los cuales puedes hacer clic para mostrar la ventana de recursos para este proyecto. Dentro encontrarás los recursos que puedes utilizar para el atributo que estás editando.

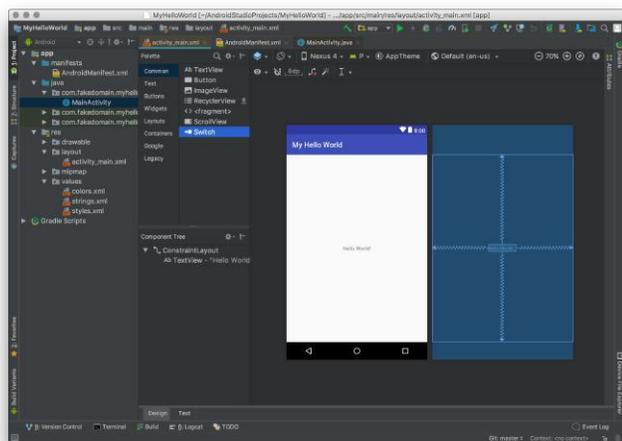


Figura 05 - Vista de diseño

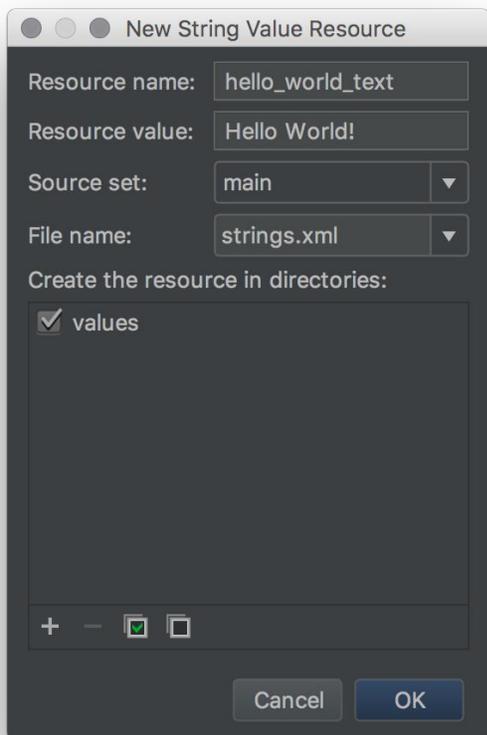


Figura 06 – Editor de recursos de valor de cadena

Como queremos hacer un cambio, haga clic en el elemento de menú “Add new resource” en la esquina superior derecha de la ventana y luego selecciona “New string Value...”, se mostrará el cuadro de diálogo “New String Value Resource”. Para “Resource name”, elije algo sensato como “hello_world_text”, y luego define el valor del recurso en “Hello World!” Puedes hacer clic en OK y el valor en el atributo de texto habrá cambiado a “@strings/hello_world_text” pero el diseño seguirá mostrando el valor, que es “¡Hola, mundo!”. Si observas strings.xml con el editor de código, verás que hay una nueva línea en el archivo, que Android Studio ha colocado por ti. ¡Genial!

¡Vamos a ejecutarlo!

Ahora tenemos que compilarlo todo y ver que se ejecuta. En la parte superior de la ventana IDE, en el centro de la barra de herramientas verás un icono en forma de martillo verde. Haz clic en él y se compilará el proyecto. Puede tardar algunos minutos dependiendo de la velocidad de tu máquina de desarrollo, pero al final el proceso se completará. A modo de comentario, a pesar de que me encantan los

dispositivos ODROID para ejecutar aplicaciones de Android y Linux en general, e incluso el poderoso ODROID-XU4, no son especialmente adecuados para el desarrollo de aplicaciones de esta naturaleza. Simplemente porque se necesita mucha memoria durante el proceso compilación del software, y compilar una aplicación con un cierto contenido al final te llevaría a forzar tu tarjeta eMMC o SD cuando Android Studio tuviese poca memoria, acortando la vida útil de tus soportes de almacenamiento. Pero bueno, tú decides, y si ODROID es con lo que sueles compilar, éste puede hacer el trabajo, aunque es posible que tengas que esperar algo más con determinadas tareas. Ahora que la aplicación está creada, si presionas el triángulo verde “Play”, la aplicación se “ejecutará” e intentará conectarse a un dispositivo Android. ¿Qué es eso? ¿No tienes un dispositivo Android a mano? ¿No has visitado la Wiki de ODROID para conectar tu ODROID-C2 utilizando su puerto USB OTG (https://wiki.odroid.com/getting_start/adb_fastboot/)? ¿No has iniciado Network ADB en tu ODROID-XU4 (https://wiki.odroid.com/odroid-xu4/troubleshooting/adb_over_tcpip/)? Aunque siempre es más provechoso probar una aplicación con un hardware real, si todo lo demás falla, siempre tienes la opción de emular un dispositivo Android.

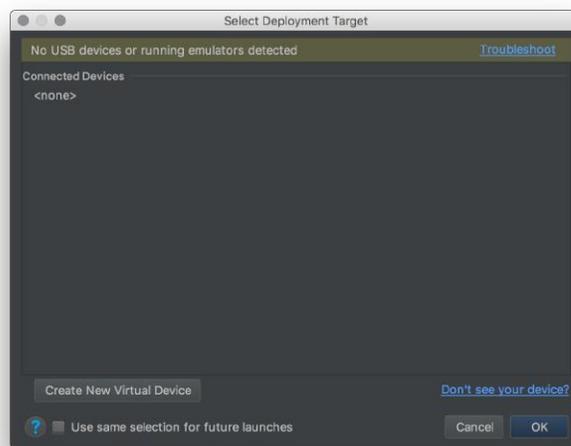


Figura 07 – Seleccionar el dispositivo para la implementación

Echar mano de lo virtual (dispositivos)

De modo que, si estás viendo el “Select Deployment Target” y está vacío, puedes hacer clic en “Create New

Virtual Device” y aparecerá un catálogo con varios dispositivos Android para hacer pruebas que puedes emular en tu ordenador, desde TVs, tablets, teléfonos hasta dispositivos Android Wear. En nuestro caso, creamos un dispositivo que emule un teléfono, un Nexus 5X. Una vez seleccionado, haz clic en “Next” y continúa eligiendo una de las imágenes del sistema “recomendadas”, que en nuestro caso sería Nougat, Oreo o la vista previa “P”. Probablemente tengas que hacer clic en el enlace “download” junto a la versión para descargar la imagen antes de continuar y sí, puede llevar un tiempo.

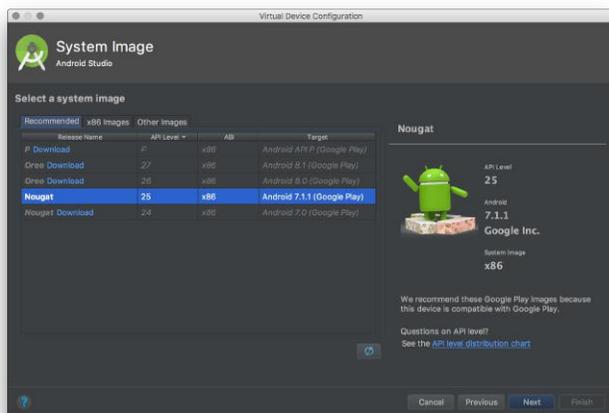


Figura 08 - Editor de configuración del dispositivo virtual

Una vez que hayas descargado la imagen, haga clic en “Next”, asigna un nombre al dispositivo virtual (o guarda el que genera por defecto Android Studio), cambia la orientación de la pantalla a modo apaisado si estás en pantallas horizontal y haga clic en “Finish”. ¡Por fin, tienes un dispositivo para iniciar tu aplicación!

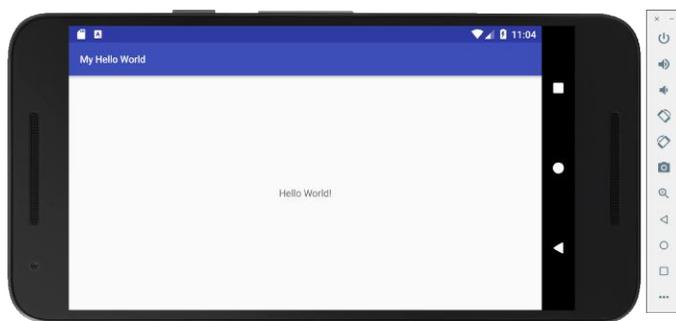


Figura 09 - Dispositivo virtual ejecutando nuestra aplicación

Si todo va según lo previsto, el dispositivo virtual arrancará la versión de Android que descargaste y lo situará en la pantalla de inicio. Unos instantes

después, Android Studio se conectará al dispositivo virtual utilizando ADB y ejecutará la aplicación. Puede que no resulte tan gratificante como tener la aplicación ejecutándose en un dispositivo Android real, pero una vez más, nunca podrás poner la excusa de que no tienes un dispositivo Android cuando lo necesitas.

Confirmar los cambios

Toda esta codificación requiere de mucho esfuerzo, y al final de cada sesión de codificación, siempre debe formar parte de tus prácticas habituales confirmar tu trabajo en curso. Soy el primero en decir que cuando trabajas solo en un repositorio de código, nadie se dará cuenta de que tu código no se compila correctamente o que contiene muchos errores y defectos. Te invito a superar esta cuestión. Lo importante es que conserves una copia de seguridad en caso de que suceda algo terrible y de repente te encuentres sin tu estación de trabajo con su almacenamiento de archivos. Para hacer esto, puedes volver al menú de VCS y seleccionar “Commit”, existe un atajo de teclado para esto, y también un completo panel de la ventana IDE usado para controlar las versiones. Asegúrate de echarle una ojeada y familiarizarte con él. Verás que al menos han cambiado dos archivos: los archivos `activity_main.xml` y `strings.xml`. Puedes echarle un vistazo a los cambios para asegurarte de que son los esperados. Ahora deberías introducir un mensaje de confirmación, en el cual se registrará un historial de los cambios realizados en el código que has comprometido en el repositorio. Yo introduciré algo escueto, como “changed text to use strings.xml”. Coloca el puntero del mouse sobre el botón “Commit” en la esquina inferior derecha, y verás que hay algunas opciones adicionales. Para nuestra demo en GitHub, vamos a continuar y seleccionaremos “Commit and Push ...” que mostrará un cuadro de diálogo que te permite agregar etiquetas al push para GitHub. Haga clic en “Push” y ¡listo! Nunca me cansaré de hacer hincapié en esta cuestión: asegúrate de verificar los cambios en tu código al final de tu día de trabajo, y no te lamentarás de irte a dormir (y es posible que incluso duermas mejor, ¡¡No hace falta que me des un abrazo!)

¿Qué es lo siguiente?

Como esta es una nueva columna, pensé que hacer una aplicación "Hello World" sería un comienzo entretenido. Pero, sinceramente, es la aplicación más básica que jamás se ha podido crear. Ahora estoy buscando opciones para trabajar en una aplicación que tenga un cierto sentido implementar en un ODROID-XU4 u ODROID-C1 + / C2 por diversión y que posiblemente implique un buen aprendizaje. Algunas ideas que me vienen fácilmente a la mente pueden ser:

- Un juego basado en Android TV, que se ejecutara en un ODROID-XU4 con LineageOS Android TV, que utilice los propios dispositivos Tablet o móviles Android de los jugadores para controlar y jugar a un juego de cartas por turnos.
- Una pantalla táctil personalizada de domótica Android, que se ejecutará en un ODROID-XU4 o ODROID-C2 con LineageOS, que utiliza servicios en segundo plano para comunicarse con otros servidores y dispositivos de domótica para monitorizar y controlar tu hogar sin usar servicios basados en la nube.
- Un Android Auto, que se ejecutara en un ODROID-C2 con pantalla táctil, que recopile información de tu coche y te permita conectar tu teléfono o tablet Android para reproducir música, navegar, comunicarte e informar del tiempo que pasas al volante.

