

ODROID

Magazine

Año Cinco
Num. #57
Sep 2018



Commodore Juegos Clásicos

PREPARATE PARA EMULAR LOS MEJORES JUEGOS DE LOS AÑOS 90 CON NUESTRO EXCELENTE TUTORIAL.

HOME ASSISTANT:
SEGUIMIENTO DE PERSONAS
CON WI-FI USANDO KISMET

PROGRAMACION 2:
LEER EL ESTADO DE LOS
BOTONES I2 EN EL ODROID-GO





Convierte un Monitor en una Gigante Tablet Android

⌚ September 1, 2018

Este artículo describe cómo usar ODROID para convertir cualquier monitor o TV en una gigante Tablet Android.



Juegos Linux: ¿No soy fanático de Comodoro?

⌚ September 1, 2018

Ahora sé que, por aquel entonces, Amstrad CPC, ZX Spectrum y Atari ST eran los grandes actores del mercado, aunque para mí lo fue y siempre los será Commodore



Campamento de Programación – Parte 3: Controlar un LED

⌚ September 1, 2018

Vamos aprender a controlar el LED azul de la parte delantera del ODROID-GO realizando pequeños ajustes en el LED con un simple GPIO de encendido/apagado, así como regular el brillo de 256 grados con PWM



Campamento de Programación – Parte 4: Leer el Estado de 12 Botones en ODROID-GO

⌚ September 1, 2018

En este artículo, aprenderemos cómo leer el estado de los botones con Arduino.



Informática de Alto Rendimiento en el Hogar: Primeros Pasos con ODROID y MPI

⌚ September 1, 2018

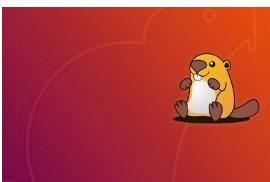
En este artículo, describiremos de forma general la instalación y configuración de un clúster básico “sin nodo principal” con el objetivo final de ejecutar programas en paralelo basados en el envío de mensajes, utilizando en particular el modelo de programación en paralelo Interfaz de paso de mensajes (MPI).



Home Assistant: Seguimiento de Personas con Wi-Fi Usando Kismet

⌚ September 1, 2018

La forma más sencilla de empezar a escuchar el espectro wifi es instalar kismet



Primeros Pasos con Ubuntu 18.04 en el ODROID-XU4: Una Guía para Principiantes

⌚ September 1, 2018

El ODROID-XU4 es básicamente un ordenador linux heterogéneo multi-proceso de ocho nucleos Linux. Al ofrecer soporte de código abierto, la placa puede ejecutar diversas versiones de Linux, incluido el último Ubuntu 18.04



Kali Linux 2018.2 en el ODROID-XU4

⌚ September 1, 2018

Kali Linux es uno de los mejores sistemas para pruebas de penetración. Está disponible para varios dispositivos basados en la tecnología ARM y por un corto periodo de tiempo también disponible para el ODROID-XU4 en el sitio web oficial de

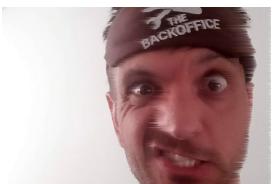
Kali.



Usando ODROID en Informática de Alto Rendimiento (HPC) – ARM Frente a la Competencia

⌚ September 1, 2018

Los núcleos ARM están siendo rápidamente utilizados por la comunidad científica de Datacenter porque los dispositivos ARM son más rápido con operaciones intensivas en matemáticas de punto flotante, por una décima parte de los costes de energía, y su hoja de ruta muestra que estar por venir una arquitectura con ➤



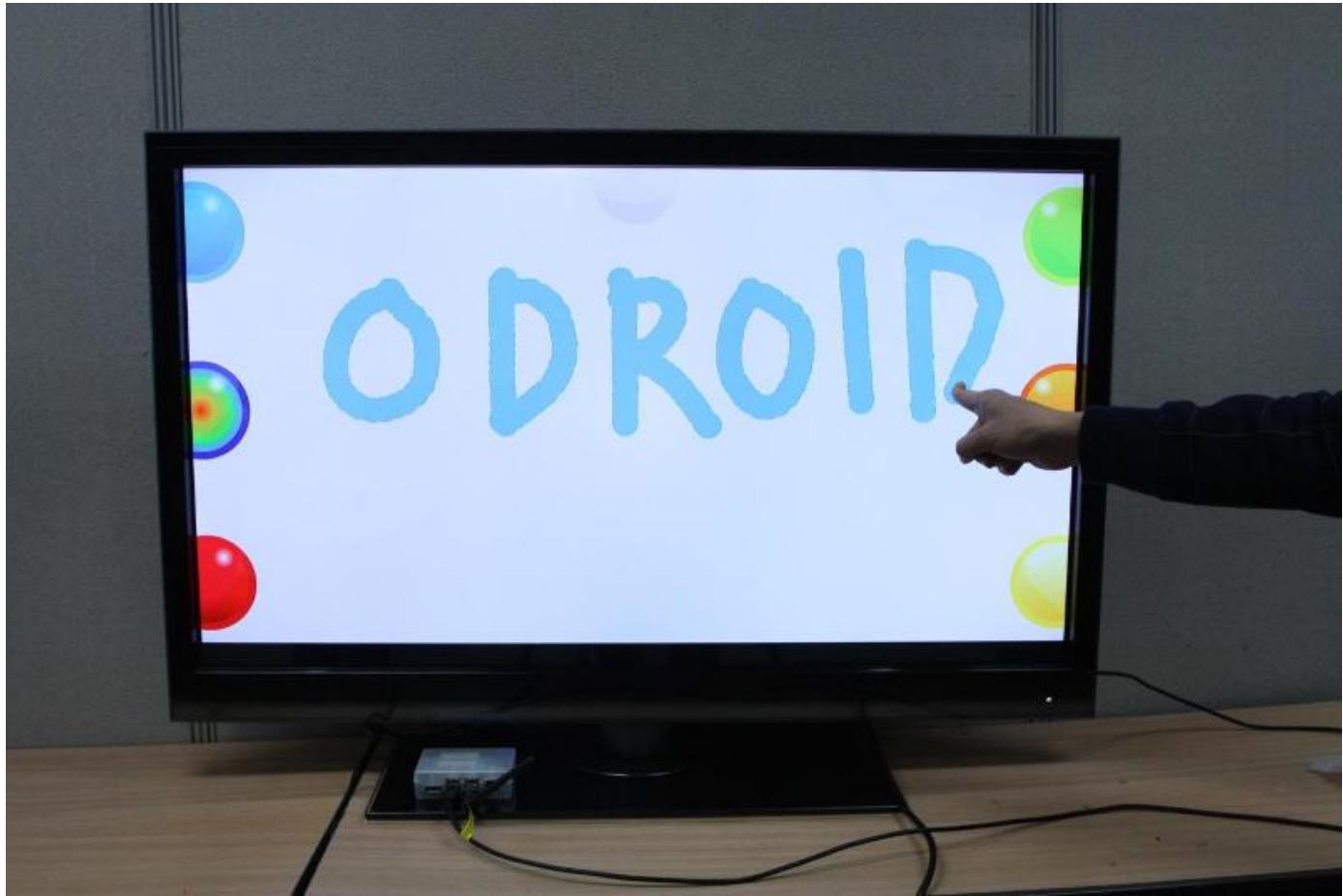
Conociendo un ODROIDian: Andrew Armstrong

⌚ September 1, 2018

Soy el CTO y cofundador de Cadonix, la primera solución de CAD eléctrica en la nube del mundo totalmente basada en el navegador. Tengo una amplia variedad de conocimientos que van desde la investigación científica, sistemas integrados, diseño electrónico, ingeniería de software y fabricación. Soy muy afortunado ya que todo ➤

Convierte un Monitor en una Gigante Tablet Android

© September 1, 2018 By Justin Lee and Charles Park ODROID-XU4, Mecaniqueo



Las pantallas táctiles son muy comunes en dispositivos como smartphones, consolas, ordenadores "todo-en-uno" y tablet. También juegan un papel importante en el diseño de aparatos digitales como los paneles de señalización, Terminales punto de venta (TPV), Dispositivos GPS, teléfonos móviles, videos juegos y libros electrónicos. Android, uno de los principales sistemas operativos para ODROID, cuenta con una interfaz de usuario muy intuitiva diseñada para usarse en pantallas táctiles. Este artículo describe cómo usar ODROID para convertir cualquier monitor o TV en una gigante Tablet Android.



Figura 1 – Una Gigante Tablet de 42 pulgadas con ODROID-XU

Tecnología Infrarroja vs Tecnología Capacitiva

Las pantallas táctiles usan principalmente tecnología por infrarrojos o tecnología capacitiva. Ésta última es la que está más entendida entre los smartphones y tablet pero también es la más cara, especialmente cuando el tamaño de la pantalla supera las 20

pulgadas. Una pantalla capacitiva sólo funciona con el dedo (o con un puntero) y puede presentar dificultades si el monitor no está correctamente montado. Teniendo en cuenta su facilidad de uso y bajo coste, vamos a utilizar la pantalla táctil por infrarrojos para este proyecto.

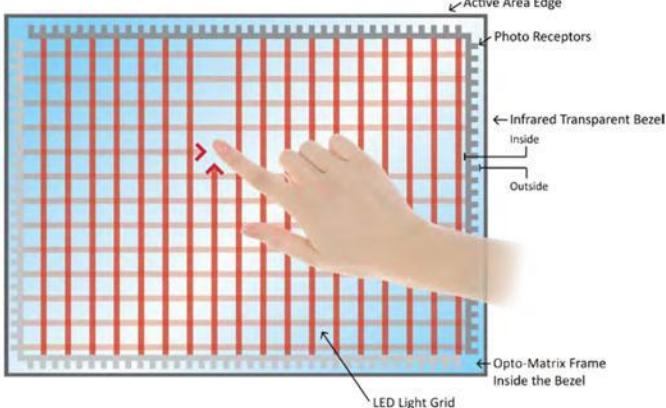


Figura 2: Principio de la pantalla táctil por IR (Infrarrojos)

Sistema de cuadrícula por Infrarrojos (IR)

Una pantalla táctil por infrarrojos utiliza una matriz LED de infrarrojos X-Y y fotodetectores en los bordes de la pantalla para detectar una interrupción en el patrón LED. Estas líneas LED se cruzan entre sí en vertical y horizontal, que ayuda a los sensores a detectar la ubicación exacta cuando se toca la pantalla. Una principal ventaja del sistema por infrarrojos es que puede detectar cualquier entrada, un dedo, un guante, un lápiz o un bolígrafo. Generalmente se utiliza en dispositivos al aire libre y terminales punto de venta en los que puede presentar cierta dificultad la utilización de una pantalla táctil.

A diferencia de las pantallas táctiles capacitivas, las pantallas táctiles por infrarrojos no requieren ningún patrón en el cristal, lo que mejora la durabilidad y claridad del sistema. Sin embargo, estas pantallas son más sensibles a la suciedad y al polvo. Además, no son aptas para superficies curvas y se pueden activar elementos erróneamente cuando el usuario desliza el dedo por la pantalla.

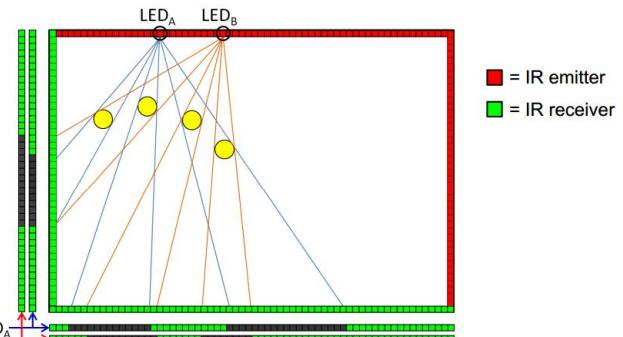


Figura 3 – Principio de la pantalla multitáctil por IR (Infrarrojos)

Cómo elegir una pantalla táctil por infrarrojos

Podemos montar nuestra propia pantalla táctil siguiendo cualquier tutorial que localicemos en internet, pero no es tarea fácil ya que el algoritmo multi-táctil es bastante complejo al igual que alinear correctamente el emisor/receptor IR, tal y como se muestra en la Figura 4.

A la hora de comprar una pantalla táctil por infrarrojos nos asegurarnos de su compatibilidad con Android. Debemos tener en cuenta los siguientes requisitos:

- 1) ¿Aparece la pantalla táctil en la lista de compatibilidad multitáctil de Linux?

<http://lii-enac.fr/en/architecture/linux-input/multitouch-devices.html>

En caso afirmativo, activar la pantalla táctil será relativamente fácil.

- 2) ¿La pantalla táctil es compatible con Windows 8?

En caso afirmativo, será necesario hacer algunas modificaciones en el Kernel y en el archivo de configuración de entrada.

- 3) ¿La pantalla dispone de un controlador específico para Android suministrado por el fabricante?

En caso afirmativo, debemos seguir sus instrucciones

En el mercado coreano y chino no hay disponibles pantallas compatibles con Linux. Algunos fabricantes de china ofrecen drivers específicos que en la práctica no son muy útiles. La mejor alternativa es hacerse con una pantalla táctil que admita Windows 8 y el estándar Plug & Play.

Es muy importante comprobar si la pantalla táctil es realmente compatible con Windows 8. Los dispositivos Plug & Play no requieren de un driver independiente para ser instalado en un PC con Windows. Si la pantalla táctil necesita un driver específico, nos está indicando que no es compatible al 100% con Windows 8 y posiblemente presentará problemas con Android.

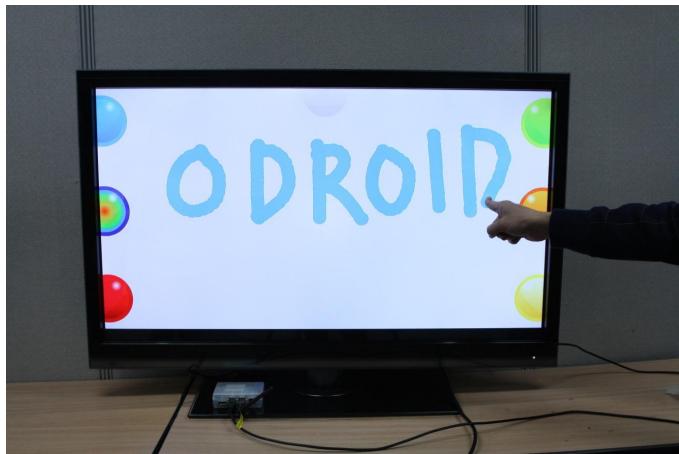


Figura 4 – Pantalla táctil de 42 puladas conectada a un ODROID-XU

Conecrtar la pantalla táctil a Android

Debemos seguir cuatro pasos para usar una pantalla táctil con Android:

Conseguir el número de identificación (ID) del proveedor y del producto desde la interfaz USB de la pantalla táctil

Modificar los archivos hid-ids.h y hidmultitouch.c, localizados en el directorio kernel/drivers/hid/

Compilar el Kernel con la opción multi-táctil habilitada, y trasladar la imagen del kernel al ODROID

Crear un archivo IDC (Configuración del dispositivo de entrada).

Paso 1: Consultar valores VID y PID

Conecta la pantalla táctil a cualquier PC con Linux. A continuación, localiza el valor VID y PID escribiendo "lsusb" en el terminal tal y como muestra la figura 5

```
odroid@odroid:~$ lsusb
Bus 002 Device 068: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 3.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
odroid@odroid:~$ lsusb
Bus 002 Device 073: ID 03fc:05d8 Elitgroup Computer Systems
Bus 002 Device 068: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
odroid@odroid:~$
```

Figura 5 – Leyendo el VID y el PID de la pantalla táctil utilizando un PC con Linux

Para determinar qué entrada está asociada a la pantalla táctil, debemos comparar la lista de dispositivos antes y después conectar la pantalla. La nueva entrada que aparezca corresponderá al dispositivo de pantalla táctil.

Durante nuestro proyecto, cuando conectamos la pantalla táctil de 23 pulgadas, aparecido la siguiente entrada VID:03FC, PID:05D8 que corresponde a una pantalla táctil fabricada por Elitegroup Computer Systems. Con una pantalla táctil de 42 pulgadas aparece la entrada VID:1870, PID:0119 que corresponde a una pantalla táctil de Nexio Co., Ltd. Cada pantalla conectada mostrará diferente información del fabricante y diferentes valores VID Y PID.

