oDrive Car • Emulación Amiga • Xbox en ODROID-H2 • Gamestation





WEBTHINGS:
ARMBIAN
ODROID-XU4
PARA INTERNET
DE LAS COSAS

ODROID-GO • RETRO-GO: UN ELEGANTE EMULADOR ODROID-GO

ROTACION PANTALLA SDL2: HIPERCARGA TU ODROID-GO ADVANCE CON GRAFICOS



SDL2 con Rotación de Pantalla: Hipercarga los Gráficos de tu ODROID-GO Advance

② April 1, 2020

He exportado con éxito SDL2 con rotación de pantalla interna, de modo que ya no necesitas una versión modificada de PPSSPP o Emulationstation.



Sistema de Emulación Amiga: Convierte tu ODROID-XU4 en una Increible Ordenador de los 80

② April 3, 2020

Este tutorial te mostrará cómo configurar un emulador de Amiga en tu ODROID-XU4 creando tu propia copia de Uae4Arm y configurándola con ROMs kickstart y archivos

Workbench de Amiga Forever o de donde sea que hayas obtenido tu copia. Se supone que tienes un ODROID-XU4 funcionando con Ubuntu 18.XX.



Car PC BMW oDrive

① April 1, 2020

Llevo casi un año trabajando en este proyecto y ahora tengo la suficiente confianza como para presentarlo al público: la todopoderosa solución suplementaria iDrive para todas aquellas personas que no tienen interés en instalar una solución de 1500 €.



Ejecutando Linux bajo Android

② April 1, 2020

El ODROID-N2 parece un sistema de TV Android perfecto: todo funciona sin problemas y convierte un televisor normal en un televisor inteligente muy valioso. Todo funciona perfecto si eso es lo único que quieres hacer con tu ODROID-N2, pero para mí, no es

suficiente. Quería usar el ODROID-N2 para reemplazar



¿Vamos a jugar a un juego? - Nintendo Drags Pokémon en Pleno Siglo 21...Pateando y Gritando

① April 1, 2020

Imagina poder compartir, intercambiar y mover tus Pokémon entre tus juegos en diferentes dispositivos a través de un servidor centralizado conectado a Internet.

Suena a algo realmente excéntrico, impresionante, futurista, ¿eh? Ahora puedes con pokémon home.



Montando una Xbox Usando un ODROID-H2

② April 1, 2020

El OG Xbox conmociono todo nuestro mundo: aún seguimos llamando al reproductor multimedia 'la xbox' a día de hoy. Después de ser modificada, se convirtió en una potencia mediática muy adelantada a su tiempo. ¡Ahora es el momento de emularla!



Retro-Go, Otro Lanzador Emulador para ODROID-GO

① April 1, 2020

Retro-Go es un lanzador con emuladores NES/GB/GBC/SMS/GameGear/Colecovision. El código del emulador está basado en Go-Play "Triforce" y el diseño del lanzador está bastante inspirado en el de pelle7; Sin embargo, no es una bifurcación.



Retro Roller: Una Experiencia de Juego Optimizada para tu ODROID-GO Advance

② April 1, 2020

Retroroller es una imagen precompilada para ODROID-GO Advance que proporciona RetroArch 32 bits y 64 bits en la única imagen estándar de CrashOverride. Entre otros

ajustes, tiene integrado un kernel personalizado que soporta el modo reposo. La idea con esta imagen es poder actualizarla a través de paquetes en lugar



WebThings en Armbian: Usando el ODROID-XU4 para el Internet de las Cosas

② April 1, 2020

Mozilla WebThings es una plataforma para controlar dispositivos domésticos.



Modificando tu ODROID-GO Advance para la Carga Inalámbrica: Un simple Proyecto de Bricolaje

② April 1, 2020

He añadido la posibilidad de cargar de forma inalámbrica mi ODROID-GO Advance



Juegos Linux: GameStation Turbo Advance

② April 1, 2020

Aquí tienen el denominado ODROID GameStation Turbo Advance. Un sistema operativo de juegos repleto de funciones para ODROID-GO Advance que utiliza X11 como backend de gráficos y el modo atracción como interfaz para emuladores, juegos

y herramientas.

SDL2 con Rotación de Pantalla: Hipercarga los Gráficos de tu ODROID-GO Advance

② April 1, 2020 By @AreaScout | Juegos, ODROID-GO Advance



He exportado con éxito SDL2 con rotación de pantalla interna, de modo que ya no necesitas una versión modificada de PPSSPP o Emulationstation. SDL2 permite mostrar una resolución de pantalla de 480x320. Hay un montón juegos gratuitos en GitHub, y probablemente algunos emuladores muy buenos que usen SDL2. También existen buenas GUI de código abierto, como nanoguisdl, que también admiten SDL2.



Figura 1 - ODROID-Go Advance ejecutando PPSSPP usando SDL2 con rotación de pantalla

Preparación

En primer lugar, necesitarás el controlador de GPU Mali como un paquete deb, para que las librerías de Mesa no lo sobrescriban:

\$ wget

https://oph.mdrjr.net/meveric/pool/go2/libm/libmal

```
i-rk/libmali-rk-bifrost-g31-rxp0-gbm_1.7-
1+deb10_arm64.deb

$ wget
https://oph.mdrjr.net/meveric/pool/go2/libm/libmal
i-rk/libmali-rk-dev_1.7-1+deb10_arm64.deb

$ sudo apt install ./libmali-rk-dev_1.7-
1+deb10_arm64.deb ./libmali-rk-bifrost-g31-rxp0-
gbm_1.7-1+deb10_arm64.deb
```

Instalación

El primer paquete son las librerías de tiempo de ejecución, y el segundo es el paquete de desarrollo que contiene las cabeceras, en caso de que estés a punto de compilar las aplicaciones. La librería SDL2 se instalará en /usr/lib/aarch64-linux-gnu y sobrescribirá cualquier librería SDL2 instalada previamente.

```
$ wget https://www.areascout.at/libsdl2-2.0-
0_2.0.10+dfsg1-1ubuntu1_arm64.deb
$ wget https://www.areascout.at/libsdl2-
dev_2.0.10+dfsg1-1ubuntu1_arm64.deb

$ sudo apt install ./libsdl2-2.0-0_2.0.10+dfsg1-
1ubuntu1_arm64.deb
$ sudo apt install ./libsdl2-dev_2.0.10+dfsg1-
1ubuntu1_arm64.deb
```

Para comentarios, preguntas y sugerencias, visita el artículo original

en https://forum.odroid.com/viewtopic.php? f=194&t=38045.

Sistema de Emulación Amiga: Convierte tu ODROID-XU4 en una Increible Ordenador de los 80

② April 3, 2020 By Brian A. Ree Linux



Cosas que necesitarás

- Un ODROID-XU4, ya sea con ventilador o disipador térmico
- Una carcasa ODROID-XU4
- Ya sea un módulo eMMC o una tarjeta micro SD
- Un adpatador para tarjeta microSD o para eMMC
- Módulo wifi; Los módulos 0, 5A y 4 pueden valer
- Ratón y teclado USB
- Pantalla HDMI
- Conexión a Internet
- Edición Amiga Forever Plus
- Un ordenador con Windows

Herramientas Necesarias

Juego de pequeños destornilladores

Introducción y objetivos del tutorial

Este tutorial te mostrará cómo configurar un emulador de Amiga en tu ODROID-XU4 creando tu propia copia de Uae4Arm y configurándola con ROMs kickstart y archivos Workbench de Amiga Forever o de donde sea que hayas obtenido tu copia. Se supone que tienes un ODROID-XU4 funcionando con Ubuntu 18.XX.

Para obtener ayuda sobre cómo escribir una imagen de sistema operativo en una tarjeta SD o módulo eMMC, visita el siguiente enlace. Asegúrate de seleccionar la pestaña ODROID-XU4 a la hora de obtener una imagen de Ubuntu.

https://wiki.odroid.com/odroidxu4/getting_started/os_installation_guide?

redirect=1#downloads A continuación, debes continuar y ensamblar la carcasa del ODROID-XU4 siguiendo las instrucciones que se encuentran en http://middlemind.net/tutorials/odroid_go/mr3_build.html#part3. Si has leído cada una de las

toda la configuración empiece a funcionar con la emulación de Amiga en tu dispositivo ODROID-XU4. El coste genérico de este proyecto es de alrededor de unos 140\$ dependiendo de las opciones de configuración que realices. Obviamente, si optar por usar tarjetas micro SD y no necesitas un adaptador USB WIFI, el coste será menor Este tutorial funciona con archivos Amiga que vienen con el paquete plus de Amiga Forever que cuesta alrededor de unos 30\$. Recomiendo comprar este paquete con todas las ROM kickstart y los archivos Workbench que necesitarás para emular los ordenadores Amiga sobre casi cualquier software de emulación, WinUAE, FS-UAE, Uae4Arm, Uae4All, etc. Necesitarás tener acceso a uno ordenador con Windows para instalar Amiga Forever y obtener una copia de los archivos necesarios. Tras instalar el software, puede ubicar los archivos del sistema operativo Amiga necesitamos en el siguiente directorio. También puedes usar tus propios archivos del sistema operativo Amiga; simplemente sustituye los archivos apropiados, donde sea necesario.

C:UsersPublicDocumentsAmiga FilesShared

Haz una copia de la siguiente carpeta en una memoria USB o en una de tus tarjetas SD utilizando el adaptador micro SD a USB

```
□df
hdf

om - (Rename to kickstarts)
dirSystem
```

Para aquellos que usan sus propios archivos del sistema operativo Amiga, necesitarán una ROM Kickstart A1200 3.1 y un archivo Workbench 3.1 .adf o .hdf. También tienes que crear las siguientes carpetas que se completarán con archivos.

```
dirGames
dirExtras
```

Ahora que tiene todos los archivos que necesitas, comencemos a configurar nuestro ODROID-XU4 para la emulación de Amiga y los juegos de Amiga.

Compilando Uae4Arm

secciones anteriores, debería ser suficiente para que Primero, hazte con una copia de la última versión de toda la configuración empiece a funcionar con la Uae4Arm de la siguiente URL: emulación de Amiga en tu dispositivo ODROID-XU4. El https://github.com/Chips-fr/uae4arm-rpi. Haz clic en coste genérico de este proyecto es de alrededor de la opción "Download zip" tal y como se muestra a unos 140\$ dependiendo de las opciones de continuación

Figura 1 - Descargando el archivo zip para uae4arm

Crea dos carpetas nuevas en tu directorio de inicio, /home/odroid. Para tu información, la contraseña del superusuario es odroid por defecto en las imágenes del sistema operativo ODROID-XU4 que estamos usando. Queremos crear una carpeta install_zips y una carpeta uae4arm. Cuando el código fuente uae4arm termine de descargarse, copia el archivo uae4arm-rpi-master.zip resultante en la carpeta install_zips. Expande y mueve la carpeta uae4arm-rpi-master a la carpeta uae4arm recién creada.

Necesitaremos instalar algunos paquetes antes de poder compilar el código uae4arm. Abre un nuevo terminal dirigiéndote a Menú -> Herramientas del sistema -> Terminal MATE. Ejecuta los siguientes comandos en la ventana del terminal.

```
$ sudo apt-get install libsdl1.2-dev -y
$ sudo apt-get install libguichan-dev -y
$ sudo apt-get install libsdl-ttf2.0-dev -y
$ sudo apt-get install libsdl-gfx1.2-dev -y
$ sudo apt-get install libxml2-dev -y
$ sudo apt-get install libflac-dev -y
$ sudo apt-get install libmpg123-dev -y
$ sudo apt-get install libmpeg2-4-dev -y
$ sudo apt-get install autoconf -y
$ sudo apt-get install libgles2-mesa-dev -y
$ sudo apt-get install libgles1 -y
$ sudo apt-get install libgles2 -y
$ sudo apt-get install libgles2 -y
```

O si quieres hacerlo todo con un gran comando...

```
$ sudo apt-get install libsdl1.2-dev -y;sudo apt-get install libguichan-dev -y;sudo apt-get install libsdl-ttf2.0-dev -y; sudo apt-get install libsdl-gfx1.2-dev -y;sudo apt-get install libxml2-dev -y;sudo apt-get install libflac-dev -y; sudo apt-get install libmpg123-dev -y;sudo apt-get install libmpg2-4-dev -y;sudo apt-get install autoconf -y; sudo apt install libgles2-mesa-dev -y;sudo apt install libgles1 -y;sudo apt install libgles2 - y;sudo apt-get install libgles4 -y;
```

Dirigete al directorio uae4arm-rpi-master usando el siguiente comando.

\$ cd /home/odroid/uae4arm/uae4arm-rpi-master

A continuación, vamos a compilar uae4arm ejecutando el siguiente comando en el terminal con el directorio actual fijando el directorio uae4arm-rpimaster.

\$ make PLATFORM=gles

Deja que la compilación siga su curso y cuando se complete, deberías tener un nuevo archivo en el directorio uae4arm-rpi-master llamado uae4arm. Copia las carpetas adf, hdf, dir v kickstarts que hemos mencionado anteriormente desde tu unidad USB o lo que sea que estés utilizando, a la carpeta uae4arm. Debes tener la siguiente estructura de carpetas. Si no estás utilizando Amiga Forever, crea la siguiente estructura de directorios y coloca tus archivos .adf, juegos, Workbench, etc. en la carpeta adf. Del mismo modo, coloca cualquier archivo .hdf, imágenes de unidad, etc., en la carpeta hdf. Tus archivos ROM kickstart van a la carpeta kickstarts. La carpeta Games se usará para los juegos del disco duro y juegos preparados con WHDLoad. Se usarán Extras para los programas y archivos de Amiga que necesitaremos transferir a la unidad de arrangue .hdf. El sistema se usa específicamente para archivos de arranque .hdf.

- uae4arm - uae4arm-rpi-master - adf - hdf - kickstarts- dir - System - Extras - Games

Deberías tener la siguiente configuración de carpetas en tu directorio de inicio de odroid y las siguientes carpetas en el directorio uae4arm.

Figura 2: Deberías tener dos carpetas "install_zips" y "uae4arm"

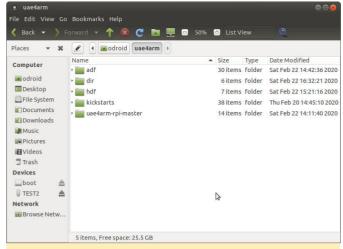


Figura 3 - El contenido de la carpeta "uae4arm"

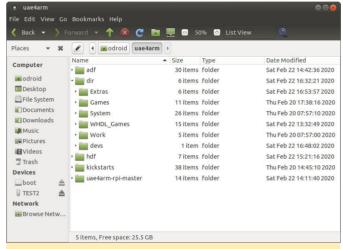


Figura 4 - El contenido de la carpeta "dir"

Necesitaremos algunos componentes de software para algunos de los pasos de configuración del sistema operativo Amiga que te mostraré. Puedes descargar WHDLoad en esta URL: : http://whdload.de/. Descarga el archivo WHDLoad_usr.lha de la página principal. Cuando finalice la descarga, mueve el archivo de la carpeta Downloads a la carpeta uae4arm/dir/Extras.