Si la lista te parece algo intimidante o si te sientes que todo esto es demasiado complicado, simplemente relájate. Estoy aquí para ayudarte y haré todo lo

posible para explicarlo todo lo más claramente posible. Reconozco que cada uno de los proyectos a los que he aludido requiere combinar una serie de elementos de hardware y de software que trabajen conjuntamente para proporcionar una solución. De modo que, no te extrañe que haya próximos artículos centrados en el hardware para configurar o, al menos, simular una plataforma de hardware que luego podremos usar para escribir aplicaciones. También habrá ediciones en las que desglosaremos los problemas más graves de los fragmentos de código a los que luego aplicaremos prácticas de arquitectura de aplicaciones basadas en Android para que funcionen de la forma que esperan que lo hagan los usuarios de dispositivos Android. Creo que será una experiencia muy gratificante para todos, y todo se hará con código abierto y estará disponible online para que los futuros desarrolladores de Android puedan utilizarlo. Si crees que estas ideas son horribles y estás convencido de que tienes alguna mejor, ¡Quiero saber de qué se trata! Siempre puede comentar la versión interactiva de ODROID Magazine de este artículo, o visitar el foro ODROID Magazine en <https://forum.odroid.com>. Lo mejor de e mis artículos es que el código con el que trabajo está disponible online para que lo puedas analizar. No dude en visitar mi cuenta de GitHub en <https://GitHub.com/randybuildsthings> y reírte de muchas de las cosas tontas que hago. Uso GitHub principalmente para jugar, espero verte por allí.

Juegos Linux en ODRROID: Juegos Saturn – Parte 4

© June 1, 2018 By Tobias Schaaf ↗ Juegos, Linux, ODRROID-XU4



Volvemos con los juegos de ODRROID-XU4/XU4 y Sega Saturn. Esta vez vamos a echar un vistazo a los juegos que empiezan por la letra "S" como Saturn o como "shmup". Existen tantos juegos para Sega Saturn que empiezan por la "S" con los que disfruto jugando, que he tenido que hacerles su propia sección.

Samurai Spirits IV – Amakusa Kourin

Samurai Spirits IV es un buen juego del estilo "beat em up" similar al King of Fighter o Street Fighter, pero en el que cuentas con armas similares a Last Blade.



Figura 01 – Pantalla de inicio de Samurai Spirits IV para Sega Saturn

Este juego requiere una expansión de memoria de 1MB para que funcione correctamente, así que asegúrese de tener el cartucho correcto seleccionado. Me gusta bastante este juego. Es uno de los mejores

juegos de lucha en 2D a los que he jugado. Las sombras y las barras de salud parpadeantes son los únicos elementos que no me gustan de este juego, pero el resto es perfecto.



Figura 02 - Tienes una buena cantidad de luchadores para elegir en este juego

El juego es muy interesante, ya que los diferentes personajes no solo son diferentes en los gráficos, sino también en el estilo de lucha. Los movimientos son fáciles de dominar, incluso las combinaciones de botones pueden funcionar bien si sabes qué movimientos necesitas combinar.



Figura 03 - Las animaciones de lucha son realmente buenas, los gráficos y el fondo se ajustan muy bien al ambiente.

Incluso puedes desarmar a tu enemigo y debilitarlo hasta que recoja nuevamente sus armas. La música es agradable, hay muchas animaciones tanto en los

personajes como en el fondo. No soy un gran fanático de los juegos de lucha, pero me he pasado unas cuantas horas jugando a éste y posiblemente continuaré jugando.

Saturn Bomberman

No creo que Bomberman necesite presentación, algunos dicen que la versión de Bomberman de Sega Saturn es la mejor versión que existe. Todas las versiones que han salido a posteriori han sido comparadas con la versión de Sega Saturn.



Figura 04 - Saturn Bomberman



Figura 05 - Excelentes animaciones, colores brillantes y una muy buena jugabilidad, no hay nada mejor cuando se trata de Bomberman

Saturn Bomberman es realmente una joya, con bonitas escenas de anime, maravillosos colores brillantes, todo tipo de power-ups y un modo de

juego muy familiar. ¡Si te gusta Bomberman, definitivamente deberías jugar a la versión de Sega Saturn! Ah, ¿he mencionado que admite hasta 10 jugadores en una pantalla?

Sexy Parodius

Este shooter es similar a Parodius, del que hablé la última vez, se trata de una serie que parodia a los juegos Gradius un tanto "Cute 'em up".



Figura 06 - Pantalla de inicio de Sexy Parodius en Sega Saturn

Tiene una curiosa peculiaridad al final de cada sección donde tienes que matar a un par de jefes, y si los vences, se irá revelando lentamente una "imagen sexy", pero si te quedas sin vida se acabó. Tienes una cantidad ilimitada de continúes, de modo que puedes pasar al siguiente nivel, pero el nivel de la imagen habrá finalizado.



Figura 07 - Pantalla de selección de personajes de Sexy Parodius

Aparte de esto, el juego es un buen "cute 'em up", aunque en realidad es bastante difícil y es posible que mueras con bastante frecuencia, pero como he dicho, afortunadamente, hay ilimitados continúes.



Figura 08 - Lucha con un jefe en Sexy Parodius

Shienryu

Shienryu es otro shoot 'em up. Es un poco más simple en cuanto a mecánica de juego, ya que no puedes elegir diferentes tipos de habilidades, y solo tienes tres tipos de armas.



Figura 09 - Shienryu



Figura 10: Gráficos muy buenos con hasta dos jugadores con muchas explosiones y enemigos en pantalla

Puedes utilizar un disparo extendido, un ataque con rayos que apunta automáticamente a todos los objetivos y va saltando de unos enemigos a otros, o un ataque con cohetes que, básicamente es una combinación de misiles autodirigidos y cohetes directos. A cada una de las armas se le puede aumentar la potencia de fuego varias veces recopilando potenciadores. Además de esto, tienes "bombas", que son ataques especiales diseñados para causar mucho daño en una extensa área de la pantalla, tal y como pueden encontrar en muchos otros juegos del estilo. El juego se ejecuta muy bien en el ODROID-XU3/XU4. Los gráficos probablemente sean la parte más impresionante del juego, con muchos detalles y animaciones, nubes transparentes,

así como una buena cantidad de desplazamiento. Es uno de los mejores "shmups" para el sistema.



Figura 11 - Los cohetes pueden causar mucho daño y algunos son autodirigidos



Figura 12 - Ningún "shmup" está completo sin jefes que derrotar

Si te gustan los "shmups", este es uno que definitivamente deberías probar. El juego también está disponible para MAME y PS1, pero la versión para Saturn es posiblemente la mejor.

Shinrei Jusaishi Taroumaru

Este juego es un tanto único, se ve y se siente un poco como la serie Shinobi. Sin embargo, en lugar de lanzar cuchillos o golpear al enemigo con los puños, recurre a tu energía interna para dispararles rayos.



Figura 13 – Pantalla de inicio de Shinrei Jusaishi Taroumaru



Figura 15 – Luchando contra un jefe junto con un enemigo capturado



Figura 14 – Cargando tu ataque antes de golpear para causar un mayor daño

En este juego, caminas de izquierda a derecha luchando contra multitud de enemigos que van desde llamas en movimiento, hasta ninjas y criaturas demoníacas, y de vez en cuando luchas contra diferentes monstruos jefes que varían bastante en fuerza y en la forma de matarlos. En mi opinión, el juego es un poco difícil de controlar. Tienes una función de apuntar automáticamente que únicamente se bloquea si aparece un enemigo frente a ti. Si el enemigo está detrás de ti, tienes que darte la vuelta, y tampoco funciona si un enemigo está justo encima.



Figura 16 – El juego incluso tiene romos rotativos

La variedad de enemigos, especialmente los jefes, es bastante impresionante, y los diferentes niveles y escenas en los que luchas son bastante buenos. No te aburrirás viendo el mismo fondo una y otra vez. El juego puede llegar a ser bastante difícil, pero aun así es muy divertido. Incluso puedes “capturar” a algunos enemigos y hacer que luchen por ti. También puedes reflejar algunos ataques o hacer que los proyectiles que van dirigidos hacia ti desaparezcan. El juego es bastante divertido, aunque no tiene función de guardado ni nada parecido, probablemente esté pensado para jugarse en una sesión y que puedas terminarlo en aproximadamente una hora más o menos.

Sonic Wings Special

Parece que la "S" no solo representa a Saturn sino también a "shmup", puesto que éste es una vez más un juego "shoot 'em up" para Sega Saturn. De nuevo, la jugabilidad es bastante simplista. No hay armas para recoger y no hay ataques con carga ni nada similar; simplemente un ataque estándar y un ataque especial "bomba".



Figura 17 – Sonic Wings Special

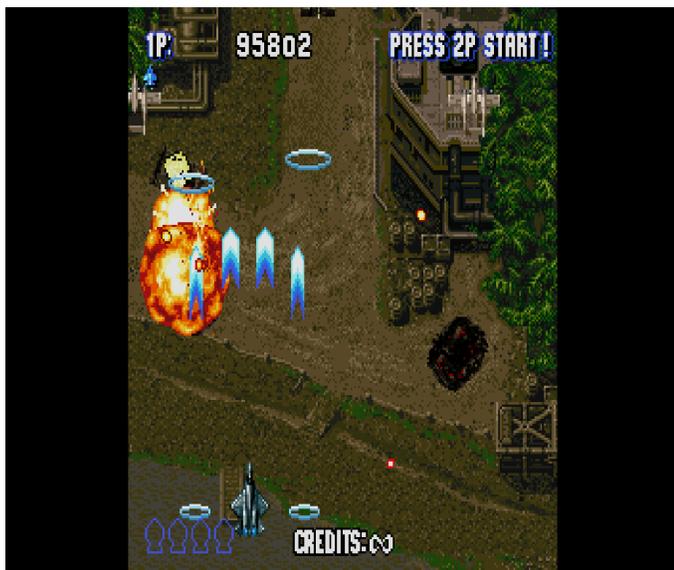


Figura 18 – Buenos gráficos y explosiones, pero sin desplazamiento ni transparencias en este juego

La falta de diferentes armas se compensa con una gran cantidad de aeronaves que puedes pilotar: puedes seleccionar entre 10 diferentes aeronaves, cada una con sus propias armas y ataques especiales. Otro añadido es que el botón C actúa como un botón de disparo turbo. Esto significa que no tienes que presionar el botón de disparo una y otra vez.



Figura 19 – Muchos y diferentes fondos y enemigos

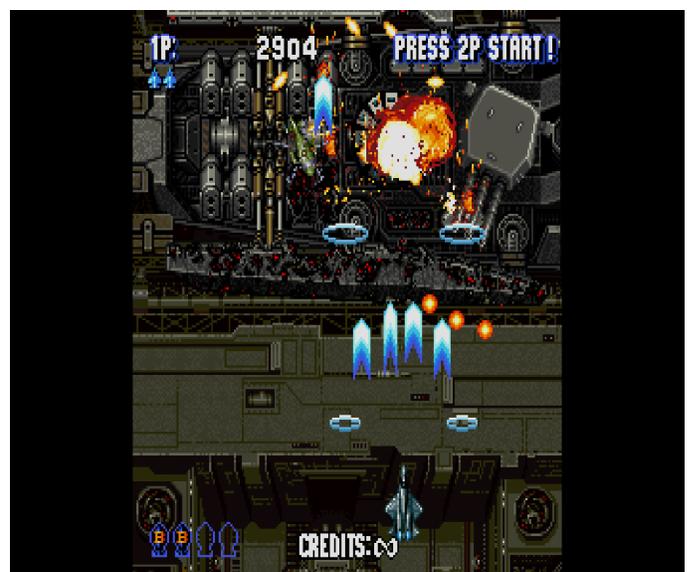


Figura 20: Muchos y exclusivos jefes, así que guarda mejor tus ataques especiales para cuando luches contra ellos.