Paso 2: Modificar archivos hid-ids.h y hidmultitouch.c

Tras descargar el correspondiente código fuente del kernel de Android desde dn.odroid.com, entra en el directorio kernels/drivers/hid/, abre el archivo hid-ids.h y añade los valores VID y el PID al final del mismo.

```

#define USB_VENDOR_ID_PRIMAX 0x0461
#define USB_DEVICE_ID_PRIMAX_KEYBOARD 0x4e05

/* Elitegroup Computer Systems */
#define USB_VENDOR_ID_ELITEGROUP 0x03fc
#define USB_DEVICE_ID_ELITEGROUP_TOUCH 0x05d8

/* NEXIO Co., Ltd */
#define USB_VENDOR_ID_NEXIO 0x1870
#define USB_DEVICE_ID_NEXIO_TOUCH 0x0119

#endif

```

Figura 6 – Ejemplo de valores PID y VID añadidos a kernels/drivers/hid/hid-ids.h.

Además, añade el nuevo ID al archivo hidmultitouch.c que debe escribirse con una estructura específica hid_device_id mt_devices.

```

static const struct hid_device_id mt_devices[] = {

/* Elitegroup Computer Systems */
{ .driver_data = MT_CLS_DEFAULT,
  HID_USB_DEVICE(USB_VENDOR_ID_ELITEGROUP,
    USB_DEVICE_ID_ELITEGROUP_TOUCH), }

/* NEXIO Co., Ltd */
{ .driver_data = MT_CLS_DEFAULT,
  HID_USB_DEVICE(USB_VENDOR_ID_NEXIO,
    USB_DEVICE_ID_NEXIO_TOUCH), }

/* 3M panels */
{ .driver_data = MT_CLS_3M,
  HID_USB_DEVICE(USB_VENDOR_ID_3M,
    USB_DEVICE_ID_3M1968) },
{ .driver_data = MT_CLS_3M,
  HID_USB_DEVICE(USB_VENDOR_ID_3M,
    USB_DEVICE_ID_3M2256) },
{ .driver_data = MT_CLS_3M,
  HID_USB_DEVICE(USB_VENDOR_ID_3M,
    USB_DEVICE_ID_3M3266) },

```

Figura 7 – Añadiendo el ID de la pantalla táctil a kernels/drivers/hid/hid-multitouch.c

Paso 3: Compilar el kernel con la opción HID-MULTITOUCH activada

Escribe “make menuconfig” para configurar el kernel, dirígete a Device Drivers > HID Devices > Special HID drivers > HID Multitouch panels y seleccionarlo para activar el driver (*), tal y como muestra la figura 8

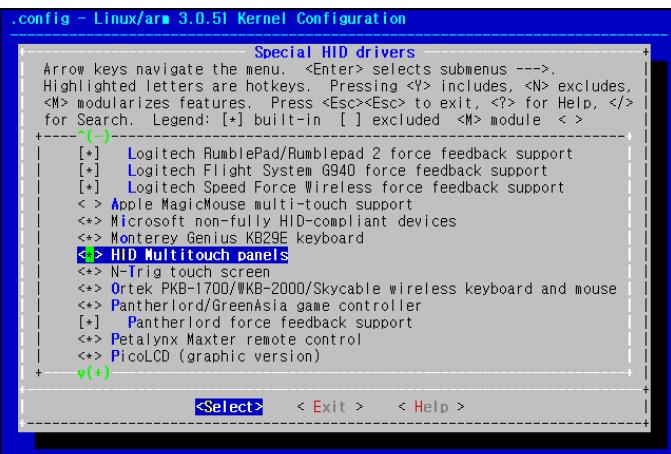


Figura 8 – Configurando la pantalla táctil con el driver activado en la configuración del Kernel

Selecciona las dos opciones que aparece a continuación ya que es un driver integrado, indicadas con un asterisco (*)

Device Drivers -> Input device support -> touch screens -> USB touch screen Driver

Device Drivers -> Input device support -> touch screens -> GeneralTouch touch screen device support

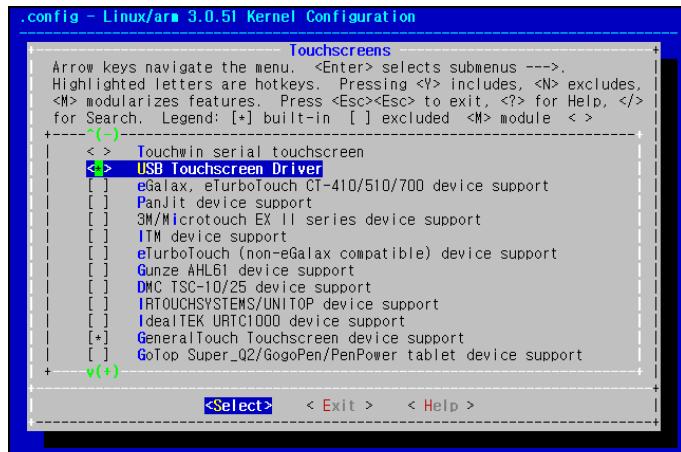


Figura 9 – Configurando la pantalla táctil con driver integrado en la configuración del Kernel

Guarda la configuración del kernel y compíalo para hacer una zImage.

Traslada la zImage a tu ODROID a través del protocolo fastboot en el u-boot.

Paso 4: Crear un archivo IDC (Configuración del dispositivo de entrada)

Si no se crea correctamente el archivo IDC, la resolución de la pantalla táctil no coincidirá con la resolución HDMI

El archivo IDC es un archivo de texto plano en formato Unix.

Aquí tienes un ejemplo. Puedes usar este archivo tal cual.

```

touch.deviceType = touchscreen
touch.orientationAware = 1
device.internal = 1

keyboard.layout = qwerty
keyboard.characterMap = qwerty2
keyboard.orientationAware = 1
keyboard.builtIn = 1
cursor.mode = navigation
cursor.orientationAware = 1

```

El nombre del archivo debe ser

Vendor_xxxx_Product_yyyy.idc (xxxx: ID Proveedor, yyyy: ID Dispositivo). Yo he creado dos archivos para

Elitegroup y Nexio. El nombre del archivo es sensible a las mayúsculas.

Vendor_03fc_Product_05d8.idc
y
Vendor_1870_Product_0119.idc

Copia los archivos IDC a tu ODROID utilizando los siguientes comandos:

```
adb remount  
adb push Vendor_03fc_Product_05d8.idc  
/system/usr/idx/.  
adb push Vendor_1870_Product_0119.idc  
/system/usr/idc/.  
adb reboot
```

Cómo fijar el panel táctil de 42 pulgadas a la pantalla de TV .

Prepara la pantalla táctil



Figura 10 – Preparando la pantalla táctil

Coloca cinta de doble cara sobre el marco de la pantalla táctil

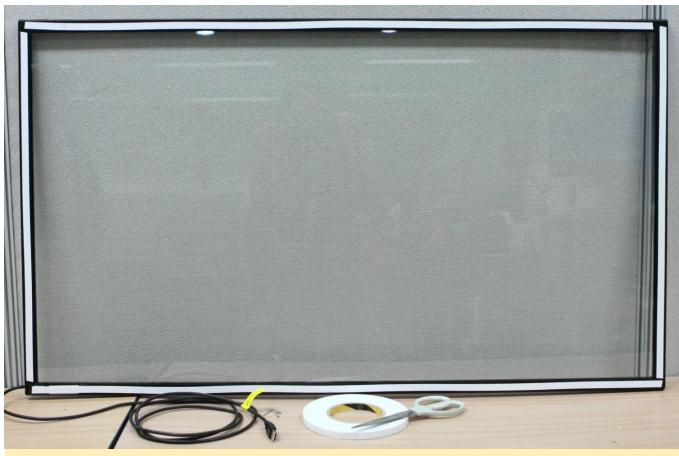


Figura 11 – Colocando la cinta de doble cara en el monitor

Acopla con cuidado la pantalla táctil al televisor y alinéala.



Figura 12 – Alineando la panta táctil con el monitor

Prueba la pantalla táctil. Nuestra pantalla llegó a detectar hasta 6 puntos.



Figura 13 – Probando la pantalla táctil

Cómo fijar el panel táctil de 23 pulgadas a la pantalla del monitor.

Esta pequeña pantalla de 23" viene con 4 correas de velcro que facilitan bastante su montaje.



Figura 14 – Panel táctil de 23 pulgadas con correas de velcro



Figura 15 – Jugando a Fruit Ninja en la pantalla táctil de 23" con ayuda de mi dedo

Conclusión

Además de su uso personal y como plataforma de juegos, ODROID es ideal como dispositivo informático para quioscos, señalización digital e investigación de interfaces entre otras aplicaciones, gracias a su alto potencial y bajo coste. Al tratarse de una plataforma abierta permite modificaciones como esta pantalla táctil.

Puedes ver los resultados de este proyecto en http://youtu.be/b8_cV_NeWQ8, y para tener más acción Minecraft con la gigante tablet Android, visita <http://www.youtube.com/watch?v=HDsnuxchxtU>.

Juegos Linux: ¿No soy fanático de Comodoro?

⌚ September 1, 2018 📝 By Tobias Schaaf 🏷 Juegos, ODROID-C2, ODROID-XU4



En recientes artículos, he dejado plasmado mi opinión sobre Nintendo y Sega. Me considero más fan de Sega que de Nintendo, aunque reconozco que disfruto bastante con la Game Boy Advance y la DS de Nintendo.

Me he criado en la Alemania del Este durante los años 80 y 90, por aquella época las consolas no me atraían demasiado. De hecho, aquí en Europa, los "micro-ordenadores" eran bastante más comunes. Esto también explica por qué yo, personalmente, prefiero modificar/trastear en lugar de simplemente encender un sistema y trabajar dentro de sus limitadas posibilidades. Esto también significa que me siento más cómodo jugando a los juegos en un PC con un ratón y un teclado que con un mando de consola. Ahora sé que, por aquel entonces, Amstrad CPC, ZX Spectrum y Atari ST eran los grandes actores del mercado, aunque para mí lo fue y siempre los será Commodore, con la que crecí. Primero con mi C64 "Brotkasten". (o "Breadbox"), y más tarde con

diferentes modelos de Amiga. En este artículo, me gustaría hablar de mis experiencias durante aquella época, y cómo me siento hoy día en relación a comodoro.

La C64 "Brotkasten"

El C64 fue el primer micro-ordenador que tuve de niño. Más exactamente, lo tenía mi padre y yo jugaba con él. Aunque de esto ya hace muchos años, tengo muy buenos recuerdos de este dispositivo.



Figura 1 – El Commodore 64 "Brotkasten"

Si no estás familiarizado con la Commodore 64, es probable que no sepas que su nombre deriva de los 64 KB de RAM con los que tenías que trabajar. Una vez cargado el sistema operativo BASIC, te quedaban 38911 Bytes de RAM libre, que es lo que te quedaba para ejecutar todos tus programas y juegos. Tenía una velocidad de cálculo de aproximadamente 1MHz y ofrecía 16 increíbles colores. Un buen potencial, probablemente comparable a la NES.

La Commodore 64 fue probablemente la más famosa por su impresionante potencial de sonido por aquel entonces. Aquí tienes un ejemplo de la música de Last Ninja https://www.youtube.com/watch?v=CoGFV_xxR64. Había, y sigue existiendo, una gran escena "demo" en torno a la C64, con usuarios creando música y vídeos con impresionantes gráficos.

Recuerdo haber jugado a un juego llamado Mafia que probablemente sea mi mejor recuerdo de la C64. Este juego tenía unos gráficos muy simplistas, pero el sistema de juego era muy divertido. Se jugaba con un teclado en lugar de un joystick, ya que a menudo necesitabas introducir números y otras cosas.



Figura 2 – Mafia en la C64

En Mafia podías hacer muchas cosas: comprar o robar coches, incluso robar las carteras de la geste. Podías visitar el casino o alquilar un lugar para vivir. Podías intentar presionar a los comercios para que te pagasen dinero a cambio de protección o preguntarles si tenían un trabajo para ti (como sicario). Podías trabajar como portero; intentar robar un banco; montar tu propia pandilla; pasar alcohol de contrabando; comprar armas; entrenarte a ti mismo o a los miembros de tu pandilla, etc.

Empiezas como un personaje desconocido con el que vas ascendiendo poco a poco a través de rangos al margen de la ley. Si hacías "malas acciones", podías conseguir un trabajo de mayor categoría, como robar bancos o pasar alcohol de contrabando. También te permitía contratar a más miembros de pandillas para realizar trabajos de mayor envergadura.

El juego tenía un error muy práctico que te permitía ir al casino y jugar con dinero negativo. Si realizabas una apuesta de -1,000,000\$ y perdías, te volvías instantáneamente rico, lo cual hacía que el juego fuese mucho más fácil, especialmente para un niño tan joven como yo por aquel entonces. También disfrutaba con un juego de texto llamado Hotel por un error similar. Construías un número negativo de piscinas en tu primer hotel y te hacías instantáneamente rico, aunque ello significaba que nunca podría alcanzar el nivel 5 estrellas, puesto que tus hoteles contaban con un número negativo de piscinas.

La C64 te ofrecía más que los simples malos gráficos y la penosa programación de estos dos juegos. De hecho, había muchos juegos impresionantes. Juegos como R-Type, Maniac Mansion, Turrican, The Last Ninja, IK + (International Karate Plus), Creatures, Boulder Dash, las infames Great Giana Sisters o las series deportivas como Summer Games, Winter Games y California Games.

Tengo muy buenos recuerdos de muchos de estos juegos. Mi madre, que nunca había jugado a ningún juego, llegó a jugar a Boulder Dash, lo cual fue increíble. Una vez, jugando a Maniac Mansion, no sabía que el juego tenía la posibilidad de guardarlo e intentamos terminarlo en un día, empezando una y otra vez desde el principio. Turrican era un sorprendente juego de plataformas/disparos con algunos elementos al estilo Metroid. IK+ es considerado uno de los mejores juegos de todos los tiempos, y puesto que Nintendo no quería tener Super Mario en ninguna otra plataforma, disponemos de The Great Giana Sisters en su lugar. The Great Giana Sisters fue un engaño un tanto descarado, aunque asombroso, que más tarde estuvo disponible para NDS y otras plataformas.



Figura 3 – R-Type para la C64



Figura 4 – Zak McKracken en la C64. Estas primeras aventuras eran bastante difíciles



Figura 5 – IK+: uno de los mejores simuladores de combate que existen



Figura 6 – Last Ninja 2: Impresionantes Gráficos, música y jugabilidad, ¡aunque condenadamente difícil!



Figura 7: Coleccionar diamantes nunca fue tan divertido como en Boulder Dash

En definitiva, la C64 fue un impresionante ordenador doméstico. No es de extrañar que, en un momento dado, fuese el ordenador doméstico más vendido de todos los tiempos. Aunque era pequeño, sigo recordando los buenos momentos que pasaba jugando en él. Aunque las nuevas versiones pueden tener mejores gráficos, a veces el sentimiento de lo original y atípico te viene a la memoria. Esto es precisamente lo que me ocurre con la C64 y algunos de sus juegos, como Boulder Dash. Hay otras versiones de este juego con gráficos mejorados, pero en mi opinión son horribles. La única y verdadera versión de este juego es la de la vieja C64 (y tal vez algunos clones de otros sistemas).

Mi época con Amiga

Aunque la C64 fue una magnífica obra con una gran flexibilidad, el Amiga supuso un increíble salto en cuanto a tecnología. Fue un paso adelante en la forma en la que lo hizo la NES cuando dio el salto a la SNES.

La historia que hay detrás de Amiga es larga y emocionante. Algunos incluso la llamarían desgarradora, con sus muchos altibajos, especialmente hacia el final. Si te interesa la historia de Amiga, te sugiero que veas la película "From Bedrooms to Billions-The Amiga Years". Es un buen documental sobre cómo Amiga cobró vida y fascinó a la gente. Si nunca has tenido un Amiga, puede que te sea un tanto difícil entender toda su esencia.

El Amiga fue una especie de "el" PC de finales de los 80 y principios de los 90. Los PC compatibles con IBM eran muy costosos. Apple tampoco era barato y apenas habían hecho cambios en bastante tiempo. Los ordenadores domésticos Atari todavía seguían siendo bastante limitados. De repente, apareció un ordenador con un cierto potencial a un precio bastante razonable. El Amiga soportaba la multitarea y se podía elegir entre 4096 colores, los mismos que en el caso de la SNES en su modo 7. Los modelos posteriores llegaron a ofrecer más de 16 millones de colores (24 bits), audio estéreo de 4 canales, discos estándar como medio de almacenamiento y mucho más.