El sitio WHDLoad se muestra a continuación.

Figura 5 - El sitio web WHDLoad

A continuación, necesitaremos una herramienta de descompresión de Amiga llamada lha. Puede descargar una copia directamente en la siguiente URL: lha.run. Cuando finalice la descarga del archivo, mueve el archivo resultante a la carpeta uae4arm/dir/Extras.

Lo último que necesitaremos son algunos juegos para hacer pruebas. La mayor parte de la biblioteca de software de Amiga está abandonada, ya que es una plataforma muerta. Para este tutorial, necesitará encontrar algunos juegos .adf para Amiga 500, 1000 o 1200. También necesitarás descargar algunos juegos preparados para el software WHDLoad desde esta URL WHDLoad Games. Los juegos deben proceder del directorio WHDLoad.

El sitio de descarga del juego se muestra a continuación.

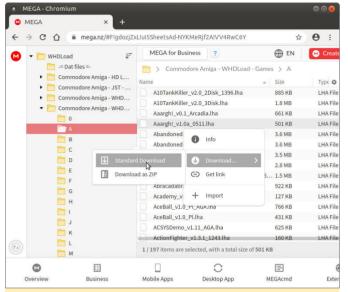


Figura 6 - Descargar el sitio web

Lo que WHDLoad hace es lanzar juegos de Amiga desde el disco duro y no desde una imagen de disco para que puedas tener una colección completa cargada en tu sistema operativo Amiga emulado. También carga la versión ROM kickstart requerida del juego. De esta manera, no tendrá que cargar un nuevo disco cada vez que quiera jugar a un juego.

Regresa a la ventana de terminal o abre una nueva. carpeta uae4arm-rpi-master, Navega la cd/home/odroid/uae4arm/uae4arm-rpi-master.

Luego, inicia uae4arm ejecutando ./uae4arm. Esto ejecutará el archivo binario que compilaste anteriormente. Si recibes un error de que el archivo no se puede ejecutar, ejecuta los siguientes comandos en el terminal.

```
sudo chmod 755 ./uae4arm
sudo chmod +x ./uae4arm
```

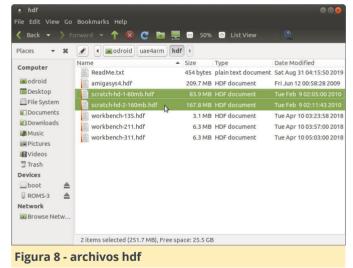
Ejecuta el comando ./uae4arm nuevamente, deberías ver la ventana emergente de la ventana del iniciador Uae4Arm tal y como se muestra a continuación.



Figura 7 - Selección de ROM

Uae4Arm configurado

En esta sección, trabajaremos con la configuración de Uae4Arm para que tengamos un emulador estable de Amiga con el que podamos lanzar un montón de juegos. Antes de entrar en detalle, necesitaremos dos archivos más. La función de creación de archivos de disco duro de Uae4Arm parece no funcionar, por lo que tendremos que solucionarlo. Dirígete a esta URL: https://scruss.com/blog/2010/02/07/amiga-blankhdf-images/ y descarga los archivos del disco duro de 80 MB y 160 MB.



Cuando los archivos terminen de descargarse, muévelos de la carpeta Downloads a la carpeta install_zips. Expande los archivos. Cuando hayas terminado, mueva los archivos resultantes a la carpeta hdf en el directorio uae4arm.

Ahora ajustaremos la configuración adecuada para crear un buen sistema Amiga emulado A1200. Dirígete a la sección de configuración de Inicio rápido y selecciona A1200 de la lista. A continuación, dirígete

a la sección CPU y FPU. Asegúrate de que JIT esté seleccionado en las secciones CPU y FPU. Selecciona la opción Fastest para ajustar la velocidad de la CPU y configurar la FPU en 68882

A continuación, dirígete a la sección ROM y carga el archivo amiga-os-3101-a1200.rom desde la carpeta kickstarts. También puedes usar un archivo equivalente si tiene uno, el archivo que estamos usando proviene de: https://www.amigaforever.com/; Este archivo, y muchos más, viene en forma de servicio adicional. La siguiente captura de pantalla muestra la pantalla ROM con el archivo apropiado seleccionado.

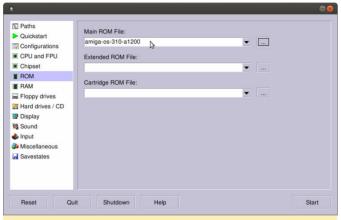


Figura 9 - Selección de Rom

A continuación, configuraremos la sección RAM. Dirígete a la sección RAM y desliza el Chip RAM a 8MB como se muestra a continuación.

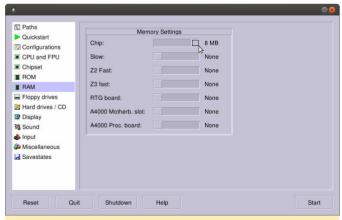


Figura 10 - Configuración de RAM

Ahora cargaremos algunos discos duros. La primera unidad será nuestra unidad de arranque. Navega hasta el directorio dir/System y selecciona el workbench-311.hdf. Una vez más, este archivo proviene del paquete Amiga Forever, pero puedes usar una versión .adf de arranque si tiene uno u otro

disco duro de arranque que tenga Workbench 3.1. La siguiente captura de pantalla muestra la configuración de la unidad.

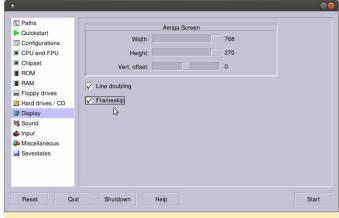


Figura 11 - Configuración del disco duro

El próximo disco duro que queremos cargar es uno de los discos en blanco que descargamos de Internet. Yo iría con el de 160 MB. Cárgalo en la lista de unidades y asegúrate de anular la selección de la opción de arranque. Lo que te voy a mostrar aquí es cómo copiar una unidad de arranque más pequeña en una unidad en blanco más grande y luego usarla como unidad de arranque. Uae4Arm hace un muy buen trabajo emulando máquinas A1200 y anteriores, pero no puedo hacer que las entradas A4000 funcionen. El software está en la versión 0.5 y puede necesitar algo más de recorrido. Otra peculiaridad es que no puede escribir nuevos archivos o carpetas en unidades basadas en directorios; Aparece un error que bloquea la aplicación. Una solución alternativa es utilizar una unidad basada en hdf. Por lo tanto, nuestro objetivo es crear una nueva unidad de arranque más grande para que podamos mover algunos juegos e instalar nuevo software.

Ajustaremos el tamaño de la pantalla y configuremos el emulador para usar el tamaño más grande posible en la sección de la Configuración de la pantalla. La siguiente captura de pantalla muestra el tamaño de pantalla más grande posible con una A1200 de 8 MB y sin tarjeta gráfica. A continuación, configuremos la entrada. Dirigete a la sección Configuración de entrada. Ajsuta la entrada como quieras, yo utilicé la siguiente configuración para empezar, ya que todas las asignaciones van directamente a las teclas con el mismo nombre.

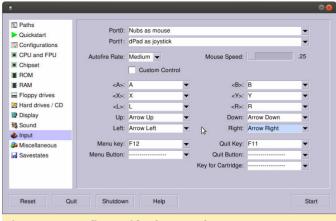


Figura 12 - Configuración de entrada

Una última cosa que hay que hacer antes de encender nuestro Amiga A1200 es volver a la sección de Configuraciones y hacer clic en el botón Save. Puedes cambiar el nombre de la configuración, pero yo uso el que aparece por defecto.

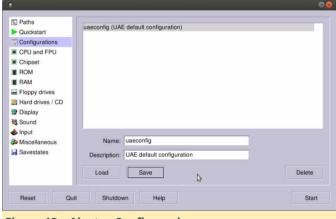
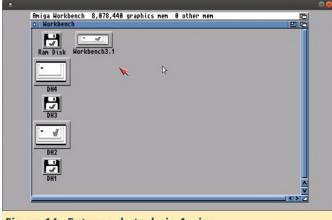


Figura 13 - Ajustar Configuraciones

Ahora que tiene todo configurado y listo para funcionar, haz clic en el botón Inicio y debería ver un entorno de trabajo de Amiga A1200 limpio y cargado tal y como se muestra a continuación.



Ejecutando juegos de Amiga * .adf

Figura 14 - Entorno de trabajo Amiga

Ahora el A1200 puede ejecutar, en su mayor parte, juegos A500, A600 y A1000. Sin embargo, no todos funcionarán. Algunos pueden requerir un conjunto de chips específico, pero la mayoría debería funcionar "directamente". Para cargar y ejecutar un juego en formato .adf, presione la tecla F12 o la tecla que haya asignado en la sección de Configuración de entrada.

Dirígete a la sección de configuración de unidades de disquete y selecciona un juego .adf al que deseas jugar. Incluso puede verificar la opción Load config con el mismo nombre que el disco para crear configuraciones específicas para el disco/juego dado. Por ahora, deja esa opción sin marcar. Selecciona tu disco de juego tal y como se muestra a continuación.



Figura 15 - Configuración del disquete

Haz clic en el botón Resume si ya iniciaste tu emulación A1200 o haz clic en el botón Start. Deberías ver un icono de disquete en el escritorio de Workbench. Haz doble clic para abrirlo. Mantenlo presionado con el botón derecho del ratón mientras el ratón se encuentra en el encabezado del entorno de trabajo, desplaza el ratón sobre Window, luego Show y selecciona All Files

Ahora deberías poder ver todos los archivos en el disquete del juego que cargaste. Encuentra el ejecutable del juego y haz doble clic en él. Ahora deberías poder jugar al juego seleccionado en tu emulador Amiga A1200 con ODROID-XU4. Si el juego falla, es posible que tengas que buscar requisitos de hardware específicos para ese juego y ajustar la configuración de tu emulador para satisfacer esos requisitos.

La siguiente captura de pantalla muestra la pantalla de un juego que he cargado desde una imagen de disquete, archivo .adf.



Figura 16 - Alien Syndrome

Usando WHDLoad para ejecutar juegos

En esta sección aprenderemos a ejecutar juegos de Amiga desde el disco duro usando WHDLoad. Este metodo tiene la ventaja adicional de utilizar un sistema flexible para iniciar el juego con la ROM kickstart adecuada. Necesitarás tener una unidad de arranque de workbench y un disco duro vacío, .hdf, que descargamos anteriormente. No estoy seguro de si este proceso funcionará con un disquete de arranque del entorno de trabajo, quizás lo haga. Simplemente no lo he probado.

Así que utilicé el Workbench-311.hdf de mi copia de Amiga Forever, pero solo se trata de una unidad de 9 MB. Te quedarás sin espacio muy rápido. Para crear una nueva unidad de arranque con más espacio libre, he cargado la unidad .hdf en blanco de 160 MB, vamos a clonar la unidad de arranque en ella.

Mi configuración inicial de disco duro se muestra a continuación. Lo más importante a recordar es que solo utilizamos discos duros basados en "directorios" como fuente de archivos. No crearemos nuevos archivos o carpetas en ellos. De modo que, la carpeta Extras es donde almacenaremos el nuevo software Amiga, DH4. La primera unidad es una unidad de arranque más pequeña, pero preconfigurada, DH2. Por último, el nuevo disco duro en blanco de 160 MB será nuestro nuevo disco de arranque y a la larga contendrá todo el nuevo software de Amiga que necesitamos, DH3.

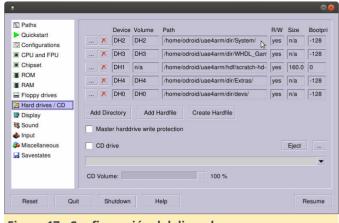


Figura 17 - Configuración del disco duro

Busca el programa AmigaSHELL, debe estar en la carpeta del sistema de la unidad Workbench 3.1. Haz doble clic en el icono del Intérprete de comandos. Y ejecuta el siguiente comando.

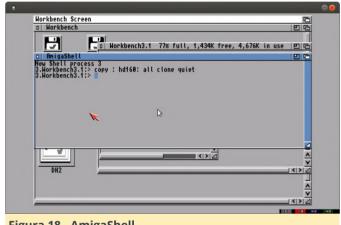


Figura 18 - AmigaShell

El comando AmigaSHELL es copy: hd160: all clone quiet, donde el segundo nombre de la unidad, hd160 es el nombre asignado a tu unidad de 160 MB. Este comando clonará la unidad y creará una nueva unidad de arranque. Cuando finalice, puede cerrar tu sesión de emulación y luego quitar la unidad más pequeña y asignar la unidad más grande de 160 MB como unidad de arranque.

Además, vas a asignar una nueva unidad basada en el directorio Extras. Este directorio debe contener los archivos lha.run y WHDLoad.lha. También debe asignar el directorio Games que tiene los juegos preparados WHDLoad. Por lo tanto, tendrás un total de tres unidades que son las siguientes.

1. Unidad de arranque más grande, 160 MB, basada en .hdf 2. Unidad de extras, basada en directorio dirExtras 3. Unidad de juegos, basada en directorio dirGames

He tenido problemas para crear nuevos archivos y carpetas en una unidad basada en directorios, no estoy seguro de cuál fue la causa. Siempre copio los programas que necesito ejecutar en la unidad de arranque .hdf. De modo que, en este caso, arrastra y suelta los archivos lha.run, WHDLoad.lha y destino .lha del juego en la unidad de arranque.

Si no puedes ver los archivos en ninguna unidad o carpeta en la que que entras, simplemente haz clic con el botón derecho en el encabezado de Workbench 3.1 y, mientras mantienes presionado el botón, arrastra el cursor a Windows, luego Show y seleccionar All Files.

Una vez que los archivos se hayan copiado, ejecute el programa lha.run haciendo doble clic en él. Asegúrate de estar usando la copia que está en la unidad de arranque. Una vez que hayas completado la instalación, abre un nuevo AmigaSHELL y escriba lha y presiona Intro. Debería ver una lista de argumentos del comando.

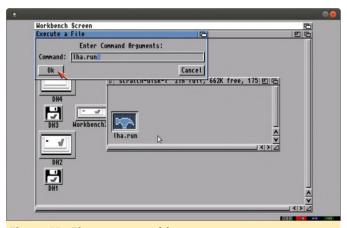


Figura 19 - Ejecutar un archivo

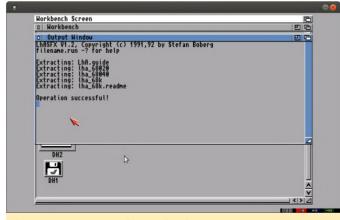


Figura 20 - Ventana de resultados

Deberías estar en la raíz de la unidad de arranque y ahí es donde se debería haber copiado WHDLoad.