Incluso si no mueres, tus armas perderán su nivel de potencia si no consigues más potenciadores con el tiempo, y una vez que te golpean, pierdes tus poderes y normalmente te dejan varado con un solo poder con tu aeronave prácticamente destruida. A veces es mejor dejar que los potenciadores vuelen por la pantalla durante un tiempo antes de recogerlos.

Steam Hearts

Aquí tienes otro shoot 'em up para Sega Saturn. Steam Hearts es una vez más un Shmup bastante bueno para Saturn, con un cierto nivel de desplazamientos y muy buenas animaciones.



Figura 21 – Portada de Steam Hearts en Sega Saturn

Lo que me gusta de este juego es que con un solo golpe no necesariamente significa que vas a estar fuera de juego. Tu nave puede recibir un par de golpes antes de caer empicada, lo cual es bastante bueno para variar.



Figura 22



Figura 23 – Una de tus armas es una espada gigante que puede balancear para desviar las balas.

Tienes un par de armas primarias y secundarias diferentes que puedes recopilar y hasta tres armas secundarias que tienen estilos diferentes. Esto te proporciona una amplia variedad de opciones a la hora de combinar tus poderes. Estas armas secundarias también se usan para tu ataque especial y serán “empleadas” precisamente para esto, lo que significa que perderás estas armas secundarias tras recurrir a tu ataque especial. Los ataques especiales son normalmente formatos más potentes de tu ataque principal y no como en muchos otros juegos una enorme pantalla que se llena con un ataque extremadamente poderoso. Por ejemplo, si tienes un láser auto-dirigido como ataque secundario y activas el ataque especial, la velocidad y fuerza de éste aumentarán durante un breve período de tiempo, y la espada aumentará su tamaño ocupando casi toda la pantalla. Este tipo de cosas pueden ser extremadamente útiles, pero no devastadoras, como ocurren en otros juegos en los que puedes matar a un jefe con solo uno o dos ataques especiales.



Figura 24 – El láser es probablemente el arma primaria más fuerte

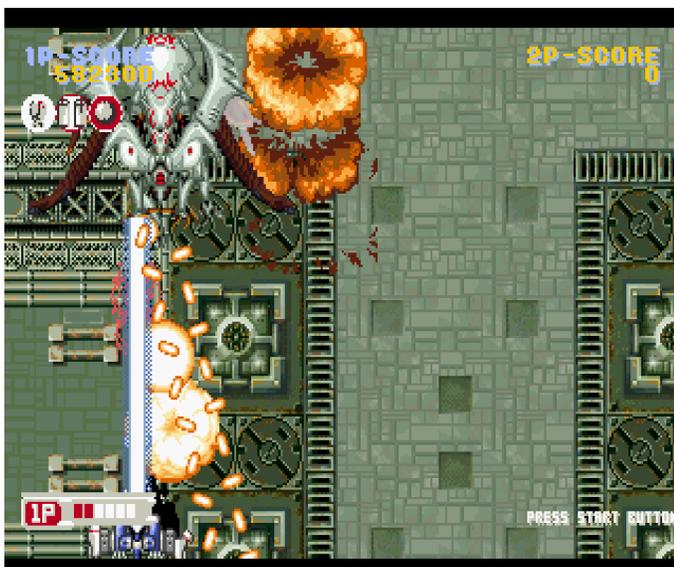


Figura 25: Matar a un jefe con el láser es mucho más rápido que con otras armas

En definitiva, es un shooter bastante conseguido. Puedes optar entre un personaje principal masculino o femenino, y una nave azul o roja.

Strikers 1945

Strikers 1945 es el último shoot 'em up de esta serie. Salió para muchos y diferentes sistemas, incluso hay una versión PSP del juego.

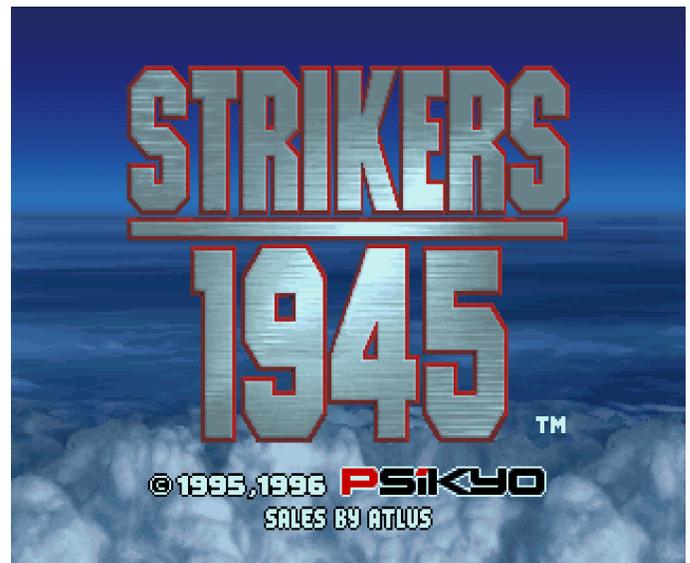


Figura 26 – Strikers 1945 para Sega Saturn ejecutándose sobre un ODROID-XU3



Figura 27 – Seleccionando tu avión favorito, cada uno tiene diferentes armas

El juego como tal no es nada del otro mundo, ya que no tiene desplazamiento paralaje, aunque los gráficos y las animaciones son bastante buenos, pero nada que no se pueda ver en otros juegos. Aun así, es bastante divertido, el nivel es corto y te enfrentas a un nuevo jefe en cada etapa que aparece de forma aleatoria. Sin correr es lo mismo, ya que el orden del nivel cambia cada vez, lo cual te proporciona un añadido para repetir el nivel. Los niveles son interesantes y los planos son lo suficientemente diferentes como para que quieras probarlos todos. Strikers 1945 puede que no sea el mejor shmup, pero lo hace todo muy bien. El sistema de juego es rápido y desafiante, pero no malintencionado. Los gráficos son

bastante buenos, incluso si no utilizas el desplazamiento paralaje. Tienes buena música, un video de introducción, continúes ilimitados, disparos automáticos con el botón derecho superior del mando y mucho más.

En general, Strikers 1945 es un shooter muy conseguido y funciona muy bien con el ODR0ID-XU4/XU3. Si te gustan los shoot 'em ups, definitivamente deberías probar este juego.



Figura 28 - Boss Battle in Striker 1945



Menciones honoríficas

Saturn Bomberman Fight!!

Este juego es un tanto interesante si lo comparamos con juegos del estilo Bomberman. No mueres con la primera bomba que estalla a tu lado, aunque sí que tienes una barra de salud. Puedes lanzar bombas,

saltar cajas, montar animales y mucho más. El juego es bastante diferente del Bomberman original, pero es bastante divertido. Además, lleva a Bomberman y a sus amigos a un mundo 3D, lo cual proporciona al juego un pequeño y agradable cambio de estilo.

Shichisei Toushin Guyferd - Crown Kaimetsu Sakusen

La primera vez que ves la introducción, instantáneamente te recuerda a una vieja serie de televisión llamada Power Rangers, pero no, este es Guyferd, que es similar, aunque diferente en muchos aspectos. Desafortunadamente, no entiendo ni una palabra de lo que se dice, lo cual es una lástima, ya que el juego incluye muchas escenas de videos probablemente de la propia serie de TV (¡y me refiero a MUCHAS!). Juegas como un crawler de mazmorras, donde caminas desde la perspectiva de la primera persona y giras a la izquierda o a la derecha y caminas de un lado a otro, hasta que encuentras elementos o un enemigo, que es cuando empiezas a luchar. El juego es muy interesante, así que es una pena que entienda tan poco.

Shining Force III

Si quieres jugar a este juego para Sega Saturn, será mejor que insertes una Memory Backup Card para guardar tus progresos del juego, ya que la memoria interna de Saturn probablemente no sea suficiente. Realmente no se puede disfrutar de los gráficos pre-renderizados como en el caso de Donkey Kong para SNES o como Shining Force III. Aún así, es un juego de rol muy interesante que utiliza muchos elementos y efectos 3D, y es uno de los mejores juegos 3D para Saturn. Es formidable y por lo que he escuchado, se supone que es un juego muy "épico" y el mejor de la serie. Se ejecuta un poco lento, con mucho texto y poca acción al principio. No he tenido realmente tiempo para jugarlo en serio y poder ofrecer una buena crítica. Ha sido bastante elogiado, así que creo no equivocarme al recomendarlo.

Silhouette Mirage

Silhouette Mirage es un juego bastante interesante que no termino de entender del todo. Puedes correr, saltar y disparar bolas de energía con los dedos, y

tienes que vencer a Mirages y Siluetas. Hay muchos disparos, golpes y cosas que puedes hacer. Incluye un tutorial que nunca he llegado a terminar, ya que te lleva mucho tiempo, y simplemente te muestra los diferentes combos de ataque que puedes hacer. Los gráficos son agradables y la música es pegadiza. Es un buen juego, aunque no he llegado entender muy bien lo que ralmente estoy haciendo.

Street Fighter Alpha - Warrior's Dream

Este juego también puede necesitar una tarjeta de memoria si quieres guardar tus buenas puntuaciones, o tienes que asegurarte de que la memoria del sistema tenga al menos 32 bloques libres. El juego es lo que se suele esperar de un juego al estilo Street Fighter. Tiene buenos gráficos, buenas animaciones y por lo general tiene buena calidad, pero no es algo extraordinario.

Street Fighter Zero

No he logrado hacer hacer funcionar Street Fighter Zero 3, aunque es una versión mejor que la versión Zero, y con la expansión de memoria de 4MB, ofrece bastantes más animaciones. Sin embargo, Street Fighter Zero es un excelente juego de lucha de la serie

Street Fighter con buenos gráficos y animaciones. Hay muchos movimientos especiales, y se ejecuta muy bien en el ODROID XU3/XU4. Si eres fanático de la saga Street Fighter, definitivamente deberías echarle un vistazo a éste.