Una de sus características más destacadas fue el Workbench. Hoy día, la mayoría lo llamaría "Escritorio".

El primer Workbench que llegó a usarse con cierta frecuencia fue la versión 1.3, que estaba ampliamente disponible en torno a 1987 en los primeros Amigas. Aunque visualmente no era muy diferente de la versión 1.0 (ya lanzada en 1985), empezó a usarse con una mayor frecuencia. Casi al mismo tiempo, Windows 2.0, que todavía se ejecutaba en DOS, aparecía en los PCs compatibles con IBM.

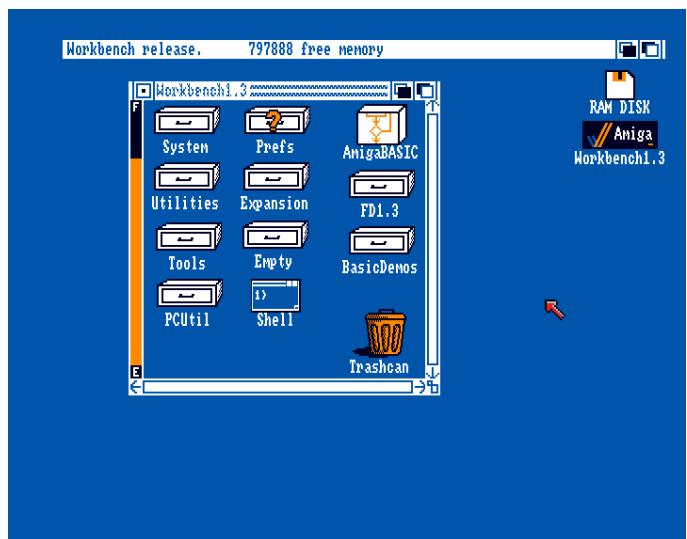


Figura 8 – Workbench 1.3 en un Amiga ejecutándose desde un HDD

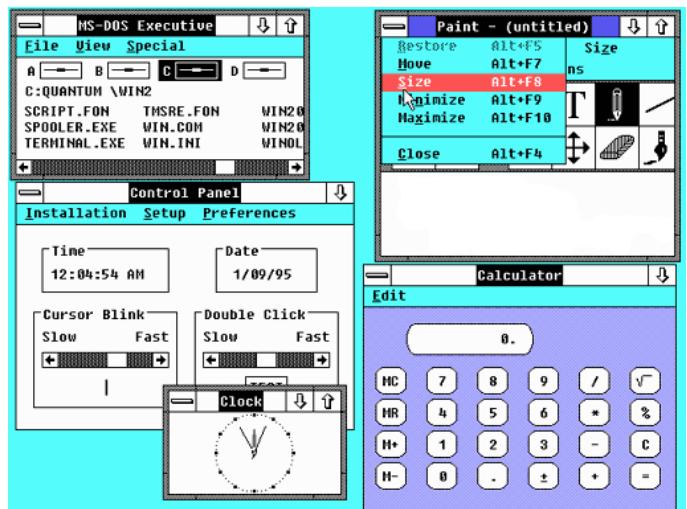


Figura 9 – Windows 2.0 en DOS para PC IBM

Fue en 1990 o 1991 cuando se lanzó Workbench 2.0 con un cambio importante en la apariencia que hizo que el Amiga pareciera un decente ordenador de escritorio. Se necesitaba una nueva versión del Kickstart (era como la BIOS del sistema) para ejecutarse, aunque se podía ejecutar en los Amigas más antiguos, siempre que tuvieran el Kickstart más reciente. Kickstart era en realidad una ROM que podía sustituirse en la placa base del sistema.

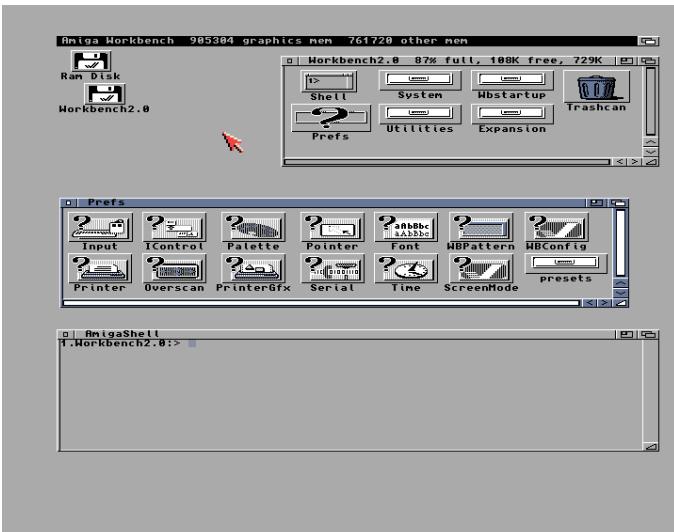


Figura 10 – Un Workbench 2.0 más optimizado para el Amiga



Figura 12 – Workbench 3.0 con imágenes de fondo para cada carpeta y escritorio

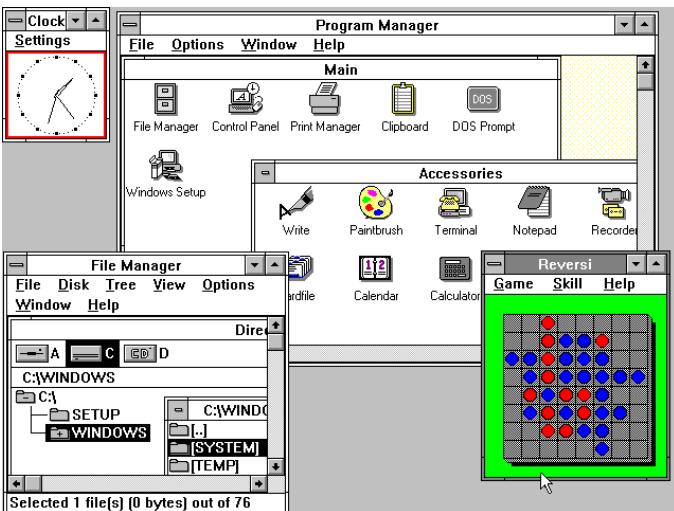


Figura 11 – Windows 3.0 para PCs con DOS

Un año después, en 1992, con las versiones más recientes de Amiga y Kickstart 3.0/3.1, se presentó una nueva versión de Workbench. Workbench 3.0 una vez más mostró gráficos algo más mejorados, pero su mayor avance radicaba en la funcionalidad.

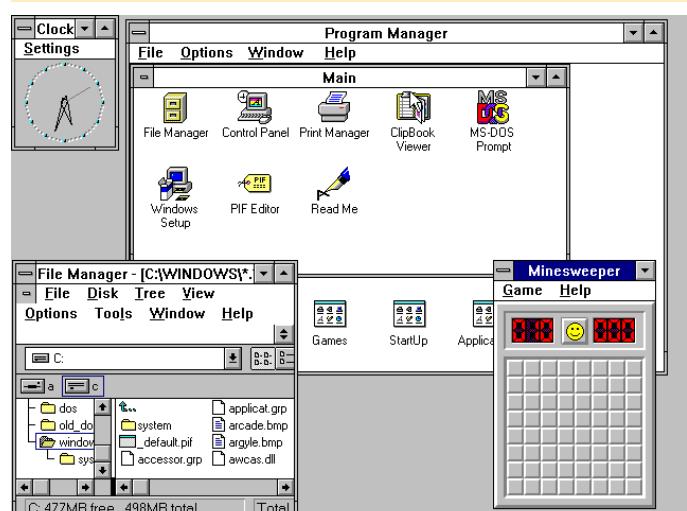


Figura 13 – Windows 3.1 para DOS

Workbench se podía ejecutar directamente desde discos o se podía instalar en un disco duro, si estaba disponible. Tenía características muy interesantes, como un menú contextual (click con el botón derecho) para operaciones de archivos. Cada carpeta se podía configurar de forma individual y los cambios que quedaban guardados para cada carpeta.

Si eres un gurú de los PCs, probablemente hayas oído hablar de algo llamado "Ram Disk". Es una técnica que usa parte de tu RAM como un disco virtual para poner datos, mejorando la seguridad o, más probablemente, la velocidad.

Por ejemplo, hay quien usa un Ram Disk para colocar los archivos temporales del navegador en una carpeta temporal, primero para eliminar automáticamente los archivos del Ram Disk cuando se reinicia el PC para no corromper el disco; y segundo, para acelerar los

tiempos de carga de dichos archivos, ya que la RAM es mucho más rápida que cualquier HDD (e incluso los SSD).

El Amiga ya contaba con esto como una función integrada en Workbench. Si tenías suficiente RAM, podías copiar un juego desde un disquete directamente al RAM Disk e iniciar lo desde allí, lo que significaba que nunca tenía que cargar el juego desde el disco, sino que podía cargar todos los datos desde la memoria, si tenías suficiente para ello.

Aunque Workbench original empezó a mejorar su apariencia en la versión 2.0, todavía quedaban más proyectos por llegar.

MagicWB mejoró enormemente la apariencia de cualquier Workbench versión 2.0 o superior, lo que permitió que tu Amiga de 1992 tuviese el siguiente aspecto:

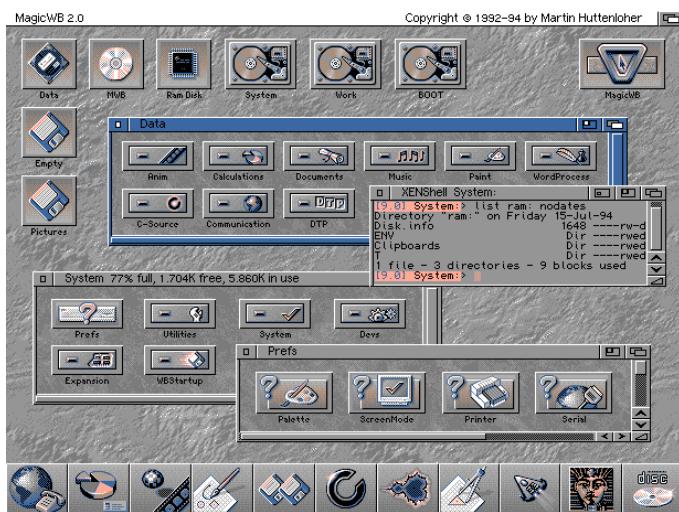


Figura 14: MagicWB mejoró la apariencia del Workbench al agregar nuevos iconos, fondos y combinaciones de colores.

El Amiga era una impresionante pieza de hardware en aquel momento, equiparándose al moderno hardware de los PCs de IBM, Apple y Atari. El Amiga fue utilizado por muchos y diferentes artistas, ya que el mod tracker (también conocido como "Protracker" o "Fast Tracker") se utilizaba para crear impresionante música con el Amiga.

Ejemplos: <https://www.youtube.com/watch?v=ecIMFa0mD1c> <https://www.youtube.com/watch?v=p9zmLQGBTIw>

Deluxe Paint para el Amiga estaba muy por delante de su tiempo y fue utilizado por muchos artistas para

crear imágenes o manipular fotos (https://www.youtube.com/watch?v=R07JURHm_jk)

De hecho, el Amiga tenía un rendimiento gráfico tan potente para su época que se llegó a utilizar en la producción de películas y televisión.

Si alguna vez has oído hablar de Babylon 5, este serie de televisión se benefició bastante del Amiga: <https://www.youtube.com/watch?v=iXmSA18cfgA>

De hecho, Video Toaster, que podría usarse de muchas y diferentes maneras, revolucionó la industria: <https://www.youtube.com/watch?v=6eNYj-Chkxw>

Después de leer que con un Amiga se puede hacer cuchas cosas creativas, uno se puede preguntar ... ¿De qué está hablando? Pensaba que estábamos hablando de juegos

Pues bien, a pesar de todo lo que se podía hacer con el Amiga, aún seguía usándose principalmente para juegos. Existen casi 1800 juegos para Amiga, algo comparable con los lanzamientos mundiales de SNES (solo 720 de los lanzamientos de SNES fueron publicados en los Estados Unidos).

Con esta impresionante colección de juegos, el Amiga tenía juegos para todos los públicos: juegos de rol, aventuras, carreras, combates. Todos los géneros de juego existían para el Amiga. La mayoría de mis recuerdos de la infancia hasta cumplir los 14 años se centra en el Amiga y sus juegos. Existían tantos títulos impresionantes para el sistema, que ni siquiera sabría por dónde empezar.

He jugado a multitud de juegos de aventuras en el Amiga, como Monkey Island 1 y 2, Indiana Jones 3 y 4, Beneath a Steel Sky (que venía con quince discos), Simon the Sorcerer, Operation Stealth, Dune, Loom y muchos más.



Figura 15 – La serie Monkey Island es una serie de aventuras extremadamente divertida. Si nunca has llegado a jugarla, es que realmente te has perdido parte de la historia de los juegos.



Figura 16 – Operation Stealth fue una detallada y compleja aventura de agente secreto en el Amiga

Muchos juegos de Amiga fueron superiores a sus homólogos para PC. Especialmente en los primeros años, el Amiga superó a otros sistemas con mejores gráficos y mejor música. Muchos juegos que luego ejecuté en el PC me parecieron un tanto extraños por esta razón, ya que era conocedor de que eran "mejores" en el Amiga que cuando los reproduces en el PC. El Amiga también contaba con muchas versiones de antiguos juegos de la C64 con gráficos mejorados.

Me resulta realmente difícil decantarme por algunos juegos de Amiga, ya que me gustan muchos de aquella época. Rodland, un pequeño y divertido juego de plataformas. Lost Patrol, en este juego debes guiar a un grupo de soldados estadounidenses varados en Vietnam hasta una base militar. Este juego es muy difícil, pero está magníficamente renderizado y es bastante complejo. La música solo ya merece un premio; es una de las mejores bandas sonoras que he escuchado en un videojuego. Persian Gulf Inferno, un juego de plataformas/tirador en el que tienes que liberar a una plataforma petrolera de terroristas del

Medio Oriente. Rainbow Island, del cual se dice que es la mejor versión de todos en Amiga. Arabian Nights, un juego de plataformas donde, aunque no está relacionado con la película de Disney, juegas con un personaje del tipo Aladdin, también tiene buena música y gráficos para ser un título de aquella época.



Figura 17 – First Samurai en el Amiga. ¡Me encantaba balancear esa espada!

First Samurai era un juego de plataformas que incluye elementos de combate, excelentes gráficos, sonidos e interpretaciones de voces muy claras. Fire and Ice era un juego de plataformas aún más impresionante, y The Chaos Engine (también conocido como Soldiers of Fortune) era un increíble juego de correr y disparar cooperativo. Flashback era un juego de aventuras de acción/plataformas, que no le gustó demasiado a la gente, pero que yo la disfruté bastante en el Amiga.



Figura 18 – Flashback fue exportado a casi todos los sistemas de aquella época, pero la versión de Amiga es la que jugué y terminé.

He jugado a más de cien juegos para Amiga, ya que tenía algunos títulos bastante sorprendentes. Hasta que conseguí mi primer PC, jugué a mi incondicional Amiga, pasando por diferentes modelos a lo largo del tiempo (mi padre los tenía y yo jugaba con ellos).

Empecé con el modelo más común, el Amiga 500, al que le conseguimos un disco duro más tarde. También nos hicimos con un Kick-Switch, que nos permitía cambiar entre Kickstart 1.3 y 2.04, y con la expansión de RAM.

Un tío mío tenía un Amiga 600 pero un poco mediocre. Tenía un pequeño y original disco duro Amiga integrado en el sistema, e incluso recurriendo a la compresión, el disco duro estaba continuamente lleno.



Figura 19: Moonstone era un juego muy divertido, pero a la vez muy sangriento



Figura 20 – Wing Commander para Amiga despertó mi pasión por los simuladores espaciales

Más tarde, conseguimos un Amiga 1200. Se trataba de un sistema nuevo y mejorado, que usaba Kickstart 3.1 y un chip AGA, que era un chip de gráficos mejorado que permitía 16,7 millones de colores en lugar de los 4,096 colores del Amiga 500. También, durante poco tiempo llegué a tener un Amiga CD32, la

primera consola de juegos basada en CD de 32 bits, pero, aunque el formato de CD era prometedor, la mayoría de los juegos eran simplemente versiones lineales de los títulos de Amiga 1200.

El Amiga era fascinante. Aunque jugué a muchos juegos con él, apenas llegue a arrancar la superficie de la inmensa colección de juegos que había disponible.

Reflexiones finales

Entonces, vuelvo a la misma pregunta que he hecho a lo largo de esta serie con respecto a otros sistemas: ¿soy un fanático de Commodore?