Desde AmigaSHELL, ejecuta lha x WHDLoad.lha. Deja que se ejecute la expansión y cuando esté completa localiza la carpeta WHDLoad en la unidad de arranque. Ábrela y ejecuta el programa de instalación. Puedes elegir opciones para ejecutar la instalación a modo de prueba y no instalarla realmente. Está bien hacerlo primero para asegurarse de que todo funciona sin problemas. Completa una instalación real y estarás casi listo para ejecutar los juegos preparados de WHDLoad. No voy a hablar sobre cómo convertir juegos .adf a juegos WHDLoad en este tutorial. Los enlaces anteriores te proporcionarán una gran cantidad de juegos preparados para WHDLoad.

Sin embargo, necesitaremos descargar un componente más de software antes de poder lanzar un juego WHDLoad. Dirigete con el navegador a: http://aminet.net y localiza skick. Descarga una copia del programa como se muestra a continuación.

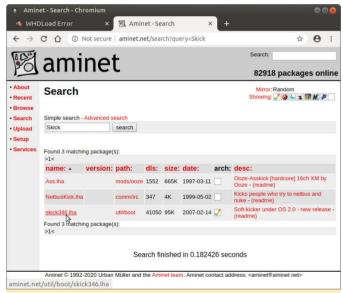


Figura 21 - El sitio web aminet.net

Apaga tu emulador Amiga A1200 y copia el nuevo archivo skick346.lha en el directorio Extras con los otros programas de Amiga. Inicia tu A1200 y copia el archivo skick346.lha en la unidad de arranque. Abre un shell, Workbench 3.1 -> System -> Shell. Crea una nueva carpeta en la unidad de arranque llamada kickstarts. Expande el archivo ejecutando este comando en el shell, lha x skick346.lha: kickstarts. Utiliza el nombre completo del archivo skick que descargaste si tiene una versión diferente.

Además, mientras tiene el shell abierto, expanda el juego preparado WHDLoad ejecutando lha x

name_of_game.lha. Ahora deberías tener nuevas carpetas en la unidad de arranque. Ejecuta el archivo de juego preparado WHDLoad y aparecerán algunos errores como los siguientes.

DOS-Error #205 (object not found) on reading "devs:kickstarts/kick40068.a4000

Haz una copia del a4000, o a500, o el archivo kickstart que se indica y muévelo a la unidad de arranque de la misma forma que moviste los otros programas de Amiga. Cópialo en la carpeta kickstarts en la unidad de arranque. Asegúrate de que se le haya cambiado el nombre para que coincida con el error. En mi caso, el error era la falta de un archivo kick34005.a500, así que utilicé una ROM kickstart 3.X del conjunto de ROM incluidas de Amiga Forever. Para obtener más información sobre qué ROM kickstart usar con respecto a Amiga Forever, dirigete a esta URL: http://www.whdload.de/docs/en/need.html.

Deberás crear un acceso directo para que exista la unidad devs: drive. Abre AmigaSHELL y ejecuta el siguiente comando, ASSIGN devs: hd0 :, donde la unidad llamada hd0 es el nombre de tu unidad de arranque o la ubicación de la carpeta kickstarts. Ahora debería tener los devs de ruta: kickstarts mapeados, y ya no deberías ver el error de kickstart que se monstraba antes.

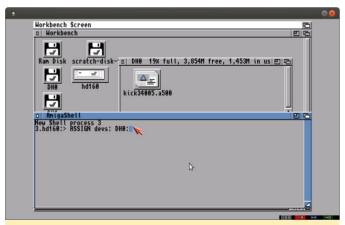


Figura 22 - Asinando devs

Una vez que superes este error, ejecuta el juego preparado WHDLoad nuevamente. Si recibes un error sobre un archivo RTB que falta, comprueba que descomprimiste el archivo skick346.lha en la ubicación adecuada. Si aún recibe un error de ROM kickstart, verifica que hayas copiado y renombrado correctamente el archivo ROM kickstart correcto.

Además, asegúrese de haber utilizado el comando ASSIGN correctamente para tener una unidad devs: kickstarts y una carpeta que contenga archivos RTB y ROM kickstart.

La carpeta Devs debería verse tal y como se muestra a continuación. Ahora contiene la ROM que copiaste y todos los archivos de configuración RTB del programa Soft kick Amiga.



Figura 23 - Devs

Yo utilicé un juego helicopter que fue preparado por WHDLoad y completé los pasos mencionados anteriormente. La siguiente captura de pantalla muestra el juego en funcionamiento. Cargado limpiamente desde el disco duro que nos da la capacidad de ejecutar un montón de juegos de forma rápida y fácil directamente desde el sistema Amiga emulado, sin intercambio de unidades de disquete.

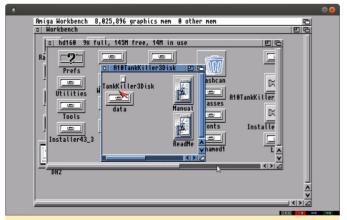


Figura 24 - Lanzando tankKiller3D



Figura 25 – Es Tiempo para divertirse

Para obtener más información, visita el artículo original

en http://middlemind.net/tutorials/odroid_go/mab_build.html.

Car PC BMW oDrive

② April 1, 2020 By @FntX Tutoriales



Llevo casi un año trabajando en este proyecto y ahora tengo la suficiente confianza como para presentarlo al público: la todopoderosa solución suplementaria iDrive para todas aquellas personas que no tienen interés en instalar una solución de 1500 €.

El proyecto consta de lo siguiente:

- Odroid N2 4GB con 32GB eMMC y Android como PC para el coche
- Pantalla Odroid VU7a+
- Carcasa de la pantalla ensamblada se instala reemplazando la pequeña guantera en el salpicadero (consulta: https://www.ebay.de/itm/Monitorhalter-f ... 1438.l2649)
- Controlador iDrive de 7 botones (renovación)
- Conjunto de consola central correspondiente para el controlador iDrive
- Placa de comandos y control personalizado con: Arduino Nano 33 IoT, Módulo Bluetooth RN42HID, interfaz CAN MCP2515 + TJA1050, sensor de voltaje INA219, 2 encabezados pwm para ventiladores (Ver:

https://github.com/Neuroquila-n8fall/od ... rpc-nano33)

- Punto de acceso 4G automotriz Teltonika RUT850
- Módulo de fuente de alimentación LTC3780
- Unos cables para conectarlo todo
- Unos cuantos retoques



Figura 1 - Pantalla VU7a+ instalada en el salpicadero que ejecuta la aplicación Torque

Ten en cuenta que el cabezal del pin en el lado del botón de encendido (lado izquierdo). Lo he soldado para tener una conexión adecuada para activar el botón de encendido de forma remota.



Figura 2 - VU7a+ montado dentro de la carcasa

En el corazón del proyecto es el Odroid N2. Se parece al ordenador que puedes encontrar en los sistemas iDrive. Sin embargo, no hace mucho ya que se enciende o se apaga y cualquier cambio en el estado de alimentación significaría presionar un botón de algún tipo. Funciona por sí solo, pero para obtener una solución integrada, el N2 debe encenderse o apagarse según el estado actual del coche y a "demanda de los conductores". Además, no tiene conectividad 4G integrada, lo que lo hace, a primera vista, inútil como sistema de información y entretenimiento con servicios de navegación y transmisión.

Puesto que las condiciones dentro de un automóvil son un tanto complicadas, tiene que ser alimentado por una fuente de alimentación que no solo proporcione suficiente potencia, sino que también tenga una cierta tolerancia a las altas temperaturas. En este aspecto, estoy muy sorprendido de que el N2 funcione bien en estas condiciones.

Arrancar y apagar con el coche

Podríamos activar la energía verificando los componentes que se conectarán si se abre el coche o se activa el encendido. Esto significaría que tan pronto como arranquemos el motor, la PC se iniciará. Esto funcionaría, pero no es lo mismo que un sistema iDrive original. Arranca muy rápido y simplemente no podemos alcanzar estas velocidades si queremos mantener el sistema Android como está.

Así que decidí piratear el bus CAN del automóvil y verificar si puedo recibir algún mensaje que indique cuál es el estado actual del automóvil y qué esperar de ello. De esta manera podríamos iniciar el N2 tan pronto como presionemos el botón "abrir" o "cerrar" en el mando. Con el tiempo que te lleva llegar hasta el automóvil, el N2 tiene suficiente para arrancar y tan

pronto como nos metamos en el coche, estará listo para hacer su trabajo.

También he experimentado con la posibilidad de tener la PC siempre encendido y ponerlo en suspensión o apagarlo si la batería baja demasiado. En pocas palabras, no llego a funcionar al final porque cambié a un MCU Due y no logré hacerlo funcionar con la correspondiente red divisoria de voltaje para monitorizar la batería. El código aún existe, así que ahora que tengo todos los servicios correctos en su lugar, podría reactivar todo el mecanismo.

Brillo de la pantalla

Un problema cuando conducimos un choche es que las condiciones de luz cambian continuamente. Esto hace que sea difícil leer una pantalla y esa es la razón por la que ahora empezamos a ver grandes pantallas en los coches actualmente Están reemplazando el salpicadero casi por completo. Estas pantallas deben ser resistentes al deslumbramiento, muy brillantes y robustas. Sin embargo, la cosa no termina ahí. La pantalla debe adaptarse a los niveles reales de luz exterior para que sea legible a la luz del sol y no te deslumbre por la noche.

Por lo general, dependeríamos de un fotodiodo para medir los niveles de luz y reaccionar ante ellos, pero en este caso, dado que ya estamos en el bus CAN del automóvil, podemos usar el sensor de nivel de luz del parabrisas. Se trata de un módulo instalado en un extremo del espejo retrovisor llamado "RLS" (Regen-Licht-Sensor = Rain-Light-Sensor). Informa de los niveles de luz actuales a intervalos y cuando cambian. Podemos usar esto para alimentar una señal PWM al regulador de luz de fondo VU7a (+) tal y como se describe en el Wiki (https://bit.ly/2UAzK1z).

Placa de comandos y control

El módulo "Comando y control" es la interfaz entre el automóvil y el N2. Se cierra la brecha entre el sistema Android y la red CAN-bus Cars y se encarga de todas las tareas que de otro modo significaría personalizar el sistema Android o escribir aplicaciones. La primera iteración de la placa de control fue una mezcla de un módulo Bluetooth Arduino Due suelto y un módulo de interfaz CAN MCP2515. ¡Que desastre!



Figura 3 - Esta es la primera interacción. Realmente es un completo desastre, pero de alguna manera necesitaba una plataforma de evaluación.

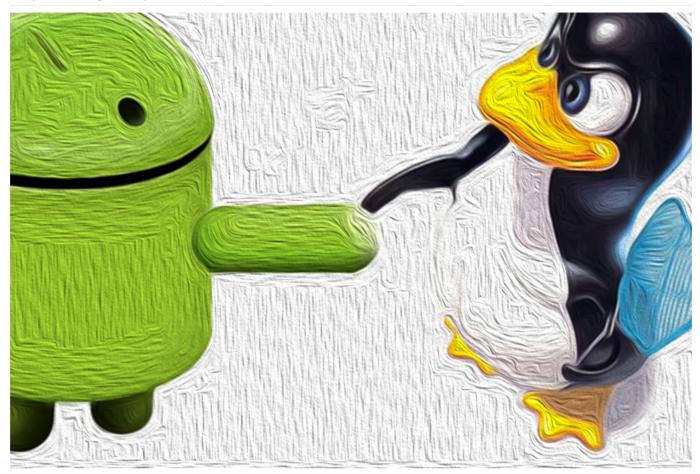


Figura 4 - Un año después, todo llega a su fin

El código fuente y los detalles se pueden encontrar en el repositorio de Github: https://bit.ly/2QMx7bR. Subiré más fotos de toda la configuración en el foro más tarde, cuando finalmente pueda cambiar el desorden de cables en la placa integrada. Para obtener más información, visite la publicación original en https://forum.odroid.com/viewtopic.php? f=182&t=37944.

Ejecutando Linux bajo Android

2 April 1, 2020 🛔 By Adrian Popa 🗁 Android, Linux, Tutoriales



El ODROID-N2 parece un sistema de TV Android perfecto: todo funciona sin problemas y convierte un televisor normal en un televisor inteligente muy valioso. Todo funciona perfecto si eso es lo único que quieres hacer con tu ODROID-N2, pero para mí, no es suficiente. Quería usar el ODROID-N2 para reemplazar un ODROID-C2 que ejecutaba Linux, tenía algunos sensores (temperatura y PIR) conectados a sus pines GPIO y también ejecutaba Kodi. Podría haber usado Linux en el ODROID-N2, pero la experiencia de Android TV es mejor como centro multimedia.

Así que me embarqué en un nuevo viaje para aprender cómo hacer que Android y Linux coexistieran en el mismo hardware y así poder transferir fácilmente mis scripts de Linux que maneja los datos del sensor del ODROID-C2 al ODROID-N2.

Si estás empezando tu proyecto desde cero, deberías analizar primero la implementación de la API de Android Things de Hardkernel

(https://magazine.odroid.com/article/android-

things/) y crear aplicaciones nativas de Android que usen los pines GPIO, pero en mi caso ya disponía de algunos scripts de Python que leían datos de temperatura y movimiento, de modo que quería reutilizarlos (https://bit.ly/2xhnGdH).

Instalar Linux bajo Android TV

Para exportar scripts existentes a Android es mejor instalar un entorno Linux en un chroot. La mejor aplicación para trabajar es Linux Deploy (https://bit.ly/2UbeKPV) que puedes encontrar en Play Store, pero desafortunadamente no está disponible para Android TV. Deberás descargar el apk de una fuente diferente (utilicé apkpure https://bit.ly/2wuiHGq) e instalarlo en ATV. Para instalar un paquete apk tienes dos opciones:

 Copia el archivo apk en una unidad USB, conéctala al ATV y usa un explorador de archivos (hay algunas opciones disponibles en Play Store) para instalarlo. O instálalo a través de adb (ya sea con un cable usb o a través de la red):

```
$ adb connect odroid-ip
$ adb install ~/downloads/Linux
Deploy_v2.6.0_apkpure.com.apk
```

Yo estoy usando la imagen ATV de voodik (https://bit.ly/3bkj3OI) con adb habilitado en red.