Street Racer

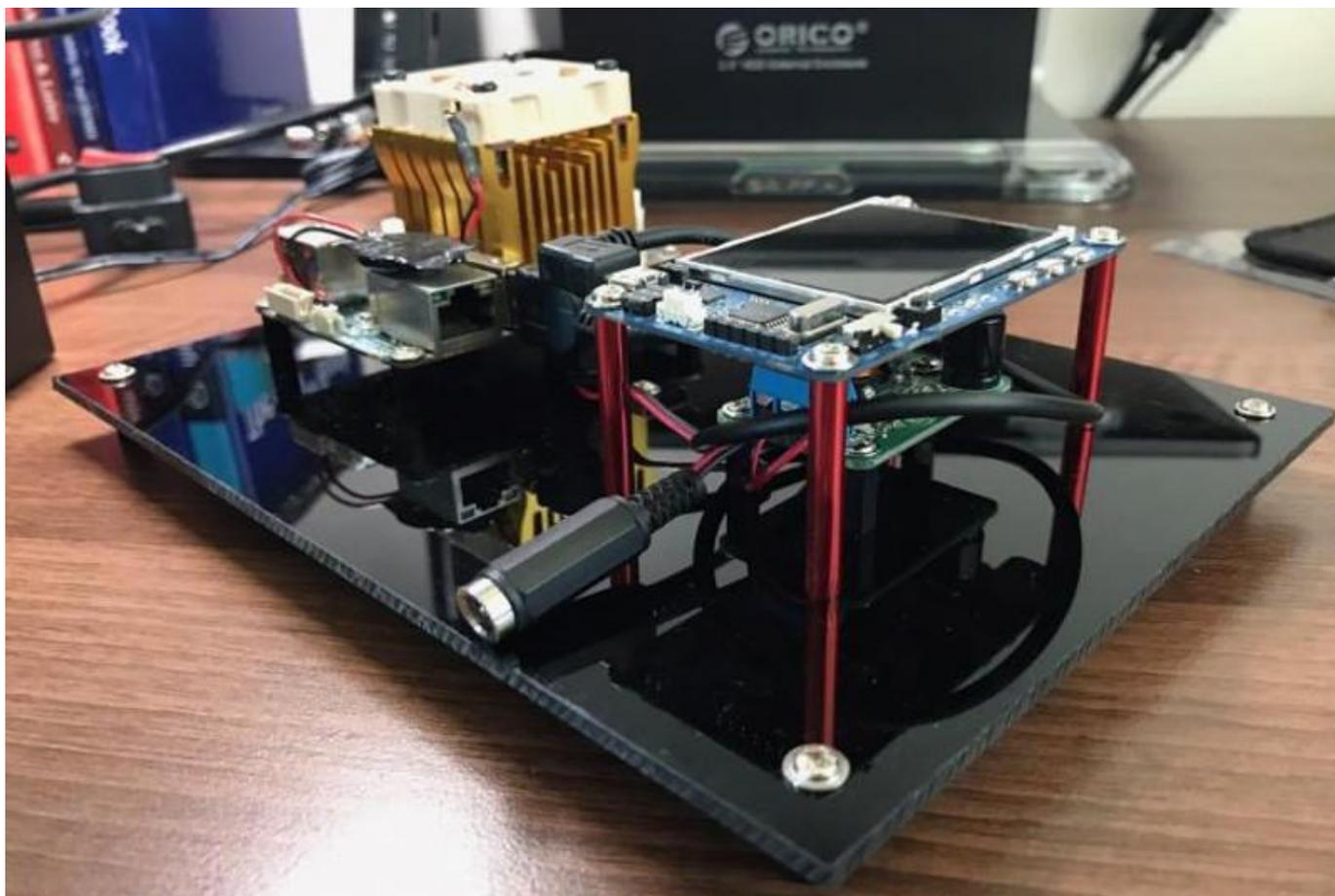
Street Racer es un pequeño e interesante juego de carreras con divertidos gráficos de dibujos animados similares a Mario Kart. Tiene algunos problemas gráficos, ya que parece que el cielo está realmente en el suelo y parece que conduces sobre él. Esto hace que la pista real sea difícil ver, pero no imposible, y aún así puedes disfrutar bastante del juego. Otra ventaja es que con este juego puedes gestionar la memoria del sistema y los estados de guardado de la memoria de backup.

Striker 1945 2

Striker 1945 2 es la segunda entrega de la serie 1945. Es un shoot'em up muy bueno, pero desafortunadamente es un poco lento. El frame skipping resulta un poco molesto, y la desactivación del mismo te permitirá jugar a cámara lenta, que se puede soportar, pero ya no resulta tan divertido.

Servidor Doméstico ODROID-XU4

© June 1, 2018 By Igor Kromin ↳ ODROID-XU4, Mecanico



En diciembre de 2017, monté mi propio servidor multimedia doméstico Odroid XU4 (<https://goo.gl/6tT6rt>) ya que tuve algunos problemas con la configuración anterior. Lamentablemente, en este montaje no tuve en cuenta ni la estética ni el cableado, solo me centré en la funcionalidad, porque necesitaba un servidor funcional y no me importaba su apariencia. En marzo de 2018, recibí mi monitor curvo con una pantalla de 34 pulgadas (<https://goo.gl/WekKF9>) y con ello el tema del cableado y la estética del servidor en general tomaron importancia, así que decidí hacer algo al respecto.

No he utilizado una carcasa totalmente cerrada, ya que me gusta el estilo de tener los distintos componentes montados en una sola lámina de acrílico, lo cual le da un aspecto más industrial. Me gustaría tener una lámina superior transparente, pero no he sido capaz de encontrar soportes que sean lo suficientemente altos como para soportar una.

El tema que he intentado resolver en este punto era el del cableado. Contaba con dos fuentes de alimentación independientes: una para el HDD y otra para el XU4. Esto no era lo idóneo, especialmente porque tenía que usar un adaptador de alimentación internacional con la fuente de alimentación XU4 (de EE. UU. a Australia). Demasiado desorden, quería usar una única fuente de alimentación, pero necesitaba dos voltajes: 12V para HDD y 5V para XU4.

Mi solución fue la de usar una fuente de alimentación de 12V para el HDD y hacerme con un convertidor DC-DC de 12V a 5V (<https://goo.gl/gKjnzG>) para alimentar el XU4. La fuente de alimentación era capaz de proporcionar suficiente corriente para ambos. Esto me resolvió el problema del voltaje. Luego, tenía que llevar los 12V al convertidor DC-DC. Dado que contaba con un cable divisor de alimentación de la base HDD que podía alimentar el ventilador de refrigeración y el dock en sí, opte por reutilizarlo. Esto implicaba volver a conectar el ventilador. Esta parte

fue fácil. Desarmé la base, taladré un pequeño orificio para pasa el cable del ventilador y conecté el ventilador directamente a su toma DC. Resultaba un poco chapucero, pero funcionaba.



Figura 1 - Volver a cablear el ventilador fue fácil



Figura 2 - Otra perspectiva del cableado del ventilador



Figura 3 - El resto de componentes del proyecto antes de montarlos

Ahora que tenía el cableado más o menos resuelto, necesitaba empezar a montarlo todo en su lugar. Como tenía una hoja negra de acrílico de 3 mm, decidí usarla como base (después de cortarla con el tamaño adecuado). Utilicé bastantes separadores de nylon negro (<https://goo.gl/auDiwD>) para colocar el XU4 y el convertidor DC-DC en su lugar. También usé los mismos separadores debajo de la base: uno en cada esquina con un separador montado en el centro.



Figura 4 - Los separadores para el XU4 y el convertidor DC-DC

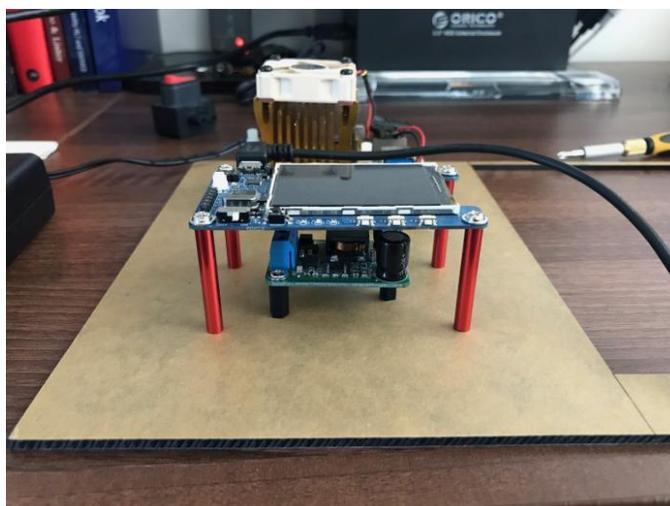


Figura 5 - Otra perspectiva de los separadores para el XU4 y el convertidor DC-DC

Para mantener todo limpio y ordenado, decidí colocar la pantalla sobre el convertidor DC-DC. Para ello utilicé unos cuantos separadores de aluminio rojos de 35 mm (<https://goo.gl/G6JTsv>). La altura era la adecuada.

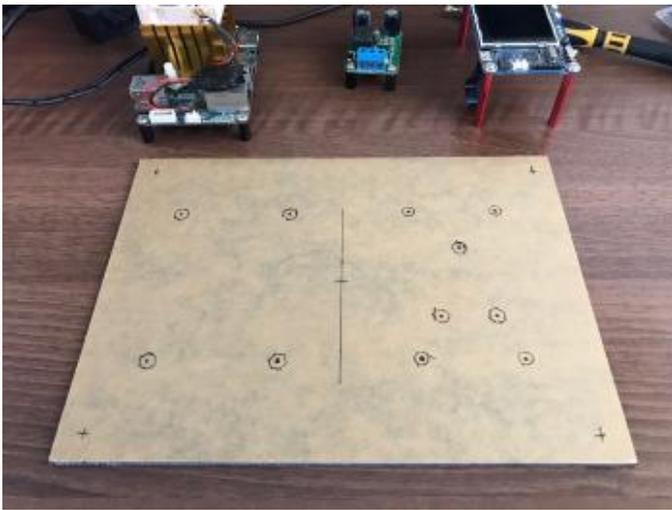


Figura 6 - La lámina acrílica lista para perforarla

Era cuestión de marcar todas las posiciones de los separadores y realizar los correspondientes agujeros.



Figura 7 - Perforando la lámina de acrílico



Figura 8 - Los componentes montados en la lámina de acrílico

Después de colocar el XU4 y el convertidor DC-DC en su lugar. Monté el cableado básico desde fuente de alimentación al convertidor DC-DC y me aseguré de

que todos los voltajes eran los correctos. Una vez verificados los voltajes, terminé con el cableado entre el convertidor DC-DC y el XU4. El conector DC en ángulo (<https://goo.gl/GvN8KY>) fue muy útil en este punto.

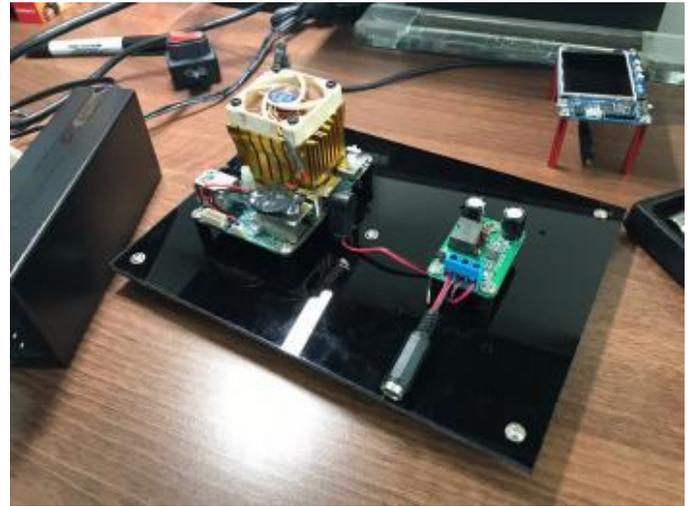


Figura 9 - Los componentes con el cableado montado

Tras conectarlo todo, el servidor estaba listo para echar a andar. Colocado detrás de mi monitor se ve limpio y ordenado, sin cables que puedan apreciarse.



Figura 10 - La vista final del servidor con los cables y láminas de acrílico montadas

El disipador de calor en el XU4 no es el original. Lo reemplacé hace un tiempo tal y como se describe en el artículo "Another Odroid XU4 standard heatsink and fan replacement" Para comentarios, preguntas y sugerencias, visita el artículo original en <https://goo.gl/6TL9BT>. Si desea leer más artículos de Igor, dirígete a <https://www.igorkromin.net/index.php/category/odroidxu4/>.