Si creciste en Europa a finales de los 80 y en los 90 y no sabías lo que era un C64 o un Amiga, probablemente vivías debajo de una roca. Siempre hablaré con orgullo de mi Amiga y de los juegos que tuve. Es posible emular los juegos de C64 y Amiga en ODROID, pudiendo revivir tus recuerdos de esa época.

Pero últimamente me dado cuenta de que, si quisiera recomendar juegos a alguien que nunca ha tenido una C64 o un Amiga, y que no ha crecido con estos sistemas, es difícil hacerle sentir lo mismo que se siente cuando eras un niño. Muchos de los juegos existentes en el C64 y Amiga fueron exportados a otros sistemas, y aunque el Amiga es probablemente la mejor plataforma para algunos de estos juegos, no a todos les gusta el tipo de juegos de estrategia, simuladores o juegos de aventuras que hicieron tan increíbles estos sistemas.

De hecho, si quisiera revivir mis juegos de aventuras, es muy probable que vaya directamente a ScummVM, donde puedo jugar a estos juegos sin emulación. Si quiero jugar a Dune 2, la serie XCOM, preferiría elegir Dune Legacy o OpenXCom, versiones mejoradas de Linux de estos juegos para ODROID. Si observas la biblioteca de juegos, encontrarás muy pocos IP que sobrevivieran al paso del tiempo. No encontrarás títulos conocidos como Super Mario Bros., The Legend of Zelda, o Sonic the Hedgehog en sistemas Commodore.

Todavía hay algunos juegos que diría “¡Oye, juegalo en el Amiga!” Estos son grandes juegos que no existen en ningún otro sistema, aunque la lista es muy corta. El

Amiga y el C64 son sistemas geniales con mucho encanto retro, pero me doy cuenta de que me atraen más los juegos SegaCD, Sega Saturn o incluso Master System que los de Amiga o C64. Tengo pensado regresar al Amiga para reproducir algunos juegos, o incluso probar algunos nuevos, pero por más que sea un buen recuerdo de mi infancia, no puedo decir que me dentanto por la Commodore. Por lo tanto, por muy difícil que me sea admitirlo, debo decir que, No, ya no soy fanático de Commodore.

Campamento de Programación – Parte 3: Controlar un LED

© September 1, 2018 By Justin Lee Mecaniqueo, Tutoriales, ODROID-GO



Vamos aprender a controlar el LED azul de la parte delantera del ODROID-GO realizando pequeños ajustes en el LED con un simple GPIO de encendido/apagado, así como regular el brillo de 256 grados con PWM. Antes de empezar, asegúrate de haber leído las dos primeras guías en https://wiki.odroid.com/odroid_go/arduino/01_arduino_setup y https://wiki.odroid.com/odroid_go/arduino/02_hello_world. Además, la guía original la puedes encontrar en la página de la wiki de ODROID-GO en https://wiki.odroid.com/odroid_go/arduino/03_blue_led_and_pwm.

Hacer parpadear del LED azul



Figura 01 – El LED azul parpadeando

Vamos a hacer que el LED parpadee continuamente. Primero, abrimos un nuevo esquema presionando el atajo CTRL-N. Controlar el LED es muy fácil, el número de pin para el LED es 2, así que vamos a definirlo con un preprocesador. A continuación, tenemos que fijar el pin en modo salida. Usa la función pinMode () para hacer esto.

```
#define PIN_BLUE_LED 2

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

pinMode(PIN_BLUE_LED, OUTPUT);
}

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

Ahora el pin está listo para ser utilizado. Puedes ajustar el nivel de la señal del pin utilizando la función digitalWrite (). Esta función tiene que estar en loop ()

para que se ejecute una y otra vez. La función delay() se utiliza para bajar la velocidad del parpadeo.

```
#define PIN_BLUE_LED 2

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

pinMode(PIN_BLUE_LED, OUTPUT);
}

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

digitalWrite(PIN_BLUE_LED, HIGH);
delay(500);
digitalWrite(PIN_BLUE_LED, LOW);
delay(500);
}
```

Presione CTRL-U para compilar y cargar el esquema. Luego, podrás ver el LED azul parpadeando.

Dar al LED un efecto progresivo



Figura 02 – el LED iluminándose progresivamente

Ajustando el valor de salida analógico, podemos hacer que el LED se ilumine de forma progresiva simulando el efecto de la "respiración". Esta técnica se conoce como PWM. Arduino nos ofrece soluciones que pueden ayudar a controlar los pines GPIO, así como las características PWM. Por lo general, estas funciones PWM son `analogRead()` y `analogWrite()`, pero todavía no están disponibles en ESP32, de modo que debemos usar las funciones `ledcRead()` y `ledcWrite()` para controlar los LED.

La forma de controlar los valores PWM en un ESP32 es diferente a otras placas Arduino. Estas funciones utilizan la función de hardware LED PWM de ESP32 que se encuentra en `ledc.h`, cabecera de la función de control LED. En ESP32, el LED PWM está compuesto por 16 canales independientes, y podemos configurarlos con un ciclo de trabajo partiendo de un periodo de onda y resolución.

Primero, debemos elegir un canal para conectar el LED. Los canales del 0 al 7 están disponibles, usaremos el canal 1 en este ejemplo. Define el canal y el LED azul a través de una macro Preprocesador.

```
#define PIN_BLUE_LED 2
#define PWM_CHANNEL 1

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

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

Set the pin mode of the LED to output:

```
#define PIN_BLUE_LED 2
#define PWM_CHANNEL 1

void setup() {
// put your setup code here, to run once:
pinMode(PIN_BLUE_LED, OUTPUT);
}

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

Conecta el pin LED al canal que hemos definido. A continuación, configura el canal para que funcione a 12 kHz con una resolución de 8 bits.

```
#define PIN_BLUE_LED 2
#define PWM_CHANNEL 1

void setup() {
// put your setup code here, to run once:
pinMode(PIN_BLUE_LED, OUTPUT);
ledcSetup(PWM_CHANNEL, 12000, 8);
}

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

Por último, añade el código del efecto "respiración" a la función `loop()`. Definimos una variable llamada `pinVal` como una variable global para evitar la repetida asignación de nueva memoria en la función `loop()`. El carácter `pinVal` no tiene firma, ya que el valor de PWM sólo abarca desde 0 a 255.

```
#define PIN_BLUE_LED 2
#define PWM_CHANNEL 1

unsigned char pinVal = 0;

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

pinMode(PIN_BLUE_LED, OUTPUT);
digitalWrite(PIN_BLUE_LED, HIGH);

ledcAttachPin(PIN_BLUE_LED, PWM_CHANNEL);
ledcSetup(PWM_CHANNEL, 12000, 8);
}

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

for (; pinVal > 0; pinVal--) {
ledcWrite(PWM_CHANNEL, pinVal);
delay(3);
}

for (; pinVal < 255; pinVal++) {
ledcWrite(PWM_CHANNEL, pinVal);
delay(3);
}
}
```

Presiona CTRL-U para compilar y cargar el esquema, luego podrás ver cómo el LED azul se va iluminado de forma progresiva.

Un ejemplo completo

Se puede cargar un ejemplo completo en el IDE de Arduino haciendo clic en el menú Files → Examples → ODROID-GO → LED or LED_PWM para importar y presionar CTRL-U para compilar y cargar.

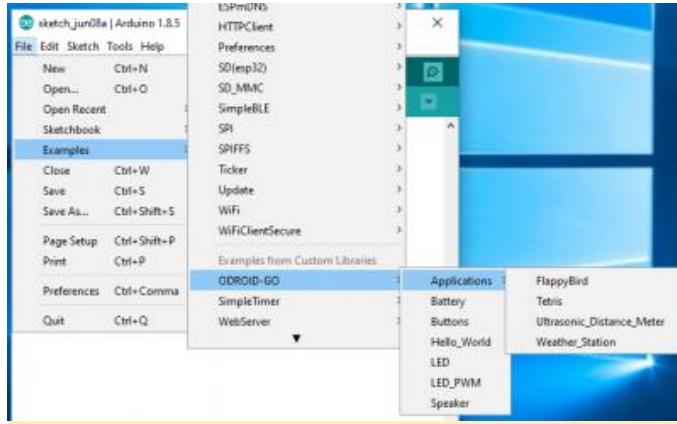


Figura 03 - Ejemplo completo cargado en IDE Arduino

Información Adicional

Para obtener información adicional y ayuda, utiliza los siguientes recursos o pregunta en los foros de ODROID:

- Consulta los documentos oficiales de Arduino. Estos proporcionan funciones habituales muy útiles con excelentes instrucciones.
- Consulta la guía de programación oficial de ESP32. La mayoría de las funciones específicas de ESP32 están incluidas en este artículo

Campamento de Programación – Parte 4: Leer el Estado de 12 Botones en ODROID-GO

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



En este artículo, aprenderemos cómo leer el estado de los botones con Arduino. En la LCD aparecerá un simple resultado. Antes de empezar, asegúrate de haber seguido estas guías:

- [Primeros pasos con Arduino](#)
- [Arduino para ODROID-GO – Hello World](#)

Consulta los documentos oficiales de Arduino. Estos proporcionan funciones habituales muy útiles y excelentes instrucciones detalladas. También puedes consultar la [guía de programación oficial de ESP32](#). La mayoría de las funciones específicas de ESP32 están incluidas en esta guía.

Obtener el estado de los botones

El ODROID-GO tiene 10 botones disponibles:

- 4 para el pad de dirección
- 2 para una acción

- 4 botones funcionales

Gracias a la librería GO, podemos obtener fácilmente el estado de cada botón. Primero, inicia la pantalla LCD con la función “GO.begin ()”. Fija el tamaño del texto en 2 ya que el tamaño de texto por defecto es demasiado pequeño para poder leerlo.

```
#include

$ void setup() {
// put your setup code here, to run once:
GO.begin();
GO.lcd.setTextSize(2);
}

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

Agrega una función independiente para mostrar el estado, y asínale el nombre “displayButtons ()”. Esta función tiene dos pasos:

- “GO.lcd.clear()” para borrar cualquier contenido de la LCD que haya sido mostrado previamente.
- “GO.lcd.setCursor()” para fijar el punto de inicio e imprimir una cadena. Los dos parámetros (0, 0) significa arriba, izquierda.

```
#include

$ void setup() {
// put your setup code here, to run once:
GO.begin();
GO.lcd.setTextSize(2);
}

$ void displayButtons() {
GO.lcd.clear();
GO.lcd.setCursor(0, 0);
}

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

Completa “displayButtons ()” con el siguiente código.

Esto mostrará si son o no presionados los botones.

Cuando se presione un botón, aparecerá la cadena de caracteres “Pressed” junto al elemento que corresponde al botón.

Todos los botones existentes para ODROID-GO están disponibles como instancias de la clase Button. La clase Button tiene algunas funciones muy útiles para hacernos saber el estado del botón. Por lo tanto, utilizaremos las funciones “isAxisPressed ()” y “isPressed ()” para saber si un botón está actualmente presionado o no.

Si se presiona el botón, esta función no devuelve 0.

- La función “isAxisPressed ()” es solo para el pad de direcciones. Si se presiona el botón, devuelve 1 o 2 para distinguir la dirección.
- La función “isPressed ()” es para el resto de botones. Si se presiona el botón, devuelve 1.

Por último, tenemos que completar la función “loop ()” para mostrar el estado en la pantalla LCD. Hay tres funciones para añadir:

- “GO.update()” que actualiza los estados de los botones para que las funciones “isPressed ()” funcionen bien.
- “displayButtons()” que añade cadenas de resultados a la pantalla LCD.
- “delay(1000)” que evita que la pantalla LCD parpadee demasiado rápido cuando actúa la función “GO.lcd.clean ()”.

```
#include

$ void setup() {
// put your setup code here, to run once:
GO.begin();
GO.lcd.setTextSize(2);
}

$ void displayButtons() {
GO.lcd.clear();
GO.lcd.setCursor(0, 0);

GO.lcd.println("/* Direction pad */");
GO.lcd.printf("Joy-Y-Up: %s
", (GO.JOY_Y.isAxisPressed() == 2) ? "Pressed"
: " ");
GO.lcd.printf("Joy-Y-Down: %s
", (GO.JOY_Y.isAxisPressed() == 1) ? "Pressed"
: " ");
GO.lcd.printf("Joy-X-Left: %s
", (GO.JOY_X.isAxisPressed() == 2) ? "Pressed"
: " ");
GO.lcd.printf("Joy-X-Right: %s
", (GO.JOY_X.isAxisPressed() == 1) ? "Pressed"
: " ");
GO.lcd.println("");
GO.lcd.println("/* Function key */");
GO.lcd.printf("Menu: %s
", (GO.BtnMenu.isPressed() == 1) ? "Pressed"
: " ");
GO.lcd.printf("Volume: %s
", (GO.BtnVolume.isPressed() == 1) ? "Pressed"
: " ");
GO.lcd.printf("Select: %s
", (GO.BtnSelect.isPressed() == 1) ? "Pressed"
: " ");
GO.lcd.printf("Start: %s
", (GO.BtnStart.isPressed() == 1) ? "Pressed"
: " ");
```

```

GO.lcd.println("");
GO.lcd.println("/* Actions */");
GO.lcd.printf("B: %s
", (GO.BtnB.isPressed() == 1) ? "Pressed" : "");
GO.lcd.printf("A: %s
", (GO.BtnA.isPressed() == 1) ? "Pressed" : "");
}

$ void loop() {
// put your main code here, to run repeatedly:
GO.update();
displayButtons();
delay(1000);
}

```

Presione CTRL-U para compilar y cargar el esquema. Luego, presiona cualquier botón para mostrar la cadena “Pressed” además de ese botón. Como habrás notado, deberás dejarlo presionarlo hasta que se

actualice la pantalla LCD debido a la función “delay (1000)”.

Un ejemplo completo

El ejemplo completo está disponible haciendo clic en el menú Files → Examples → ODROID-GO → Buttons para importar y presiona CTRL-U para compilar/cargar.

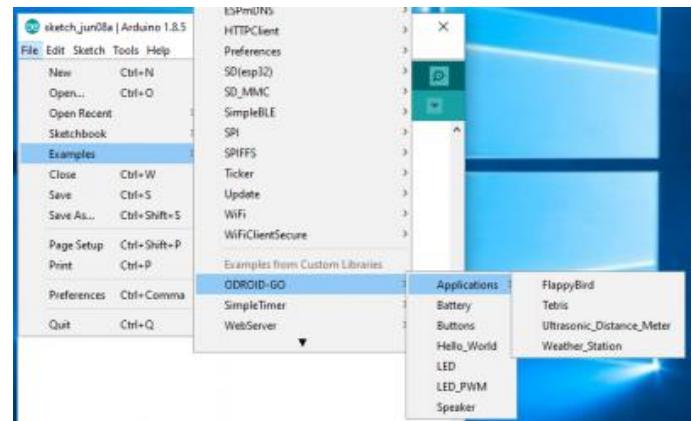
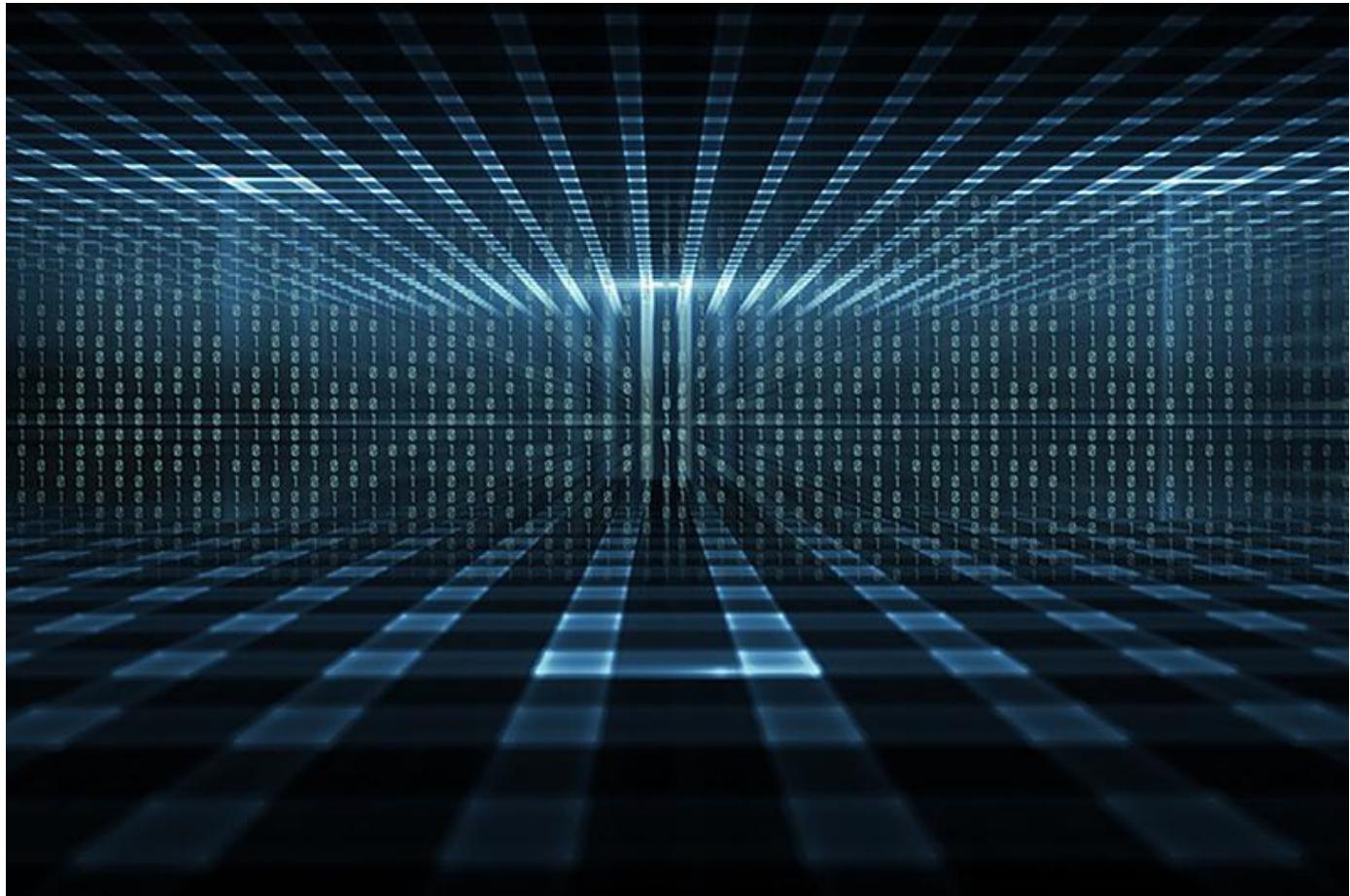


