Debian, Tizen, Lakka y Android Marshmallow para XU4 • Caja XU4

ODROID Magazine

¡Nuestro 3er año!

Año Tres
Num.#25
Ene 2016

Para jugadores serios

Hardkernel presenta su nuevo Universal Motion Joypad

JUEGOS LINUX

- Compilación de Kernel
- Control de la CPU del XU4
- Instalador Universal ODROID

Qué defendemos.

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

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

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

Simple, moderno y único.
De modo que tienes a tu alcance lo mejor

HARDKERNEL

Pollin ELECTRONIC

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

Dirección: Max-Pollin-Straße 1
85104 Pförring Alemania

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

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

EDITORIAL

ODROID Magazine celebra el inicio de su tercer año. Todos nuestros colaboradores nos sorprenden constantemente con sus innovaciones. Cada mes, nuestros columnistas regulares Tobías, Venkat y Nanik escriben fantásticos tutoriales y estudios muy completos que ayudan a nuestros lectores a encontrar nuevas formas de trabajar con sus ODROIDs disfrutando de ellos cada día. Nos hace mucha ilusión continuar elaborando la mejor revista que podemos. Espero que la comunidad ODROID en todo el mundo continúe enviándonos grandes y fantásticos artículos, ya que eso es lo que realmente hace que nuestra publicación prospere.

Este mes, presentamos el nuevo Universal Motion Joypad para que el manejo de los juegos sea más divertido. Tobias presenta un serie de juegos de estrategia que seguro te mantendrán entretenido, y destacamos algunos sistemas operativos innovadores como Tizen, Lakka, Debian Jessie Server y Android Marshmallow. También mostramos una moderna caja de aluminio para el ODROID-XU4, te enseñamos cómo regular tu CPU para reducir el ruido del ventilador, detallamos los pasos para compilar un kernel y presentamos una forma muy cómoda de instalar un nuevo sistema operativo de arranque dual o simple. No te olvides de visitar nuestro nuevo sitio web en http://magazine.odroid.com, ahora con un listado de artículos en el que se puede realizar búsquedas.

ODROID Magazine, que se publica mensualmente en http://magazine.odroid.com/, es la fuente de todas las cosas ODROIDianas. • Hard Kernel, Ltd. • 704 Anyang K-Center, Gwanyang, Dongan, Anyang, Gyeonggi, South Korea, 431-815 • fabricantes de la familia ODROID de placas de desarrollo quad-core y la primera arquitectura ARM "big.LITTLE" del mundo basada en una única placa.
Para información sobre cómo enviar articulos, contacta con odroidmagazine@gmail.com, o visita http://bit.ly/1yplmXs.
Únete a la comunidad ODROID con miembros en más de 135 países en http://forum.odroid.com/ y explora las nuevas tecnologías que te ofrece Hardkernel en http://www.hardkernel.com/.

HARDKERNEL

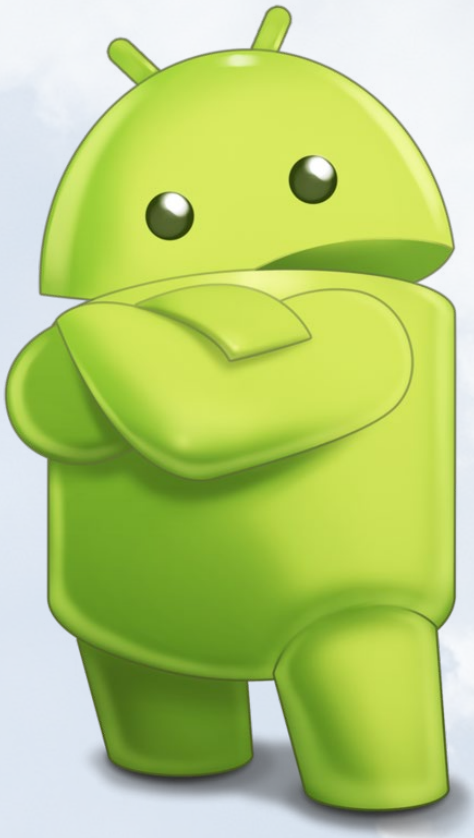
ODROID Magazine

Rob Roy, Editor Jefe

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

Manuel Adamuz, Editor Español

Tengo 31 años y vivo en Sevilla, España, aunque nací en Granada. Estoy casado con una mujer maravillosa y tengo un hijo. Hace unos años trabajé como técnico informático y programador, pero mi trabajo actual está relacionado con la gestión de calidad y las tecnologías de la información: ISO 9001, ISO 27001, ISO 20000 Soy un apasionado de la informática, especialmente de los microordenadores como el ODROID, Raspberry Pi, etc. Me encanta experimentar con estos equipos y traducir ODROID Magazine. Mi esposa dice que estoy loco porque sólo pienso en ODROID. Mi otra afición es la bicicleta de montaña, a veces participo en competiciones semiprofesionales.



Bruno Doiche, Editor Artístico Senior

Bruno pasó un par de meses “esnifando” redes y trabajando como loco para conseguir que todas nuestras publicaciones estén listas. Como de costumbre era una locura. ¡Pero una locura muy divertida!

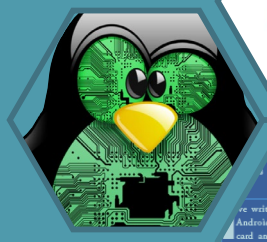
Nicole Scott, Editor Artístico

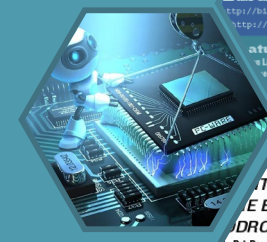
Soy una experta en Producción Transmedia y Estrategia Digital especializa en la optimización online y estrategias de marketing, administración de medios sociales y producción multimedia impresa, web, vídeo y cine. Gestiono múltiples cuentas con agencias y productores de cine, desde Analytics y Adwords a la edición de vídeo y maquetación DVD. Tengo un ODROID-U3 que utilizo para ejecutar un servidor web sandbox. Vivo en el área de la Bahía de California, y disfruta haciendo senderismo, acampada y tocando música. Visita mi web enhttp://www.nicolecscott.com.

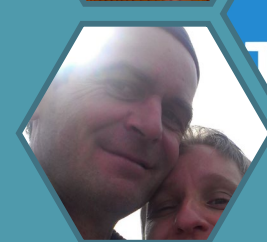
James LeFevour, Editor Artístico

Soy un especialista en medios digitales que disfruta trabajando como freelance en marketing de redes sociales y administración de sitios web. Cuanto más aprendo sobre las posibilidades de ODROID más me ilusiona probar cosas nuevas con él. Me traslade a San Diego desde el Medio Oeste de los EE.UU. Todavía estoy bastante enamorado de muchos aspectos que la mayoría de la gente de la Costa Oeste ya da por sentado. Vivo con mi encantadora esposa y nuestro adorable conejo mascota; el cual mantiene mis libros y material informático en constante peligro.

INDEX



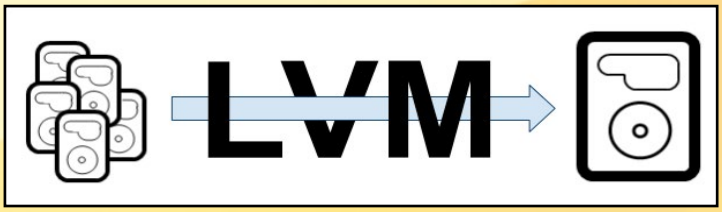




VOLUMENES LOGICOS CON POCO SUMISTRO

LA CAPACIDAD DE ASIGNAR DINAMICAMENTE ESPACIO

por David Gabriel



Cuando empiezas a tener múltiples aplicaciones ejecutándose en un único sistema, administrar el espacio asignado a cada una de ellas empieza a ser un poco complicado. Normalmente ajustas el tamaño para que sea suficiente y puedan ejecutarse durante algún tiempo, además cuentan con un generoso buffer para evitar quedarte sin almacenamiento cuando lo necesitan. Sin embargo, Las necesidades de algunos sistemas de archivos tienden a aumentar rápidamente con el tiempo, como son las bases de datos, mientras que otros lo hacen más despacio, como los servidores web. Esto da lugar a tener grandes cantidades de espacio asignado cuando realmente no siempre se está utilizando, aunque esté vinculado a un volumen en concreto. Este tipo de modelo se conoce como "Alta disponibilidad" y puede ser ineficaz en ciertas ocasiones.

Para casos como este, podemos aprovecharnos de otra gran característica que el LVM ofrece denominada "Disponibilidad dinámica". Ofrece la posibilidad de asignar dinámicamente espacio a los volúmenes según sea necesario. Normalmente, los bloques se escriben tan pronto como se crea el volumen lógico. En un volumen lógico con disponibilidad dinámica, los bloques se asignan como se escriben. De este modo, podemos tener un VL con un tamaño virtual que es mucho más grande que el almacenamiento físico disponible. Luego se puede aumentar a medida que sea necesario.

La "Disponibilidad dinámica" se puede comparar con un banco. El banco almacena el dinero de todos sus depositantes, pero todo ese dinero no existe físicamente a la vez. Si todas las persona decidieran retirar su dinero al mismo tiempo, el banco no tendría suficiente disponible para todos. Tendría que conseguir efectivo a partir de sus inversiones y préstamos, o mediante la venta de acciones. Otro ejemplo es una compañía de seguros. Existe una cierta cantidad de personas aseguradas, pero no todas ellas reclaman daños todos los años. De modo que la compañía no tiene que tener dinero para cubrir a todos los asegurados al mismo tiempo. Esto concepto también se puede aplicar a otros recursos como cuando se demanda más memoria o CPU, especialmente si se utilizan máquinas virtuales, aunque nosotros nos centraremos en el uso del espacio.

Para crear un volumen de disponibilidad dinámica, primero se debe crear un VL de fondo común, que procede de la combinación de dos VLs. Un VL para grandes datos que almacenará los bloques para el VL y otro VL que almacenará los metadatos. El VL de los metadatos mantiene un registro de qué bloque pertenece a qué VL dentro del fondo común, junto con alguna otra información.

Para crear el fondo común, usa los siguientes comandos:

```
$ lvcreate -n thinpool -L 10G rootvg
$ lvcreate -n metathinlv -L 500M rootvg
$ lvconvert --thinpool rootvg/thinpool \
  --poolmetadata rootvg/thinmetalv
```

Los primeros dos comandos crean los VLs de datos y metadatos, y el último es para fusionarlos y convertirlos en un VL de fondo común. Después, puedes crear los volúmenes lógicos que desees dentro del fondo común:

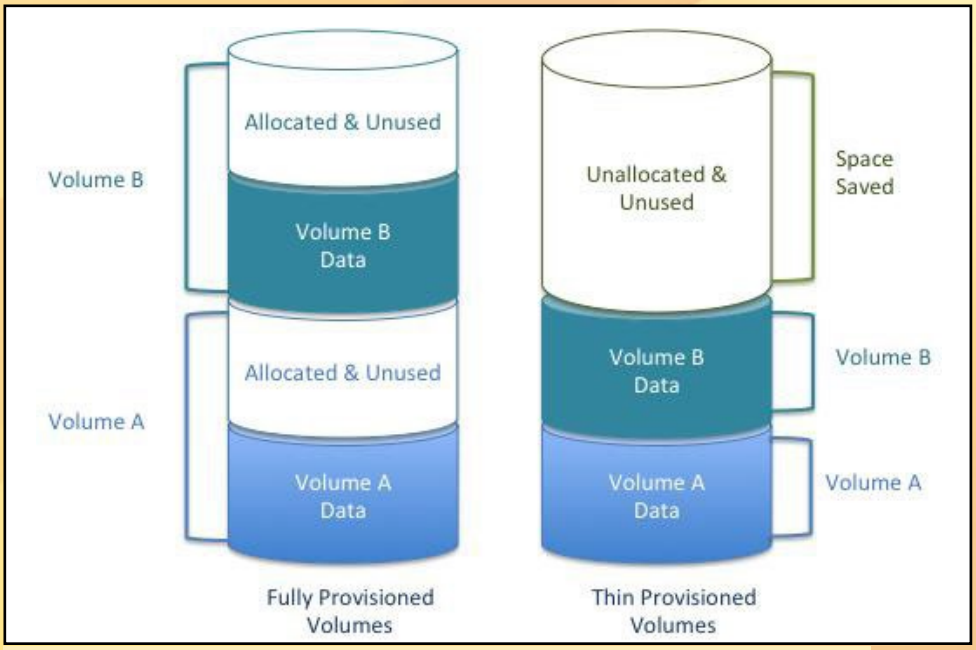
```
$ lvcreate -n mysqllv -V 1T \
  --thinpool rootvg/thinpool
```

Puedes comprobar tu VL recién creado con el comando lvs. Esto creará un nuevo VL a partir del fondo común con 1TB. Fíjate que claramente no tenemos 1 TB de espacio disponible, ya que el fondo común sólo tiene 10 GB, pero el sistema nos permitirá crear ese espacio virtual porque los bloques se irán asignando según sea necesario. Si no quieres preocuparte por la parte de los metadatos puede crear el volumen de fondo común en un solo paso:

```
$ lvcreate --thinpool thinpool -L 10G rootvg
```

O incluso crear el fondo común y un VL dentro:

```
$ lvcreate -L 10G -V 1T -n mysqllv \
  --thinpool rootvgvg/thinpool
```

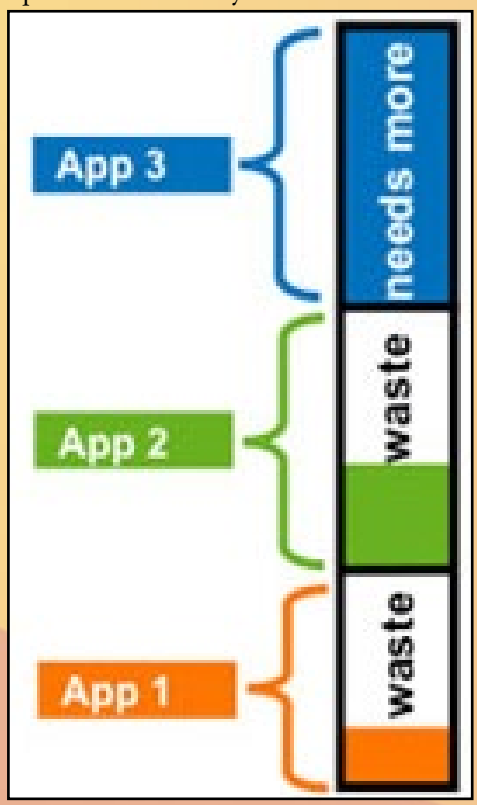


Una rápida comparativa entre "Alta Disponibilidad" y "Disponibilidad Dinámica"

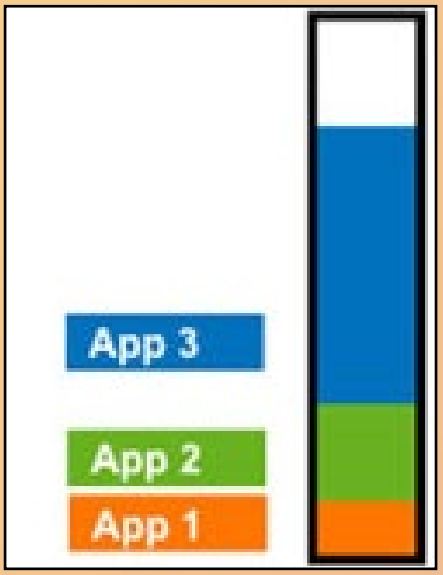
Ya deberías entenderlo, aunque tines que controlar muy de cerca el uso de espacio en los sistemas de archivos de disponibilidad dinámica. Si es posible, fija alertas que se disparen cuando la asignación de espacio aumente demasiado para evitar la sobre-asignación, pudiendo dañar los datos. Una vez que el uso empiece a crecer, debes ampliar el VL de fondo común tal como lo harías con un VL estándar. El sistema intentará escribir en un bloque que no existe, de modo que debe estar muy atento a la hora de utilizar la "Disponibilidad dinámica".

La asignación de almacenamiento desde el fondo común puede hacer que los VLs se fragmenten. Generalmente los VLs estándar evitan este problema asignando un único bloque completo en el disco. A menos que tengan un fondo realmente grande, esto no debería afectar al rendimiento del disco con el tiempo.

La disponibilidad dinámica puede ser muy útil si se usa adecuadamente. Evita tener espacio asignado si realmente no lo utilizas, de modo que los bloques irán a donde más se necesite. No obstante, si el administrador no está pendiente del sistema, podría convertirse en un verdadero quebradero de cabeza.



Puede hacer más con menos y ser flexible



DEBIAN JESSIE PARA ODROID-XU4

UNA DIMINUTA IMAGEN DE SERVIDOR

por Tobias Schaaf

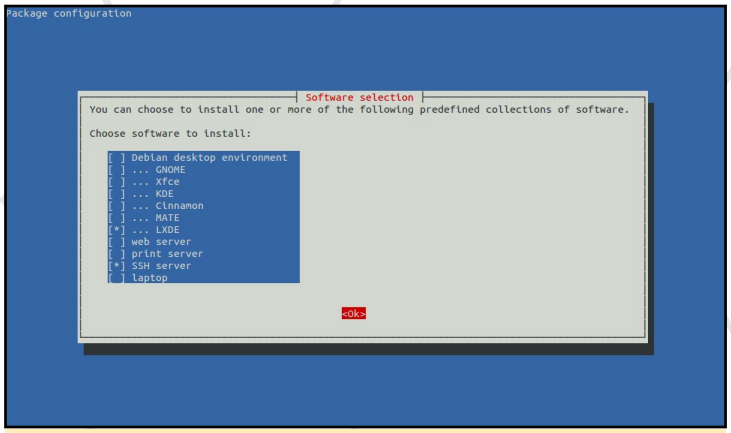
Recientemente he liberado una pequeña imagen Debian Jessie que se puede utilizar como PC de escritorio o como servidor, totalmente configurable. La imagen es un servidor básico con sólo el usuario root. Sin embargo, tiene todos mis repositorios ya añadidos, lo cual permite instalar y actualizar fácilmente paquetes como un kernel diferente, Kodi, Chromium o cuantos paquetes se quiera. El archivo de imagen está disponible en `http://bit.ly/1mf2CcD`, y es compatible con tarjetas microSD y módulos eMMC. Su tamaño total es de 99MB comprimido y 472MB descomprimido.