Carputer – Pantalla Táctil Android de 7”

© June 1, 2018 By Steve Jackson (@zurkeyon@gmail.com) Android



Este es un CarPuter Android de pantalla táctil de 7” con GPS USB muy preciso, Bluetooth 4, entrada y salida de audio de 3.5MM, WiFi y una pantalla magnética ajustable. Tienes disponible un archivo sketchup para personalizarlo y cambiar el tamaño según sea necesario en <https://www.thingiverse.com/thing:2720349>. Como técnico de Servicio Locales TI de 28 años, he montado esta unidad por necesidad para reemplazar la pequeña pantalla del teléfono y que me sea de más utilidad a la hora de viajar. Existen unidades prefabricadas que puedes comprar y lograr la misma tarea, pero ya tenía la pantalla Vu7 y ODROID-XU4 de otro proyecto, así que opté por readaptarlos en algo que pudiera serme de más aplicación

Ten en cuenta que algunas de las siguientes aplicaciones/funciones son ilegales usarlas mientras se conduce. No puedes usar aplicaciones que te permiten escribir mientras conduces un vehículo a motor en la mayoría de los estados. Comprueba tu

legislación y no la violes al utilizar este dispositivo. No me hago responsable del uso o aplicación inadecuada de este dispositivo y de la posible violación de la legislación aplicable.

Esta versión del dispositivo es solo para actividades educativas/mecanico. Apártate a un lado del camino, detente sin ocasionar daños y luego utilízalo. Por lo general, me encuentro estacionado antes o después de activar un servicio cuando uso este dispositivo, o me detengo y me pongo a salvo, luego hago lo que tengo que hacer. Sé inteligente cuando uses este dispositivo, y no vaya a sufrir un accidente por utilizar lo que básicamente es un juguete friki con algunas aplicaciones de negocios.



Este proyecto carputer es una forma muy económica de modernizar tu vehículo

GPS

Las unidades GPS USB son mucho más sensibles y precisas que las incorporadas en tu teléfono. Esto implica un gran ahorro de tiempo, ya que me permite anticipar mejor mis giros y evitar esperas innecesarias y dar vueltas en U. También localiza instantáneamente los satélites sin esperas.

Portal RepairShopr

Utilizo este calendario/portal para ver mis citas en tiempo real y para colocar la dirección del cliente en la cola de visitas del GPS. Cero escribo y cero hablo con Google. Con un teclado Bluetooth RII, también puedo sentarme y añadir notas del sitio en la intimidad de mi camioneta mientras escucho las noticias o música.

Musica

Uso Spotify, Pandora y multitud de servicios de música gratuitos para proporcionar sonido a mi camioneta. Todos ellos se integran muy bien con la configuración permitiéndome controlar otros programas, así como también los botones de la unidad principal de sonido del automóvil.

Panel de Mandos

El panel de mandos incluye un velocímetro digital (GPS), fecha, hora, clima, controles de audio e incluso un visualizador de audio, todo esto es posible gracias a Car Launcher Pro.

WAZE

WAZE es un buscador de zonas de control de velocidad, policías y accidentes con información de tráfico en tiempo real, un GPS alternativo a Google Maps y mucho más. Éste incluso te muestra los límites de velocidad con muy buena precisión, aunque no siempre son correctos, así que estate atento a las señales de las calles.

Torque PRO

Puedes emparejar Torque PRO con un ELM327 o con un escáner Bluetooth OBD-II similar y acabarán teniendo un sistema de información del motor en tiempo real. Cuenta con una muy completa gama de indicadores digitales que se pueden personalizar con muchas configuraciones diferentes. Para los coches con o sin Turbo, hay indicadores de potencia y de vacío, indicadores de aceleración e incluso lee los códigos del motor de la mayoría de los automóviles y camionetas OBD-II. Existen otras aplicaciones que también hacen esto, pero Torque Pro, que es una aplicación de pago, funciona perfectamente y cubre todas mis necesidades. Es agradable y muy estable en las imágenes de Android para ODROID-XU4.

Probando los límites del Wifi y su alcance

Las pruebas WiFi de un edificio amplio para comprobar la cobertura inalámbrica, como es un complejo de apartamentos o un hotel, pueden llegar a ser un poco monótonas si tienes que caminar. WiFi Analyzer es una gran utilidad que proporciona un ping audible para comprobar la potencia del WiFi. Te permite recorrer toda una propiedad y trazar rápidamente los límites del WiFi sobre el plano de un lugar, revelándote las brechas de cobertura con bastante facilidad, todo ello desde la comodidad de tu aire acondicionado, lo cual es genial cuando se está aquí en la soleada Florida. Esta herramienta me permite estar online las 24 horas del día con algo mucho más útil que la pequeña pantalla de mi teléfono. Cuando mi teléfono está cerca, mi punto de acceso se conecta y listo.

Usos adicionales

- Señalización LED digital, usando Neopixels con la apropiada configuración de batería para ejecutarlos

- Controles de LED personalizados, luces de neón y otra iluminación para los efectos y las luces
- Interfaz de sistema de combustible personalizado y controles No2
- Vehículo para investigador/detective privado e interfaz de seguimiento GPS para las investigaciones
- Cámara de respaldo o cámara del panel de instrumentos
- Estación base de drones FPV desplegable para inspecciones y seguridad de lugares/edificios
- Sistema de películas y videojuegos con reposacabezas trasero para entretener a pasajeros y niños
- Informe de carga EV y motor eléctrico, AC, regenerativo y otros controles

Componentes

- Ordenador ODRO ODROID-XU4 con eMMC de 32 GB + batería RTC para mantener el registro del tiempo: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825, http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145628097465, http://www.hardkernel.com/main/products/prdt_info.php?g_code=G137508214939, un disipador de calor con ventilador que se recomienda encarecidamente para el XU4 en esta configuración.
- Pantalla Táctil ODROID-VU7 + 5v: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G144549683088
- Hub USB alimentado con 4-5 puerto: mi versión incluye un cable de alimentación modificado para alcanzar una potencia de 5v desde el convertidor DC-DC (<https://goo.gl/WZmPUA>)
- Cable de extensión USB opcional de medio metro (<https://goo.gl/GDQzkv>)
- Fuente de alimentación de 12v a 5v para alimentar ODROID-XU4 y el HUB USB con alimentación (8 AMP+) <https://goo.gl/mwT1ng>. También necesitarás un ventilador de 12V de 40 mm x 40 mm x 10 mm para enfriar hasta 10-12 amperios
- Adaptador WiFi Nano USB ODROID: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145431421052
- Adaptador Bluetooth 4 USB ODROID: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G141656580698
- Un pequeño Cable HDMI, lo más delgado posible, para el video de la pantalla: El cable HDMI incluido con el

ODROID-CU7 es demasiado grueso (<https://goo.gl/HuQdFy>)

- Un Micro cable USB a USB, tan delgado como te sea posible, para la función táctil de la pantalla: La pantalla ODROID-VU7 viene con este componente.
- Imanes magnéticos K y J redondos de neodimio (2), para mantener la pantalla en su lugar: <https://goo.gl/SeuHpM>
- Material ABS, del color que quieras: <https://goo.gl/nYJPB4>. Funciona muy bien, aunque lo barato sale caro. Estoy seguro de que podría haber encontrado algo mejor.
- Cable par trenzado de un tamaño decente. No excesivamente pequeño, ya que se moverán más de 5 amps.
- Tarjeta de sonido ODROID USB 5.1 con línea de entrada, la del micrófono para que puedas hablar con Google: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G141629380416
- Unidad GPS USB Gmouse, compatible con ODROID: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G142502154078
- Conector DC para ODROID-XU4: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G141440511056
- Micrófono para PC <https://goo.gl/YQGT9p>
- Esmalte de uñas Sally Hansen Insta-Dri: bueno para darle a la pantalla un acabado brillante, el cual está disponible en Walgreens y tiendas similares.
- Un interruptor de encendido/apagado

Utiliza una combinación de hardware M2 y M3. También disponía de unos cuantos tornillos de rosca gruesa que procedían de otro proyecto, con tamaño similar al hilo grueso M2. No obstante deberías poder ver lo que realmente necesitas partiendo del tamaño de los agujeros.

Próximas funciones

Encendido/apagado automático al arrancar el vehículo: Desafortunadamente, no he tenido suficiente tiempo ni la curiosidad por utilizar este circuito e implementarlo en el diseño. ¡Cualquiera que esté interesado puede personalizar uno y mostrarnos cómo lo ha hecho!

Botones complicados/control del joystick: estaba considerando la posibilidad de utilizar una palanca

con un botón para los controles más complicados, pero no he logrado decidir dónde colocarla.

Double Din, versión completamente independiente con amplificador integrado de 4 canales, con línea de entrada/salida: sistema de sujeción adicional, para facilitar su montaje en el hueco. La combinación de cuñas y tornillos en este punto dificulta la entrada y la salida.

Controles de iluminación reactivos al sonido o HAT: Esto realmente haría resaltar este dispositivo sobre el resto de cosas. Ninguna unidad de Android comercial puede hacer esto sin los componentes adicionales. Un HAT permitiría que el Xu4 fuese un verdadero panel de instrumentos personalizado para el coche.

Impresión

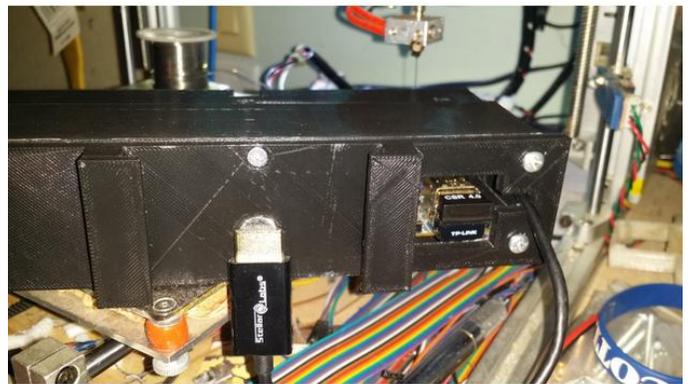
Utiliza ABS o PETG para lograr mejores resultados. Espera algunas deformaciones, al trabajar con los bordes de los extremos de tu superficie de impresión y problemas de separación en las capas. Un recinto caliente funciona muy bien. Use cartón, papel de corcho, papel de aluminio y una lámpara de calor o luces de seguimiento; simplemente ten cuidado con el calor y no dejes de vigilarlo.



Carcasa Carputer para la pantalla



Vista lateral de la carcasa del ordenador



Vista posterior de la carcasa del ordenador

Montaje posterior a la impresión

Tras imprimir las partes, conecté el módulo eMMC en el ODROID e instalé el sistema operativo Android. Haz un recorrido de prueba sin instalar la unidad en el vehículo y asegúrate de que todo funciona correctamente. Recomiendo ejecutar la unidad durante varias horas para asegurarse que no hay problemas de calor con la fuente de alimentación.



Pantalla instalada en la carcasa del ordenador

La fuente de alimentación DROK me fue mejor cuando estaba configurada a unos 5.7v. Se sabe que el VU7 parpadea ligeramente cuando el XU4 trabaja al 100% debido a la caída de corriente en el riel de 5v. El HUB intenta solucionar esta cuestión, pero tienes que subir un poco la energía. 5.7 fue mi "punto óptimo" permitiéndome reducir considerablemente cantidad de parpadeos.

¡Las configuraciones hace trabajar a las CPUs al máximo! Haz que se caliente todo hasta no poder más y ponle al XU4 tanta carga de trabajo como te sea posible. Los programas para probar el rendimiento funcionan bien para esto. No querrás meterle fuego al salpicadero de tu coche. Si alguno de los componentes se calienta demasiado hasta el punto de sentirlo excesivamente caliente al tocarlo, es que está funcionando a demasiada temperatura como para incluirlo en el salpicadero, lo cual significa que debes añadir ventiladores.