Figura 1 – Un ejemplo completo

Informática de Alto Rendimiento en el Hogar: Primeros Pasos con ODROID y MPI

© September 1, 2018 By Cooper Filby and Anthony Skjellum ↳ Linux, Mecaniqueo, Tutoriales



En este artículo, Cooper Filby y Anthony Skjellum de Runtime Computing Solutions LLC (<http://www.runtimecomputing.com>), describen de forma general la instalación y configuración de un clúster básico “sin nodo principal” con el objetivo final de ejecutar programas en paralelo basados en el envío de mensajes, utilizando en particular el modelo de programación en paralelo Interfaz de paso de mensajes (MPI). Existen numerosos sistemas operativos Linux precompilados disponibles para las placas ODROID desde el sitio web de Hardkernel. Para empezar, descarga la imagen de Ubuntu Server para tu modelo exacto de ODROID y extrae la imagen archivada IMG.XZ utilizando la herramienta 7zip en Windows o escribiendo “xz” desde la línea de comandos de Linux. Por último, debes copiar la imagen a una tarjeta SD o módulo eMMC con el comando “dd” en Linux/OS X o utilizar la herramienta Win32DiskImager.exe para ODROID en Windows.

Para obtener información de la instalación del sistema operativo, consulta el artículo de Bohdan Lechnowsky titulado “Instalación de un sistema operativo en ODROID” de la edición de enero 2014 de ODROID Magazine. Recomendamos utilizar los módulos eMMC disponibles en Hardkernel, para un mejor rendimiento. Sin embargo, las tarjetas SD también son válidas.

Conectando a tu ODROID

Puesto que hemos optado por utilizar Ubuntu Server en nuestros dispositivos ODROID, podemos conectarnos a nuestros sistemas XU-E (que llamaremos nodos) a través del protocolo “ssh” usando Terminal (o Putty si estamos en Windows) para continuar con la configuración de nuestro clúster. Debido a los potenciales conflictos iniciales con las direcciones MAC y los hostname que resolveremos más adelante, necesitamos reiniciar el

primer ODROID y realizar algunos ajustes antes de pasar al segundo.

[Nota: Si es posible, es recomendable disponer de una máquina con Linux o Windows que te ayudará con la instalación y reinicio del clúster, solucionar problemas de hardware y corregir posibles errores. Una alternativa a la utilización de un equipo independiente es conectar un teclado USB y un monitor HDMI en el primer ODROID y utilizarlo directamente para arrancar el clúster en lugar de usar SSH tal como se describe en el siguiente párrafo. Presiona Ctrl-Alt-F1 para utilizar la consola framebuffer si X11 no se ejecuta]

Para conectarte a tu ODROID ha de conocer el hostname o dirección IP del dispositivo. En la imagen de Ubuntu Server utilizada en nuestro clúster XU+E el hostname por defecto es "odroid-server", mientras que para el resto de imágenes que hemos utilizado, el hostname usado es "odroid". La mayoría de las redes domésticas deberían soportar DNS por defecto, lo que te permite conectarse simplemente con el hostname. Si esto no funciona, puedes conectarte mediante la dirección IP que el router ha asignado al ODROID. Si no se resuelve ningún hostname entra en el panel de control del router para localizar la dirección IP que necesitas. Esta información suele estar en la tabla de clientes DHCP dentro del panel de administración del router.

Dado que se han utilizado copias idénticas de la imagen en ambos nodos, por defecto presentan un conflicto de hostname. Este problema lo resolveremos iniciándolos uno por uno y luego cambiando la configuración de red de forma individual. Si no puedes acceder al panel de control del router, puedes utilizar el comando "nmap" para escanear la red en busca de los dispositivos ODROID. Ej.: "nmap 192.168.1.0/24". Busca un host que tenga el puerto 22 abierto

Enciende uno de los ODROID y a continuación introduce "ssh odroid@ubuntu-server" (o "ssh odroid@xxx.xx.xx.xxx", si utilizas la dirección IP) en el Terminal o con la herramienta Putty del ordenador principal. Esto establecerá una conexión segura con el dispositivo ODROID. Utiliza "odroid" como contraseña para iniciar sesión.

Una vez que aparezca el prompt de comandos, deberías ejecutar "sudo apt-get update && sudo apt-get upgrade" para asegurarnos de que el sistema operativo esté actualizado. Además, se recomienda ejecutar el comando "passwd" para cambiar la contraseña del usuario odroid, o crear nuevas cuentas de usuario con el comando "adduser", como por ejemplo ejecutar "sudo adduser kilroy". (En términos generales, debes tener en cuenta tres cosas a la hora de determinar las contraseñas de los nodos: que tengas una cierta extensión, que sean difíciles de adivinar y que estén guardadas en un lugar seguro).

Configurando la red

Antes de tener los dispositivos ODROID operativos y funcionando, debemos realizar algunos cambios en la configuración de la red y eliminar los conflictos de direcciones MAC y hostname que pueden aparecer en tu clúster ODROID. Para cambiar el hostname tenemos que editar dos archivos /etc/hostname y /etc/hosts cambiando "odroid-server" por otro nombre y reiniciar la máquina para que los cambios tengan efecto. A efectos de este artículo, utilizaremos odroid-server0 y odroidserver1 para referirnos al primer y segundo ODROID respectivamente. De forma alternativa y si tu sistema operativo lo permite, puede ejecutar "sudo odroid-config" para cambiar el hostname y utilizar cualquier otro nombre. Cada hostname debe ser único para cada nodo.

El conflicto de las direcciones MAC es una cuestión delicada que detectamos cuando configuramos varios dispositivos ODROID XU+E. Nos hemos encontrado que, por defecto, el componente Ethernet integrado en todos los dispositivos compartían la misma dirección MAC, lo cual hacía imposible trabajar en un único ODROID si conectábamos múltiples en la misma red. Como ambos ODROID tienen direcciones MAC idénticas, podemos hacer dos cosas: 1) configurar uno (o ambos) ODROID para usarlo con direcciones MAC diferentes. 2) Configurar dongles Ethernet USB para que todos tengan direcciones MAC únicas. No importa que valores específicos se elijan, siempre y cuando sean únicos en tu Red de Área Local (LAN).

Para cambiar la dirección MAC de un dispositivo, edita /etc/network/interfaces con un editor de texto y añade la línea “hwaddress ether newmac”, donde Newmac es una dirección con el formato “b6:8d:67:7b:cb:e0”, debajo de las siguientes líneas.

```
auto eth0
iface eth0 inet dhcp
```

A continuación, reinicia el ODROID para que los cambios tengan efecto. Asegúrate de verificar la nueva dirección con el comando “ifconfig”.

Alternativamente, se puede optar por conectar un adaptador Ethernet USB en la ranura USB 3.0 y luego, ejecutar “ifconfig-a | eth grep” que debería devolver algo similar a esto:

```
eth0 Link encap:Ethernet HWaddr
b6:8d:67:7b:cb:e0
eth2 Link encap:Ethernet HWaddr
00:13:3b:99:92:b1
```

Por defecto, eth0 será el componente Ethernet 10/100 integrado, mientras que el segundo dispositivo de Ethernet (en este caso, eth2) será el adaptador Ethernet USB. Si sólo aparece eth0, intenta conectar de nuevo el adaptador Ethernet USB y/o verifica que funciona en otra máquina. Para configurar el adaptador de forma que use DHCP en el arranque y obtener una dirección IP, modifica /etc/network/interfaces y agrega las siguientes líneas entre las entradas “auto lo” y “auto eth0”:

```
auto eth2
iface eth2 inet dhcp
```

Utiliza el ID correcto del dispositivo de red que has localizado previamente a través ifconfig (en este caso, eth2). A continuación, apaga el ODROID, conecta el cable de red al adaptador Ethernet USB y vuelve a encender de su nuevo tu ODROID. Si por alguna razón, no hay conexión prueba a desenchufar y enchufar de nuevo el cable y verifica que el adaptador Ethernet USB aparece en pantalla usando el comando “ifconfig-a”. También se puede dar el caso de que el ID del dispositivo Ethernet haya cambiado si el adaptador ha sido desconectado. En este caso debemos actualizar /etc/network/interfaces en consecuencia

Llegado a este punto el primer ODROID está configurado y accesible en red. Antes de adéntranos en la Interfaz de Paso de Mensajes (MPI), configura el segundo ODROID siguiendo los mismos pasos utilizados para configurar el primer ODROID.

Interfaz de Paso de Mensajes(MPI)

Ahora que tenemos ambos nodos configurados correctamente. Podemos empezar a ejecutar trabajos HPC en nuestro clúster formado por dos nodos. Un entorno de programación paralelo como MPI nos facilitará esta labor, el cual se encarga de iniciar los procesos que conforman el modelo de programación paralelo. Proporciona una Interfaz de programación de aplicaciones (API) que permite la comunicación entre procesos secuenciales usados para desarrollar trabajos de programación. Para lograr esto vamos a utilizar la Interfaz de Paso de Mensajes (MPI), que proporciona una API que permite a los nodos enviar y recibir mensajes durante la ejecución de tareas. El comando “mpirun” o “mpiexec” inicia todos los procesos necesarios para controlar tus dispositivos ODROID de forma segura. Podemos descargar cualquiera de las dos implementaciones MPI de código abierto que existen actualmente: MPICH y OpenMPI. Para nuestro clúster ODROID en red ambas opciones son igual de válidas

Para instalar MPICH, ejecuta “sudo apt-get install mpich2”, y para instalar OpenMPI ejecuta “sudo apt-get install openmpi-bin”

Cosas que puedes hacer una vez cargado MPI:

- 1) Ejecutar programas de ejemplo que usen múltiples núcleos en un único ODROID
- 2) Ejecutar programas de ejemplo que utilice los 8 núcleos de nuestros dispositivos ODROID
- 3) Aprender a compilar nuestros propios programas MPI

En este artículo nos hemos centrado en los dos primeros usos. Puedes echar un vistazo a ejemplos de programas desarrollados con OpenMPI y MPICH para seguir aprendiendo. También existen excelentes tutoriales en Internet y numerosos libros de programación MPI como “Using MPI” de MIT Press (Uno de los coautores de este libro).

Mejorando el sistema

El contenido de este artículo representa una pequeña fracción de lo que podemos llegar a realizar con nuestro clúster. No obstante, esta configuración es más que suficiente para hacer funcionar dos nodos con unos cuantos usuarios. Si queremos que nuestro clúster crezca tendremos que hacer uso de un nodo principal que nos permita manejar un mayor número de usuario y nodos conectados. Además de poder

ocultar el tráfico del clúster del resto de la red, este nodo principal albergara los servicios que facilitaran la gestión del clúster, como LDAP para la gestión de documentos, Puppet para la gestión de contenido, NFS para compartir archivos, y otros servicios y herramientas de redes.

En la segunda entrega de esta serie de tutoriales convertiremos odroid-server0 en un nodo principal.

Home Assistant: Seguimiento de Personas con Wi-Fi Usando Kismet

⌚ September 1, 2018 📝 By Adrian Popa ⚡ Linux, Mecaniqueo, Tutoriales



Parece ser que el seguimiento de las personas se ha convertido en un mercado multimillonario (<https://goo.gl/T1XZS8>), el cual se puede usar para crear perfiles sobre personas (conocer los hábitos de una persona y monetizar en función de éstos) o para optimizar las tiendas y lugares dónde va la gente. Por ejemplo, si una tienda de ropa puede rastrear tus movimientos (y recordar dónde has estado en visitas anteriores), pueden crear un perfil con los tipos de ropa que te gusta y pueden bombardearte posteriormente con anuncios personalizados. El seguimiento de personas se puede realizar de muchas maneras, desde el reconocimiento facial hasta el seguimiento por wifi/bluetooth, con una precisión variable. Al igual que con cualquier tecnología, el seguimiento se puede utilizar para hacer el bien (por ejemplo, encontrar personas sepultadas tras un terremoto) o el mal (acechar a tu vecina de al lado).

En mi caso, quiero seguirle la pista a la niñera para saber si está en casa o no, de esta forma sé si tengo que ir a recoger a mi hijo al parque. Esto es bastante fácil de hacer con el seguimiento wifi y Home Assistant. El problema que tengo es que el teléfono de mi niñera no se conecta a mi red wifi, así que necesito usar una forma pasiva de monitorización.

Ya hemos comentado cómo funciona el wifi (<https://goo.gl/yWD2j1>) y también cómo se puede detectar (<https://goo.gl/uEsdMo>). En pocas palabras: los clientes y los puntos de acceso transmiten regularmente su SSID o sondean buscando cualquier SSID de una determinada área. Todo el tráfico inalámbrico (incluso el tráfico encriptado) tiene información de capa 2 (direcciones MAC) sin cifrar y ésta puede ser rastreada. Para escuchar el tráfico wifi, debes tener una tarjeta inalámbrica que admita el modo monitor (los adaptadores wifi HardKernel lo

admiten). Normalmente, las direcciones MAC son únicas por dispositivo y se pueden utilizar para rastrear un dispositivo específico, de modo que cualquier persona que tenga una tarjeta WiFi con modo monitor puede rastrearte (como estamos a punto de ver).

Instalación del súper vanguardista Kismet

La forma más sencilla de empezar a escuchar el espectro wifi es instalar kismet. Kismet se encarga de poner tu adaptador wifi en modo monitor y, además, puede registrar todo el tráfico detectado. Para acceder a la nueva interfaz de usuario y también para tener una REST API, necesitarás instalar kismet desde la fuente en lugar de hacerlo desde un administrador de paquetes (tanto Ubuntu 16.04 como 18.04 tienen versiones demasiado antiguas de kismet). Tienes instrucciones generales sobre ello en:

<https://goo.gl/qjwLkb>.