Una vez instalado Linux Deploy, usa un ratón/teclado (no está diseñado para ATV y la entrada del mando a distancia no cubre todas las opciones) para configurar e implementar la distribución la Linux que elijas en el ODROID. Si no tienes acceso a un dispositivo de entrada adecuado, puede instalar scrcpy (https://github.com/Genymobile/scrcpy) y acceder a Android de forma similar a un escritorio remoto desde tu ordenador. También es posible que necesites instalar un launcher adicional para ver aplicaciones que no sean ATV, como el Sideload Launcher

(https://play.google.com/store/apps/details? id=eu.chainfire.tv.sideloadlauncher&hl=en).

Inicia la aplicación y selecciona el icono de propiedades y configura tu entorno. Ten en cuenta que requiere acceso root, así que asegúrate de otorgarle root. Aquí puede elegir tu distribución favorita (yo usé Ubuntu 18.04), puede elegir el tamaño de la imagen rootfs (yo elegí 8G) y configurar las credenciales por defecto

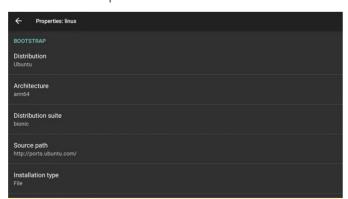


Figura 1 - Propiedades de configuración de implementación de Linux

En la sección INIT, asegúrate de habilitar un sistema de inicialización. Yo opté por sys-v que usa scripts en /etc/rc3.d/.

Bajo las opciones de montaje, quería mostrar algunas configuraciones de Android hacia la imagen invitada de Linux, así que asigné / y /sdcard como puntos de montaje internos en Linux. Afortunadamente, /dev y /sys ya están compartidos.



Figura 2: Puntos de montaje de la implementación de Linux

En teoría, puedes instalar en un directorio y beneficiarte de todo el espacio del host, pero en mi caso esa opción no funcionaba

Cuando estés listo, puedes seleccionar Iniciar y el programa empezará a descargar los paquetes necesarios y creará una imagen mínima. Esto puede llevar un tiempo, pero cuando esté listo, puede continuar y empezar con esa imagen. Si has habilitado ssh, ahora deberías poder iniciar sesión en la IP de tu ODROID con ssh en el puerto 22. Ahora puedes usar apt para actualizar la lista de paquetes e instalar los paquetes que necesites.

En mi caso, necesitaba python3 y un montón de módulos de python, así que los instalé con:

```
# apt-get install python3-pip python3-paho-mqtt
```

Como necesito usar los pines GPIO, quería instalar la rama wiringpi de Hardkernel para el ODROID-N2. La forma más sencilla de hacerlo (sin compilar nada) es tomarlo del PPA de Hardkernel tal y como se describe en la wiki en https://wiki.odroid.com/odroid-n2/application_note/gpio/wiringpi:

```
# apt install software-properties-common
# add-apt-repository ppa:hardkernel/ppa
# apt update
# apt install odroid-wiringpi odroid-wiringpi-
python
```

Una vez instalado, verifica que wiringpi pueda acceder a los pines GPIO y leer su estado

```
# gpio readall
```

Si todo está en su lugar, transfiere tus scripts y conecta el hardware. En mi caso, usr un sensor de temperatura DS18B20

(https://www.adafruit.com/product/381) que utiliza un cable para la comunicación y un sensor PIR HC-SR501 (https://www.hardkernel.com/shop/motion-detector-hc-sr501/) El sensor PIR es el más sencillo para trabajar, ya que requiere un pin GPIO de entrada que te permite leer su estado. Yo estoy usando el pin físico 12, que corresponde al pin wiringpi 1 si echas un vistazo a la distribución

(https://wiki.odroid.com/odroid-n2/hardware/expansion_connectors).

El sensor de temperatura es algo más complicado porque necesita soporte de un único hilo de ejecución en el kernel. En mi caso, el kernel ATV en este momento no tenía habilitado el soporte de único hilo:

```
$ zcat /proc/config.gz | grep -i W1
# CONFIG_W1 is not set
```

Echa un vistazo a la próxima sección sobre compilación del kernel y flasheo.

Una vez que tus programas ejecutan correctamente, debes iniciarlos cuando se inicia Linux (Linux Deploy se encarga de iniciarlos cuando Android se inicia si activas el inicio automático en la configuración de Linux Deploy). Desafortunadamente, puesto que Linux Deploy se ejecuta en un chroot si usas el comando service en Linux para iniciar y detener servicios, recibirás un aviso de que se está ejecutando en un chroot y no hará nada. En mi caso, necesitaba únicamente dos scripts, así que tuve que crear sys-v para mis procesos. Puedes obtener una plantilla de los script en https://github.com/fhd/initscript-template. /etc/init.d/pir-mqtt-agent, Crea configúralo para que apunte a tu script, hazlo ejecutable y vincúlalo a /etc/rc3.d:

```
# chmod a+x /etc/init.d/pir-mqtt-agent
# cd /etc/rc3.d/
# ln -s ../init.d/pir-mqtt-agent S10pir-mqtt-agent
```

SYS-V puede parecer arcaico, pero ha funcionado muy bien durante los últimos 20 años, y es más fácil de entender que systemd. Sin embargo, ¿qué sucede si los scripts fallan por alguna razón (por ejemplo, el sensor no está, la red está offline, etc.)? Sin systemd para hacer el servicio de mantenimiento, no hay nada

que reinicia tus scripts muertos. Opté por la ruta rápida y chapucera y configuré cron para reiniciar periódicamente los scripts (incluso si no están activos). Esto funciona y cubre mis necesidades, aunque hay que tenerlo en cuenta a la hora crear un sistema de gestión de procesos más robusto.

```
# crontab -1
0 * * * * /usr/sbin/service pir-mqtt-service
restart
```

Compilar el kernel de Android

Compilar el kernel de Android es casi lo mismo que compilar el kernel de Linux, aunque tiene algunas peculiaridades. Algunos de los cambios más importantes son:

- La compilación se realiza en un sistema x86_64 (compilación cruzada)
- zlmage necesita combinarse con initrd y empaquetarse en una imagen de arranque de 16M
- La imagen de arranque debe actualizarse a la partición de arranque de Android
- El dtb vive en una partición dtbs y necesita ser flasheado también en caso de que le estés haciendo cambios.

Primero deberás configurar tu entorno de compilación cruzada. Suponiendo que estés en un sistema Ubuntu 18.04 x86_64, puede:

\$ sudo apt-get -y install bc curl gcc git libssl-

```
dev
    libncurses5-dev lzop make u-boot-tools device-
tree-compiler
$ mkdir ~/toolchain
$ cd ~/toolchain
$ wget
https://releases.linaro.org/components/toolchain/b
inaries/6.3-2017.02/aarch64-linux-gnu/gcc-linaro-
6.3.1-2017.02-x86_64_aarch64-linux-gnu.tar.xz
$ unxz gcc-linaro-6.3.1-2017.02-x86_64_aarch64-
linux-gnu.tar.xz
$ tar xvf gcc-linaro-6.3.1-2017.02-x86_64_aarch64-
linux-gnu.tar
```

A continuación, descárgate la fuente del kernel de Android:

```
$ mkdir ~/development
$ cd ~/development
$ git clone --depth 1
https://github.com/hardkernel/linux.git -b
odroidn2-4.9.y-android
$ cd linux
```

Ahora, ajusta algunas variables de entorno para que el entorno de compilación sepa que necesita usar el compilador cruzado. Deberás volver a ejecutar estos comandos cada vez que inicies un nuevo terminal (no se mantienen):

```
$ export ARCH=arm64
$ export CROSS_COMPILE=aarch64-linux-gnu-
$ export PATH=~/toolchain/gcc-linaro-6.3.1-
2017.02-x86_64_aarch64-linux-gnu/bin/:$PATH
```

Ahora puedes empezar desde una configuración base, generalmente dentro del directorio de Linux:

```
$ make odroidn2_android_defconfig
```

Si deseas empezar con una configuración diferente, puedes extraer la configuración actual del kernel del sistema Android en ejecución desde /proc/config.gz, comprimirla y guardarla como ~ /development/linux/.config.

Ahora puedes ejecutar

```
$ make menuconfig

adrianp@frost:/home/adrianp/development/linux -- + ×

.config - Linux/arm64 4.9.113 Kernel Configuration

Arrow keys navigate the menu. <finter> selects submenus ---> (or empty submenus ---). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <N> modularizes features. Press <Esc> Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

General setup --->

[*] Enable loadable module support --->

Evaluation = Platform selection --->

Bus support --->

Bus support --->

Bus support selection --->

Bus support --->

Power management options --->

Userspace binary formats --->

Power Management --->

Pub Power Management --->

**Select** < Exit > < Help > < Save > < Load >
```

Figura 3 - make menuconfig

Aquí puedes buscar los elementos que necesitas y habilitarlos. Por razones de eficiencia, puede localizar una opción de configuración en el menú escribiendo '/' junto con el nombre de la opción (por ejemplo, en mi caso W1 - abreviatura de onewire). El resultado te indicará dónde puede encontrar esa opción (en mi

caso, en drivers de dispositivo), cuál es su estado actual y si depende de otras opciones (y cuáles son sus estados). Si no puedes encontrar una opción en el menú, lo más probable es que esté oculta porque no se cumplen sus dependencias. Usando la función de búsqueda puede encontrar la dependencia que falta.



Figura 4: Buscando opciones de kernel

Una vez que llegue al menú, puede encontrar el elemento presionando ALT y la letra de acceso directo resaltada y deberías navegar a través de todas las opciones en el menú visible que comparte el mismo acceso directo.

Cuando habilitas una opción, generalmente tiene dos elecciones: integrada o como módulo. Tradicionalmente, Linux usa módulos y tiene la capacidad de cargar automáticamente los módulos a medida que se necesiten, pero Android es un poco diferente y, aunque puede usar módulos, no los carga automáticamente. La opción más fácil es compilarlo, pero si necesitas demasiadas opciones, puede hacer que el kernel crezca demasiado y no se ajuste (junto con el initrd) a la partición de arranque de 16M.

Si va a compilar módulos, deberás ubicarlo manualmente en el entorno de compilación (por ejemplo, drivers/w1/masters/w1-gpio.ko) y copiarlo en /vendor/lib/modules/ en Android. Ten en cuenta que la partición /vendor se monta con solo lectura y debe volver a montarse antes de copiar:

:/ # mount -o remount,rw /vendor \$ adb push drivers/w1/masters/w1-gpio.ko /vendor/lib/modules/

Para cargar nuevos módulos al inicio deberá editar init.odroidn2.system.rc tal y como se describe en:https://github.com/codewalkerster/android_device

_hardkernel_odroidn2/blob/s922_9.0.0_master/init.od roidn2.system.rc#L96

Para cubrir mis necesidades (soporte onewire) necesitaba habilitar los siguientes módulos (y tenerlos integrados):

```
CONFIG_W1
CONFIG_W1_MASTER_GPIO
CONFIG_W1_SLAVE_THERM
```

Ok, ahora que la configuración del kernel está lista, puedes guardarla y comenzar a compilar el kernel:

```
$ make -j$(nproc)
```

La compilación debería llevar un tiempo la primera vez, pero las ejecuciones posteriores deberían compilar únicamente las diferencias. Echa un vistazo a cualquier error, si se compila correctamente, debería obtener una

~/development/linux/arch/arm64/boot/Image.gz.

```
$ 1s -1
~/development/linux/arch/arm64/boot/Image.gz
```

Si necesitas realizar cambios en el dtb (por ejemplo, habilitar o cambiar la configuración), puede usar fdtput (parte del paquete del compilador del árbol de dispositivos) para realizar cambios en el dtb. En mi caso, necesitaba habilitar explícitamente el soporte para un único hilo de ejecución en el dtb, de lo contrario el controlador no haría nada:

```
$ fdtput -t s
~/development/linux/arch/arm64/boot/dts/amlogic/me
son64_odroidn2_android.dtb /onewire status okay
```

Una vez que el kernel y el dtb estén en orden, debemos preparar el bootimg. Para hacer esto, primero extrae la partición de arranque de la versión ATV en funcionamiento, para que podamos mantener el initrd de la imagen original. A continuación, instalamos bootimg, que es una herramienta para empaquetar y desempaquetar imágenes de arranque de Android.

```
$ adb shell
# if=/dev/block/boot of=/dev/shm/boot.img
# exit
$ mkdir ~/development/bootimg
$ cd ~/development/bootimg
$ adb pull /dev/shm/boot.img
$ sudo apt-get install abootimg
```

Ahora podemos inspeccionar boot.img y extraer el initrd y la configuración:

```
$ file boot.img
boot.img: Android bootimg, kernel (0x1080000),
ramdisk (0x1000000), page size: 2048, cmdline
(otg_device=1 buildvariant=userdebug)
$ abootimg -x boot.img
```

El último paso es crear el nuevo bootimg con el nuevo kernel:

```
$ abootimg --create boot-new.img -f bootimg.cfg -k
~/development/linux/arch/arm64/boot/Image.gz -r
initrd.img
```

Ok, ahora estás listo para grabar el nuevo kernel y dtb:

```
$ adb push boot-new.img /dev/shm
$ adb push
~/development/linux/arch/arm64/boot/dts/amlogic/me
son64_odroidn2_android.dtb /dev/shm
$ adb shell
# dd if=/dev/shm/boot-new.img of=/dev/block/boot
# dd if=/dev/shm/meson64_odroidn2_android.dtb
of=/dev/block/dtbs
```

Una vez que reinicies, tu nuevo kernel debería hacerse cargo del sistema, puedes verificar que cuenta con la funcionalidad que necesitas. Si no se reinicia, deberás recuperar la placa al actualizar el boot.img anterior mediante el modo fastboot. ¡Disfruta de tu servidor Linux con una GUI de Android TV!

¿Vamos a jugar a un juego? - Nintendo Drags Pokémon en Pleno Siglo 21...Pateando y Gritando

② April 1, 2020 ♣ By Dave Prochnow ▷ Juegos, Tutoriales



Imagina poder compartir, intercambiar y mover tus Pokémon entre tus juegos en diferentes dispositivos a través de un servidor centralizado conectado a Internet. Suena a algo realmente excéntrico, impresionante, futurista, ¿eh? Bueno, quizás si este titular se hubiera anunciado a principios de 1999, podría haberlo sido. Más bien, es el anuncio del "servicio" completamente válido que Nintendo ha lanzado recientemente. Incluso tienen la temeridad de etiquetar este servicio en la nube como Pokémon HOME.

Ubicado en http://home.pokemon.com, el sitio web ofrece una introducción a Pokémon HOME, así como una presentación de los dos servicios de suscripción disponibles que muestra las diferencias entre las versiones de Nintendo Switch y Android. Para usar la versión de Android y aprovechar el nuevo servicio en la nube, deberás ir a Google Play y descargar la aplicación gratuita Pokémon HOME y suscribirte.