Abriendo el salpicadero

Busca sitios para que el hub USB y la fuente de alimentación no entren en contacto con la puesta a tierra de la carrocería, o podrías tener problemas con los cortocircuitos. Asegúrate de que todos los componentes estén debidamente acotados con suficiente ventilación con el objeto proteger los componentes electrónicos. La estructura Din se atornilla al marco, o se pueden usar pestañas con rosca ajustándola en su lugar de esta forma.

Disponer de una línea hacia un AUX u otra fuente sigue siendo un problema en este proyecto. Aún estoy buscando una salida AUX limpia para el ODROID-XU4 mientras esté conectado a los 12 voltios del coche. De momento, la unidad sólo es capaz de proporcionar audio limpio a través del Bluetooth a la unidad principal del coche.

Algunos han sugerido probar el [conector de salida de 3.5 mm](#) en el Boombonnet, y simplemente dejar los pequeños altavoces fuera. Siendo I2S, es posible que no experimente tanto ruido. La fuente de alimentación que utilice para ello inyectaba un 50% de ruido en la señal de audio, de modo que era totalmente inservible.



Ordenador instalado en el salpicadero

Los filtros de ruido y de puesta a tierra recortan las frecuencias de los graves y otras frecuencias, y distorsionan el sonido en cualquier sistema que usará un amplificador y subwoofer. Créeme. Los probé todos. Ninguno llegó a sonar bien después de conectarlos. El ruido desapareció, pero también lo hizo el 50% de mis frecuencias de sonido.

Consideraciones adicionales

Ten mucho cuidado al usar imanes de neodimio en cualquier proyecto. Son muy conocidos por unirse y romperse, causando que el metal afilado salga volando a gran velocidad, posiblemente cegando al usuario, ¡Así que ten cuidado! Asegúrate de que tus imanes estén orientados de la manera correcta antes de colocarlos en la posición final, o los estarás forzando y probablemente se harán añicos en el proceso.

Todos los intentos de imprimir esto en PLA terminaba en fracaso a los pocos días. Se deformaba dentro del automóvil debido a las altas temperaturas que alcanzaba el salpicadero No te molestes en intentarlo; no funciona.

Para imprimir los componentes y para obtener más detalles, visita el artículo original en <https://www.thingiverse.com/thing:2720349>.

Introducción a los Conceptos Básicos de BASH – Parte 2: Comandos BASH Útiles para Ordenadores de placa Reducida

© June 1, 2018 By Erik Koennecke Linux, Tutoriales



La última vez aprendimos los comandos 'ls' y 'tree'. Aunque analizar cosas puede ser interesante, resulta más divertido en realidad hacer algo con nuestros propios datos. Este artículo contiene una lista de los comandos más comunes a la hora de manipular datos. El comando y su explicación son muy breves con el fin de evitar escribir una enciclopedia de Linux y que llegue a aburrirte hasta cansarte. Simplemente debería llamarte la atención la existencia de un determinado comando BASH.

A modo de recordatorio, cada vez que queramos conocer mejor algo, escribiremos "man" en una ventana de Terminal y nos devolverá la sintaxis de uso, así como todas las opciones disponibles para ese programa en concreto. Ten cuidado, sin embargo, porque las páginas man a menudo tienen contenido que se ha ido acumulando durante décadas y pueden ser difíciles de digerir, especialmente para los

principiantes. Si quieres comenzar a trabajar de inmediato, abre www.explainshell.com en tu navegador, aquí encontrara explicaciones de los distintos comandos, como los ejemplos que vamos a ver a continuación. Para obtener información útil sobre cómo crear tus propios comandos, usa las páginas TLDR en tldr.oostera.io. Compara el manual del comando tar con la versión TLDR y verás a qué me refiero.

Comandos básicos de BASH

Los comandos básicos más utilizados son:

- cd – cambiar de directorio. 'Cd /' cambia al directorio raíz, 'cd ~' vuelve al directorio de inicio del usuario.
- mkdir – Crear un directorio. Prueba con 'mkdir ~/test' para crear un subdirectorio dentro de tu carpeta de inicio llamada prueba.
- cp – copiar archivos. 'Cp /path/to/file /home/odroid/test/' genera una copia en el directorio

que hemos creado antes. Con la opción -R, puedes copiar archivos y directorios de forma recursiva.

- mv – mover archivos. La sintaxis es igual que cp, pero el archivo se elimina una vez que la copia se haya realizado correctamente.
- rm – eliminar archivos.

```
$ rm exampleFile.txt
```

- cat – concatenar archivos. cat lee desde entradas estándar y escribe resultados estándar si no se proporcionan otras opciones. Si usas el archivo cat, se escribe el contenido del archivo en la consola, el resultado estándar. Pruebe con 'cat/etc/fstab' para ver un ejemplo.
- less – si deseas leer archivos más largos en consola. 'Less filename' muestra el contenido del archivo y hace una pausa después de cada página. Puede avanzar con "espacio" y regresar con "b" y "q" para salir.

```
$ less exampleFile.txt
```

- head y tail – por defecto, el archivo head muestra las primeras 10 líneas del archivo, mientras que la tail de un archivo muestra las últimas 10 líneas del archivo.

```
$ tail exampleFile.txt  
$ head exampleFile.txt
```

- tar y zip – para archivar archivos. tar cf archive.tar /home/odroid/test/ archiva el directorio de prueba y todos sus contenidos. Si añades el parámetro z (tar czf), el archivo se comprime posteriormente.
- sudo y su – para todo lo que requiera más privilegios, antepones sudo para ejecutar el comando como root, el superusuario. cat/etc/sudoers no funcionará. Si usas sudo cat /etc/sudoers, puedes leer el archivo. su permite 'cambiar de usuario' Si ha asignado una contraseña al usuario root o tienes otros usuarios en el sistema, puedes pasarte a ellos con este comando. Con sudo -i, también puede cambiar a un prompt de comandos con más privilegios y trabajar como root hasta que te salgas.

```
$ sudo -i
```

- exit – para salir del prompt de comandos o del shell. Útil también para conexiones remotas.

```
$ exit
```

- ssh – puedes iniciar sesión en una consola remota de tu ODROID con esto. Aquí es donde realmente destaca la consola BASH. Siéntate frente a tu PC, inicie sesión en el ODROID donde sea que esté y trabaja como si estuviera sentado frente a él. En este caso, utilizar una GUI es más bien un inconveniente. Ésta es la razón por la cual la consola y BASH en tu SBC son más importantes. En un PC de escritorio, es posible que puedas evitar la línea de comandos la mayor parte del tiempo. Para conectarse por ssh, necesitas la dirección IP del ordenador o del ODROID y el nombre de usuario con el que desea iniciar sesión. Por ejemplo, el usuario 'odroid' y la IP '192.168.0.1'.

```
$ ssh odroid@192.168.0.1
```

- grep – muestra los archivos que contienen un cierto patrón de texto. ¿Desea saber qué archivos del directorio /etc/ contienen la palabra 'Ubuntu'? grep -rnw '/etc/' -e 'Ubuntu' te da la respuesta.

```
$ grep -rnw '/etc/' -e Ubuntu
```

- find – Puedes usar find para buscar archivos usando casi todos los criterios que puedas imaginar y actuar sobre ellos. El ejemplo más simple: find/home/ -name '* test *' busca todos los archivos y directorios que contienen 'test'. find / home/odroid/ -type f -size +1M -mtime 30 es un ejemplo más elaborado. Encuentra todos los archivos (no directorios) en /home/odroid que ocupen más de 1 MB y que hayan sido modificados en los últimos 30 días. También puede usar Explainshell.com para averiguar lo que hace.



Figura 01 – explicahell.com explicando nuestro comando find

- `sort` – ordenar los archivos.
- `df` y `du` - el espacio libre y el uso del disco muestran cuánto espacio le queda con `df` o cuánto usan tus archivos y carpetas individuales con `du -ch`.
- `ps` y `kill`, `killall` – `ps` lista los procesos en ejecución, si quieres ver todo lo que hay funcionando, usa `ps aux`. El número de proceso utilizado con `kill`, así que `kill`, matará el proceso al enviar una señal para que finalizar el proceso. `killall` mata todos los procesos.

```
$ ps aux
$ kill
```

- `mount` y `umount` – monta dispositivos externos y unidades con `mount /dev/ /path`, desmóntalos con `umount /dev/`.
- `dd` – con `dd`, puede escribir desde cualquier lugar a cualquier lugar. También se conoce como destructor de disco por alguna razón, ¡así que ten cuidado! `dd if=/dev/mmcblk0 of=/home/odroid/mbrbackup bs=512 count=1` hace una copia del registro de inicio maestro. Muy práctico cuando quieres grabar una nueva tarjeta SD.

```
$ dd if=/input/path of=/output/path
```

- `shutdown` y `reboot`: apaga y reinicia tu SBC. Hay varias opciones disponibles

Ampliaciones de BASH y comodines

- `*y?` – `ls *.txt` muestra todos los archivos que terminan en `.txt`. Usando `ls Image??.jpg`, puedes mostrar los archivos con nombres como `Image00.jpg` o `ImageAB.jpg`.
- `[abc]` – `ls /dev/sd[abcd]1` muestra las particiones `/dev/sda1`, `/dev/sdb1`, `/dev/sdc1` y `/dev/sdd1`.
- `|` como selector – Con `mount | grep 'mmc|sd'`, muestra únicamente las eMMC, tarjetas SD o dispositivos externos montados. Se extiende a 'todo con mmc o sd en el nombre'.

Comandos BASH avanzados – redirección, flujo de datos y bucles

- `stdin`, `stdout` y `stderr` – estos tres archivos (recuerda que todo en UNIX es un archivo) siempre existen. `stdin`

es el teclado, `stdout` es la pantalla y `stderr` son los mensajes de error que aparecen en pantalla.

- `|` – Puedes canalizar el resultado de un programa en el siguiente. El resultado del ejemplo anterior con `find` es mucho más fácil de leer si usas `find /home/odroid -type f -size + 1M -mtime -30 -exec du -h {} ; 2>/dev/null | sort -n`. El resultado de `find` es usado por `du` para mostrar los tamaños, y todos los errores de `stderr` van a `/dev/null`, el receptor para datos innecesarios. Una vez más, si pegas este comando en explainhell.com, verás una vista desglosada de lo que hace cada parte.

```
$ find /home/odroid -type f -size +1M -mtime -30 -exec du -h {} ; 2>/dev/null | sort -n
```

- `/dev/zero` y `/dev/null` son la fuente para los bytes nulos y el receptor para todo lo que no necesitas.
- `>` y `<`, `>>` y `<<` - Estos redirigen la salida o la entrada a archivos. `ls *.txt > texts.list` genera una lista de archivos de texto en el directorio actual. `ls Documents/ *.txt >> texts.list` añadiría la lista de archivos `.txt` al directorio de Documentos. `2>` es el resultado de `stderr`. La parte `2>/dev/null` en el ejemplo anterior redirige todos los mensajes de error fuera de la consola al Big Bit Bucket.

`for`, `do`, `done` – Los bucles BASH se pueden crear con estos comandos.

`if`, `then`, `else` – creación de bucle avanzado, principalmente para scripts.

Ejemplo de aplicación práctica

Hagamos un ejemplo para ver el potencial de shell y de los bucles. La carpeta `Imágenes` contiene varios subdirectorios, cada uno de los cuales deseas comprimirlo en un archivo individual para enviarlos a otra persona. Esto se puede hacer fácilmente con la GUI si solo tiene carpetas de tus viajes con la etiqueta 'EE. UU.', 'Corea del Sur' y 'España': haces clic derecho, seleccionas comprimir y listo.