Primero necesitarás instalar algunas herramientas de desarrollo:

```
$ sudo apt-get install build-essential git  
libmicrohttpd-dev  
pkg-config zlib1g-dev libnl-3-dev libnl-genl-  
3-dev libcap-dev  
libpcap-dev libncurses5-dev libnm-dev libdw-  
dev libsdl1.2debian-dev  
libprotobuf-dev libprotobuf-c-dev protobuf-  
compiler  
protobuf-c-compiler  
$ sudo apt-get install python python-  
setuptools python-protobuf  
python-sqlite python-requests
```

La compilación de kismet necesitará más memoria RAM de la que probablemente tengas, de modo que no es mala idea activar el intercambio de memoria swap basado en disco. Yo opte por 1G swap para mi C2, pero si tienes más espacio disponible, puedes crear un archivo swap más grande:

```
$ dd if=/dev/zero of=/swap bs=1M count=1000  
$ mkswap /swap  
$ swapon /swap
```

A continuación, puedes hacerte con la última copia del desarrollo de Kismet, compilarla e instalarla:

```
$ git clone  
https://www.kismetwireless.net/git/kismet.git  
$ cd kismet  
$ ./configure  
$ make -j 4  
$ sudo make suidinstall
```

Una vez hecho esto, puede desactivar el intercambio swap que creaste y recuperar tu espacio en disco:

```
$ sudo swapoff /swap  
$ sudo rm -f /swap
```

Kismet se instalará en /usr/local, con su configuración ubicada en /usr/local/etc/kismet.

Puedes crear un nuevo script de inicio systemd para kismet:

```
$ cat /etc/systemd/system/kismet.service  
[Unit]  
Description=Kismet wifi monitor  
  
[Service]  
Type=forking  
ExecStart=/usr/local/bin/kismet -c wlan0 --no-  
curses-wrapper --daemonize -n  
WatchdogSec=3600  
Restart=always  
  
[Install]  
WantedBy=multi-user.target  
$ sudo systemctl enable kismet  
$ sudo systemctl start kismet
```

El script de inicio systemd inicia kismet como un demonio, sin registro, vinculado a wlan0 (que se pondrá en modo monitor). Yo he tenido algunos problemas con los puertos USB de mi C2 (principalmente porque he estado ejecutando varios gadgets desde un hub no alimentado) y, a veces, el adaptador wifi se bloqueaba y necesitaba reiniciarse. Si reinicio kismet cada hora, el problema desaparece, esto es lo que hace precisamente WatchdogSec.

Una vez iniciado, puedes conectarte a su interfaz web en <http://:2501/>. Tienes acceso de solo lectura sin autentificarte, pero deberás crear un usuario si quieres cambiar la configuración. Cuando inicies kismet por primera vez, se generará y almacenará

una contraseña aleatoria para el usuario kismet en /root/.kismet/kismet_httpd.conf.

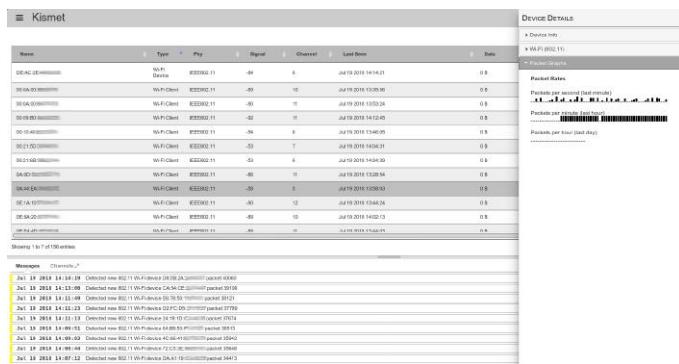


Figura 1 – UI de Kismet

Conocer la MAC

Para rastrear a alguien a través de wifi, necesitarás conseguir su dirección MAC de una forma u otra. Puedes solicitarla o utilizar la ingeniería social para averiguarla (p. Ej., Solicitar ver la configuración de red para solucionar algún problema ficticio), o tendrás que trabajar un poco en el caso de que no tengas acceso en absoluto al dispositivo que quieras rastrear.

Si estás atascado en el último caso, tendrás que hacer algunas suposiciones:

- La persona rastreada tiene el wifi abierto y hacer solicitudes de sondeo
- La persona rastreada tiene una MAC estática
- Saber cuándo la persona rastreada se encuentra dentro del alcance de detección

En este caso, aquí tienes la estrategia a seguir. Inicia kismet en modo monitor y haz que se registre el tráfico ambiental en un archivo (base de datos sqlite3). Detén la captura antes de que la persona de interés se encuentre dentro del alcance y realiza una nueva captura cuando la persona esté cerca. Deja la captura activa durante el mayor tiempo posible (al menos 10-15 minutos) y detenla cuando la persona ya no se encuentre cerca. A continuación, extrae las direcciones MAC de ambos conjuntos de datos y marca las diferencias. Te interesarán las MAC que aparecen cuando el objetivo estaba presente pero que no se encuentran en el primer conjunto de datos (elimina las MAC encontradas en el primer conjunto de datos del segundo). Deberás quedarte con un

listado más pequeño de MAC, una de las cuales es la MAC del objetivo. Puedes afinar aún más este listado teniendo en cuenta los registros de fecha y hora: elimina las MAC que aparecen después de que el objetivo ya no se encontrase cerca o antes de que aparezca el objetivo. Si aún cuentas con una lista de varias MAC y no puedes excluir algunas basándote en el fabricante (por ejemplo, todos son teléfonos Samsung), tendrás que repetir el proceso en otro momento y ver qué MAC del nuevo registro son las mismas que las posibles MAC del objetivo. El proceso de eliminación debe terminar con solo una MAC que aparece en todas las capturas y no es "regular". Si estás haciendo esto en un área "tranquila" (por ejemplo, una casa), encontrará la MAC con bastante rapidez. Si, en cambio, estás haciendo esto en un área concurrida, tendrás que registrar el tráfico del ambiente en varias ocasiones.

Veamos el plan en acción.

Primero necesitas llevar a cabo la captura. Para activar esto, edita /usr/local/etc/kismet_logging.conf y cambia log_prefix=/tmp/.

A continuación, edita el servicio systemd y elimina el parámetro de la línea de comandos "-n" para que el registro finalice:

```
$ sudo sed -i 's/--daemonize -n--daemonize/' /etc/systemd/system/kismet.service
$ sudo systemctl daemon-reload
$ sudo service kismet restart
```

Ahora obtendrás archivos de registro por hora en /tmp con los dispositivos que ha visto kismet. Una vez que tenga suficientes datos, puedes desactivar el registro y reiniciar kismet editando el servicio systemd y agregando "-n".

Después, deberás dividir los archivos que has recopilado en dos grupos. El directorio "0" contendrá los archivos donde se sabe que el objetivo no ha estado presente, mientras que el directorio "1" contendrá los archivos donde el objetivo podría haber estado presente. Puedes hacer esta división teniendo en cuenta el tiempo (tienes intervalos de una hora por defecto).

```
$ mkdir 0 1
$ sudo mv /tmp/Kismet-20180718-07-33-00-
1.kismet 0/
$ sudo mv /tmp/Kismet-20180718-10-33-03-
1.kismet 1/
```

Ahora viene el trabajo duro. Necesitamos extraer todos las MAC de todos los archivos de ambas carpetas y compilar dos listas: MAC de la carpeta 0 y de la carpeta 1. Podemos obtener datos del registro log de kismet utilizando sqlite3 con una sintaxis SQL. Como sqlite3 no admite comodines para los nombres de archivos, debemos hacer algunas artimañas con bash y repetirlas sobre cada archivo.

```
$ sudo apt-get install sqlite3 bc
$ for file in 0/*.kismet; do sqlite3 -csv
"$file" 'select
devmac,first_time,last_time from devices;' | 
tee -a 0/0.dump; done
$ for file in 1/*.kismet; do sqlite3 -csv
"$file" 'select
devmac,first_time,last_time from devices;' | 
tee -a 1/1.dump; done
```

Para ambos directorios, hemos descargado la MAC, la hora de inicio y la hora de finalización de Kismet. El siguiente paso es unir todos los datos (ya que tener múltiples archivos da lugar a entradas duplicadas). Utilizaremos sort + uniq y conservaremos solo las direcciones MAC. En mi caso, terminé con 701 MAC que sabía que no eran el objetivo, y 229 MAC que podrían ser el objetivo.

Llegados a este punto, tenemos que reducir aún más esta lista potencial eliminando las MAC que sólo han aparecido un breve periodo de tiempo (durante menos de 100 segundos) porque es probable que pertenezcan a personas o coches que han pasado de largo. Esto reduce aún más el número de MAC ajustándose a 152 en mi caso.

```
$ cat 0/0.dump | cut -f 1 -d ',' | sort -u >
0/0.mac
$ while read line; do start=`echo $line | cut
-f 2 -d ','`;
end=`echo $line | cut -f 3 -d ','`;
difference=`echo $end-$start|bc`; if [
$difference" -gt 100 ]; then echo "$line" |
tee -a 1/1_filtered.dump; fi; done < 1/1.dump
```

```
$ cat 1/1_filtered.dump | cut -f 1 -d ',' |
sort -u > 1/1.mac
```

Ahora comienza el proceso de eliminación: eliminamos las líneas coincidentes de los dos archivos. Para esto podemos usar grep -v para imprimir las líneas que no coinciden y buscamos las contenidas de 0/0.mac dentro de 1/1.mac:

```
$ grep -v -f 0/0.mac 1/1.mac > potential.mac
```

Esto me deja con solo 27 MACs para seguir investigando. Podría filtrar por el proveedor del dispositivo (si supiera cual es). Una forma de mantener solo los dispositivos Samsung es buscar su MAC en la base de datos OUI. Puedes instalar una copia local de la base de datos con el paquete ieee-data y puede buscar todas las MAC y conservar únicamente las registradas en Samsung.

```
$ sudo apt-get install ieee-data
$ while read line; do match=`echo $line | cut
-c 1-8 | sed 's/:/:/g' | xargs -n 1 -I{} grep
{} /usr/share/ieee-data/oui.txt | grep -i
Samsung`; if [ -n "$match" ]; then echo
"$line" | tee -a samsung.mac; fi; done
<potential.mac
```

Reduco a 4 las MACs tras aplicar este filtro, así que me estoy acercando bastante. Para afinar aún más, necesito realizar más escuchas cuando el objetivo esté cerca y encontrar qué MAC de del nuevo registro se encuentran en el registro anterior hasta que me quede solo con una. O bien, podría monitorizar las 4 y ver dentro de Home Assistant cuál de ellas actúa correctamente.

Añadir un device_tracker personalizado en Home Assistant

La ventaja del nuevo kismet que acabas de instalar es que puedes hacerle consultar a través de su REST API y obtener una respuesta JSON de los nuevos dispositivos que coincidan con tus criterios de búsqueda. La documentación de la REST API está disponible aquí: <https://goo.gl/JFpnZM>.

La idea es implementar un device_tracker personalizado como un módulo en Home Assistant que cogerá una lista de direcciones MAC o una lista de SSID y preguntará a una instancia de kismet si vio

estas MACs/SSIDs en los últimos 30 segundos. Si Kismet los ha visto, reportará sus nombres, que a su vez se transmitirán a Home Assistant.

Actualmente, kismet device_tracker se puede usar como un componente personalizado. Es de esperar que, en el futuro, cuando la REST API se estabilice, pueda fusionarse directamente en Home Assistant. Puedes instalarlo siguiendo estos pasos:

```
$ sudo su - homeassistant
$ cd .homeassistant
$ mkdir -p custom_components/device_tracker
$ cd custom_components/device_tracker
$ wget -O kismet.py https://goo.gl/WPGZZA
```

Necesitarás editar configuration.yaml y añadir el componente:

```
device_tracker:
  - platform: kismet
    interval_seconds: 30
    host: 192.168.1.15
    port: 2501
    consider_home: 420
    clients:
      - 84:98:66:47:cf:b9
      - d0:31:69:38:f2:99
    ssids:
      - DIGI
```

Para una depuración adicional (aunque con ruido), puedes configurar el componente para depurar también dentro del programa registrador:

```
logger:
  default: error
  logs:
    custom_components.device_tracker.kismet: debug
```

Una vez que reinicies Home Assistant, el componente se conectará periódicamente a la instancia de kismet y consultará los dispositivos que enumeraste. Obtendrá resultados para el último período "interval_seconds" (por ejemplo, durante los últimos 30 segundos), de modo que incluso si el dispositivo móvil estuvo activo durante un tiempo en ese intervalo, se detectará y se informará de ello. La duración de interval_seconds solo afecta a la rapidez con que se ve un dispositivo, no si se ve o no. Los parámetros tienen los siguientes significados:

- interval_seconds – con qué frecuencia se solicita actualizaciones desde el servidor kismet
- host – la IP/FQDN del servidor kismet (127.0.0.1 por defecto)
- port – el puerto donde se ejecuta kismet (2501 por defecto)
- consider_home – ¿durante cuánto tiempo se debe considerar que un cliente permanece en casa si no se ha visto? (7 minutos es aceptable para clientes con wifi abierto, aunque no esté conectado a una red. Puede depender de un dispositivo a otro)
- clients – una lista de direcciones MAC para buscar. Puede ser una expresión regular
- ssids – una lista de SSID para buscar. Puede ser una expresión regular

Los dispositivos que sean detectados se agregan a known_devices.yaml, al cual se puede acceder como una entidad dentro de Home Assistant. Una vez descubiertos los dispositivos, puede personalizar su nombre y añadirles una imagen editando known_devices.yaml

```
the_nanny:
  hide_if_away: false
  icon:
  mac: D0:31:69:1[REDACTED]
  name: The Nanny
  picture: /local/the_nanny.jpg
  track: true
```

Figura 2 – Entrada known_devices.yaml

Puedes añadir las entidades para ser rastreadas dentro de un grupo y mostrarlas en la interfaz de usuario principal de Home Assistant

```
group:
  people:
    name: People
    view: yes
    entities:
      - device_tracker.the_nanny
      - device_tracker.samsungj3
      - device_tracker.nexus5
```

People

	The Nanny	Home
	samsung-j3	Away
	nexus5	Away

Figura 3 – Seguimiento de personas

Si estás utilizando la interfaz de usuario personalizada (<https://github.com/andrey-git/home-assistant-custom-ui/>) con Home Assistant también puede mostrar la última hora modificada (por ejemplo, "hace 2 minutos") debajo del nombre de la entidad para contar con una indicación rápida de cuánto tiempo hace que una persona llegó o se fue. Para hacerlo, asegúrate de estar en la última versión de CustomUI (ejecuta `./update-custom-ui.sh` desde dentro de `~homeassistant/.homeassistant`) y añade lo siguiente en la configuración:

```
homeassistant:  
  customize:  
    device_tracker.the_nanny:  
      custom_ui_state_card: state-card-custom-ui  
      show_last_changed: true
```

Conclusión

Entonces, ¿qué tan funciona? Depende del dispositivo que quieras rastrear. Los dispositivos más antiguos/más baratos no intentan ocultar su MAC y se detectan con facilidad (como mi Samsung J3). Los dispositivos más nuevos/caros utilizan MACs aleatorias para hacer solicitudes de sondeo y son más difíciles de concretar. Sin embargo, no es imposible. Todos los teléfonos usan su dirección MAC real cuando se conectan a un punto de acceso conocido. Por lo tanto, si (independientemente) transmites una lista de nombres de puntos de acceso populares del área del objetivo (por ejemplo, Starbucks, McDonalds, etc.), puedes convencer a su wifi para que se de late e intente conectarse a tu punto de acceso. Este método dejará rastros, porque todos esos puntos de acceso wifi serán visibles en la lista de redes para todos los clientes, lo cual levantará sospechas, aunque este artículo te ayudará a empezar: <https://goo.gl/rXg2so>.

Dependiendo del comportamiento del teléfono, es posible que entre en modo reposo de vez en cuando y es posible que se pierdan algunas solicitudes de sondeo (especialmente si el teléfono tiene batería), pero debería ser bastante visible cuando su pantalla esté encendida o el usuario esté haciendo una llamada. Para mejorar la precisión, es posible que necesites añadir más oyentes kismet alrededor de tu casa para cubrir más canales o puntos ciegos.