Figura 1 - Pokémon HOME

Sobre la marcha, tiene una opción para tu plan de suscripción: gratis o de pago. El plan gratuito es lo suficientemente rico en funciones como para usarlo de inmediato con tu colección de Pokémon. Puedes optar fácilmente por el plan gratuito y probar las aguas de Pokémon HOME antes de lanzarte con tu tarjeta de crédito.

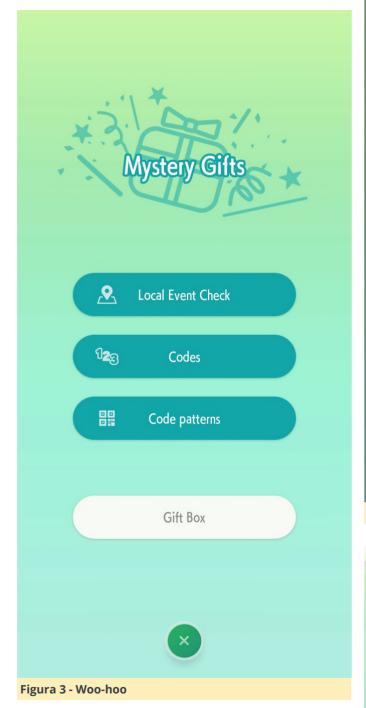
Los planes disponibles son Básico (el plan gratuito) y Premium (el plan de pago). La suscripción al plan Premium está disponible por un mes a 2.99\$, tres meses por 4.99\$ y un año por 15.99\$. Además de las diferencias entre el plan básico y el plan Premium, también hay diferencias en las características de la plataforma entre Nintendo Switch y Android. Algunas de las diferencias de características entre los planes y las plataformas son:

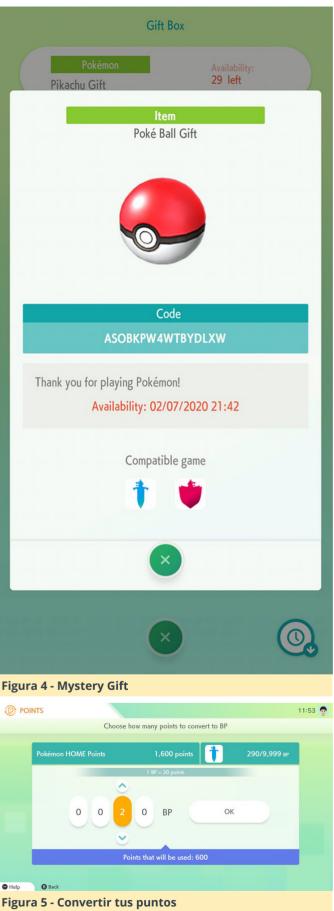
- 1. Mover Pokémon desde el Pokémon Bank patrocinado por 3DS solo está disponible con una suscripción Premium. Esta es una característica incluida para Android.
- 2. La cantidad de Pokémon que se pueden depositar en Pokémon HOME está limitada a 30 en el plan Básico y a 6.000 en el plan Premium.
- 3. El número de Pokémon que se pueden colocar en la función Wonder Box al mismo tiempo está restringido a 3 para usuarios básicos y 10 para usuarios premium.
- 4. La cantidad de Pokémon que se pueden colocar en el GTS (también conocido como Global Trade System) a la vez son 1 con Basic y 3 con Premium. Esta característica también está habilitada para Android.
- 5. Los usuarios básicos solo pueden usar salas prefabricadas en Room Trade, mientras que los usos Premium también pueden crear nuevas salas de usuarios. Los usuarios de Android tendrán esta posibilidad
- 6. Solo los suscriptores Premium pueden usar la función Judge. Esta característica está disponible para Android.



En la actualidad, la versión de Android de Nintendo Pokémon HOME solo admite transferencias del Pokémon Bank de Nintendo 3DS, ¡pero NO Pokémon Let's Go Eevee! o Pokémon Let's Go Pikachu! Aunque se especula que la integración de Pokémon GO pronto se lanzará para dispositivos móviles. Por lo tanto, los jugadores inteligentes ODROID pueden optar por el plan básico gratuito hasta que se haya activado la opción Pokémon GO y luego decidir si continuar con el plan básico o actualizar al plan Premium.

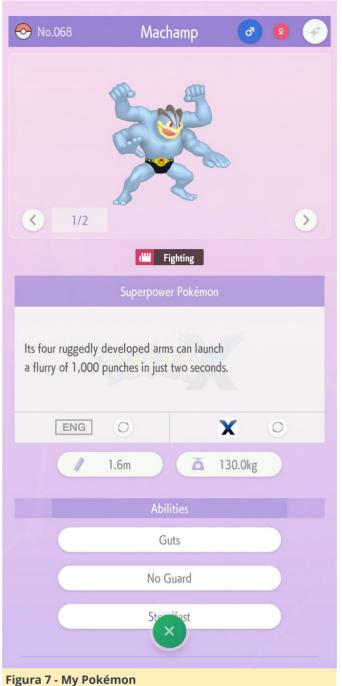
Junto con las transferencias dentro del juego Pokémon de Pokémon HOME, también hay características para intercambiar Pokémon a través del Global Trade System y eventos de Room Trade, la posibilidad de examinar Pokémon en la Wonder Box, la oportunidad de recibir regalos "misteriosos" gratuitos, y convertir tus puntos acumulados en Battle Points (BP). Si bien los usuarios de Android podrán recibir Mystery Gifts, el intercambio de Battle Point se limita solo a la versión de Nintendo Switch.





Una última característica digna de mencionar es que dentro de Pokémon HOME hay la posibilidad de registrar todos tus Pokémon agregados en la National Pokédex. Dentro de este diccionario, puedes leer acerca de las habilidades y movimientos de Pokémon específicos que los Pokémon pueden aprender. Esta es una inclusión encantadora e informativa que se encuentra dentro de la versión de Android. También puede registrar los formularios Mega Evolve y Gigantimax en esta base de datos móvil





Si bien parece que Nintendo tardó una eternidad en ponerse al día con el resto de la industria del juego en lo que respecta a los juegos basados en la nube, Pokémon HOME es un producto rico en funciones que permite la adopción generalizada del ecosistema Pokémon entre jugadores activos. Aún más notable, esta experiencia compartida podría crecer más de 1000 veces con la llegada de la integración propuesta de Pokémon GO. Eso se traducira en ver incluso a más personas deambulando sin rumbo mirando su dispositivo inteligente en busca de otro gimnasio Pokémon. Gracias Nintendo y bienvenidos al siglo XXI.

Montando una Xbox Usando un ODROID-H2

② April 1, 2020 🎍 By @teeedubb 🗁 ODROID-H2, Tutoriales



Montar un clon de Xbox es algo que he querido hacer desde hace mucho tiempo, pero nunca lo he logrado por varias razones, y ahora finalmente lo he conseguido. A continuación, te mostrare básicamente el registro del montaje, porque sé que, si no lo hago ahora, todo lo que mostraría sería un par de fotos de un proyecto terminado



Figura 1: Montar un clon de Xbox usando un ODROID-H2

¿Por qué estoy haciendo esto? El OG Xbox conmociono todo nuestro mundo: aún seguimos llamando al reproductor multimedia 'la xbox' a día de hoy. Después de ser modificada, se convirtió en una potencia mediática muy adelantada a su tiempo. Permite cargar juegos desde el HDD, emuladores y, por supuesto, XBMP. Desafortunadamente, debido a los avances de la tecnología, tuvo que ser retirada. Con mi carcasa s912 me estaba dando dolores de cabeza y la placa que quería finalmente estaba de nuevo en stock, decidí que era hora de empezar con este proyecto y llevar la Xbox a la habitación de estar.

El hardware que he usado es un ODROID-H2. Elegí este por la GPU Intel, que tiene un gran soporte en Linux, y con soporte HDR próximamente. Además, la CPU x86 supera a muchos sistemas ARM cubriendo más posibilidades en cuanto software, como emuladores, Firefox, Steam y Project M (¡No es un reproductor multimedia si no tiene Project M!). La placa cuenta con 2 salidas de video (HDMI2.0 y DP ++) y con un adaptador pasivo de DP a HDMI, el puerto DP++ se convierte en un puerto HDMI que me permite enviar video 4k60 a mi televisor y enviar

audio passthrough a mi AVR HDMI 1.4 al mismo tiempo. La placa funciona bien, tiene una fuente de alimentación externa y es pequeña, lo cual me permitirá introducir algunas cosas en la carcasa y mantenerla ordenada y silenciosa al mismo tiempo. Pensé en una placa mini-itx, pero el ODROID-H2 tiene casi todo lo que necesito a un precio bastante razonable.

El plan es mantener el exterior bastante abastecido. No voy a utilizar una unidad óptica, de modo que la placa frontal del DVD se adjuntará a la carcasa. El LED se cambiará a blanco y los botones frontales se usarán para encender y reiniciar. El botón más pequeño tiene una impresión de encendido y el más grande tiene una impresión de expulsión y voy a pintarlos de un color similar dejándolos en blanco; quiero usar el botón grande para encender. Pensé en conectar la alimentación y restablecer las calcomanías a los botones, pero no creo que sea tan importante ya que el 99% del tiempo se usará un mando para encenderlo. Los dos puertos izquierdos controlador estarán cubiertos con un poco de acrílico ahumado, con un receptor IR escondido detrás de uno y pensé en poner un puerto para auriculares en el otro, aunque nunca he tenido la necesidad de tener uno, dispongo de una tarjeta de sonido USB o un USB DAC y amplificador que se pueden usar si alguna vez lo necesitase. Los dos puertos para los mandos serán un puerto USB 2 y 3, que tampoco necesito a menudo, pero serán útiles para la conectar una unidad USB ocasionalmente, los mandos con cable y para cargar los mandos DS3 que usaré. Pensé en agregar algún tipo de marca Kodi en el frontral, pero creo que la superficie texturizada dificultaría obtener un buen acabado, por lo que probablemente no me centre en ello.

En el interior estará la placa, 2x 4gb RAM, un SSD de 60gb para el sistema operativo y un HDD de 2.5" de 320gb para algunas copias de seguridad y almacenamiento multimedia. Estas son unidades que tenía como repuestos, el de 320gb se llenará muy rápido, probablemente lo cambiaré muy pronto, tal vez a un viejo HDD de 3.5 "que tengo. Usaré la ranura NVME para una tarjeta Intel WiFi plus BT 4 (7265ngw) a través de un adaptador, lo que me permitirá usar

rango BT para los mandos. También habrá un hub USB3 de 4 puertos en el interior para proporcionar puertos para el receptor IR, el módulo BT, un arduino para controlar la retroiluminación de la TV y un receptor unificador Logitech. Los puertos USB frontales se conectarán directamente a los puertos posteriores, lo que dejará 1 puerto USB2 de repuesto en la parte posterior. El ODROID-H2 aproximadamente 10x10cm y la Xbox mide aproximadamente 35x30cm, por lo que no veo ningún problema para ajustar componentes. Compré un ventilador de 5V y 92 mm con el ODROID-H2, demasiado grande para ubicarlo en la misma ubicación del ventilador original, así que lo voy a montar en el disipador térmico. Sobresaldrá un poco, de modo que moverá un poco de aire alrededor de toda la carcasa, lo cual ayudará a mantener frescos el resto de componentes. Estaré al tanto de las temperatura y agregaré un ventilador trasero si fuera necesario, pero muchas personas ejecutan estos ventiladores sin problema, así que no creo que sea un problema.

El sistema operativo utilizado será Xubuntu 19.10, por ahora, y lo actualizaré a la versión 20.04 LTS y luego lo mantendré en las versiones LTS. Como se utilizarán múltiples aplicaciones, quería un entorno de escritorio simple, liviano У personalizable. Generalmente ejecuto versiones LTS para este tipo de aplicaciones, pero al principio tuve algunos problemas e intenté algunas versiones de Xubuntu + Ubuntu + Libreelec mientras solucionaba el problema y finalmente lo solucioné con la 19.10 y con 20.04 tan cerca, decidí quedarme con él porque no quería hacer otra reinstalación. Si supiera lo que sé ahora, me quedaría con 18.04 porque tiene más software precompilado disponible (por ejemplo, modo de atracción, estación de emulación y un PPA de emulación).

Puesto que estamos ante una configuración de monitor dual, aunque no en el sentido tradicional ya que los dos monitores son un único televisor, he desactivado el uso del monitor AVR mediante algunos trucos xrandr porque tener un segundo monitor que no era fácilmente accesible era bastante molesto. La

dobles antenas internas y proporcionar un buen duplicación no funciona debido a las diferentes rango BT para los mandos. También habrá un hub USB3 de 4 puertos en el interior para proporcionar deshabilitar el segundo monitor también puertos para el receptor IR, el módulo BT, un arduino para controlar la retroiluminación de la TV y un con esto es que el conmutador alt-tab aparece dos receptor unificador Logitech. Los puertos USB veces, lo cual no es un gran problema porque frontales se conectarán directamente a los puertos realmente solo ejecutaré un programa gráfico al posteriores, lo que dejará 1 puerto USB2 de repuesto mismo tiempo, aunque buscando por google parece en la parte posterior. El ODROID-H2 mide ser que hay algunas alternativas posibles para aproximadamente 10x10cm y la Xbox mide solucionar esta cuestión.

Hasta ahora, me ha sorprendido bastante esta pequeña máquina y la recomiendo para el uso de HTPC: reproduce contenido 4k60 sin problemas, HDR próximamente y no tengo que comprar un nuevo AVR para conseguir video 4k con audio passthrough. Tengo un script que cambia automáticamente las salidas de audio dependiendo de si el AVR está encendido o apagado y, a diferencia de sistema s912, no envía audio a múltiples salidas al mismo tiempo, lo cual era de esperar. Falta rendimiento 3D, pero es algo con lo que ya contaba. A 4k, el GUI Kodi + Project M lag, aunque se suaviza a 1080p. A 1080p, el mapa 3D de Broforce aparece entrecortado, pero las secciones 2D son suaves. Super Meat Boy funciona sin problemas a 1080p. Intenté Portal 2 pero se carga con una pantalla en blanco. He leído que usar DXVK puede mejorar el rendimiento, pero aún no he profundizado en ello. Además, siempre existe la posibilidad de hacer Steaming en un entorno doméstico para cualquier cosa que sea demasiado exigente.

Para finalizar, algunas fotos del ODROID-H2 en su ubicación temporal. A continuación, desmantelaré la Xbox y encontraré un hogar para los componentes no utilizados.