Ahora piensa en el mismo problema, pero con las carpetas generadas por un programa desde una cámara conectada al ODRROID. Si tiene 100 carpetas o más, esta tarea se vuelve tremendamente repetitiva. ¡Aquí es donde BASH entra en acción!

Si usamos el comando `'ls -F'` para mostrar un directorio, los símbolos se añaden a los nombres de

acuerdo con su tipo. Los directorios tienen una barra añadida, que también es como los ve BASH. Haz memoria: la barra inclinada puede ser parte del nombre de la ruta de un archivo en este directorio, de modo que reciben una barra inclinada:

```
$ for i in */; do zip -r "${i%}/.zip" "$i";  
done
```

Este comando coge todos los directorios que hay dentro del directorio en el que nos encontramos: la parte "for i in */;" Para cada uno de ellos, hacemos lo siguiente: zip -r "\$ {i% /}. zip", donde \$i es la variable i que introducíamos antes, i%/ es el nombre sin la

barra y "\$ {i % /}. zip" es el nombre del directorio con .zip adjunto; el directorio 2018 05 20 daría el archivo 2018 05 20.zip. Las comillas evitan que los nombres con espacios sean tratados como dos nombres diferentes, lo cual podría tener consecuencias desastrosas. Después de que este comando se haya ejecutado para todo i, habremos terminado.

Una breve frase ingeniosa ahorra mucho trabajo tedioso. Esto es lo que hace a BASH tan atractivo. En el próximo artículo, veremos el arranque y el inicio de sesión, y personalizaremos el prompt de BASH.

Juegos Linux: Emulación Nintendo 64 – Parte 2

© June 1, 2018 By Tobias Schaaf Juegos, Linux



En la primera parte de este artículo presentamos la última versión del emulador de Nintendo 64 para Linux y comparamos su rendimiento en todas las placas ODDROID actuales. En esta segunda parte hablaremos de algunos de los juegos más populares de Nintendo 64, como son Mario Kart, Mario Party, Paper Mario, Star Fox, Star Wars, Starcraft, Super Mario, Super Smash Bros y Legend of Zelda.

Mario Kart 64

Mario Kart es una franquicia muy conocida de Nintendo como juego de carreras, que tiene como protagonistas a los personajes más famosos de Nintendo como Mario, Luigi, Peach, Yoshi, Donkey Kong, Bowser y algunos otros. Una de las grandes ventajas de este juego es que se puede jugar con hasta 4 jugadores al mismo tiempo. En realidad, no soy un fan de la serie, especialmente de la versión de Nintendo 64, que gráficamente es muy pobre en mi opinión. Aunque Nintendo 64 es conocida por su

potencial 3D, Mario Kart 64 usa en su mayor parte sprites 2D, que no se ven del todo bien. Los únicos elementos en 3D que presenta el juego son los terrenos por los que conduces y algunos obstáculos y puentes, lo cual hace que el juego sea muy poco atractivo a la hora de jugar.



La primera vez que ejecuté el juego sin frameskip, iba bastante lento. Puesto que el juego usa en su mayoría sprites 2D, realmente me hizo preguntarme por qué necesitaba tanta potencia de CPU. Sin embargo, una vez que activé el frame skipping, empezó a ir bien en el U3. Hay un pequeño retraso en el sonido cuando se usa el menú, pero nada que sea realmente problemático. Las carreras dentro del juego funcionan bien sin retrasos ni ralentizaciones, y el modo multijugador con varios mandos funciona a la perfección.

C1

Aunque el menú es lento, la experiencia dentro del juego es buena y parece funcionar a máxima velocidad con el plugin Rice. Definitivamente es jugable, aunque logras una experiencia mucho mejor en el U3 o XU3. Cuando volví a probarlo con glide64mk2, el juego funcionaba bien, aunque presentaba algunos fallos técnicos con las sombras y las texturas del suelo.

XU3

Mario Kart 64 no tiene problemas con el XU3. Funciona a toda velocidad y puede controlarse fácilmente con un mando Xbox 360.

Mario Party

Mario Party es un tipo de juego de mesa en el que juegas con o contra hasta 4 jugadores en diferentes tipos de minijuegos. El juego es bastante divertido, aunque a veces me resulta difícil entender los controles para ciertos minijuegos. Probablemente sea adecuado para todas las edades, desde niños pequeños hasta adultos como juego de grupo, o simplemente para pasar un buen rato.



Figura 2 - Mario Party

U3

La experiencia con el U3 es impecable, el juego se ejecuta a toda velocidad sin problemas. Una vez observé un parpadeo en la pantalla dividida de un mini-juego, pero en el momento en el que comenzó la acción, desapareció. Por lo tanto, puedo afirmar que es totalmente jugable.

C1 - plugin rice

El menú era un poco lento al principio, pero cuando pasé al mapa para seleccionar un juego, el movimiento se volvió más fluido, como en Mario Kart. Sin embargo, cuando intenté iniciar un juego, sólo vi una pantalla blanca. Escuchaba que todo se ejecutaba en segundo plano, y al hacer clic en los botones se activaban ciertas acciones que podía oír, pero me era imposible ver nada a parte de esa pantalla en blanco. Cuando lo intenté de nuevo con un modo de juego diferente, pude ver algunas partes del juego, pero faltaban otras importantes, y en el momento en que empecé un mini-juego, simplemente obtuve una pantalla negra.

C1 - glide64mk2

Aunque el juego no llegó a funcionar con el complemento rice, sí que lo hizo con glide64mk2, pero iba un poco lento en ocasiones. La mayor parte de las escenas del juego se ejecutan a toda velocidad, de modo que considero que este juego se puede jugar con glide64mk2 a 16 bits.

XU3

El XU3 no tuvo ningún problema al ejecutar este juego. Funcionó sin problemas, lo cual no me sorprendió teniendo en cuenta que también funcionaba en el U3. En general, la experiencia de juego es bastante agradable.

Paper Mario

Paper Mario es una mezcla entre un juego de saltar y correr como Super Mario y un juego de rol como Final Fantasy. Tiene buenos gráficos y aunque el mundo es 3D, Mario en sí está en 2D. En realidad, es una figura de papel. El sistema de juego es único y muy divertido. Es difícil de describir, aunque definitivamente deberías probarlo.



Figura 3 - Paper Mario

U3

La experiencia de U3 con Paper Mario es realmente buena. Me encontré con algunos problemas gráficos en sombras, fuentes y burbujas de dialogo. No podía leer lo que decían las estrellas, aunque valía poco la pena. Sí que podía leer todo lo demás, de modo que en realidad no interfería con el juego. La velocidad en general era muy buena, disfruté bastante del juego en el U3.

C1 - plugin rice

La experiencia en el C1 es difícil de describir. Al principio, el juego no funcionaba en absoluto. Después de una intro bastante lenta, el menú principal no aparecía. Tras otros 10 o 15 minutos, apareció otro tipo de intro, que básicamente era una imagen de fondo desplazándose. 10 a 20 minutos

más tarde, la imagen cambió de nuevo y de repente vi el menú de inicio. Creé un nuevo estado de guardado y comencé un nuevo juego. Una vez más, me apareció una única imagen de fondo. Parece que el juego no funcionaba de ninguna manera en el C1, o podría tardar horas en iniciarse. El C1 debería poder ejecutar el juego a una velocidad bastante aceptable, pero desafortunadamente, los drivers deficientes y la falta de compatibilidad con los gráficos hace que el sistema no funcione correctamente.

C1 - glide64mk2

Este juego funciona con glide64mk2 a toda velocidad. Al igual que ocurre con el U3, tiene problemas técnicos con las sombras y las texturas del suelo, pero a parte de eso, el juego funciona bastante bien.

XU3

El nucleo libretro hace muy buen trabajo con este juego. No se muestran ninguno de los fallos del U3 con glide64mk2. Las sombras son perfectas, las burbujas del dialogos no presentan problemas, y se podía leer lo que decían las estrellas. La velocidad en general también es perfecta. Realmente disfruto jugando a este juego en el XU3.

Star Fox 64

Star Fox 64 es una nueva versión del juego Star Fox de Super Famicom/SNES, que fue uno de los primeros shooters espaciales en 3D. La versión N64 fue famosa por sus muy buenos gráficos y especialmente por sus interpretaciones de voz. Las líneas de comunicación a menudo divertidas de tus compañeros a través de la radio, las intensas batallas y los buenos gráficos hacen que este juego sea realmente divertido.



Figura 4 - Star Fox 64

U3

El juego funciona muy bien en el U3. Aparecen algunas ralentizaciones en el mapa de galaxias donde seleccionas la misión, y las sombras son muy oscuras. La iluminación no funciona del todo bien, lo que significa que el juego es muy oscuro en algunas escenas. A parte de eso, el juego funciona perfectamente a toda velocidad.

C1 - plugin rice

El C1 le va bien este juego. El plugin de video rice se ve mucho mejor a la hora de renderizan las sombras que el glide64mk2 en el U3, de modo que las escenas no son tan oscuras. El rendimiento del C1 es un poco más lento que en el U3, y el informe de la misión presenta algo de retardo. Mientras que el U3 presenta cierta ralentización en el mapa de Galaxia, donde puedes seleccionar tu misión, el C1 se cuelga de mala manera, pero como es solo para seleccionar tu misión, no afecta demasiado al juego. Cuando finalmente te encuentras en plena cacería disparando a todo lo que se menea, el juego funciona a toda velocidad sin problemas, realmente resulta divertido jugarlo en el C1.

C1 - glide64mk2

Similar al U3, la experiencia de juego es bastante buena. La velocidad con la que se ejecuta es la misma que en el U3 y presenta los mismos problemas con las sombras, pero aparte de eso, la experiencia de juego es agradable y solo se ralentiza en el mapa de la galaxia.

XU3

Como de costumbre, la experiencia de XU3 es la mejor. El juego funciona sin problemas, aunque se ralentiza un poco en el mapa de la galaxia. Los gráficos se ven muy bien en el XU3, y el juego funciona sin complicaciones.

Star Wars Episode 1 - Racer

Jugué a este juego hace muchos años en el PC con mi tarjeta gráfica Voodoo 3DFX, que usaba el "deslizamiento" el cual está incluido en algunos de los plugins gráficos de mupen64plus. El juego se centra en el Pod Racer del Episodio 1 de Star Wars. Es un juego de carreras muy rápido con buenos gráficos y objetos que se pueden destruir, puedes actualizar tu pod para hacerlo más rápido o más fácil de manejar. Este juego de hecho usa el paquete de expansión de memoria en el N64 que mejoraba los gráficos, y es compatible con el dispositivo rumble pak de Nintendo. Sin embargo, la versión N64 no es comparable con la versión para PC en términos de gráficos, y también le falta el modo multijugador, aunque sigue siendo un buen juego de carreras.



Figura 5 - Star Wars Episode 1 - Racer

U3

La experiencia en el U3 es muy buena. El juego funciona con fluidez y rapidez, y no parece tener ningún tipo de problema. Algunas sombras son demasiado oscuras, pero es algo que sólo ocurre en el menú.

C1 - plugin rice

Una vez más, el C1 tiene problemas con este juego relacionados con el plugin de video rice, los mismos problemas aparecen en el U3 cuando el plugin de video se cambia a rice. La imagen se distorsiona y corta en algunas escenas. El juego funciona a la perfección usando glide64mk2 a toda velocidad y sin problemas.

XU3

El juego funciona muy bien en el XU3. Finalmente descubrí cómo usar los propulsores, y también vi una opción para dos jugadores. Parece que, si el juego encuentra más de un mando conectado, ofrece una opción multijugador. La experiencia de juego fue impecable y a toda velocidad.