En el primer arranque, la imagen redimensionará la partición del sistema de archivos raíz y configura SSH, luego se reiniciará automáticamente tras completar la configuración inicial. Es en ese momento cuando la imagen está lista para usarse. El kernel y los principales archivos ya están instalados, si necesitas compilar tus propios drivers. Algunas herramientas básicas como ntp, htop, mc, vim, y bash también están disponibles por comodidad.

Añadir un escritorio

Puesto que todos mis repositorios están disponibles con paquetes pre-compilados para Debian Jessie, es fácil convertir esta imagen en una imagen de escritorio siguiendo estos pasos. En primer lugar, actualizar la lista de paquetes:

```
# apt-get update
```



Tasksel permite instalar con facilidad muchos paquetes de software

A continuación, ejecuta tasksel para elegir el entorno de escritorio deseado, tal y como se muestra en la Figura anterior:

```
# tasksel
```

Ten en cuenta que no todos los entornos de escritorio de Debian funcionan a la perfección en el ODROID. Las mejores opciones son LXDE o MATE, pero XFCE o KDE deberían funcionar también. Tasksel le llevará un tiempo descargar e instalar todos los paquetes necesarios para una imagen de escritorio. Además necesita al menos 1 GB de espacio adicional en disco, aunque se recomienda 2 GB.

Una vez que el entorno de escritorio se haya instalado, tenemos que instalar los driver X11 framebuffer:

```
# apt-get install xf86-video-armsoc-odroid
```

Los drivers de la GPU Mali para la aceleración 3D también se pueden instalar:

```
# apt-get install malit628-odroid
```

Necesitarás el xorg.conf apropiado para los drivers framebuffer:

```
# cd /etc/X11
```

```
# wget \
  http://oph.mdrjr.net/meveric/other/xorg.conf
```

Si eres un usuario más experimentado, siempre puedes instalar los paquetes de forma manual, manteniendo así la imagen tan pequeña como desees en lugar de usar tasksel. También es recomendable crear una nueva cuenta de usuario para el entorno de escritorio en lugar de iniciar sesión como root:

```
# adduser odroid
```

Una vez que se hayan completado todos los pasos de instalación, reinicia el ODROID. Tras unos minutos, deberías ver la pantalla gráfica de inicio de sesión de tu imagen Debian Jessie

Ahora que tienes tu escritorio funcionando, puede instalar fácilmente todos los paquetes disponibles en mi repositorio como Kodi o XBMC.

Para instalar kodi escribe en una ventana de terminal:

```
# apt-get install kodi-odroid
```

Para instalar XBMC en su lugar, escribe lo siguiente en una ventana de terminal:

```
# apt-get install xbmc-odroid
```

A continuación, instala el firmware necesario para la descodificación por hardware en XBMC y kodi

```
# apt-get install firmware-samsung
```

El paquete TVHeadend te permite ver la televisión en directo utilizando Kodi:

```
# apt-get install tvheadend
```

También puedes instalar una versión optimizada del navegador Chromium desde mi repositorio:

```
# apt-get install chromium-browser-odroid
```

El IDE de Arduino es necesario para programar con la placa ODUINO, ODROID SHOW, y otros dispositivos electrónicos basados en Arduino:

```
# apt-get install arduino
```

También recomiendo instalar ffmpeg desde el repositorio de Debian, que es una muy buena herramienta para la visualización y conversión de videos, está mantenido por desarrolladores experimentados de Debian.

Notas

La mayoría de los paquetes disponibles en mi repositorio están basados en X11. Por ejemplo, el paquete malit628-odroid son los driver X11 para la GPU y Kodi sólo está disponible para X11, de modo que necesitas un entorno de escritorio para utilizar cualquiera de estos paquetes. Además, yo no he instalado todos los drivers y firmwares disponibles. Si deseas utilizar el módulo 4 Wifi, tendrás que instalar el paquete firmware-ralink disponible en el repositorio estándar de Debian.

Cambiar el idioma

Si has instalado un entorno de escritorio, ya deberías tener una configuración de teclado instalada. Debian te pregunta en la primera instalación por la distribución del teclado a utilizar en tu escritorio X11. Para la consola, también necesitas instalar consoleSetup:

```
# apt-get install console-setup keyboard-configuration
# dpkg-reconfigure keyboard-configuration
```

Es probable que quieras ajustar también la zona horaria:

```
# dpkg-reconfigure tzdata
```

Soporte para CEC

Para tener soporte CEC, instala el paquete libcec:

```
# apt-get install libcec
```

Si fuera necesario, también puedes instalar las herramientas de cec escribiendo:

```
# apt-get install cec-utils
```

Después de esto, necesitas añadir una nueva regla udev para que puedas acceder al dispositivo CEC:

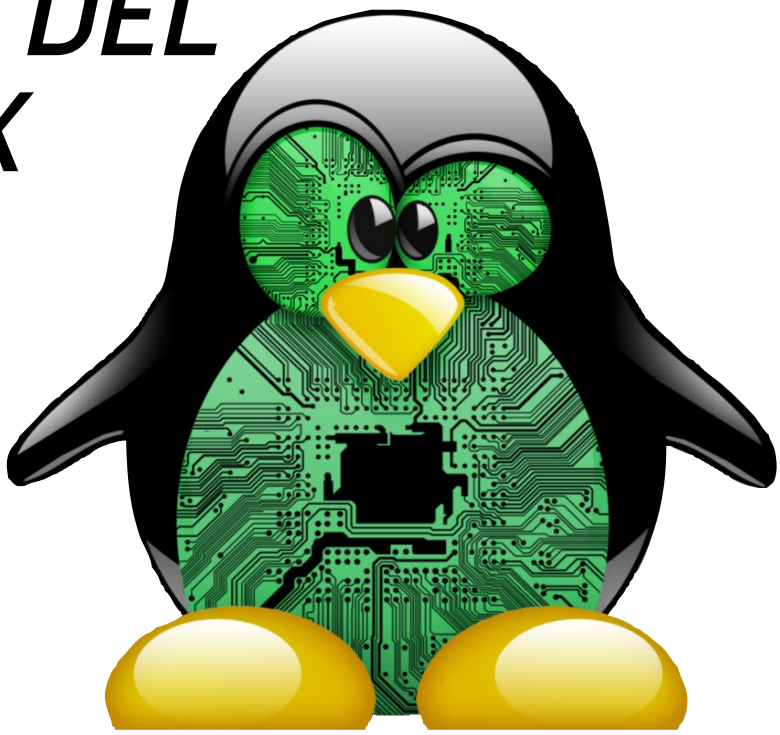
```
# echo 'KERNEL=="CEC",SUBSYSTEM=="misc",MODE="0666"'\
  > \ /etc/udev/rules.d/20-hkl_cec.rules
```

Si encuentras algún error por favor avísame y si cuentas con otro modelo además del ODROID-XU3 o XU4, puedo convertir fácilmente esta imagen para que se pueda usar con el ODROID-X, X2, U2, U3, o C1. Si tiene comentarios, preguntas o sugerencias, por favor visita el hilo original en `http://bit.ly/1k5qldS`.

COMPILACION DEL KERNEL LINUX

COMO PERSONALIZAR TU SISTEMA OPERATIVO

por Uli Middelberg



¿Quién no querría controlar totalmente su propio kernel? ¡Por supuesto, nosotros lo hacemos!

Este tutorial cubre algunos aspectos de la compilación de tu propio kernel Linux para tu dispositivo ARM. La mayoría de las distribuciones de Linux para plataformas de PC x86 mantienen un kernel Linux que soporta una amplia gama de dispositivos hardware, por lo que es muy poco probable que tengas que compilar tu propio kernel desde el código fuente si utilizas dispositivos x86. Sin embargo, en las plataformas ARM, el kernel Linux es proporcionado por el fabricante del system on chip (SoC) o de la placa de desarrollo. En muchos casos, estos kernel sólo incluyen funciones mínimas y los drivers del dispositivo, por lo que tienen que modificarse para incluir más funcionalidades.

Además, es posible que quieras incluir funciones específicas que sólo se proporcionan a modo de parche para el código fuente del kernel, como la optimización de la seguridad (bit.ly/1wcJIa3) o la capacidad de trabajar en tiempo real (bit.ly/1OQlDcv). Algunos usos imponen requisitos especiales. Por ejemplo, es posible que prefieras apagar el módulo del kernel que da soporte a entornos de alta seguridad y compilar un kernel monolítico en su lugar. O bien, puede que necesites limitar los recursos disponibles para compilar una imagen del kernel muy pequeña.

Los recientes Dispositivos ARM se han vuelto muy potentes, así que prefiero compilar el kernel directamente en el propio dispositivo en lugar de usar la compilación cruzada, con el fin de reducir el nivel de complejidad.

Componentes del kernel

El kenel de Linux consta de los siguientes componentes:

- **Imagen del Kernel <particion-arranque>/zImage o <particion-arranque>/uImage, según configuración y capacidades de U-Boot**
- **Binario en árbol del dispositivo, una descripción del dispositivo de bajo nivel, específica para el dispositivo <particion-arranque>/<placa> .dtb**
- **Módulos del Kernel /lib/modules/<version-kernel>/ ***
- **Firmware del dispositivo /lib/firmware/ ***

Estos componentes se compilan a partir de las fuentes del kernel con la ayuda de la utilidad make. Por lo general, la imagen del kernel y el binario en árbol del dispositivos se cargan desde una pequeña partición de arranque vfat, que está montada como /boot o /media/boot. El resto reside en el sistema de archivos raíz.

Preparar entorno de trabajo

A parte de la utilidad make, se necesitan otras utilidades para compilar el kernel de Linux. Por ejemplo, con Ubuntu, tendrás que instalar los siguientes paquetes:

```
$ sudo apt-get -y install \
bc curl gcc git libncurses5-dev \
lzop make u-boot-tools
```

Compilador por defecto

En el momento de la publicación de este artículo, la mayoría de las distribuciones de Linux actualizaron su compilador por defecto a gcc versión 5. Si tu distribución sigue utilizando la versión 4.8 por defecto, deberías considerar la posibilidad de cambar a la versión 5. Ubuntu 14.04, por ejemplo, utiliza la 4,8 . Para actualizar a la versión 5, escribe los siguientes comandos en una ventana de terminal:

```
$ sudo apt-get -y install \
  python-software-properties;
$ sudo add-apt-repository -y \
  ppa:ubuntu-toolchain-r/test;
```

```
$ sudo apt-get update
$ sudo apt-get -y install gcc-5 g++-5
$ sudo update-alternatives --install \
  /usr/bin/gcc gcc /usr/bin/gcc-5 50
$ sudo update-alternatives --install \
  /usr/bin/g++ g++ /usr/bin/g++-5 50
```

El comando update-alternatives te ayuda a definir el comando por defecto que se ejecutará si las diferentes versiones del mismo comando se instalan al mismo tiempo. Puede comprobar la versión de gcc escribiendo lo siguiente:

```
$ gcc --version
```

U-boot

Compilar un kernel personalizado siempre conlleva el riesgo de que el nuevo kernel no llegue a arrancar por diversas razones. Recomiendo definir una macro u-boot que cargue y arranque un kernel para pruebas antes de sobrescribir el kernel por defecto existente. Puede encontrar más información sobre u-boot en bit.ly/1LMcDHM.

Descargar las fuentes

La página web www.kernel.org proporciona las fuentes del Kernel Linux estándar:

```
$ curl -sSL https://cdn.kernel.org/pub/linux/\
  kernel/v4.x/linux-4.3.tar.xz | \
  unxz | tar -xvf -
```

Puedes obtener las versiones más recientes desde este sitio web, pero sólo unos cuantos dispositivos ARM serán capaces de arrancar un kernel estándar sin modificar. Incluso si llegan a iniciarse, es muy probable que algunos dispositivos presente problemas de compatibilidad debido a que faltan drivers, como son lo que dan soporte a la aceleración gráfica.

Para conseguir un kernel compatible con tu placa ARM, es necesario buscar las fuentes del kernel proporcionadas por el proveedor de la placa. A menudo, estas fuentes contienen parches adicionales para una versión del kernel con soporte a largo plazo. Muchos proveedores, como Hardkernel, utilizan Github para proporcionar y administrar sus archivos fuente del kernel específico, lo que hace que sea muy fácil añadir nuestras propias aportaciones. Por lo general, los proveedores organizan las fuentes del kernel por cada placa dentro de las divisiones Github designadas.

El comando git clone crea una copia local del repositorio:

```
$ git clone --depth 1 --single-branch \
  -b <branch> \
  <URL to the repository>
```

Esta copia local sólo contiene la división/rama especificada en <branch> sin información sobre los commits anteriores. El argumento "--depth 1" reduce el tamaño de la descarga limitando el árbol de commit al más reciente. Por ejemplo, si escribe lo siguiente, la última revisión del código fuente del kernel para el ODROID-C1 se guardará en un directorio llamado "linux":

```
$ git clone --depth 1 --single-branch \
-b odroidc-3.10.y \
https://github.com/hardkernel/linux
```

Compilar el kernel personalizado

Ahora, estás listo para empezar a compilar tu propio kernel. Tras descargar y extraer el código fuente del kernel, empieza creando un archivo de configuración llamado .config. Este archivo de texto incluirá parámetros importantes para tu kernel, una línea por cada opción o parámetro del kernel.

```
cd linux
make <default_config>
less .config
```

Puedes encontrar la configuración por defecto para tu dispositivo ARM en el directorio ./arch/arm/configs/. Estos son los archivos de configuración para las placas mencionadas:

Device	default configuration
ODROID C1(+)	odroidc_defconfig
ODROID U3	odroidu_defconfig
ODROID XU3	odroidxu3_defconfig
ODROID XU4	odroidxu4_defconfig

Una vez que el .config de configuración del kernel haya sido creado, puedes modificarlo, ya sea con un editor de texto, o bien usando el siguiente comando para cambiar la configuración del kernel de forma interactiva:

```
$ make menuconfig
```

Si se activa una opción de configuración, puedes leer la configuración existente de un kernel en ejecución:

```
$ cat /proc/config.gz | gunzip | less
```

Cuando hayas terminado con la configuración del kernel, crea una nueva configuración por defecto:

```
$ make savedefconfig
```

Este comando crea un archivo llamado defconfig desde el

archivo .config, que contiene sólo los cambios con respecto a los valores predeterminados de la configuración del kernel, reduciendo el tamaño del archivo aproximadamente en un 15% o 20% con respecto al archivo .config original:

El siguiente paso es compilar el binario en árbol del dispositivo, los módulos del kernel y la imagen del kenel, que puede ser un archivo zImage o uImage. Este es el paso que necesita más tiempo, incluso con la ejecución en paralelo utilizando la opción -j4, se necesita alrededor de una hora para compilar el kernel del C1 en el propio C1, y unos 20 minutos en el XU4. Para iniciar la compilación, escribe el siguiente comando:

```
$ make -j4 zImage uImage dtbs modules
```

Instalar el kernel personalizado

Como se ha mencionado anteriormente, es posible que quieras probar tu nuevo kernel antes de reemplazar el existente. Este paso requiere conocer un poco el u-boot y la configuración u-boot de tu placa en concreto. Es necesario definir una macro u-boot que arranque tu kernel personalizado en lugar del sistema por defecto. Cuando haya terminado con las pruebas, puedes instalar el nuevo kernel para que se convierta en el sistema por defecto con estos comandos:

```
$ sudo cp ./auch/arm/boot/[u|z]Image \
./auch/arm/boot/dts/*.dtb <boot-partition>
$ sudo make modules_install
$ sudo make firmware_install
```

La imagen del kernel y el binario en árbol del dispositivo se instalan en la partición de arranque, mientras que los módulos del kernel y el firmware del dispositivo se copian en el sistema de archivos raíz. Si está ejecutando diferentes instalaciones de Linux en diferentes particiones de tu dispositivo de almacenamiento SD o eMMC con la misma imagen del kernel, también es necesario instalar los módulos del kernel y firmware del dispositivo en cada una de las particiones con los siguientes comandos, repitiéndolos para cada partición:

```
$ sudo make modules_install INSTALL_MOD_PATH=<path>
$ sudo make firmware_install INSTALL_FW_PATH=<path>
```

Puede obtener una lista de todos los parámetros de make escribiendo lo siguiente:

```
$ make help
```

Ejemplos

ODROID-C1 and ODROID-C1+ bit.ly/1SgNPut

```
$ git clone --depth 1 --single-branch \
-b odroidc-3.10.y \
https://github.com/hardkernel/linux
$ cd linux
$ make odroidc_defconfig
$ make -j 4 uImage dtbs modules
$ sudo cp arch/arm/boot/uImage \
arch/arm/boot/dts/*.dtb /media/boot
$ sudo make modules_install
$ sudo make firmware_install
```

ODROID-C1 Mainline (Experimental) bit.ly/1ZuG5XK

```
$ curl -sSL \ https://cdn.kernel.org/pub/linux/\
kernel/v4.x/testing/linux-4.4-rc2.tar.xz | unxz | \
tar -xvf -
$ cd linux
$ make multi_v7_defconfig
$ make -j 4 LOADADDR=0x00208000 \
uImage dtbs modules
$ sudo cp arch/arm/boot/uImage \
arch/arm/boot/dts/*.dtb /media/boot
$ sudo make modules_install
$ sudo make firmware_install
```

ODROID-U3 bit.ly/1kl4Fue

```
$ git clone --depth 1 --single-branch \
-b odroid-3.8.y \
https://github.com/hardkernel/linux
$ cd linux
$ make odroidu_defconfig
$ make -j 4 zImage dtbs modules
$ sudo cp arch/arm/boot/zImage \
arch/arm/boot/dts/*.dtb /media/boot
$ sudo make modules_install
$ sudo make firmware_install
```

ODROID-XU3 bit.ly/1YIToBI

```
$ git clone --depth 1 --single-branch \
-b odroidxu3-3.10.y \
https://github.com/hardkernel/linux
$ cd linux
$ make odroidxu3_defconfig
$ make -j 8 zImage dtbs modules
$ sudo cp arch/arm/boot/zImage \
arch/arm/boot/dts/*.dtb /media/boot
$ sudo make modules_install
$ sudo make firmware_install
```

ODROID-XU4 bit.ly/1J9ZVn1

```
$ git clone --depth 1 --single-branch \
-b odroidxu4-v4.2 \
```

```
https://github.com/tobetter/linux
$ cd linux
$ make odroidxu4_defconfig
$ make -j 8 zImage dtbs modules
$ sudo cp arch/arm/boot/zImage \
arch/arm/boot/dts/*.dtb /media/boot
$ sudo make modules_install
$ sudo make firmware_install
```

Compilar un kernel de prueba

Copia el contenido de /media/boot a un nuevo directorio en la partición de inicio, como /media/boot/backup. Si el kernel de prueba tiene la misma versión que la actual, debe definir una extensión. Por ejemplo, añadiendo "-dev" con CONFIG_LOCALVERSION = "- dev" en el .config de configuración del kernel con el fin de evitar que los módulos del kernel actual se sobrescriban. Copia la imagen del kernel y el binario en árbol del dispositivo a /media/boot/test en lugar de /media/boot. Puedes retocar el archivo /media/boot/boot.ini y modificar la ruta para cargar el kernel y el binario en árbol del dispositivo. Aquí tienes un ejemplo de boot.ini para el ODROID-C1:

```
setenv prefix '/test/'
...
fatload mmc 0:1 0x21000000 ${prefix}uImage
fatload mmc 0:1 0x22000000 uInitrd
fatload mmc 0:1 0x21800000 ${prefix}meson8b_odroidc.
dtb
...
```

Si quieres volver a tu imagen backup del kernel, cambia el prefijo de la variable u-boot a /backup/. Si tienes acceso a u-boot a través de la consola serie, puede definir una macro u-boot, que cargue la imagen del kernel y el binario en árbol del dispositivo desde /media/boot/test. Por favor, consulta `bit.ly/1JBhhnQ` para más detalles. No olvides ejecutar el siguiente comando antes de compilar otro kernel:

```
$ make clean
```

Si tienes comentarios, preguntas o sugerencias, por favor visita el post original en `http://bit.ly/1NVRprY`

INSTALADOR DE IMAGENES UNIVERSAL

por @loboris

He escrito un nuevo instalador universal con el que puedes instalar Android, Linux o ambos como un sistema de arranque dual, desde la SD y/o una unidad USB. Puedes descargarlo en `http://bit.ly/1khGRrg`, el codigo fuente esta disponible en `http://bit.ly/1khGVHB`.