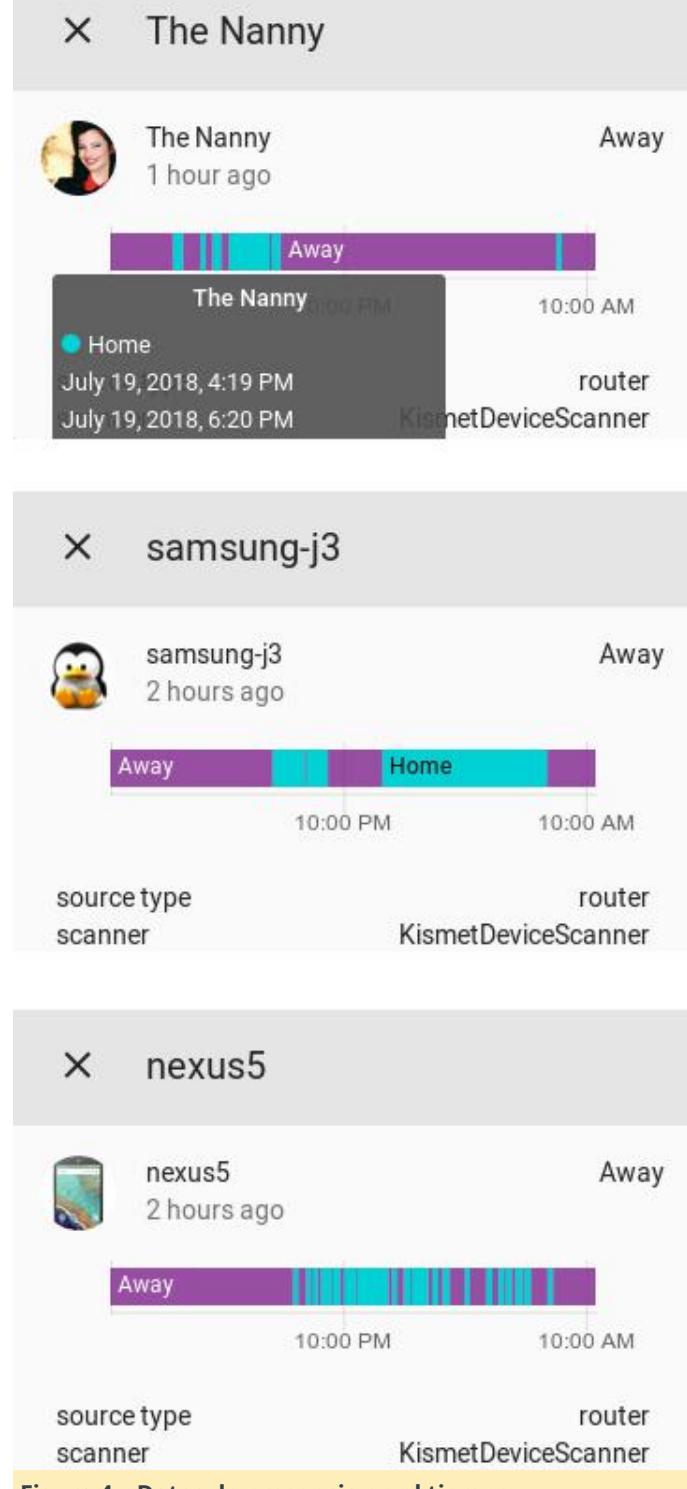


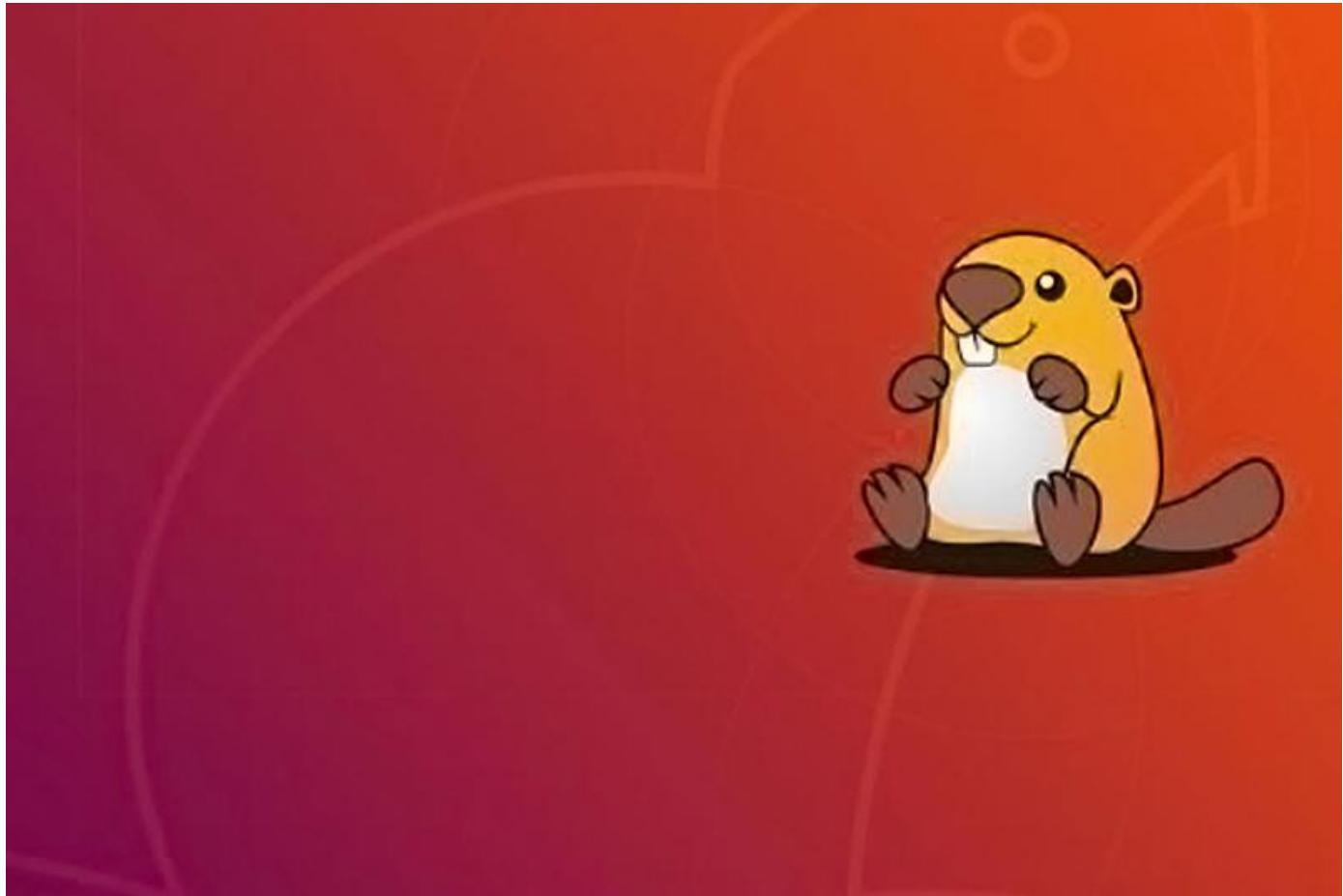
Figura 4 – Datos de presencia en el tiempo

¿Qué puedes hacer para evitar la detección y el seguimiento? Sencillo. Apaga tu wifi cuando no la utilices. Si no tienes lo último en teléfono móvil (Android P parece haber incluido las MAC aleatorios), puedes usar aplicaciones de terceros como Pri-Fi (<https://play.google.com/store/apps/details?id=eu.chainfire.pryfi>) por chanfire (el creador de SuperSU) para hacer los mismo. Aunque dependiendo de la tecnología de la persona que está rastreando, la mayoría de la gente no se molestará en ocultarse.

(<https://play.google.com/store/apps/details?id=eu.chainfire.pryfi>)

Primeros Pasos con Ubuntu 18.04 en el ODROID-XU4: Una Guía para Principiantes

⌚ September 1, 2018 📝 By Melissas Miltiadis 📱 Linux, ODROID-XU4, Tutoriales



El ODROID-XU4 es básicamente un ordenador linux heterogéneo multi-proceso de ocho núcleos Linux. Al ofrecer soporte de código abierto, la placa puede ejecutar diversas versiones de Linux, incluido el último Ubuntu 18.04 y las últimas versiones de Android. Al implementar las interfaces eMMC 5.0, USB 3.0 y Ethernet Gigabit, el ODROID-XU4 cuenta con increíbles velocidades de transferencia de datos, una característica que cada vez es más necesaria para soportar la capacidad de procesamiento avanzada de los dispositivos ARM. Esto permite a los usuarios realmente experimentar una mejoría en términos informáticos, especialmente con un arranque más rápido, navegación web más fluida, sistema de redes más avanzado y juegos en 3D. Para obtener más detalles técnicos sobre la placa, puedes visitar la página de especificaciones del producto de Hardkernel en <https://goo.gl/bQ5szX>.

ODROID XU4 es compatible con Linux Kernel 4.14 LTS y puede ejecutar el último Ubuntu 18.04 bastante bien. En esta guía paso a paso veremos cómo instalar y ejecutar este sistema operativo en la placa junto con algunas aplicaciones que deberías tener, así como las ventajas y desventajas de ejecutar dicho sistema operativo en este dispositivo.

Los materiales necesarios para ejecutar Ubuntu 18.04 en un ODROID-XU4 son:

- ODROID-XU4 (<https://goo.gl/19qHGB>)
- Fuente de alimentación 5V/2A US: (<https://goo.gl/6z5XLC>), EU: (<https://goo.gl/bgmxBQ>), Asia/Korea (<https://goo.gl/TqLrr8>)
- Módulo eMMC preinstalado con Linux (<https://goo.gl/RxM5y9>) o tarjeta microSD: (<https://goo.gl/y3TdEk>)
- Cable HDMI: (<https://goo.gl/8TX8Y6>)
- Monitor o TV con un puerto HDMI

Es hora de empezar con el procedimiento de instalación.

Instalación

Primero, descarga el sistema operativo Ubuntu 18.04 (20180501) desde el sitio web de Hardkernel en <https://goo.gl/ErjTVT>. Asegúrate de esperar a que se complete la descarga. Puedes descargar la imagen desde la página de la wiki de Hardkernel en <https://goo.gl/JYvNGY>.



¡Ubuntu 18 ya está entre nosotros!

Para instalar o “flashear”, Ubuntu 18.04 en la tarjeta de memoria, recomendamos usar Etcher, tal y como se describe en <http://bit.ly/2f61k5x>. Puedes descargar etcher desde <https://etcher.io/>.

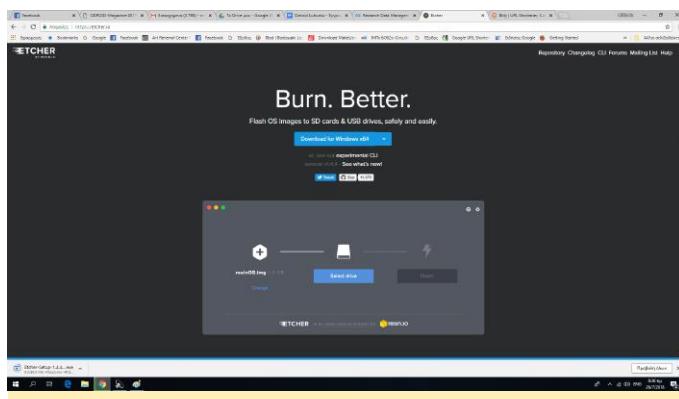


Figura 01 – Etcher

Etcher funciona en Mac OS, Linux y Windows, y es la opción más fácil para la mayoría de los usuarios. Etcher también permite escribir las imágenes de SO directamente desde el archivo zip, sin necesidad de descomprimir. Para instalar el sistema operativo en un módulo eMMC, necesitarás un lector de módulos eMMC (<https://goo.gl/A5LVTR>) y un multi-lector USB (<http://bit.ly/2f61k5x>) para conectarlo a tu PC.

Primer Arranque

En el primer arranque, “cuidado con el hueco”. Según la Wiki de Hardkernel, la función automática para redimensionar el tamaño de RootFS ha cambiado.

Una vez finalizado el redimensionamiento automático, la energía se apagará automáticamente. Espera un par de minutos y presiona el botón de encendido si el LED azul está apagado. La imagen arrancará en 30-40 segundos, la Figura 2 muestra nuestro reciente instalado Ubuntu 18.04 con el escritorio Mate.



Figura 02

Lo primero que debes hacer es ajustar la hora y la fecha. Desde el menú, haga clic en Administration->Time and Date, tal y como se muestra en la Figura 3.

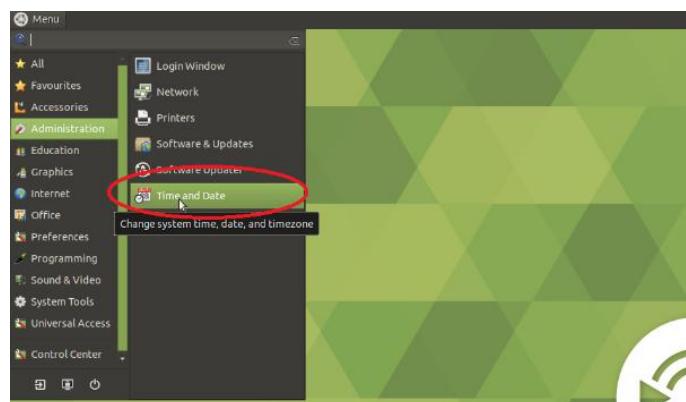


Figura 03

A continuación, actualiza el sistema y el software kernel. No olvides dist-upgrade (la contraseña es “odroid”):

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

Luego, instala tu idioma preferido (el mío es el griego), y finalmente agrega el estado de accesibilidad del teclado al panel, tal y como se muestra en las Figuras 4 – 7.

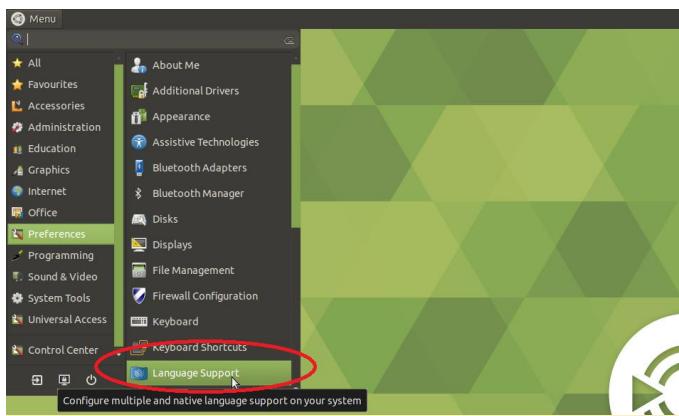


Figura 04

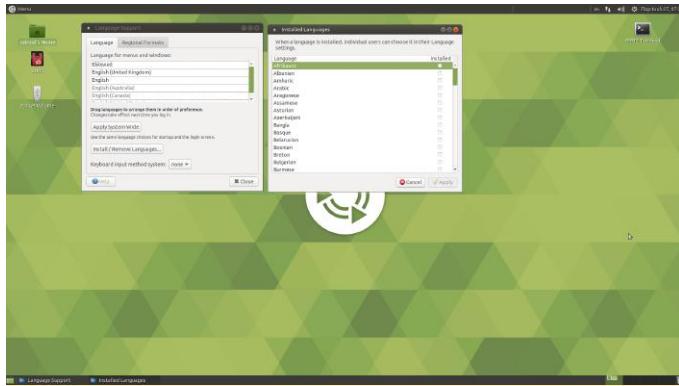


Figura 05

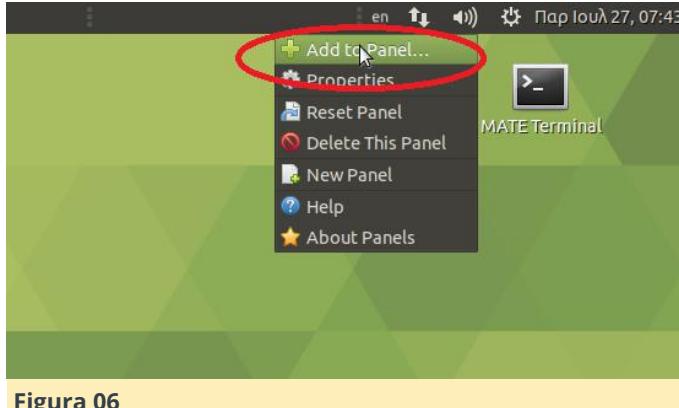


Figura 06

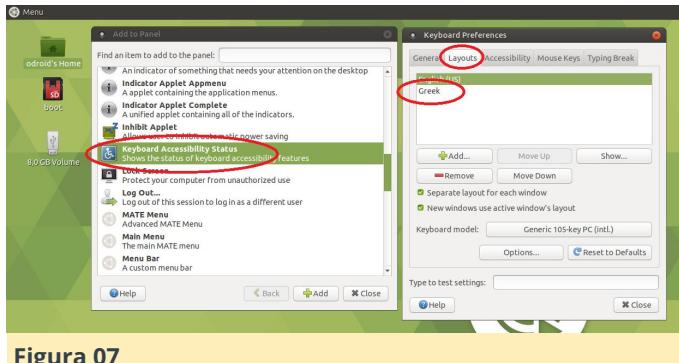


Figura 07

Instalar aplicaciones y probar el rendimiento La imagen Ubuntu 18.04 de Hardkernel viene preinstalada con muchas aplicaciones, como el navegador Chromium, Thunderbird (cliente de correo electrónico), LibreOffice y reproductor multimedia MPV, solo por nombrar algunos y, por supuesto, Kodi. Sin embargo, se necesitan algunos más para una completa experiencia de sistema operativo de escritorio.