Como he estado usando el ODROID-H2 durante algunos meses, aquí tienes mis impresiones sobre el dispositivo:

Como he mencionado anteriormente, lo estoy usando como HTPC y para mi propósito, el ODROID-H2 es increíble y cumple todos los requisitos: reproducción impecable de kodi 4k60 HDR (he probado una versión de prueba de libreelec y HDR funciona en esta placa), CPU x86, emulación y steam, navegadores web, puertos SATA y gemelos HDMI / DP ++, disponibilidad

de Linux principal y probablemente mucho más. La CPU Intel J4105 tiene problemas con los gráficos 3D y la emulación más pesada (por ejemplo, gzdoom no funcionaría con 60 fps a 1080p). He descubierto que 720p con juegos 3D es posible con juegos simples (por ejemplo, LEGO Starwars funciona bien, pero al Serious Sam 3 BFE no se puede jugar). La emulación PSX, DC funciona a toda velocidad, pero Super Mario Sunshine estaba muy entrecortado y Mario Kart Wii corría cerca de 60 fps, pero el nuevo Super Mario se entrecortaba

El uso de escritorio con el SSD SATA es lo suficientemente ágil, sin embargo, realmente no he hecho nada demasiado exigente con ello. Steaming a nivel doméstico funciona bien y usa vaapi, pero los navegadores web no usan vaapi, por lo que tendrá dificultades para reproducir YouTube a 4k a través de un navegador (existe una compilación de Chromium con soporte vaapi, pero no funciona a nivel local).

La temperatura oscila alrededor de los 50 °C, en mi caso, y el ventilador está en silencio (he ajustado la curva del ventilador en 10'). Solo desearía que esta placa tuviera una ranura NVMe dual, como he visto en algunas placas base, para que acepte tanto una tarjeta NVMe como WiFi / BT. Esto le daría al ODROID-H2 una gran conectividad sin el uso de adaptadores. Si bien esta máquina me durará varios años, espero que Hardkernel continúe fabricando este tipo de placas y posiblemente placas base Mini-STX sin CPU integrada.



Figura 2: Componentes del donante XBox que se utilizarán



Figura 3 - Desafortunadamente, se encendió, pero no emitía video, y el DVD-ROM tampoco se abría



Figura 4: Componentes que no usaré, con un chip de modificación instalado. La consola no se inició, así que no sabía en qué estado se encontraban estos componentes



Figura 5: Lo compré una o dos semanas después del lanzamiento. Esta es la mejor consola de todas, en mi opinión



Figura 6 - Primero, coloqué el botón frontal y lo pinté. Los botones están sucios, pero mi espacio de trabajo estaba limpio en ese momento



Figura 7: Se ha aplicado masilla



Figura 8: Así es como se ve, después de dos capas de masilla y lijado con una lija gruesa para mantener la textura.



Figura 9: Así es como se muestra después de cubrir los botones con 'plata satinada'. Era una buena combinación con el color de la tapa.



Figura 10 - Opte por un color "gris medio" en lugar de plateado

El color plateado resultó ser demasiado brillante, así que cambié a un 'gris medio'. El botón ahora tiene una textura diferente y aunque el color no es una combinación perfecta, es suficiente. Todo lo que se puede ver cuando está instalado en la carcasa, ha sido pintado. Puede ver los contornos de los botones originales si miras lo suficientemente cerca, pero a unos 50 cm ya no podrás verlos.

Referencias

https://imgur.com/gallery/oSOHuBt#LdS14Ui

Retro-Go, Otro Lanzador Emulador para ODROID-GO

② April 1, 2020 🎍 By @ducalex 🗁 Juegos, ODROID-GO



Retro-Go es un lanzador con emuladores NES/GB/GBC/SMS/GameGear/Colecovision. El código del emulador está basado en Go-Play "Triforce" y el diseño del lanzador está bastante inspirado en el de pelle7; Sin embargo, no es una bifurcación

Características

- Menús dentro del juego
- Transiciones más suaves/rápidas
- Ajustes RTC y conserva el tiempo entre jugadas
- El estilo del lanzador es personalizable
- Ocultar emuladores no deseados
- Soporte NES PAL
- Compatibilidad mejorada de juegos
- Más opciones de escalado
- Filtrado bilineal
- Se han corregido muchos bloqueos
- Fácil de compilar
- ¡Y mucho más!

Detalles

Te animo a leer el archivo README (https://github.com/ducalex/retro-go/blob/master/README.md) para ver las asociaciones de teclas y más detalles.

Código, activos y lanzamientos:

Repositorio: https://github.com/ducalex/retro-go
Lanzamientos: https://github.com/ducalex/retro-

go/releases



Figura 1 - Vista previa de capturas de pantalla

Para ver el artículo original y el último enlace de descarga, consultea el siguiente post en: https://forum.odroid.com/viewtopic.php? f=159&t=37599

Retro Roller: Una Experiencia de Juego Optimizada para tu ODROID-GO Advance



Retroroller es una imagen precompilada para ODROID-GO Advance que proporciona RetroArch 32 bits y 64 bits en la única imagen estándar de CrashOverride. Entre otros ajustes, tiene integrado un kernel personalizado que soporta el modo reposo. La idea con esta imagen es poder actualizarla a través de paquetes en lugar de grabarla de nuevo. Ten en cuenta que cualquier actualización importante del sistema operativo cómo lo será la próxima imagen de Ubuntu de CrashOverride requerirá una actualización importante. Esta imagen incluye lo que yo suelo ejecutar en mi ODROID-GO Advance y, como tal, lo más probable es que no la convierta en un sumidero de aplicaciones como ocurre en otras imágenes.



Figura 1 - La imagen de juego de Roller Retro para ODROID-GO Advance está basada en RetroArch

Como actualizar

Si te indica que actualices los archivos de configuración, elije la acción que quieras. Probablemente deberías hacer una copia de seguridad de los archivos que modificante primero, incluidos los archivos de configuración y los scripts de RetroArch hasta que te sientas cómodo con las actualizaciones de dpkg. Tan solo necesitas descargar

el último paquete de Deb e instalarlo. Esto significa que no tiene que instalar cada versión hasta llegar a la última.

En los siguientes comandos, sustituye la versión del paquete más reciente:

```
$ sudo dpkg -i retroroller_.deb
$ sudo apt-get update && sudo apt-get -f install
```

Atajos de teclado

Puedes utilizar las siguientes combinaciones de teclas:

```
F3 + left/right -> volume
F3 + up/down -> brightness
power -> suspend
F3 + power -> shutdown
F3 + L -> perfnorm
F3 + R -> perfmax
```

RetroArch también tiene teclas específicas:

```
F3 -> Hotkey
L + R -> Menu
```

Puedes usar la opción "Customize Settings" -> "nput" para cambiar las combinaciones de las teclas a tu gusto.

Características

- Compatibilidad con aplicaciones de 64 bits y 32 bits (gracias a la imagen estándar de CrashOverride)
- RetroArch 64 bits y 32 bits preinstalados
- El led azul parpadeante está apagado
- perfmax también ajusta el regulador mem en high
- Kernel personalizado con modo reposo habilitado
- Configuración de aplicaciones RetroArch para descargar núcleos del sitio de Safarikniv
- Es_system.cfg de NOirBRight utiliza retroarch cuando está disponible
- Apagar/Reiniciar funciona
- El cambio automático de auriculares/altavoces funciona

Créditos

 CrashOverride por todo su gran trabajo en la creación de la imagen de Stock, libgo2,

- Vrsión retroarch inicial e innumerables cosas para esta plataforma.
- El equipo de Lakka por sus parches, pruebas e integración de RetroArch.
- NOirBRight por sus extensas pruebas y comentarios.
- Safarikniv para alojar núcleos RetroArch de 32 bits y 64 bits para descargar.

¡Las peticiones y comentarios de la comunidad son bienvenidas!

Versiones

La versión inicial y la imagen basada están disponibles

en https://mega.nz/#!etlnUTjT!nxOhiLzG03jbXgcyiUeh7_B6ovHBqvDFGWTQ4uR58Ho. Las actualizaciones estan disponibles

en https://github.com/valadaa48/retroroller/releases.

Para actualizar la imagen, debe tener acceso SSH (a través de WiFi o Ethernet). Primero debes copiar el archivo .deb en ODROID-GO Advance usando WinSCP o Filezilla::

```
$ scp retroroller 1.0-2.deb odroid@OGA IP ADDRESS
```

Luego, conéctate por SSH al ODROID usando la línea de comando de Linux en otro ordenador, o usa PuTTY en Windows:

```
$ ssh odroid@OGA_IP_ADDRESS
```

Después, actualiza el paquete:

```
$ sudo dpkg -i retroroller_1.0-2.deb
```

Cuando se te pregunte si desea sobrescribir retroroller_boot.sh, responda Y o I:

Configuración del archivo '/opt/retroroller/scripts/retroroller_boot.sh' ==> Archivo en el sistema creado por ti por un script. ==> Archivo del paquete proporcionado por el mantenedor del paquetes ¿Qué te gustaría hacer con esto? Tus opciones son: Y o I: instala la versión del mantenedor del paquete N o O: mantiene tu versión instalada actualmente D: muestra las diferencias entre las versiones Z: Inicia un shell para examinar la situación La acción por defecto es mantener tu versión actual. *** retroroller_boot.sh (Y/I/N/O/D/Z)

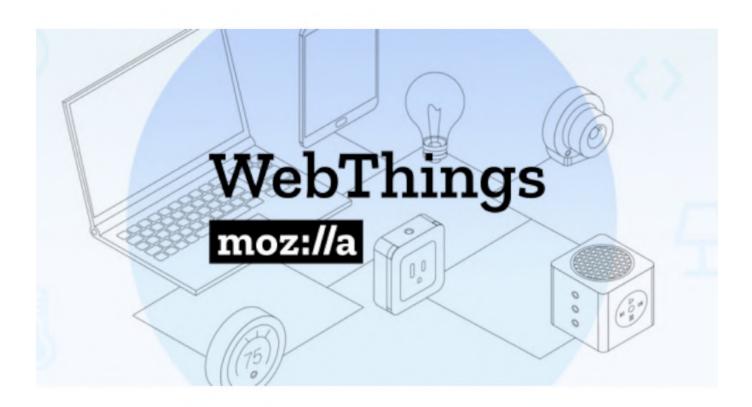
[default=N] ? I Instalando una nueva versión del sugerencias, archivo de configuración en <a href="https://doi.org/10.2016/j.je/http

sugerencias, visita el artículo original en https://forum.odroid.com/viewtopic.php? f=193&t=38140.

El hilo de version no oficial está disponible en https://forum.odroid.com/viewtopic.php?
f=193&t=38016. Para comentarios, preguntas y

WebThings en Armbian: Usando el ODROID-XU4 para el Internet de las Cosas

2 April 1, 2020 By Philippe Coval Desarrollo, ODROID-XU4, Mecaniqueo



Mozilla WebThings es una plataforma para controlar dispositivos domésticos. El objetivo principal es brindar a los usuarios la capacidad de configurar y controlar un "hogar inteligente" por su cuenta sin depender de servicios de terceros. Este artículo te dirá por qué y cómo configurarlo en ODROID-XU4.

¿Por qué?

Durante años, los dispositivos que operan en la nube han tenido limitaciones. Los problemas de interoperabilidad y confiabilidad se salen de este artículo, pero uno que debe abordarse es la privacidad. Personalmente, considero que tomar o revelar (cualquier) información del usuario sin tu permiso explícito es un problema que todos deberían considerar. Incluso si no tienes nada que ocultar, suscita duda el compartir los números de su tarjeta de crédito, de modo que es importante ser cuidadoso antes de adoptar un servicio o conectar dispositivos

porque tu política de privacidad también afectará a otros miembros de la familia o incluso a tus invitados. Hemos visto algunos progresos a la hora de reforzar la privacidad en los últimos años gracias a ciertas regulaciones. Europa comenzó con GDPR, que incluye directrices importantes a la hora de usar software que maneja datos personales. Una muy importante específicamente, está recogida en artículo 25, que exige "Privacidad en el diseño" en los sistemas técnicos. Esto significa que deben estar diseñados para que los usuarios puedan mantener la privacidad desde cero.



Figura 1 - GPDR

¿Quien?

Es fácil predecir que el mercado de dispositivos conectados cambiará tarde o temprano. Por ahora, puedes mantenerte a la vanguardia y probar proyectos FLOSS que no esperaron a las regulaciones que abordaban la privacidad. Desde los inicios del desarrollo de la plataforma Mozilla IoT, el proyecto ha sido diseñado explícitamente para evitar que tus datos vayan a parar a la nube de otra persona.



Figura 2 - NoCloud

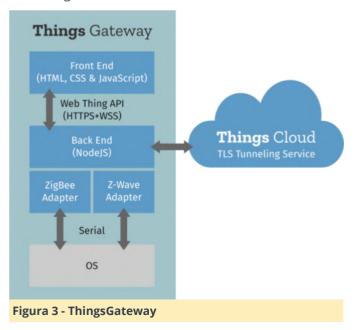
La clave es bastante simple: todos los dispositivos domésticos están conectados únicamente a la red

doméstica, de modo que los datos simplemente permanecen en local, no sale nada del entorno local. Sin entrar en demasiados detalles, existen muchos estándares IoT hoy en día, y probablemente ninguno se ajuste a todos los posibles usos. Dicho esto, una tecnología común puede abstraer muchas superposiciones. Esta es la misión de W3C, la organización conocida por la estandarización de la red mundial (HTML, HTTP y XML). El grupo de trabajo "Web of Things" (WT) del W3C se compromete a abordar algunos problemas de IoT utilizando las tecnologías web existentes y rellenar el vacío que falta. Mientras que algunos describen WoT como el "HTML para IoT", creemos que los protocolos de WoT son principalmente para máquinas en lugar de para humanos, técnicamente estamos hablando de API REST y Esquema JSON para la semántica.

¿Qué?

El marco de WebThings se basa en las recomendaciones propuestas por WoT simplificadas. Actualmente, la implementación se compone de:

WebThings API para dispositivos que hablan REST en HTTP. La puerta de enlace de WebThings puede controlar dispositivos desde una buena aplicación web. Los adaptadores adicionales de la puerta de enlace de WebThings para traducir dispositivos o servicios que no son de WebThings a la API de WebThings.



¿Cómo?

En este artículo, cubriré cómo un ODROID-XU4 puede ser un ordenador de placa reducida totalmente válido para alojar el software WebThings que se desarrolló originalmente para la Raspberry Pi 3.



Figura 4 - ODROID-XU4

Seguir estas instrucciones también debería funcionar para otros ODROID con pequeñas adaptaciones. Si necesitas elegir uno, te sugiero el ODROID-H2, ya que el desarrollo es algo más sencillo en la arquitectura de CPU x86_64. La configuración no es sencilla, ya que necesitarás un ordenador host GNU / Linux, una tarjeta SD, un puerto serie y un enlace Ethernet.

Configurar Armbian

La imagen de referencia de Mozilla está basada en Raspbian, un puerto Debian para Raspberry Pi, esta imagen no funcionará en dispositivos que no sean Pi. Con suerte, el proyecto Armbian acaba de lanzar imágenes para arrancar el último Debian-10 en varias placas ARM. La versión de Debian o Ubuntu de Armbian funcionará en ODROID-XU4, descargaremos la imagen mínima y luego la volcaremos a una tarjeta SD.