Características

Basado en initramfs Linux - Instalador interactivo El usuario puede configurar tamaño de las particiones deseadas y el destino de la instalación - Se puede instalar en una tarjeta SD o módulo eMMC - Usa el upgrade.zip estándar de Android y los archivos de imagen de Linux como fuentes - Las fuentes de instalación de Android (update.zip) se puede colocar en la tarjeta SD o en una memoria USB - Las fuentes de instalación de Linux se debe colocar en la unidad USB - Las fuentes de instalación deben ubicarse en la primera partición de la unidad USB - Compatible con instalaciones de arranque dual (Android y Linux) - Probado con todas las versiones de Android y Linux para XU3/XU4

Uso

Primero, descarga la imagen del instalador pre-compilada para la tarjeta SD, y usa MD5 para verificar su integridad. Descomprime el archivo xz. El universal_install_small.img es una imagen de 200 MB y el universal_install.img es una imagen de 2GB. Escribe la imagen en una tarjeta SD con el comando dd bajo Linux o con el software de grabación de imágenes para Windows. Después, puedes ampliar la partición FAT de la SD para ajustarla al tamaño de la tarjeta, pero ten cuidado de no cambiar el sector de inicio de la partición.

Después, copia las fuentes de instalación, que es el archivo update.zip para Android y el archivo .img para Linux, a la primera partición de tu unidad USB. Cambiar el nombre de la instalación de Linux a linux.img. Si sólo quieres instalar Android, puedes copiar update.zip directamente a la tarjeta SD sin necesidad de utilizar la unidad USB. Ajusta el interruptor de arranque del ODROID para arrancar desde la tarjeta SD, conecta el dispositivo USB, inserte la tarjeta SD y enciéndelo. Sigue las instrucciones para seleccionar los tamaños de las particiones deseadas y destino de la instalación (tarjeta SD o eMMC).

Si estás interesado en saber cómo funciona, descarga universal_install_source.tar.gz. Puedes crear una tarjeta SD de arranque con el instalador universal, o simplemente analizar los scripts de instalación para conocer mejor el proceso de arranque del ODROID y de initramfs. Las funciones de arranque dual son las mismas que las descritas en el artículo http://bit.ly/1j9r6TG.

Al instalar un único sistema operativo, tienes libertad para seleccionar el tamaño de las particiones, y puedes instalarlo en eMMC sin extraerlo. Por otro lado, el CM-12.1 Android 5.1.1 Lollipop y CM-12.1 Android TV 5.1.1 Lollipop no tienen imagen de instalación. Pronto añadiré más opciones, como la posibilidad de hacer backup y restaurar funciones. Si tiene preguntas, comentarios o sugerencias, visita el post original en `http://bit.ly/1PbDhMb`.

VENTILADOR Y CPU BAJO TU CONTROL

DOMINA LA CANTIDAD DE CALOR QUE GENERA EL XU3 Y XU4 CUANDO NO NECESITES TODO EL POTENCIAL DEL OCTA-CORE

por Adrian Popa

Si quieres mejorar el ruido del ventilador de tu ODROIDXU3 o ODROID-XU4, cuentas con algunas opciones. Puede cambiar el ventilador (http://bit.ly/1BeKEqw), o montar un impresionante disipador de calora a modo de caja (http://bit.ly/1T28KQ3). Sin embargo, si normalmente utilizas tu XU3/4 como servidor, hay otra forma de minimizar el ruido sin tener que cambiar el hardware.

Si actualizas la configuración del regulador del procesador y limitas la frecuencia máxima a unos 600 MHz, el ODROID raramente superará los 65 ºC, incluso con una alta carga de trabajo. Para cambiar la frecuencia y los reguladores, necesitas editar los archivos pseudo especiales dentro del directorio /proc. Para facilitarme el trabajo, he escrito un pequeño script en Perl con el que puedes ajustar y mostrar los valores actuales del llamado "odroid-cpu-control". Está disponible en http://bit.ly/1IUZ0qz, y el hilo de soporte en http://bit.ly/1RSrjb6.

El script puede mostrar los reguladores y las frecuencias actuales, y fijar otras nuevas, lo cual requiere privilegios de root. Por ejemplo, para mostrar los reguladores y frecuencias máximas y mínimas actuales, puedes escribir el siguiente comando.

```
$ odroid-cpu-control -l
```

```
adrianp@procyon:~$ odroid-cpu-control -l
CPU0: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [1.40GHz]
CPU1: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [1.40GHz]
CPU2: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [1.40GHz]
CPU3: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [1.40GHz]
CPU4: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [2.00GHz]
CPU5: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [2.00GHz]
CPU6: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [2.00GHz]
CPU7: governor powersave        current 200.00MHz       min 200.00MHz [200.00MHz]       max 600.00MHz [2.00GHz]
```

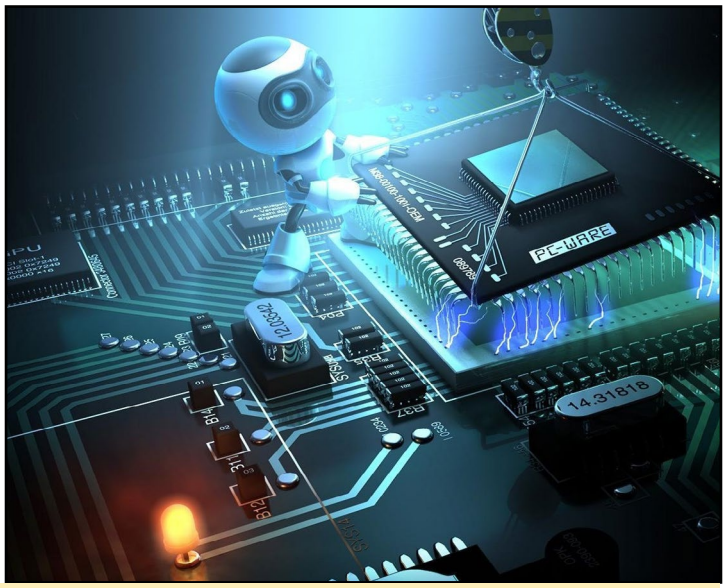
Listado de valores actuales

Puedes ser más específico a la hora de mostrar los datos especificando el/los núcleo(s) de la CPU y el/los parámetro(s) que deseas que aparezcan. Por ejemplo, para mostrar la frecuencia actual y la frecuencia máxima para los núcleos 1, 4, 5 y 6, puedes escribir lo siguiente, tal y como muestra la siguiente imagen:

```
$ odroid-cpu-control -l -f -M -c 1,4-6
```

```
adrianp@procyon:~$ odroid-cpu-control -l -f -M -c 1,4-6
CPU1: current 200.00MHz max 600.00MHz [1.40GHz]
CPU4: current 200.00MHz max 600.00MHz [2.00GHz]
CPU5: current 200.00MHz max 600.00MHz [2.00GHz]
CPU6: current 200.00MHz max 600.00MHz [2.00GHz]
```

Listado de núcleos seleccionados



A veces no se necesita usar todo el potencial que ofrece el ODROID

Las funciones de los parámetros se explican si ejecutas el script sin argumentos. En resumen, -l gestiona el listado, -f muestra la frecuencia actual, -m maneja la frecuencia mínima, -M maneja la frecuencia máxima, -g es el regulador y -c especifica los núcleos sobre los que deseas incidir. La sintaxis de los núcleos es flexible. Puede separarlos por una coma y sin espacios, o usar un guión para fijar rangos. Sin el argumento -c, todos los núcleos de la CPU serán seleccionados por defecto.

El script también ofrece una forma inteligente de no perder de vista los parámetros actuales, de manera similar a cómo funciona el comando "top". Si agregas -i 1 a la sintaxis de la lista, entonces, los datos se actualizarán cada segundo. Esto es útil sobre todo si se analiza la frecuencia actual de los núcleos cuando se realiza tareas que consumen mucha CPU.

Para realizar cambios en la configuración de la CPU, es obvio que necesitas ejecutar el script como usuario root usando sudo. La sintaxis cambia un poco: reemplazas -l por -s (para los ajustes). Las opciones -m, -M y -g utilizar parámetros adicionales. Para cambiar la frecuencia mínima y máxima, necesitas especificar la nueva frecuencia en MHz o GHz con el sufijo M o G. Los reguladores disponibles se muestran cuando se ejecuta " odroid-cpu-control -h ". Por ejemplo:

```
$ sudo odroid-cpu-control -s -m 300M -M 1.2G -g onde-
mand -c 0,4
```

```
adrianp@procyon:~$ sudo odroid-cpu-control -s -m 300M -M 1.2G -g ondemand -c 0,4
CPU0: governor powersave -> ondemand
CPU0: min 200.00MHz [200.00MHz] -> 300.00MHz [200.00MHz]
CPU0: max 600.00MHz [1.40GHz] -> 1.20GHz [1.40GHz]
CPU4: governor powersave -> ondemand
CPU4: min 200.00MHz [200.00MHz] -> 300.00MHz [200.00MHz]
CPU4: max 600.00MHz [2.00GHz] -> 1.20GHz [2.00GHz]
```

Configuración min/max/regulador

En el comando anterior, ajustamos la frecuencia mínima a 300 MHz y la frecuencia máxima a 1,2 GHz para los núcleos 0 y 4, mientras que cambiamos el regulador a OnDemand. Ten en cuenta que en las plataformas XU3 y XU4, la arquitectura big.LITTLE fija los mismos parámetros para los núcleos 0-3 y 4-7. Por ejemplo, si cambiamos algo en el núcleo 0 cambiará también lo mismo en los núcleos 1, 2 y 3. Esto significa que el comando en la práctica cambia la configuración en todos los núcleos. Si tus valores de frecuencia no son válidos, como que supean la frecuencia máxima que soporte ese núcleo, se redondearán al valor válido más cercano limitando la frecuencia máxima. Debido a este redondeo, el siguiente comando es perfectamente válido, aunque no hará overclock en tu CPU:

```
$ sudo odroid-cpu-control -s -m 10M -M 4.5G -c 0,4
```

```
adrianp@procyon:~$ sudo odroid-cpu-control -s -m 10M -M 4.5G -c 0,4
CPU0: min 200.00MHz [200.00MHz] -> 200.00MHz [200.00MHz]
CPU0: max 1.40GHz [1.40GHz] -> 1.40GHz [1.40GHz]
CPU4: min 200.00MHz [200.00MHz] -> 200.00MHz [200.00MHz]
CPU4: max 2.00GHz [2.00GHz] -> 2.00GHz [2.00GHz]
```

Usando un parámetro que supera los límites

Tal vez te preguntes cómo podría ayudarte el comando anterior. En primer lugar, necesitas averiguar qué regulador de CPU es el mejor para tu caso en concreto. En mi opinión, el regulador por defecto "performance" es un desperdicio, ya que mantiene la frecuencia alta y hace que la temperatura del sistema se eleve. Otros reguladores aumentan o disminuyen la frecuencia del núcleo basándose en la carga de trabajo usando diferentes algoritmos, con la opción de "ahorro de energía" que mantiene todas las frecuencias al mínimo.

Puedes combinar la configuración de la frecuencia mínima/máxima y del regulador para proporcionar a tu sistema el mejor equilibrio entre capacidad de respuesta y temperatura. Podría ser posible encontrar una combinación donde puedas utilizar la XU3/4 con Kodi sin escuchar el ventilador. Otras plataformas podrían beneficiarse de la reducción de la frecuencia de la CPU para bajar la temperatura, o elevar la duración de la batería si el sistema usa una.

Habrá momentos en los que una determinada configuración no sea la más óptima. En ese caso, puede utilizar la aplicación "cron" para configurar varios perfiles para diferentes momentos, o simplemente cambiar la configuración sobre la marcha utilizando las líneas de comandos anteriores. Por ejemplo, puede limitar la frecuencia máxima durante la noche y elevarla por la mañana.

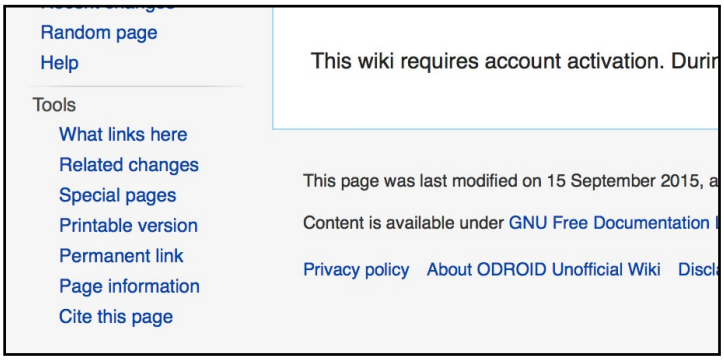
WIKI DE LA COMUNIDAD

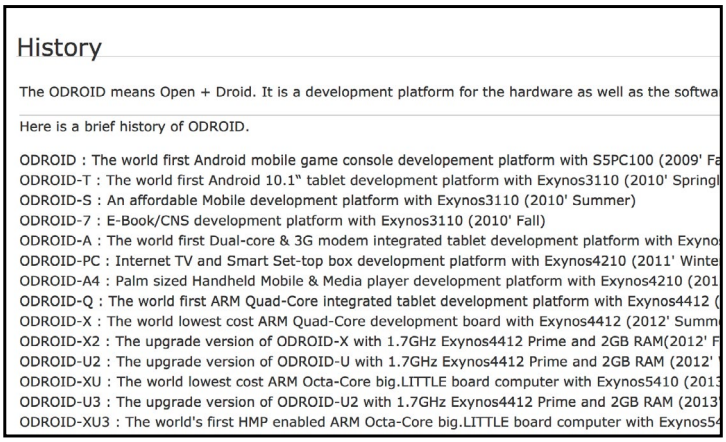
CONTRIBUYE A AMPLIAR LA BASE DE CONOCIMIENTO DE ODROID

por Rob Roy

Hardkernel ha puesto en marcha recientemente un gran recurso para los ODROIDians que deseen aportar sus conocimientos a una wiki de la comunidad, disponible en `http://wiki.odroid.in`. Está hecha con la intención de complementar la wiki oficial de Hardkernel de `http://bit.ly/1R6DOgZ`, útil para que publiques tus consejos, enlaces a imágenes de la comunidad, proyectos y cualquier otra cosa que pueda ser beneficiosa para la comunidad de Hardkernel.

Si deseas participar, haz clic en el botón "Request Account" en la parte superior derecha, e incluye tu nombre de usuario del foro ODROID en la sección "Personal Biography". Para comentarios, preguntas y sugerencias, por favor visite el hilo del foro original ent `http://bit.ly/1QDMNoT`.