Star Wars: Rogue Squadron

Ha sido catalogado como uno de los mejores juegos de N64 jamás creados, donde haces volar un X-Wing y luchas contra el malvado Imperio. Jugué al juego en el PC cuando salió, y era bastante divertido. Tenía ganas de probarlo en el ODROID. He leído que este juego necesitaba el paquete de expansión de memoria para poder iniciarse. Sin embargo, no importaba lo que intentase, no logre hacer que este juego funcionara en ninguna plataforma ni con ningún plugin de gráficos. Los emuladores mupen64plus y el núcleo libretto se bloqueaban o dejaban de responder.



Pictura 6 – Star Wars: Rogue Squadron

StarCraft 64

StarCraft es un juego de estrategia en tiempo real muy famoso. Es uno de los mejores juegos de

estrategia jamás creados, aún se juega en torneos de juegos profesionales. El juego de Nintendo 64 es un remake muy bueno con gráficos reducidos, videos eliminados y poca música. Es un buen juego de estrategia, y me pareció muy interesante poder jugarlo en un emulador de Nintendo 64.



Pictura 7 – StarCraft 64

U3

El juego funciona sorprendentemente bien en el U3. Presenta algunos problemas de velocidad en el menú, pero tan pronto como empiezas a jugar, funciona bien, aunque el sonido presenta un poco de retardo, especialmente en las batallas más grandes. Puedes escuchar como nueren las unidades después de haber desaparecido de la pantalla.

C1 – plugin rice

StarCraft 64 funciona sorprendentemente bien en C1. Parece que funciona mejor utilizando el plugin de video Rice. Si utilizamos el plugin glide64mk2, el menú se vuelve tan lento que no puedes seleccionar la misión a la que deseas jugar. Por lo tanto, el juego no se puede jugar con glide64mk2. La velocidad dentro del juego probablemente sea la adecuada, pero como no puedes pasar del menú, no hay forma de saberlo.

XU3

Realmente tuve muchos problemas para ejecutar StarCraft 64 en el XU3. El juego iba muy lento al principio, y al cambiar de glide64 a rice o gln64 aparecieron cosas extrañas. Rice y gln64 eran realmente rápidos en el menú de XU3, y todo iba a

máxima velocidad. Pero tanto rice como gln64 presentaban problemas gráficos importantes, lo cual hacía que el juego no se pudiera jugar. Tras algunas investigaciones sobre las ralentizaciones en glide64, descubrí que reduciendo la resolución aumentaba la velocidad. El juego se muestra en 1080p sin importar la resolución seleccionada, pero la resolución a la que se representan los personajes y los objetos se puede cambiar en XU3. Descubrí que con una resolución de 800×600 o menor se lograba un mejor rendimiento.

Super Mario 64

Super Mario 64 fue el título de lanzamiento para el N64, ¡y vaya que sí fue un lanzamiento! Este juego impulsó al N64 a la cima de su clase mostrando de qué era capaz la consola y, una vez más, convirtió a Mario en la estrella de la franquicia de Nintendo.



Figura 8 - Super Mario 64

U3

En el U3, Mario 64 tiene algunos fallos técnicos con sombras, texturas e iluminación, pero aparte de eso, el juego se ejecuta a máxima velocidad.

C1 - plugin rice

Mario 64 parece ejecutarse un poco por debajo de la velocidad máxima en el C1, pero todavía se puede jugar con el complemento de gráficos rice. La velocidad es un poco mejor con glide64mk2 que con el plugin rice, pero ocasionalmente cae por debajo de la velocidad máxima. También tiene los mismos problemas que el plugin glide64mk2 en el U3 con el renderizado de texturas del suelo y las sombras.

XU3

El juego funciona bien en el XU3, sin tipo de problemas ni errores.

Super Smash Bros

Este juego introdujo un nuevo género de juegos de lucha. Fue un gran éxito en el N64 y dio lugar a muchas secuelas. Puedes elegir entre los famosos personajes de Nintendo como Mario, Yoshi, Princess Peach y muchos más, y luchar contra otros personajes.



Figura 9 - Super Smash Bros

U3

La experiencia de juego de Super Smash Bros en el U3 con el plugin mupen64plus y glide64mk2 es muy agradable. Incluso el menú funciona a una velocidad decente. Existen algunos fallos con las sombras y los textos, pero nada serio, tan sólo los problemas con los textos son realmente apreciables.

C1 - plugin rice

presenta retardos. No obstante, Super Smash Bros funciona bastante mejor con el plugin glide64mk2, y puedes jugarlo prácticamente a toda velocidad, aunque tiene los mismos fallos que la versión del U3.

XU3

Aunque en el menú, aparecen algunos retrasos y ralentizaciones, el juego funciona perfectamente. Es realmente divertido de jugar.

The Legend of Zelda: Majora's Mask

No conozco muy bien los juegos de Legend of Zelda de N64, pero sí sé que este juego te lleva unas 72

horas recorrer todo el mundo, y dispones de diferentes máscaras que te ayudan en tu causa. Puedes usar la "Ocarina of Time" para retroceder en el tiempo y empezar las 72 horas una y otra vez hasta que finalices el juego.



Figura 10 - The Legend of Zelda: Majora's Mask

U3

Aunque la velocidad del juego es bastante buena, el plugin glide64mk2 una vez más tiene problemas por ser demasiado oscuro. Como no es posible activar el efecto borroso, el juego se mantiene a toda velocidad todo el tiempo. Sin embargo, debido a que es demasiado oscuro, a veces resulta difícil encontrar la forma de hacer las cosas, aunque no es tan oscuro como cuando se juega en el XU3, donde no ves prácticamente nada. Considero que es completamente jugable.

C1 - plugin rice

El juego funciona sorprendentemente bien en el ODROID-C1 con el plugin rice. No hubo problemas gráficos, aunque la introducción y algunas escenas presentan ligeros desfases. En general, el juego es muy jugable en C1 con el complemento rice.

C1 - glide64mk2

El juego se ejecuta a velocidad casi completa, aunque presenta el mismo problema de oscuridad que el U3. Rice es probablemente el mejor plugin para este juego a la hora de jugar en el ODROID-C1.

XU3

La experiencia del juego en general es bastante buena. Cuando hay escenas con efectos borrosos, el juego se ralentiza y se vuelve más lento. Sin embargo, dado que solo ocurre en determinadas escenas, el juego

se ejecuta bastante bien. Sin embargo, existe otro problema relacionado con el plugin glide, los gráficos son demasiado oscuros, lo que dificulta elegir el camino a seguir. Se volvía tan oscuro que tuve que cambiar al plugin gln64, que presenta pequeños fallos sobre el terreno, aunque por otro lado funcionaba perfectamente a toda velocidad. Resulta tan oscuro que no logras ver a dónde tienes que ir, de modo que usar gln64 como plugin para este juego es la mejor opción.

The Legend of Zelda: Ocarina of Time

Se trata del predecesor de Majora's Mask. De hecho, me costó bastante disfrutar del juego, pero sé que mejorará con el tiempo. Debe haber una razón por la que tanta gente lo tienen en su lista de los 10 mejores, así que decidí probarlo.



Figura 11 - The Legend of Zelda: Ocarina of Time

U3

En general, el juego funciona bien y se ejecuta a toda velocidad, con algunos problemas menores con las sombras y las texturas del suelo. En algunos lugares, es demasiado oscuro, pero aún así es completamente jugable.

C1 - plugin rice

Tal y como ocurre en el otro juego Legend of Zelda, éste funciona muy bien en el ODROID-C1 usando el plugin rice. Con el plugin glide64mk2, el juego no alcanza su velocidad máxima y muestra los típicos problemas con las sombras y las texturas del suelo.

XU3

La experiencia en el XU3 es excelente. Hasta ahora, no he visto fallos ni ralentizaciones, aunque no he llegado demasiado lejos en el juego. Es una experiencia realmente agradable.

Texturas de Alta Resolución

Después de probar diferentes juegos, estuve revisando qué más se podía hacer con los emuladores, y descubrí que existen algunos paquetes de texturas de alta resolución que mejoran considerablemente los gráficos. Probé algunos de ellos para ver si funcionarían en los ODROID. El emulador independiente Mupen64plus ofrece la posibilidad de utilizar texturas de alta resolución para juegos N64 que pueden mejorar la experiencia de juego dando un nuevo aspecto a los juegos, aunque esta opción no está disponible para otros emuladores.



Figura 12 - Super Mario 64 sin texturas de alta resolución



Figura 13 - Super Mario 64 con texturas de alta resolución

Para usar las texturas de alta resolución, descárgalas desde <http://bit.ly/1Jvpahr> y cópialas en el directorio ~/.local/share/mupen64plus/hires_texture/. Algunas de las texturas reescriben por completo los gráficos del juego. Asegúrate de colocar las texturas en una carpeta con el "nombre corto" del juego en mayúsculas. Por ejemplo, Mario 64 es "SUPER MARIO 64", y Mario Kart 64 es "MARIOKART64".



Figura 14 - Completa transformación de las texturas de Mario 64

Conclusiones

La emulación de Nintendo 64 por lo general funciona bastante bien en los dispositivos ODROID, especialmente en el U3 y XU3. El C1 tiene muchos problemas que le impiden ofrecer la misma experiencia de juego que el resto de dispositivos ODROID. El plugin rice, que funciona sin tener que cambiar la configuración de profundidad del color en la imagen, tiene problemas importantes con muchos juegos, pero realiza un buen trabajo en otros. El plugin glide64mk2 solo funciona con 16 bits, y aunque la mayoría de los juegos se ejecutan muy bien, los que funcionan mejor con el plugin rice requieren reiniciar para poder usarlo, ya que rice no funciona con 16 bits. Esto básicamente era un incordio, ya que siempre tenía que reiniciar el ODROID por completo para cambiar entre los diferentes plugins gráficos en el C1. El U3 y XU3 pueden hacer esto sin tener que reiniciar todo el sistema, lo cual hace que sea mucho más fácil cambiar entre los diferentes plugins. Además, al utilizar una profundidad de color de 16 bits se evita que diferentes aplicaciones, como XBMC, se ejecuten correctamente, lo cual hace que elijas un

interfaz de emulador que realmente soporte el modo 16 bits, o te ves obligado a iniciar los juegos de N64 a través de una ventana de Terminal. Todo esto me hace pensar que el C1 no es realmente adecuado para N64, al menos en Linux. Creo que la mejor forma de ejecutar juegos N64 en el C1 es probablemente a través de aplicaciones de Android o una versión sumamente modificada que utilice drivers fbdev y algunos scripts que puedan cambiar la profundidad del color y las aplicaciones a ejecutar. Esa configuración sería un inconveniente y por supuesto no apta para principiantes. El U3 y el XU3 son muy buenos a la hora de emular juegos de la N64. Poder cambiar fácilmente entre núcleos gráficos es una gran ventaja sobre el C1. Los juegos N64 parecen necesitar

en ocasiones algunos ajustes, y si nos fijamos en las opciones de configuración de glide64mk2 o rice en el emulador independiente mupen64plus, hay muchas opciones para elegir. El XU3 es la única placa que puede usar el núcleo libretro de mupen64plus con Retroarch en este momento. Integra los mandos muy bien pudiendo adaptar fácilmente el diseño de tu gamepad a tus propias necesidades y es compatible con diferentes mandos. Además, el XU3 tiene una potencia de CPU adicional, que a menudo marca la diferencia entre la velocidad máxima o la "casi" velocidad máxima. El U3 hace un muy buen trabajo emulando los juegos de N64, y ser capaz de usar texturas de alta resolución en mupen64plus es algo realmente genial.