En un sistema GNU / Linux, utilicé un adaptador USB SD y estas líneas de comando:

```
$ lsblk # Will list your disks make sure to use
the right one
$ disk='/dev/disk/by-id/usb-Generic-
_USB3.0_CRW_-0_000000021716-0:0' #
$ file "$disk" # TODO
$
release="Armbian_20.02.1_ODROIDxu4_buster_current_
5.4.19 minimal"
```

```
$
url="https://dl.armbian.com/odroidxu4/archive/${re
lease}.7z"

$ sudo sync
$ sudo apt-get install curl 7zip-full time #
Install those tools on Debian or adapt
$ time curl -0 "$url" # 3sec 155313070c
$ time 7z e -so "$release.7z" "$release.img" |
sudo dd bs=4MB of="$disk"
#| 562036736 bytes (562 MB, 536 MiB) copied,
142.735 s, 3.9 MB/s
```

Arranque

Inserta la imagen SD en ODROID-XU4. Si tiene el puerto serie configurado, debería poder ver el registro de arranque:

```
$ screen /dev/ttyUSB0 115200
# U-Boot 2017.05-armbian (Feb 17 2020 - 07:52:44
+0100) for ODROID-XU4
#|
#| CPU: Exynos5422 @ 800 MHz
#| Model: ODROID XU4 based on EXYNOS5422
#| Board: ODROID XU4 based on EXYNOS5422
#| Type: xu4
#| DRAM: 2 GiB
#| MMC: EXYNOS DWMMC: 0, EXYNOS DWMMC: 1
#| MMC Device 0 ( SD ): 7.4 GiB
#| mmc_init: -5, time 4
#| *** Warning - bad CRC, using default
environment
#|
# In: serial
# | Out: serial
#| Err: serial
#| Net: No ethernet found.
#| Press quickly 'Enter' twice to stop autoboot: 0
# | (...)
```

Un par de minutos después, deberías poder iniciar sesión como root con la contraseña por defecto "1234". Luego el sistema te preguntará el usuario predeterminado de configuración:

```
#| Armbian 20.02.1 Buster ttySAC2
#| odroidxu4 login: root
#| Password: 1234
#| You are required to change your password
immediately (root enforced)
#| Changing password for root.
```

```
# | (current) UNIX password:
#| Enter new UNIX password:
#| Retype new UNIX password:
#| / _ __| |_ __ (_) __| | / / | | | | |
#| | | | | | / _ ` | '__/ _ \| | / | ` | / | | | | | | | |
#| | | | | (_| | | | (_) | | (_| | / \| | | | | | __ |
#| ___/ __,_|_ | ___/|__,_| /__/__/ |_|
#| Welcome to Armbian buster with Linux 5.4.19-
odroidxu4
#| System load: 2.23 0.79 0.28 Up time: 1 min
#| Memory usage: 6 % of 1993MB IP: 192.168.1.232
#| CPU temp: 32°C
#| Usage of /: 29% of 7.1G
#|
#| New to Armbian? Check the documentation first:
https://docs.armbian.com
#|
#|
#| Thank you for choosing Armbian! Support:
www.armbian.com
#| Creating a new user account. Press to abort
#| Desktop environment will not be enabled if you
abort the new user creation
#|
#| Please provide a username (eg. your forename):
#| Trying to add user user
# | Adding user `user' ...
#| Adding new group `user' (1000) ...
#| Adding new user `user' (1000) with group `user'
#| Creating home directory `/home/user' ...
#| Copying files from `/etc/skel' ...
#| Enter new UNIX password:
#| Retype new UNIX password:
#| passwd: password updated successfully
#| Changing the user information for user
#| Enter the new value, or press ENTER for the
default
#| Full Name []: User
#| Room Number []:
#| Work Phone []:
#| Home Phone []:
#| Other []:
# | Is the information correct? [Y/n] y
#| Dear User, your account user has been created
and is sudo enabled.
#| Please use this account for your daily work
```

from now on.

Una vez hecho esto, inspeccionemos el sistema e instalemos un servicio DNS multidifusión que ayudará a conectarnos de forma remota mediante SSH:

```
root@odroidxu4:~# cat /etc/os-release
#| PRETTY NAME="Debian GNU/Linux 10 (buster)"
#| NAME="Debian GNU/Linux"
#| VERSION ID="10"
#| VERSION="10 (buster)"
#| (...)
root@odroidxu4:~# df
#| Filesystem 1K-blocks Used Available Use%
Mounted on
#| udev 950928 0 950928 0% /dev
#| tmpfs 204128 6592 197536 4% /run
#| /dev/mmcblk1p1 7505192 498264 6915480 7% /
#| tmpfs 1020628 0 1020628 0% /dev/shm
#| tmpfs 5120 0 5120 0% /run/lock
#| tmpfs 1020628 0 1020628 0% /sys/fs/cgroup
#| tmpfs 1020628 0 1020628 0% /tmp
#| /dev/zram0 49584 632 45368 2% /var/log
#| tmpfs 204124 0 204124 0% /run/user/0
root@odroidxu4:~# sudo apt-get update
root@odroidxu4:~# sudo apt-get install avahi-
daemon
root@odroidxu4:~# reboot
```

Una vez reiniciado, iniciaremos sesión con SSH con la dirección hostname.local en lugar de IP que puede ser diferente para cada uno de nosotros:

```
$ ssh root@odroidxu4.local
```

Almacenamiento extra

Opcionalmente, puedes omitir o usar este truco para preservar la vida útil de la tarjeta SD. Acabo de conectar un par de memorias USB (de 4GB cada una) en los dos puertos USB3 de ODROID. Este almacenamiento masivo USB adicional se usará para intercambiar memoria y docker. Una vez montado de la siguiente forma:

```
$ sudo=$(which sudo)

$ dev="/dev/disk/by-id/usb-Generic_Mass-Storage-
0:0" # TODO
$ file "$dev"
$ fdisk -1 $dev || lsblk # Update previous line
```

```
$ dev='/dev/sda' # TODO update
$ label="docker"
$ yes | ${sudo} mkfs.ext4 -L "$label" "$dev" #
TODO: verify $disk variable
$ dev=/dev/sdb # TODO: update if needed
$ label="swap"
$ fdisk -1 $dev
$ yes | $sudo mkswap -L "$label" "$dev"
#| total used free shared buff/cache available
#| Mem: 2041260 107724 1759992 6592 173544 1865772
#| Swap: 1020628 0 1020628
$ sudo swapoff -a
$ free
#| total used free shared buff/cache available
#| Mem: 2041260 106224 1761472 6592 173564 1867272
#| Swap: 0 0 0
$ sudo swapon "/dev/disk/by-label/swap"
$ free
#| total used free shared buff/cache available
#| Mem: 2041260 107912 1759716 6592 173632 1865584
#| Swap: 3943420 0 3943420
```

Instala docker para usar el otro disco USB:

```
$ sudo apt-get install docker.io time git lsb-
release file
#| Need to get 55.9 MB of archives.
#| After this operation, 255 MB of additional disk
space will be used.
#| Do you want to continue? [Y/n] Y
#| (...)
$ dev="/dev/disk/by-label/docker"
$ mnt="/var/lib/docker"
$ df -h "$mnt"
# /dev/mmcblk1p1 7.2G 1.2G 6.0G 17% /
$ sudo systemctl stop docker
$ sudo sync
$ sudo mkdir -p "$mnt"
$ sudo mount "$dev" "$mnt"
$ df -h "$mnt" # /dev/sda 3.7G 16M 3.5G 1%
/var/lib/docker
$ sudo systemctl restart docker
$ sudo docker version # 18.09.7
```

Compilar binarios

Hay varias formas de usar la plataforma Mozilla http://odroidxu4.local:8080/. La página WebThings. La más simple sería usar el paquete deb debería aparecer de la siguiente forma:

creado para Raspbian, pero la versión ARMv6 no está optimizada para nuestra CPU ARMv7. Así que intentemos compilarlo nuevamente en el dispositivo usando Docker para asegurarnos de que todo el proceso sea replicable. Es para la última versión 0.11.0, así que he publicado el paquete "webthingsgateway_0.11.0-1_armhf-debian-buster.deb" en https://bintray.com/rzr/deb/webthings-gateway#files. Tienes total libertad para instalarlo o recompilarlo en el dispositivo utilizando los siguientes pasos:

```
$ sudo apt-get install docker.io time git lsb-
release file
$ sudo apt-get install time screen
$ url=https://github.com/mozilla-iot/gateway-
deb.git
$ git clone --depth 1 --recursive "${url}"
$ cd gateway-deb
$ sudo time bash ./build-docker.sh
$ sudo docker image ls
#| REPOSITORY TAG IMAGE ID CREATED SIZE
#| c0edf15b50a7 About an hour ago 122MB
#| gatewaydeb default latest 32e5bdf8321c 8 hours
ago 843MB
#| 51a23a2e7130 9 hours ago 1.04GB
#| Debian 10 3eee7456d779 3 weeks ago 92.8MB
$ du -hsc ./dist/*
# | 21M ./dist/WebThings-gateway 0.11.0-1 armhf-
debian-buster.deb
$ sudo chmod -Rv 700 ./dist/WebThings-
gateway *.deb
$ sudo apt install -y ./dist/WebThings-
gateway *.deb
#| Need to get 0 B/40.5 MB of archives.
# | After this operation, 180 MB of additional disk
space will be used.
#| Do you want to continue? [Y/n] Y
#| (...)
$ sudo reboot
```

Webapp

Ahora la parte más difícil está hecha (¡no más líneas de comando!) Usamos el usuario y conectemos a la placa apuntando yu navegador a http://odroidxu4.local:8080/. La página de bienvenida debería aparecer de la siguiente forma:

```
| Mozilla IoT
| Welcome
| Choose a secure web address for your gateway:
| [subdomain].mozilla-iot.org
| [Email]
| [Create]
| [Skip]
```

La puerta de enlace se puede registrar en mozilla.org para la administración remota, esto es opcional, así que omite el subdominio ya que no usaremos la puerta de enlace de Internet para nuestro primer experimento. En la siguiente página te pedirá que crees unas credenciales (locales):

```
| Welcome
| Create your first user account:
| user: [user]
| email: [user@localhost]
| password: [password]
| password: [password]
| [Next]
```

Y ahora estamos listos para agregar algunas WebThings, y puede comenzar a llenar tu cuadro de mandos con recursos virtuales. Primero presiona el ícono de "menú hamburguesa" en la esquina superior izquierda, dirígete a la página de configuración y luego a la página de "complementos" en https://odroidxu4.local/settings/add-ons/ y habilita un adaptador " Virtual Things ":

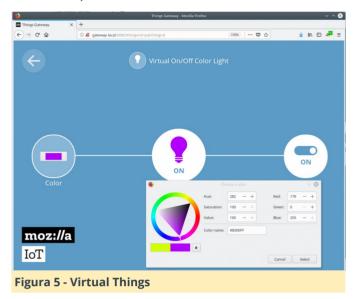
```
| Mozilla IoT Virtual Things Adapter
| by Mozilla IoT
```

Una vez habilitado, debe aparecer en la página de adaptadores en https:

//odroidxu4.local/settings/adapters. A continuación, puedes volver a la primera " Things page " (es la primera entrada en el menú). Podemos comenzar a agregar "cosas" presionando el botón inferior "+" en https: //odroidxu4.local/things, luego presione " Done " en la parte inferior:

```
| Virtual On/Off Color Light
| Color Light
| Save
```

Desde este punto, puede decidir controlar una lámpara virtual desde la interfaz de usuario e incluso establecer algunas reglas básicas (segunda entrada en el menú) con más recursos virtuales.



Únete a la comunidad

Acabamos de validar que nuestra configuración funciona con dispositivos simulados, por lo que el siguiente paso es buscar otros complementos. Actualmente hay más de 100, que es un gran escaparate de contribuciones de la comunidad. Un uso típico de un adaptador adicional es conectar un dispositivo que usa otro protocolo, y el complemento solo se traduce en abstracciones de WebThings. La puerta de enlace puede admitir dispositivos Zigbee, ZWave que encontrarás en el mercado. Dicho esto, algunos complementos podrían no funcionar perfectamente en nuestra configuración de ODROID-XU4, así que por favor infórma de problemas con proyectos relacionados con @rzr en GitHub y estaré encantado de compartir sugerencias o soluciones.

Va incluso más allá de eso. Cualquier cosa que tenga una API se puede administrar utilizando la plataforma WebThings. Como ejemplo, hice el adaptador Activity Pub que permite publicar un mensaje público para enviar a la red social cuando se cumplen algunas condiciones. La automatización es posible utilizando el motor de reglas donde los usuarios pueden vincular sensores a actuadores para establecer "comportamientos inteligentes". El control remoto es posible desde una aplicación web progresiva servida por la puerta de enlace. En mi opinión, esto es mucho más simple y confiable que cuando te ves obligado a instalar una aplicación de una tienda en teléfonos

personales para obtener acceso a los dispositivos que compras

El proyecto WebThings de Mozilla es una buena demostración de cómo un servicio podría tener el mismo aspecto para los usuarios, pero funciona de manera totalmente diferente a lo que encontrarás hoy en el mercado de loT. Tus comentarios son bienvenidos, ya que WebThings se puede mejorar.

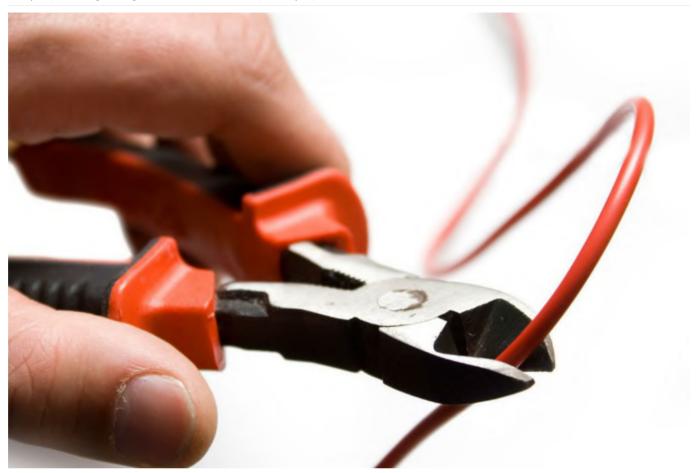
Recursos

- https://iot.mozilla.org/
- https://blog.mozilla.org/blog/2018/02/06/announcing
 -project-things-open-framework-connecting-devices-

web/

- https://fosdem.org/2020/schedule/speaker/philippe_ coval/
- https://www.hardkernel.com/shop/odroid-xu4special-price/
- https://www.armbian.com/
- https://edpb.europa.eu/our-work-tools/publicconsultations-art-704/2019/guidelines-42019-article-25-data-protection-design_en https://mastodon.social/@rzr/103805535349436510
- https://twitter.com/RzrFreeFr/status/1237784387346 989057

Modificando tu ODROID-GO Advance para la Carga Inalámbrica: Un simple Proyecto de Bricolaje



He añadido la posibilidad de cargar de forma inalámbrica mi ODROID-GO Advance usando un receptor de carga inalámbrico y una placa de conexión USB. Tuve los componentes por ahí, y literalmente solo me ha llevado un par de minutos y ha sido increíblemente simple. ¡Lo más difícil fue abrir la carcasa del ODROID-GO Advance sin romper los clips!