PAGINA WEB DE ODROID MAGAZINE

NUEVO DISEÑO PARA NUESTRO TERCER AÑO

por Rob Roy

Ya que estamos celebrando el inicio de nuestro tercer año de publicación, ¡ODROID Magazine ha creado una nueva página web! Compruébalo en `http://magazine.odroid.com`. Cuenta con varias mejoras:

Formulario busqueda rápida de artículos

Índices de Contenidos con enlaces a los artículos individuales
`magazine.odroid.com/articles`

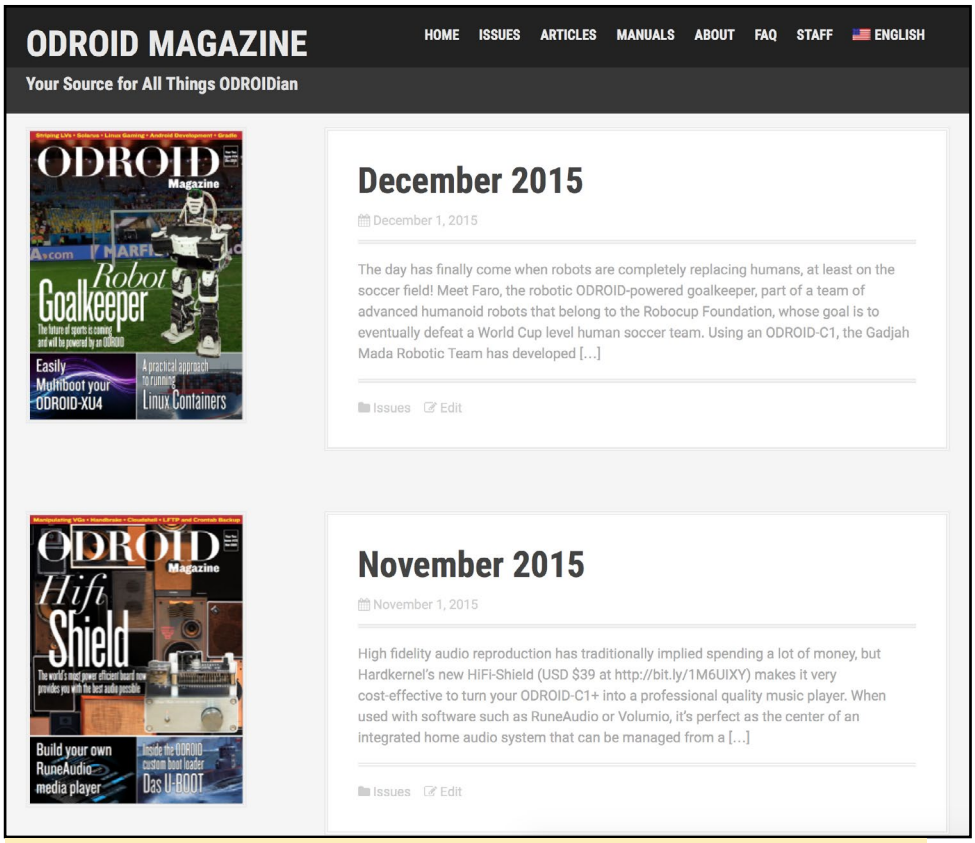
Sección de Manuales de usuario

Sección para Donaciones
`magazine.odroid.com/about`

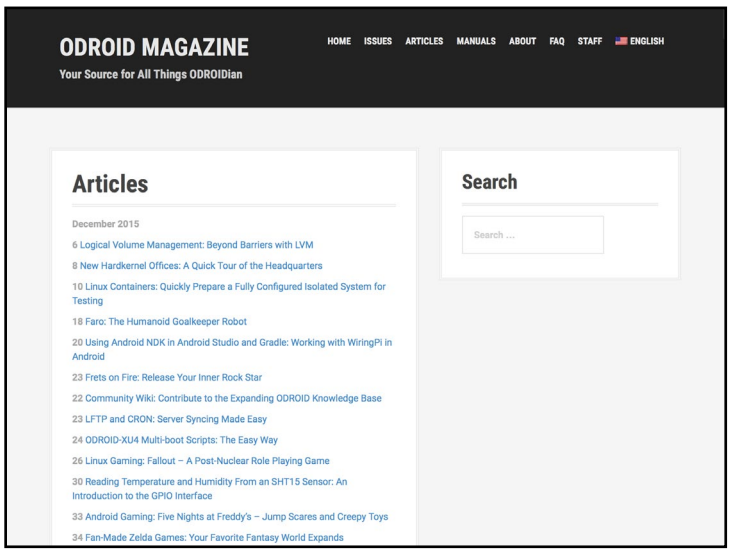
Rápido Acceso a los números individuales escribiendo magazine.odroid.com/YYYYMM
`magazine.odroid.com/201512`

Compatible con PC de escritorio y dispositivos móviles

Disponible en Español



Página Principal de la nueva web de ODROID Magazine mostrando los últimos números



El nuevo sitio cuenta con un índice de contenidos que vincula todos los artículo de cada número

Seguiremos actualizando el diseño y la apariencia de la página en los próximos meses, así que estate atento a las mejoras. Para comentarios, preguntas o sugerencias, por favor visita el post original en `http://bit.ly/1k5x8Ea`.

ANDROID 6.0 MARSHMALLOW PARA ODROID-XU4

EL ANDROID MAS RECIENTE PARA TU NUEVO ODROID

por @voodik

El Usuario @voodik ha publicado recientemente una versión de Android 6.0.1 CyanogenMod 13.0 para el ODROID-XU4. Sus características incluyen el kernel Linux versión 3.10.9, OpenGL ES 1.1 / 2.0 / 3.0 y OpenCL 1.1 EP. Permite hasta 8 usuarios, un punto de acceso Wi-Fi portátil, y soporte para HDMI-CEC.

Este mes, Android Marshmallow ha comenzado su lanzamiento internacional para smartphones, ofrece varias mejoras con respecto a Lollipop, incluyendo actualizaciones visuales, gestión inteligente de la batería, y características de seguridad para las aplicaciones más robustas. @voodik también ha incluido Google Play en su compilación, por lo que no es necesario instalarlo por separado.

Instalación

Se necesita en primer lugar una tarjeta SD o módulo eMMC que esté preparada para almacenar una de las imágenes de auto-instalación, disponible en `http://bit.ly/1QO9Wcn`. Puedes encontrar información detallada sobre cómo grabar imágenes en la Wiki de Hardkernel en `http://bit.ly/1Vk9u4o`.

Si utilizas una tarjeta SD, graba la imagen sd_ installer en la tarjeta, luego, inserta la tarjeta SD en el ODROID-XU4. Si usas un módulo eMMC, la imagen de auto-instalación denominada sd2emmc_installer debe grabarse en una tarjeta SD temporal, después tanto la tarjeta SD y el módulo eMMC deben introducirse en el ODROID-XU4.

Con el interruptor de hardware de soporte de arranque fijado en "tarjeta SD", enciende el ODROID-XU4 para instalar el sistema operativo. El LED azul permanecerá encendido durante el proceso y el ventilador arrancará. Una vez que la instalación haya terminado, el ODROID se apagará automáticamente. Si utilizas el módulo eMMC, cambia el interruptor de hardware de soporte de arranque a "eMMC". Después, encendido el ODROID y a disfrutar de Android Marshmallow.

Problemas conocidos

La versión todavía está en desarrollo, el soporte para Bluetooth y USB-3G aún están pendiente. La barra de búsqueda de Google tiene que ser desactivada desde la configuración del Launcher y añadirla de nuevo desde los widgets para que se visualice correctamente, y el MTP debe desactivarse, luego reactivarlo desde el menú Developer Options -> Select USB Configuration.

Trucos

Para conseguir que el WiFi funcione, coloca el nombre del módulo correcto en build.prop. Por ejemplo, para utilizar el módulo RealTek 8192cu por defecto, añade o actualizar la siguiente línea:

```
wlan.modname=8192cu
```

Para el módulo de Realtek 8188eu, utiliza la siguiente línea:

```
wlan.modname=8188eu
```

Para los módulos Ralink RT33XX/RT35XX/RT53XX/RT55XX debe quedar especificado así:

```
wlan.modname=rt2800usb
```

Para habilitar el GPS USB, introduce el TTY correcto y la velocidad en el archivo build.prop:

```
ro.kernel.android.gps=ttyACM0
ro.kernel.android.gps.speed=9600
```

Para comentarios, preguntas o sugerencias, por favor visita el post original en `http://bit.ly/1YEt9BG`.

UNIVERSAL MOTION JOYPAD

¿ESTAS LISTO PARA CONDUCIR UN COCHE DE CARRERAS?

por John Lee y Charles Park

El Universal Motion Joypad es el nuevo mando para juegos de Hardkernel , similar al volante que hay disponible para la Nintendo Wii. Ha sido fabricado utilizando la USB-IOBOARD y un sensor de movimiento auxiliar, te permite jugar a muchos juegos que hay disponibles para teléfonos móviles y que normalmente utilizan el giroscopio del teléfono. Está disponible para su compra en la tienda Hardkernel (`http://bit.ly/1Sbe46q`) por 40$, ¡Te proporcionará muchas horas de diversión!

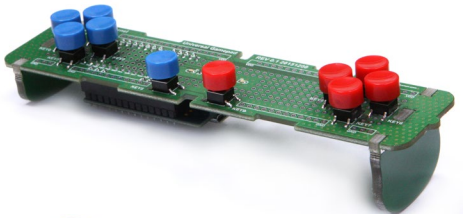
Figura 1 - Primer plano del Universal Motion Joypad

Características

El Joypad es personalizable ajustando el archivo boot.ini del ODROID sin necesidad de volver a compilar ninguno de los paquetes. La placa de circuito impreso (PCB) está diseñada como una placa universal. Por lo tanto, si los usuarios pueden hacer soldaduras por sí mismos, pueden conectar interruptores o dispositivos de entrada.

Componentes

- Sensor de Acelerómetro
- Placa universal
- 10 botones EA redondos
- LED de estado 2EA
- MCU de 8-bit
- Interfaz USB



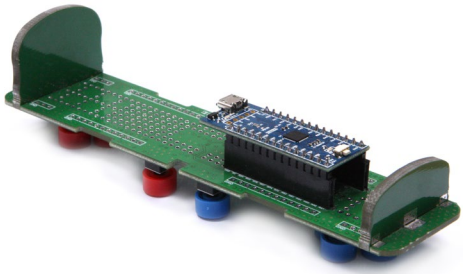


Figura 2a y 2b - Vista superior e inferior de la PCB del Joypad

Diseño

El Universal Motion Joypad se compone de una simple estructura de hardware. El MCU PIC18F45K50 dentro de la USB-IOBOARD lee los botones y el sensor acelerómetro BMA150, y controla los LED de estado.

El dispositivo USB-IOBOARD

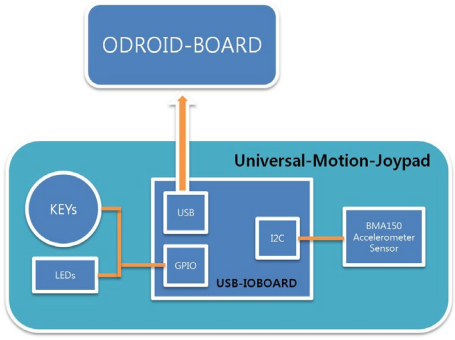


Figura 3 - Diagrama por bloques del Joypad

utiliza el microcontrolador (MCU) PIC18F45K50 fabricado por Microchip, que activa las funciones de reloj, serie y GPIO. Como muestra la Figura 4, la GPIO y I2C se utilizan para controlar el Universal Motion Joypad. El USB-IOBOARD lee la señal de los botones a través de los puertos GPIO. Los botones normalmente muestran un estado HIGH, y cuando se pulsan presentan un estado LOW.

El sensor acelerómetro está conectado al bus I2C. Cuando arranca el MCU, se activan los correspondientes registros. Cada botón y el sensor acelerómetro leen los valores a intervalos regulares utilizando el reloj del MCU, los valores se almacenan en forma de paquetes de 10 bytes y se transmite al ODROID a través de la interfaz USB.

Cada paquete de 10 bytes contiene una posición de cabecera y de cola de 1 byte cada una, junto con 8 bytes de datos. Los datos constan de 6 bytes de datos del sensor de aceleración y 2 bytes

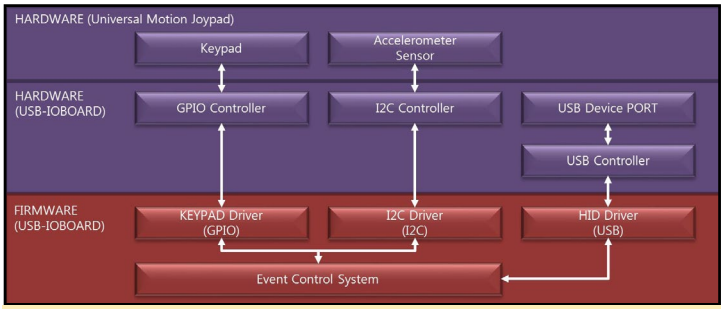


Figura 4 - Diagrama de la Interfaz del Joypad

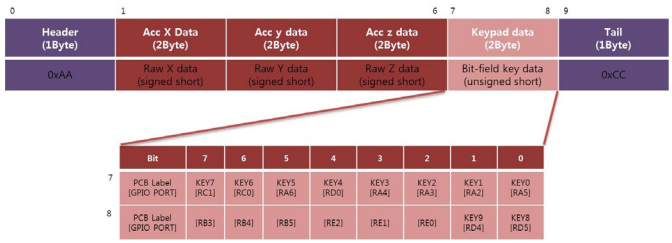


Figura 5 - Asignaciones de los bytes con los paquetes de datos

de datos de las teclas. Los valores KEY0 - KEY9 de la PCB son asignados a las correspondientes GPIO como se muestra en la Figura 5, con la opción de utilizar puertos de entrada adicionales. El puerto GPIO y la asignación de teclas pueden configurarse fácilmente con el archivo boot.ini del ODROID.

Entorno de trabajo de Android

Como muestra la Figura 6, el marco de trabajo es una aplicación que usa el entorno de trabajo del sensor para obtener datos del mismo. Se comunica con la capa C++ a través de la interfaz nativa de Java (JNI). La capa intermedia de la librería del sensor se compone del gestor del sensor, el servicio del sensor y de la capa de abstracción de hardware del sensor.

El subsistema de entrada es un entorno de trabajo Linux genérico para todos los dispositivos de entrada como teclados, ratones y pantallas táctiles, que define un conjunto de eventos estándar. Se comunica con el espacio de usuario con la interfaz /sys/class/input. El evento dispositivo (evdev) ofrece una forma genérica de que los eventos de dispositivos de entrada sean accesibles en /dev/input/eventX. El driver de la interfaz del sensor se comunica con la USB-IOBOARD a través del bus USB.

Beach Buggy Racing

Uno de nuestros juegos de carreras favoritos para Android es Beach Buggy Racing. Es un juego al estilo Mario Kart que cuenta con increíbles gráficos, una suave jugabilidad y hasta 4 jugadores. El juego es compatible con muchos joysticks USB y pantallas táctiles HDMI, pero es mucho más divertido y fácil de manejar con el Universal Motion Joypad.

Figura 6 - Diagrama del entorno de trabajo de los drivers del Joypad

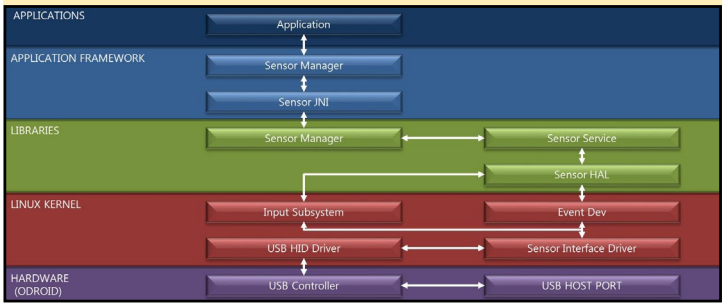


Figura 7 - Beach Buggy Racing

Para permitir que el movimiento Joypad universal como un controlador, añadir los siguientes parámetros en el archivo boot.ini en la partición FAT32 Android:

```
setenv orientation "3"
setenv bts "0:M:L,1:M:U,2:M:R,3:M:D,4:T:71:203,5:K:10
2,6:T:1214:212,7:T:1253:43,8:M:1,9:M:r"
```

A continuación, actualice la "bootargs" variable de acuerdo con el modelo de ODROID específica como se muestra a continuación. Después de guardar el archivo, reinicie el ODROID para los nuevos parámetros surtan efecto.

ODROID-XU3/4

```
setenv bootargs "fb_x_res=${fb_x_res}
fb_y_res=${fb_y_res} hdmi_phy_res=${hdmi_phy_res}
edid=${edid} hpd=${hpd} led_blink=${led_blink}
acc_orientation=${orientation} button_map=${bts}
usbhid.quirks=0x04d8:0x003f:0x0004"
```

ODROID-C2

```
setenv bootargs "${rootopt} ${consoleopt}
hdmimode=${hdmimode} hdmitx=${cecconfig}
vout=${vout_mode} disablehpd=${disablehpd}
logo=${logoopt} ${androidopt} ${selinuxopt}
suspend_hdmiphy=${suspend_hdmiphy}
acc_orientation=${orientation} button_map=${bts}
usbhid.quirks=0x04d8:0x003f:0x0004"
```

Estos cambios envían tres parámetros al kernel para asignar las teclas. Ten en cuenta que otros juegos pueden necesitar una orientación diferente que se ajusta en el parámetro acc_orientation. El button_ map fija la asignación de tecla de cada botón.

KEY	type	value1	value2
KEY Number (PCB Layout)	M	U : Move up mouse cursor D : Move down mouse cursor L : Move left mouse cursor R : Move right mouse cursor l : click mouse left button m : click mouse middle button r : click mouse right button	
	T	Touch X position	Touch Y position
	K	Key code	

Figura 8a - tabla asignación de botones

Accelerometer sensor orientation				
Sensor PCB	2	3	0	1
PCB Sensor	7	4	5	6

Figura 8b - tabla asignación Acelerómetro

Cada tecla se puede asignar a un evento de ratón, un evento de pantalla táctil o un evento de teclado. El parámetro usbhid.quirks fuerza a la USB-IOBOARD para que funcione como driver de entrada genérico en lugar de dispositivo HID estándar. Una explicación detallada de los parámetros button_map y acc_orientation se muestran en las Figuras 8a y 8b.

Asignaciones de botones

La asignación de los botones del keypad se puede ajustar utilizando el argumento de arranque "button_map". El número de la etiqueta de la tecla se utiliza como primer parámetro. El segundo parámetro especifica la operación, con el tercer y cuarto parámetro se ajustan los valores correspondientes a la operación. Los botones también se pueden asignar a la funcionalidad del ratón o a la pantalla táctil en lugar de teclas normales.

```
button_map=[KEY]:[type]:[value1]
:[value2]
```

Por ejemplo, para asignar la entrada KEY0 a un clic izquierdo del ratón y la KEY1 a unas las coordenadas táctiles x = 200, y = 200, los parámetros serían:

```
button_map=0:M:l,1:T:1200:200
```

Puede obtener las coordenadas de una entrada táctil, activando el parámetro "Settings" -> "Developer options" -> "Pointer location" en Android.

Asignaciones del Acelerómetro

El argumento "acc_orientation" se puede utilizar para ajustar el movimiento del Joypad en cualquier dirección. Se cambia configurando el parámetro del sensor del acelerómetro:

```
acc_orientation=<value>
```

Por ejemplo, si quieres que el valor de la rotación del Joypad llegue a los 90 grados hacia la derecha, debes ajustar el valor de acc_orientation a 2:

```
acc_orientation=2
```

Compatibilidad con sistemas operativos

Para utilizar el Joypad, primero tiene que actualizar el sistema operativo Android a la última versión:

- ODROID-XU3/XU4 Android version 4.4.4 (v3.3), fecha publicación 22 Dic, 2015, version del kernel 3.10.9: http://bit.ly/1Ossuyo
- ODROID-C1 Android version 4.4.2 (v2.1), fecha publicación 11 Dic, 2015, version del kernel 3.10.33: http://bit.ly/1JB0jWs

Asignación de teclas Beach Buggy Racing

Para que puedes empezar rápidamente a jugar con Beach Buggy Racing, aquí tienes la asignación de teclas del joypad para el juego, que debes añadir al archivo boot.ini:

```
setenv bts "0:M:L,1:M:U,2:M:R,3:M
:D,4:T:71:203,5:K:102,6:T:1214:21
2,7:T:1253:43,8:M:l,9:M:r"
```

Esta cadena de parámetros configura las entradas con los siguientes valores:

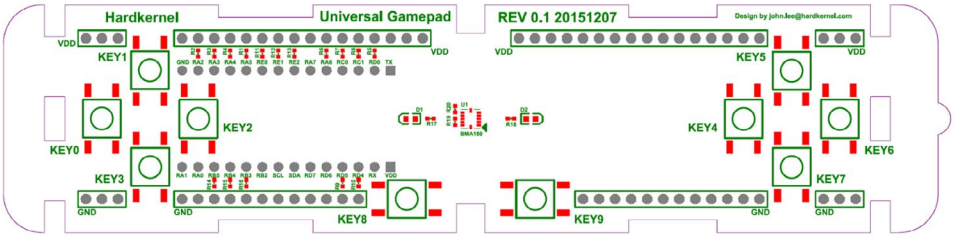
```
KEY0 = Mouse Left direction
KEY1 = Mouse Up direction
KEY2 = Mouse Right direction
KEY3 = Mouse Down direction
KEY4 = Touch Click (Left up side
item click : X = 71, Y = 203)
KEY5 = Key (Home : keycode is
102)
KEY6 = Touch Click (Right up side
item click : X = 1214, Y = 212)
KEY7 = Touch Click (Game Pause
click : X = 1253, Y = 43)
KEY8 = Mouse Left button
KEY9 = Mouse Right button
```

Las coordenadas X-Y de la pantalla táctil están ajustadas para una resolución de pantalla HDMI de 1280x720. Si estás usando una resolución diferente como 1920x1080, las coordenadas deben cambiarse. Como se ha mencionado anteriormente, las coordenadas táctiles se pueden obtener a partir de la configuración para desarrolladores "Pointer Location". Asegúrate de reiniciar el ODROID tras modificar el archivo boot.ini.

Información del hardware

- Sensor Acelerómetro con Sistema Micro electromecánico (MEMS)
- Brújula electrónica con sensor BMA150, con un sensor de campo magnético de tres ejes y un acelerómetro de tres ejes

Figura 9 - Diagrama PCB del Joypad



Código fuente

Puede aprender cómo implementar la HAL de la interfaz del sensor, y descubrir lo que ha cambiado en el código fuente al añadir los drivers a Android, revisando los siguiente commits de GitHub:

ODROID-XU3/4

Librería sensor libsensor
`http://bit.ly/1MATKDh`

Librería sensor.odroidc.so
`http://bit.ly/1YIFgIB`

usbio-keypad.idc
`http://bit.ly/1PpgYB0`

Driver placa sendor USBIO
`http://bit.ly/1OmbLX9`

Fichero configuración kernel
`http://bit.ly/1Ost5jG`

Driver Key pad USBIO
`http://bit.ly/1J9PRur`

ODROID-CI

Librería sensor libsensor
`http://bit.ly/1MATKDh`

Librería sensor.odroidc.so
`http://bit.ly/1kkTTEq`

usbio-keypad.idc
`http://bit.ly/1TlxlQ1`

Driver Placa sendor USBIO
`http://bit.ly/1mbuGy3`

Fichero configuración kernel
`http://bit.ly/1R1wAfJ`

Driver key pad USBIO
`http://bit.ly/1mbuLSq`

Arbol de fuentes Firmware MCU USB-IOBOARD
`http://bit.ly/1R1wG75`

Descarga firmware MCU
`http://bit.ly/1QSV1NR`

Echa un vistazo al Universal Motion Joypad en acción en `http://youtu.be/xSlPJzXEsIo`.

Beach Bunny Racing también es divertido

RESPETAR EL TRABAJO DE TU COMPAÑERO

PLEASE, DON'T TOUCH ANYTHING DEMUESTRA QUE LA IGNORANCIA PUEDE SER MUY DIVERTIDA.

por Bruno Doiche

Te quedas al cargo de un escritorio con un gran botón rojo, mientras que tu compañero se toma un descanso para ir al baño, te da instrucciones de que no toques nada. ¿Y Cuál es tu primer impulso? ¡Pues tocar algo, por supuesto!

Esta es la premisa de Please Don't Touch Anything: estás justamente sentado en un escritorio controlando un misterioso aparato que tiene múltiples controles lógicos e ilógicos capaces de acabar con la civilización tal como la conocemos.

Con un montón de finales distintos y asombrosa creatividad, ¡pasarás horas haciendo combinaciones de efectos posibles e imposibles que este juego ofrece!

Please Don't Touch Anything está disponible para su descarga desde la tienda Play Google en el siguiente enlace:

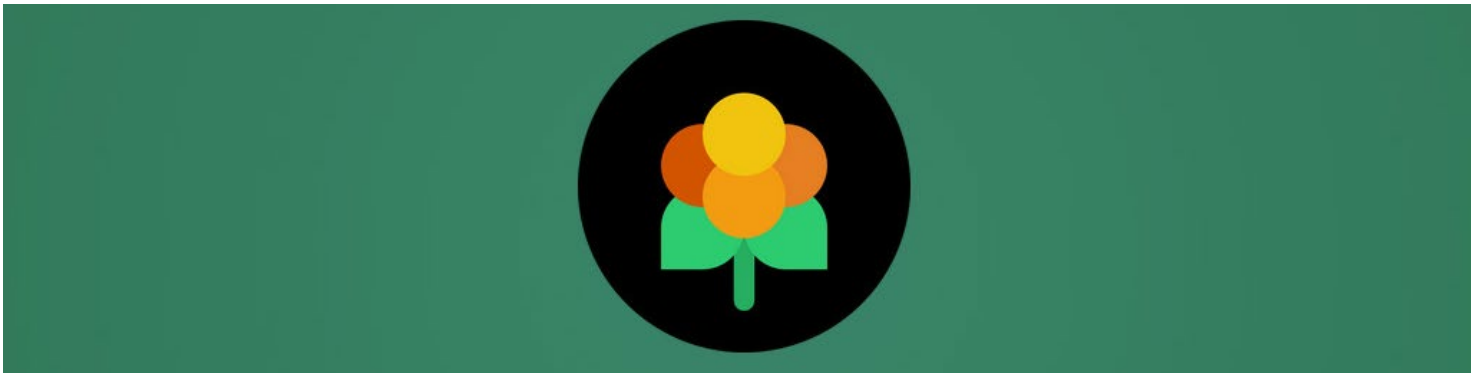
`https://play.google.com/store/apps/details?id=com.fourquarters.PleaseDontTouchAnything`

Abarcando desde lo obvio a la absoluta locura, este juego consumirá muchas horas de tu tiempo, ¡sacarle el máximo partido!

LAKKA PARA ODROID-XU4

EL MEJOR SISTEMA PARA JUEGOS

editado por Andrew Ruggeri



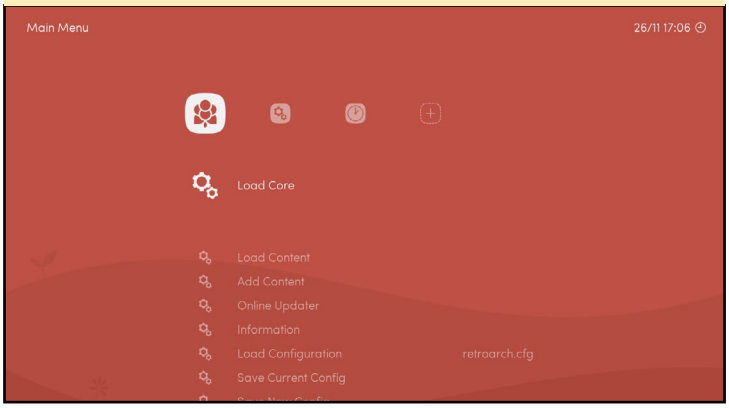
Lakka es una distribución Linux muy liviana que permite transformar un pequeño ordenador en una consola de juegos en toda regla. El sistema operativo, que puede funcionar tanto en el ODROID-XU4 como en el ODROID-C1, cuenta con una gran cantidad de emuladores diferentes, conocidos como "cores", así como una atractiva interfaz de usuario. Los desarrolladores de Lakka han anunciado recientemente el lanzamiento de una nueva e importante versión de Lakka. Esta nueva versión todavía está basada en OpenELEC 5, pero incluye el último RetroArch, con muchos cambios en la interfaz gráfica. Los enlaces de descarga y la guía de instalación se han actualizado e incluye una nueva forma de lanzar juegos.

Características

Cuando arrancas una nueva instalación de Lakka, se empieza con 4 pestañas, como se muestra en la Figura 1.

- El menú RetroArch, para lanzar las ROMs manualmente, y cerrar el sistema operativo
- La pestaña Settings, para configurar tu copia de Lakka
- La pestaña History, para consultar los juegos a los que ha jugado recientemente
- La pestaña +, para añadir nuevas pestañas.

Figura 1 - Pestañas Lakka



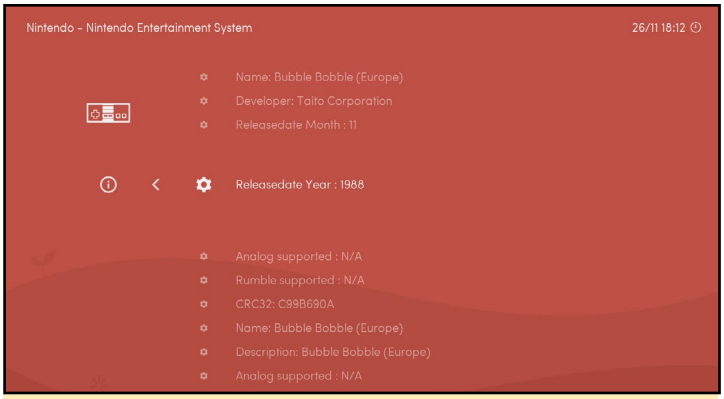
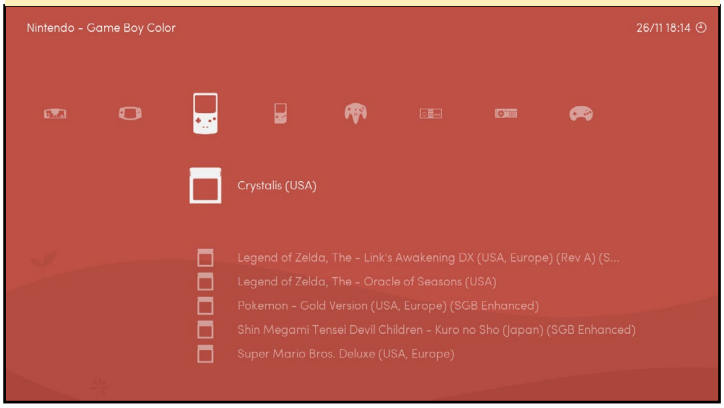


Figura 2 – Lakka, ahora incluye una base de datos de juegos

- Base de datos, Búsqueda rápida y listas de juegos

El nuevo RetroArch incluido en Lakka introduce una base de datos de juegos. Esta base de datos no es más que los archivos DAT de la famosa iniciativa No-Intro bit.ly/1RKJT5L, convertidos a binario por razones de rendimiento. Contiene metadatos sobre los juegos, como son el editor comercial y la fecha de lanzamiento. Con esto, es fácil consultar los juegos creados por el mismo equipo. Esta base de datos también incluye las sumas de verificaciones de las ROMs, RetroArch utiliza estas sumas de verificación para comparar tus ROMs con

Figura 3 - Lakka tiene un nuevo sistema de búsqueda



las entradas de la base de datos.

Dado que los usuarios se quejaban constantemente de la errónea clasificación de los juegos en las versiones anteriores de Lakka, se ha implementado un sistema de búsqueda rápida que te permite consultar todas tus ROMs comparándolas con la base de datos, generando así una única lista de juegos. Es por ello que te recomendamos utilizar la colección de Roms de No-Intro a partir de ahora, ya que la lectura se hace correctamente. Nos aseguramos de que no

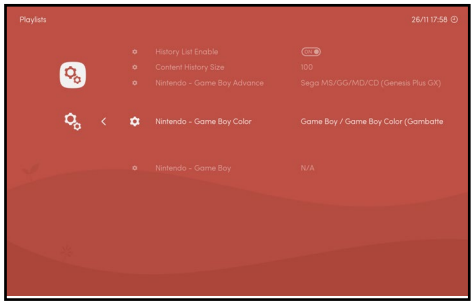


Figura 4 - Lakka asigna un "core" por defecto a cada lista de juegos

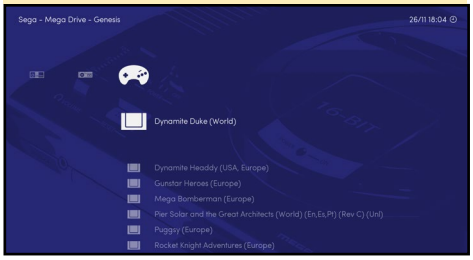
funcionen las listas de juegos personalizas que no hagan referencia a un sistema de juego o que contengan ROMs que no forman parte de No-Intro.

Se asigna un core libretro por defecto a cada lista de juegos. Puede personalizar esta opción en la configuración de las listas de juegos. También puede eliminar completamente esta vinculación pulsando START. RetroArch te permitirá elegir el core sobre una base por ROM.

Fondos dinámicos y boxarts

Los desarrolladores de RetroArch recientemente han añadido un sistema E/S sin bloqueos que nos permite cargar imágenes sin llegar a causar retardos en

Figura 5 – Se pueden seleccionar imágenes de fondo personalizadas



la interfaz. Con esto, podemos poner en marcha una primera versión de dos funciones muy conocidas: fondos dinámicos y boxarts.

Con fondos dinámicos, puedes colocar una imagen de fondo personalizada en cada entrada del menú horizontal:

- **Activa la función en "Menu Settings"**
- **Introduce tu directorio de fondos dinámico en "Directory Settings" para que apunte a la carpeta que contiene las imágenes.**

Nombra las imágenes tras los títulos que aparecen en la esquina superior izquierda de nuestra interfaz. El formato debe ser RGBA PNG. Esta función

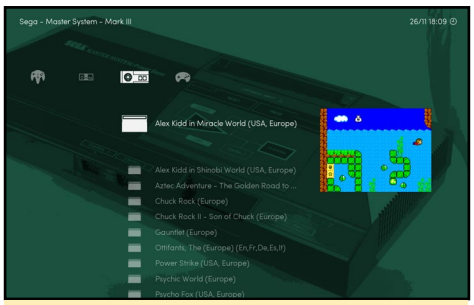


Figura 6 – También se puedes visualizar los Boxarts de los juegos

flaquea cuando el hardware es pobre, pero funciona muy bien en los PCs. Se proporcionan tres paquetes listos para usar en esta versión dentro del directorio /usr/share/retroarch-assets/wallpapers/.

Los Boxarts funciona exactamente igual, excepto que tienen que seguir una estructura de directorios para evitar conflictos de nombres. Por ejemplo, "boxarts/Nintendo - GameBoy/Named_Snaps /Tetris (World).png". Las imágenes tienen que estar en formato RGBA PNG, y tienen que ser nombradas teniendo en cuenta las entradas en la lista vertical. Aún no ofrecemos paquetes de Boxart, pero el proyecto No Intro Screenshot Reloaded (http://bit.ly/1O9Vuoi) proporciona algunos paquetes de capturas de pantalla.

DRM/KMS

Un grave error con la GPU Radeon nos impedía proporcionar desarrollo DRM/ KMS de Lakka para PCs. El usuario @lugaidster en el seguimiento de incidencias de RetroArch encontró una solución que podemos utilizar en Lakka, permitiéndonos eliminar la enorme dependencia de X11. Sin embargo, algunos usuarios han informado de bastantes incompatibilidades y del malos resultados con los desarrollos DRM/KMS. Seguimos creyendo que el X11 gratuito es el futuro, y que los proveedores pronto lanzará drivers que seran compatibles con DRM/KMS. Mientras tanto, puedo ofrecer desarrollos X11 extraoficiales.

Corrección de errores

El error con los 55 FPS que afectaba al ODROID-XU3 y XU4 finalmente se ha solucionado. No se trataba de un error de visualización. Sólo necesitábamos fijar la tasa de audio a 44100 en lugar de los 48000 por defecto. Un espacial agradecimiento a @maister que encontró el origen del problema. Estamos muy contentos de proporcionar una versión estable para el XU3 y XU4 después de todo este tiempo.

Funciones que faltan

El sistema de búsqueda está en proceso. Los siguientes sistemas todavía tienen que ser implementados:

- **Busqueda Arcade**
- **Busqueda PCE-CD**
- **Busqueda Sega CD**

Actualización

Los archivos de configuración de RetroArch y la estructura de directorios de la partición de almacenamiento han cambiado bastante. De modo que se recomienda utilizar la ruta de actualización habitual, y hacer una nueva instalación en esta ocasión. No olvides de hacer una copia de seguridad de tus juegos y partidas guardas. Si tiene preguntas o comentarios, por favor visite el post original en `http://bit.ly/1PSklD7`.

JUGOS LINUX

JUEGOS DE ESTRATEGIA EN EL ODROID PARTE I

por Tobias Schaaf

Siempre he sido un jugador de PC, y uno de mis géneros favoritos está ampliamente disponibles para los ordenadores, pero rara vez lo encuentras en las consolas: ¡Los juegos de estrategia! Realmente me encantan, incluso de niño ya me fascinaban este tipo de juegos. En nuestro viejo Amiga, en el que arrancaba Dune 2, este era un género que lograba mantenerme ocupado durante horas y horas. El tema de planificar tu estrategia, crear un ejército y aplastar a tu enemigo me atrajo mucho. Tengo muy buenos recuerdos de impresionantes sesiones multijugador con mis amigos jugando StarCraft y al Total Annihilation. Incluso hoy en día, me siguen gustado mucho este tipo de juegos. Estoy deseando hacerme con la trilogía de StarCraft 2. Puesto que son tantos los juegos de estrategia, es muy difícil decidir cuáles son los mejores. Me voy a centrar en algunos juegos de DOS algo antiguos que son bastante buenos, y que se pueden ejecutar en la plataforma ODROID usando DOSBox.

Figura 1 - Panzer General, un simulador de la 2ª Guerra Mundial, donde guías a ejércitos en batallas. Cada pérdida es una necesidad previamente calculada.