El receptor de carga inalámbrica se puede encontrar en los sitios habituales, como AliExpress, eBay o Amazon, aunque en Amazon es un poco más caro. La placa de conexión USB me costó un par de dólares de Adafruit, pero también se puede comprar en otros sitios.

Probablemente luego añadire un concentrador USB o carga USB C, pero honestamente, ahora tengo almohadillas de carga inalámbrica en toda la carcasa, así que no creo que alguna vez lo cargue de otra forma

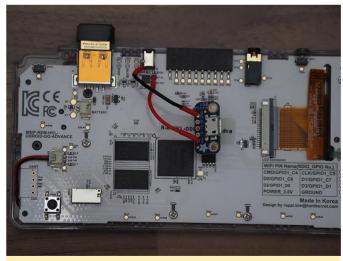


Figura 1: Primer plano de la soldadura necesaria para agregar carga inalámbrica al ODROID-GO Advance.

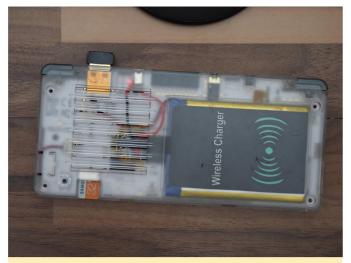


Figura 2: el cargador inalámbrico conectado dentro de la carcasa del ODROID-GO Advance.



Figura 3: el cargador inalámbrico ahora está cargando el ODROID-GO Advance.

Para comentarios, preguntas y sugerencias, visita el artículo original en

https://forum.odroid.com/viewtopic.php? f=193&t=38193.

Juegos Linux: GameStation Turbo Advance

② April 1, 2020 🌡 By Tobias Schaaf 🖹 Juegos, ODROID-GO Advance, ODROID-GO, Tutoriales



Aquí tienen el denominado ODROID GameStation Turbo Advance. Un sistema operativo de juegos repleto de funciones para ODROID-GO Advance que utiliza X11 como backend de gráficos y el modo atracción como interfaz para emuladores, juegos y herramientas.





Figura 2 - Juega a tus juegos favoritos de Amiga

Figura 1 - ODROID-GO Advance ejecutando GameStation Turbo Advance



Figura 3 - Juega a tus juegos favoritos de Amiga



Figura 4 - Juega a tus juegos favortios de Atari ST



Figura 5 - Juega a tus juegos favoritos de ScummVM



Figura 6 - Juega a tus juegos favoritos de Nintendo DS

El objetivo es tener una IU brillante y llamativa, así como una manera fácil de integrar un nuevo software.

Está basado en **Debian Buster** de @ Meveric con muchas modificaciones para que funcione como una imagen de juegos. Ten en cuenta que esta imagen e un "trabajo en progreso", ya que todavía hay algunas cosas que necesito mejorar, pero está en un punto en el que se puede compartir con "la gente".

Emuladores actualmente admitidos

- Amstrad CPC
- Atari 2600
- Atari 5200
- Atari 7800
- Atari 8bit
- Atari Lynx
- Atari ST
- Bandai Wonderswan
- Bandai Wonderswan Color
- Capcom Play System
- Capcom Play System I
- Capcom Play System II
- Commodore 64
- Commodore Amiga
- Colecovision
- GCE Vectex
- Magnavox Odyssey 2
- Mattel Intellivision
- Microsoft MSX
- Microsoft MSX2
- NEC SuperGrafx
- NEC Turbografx-16

- NEC Turbografx-CD
- Nintendo 64
- Nintendo DS
- Nintendo Entertainment System
- Nintendo Famicom Disk System
- Nintendo Gameboy
- Nintendo Gameboy Color
- Nintendo Gameboy Advanced
- Nintendo Virtual Boy
- Final Burn Alpha
- Sammy Atomiswave
- ScummVM
- Sega 32x
- Sega CD
- Sega Dreamcast
- Sega Gamegear
- Sega Genesis/Megadrive
- Sega Master System
- Sega Naomi
- Sega SG-1000
- Sharp X68000
- Sinclair ZX 81
- Sinclair ZX Spectrum
- SNK Neo Geo AES
- SNK Neo Geo CD
- SNK Neo Geo Pocket
- SNK Neo Geo Pocket Color
- Sony Playstation
- Sony Playstation Portable
- Sony Playstation Portable Minis
- Super Nintendo Entertainment System

La ruta de la BIOS por defecto es /home/odroid/ROMS, debes colocar tus archivos de BIOS en esta carpeta (si fuese de otra manera, consulta las notas disponibles en el enlace al final del artículo). También es la ruta donde debe colocar tus archivos ROM, en las subcarpetas respectivas. Está disponible para descargar

en https://oph.mdrjr.net/meveric/images/OG . GO2.img.xz md5 sha512 sig.

La imagen utiliza drivers de 32 bits y de 64 bits, lo que te permite ejecutar muchas aplicaciones ya existentes para armhf, incluso si aún no están exportadas a ARM64.

Cómo empezar

Copia tus archivos rom dentro de las carpetas de /home/odroid/ROMS/ que corresponden con el sistema para el que son las roms (por ejemplo, los juegos de Super Nintendo Entertainment System va dentro de /home/odroid/ROMS/SNES/). Cuando inicies la imagen, dirígete a "Attract Mode Setup", tienes dos opciones

- Escanear juegos y artwork
- Escanear sólo juegos

Para las artwork, necesita una conexión a Internet, de modo que debes tener un adaptador USB a LAN instalado y funcionando correctamente. Debería funcionar directamente sin configurar nada. "Al escanear solo los juegos importarán tus juegos sin descargar ninguna artwork. Descargar artwork puede llevar mucho tiempo (y necesitar espacio en disco) dependiendo de tu librería. No se recomienda escanear miles de juegos a la vez, a menos que disponga de varias horas libres.

Verás una barra de progreso mientras se importan los juegos y las artworks. Esta barra de progreso solo muestra el progreso general de todos los emuladores, no para cada ROM individual, lo que significa que dependiendo de cuántos juegos importes, el porcentaje del proceso no aumentará hasta que haya terminado con el sistema que este importando.

Actualizar regularmente:

Puesto que constantemente hay nuevos desarrollos para esta imagen y ODROID en general, se recomienda instalar todas las actualizaciones:

\$ apt-get update && apt-get upgrade && apt-get
dist-upgrade && apt-get autoremove

Esto garantizará disponer de los últimos parches y correcciones, así como nuevos emuladores cuando estén disponibles.

Configurar WLAN desde la línea de comandos

Edita el archivo /etc/network/interfaces o cree uno nuevo en /etc/network/interfaces.d/ y añade la siguiente línea:

auto wlan0
iface wlan0 inet dhcp
 wpa-essid
 wpa-psk

Menú principal

El menú principal funciona en Attract Mode y utiliza el siguiente esquema de botones:



Figura 7 - Esquema de botones

Aunque los botones funcionan en cada submenú, el resultado del botón que presiones y la carga actual de la batería solo se muestran en el menú principal. Attract Mode cuenta con un modo de protector de pantalla, que se activará tras aproximadamente 3 minutos de inactividad. Mostrará vistas previas de los videos e imágenes de la selección actual, esto significa que el menú principal mostrará videos de los diferentes sistemas, mientras que en el submenú de un sistema mostrará vistas previas de los diferentes juegos. Finalmente, Speaker Select permite solo auriculares, Auriculales + altavoz, solo Altavoz y OFF

Retroarch

Muchos de los emuladores funcionan con retroarch y utilizan el diseño de botones predeterminado de retroarch. Las teclas especiales se asignan de la siguiente forma::



Figura 8 - Asignación de Teclas especiales

Si desea hacer cambios en la configuración de retroarch, te aconsejo que salgas de retroarch si quieres guardar los cambios. Si vuelves al juego y luego sales del emulador, es posible que los cambios no se guarden.

Emulador Amiga - FS - UAE

Para Amiga, estoy usando FS-UAE puesto que su interfaz de usuario es mucho más amigable con el controlador que otros emuladores. También mapeé el movimiento y los botones del ratón, así como las teclas especiales para Amiga. El mapeo es el siguiente:



Figura 9 - Mapeo de las teclas en Amiga

El emulador está configurado actualmente para juegos compatibles con Amiga 500, por lo que tus juegos AGA o CD32 no funcionarán. Puesto que muchos juegos vienen con múltiples discos, puede comprimir todos los archivos adf en un solo archivo. Los discos se insertarán de DF0 a DF3, aunque puede

cambiar los discos a través del menú, incluso si un asignación de los botones siempre es un tanto juego solo es compatible con DF0 o tiene más de 4 polémica, así es como yo asigné los botones: discos.

El emulador utiliza carpetas independientes para los archivos Kickstart que se pueden encontrar en: /home/odroid/Documents/FS-UAE/Kickstarts v en: /home/odroid/.config/fs-uae/ puede encontrar los archivos de configuración para tus diferentes juegos. Luego, por ejemplo, puedes usar diferentes Kickstarts si quieres, o cambiar la configuración de la memoria, etc.

Atari ST - Hatari

Atari ST se ejecuta en el emulador independiente Hatari, no en un núcleo libretro. Tiene buen rendimiento y la asignación de botones es bastante mejor. El núcleo libretro solo funciona en muy pocos juegos. La distribución de los botones es la siguiente



Figura 10 - Mapeo de botones Atari ST

Algunos botones todavía están libres y pueden asignarse más adelante. Hay una forma de usar el menú para cambiar de disco, pero es bastante difícil. No es sencillo navegar por el menú, por lo tanto, sugiero usar juegos de un único disco o juegos que usen una imagen de disco duro.

Nintendo 64 - Mupen64plus

El emulador independiente Mupen64plus es bastante más rápido que las alternativas del núcleo libretro, de modo que, utilizo la versión independiente con el núcleo de video RICE, que es el núcleo de video más rápido en este momento, aunque no es perfecto. La



Figura 11 - Mapeo de botones en Nintendo 64

Este es el único emulador en el que debes presionar dos botones para guardar y cargar estados de guardado, por desgracia, no cuenta con suficientes botones. El rendimiento y la calidad de la emulación en general difiere bastante entre los juegos.

Nintendo DS - DraStic

DraStic de @Exophase es bastante rápido, pero es un emulador de Nintendo DS de código cerrado y, por lo tanto, es un poco complicado de ponerse a trabajar. El rendimiento es impresionante y puedes jugar a NDS muy bien en ODROID-GO Advance. La asignación de botones que tiene es la siguiente:



Figura 12 - Mapeo de botones en Nintendo DS

Desafortunadamente, la NDS DSi) (0 tiene demasiadas funciones para asignarlas todos, por lo que la emulación del ruido, abrir y cerrar la tapa no son compatibles, lo que hace que a algunos juegos no

sea posible jugar. Parece que de vez en cuando DraStic no reconoce ninguna entrada (aparte del puntero del ratón). No estoy seguro de por qué, no estoy seguro de cómo evitarlo. Descubrí que reiniciando el emulador parece resolver el tema. Simplemente presione el botón Salir del emulador y lo intente nuevamente. Asegúrate de haber realizado todas las actualizaciones del sistema antes de ejecutarlo

Playstation Portable - PPSSPPSDL

PPSSPPSDL es un emulador muy conocido para PSP y funciona bastante bien en ODROID-GO Advance. No es tan bueno como en otros dispositivos, como ODROID-XU4 u ODROID-N2, pero sigue siendo lo suficientemente bueno para la mayoría de los juegos. El esquema de botones es el siguiente:



Figura 13 - Mapeo de botones PSP

ScummVM Hints

El formato de los juegos ScummVM es un poco diferente del resto de emuladores. En la carpeta /home/odroid/ROMS/SCUMMVM/ debes crear una carpeta con el nombre del juego al que deseas jugar, Beneath Steel por ejemplo, Sky (/home/odroid/ROMS/SCUMMVM/Beneath а Steel Sky/). En esta carpeta, se copiarán todos los archivos de los juegos ScummVM que elijas. El siguiente paso es crear un archivo con el mismo nombre y la extensión .svm en la misma carpeta "Beneath a Steel Sky.svm" (/ home/odroid/ROMS/SCUMMVM/Beneath a Steel Sky/Beneath a Steel Sky.svm) Dentro del

archivo pones la identificación del juego. En nuestro ejemplo, esto sería "sky". Puede encontrar los ID del juego en la página de inicio de ScummVM en la columna "Game Short Name".

Sharp X68000 Hints

El emulador Sharp X68000 admite imágenes de disco único (.dim) e imágenes de disco múltiple .m3u. .m3u es una lista de "reproducción", que solo contiene la ruta/nombre de los archivos .dim. Si tiene juegos que requieren múltiples discos, debes colocar todos los discos dentro de una subcarpeta dentro de: /home/odroid/ROMS/X68000/ y crear un archivo .m3u que contenga los nombres de las imágenes del disco.

Por ejemplo: Crea una carpeta /home/odroid/ROMS/X68000/Gradius II

\$ mkdir -p "/home/odroid/ROMS/X68000/Gradius II"

Copie todos los archivos * .dim dentro de esta carpeta

/home/odroid/ROMS/X68000/Gradius II/Gradius II Gofer No Yabou (1992)(Konami)(Disk 1 of 2)(Disk A).dim

/home/odroid/ROMS/X68000/Gradius II/Gradius II Gofer No Yabou (1992)(Konami)(Disk 2 of 2)(Disk B).dim

Crea un archivo .m3u que contenga la ruta de los archivos ROMS:

\$ cat "/home/odroid/ROMS/X68000/Gradius II.m3u"
Gradius II/Gradius II Gofer No Yabou (1992)
(Konami)(Disk 1 of 2)(Disk A).dim
Gradius II/Gradius II Gofer No Yabou (1992)
(Konami)(Disk 2 of 2)(Disk B).dim

Sony Playstation

Al igual que Sharp X68000, puedes crear archivos y carpetas m3u para juegos con varios CD.. Puedes cambiar los CD desde el menú retroarch. Para obtener más información, consulta el artículo original, ubicado

en https://forum.odroid.com/viewtopic.php? f=193&t=38177.