Figura 2 - Settlers 2 es uno de esos juegos que te mantendrán ocupado durante muchas horas simplemente planificando tus rutas comerciales

Subgéneros

Existen muchos y diferentes subgéneros de juegos de estrategia. Algunos son juegos de estrategia táctica como History Line 1914-1918, Battle Isle, Gettysburg, y Panzer General, donde tienes una cantidad limitada de unidades y tienen que planificar tus movimientos para alcanzar tu objetivo, hacer retroceder al enemigo, o conquistar un punto de interés. Cuentas con opciones limitadas para aumentar el número de unidades, como llamar a las unidades de apoyo o invertir el poco dinero del que dispones para comprar unidades de reemplazo. Por lo general, no te desases de unidades tras atacar a un enemigo hasta que no hayas ganado la batalla. Más bien, planificas cada movimiento y saber que cada unidad cuenta.

Otro subgénero de estrategia son los juegos de simulación, en el que tiene que manejar mucho más que tus tropas y tu base. Generalmente, tienes que ajustar y distribuir los recursos, tener presente la diplomacia, organizar diversos sitios y lidiar con múltiples conflictos al mismo tiempo. El juego más conocido de este tipo es, probablemente la serie Civilization, pero hay muchos más, como Imperium Galactica, Master of Orion, Fragile Alliance, la serie Settlers y la serie Anno.

El último subgénero que quiero mencionar dentro de los juegos de estrategia son los juegos de construcción de bases, incluyendo juegos como los ya mencionados Dune 2, StarCraft y la famosa serie Command and Conquer entre otros. Estos son mis juegos de estrategia favoritos, donde tienes que construir una base, crear un gran ejército y tratar de aplastar a tu enemigo, al mismo tiempo que desarrollas tus defensas para retener los ataques de tus enemigos.

Existen más subgéneros de juegos de estrategia, como los juegos de estrategia por turnos como mi serie favorita XCOM, o la serie Heroes of Might and Magic, pero por ahora quiero limitar el número y géneros de juegos a unos cuantos. Ya que con lo que más disfruto son con los juegos de construcción de bases y ejércitos, me concentraré en estos.

Preparación

Puesto que los juegos en los que me quiero centrar son para DOS, usare mi

fiel emulador DOSBox para ejecutarlos. Dispones de una versión de DOSBox funcional en mi repositorio, debería serte fácil conseguir ejecutarlos en DOSBox. En primer lugar, crea una carpeta para almacenar los juegos y una subcarpeta llamada CDs para almacenar los archivos ISO, luego instalar DOSBox:

```
$ mkdir -p DOS/CDs
$ sudo apt-get install dosbox-
odroid
```

Inicia DOSBox una vez que crees el archivo de configuración por defecto. Después, salte de inmediato. Abre /home/odroid/.dosbox/dosbox-SVN.conf con un editor de texto y cambia las siguientes líneas, después vuelve a iniciar DOSBox desde el menú de Aplicaciones:

```
[sdl]
fullscreen=true
fullresolution=desktop
windowresolution=1024x768
output=overlay
[dosbox]
memsize=31
[render]
 frameskip=3
aspect=true
[cpu]
core=dynamic
cycles=auto
```

Por comodidad, también deberías añadir estas líneas al final del archivo de configuración que se encuentra en /home/odroid/.dosbox/, reemplazando MyISO.iso por el nombre de tu ISO:

```
[autoexec]
mount c: /home/odroid/DOS -free-
size 1024
imgmount d: /home/odroid/CDs/
MyISO.iso -t iso
c:
```

Una vez que el emulador se haya iniciado, la carpeta DOS automáticamente será montada como la unidad C: y MyISO.iso será montada como D: como si fuera una unidad de CD-ROM.

Ten en cuenta que he añadido la opción –freesize 1024 para la montar la unidad C: en DOSBox, que es necesaria para los juegos que comprueban el tamaño del espacio disponible en el disco duro. Sin esta opción, la unidad C: sólo se monta con unos 300 MB libre, a pesar de que el tamaño de la tarjeta SD o módulo eMMC es mucho mayor.

Command And Conquer

Uno de los primeros juegos de este género fue Dune 2, la gente lo suele llaman el abuelo de todos los juegos de estrategia en tiempo real (RTS). Fue el primer gran éxito tras Westwood, en el cual creas una base y unidades y las envías a la batalla. Mientras que con Dune 2 juegas en un mundo lejano con grandes edificios luchando unos contras otros por el territorio y el poder, Command and Conquer nos trae de vuelta a Tierra y nos hace participar en una guerra mundial, donde existen dos grandes bandos, GDI y NOD que luchan entre sí.

Figura 3 - Command and Conquer utilizaba un montón de películas y pequeñas escenas para contar la historia que era realmente buena por el aquel entonces.

Command and Conquer era un juego más largo que Dune 2, con una historia a seguir donde te identificas como el comandante. Tus superiores te hablaban directamente en pequeñas escenas, y se puede seguir la evolución de la guerra en partes de noticias y películas que hablan sobre el impacto de tu última misión. Estas características, combinadas con un impresionante sistema de combate, la progresión de las construcciones y las unidades de tu arsenal, lo convierten en un juego único de su clase.

Figura 4 - La construcción de tu base y las unidades era muy divertido en Command and Conquer.

Siempre recordaré las muchas horas a las que he jugado a este juego, realmente es divertido jugarlo en el ODROID. Me trae muy buenos recuerdos de una de las mejores series de la historia de los videojuegos. Command and Conquer 1 y 2 (Alerta Roja) están disponibles para DOS y funcionan muy bien en Android.

Earth 2140

¡Me encanta este juego! No es muy conocido y los posteriores, como Earth 2150, 2160 y sus varias extensiones eran divertidas y tenía muy buenos gráficos en 3D por aquel entonces, pero ninguno llego a superar al original. Este juego era más avanzado a nivel gráficos que el Command and Conquer. Se creó a una resolución muy alta, incluso para DOS, y los efectos eran bastante mejores que lo que ofrecía el Command and Conquer.

Figura 5 - El humo, el terreno dañado y el fuego con una apariencia muy realista, son sólo algunos de los impresionantes efectos de Earth 2140

En Earth 2140, puedes jugar al lado de la Eurasian Dynasty (ED) o en el bando de los United Civilized States (UCS). Ambos bandos son muy diferentes y hacen que el juego sea muy particular. Como ED, normalmente tienen soldados y tanques para combatir a tu enemigo. Como UCS, utilizas cyborgs y robots de combate, muy parecidos a los tipicos de las películas de Terminator.

Por desgracia, el juego carece de una buena historia de fondo, pero puedes apreciar en qué se destino el dinero. La interfaz y el juego en sí cuentan con unos gráficos realmente impresionantes para un juego de DOS de su época. Lo que mejor recuerdo son los tanques de láser de los ED, realmente me encantaban. Un grupo de tanques de láser puede abrasar a un enemigo en cuestión de segundos. El láser penetraba en la armadura como si fuera a cortar mantequilla, y era impresionante ver diez o veinte rayos láser sobre un único objetivo enemigo.

Los del bando de UCS, por otro lado, intimidaban con robos que podían lanzar una especie de plasma devastador sobre el enemigo, y muchas de sus unidades contaban con granadas napalm, que hacían daño con el tiempo, así como daño al AOE. Este es, sin duda uno de los próximos juegos en mi lista al que jugaré en el ODROID, realmente estoy deseando probarlo. Si te gustan los juegos de estrategia, pero no ha oído hablar de la Eart 2140, hazte con una copia desde GoG.com y pruébalo, vale la pena.

Gene Wars

Gene Wars es muy particular, es uno de los juegos al que actualmente estoy jugando en mi ODROID. Está hecho por Bullfrog, una de mis compañías favoritas de la época. En este juego, puedes criar diferentes tipos de animales, además de cruzarlos para alcanzar tus objetivos, que van desde conseguir una determinada cantidad de puntos o investigar algo, a la destrucción de la base de los enemigos.

Dispones de cuatro tipos de trabajadoress: un ingeniero que construye y mejora tus edificios, un botánico con el que cultivas diferentes plantas, un guarda forestal que te ayuda a criar y cruzar tus animales, y un experto en genética, para investigar nuevas especies y curar tus criaturas cuando presente heridas. Este juego es fascinante ya que combina el típico objetivo de "construir una base con un ejército" con aspectos de crecimiento y de cruce de especies. Hay incluso una zona botánica, donde tiene que decidir qué plantas cultivar para producir recursos y/o alimentos para tus criaturas en un terreno específico.

El terreno es otra cuestión interesante del juego. Éste se puede cambiar: las colinas pueden ser allanadas por el constructor y así conseguir más espacio para edificios, y si un edificio explota puede dejar un gran cráter en el suelo, un efecto muy avanzado para ser un juego DOS. Por último usa tus criaturas y diferentes trabajadores para realizar las tareas que te son encomendadas por una especie alienígena que te vigila tanto a ti como a tus enemigos. Estas tareas incluyen tener un determinado tipo de especies en un lugar específico, con las que ganaras puntos y complacerás a los alienígenas, o cultivar plantas en diferentes lugares. A menudo, habrá un jugador enemigo con tus mismas capacidades y objetivos, así que o bien tiene que ser más rápido que él o obligarle a que se retire, lo cual a veces puede resultar muy difícil.

Algunas criaturas sólo pueden realizar ciertos tipos de tareas. Por ejemplo, las primeras criaturas con las que cuentas son mulas, que tienen bastante fuerza para trabajar, pero que no son muy buenas a la hora de atacar. Se pueden utilizar para cosechar las plantas y convertirlas en "goop", que es el material básico de construcción. Más adelante, encontrarás cangrejos, que también puedes clonar usando la investigación. Los cangrejos son más fuertes que las mulas y son excelentes luchadores. Lamentablemente, no son muy buenos para el trabajo diario. Pueden talar un árbol, pero no pueden portearlo como lo hace una mula.

Más adelante, podrás cruzar estas especies, creando criaturas con cualidades de defensa y ataque más fuertes que una mula sin que dejen de transportar mercancías. Este tipo de cruces se puede hacer con todas las especies que encuentres en el planeta. Hay un total de 5 especies diferentes, y todas ellas pueden ser cruzadas entre sí. Cada raza cruzada puede tener dos resultados, dependiendo de los genes que se vuelvan dominantes, lo que da lugar a un total de 25 criaturas diferentes que puedes conseguir a lo largo del juego. Si lo haces bien y los alienígenas están contentos contigo, es probable que consigas un monolito que se puede utilizar para aumentar las capacidades de tus trabajadores y de las criaturas.

El juego incluye varios idiomas. No reinventa el género, pero utiliza aspectos muy particulares de este tipo de juegos de estrategia. Te aconsejo que guardes el juego a menudo y en diferentes ranuras. Puesto que no existe "un reinicio de nivel", tienes que cargar una partida guardada si fallas en un nivel y sólo se puede iniciar una nueva partida desde el menú de inicio, de modo que guardar el juego costantemente es fundamental.

Figura 6 - ¿Es una mula, o se trata de un cangrejo? No, ¡es una mula-cangrejo! Umm, quiero decir un cangrejo-mula

Fragile Alliance

Si eres un fan del Amiga, puede que hayas oído hablar de un juego llamado K240. Fragile Alliance es similar a este juego. Eres un operativo de una compañía minera llamada Tetra Corp, y tu principal objetivo es el de conseguir créditos para tu empresa. Para ello, se supone que

tienes que recoger minerales raros de los asteroides. Tienes que crear un entorno de vida estable con filtros de radiación, emplazamientos vitales para tus trabajos, minas para recoger minerales, instalaciones de entretenimiento para mantener contentos a tus trabajadores, además de centros de seguridad para mantener el orden en tus asteroides. Más tarde, podrás fabricar tus propias naves espaciales que tienen diferentes "slots", que puedes equipar con diversas armas o equipos de defensa. Incluso puedes construir estaciones espaciales y grandes navíos.

Más adelante, conocerás a otras especies y tendrás que interactuar con ellas haciendo uso de la diplomacia o la acción militar. Puede colonizar nuevos asteroides para conseguir más recursos y créditos, o conquistarlos si están ocupados por otras especies. Si prefieres destruir al enemigo antes de intentar hacerte con un asteroide, cuentas con un amplio arsenal de misiles para hacerlo igualmente. Hay contrabandistas y comerciantes para vender y comprar productos, y puedes tener supervisores que

Figura 7 - En Fragile Alliance te asientas sobre asteroides, los defiendes y creas ejércitos y misiles para aumentar tu poder y riqueza.

te ayudarán a crear mejores asteroides. También puede usar espías para sabotear las producciones y asteroides enemigos.

Existen muchas tácticas diferentes para luchar contra los enemigos, y el arsenal de armas en este juego es impresionante, especialmente los misiles. Hay incluso un denominado misil éstasis que te permite congelar todo un asteroide y todo lo que está a su alcance. Una táctica muy común es, por ejemplo, congelar un asteroide con un misil éstasis y luego, enviar varios misiles y naves al asteroide. Una vez que el éxtasis desaparece, todos tus misiles y naves empezarán a atacarlo a la vez. Este juego también cuenta con un modo multijugador muy bueno, que puede mantenerte ocupado durante horas y horas jugando en único mapa.

Z

Otro de mis juegos favoritos de estrategia se llama Z. En este juego, no puedes construir nuevos edificios para producir unidades, pero puedes conquistar sectores que tienen fábricas predefinidas, que producen diferentes tipos de unidades. Este juego es todo táctica y velocidad. Cuantos más sectores tengas, más rápido se construyen las nuevas unidades. Cuantos más sectores tenga el enemigo, más rápido construye sus unidades y más fuerte se vuelve, así que es muy importante ser rápido a la hora de conquista sectores y mantenerlos.

El juego es bastante difícil, aunque tiene un cierto toque humorístico con escenas muy divertidas y cuenta con un

Figura 8 - Z es un juego muy bueno sobre robots que luchan en la guerra del futuro por territorios. No enfades el General Zod

sistema de lucha muy trabajado. Los gráficos son realmente buenos y hay muchas unidades interesantes por descubrir y construir. Si eres un fan de los juegos de táctica rápida, Z está hecho para ti.

Mención especial

Existen muchos más juegos de estrategia disponibles para DOS, como Wing Commander Armada, que combina elementos de estrategia de conquista de planetas, construcción de defensas y creación de nuevas naves espaciales, con elementos de un juego de disparos cuando luchas contra un enemigo o necesitas defender tu sistema. Incluso con un batallón débil, puedes defenderte de toda una flota enemiga, si cuentas con buenas habilidades para pilotar.

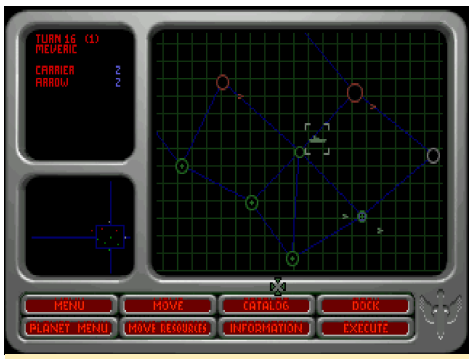


Figura 9 – En Wing Commander Armada, construyes flotas, recursos mineros y combates aéreos contra tus enemigos

El clásico Civilization es una serie muy conocida que también empezó en DOS, y con el que Sid Meier escribió la historia del juego. Muchos juegos siguieron la serie Civilization, y ahora casi todo el mundo reconoce el nombre de Sid Meier en diversos juegos. Además de Civilization, siempre me encanto Colonization, que es un juego sobre la colonización de América y la Guerra de la Independencia. Colonization es uno de mis favoritos de todos los tiempos en Amiga.

Hay muchos otros juegos de estrategia para DOS como Constructor, donde eres dueño de una empresa de construcción y construyes casas para personas de todas las clases. Sueles luchar contra cucarachas gigantes, o contratar matones que aterroricen a la competencia. Krush Kill 'N Destroy (KKnD) es un juego de estrategia post-apocalíptico que combina elementos de Command and Conquer con elementos de Fallout y Mad Max, creando un entorno muy peculiar para ser un juego de estrategia.

Esta lista de juegos de estrategia en DOS es lo suficientemente consistente como para mantenerte ocupado durante meses. Gracias a la plataforma Android, puedes reproducir estos juegos en su moderno TV. ¡Revive o experimenta por primera vez estas impresionantes joyas de la historia de los videojuegos!

CAJA PARA ODROID-XU4

ELEGANTE, MODERNA Y SILENCIOSA

por @Chris Oi

He creado mi propia caja mecanizada para mi ODROID-XU4. Está hecha del aluminio que se utiliza en los aviones con refrigeración pasiva utilizando un núcleo de cobre. Primero cree un archivo 3D utilizando el software Inventor para que fuese más fácil fabricar la misma, se puede descargar desde http://bit.ly/1Jfd35b. Las imágenes que se muestran a continuación detallan el progreso de construcción, que consiste en una pieza inferior elevada con unas goma, una pieza superior con hendiduras para la refrigeración y un logo perso-nalizado de ODROID.

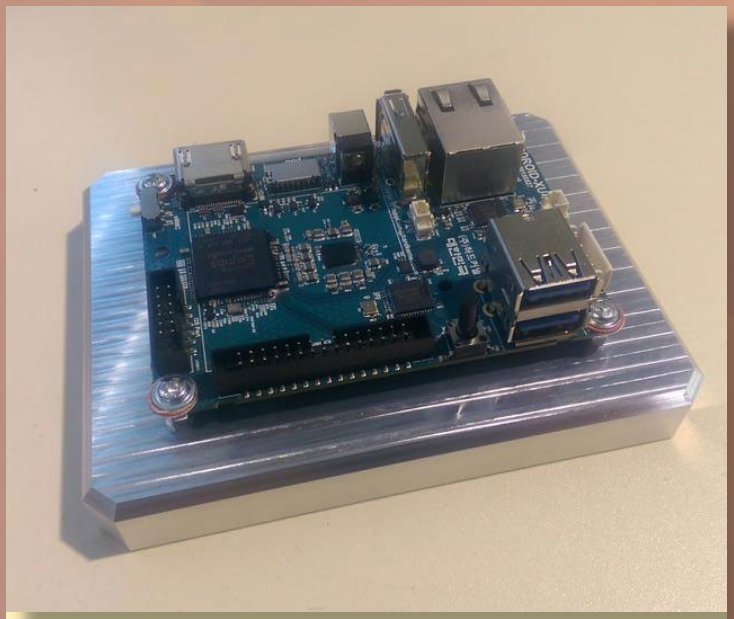


Figura I - ODROID-XU4 unido a la base

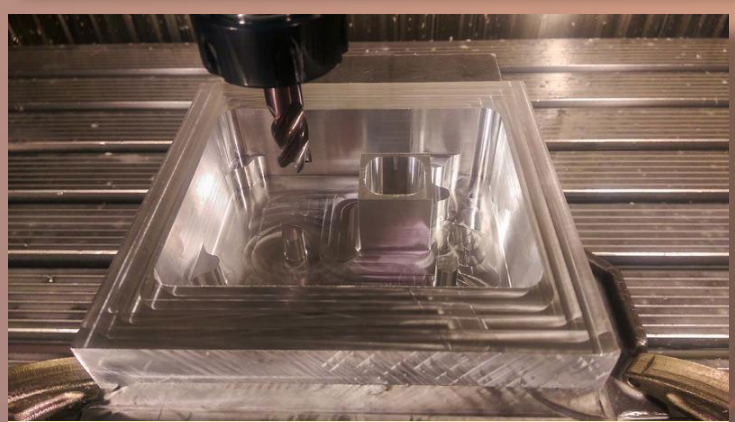


Figura 2 - Fresado el interior del parte superior de la caja

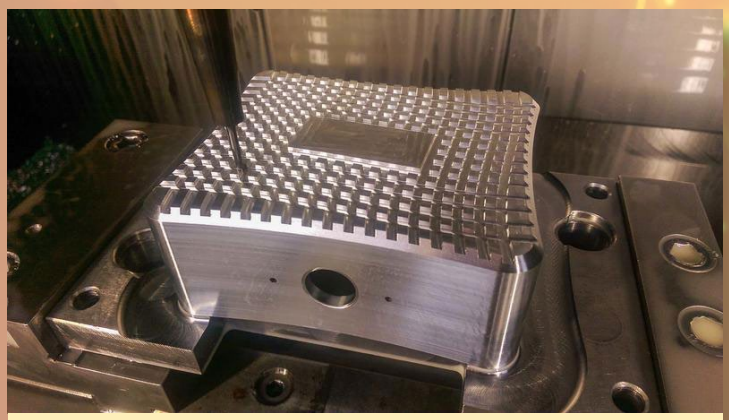


Figura 3 - Creando las hendiduras para la refrigeración en el exterior de la parte superior de la caja

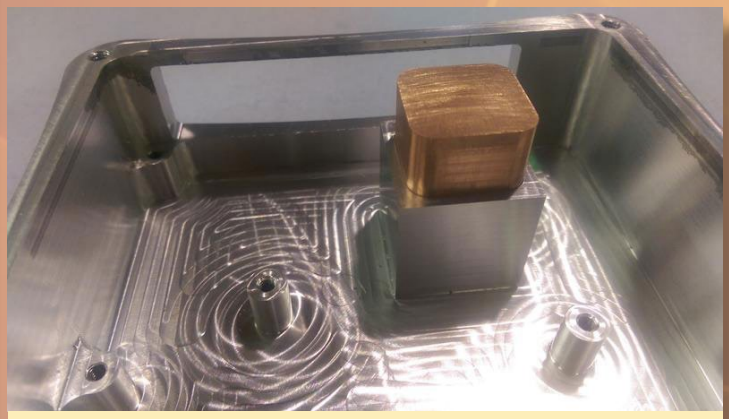


Figura 4 - El núcleo de cobre unido al interior de la caja para la refrigeración pasiva

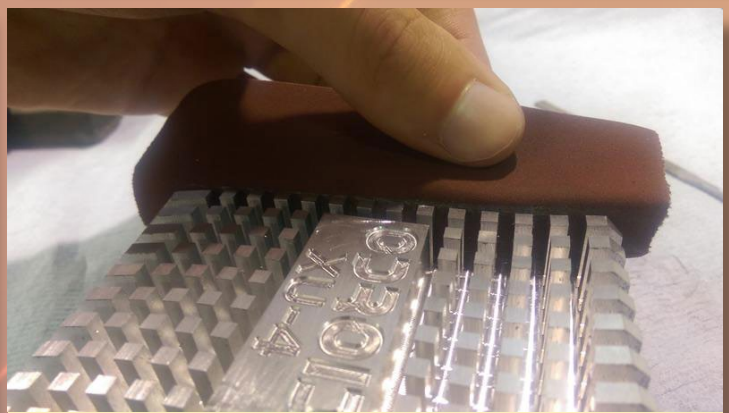


Figura 5 - Puliendo el prototipo final

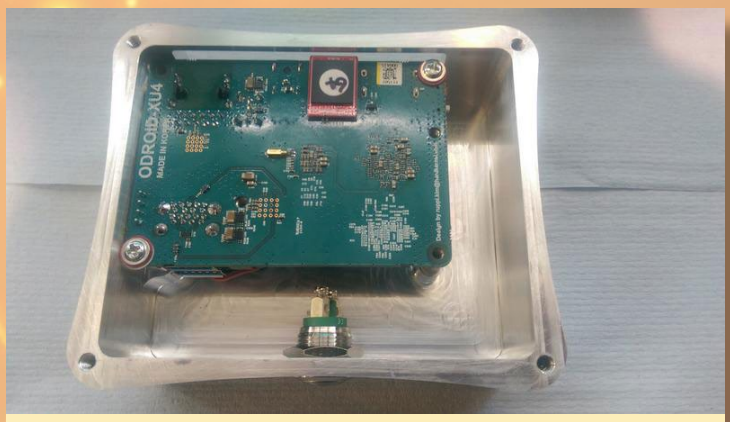


Figura 6 - ODROID-XU4 dentro del área de refrigeración de la parte superior

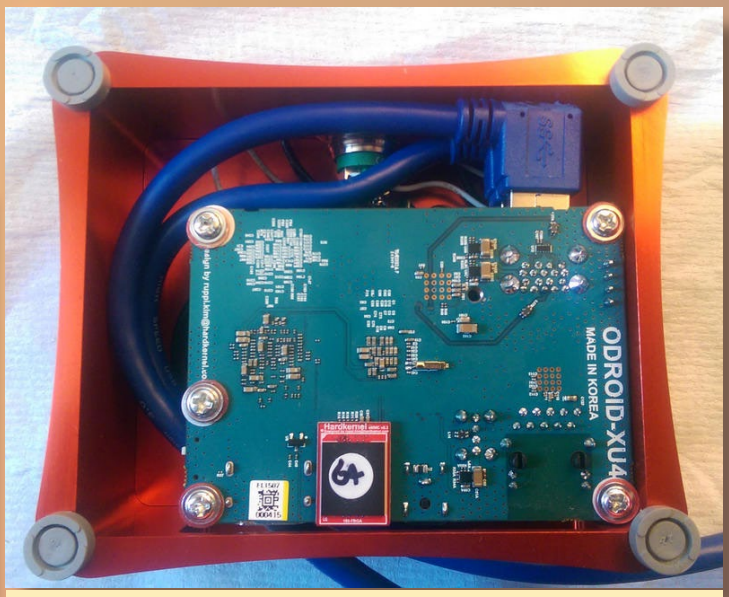


Figura 9 - El montaje final de la caja sin la parte inferior

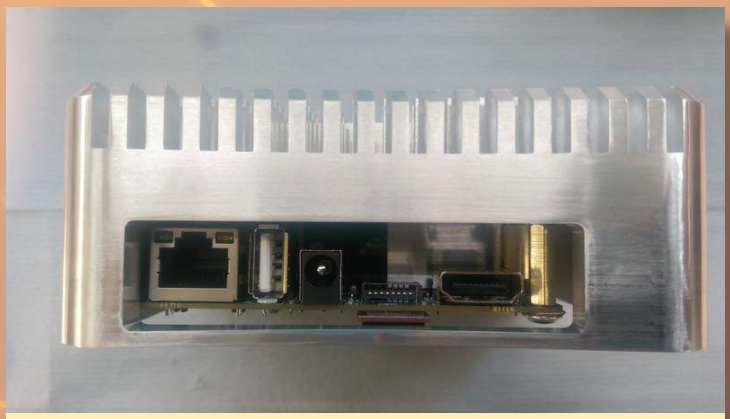


Figura 7 – Vista posterior de la caja

Figura 10 - El producto terminado

Figura 8 - Al producto final se le aplicó una anodización de color rojo

Para comentarios, preguntas y sugerencias, por favor visita el post original en `http://bit.ly/1LjgnM4`.

Este es nuestro intento de replicar la caja de aluminio del ODROID-XU4 - ¡creemos que lo hemos conseguido!

S.O. DESTACADO

TIZEN PARA ODROID-XU4

editado por Andrew Ruggeri

Tizen es un sistema operativo embebido basado en Linux que se ejecuta en una amplia gama de dispositivos, incluyendo teléfonos inteligentes, tabletas, televisores inteligentes, ordenadores, cámaras inteligentes, dispositivos portátiles como los smartwatches, reproductores Blu-ray, impresoras y electrodomésticos inteligentes. Su objetivo es ofrecer una experiencia de usuario uniforme en los dispositivos. Tizen está desarrollado por Samsung e Intel, por lo que no es de extrañar que su origen se remonte a Bada y MeeGo. Este artículo detalla los pasos necesarios para crear una imagen de arranque de Tizen 3 para el ODROID-XU3 o el XU4. El artículo original lo puedes encontrar en bit.ly/1O52IhH.

Preparar el binario

Para hacer una tarjeta de arranque, necesitas conseguir todos los archivos que se incluyen en la siguiente tabla y guardarlos en un único directorio de un PC host que sea capaz de grabarlos una tarjeta microSD:

Nombre	Tipo
bl1.bin.hardkernel	binary
bl2.bin.hardkernel.1mb_uboot	binary
tzsw.bin.hardkernel	binary
u-boot-mmc.bin	binary
sd_fusing_xu4.sh	shell script

Para obtener estos archivos, primero descarga los binarios del pre-gestor de arranque escribiendo los siguientes comandos en una ventana de terminal:

```
$ wget https://github.com/hardkernel/u-boot/raw/
odroidxu3-v2012.07/\
sd_fuse/hardkernel_1mb_uboot/bl1.bin.hardkernel
$ wget https://github.com/hardkernel/u-boot/raw/
odroidxu3-v2012.07/\
sd_fuse/hardkernel_1mb_uboot/bl2.bin.hardkernel.1mb_
uboot
$ wget https://github.com/hardkernel/u-boot/raw/
odroidxu3-v2012.07/\
sd_fuse/hardkernel_1mb_uboot/tzsw.bin.hardkernel
```

Descarga el último u-boot y el kernel desde bit.ly/1QGBzDW, extrae el contenido de la imagen tarball y cambia los nombres de los binarios u-boot:

```
$ tar xvf tizen-tv_xxxxxxxx.x_\
tv-boot-armv7l-odroidxu3.tar.gz
```

Guarda el siguiente script en un archivo llamado “sd_fusing_xu4.sh”:

```
#!/bin/bash

declare FORMAT=""
declare DEVICE=""

# Binaires array for fusing
declare -a FUSING_BINARY_ARRAY
declare -i FUSING_BINARY_NUM=0

declare CONV_ASCII=""
declare -i FUS_ENTRY_NUM=0
declare -r FUSING_IMG="fusing.img"

# binary name | part number | offset | bs
declare -a PART_TABLE=(
      "bl1.bin.hardkernel"          ""      1
512
      "bl2.bin.hardkernel.1mb_uboot" ""     31
512
      "u-boot-mmc.bin"              ""      63
512
      "tzsw.bin.hardkernel"         ""      2111
512
      "params.bin"                  ""      6272
512
      "boot.img"                    1       0
512
      "rootfs.img"                  2       0
4M
      "system-data.img"             3       0
4M
```

```
	"user.img"
5	0	4M
	"modules.img"
6	0	512
	$FUSING_IMG
""	3072	512
	)

declare -r -i PART_TABLE_ROW=4
declare -r -i PART_TABLE_
COL=${#PART_TABLE[*]}/${PART_TA-
BLE_ROW}

# partition table support
function get_index_use_name () {
	local -r binary_name=$1

	for ((idx=0;idx<$PART_TA-
BLE_COL;idx++)); do
		if [ ${PART_
TABLE[idx * ${PART_TABLE_ROW} +
0]} == $binary_name ]; then
			return $idx
		fi
	done

	# return out of bound index
	return $idx
}

# fusing feature
function convert_num_to_ascii ()
{
	local number=$1

	CONV_ASCII=$(printf
\\$(printf '%03o' $number))
}

function print_message () {
	local color=$1
	local message=$2

	tput setaf $color
	tput bold
	echo ""
	echo $message
	tput sgr 0
}

function add_fusing_entry () {
	local name=$1
	local offset=$2
	local size=$3

	FUS_ENTRY_NUM=$((FUS_ENTRY_
NUM + 1))

	echo -n "$name" > entry_
name
	cat entry_name /dev/zero |
head -c 32 >> entry

	echo -n "" > entry_offset
	for ((i=0; i < 4; i++))
	do
		declare -i var;
		var=$(( ($offset >>
(i*8)) & 0xFF ))
		convert_num_to_
ascii $var
		echo -n $CONV_ASCII
> tmp
		cat tmp /dev/zero |
head -c 1 >> entry_offset
	done
	cat entry_offset /dev/zero
| head -c 4 >> entry

	echo -n "" > entry_size
	for ((i=0; i < 4; i++))
	do
		declare -i var;
		var=$(( ($size >>
(i*8)) & 0xFF ))
		convert_num_to_
ascii $var
		echo -n $CONV_ASCII
> tmp
		cat tmp /dev/zero |
head -c 1 >> entry_size
	done
	cat entry_size /dev/zero |
head -c 4 >> entry

	rm tmp
	rm entry_name
	rm entry_offset
	rm entry_size
}

function make_fusing_header () {
	# header magic
	echo -n "BFUS" > fus_hdr_
magic
	cat fus_hdr_magic | head -c
4 > fus_hdr

	# entry number: 1 byte
	convert_num_to_ascii $FUS_
ENTRY_NUM
	echo -n $CONV_ASCII > fus_
hdr_entry_num
	cat fus_hdr_entry_num /dev/
zero | head -c 4 >> fus_hdr

	rm fus_hdr_magic
	rm fus_hdr_entry_num
}

function make_fusing_struct {
	if [ -f entry ];then
		make_fusing_header
		cat fus_hdr entry /
dev/zero | head -c 512 > $FUS-
ING_IMG
		rm fus_hdr entry

		# Write Fusing
Magic Number */
		fusing_image $FUS-
ING_IMG
		rm $FUSING_IMG
	fi
}

function fusing_image () {
	local -r fusing_img=$1

	# get binary info using
basename
	get_index_use_name $(base-
name $fusing_img)
	local -r -i part_idx=$?

	if [ $part_idx -ne $PART_
TABLE_COL ];then
		local -r
device=$DEVICE${PART_
TABLE[${part_idx} * ${PART_TABLE_
ROW} + 1]}
		local -r
seek=${PART_TABLE[${part_idx} *
```

```
${PART_TABLE_ROW} + 2]}
            local -r bs=${PART_
TABLE[${part_idx} * ${PART_TABLE_
ROW} + 3]}
      else
            echo "Not supported
binary: $fusing_img"
            return
      fi

      local -r input_size=`du -b
$fusing_img | awk '{print $1}'`

      print_message 2 "[Fusing
$1]"
      dd if=$fusing_img | pv
-s $input_size | dd of=$device
seek=$seek bs=$bs

      if [ $(basename $fusing_
img) == "u-boot-mmc.bin" ];then
            add_fusing_entry
"u-boot" $seek 2048
      fi
}

function fuse_image_tarball () {
      local -r filepath=$1
      local -r temp_dir="tar_tmp"

      mkdir -p $temp_dir
      tar xvf $filepath -C $temp_
dir
      cd $temp_dir

      for file in *
      do
            fusing_image $file
      done

      cd ..
      rm -rf $temp_dir
      eval sync
}

function fuse_image () {

      if [ "$FUSING_BINARY_NUM"
== 0 ]; then
            return
      fi

      for ((fuse_idx = 0 ; fuse_
idx < $FUSING_BINARY_NUM ; fuse_
idx++))
      do
            local
filename=${FUSING_BINARY_
ARRAY[fuse_idx]}

            case "$filename" in
                *.tar | *.tar.
gz)
                    fuse_image_
tarball $filename
                    ;;
                *)
                    fusing_image
$filename
                    ;;
            esac
      done
      echo ""
}

# partition format
function mkpart_3 () {
      local -r DISK=$DEVICE
      local -r SIZE=`sfdisk -s
$DISK`
      local -r SIZE_MB=$((SIZE >>
10))

      local -r BOOT_SZ=64
      local -r ROOTFS_SZ=3072
      local -r DATA_SZ=512
      local -r MODULE_SZ=20

      let "USER_SZ = $SIZE_MB -
$BOOT_SZ - $ROOTFS_SZ - $DATA_SZ
- $MODULE_SZ - 4"

      local -r BOOT=boot
      local -r ROOTFS=rootfs
      local -r SYSTEMDATA=system-
data
      local -r USER=user
      local -r MODULE=modules

      if [[ $USER_SZ -le 100 ]]
      then
            echo "We recommend
to use more than 4GB disk"
            exit 0
      fi

      echo "====================
==================="
      echo "Label          dev
size"
      echo "====================
==================="
      echo $BOOT"            "
$DISK"1       " $BOOT_SZ "MB"
      echo $ROOTFS"          "
$DISK"2       " $ROOTFS_SZ "MB"
      echo $SYSTEMDATA"      "
$DISK"3       " $DATA_SZ "MB"
      echo "[Extend]""       "
$DISK"4"
      echo " "$USER"         "
$DISK"5       " $USER_SZ "MB"
      echo " "$MODULE"
" $DISK"6     " $MODULE_SZ "MB"

      local MOUNT_LIST=`mount |
grep $DISK | awk '{print $1}'`
      for mnt in $MOUNT_LIST
      do
            umount $mnt
      done

      echo "Remove partition
table..."
      dd if=/dev/zero of=$DISK
bs=512 count=16 conv=notrunc

      sfdisk --in-order --Linux
--unit M $DISK <<-__EOF__
      4,$BOOT_SZ,0xE,*
      ,$ROOTFS_SZ,,-
      ,$DATA_SZ,,-
      ,,E,-
      ,$USER_SZ,,-
      ,$MODULE_SZ,,-
      __EOF__

      mkfs.vfat -F 16 ${DISK}1 -n
$BOOT
      mkfs.ext4 -q ${DISK}2 -L
$ROOTFS -F
      mkfs.ext4 -q ${DISK}3 -L
$SYSTEMDATA -F
```

```
	mkfs.ext4 -q ${DISK}5 -L
$USER -F
	mkfs.ext4 -q ${DISK}6 -L
$MODULE -F
}

function show_usage () {
	echo "- Usage:"
	echo "  sudo ./sd_fusing*.
sh -d <device> [-b <path> <path>
..] [--format]"
}

function check_partition_format
() {
	if [ "$FORMAT" != "2" ];
then
		echo "-------------
----------"
		echo "Skip $DEVICE
format"
		echo "-------------
----------"
		return 0
	fi

	echo "---------------------
----------"
	echo "Start $DEVICE format"
	echo ""
	mkpart_3
	echo "End $DEVICE format"
	echo "---------------------
----------"
	echo ""
}

function check_args () {
	if [ "$DEVICE" == "" ];
then
		echo "$(tput setaf
1)$(tput bold)- Device node is
empty!"
		show_usage
		tput sgr 0
		exit 0
	fi

	if [ "$DEVICE" != "" ];
then
		echo "Device: $DE-
VICE"
	fi

	if [ "$FUSING_BINARY_NUM"
!= 0 ]; then
		echo "Fusing bina-
ry: "
		for ((bid = 0 ; bid
< $FUSING_BINARY_NUM ; bid++))
		do
			echo "
${FUSING_BINARY_ARRAY[bid]}"
		done
		echo ""
	fi

	if [ "$FORMAT" == "1" ];
then
		echo ""
		echo "$(tput setaf
3)$(tput bold)$DEVICE will be
formatted, Is it OK? [y/n]"
		tput sgr 0
		read input
		if [ "$input" ==
"y" ] || [ "$input" == "Y" ];
then
			FORMAT=2
		else
			FORMAT=0
		fi
	fi
}

function print_logo () {
	echo ""
	echo "Odroid-XU4 download-
er, version 0.5"
	echo "Authors: Inha Song
<ideal.song@samsung.com>"
	echo ""
}

print_logo

function add_fusing_binary() {
	local declare binary_
name=$1
	FUSING_BINA-
RY_ARRAY[$FUSING_BINARY_
NUM]=$binary_name
	FUSING_BINARY_
NUM=$((FUSING_BINARY_NUM + 1))
}


declare -i binary_option=0

while test $# -ne 0; do
	option=$1
	shift

	case $option in
	--f | --format)
		FORMAT="1"
		binary_option=0
		;;
	-d)
		DEVICE=$1
		binary_option=0
		shift
		;;
	-b)
		add_fusing_binary
$1
		binary_option=1
		shift
		;;
	*)
		if [ $binary_option
== 1 ];then
			add_fusing_
binary $option
		else
			echo "Unkown
command: $option"
			exit
		fi
		;;
	esac
done

check_args
check_partition_format
fuse_image

make_fusing_struct
```

Para crear el archivo ejecutable abre una ventana de terminal en el directorio donde se creó el script bash y escribe lo

siguiente:

```
$ chmod u+x sd_fusing_xu4.sh
```

Luego, instala las herramientas pv:

```
$ sudo apt-get install pv
```

Instalar en microSD

Conecta la tarjeta microSD usando un lector de tarjetas y comprueba el nodo del dispositivo.

```
$ sudo fdisk -l
```

El siguiente es un ejemplo del resultado del comando fdisk.

```
..........
Partition table entries are not
in disk order
Disk /dev/sdb: 32.0 GB,
32010928128 bytes
64 heads, 32 sectors/track, 30528
cylinders, total 62521344 sectors
Units = sectors of 1 * 512 = 512
bytes
Sector size (logical/physical):
512 bytes / 512 bytes
I/O size (minimum/optimal): 512
bytes / 512 bytes
Disk identifier: 0x00000000
Device Boot Start End Blocks Id
System
/dev/sdb1 * 8192 139263 65536 e
W95 FAT16 (LBA)
..........
```

Ejecuta “sd_fusing_xu4.sh”, seguido del nodo del dispositivo, que es /dev/sdb en el siguiente ejemplo. Ten en cuenta que el script puede tardar unos minutos.

```
$ sudo ./sd_fusing_xu4.sh -d \
/dev/sdb -b bl1.bin.hardkernel \
bl2.bin.hardkernel.1mb_uboot \
tzsw.bin.hardkernel \
u-boot-mmc.bin
```

Inserta tu micro SD en el ODROID y selecciona la SD como sistema de arranque con el interruptor de hardware.

Instalar en eMMC

Conecta el módulo eMMC a un ordenador de escritorio usando un lector de tarjetas y comprueba el nodo del dispositivo:

```
$ sudo fdisk -l
..........
Partition table entries are not
in disk order
Disk /dev/sdb: 7948 MB,
7948206080 bytes
245 heads, 62 sectors/track, 1021
cylinders, total 15523840 sectors
Units = sectors of 1 * 512 = 512
bytes
Sector size (logical/physical):
512 bytes / 512 bytes
I/O size (minimum/optimal): 512
bytes / 512 bytes
Disk identifier: 0x00000000
Device Boot Start End Blocks Id
System
/dev/sdb1 * 8192 139263 65536 e
W95 FAT16 (LBA).......
```

Ejecutar el script “sd_fusing_xu4.sh” seguido del nodo del dispositivo

```
$ sudo ./sd_fusing_xu4.sh -d /
dev/sdb --format
```

Pulse la tecla “y” para formatear, puede llevar unos minutos. Espere a que finalice el script. El siguiente es un ejemplo del resultado del script:

```
Device: /dev/sdb
/dev/sdb will be formatted, Is it
OK? [y/n]
y
------------------------------
Start /dev/sdb format
..........
```

Después, descarga la última imagen de arranque y de la plataforma desde bit.ly/1J2ytri y bit.ly/1S6BEjE. Conecta el eMMC al PC usando un lector de tarjetas y comprueba el nodo del dispositivo:

```
$ sudo fdisk -l
..........
Partition table entries are not
in disk order
Disk /dev/sdb: 7948 MB,
7948206080 bytes
245 heads, 62 sectors/track, 1021
cylinders, total 15523840 sectors
Units = sectors of 1 * 512 = 512
bytes
Sector size (logical/physical):
512 bytes / 512 bytes
I/O size (minimum/optimal): 512
bytes / 512 bytes
Disk identifier: 0x00000000
Device Boot Start End Blocks Id
System
/dev/sdb1 * 8192 139263 65536 e
W95 FAT16 (LBA).......
```

Ejecuta el script “sd_fusing_xu4.sh”, seguido del nodo del dispositivo. Ten en cuenta que “/dev/sdb” se utiliza en los siguientes comandos de ejemplo:

```
$ sudo ./sd_fusing_xu4.sh -d /
dev/sdb -b \
tizen-tv_20150824.1_tv-boot-arm-
v7l-odroidxu3.tar.gz \
tizen-tv_20150824.1_tv-wayland-
armv7l-odroidu3.tar.gz
```

Arrancar Tizen

Inserta la tarjeta microSD o módulo eMMC en el ODROID y selecciona el modo de arranque SD con el interruptor de hardware. Conecta la alimentación, entra en el promp u-boot y ejecuta los siguientes comandos:

```
# mmc dev 0
# mmc read 0x50000000 0x1 0xa3e
# mmc dev 1 1
# mmc write 0x50000000 0x0 0xa3e
# mmc dev 1 0
```

Después, cambiar el soporte de arranque a eMMC y reiniciar. Ahora, puedes usar el módulo eMMC con Tizen

CONOCIENDO UN ODROIDIAN

GEORG MILL, INNOVADOR Y CREATIVO "MAKER" DE HARDWARE

editado por Rob Roy

Por favor, háblanos un poco sobre ti.

Mi nombre es Georg Mill. Mi esposa y yo vivimos en Düsseldorf Alemania. Como dice el sitio web de Düsseldorf, "Düsseldorf es una bulliciosa metrópolis en el corazón de Europa con mucho que ofrecer a los residentes y los visitantes." He disfrutado viviendo en esta ciudad durante más de 10 años.

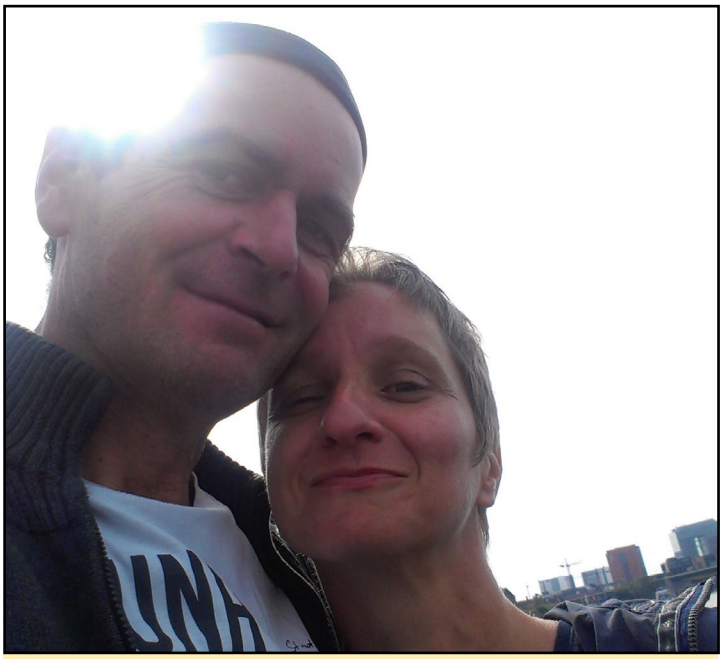
¿Cómo fueron tus inicios con los ordenadores?

Informático es mi segundo trabajo. Mi primer trabajo fue como impresor offset. De modo que no vengo de una Universidad, aunque aprendí lo básico en Heidelberg en 1999, en una escuela de formación profesional de informática centrada en el desarrollo web.

Mi primer contacto con los ordenadores vino de la mano de mi hermano, que desarrollo un dispositivo hardware para Pong cuando tenía 17 años a principios de los 70. Este dispositivo podía conectarse a un televisor. Mis padres nos regalaron ese año por Navidad: un kit de hardware electrónico para construir cosas como una radio FM. Por aquel entonces, los ordenadores eran raros y bastante caros, así que no teníamos uno. Incluso en nuestra escuela, no había nada de eso. Finalmente conseguí mi primer ordenador cuando finalice mis exámenes del instituto de enseñanza secundaria. Los teléfonos inteligentes y dispositivos portátiles no estaban inventados todavía. Nadie sabía lo qué era un "maker" , pero nosotros jugueteábamos con un experimentador de electrónica llamado "Kosmos Elektronikus Radio + Elektronik 1 y 2".

Con estos componentes electrónicos, mi hermano y yo fuimos capaces de construir el nuestro propio, un radio FM totalmente funcional. Mis padres se quedaron asombrados cuando les presentamos con orgullo este pequeño y extraño dispositivo. Eso pasó hace mucho tiempo. Ahora Tengo 51 años, y los dispositivos electrónicos son baratos, fáciles de usar y están al alcance de casi de todo el mundo.

Kosmos Elektronikus Radio



Georg y su esposa

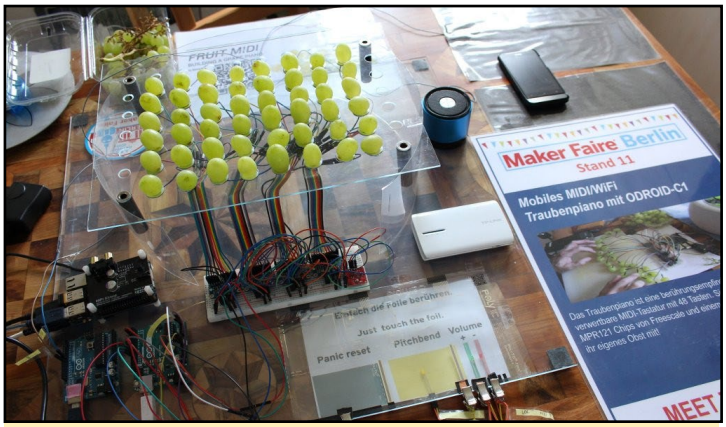
¿Qué te atrajo de la plataforma ODROID?

Hace algunos años, comencé a juguetear con un Arduino, me recordó los primeros días de mi infancia. Nos hubiera encantado que dispositivos como los ODROIDs estuviesen disponibles por aquel entonces. He utilizado Arduino para crear una batería electrónica desde cero. Las almohadillas de la batería amortiguaban los ruidos de los tambores hechos de malla de mosquitera y anillos de madera con algunos elementos piezoeléctricos. Finalmente desarrolle una versión mejor, la cual se describe en `http://bit.ly/1Yduav0`. La llamé YAAMI-Drum (`http://bit.ly/1Qfa18W`) y hacía lo que exactamente yo quería: una forma silenciosa de tocar la batería en nuestra casa sin molestar a los vecinos que vivían justo debajo de nosotros.

Todo el mundo conoce la Raspberry Pi, fue mi primer Ordenador de placa reducida (SBC). Me he dado cuenta que estos pequeños dispositivos son incluso mejores que un ordenador "normal" porque se puede conectar una gran cantidad de sensores, entradas y salidas a sus GPIOs. Me puse muy contento cuando desarrollé mi primer sistema de vigilancia con cámaras y radio internet por red con él. Después, intente desarrollar una solución para reproducir música en vivo procesando audio en tiempo real y encontré el ODROID-C1 original. El ODROID-C1+ parece que se puede utilizar con gran cantidad de juegos y Kodi (antes conocido como XBMC), pero lo utilizó para desarrollar mi propio kernel en tiempo real (rt_pre-

empt) y reproducir música en vivo con él. Otro proyecto es un sistema time-lapse para una cámara réflex digital EOS. Se pueden encontrar algunos ejemplos muy buenos en http://bit.ly/1ErAzMK, que me fueron de gran ayuda. Mi versión la puedes encontrar en http://bit.ly/1Ydufi7 y http://bit.ly/1miU5pb.

Uno de mis proyectos favoritos con el ODROID-C1+ es un proyecto de audio llamado "Traubenpiano", que significa "el piano con uvas". Es un teclado MIDI vía Wifi con funciones completas (http://bit.ly/1UgIRgE) que cuenta con dispositivos muy especiales como teclas: ¡las uvas! Fue presentado por primera vez en la Maker Faire Berlín 2015 http://bit.ly/1NjEf6L y en la Codemotionworld 2015 en Berlín (http://bit.ly/1jX8fKV). Lo verás la próxima vez aquí en Alemania en la Makerfaire-Ruhr 2016 en marzo (http://bit.ly/1OqeaAA).



Berlin 2015

¿Cuál es tu ODROID favorito?

Puesto que sólo tengo un C1+ en estos momentos, la respuesta es simple: el ODROID-C1+ con una placa shield de audio. Sin embargo, en el futuro intentaré conseguir un XU4 que también es una buena plataforma para el piano con uvas.

Tus proyectos hardware, especialmente el piano con uvas y el plátano discoteca, son muy creativos. ¿Qué te motivó a desarrollarlos?

Por supuesto, el "piano de uva" no es algo que se puede utilizar para producir música en serio. Es un proyecto que he desarrollado para ayudar a los niños de un hospicio aquí en Düsseldorf llamado "Rainbowland", o como decimos en alemán "Regenbogenland" (http://bit.ly/1Oqetvh). Los niños sufren de enfermedades muy raras, tan raras que no hay esperanza de conseguir una medicina que les ayudara a volver a tener vida normal y saludable. La mayoría de estos niños les gusta escuchar música, o incluso intentan hacer música ellos mismos. Dado que la mayoría de ellos han perdido el control de sus músculos, tienen que buscar otra manera de jugar. Por esta razón, empecé a desarrollar algunos dispositivos táctiles especiales que les permitieran hacer música sin necesidad de mover demasiado sus dedos o brazos. El dispositivo tiene un bajo consumo de energía y es fácil de usar, que es como deben trabajar estos instrumentos. Actualmente sigo trabajando en su desarrollo, y aún está lejos de estar listo para su uso.

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

Sería bueno si el ODROID-XU4 tuviese un Shield Hi-Fi como el que está disponible para el ODROID-C1+, o al menos una buena entrada y salida de audio de calidad.

¿Qué aficiones e intereses tienes aparte de los ordenadores?

A los 15 años, mi padre me compró un simple tambor Jazz de una compañía llamada Hoshino. Como resultado, la relación con nuestros vecinos se volvió un poco más tensa. Querían ver un partido de fútbol en la televisión, al mismo tiempo que yo quería aprender a tocar el tambor. Recientemente, he aprendido mucho de algunos bateristas de jazz profesionales aquí en Düsseldorf y Cologne, donde es posible tocar la batería sin molestar a la gente incluso sentado en la misma habitación.

Me gusta viajar y hacer pequeñas travesías con mi esposa. Ella tiene miedo a volar en un avión, por lo que nuestros destinos se limitan a los que se pueden llegar en coche o en tren. A menudo disfrutamos de la costa en pequeñas islas de los Países Bajos y Dinamarca durante las vacaciones. Allí el viento es muy constante, razón por la que me gusta volar cometas acrobáticas. La mayoría de la gente cree que no soy capaz de controlarlas, pero la verdad es que me encanta volarlas en círculos con rapidez y con un control total sobre ellas.

Cuando la programación se hace demasiado pesada, me gusta coger mi bicicleta de montaña y rodar por los bosques cercanos y a lo largo de Düsseldorf, que tiene muchos árboles. Si esto no es suficiente, consigo un poco de energía positiva practicando Aikido.

¿Qué consejo le darías a alguien que quiere aprender más sobre programación?

Mi único consejo es que sienta curiosidad por todo. Si estás interesado en la programación, tienes que ser consciente que nadie ha nacido con conocimientos sobre electrónica o con formación en TI. Es necesario mucha dedicación para aprender a hacer que las cosas funcionen. Grandes nombres como Linus Torvalds y Thomas Gleixner empezaron sus carreras en alguna parte. Hoy en día, ni siquiera necesitas ser un ingeniero o experto informático para conseguir hacer funcionar algo de domótica. Sólo tienes que sentir curiosidad sobre cómo funcionan las cosas. No pienses en grande. Empezando con pocas aspiraciones y pequeños proyectos, a menudo se hace más divertido que desarrollar un superordenador con miles de ODROIDS, ¡incluso si es sólo un piano con uvas!