Instala SMPlayer, uno de los mejores reproductores en Linux. De modo que desde la consola simplemente escribe:

```
$ sudo apt install smplayer
```

y luego selecciona Menu→ Sound and Video→ SMPlayer en el menú, tal y como se muestra en la Figura 8.

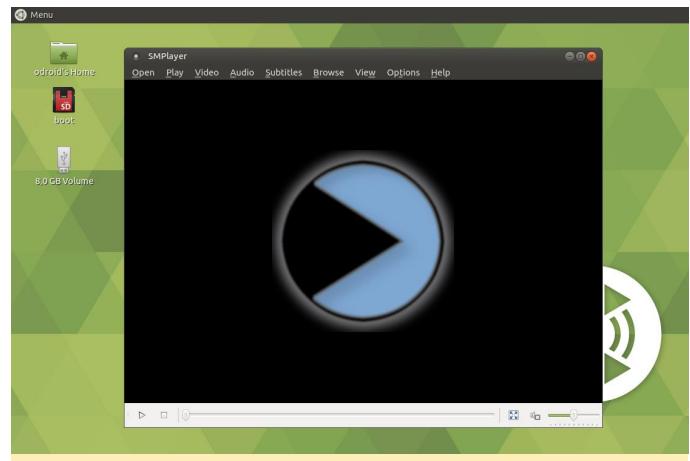


Figura 08

Instala GIMP, el mejor editor de gráficos para Linux, que se ejecuta muy rápido y funciona bastante bien:

```
$ sudo add-apt-repository ppa:otto-kesselgulasch/gimp
$ sudo apt-get update
$ sudo apt-get install gimp
```

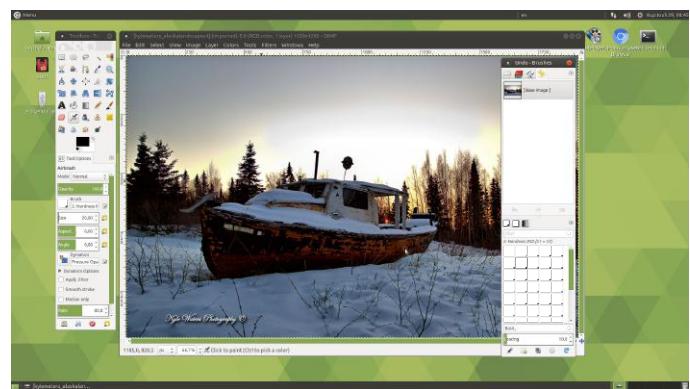


Figura 09

También puede probar los gráficos 3D usando glmark2-es2 y es2gears con resultados espectaculares:

```
$ glmark2-es2  
$ es2gears
```

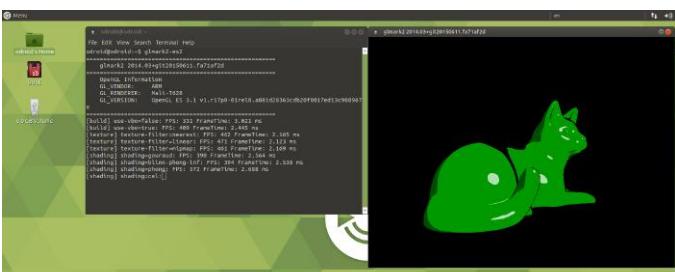


Figura 10

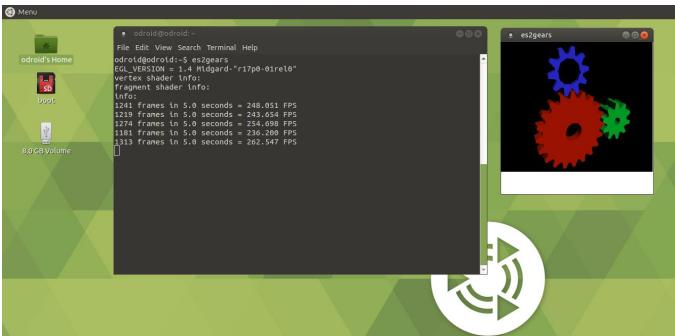


Figura 11

Finalmente, probaremos la función WebGL dentro del navegador Chromium dirigiéndonos a la página de muestra de WebGL en <http://webglsamples.org/>. Conseguimos casi 30 FPS con 500 peces en el tanque en la prueba de acuario disponible en <http://webglsamples.org/aquarium/aquarium.html>, que es una experiencia satisfactoria para una placa tan “pequeña”.

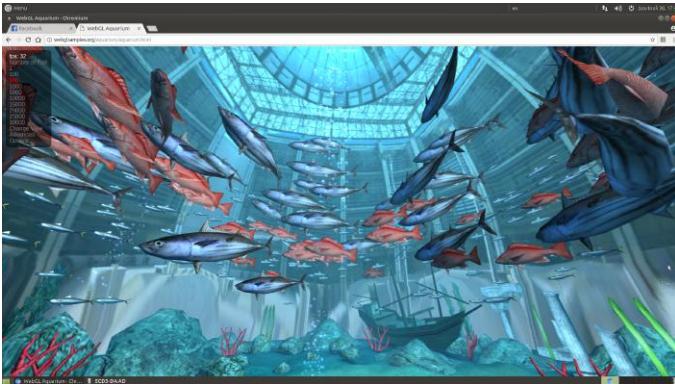


Figura 12

Problemas conocidos

De acuerdo con la página wiki de Hardkernel (<https://goo.gl/wuV9Vx>) debemos tener en cuenta las siguientes cuestiones

1. El acceso a la GPU Mali podría ser bloqueado por un paquete EGL reciente de Canonical. En este caso, necesitarás instalar el driver de Mali manualmente.

```
$ sudo apt-get install mali-x11 --reinstall
```

2. VLC no se inicia debido a un problema de compatibilidad con el driver de video. La razón es que VLC no es compatible con una plataforma genérica ARM Linux. Las actualizaciones recientes lo dejaron aún peor. Probé con un archivo H.264 en Kodi y funcionó sin ningún problema. Sin embargo, los archivos MPEG4/MPEG2 no funcionan bien con el decodificador de hardware. Tenía muchos fallos. El sistema operativo en sí no tiene nada que ver con esto. Forma parte de los programas que se ejecutan en él. Algunas investigaciones han revelado que Ubuntu 18.04 utiliza Kodi con hardware acelerado ffmpeg en lugar del decodificador de hardware MFC de Oversun. La implementación de ffmpeg parece no ser muy estable, mientras que MFC en Kodi hasta la versión 17.6 ha funcionado bastante bien. Las versiones más antiguas de Ubuntu, como la imagen 16.04, aún utilizan la versión de Kodi que usa MFC en lugar de ffmpeg, lo cual hace que se ejecute de manera más estable. Tendremos que esperar hasta que Hardkernel solucione el problema. 3. La primera vez que ejecutamos Chromium necesita alrededor de un minuto debido al proceso de inicialización interna. Después de eso, arrancará en 1-2 segundos. 4. Algunos controladores de comunicación USB a SATA que soportan UASP no son compatibles. Si tu disco duro externo no funciona correctamente, según Hardkernel, agrega su VID/PID en boot.ini. Utiliza los argumentos para deshabilitar la función UAS de la siguiente forma:

```
$ usb-storage.quirks=0x0bc2:0x2322:u
```

5. Cambia el regulador de la CPU a “ondemand”, ejecuta este comando y reinicia: \$ echo ‘GOVERNOR=”ondemand”’ | sudo tee /etc/default/cpufrequtils; systemctl mask ondemand; A continuación, instala la extensión “h264ify” desde Chrome Web Store para el navegador Chrome, que permite mejorar la calidad del video de Youtube en el navegador Chromium. La mejoría es sustancial, los videos 720p/30fps se reproducen bastante bien con la extensión.

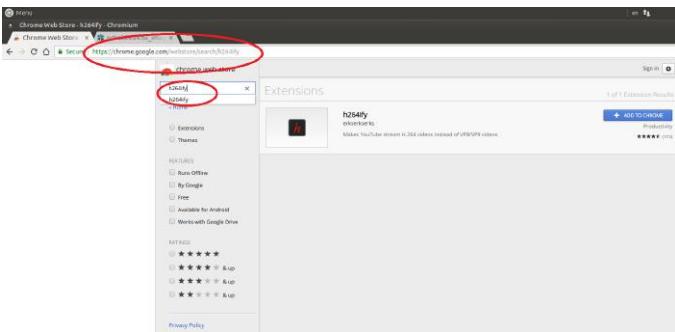


Figura 13

Ventajas Ubuntu 18.04 es bastante estable en ODROID XU4. Los programas se ejecutan muy rápidamente y la navegación por internet es muy fluida. El rendimiento de la placa con gráficos acelerados en 3D funciona bastante bien en demos y en algunas demostraciones de WebGL. La decodificación de hardware H.264 en Kodi y FFmpeg

funciona bien hasta 1080p, y en Youtube hasta 720p, Los tiempos de carga de los programas son muy rápidos gracias al módulo flash eMMC (<https://goo.gl/r5Zi8M>), y la capacidad multitarea es decente. En resumen, podríamos decir que Ubuntu 18.04 en la placa ODROID-XU4 ha sido una experiencia muy satisfactoria, muy similar a la de un ordenador de escritorio. A pesar de algunos problemas de rendimiento con la reproducción de archivos AVI, en la mayoría de los casos el sistema funciona bien y de manera fiable con unos tiempos de carga de programas bastante bajos. La codificación de video de hardware, la aceleración de gráficos 3D para OpenGL ES y la mayoría de las funciones funcionan como era de esperar. ¡Darle una oportunidad!

Kali Linux 2018.2 en el ODROID-XU4

© September 1, 2018 By Philipp Rutherford ➔ Linux, Tutoriales



Nota para nuestros lectores: la configuración detallada en este artículo se ha realizado con una pantalla táctil y un adaptador Alfa AWUS1900 Wi-Fi compatible.

Kali Linux es uno de los mejores sistemas para pruebas de penetración. Está disponible para varios dispositivos basados en tecnología ARM y por un corto periodo de tiempo también estaba disponible para el ODROID-XU4 en el sitio web oficial de Kali. Lamentablemente, la imagen ya no está disponible. En este artículo vamos a cubrir las siguientes cuestiones:

- Cómo instalar Kali Linux 2018.2 en ODROID-XU4
- Cómo conseguir que la pantalla táctil reconozca los gestos multitáctiles
- Cómo hacer que el adaptador Alfa AWUS1900 funcione con ODROID-XU4

- Tu asumes la plena responsabilidad de tus acciones
- Esta guía es fácil de seguir, sin embargo, no te salte ninguna sección por fácil que te parezca y léelas todas hasta la última.
- Asegúrate de hacer una copia de seguridad de tus datos
- No te preocupes por los mensajes de advertencia durante la ejecución del comando make
- Pueden aparecer algunos fallos, aunque actualmente no se ha detectado ninguno. No te olvides de publicar cualquier problema que encuentres en el hilo del foro en <https://forum.odroid.com/viewtopic.php?f=96&t=31737>
- El autor permite que cualquier persona comparta el material que figura a continuación, pero se debe incluir este artículo como fuente.

Descargo de responsabilidad



Figura 1 – Kali Linux 2018.2 en ODROID-XU4 con el adaptador Wifi Alfa conectado

Instalación Descarga la imagen ‘Kali Linux Odroidxu3’ desde sitio web oficial de Kali Linux: <https://www.offensive-security.com/kali-linux-arm-images/> y grábala en un módulo eMMC o tarjeta microSD. Asegúrate de que tu XU4 esté conectado a Internet. Arranca tu XU4 e inicia sesión utilizando el usuario ‘root’ y la contraseña ‘toor’. Elije usar la configuración por defecto en el Panel. Si estás usando una pantalla multitáctil, como el ODROID-VU5, notarás que los gráficos aparecerán con fallos y que la pantalla táctil no reacciona al tocarla. No obstante, los puertos USB 3.0 deberían funcionar bien. Vamos a compilar y actualizar el kernel de Linux a v4.14, lo cual mejorará significativamente el rendimiento general y solucionará los fallos que hemos comentado.

¡ADVERTENCIA! Si está usando una pantalla multitáctil, asegúrate de que tu XU4 no entre en modo reposo durante la actualización del kernel, de lo contrario, no podrás encenderla sin tener que reiniciar. Si tu pantalla empieza a parpadear, significa que el XU4 está a punto de entrar en modo reposo. Se recomienda aumentar temporalmente los tres valores en [Applications > Settings > Power Manager > Display] a su valor máximo.

Inicia el terminal y ahora estamos listos para empezar a prepararlo todo. Lleva a cabo el proceso básico de actualización del software y luego instala algunos paquetes que son necesarios para continuar con los siguientes pasos:

```
$ apt update && apt dist-upgrade -y  
$ apt install gcc g++ build-essential libssl-dev bc -y
```

Una vez hecho esto, comenzaremos a compilar y actualizar el kernel:

```
$ cd /usr/src  
$ git clone --depth 1  
https://github.com/hardkernel/linux -b  
odroidxu4-4.14.y kernel-4.14  
$ cd kernel-4.14  
$ make odroidxu4_defconfig  
$ make -j8
```

Nota: la ejecución del comando anterior durará aproximadamente unos 30 minutos.

```
$ make modules_install  
$ mount -o rw /dev/mmcblk0p1 /boot  
$ cp -f arch/arm/boot/zImage /boot  
$ cp -f arch/arm/boot/dts/exynos5422-  
odroid*dtb /boot  
$ sync
```

Ahora tenemos que elegir los drivers adecuados para la GPU. Existe un paquete mali-t62x-x11-driver disponible a través de apt, pero el XU4 presenta algunos fallos de dependencia al instalarlo. Por lo tanto, optaremos por el driver del repositorio de Ubuntu de Hardkernel y lo adaptaremos para que funcione con Kali Linux:

```
$ wget -O mali http://deb.odroid.in/5422-  
s/pool/main/m/mali-x11/mali-x11_20180717-  
r17p0-a52903b-11_armhf.deb  
$ dpkg -i mali && rm mali  
$ cd /usr/lib/arm-linux-gnueabihf  
$ sos='libEGL_mesa.so.0.0.0 libEGL.so.1.1.0  
libGLESv2.so.2.1.0'  
$ for so in $sos; do rm -rf $so; ln -s mali-  
egl/libmali.so $so; done
```

Ahora el driver está listo. Elimina los paquetes innecesarios y reinicia:

```
$ apt autoremove -y  
$ reboot
```

Tras reiniciar el XU4, verifica que el kernel se haya actualizado correctamente:

```
$ uname -r
```

Deberías ver 4.14.55+. La pantalla táctil y el resto de módulos deberían funcionar en este momento. Disfruta de tu Kali Linux actualizado en el XU4.

Reconocimiento de gestos multitáctiles

Aunque la pantalla táctil ahora funciona, solo es capaz de soportar toques individuales. Vamos a enseñarle a reconocer gestos multitáctiles. Existe una herramienta llamada 'toucheegg', que permite gestionar la actividad multitáctil, aunque presenta errores con el modo individual. Por lo tanto, instalaremos otra herramienta llamada 'tatchi', que nos permitirá cambiar entre el modo simple (por defecto) y el multitáctil (toucheegg) con solo un clic. Instalar toucheegg es un poco complicado debido a sus dependencias. Así que, vamos a hacernos con ella:

```
$ apt install netselect-apt libqt4-dev  
libxtst-dev libxv-dev multiarch-support -y  
$mirror=$(echo $(netselect-apt jessie -a  
armhf -o /dev/null) | awk '{print  
$1;}')pool/main/g/geis/libgeis"  
$debs='1 -dev'  
$for deb in $debs; do temp=$(mktemp); wget -O  
$temp $mirror$deb'_2.2.16-1+b1_armhf.deb';  
dpkg -i $temp; apt install -f -y; rm $temp;  
done
```

Ahora estamos listos para instalar touchegg:

```
$ git clone  
https://github.com/JoseExposito/touchegg  
$ cd touchegg  
$ qmake  
$ make  
$ make install
```

Ahora tenemos que instalar tatchi:

```
$ git clone https://github.com/mopo3ob/tatchi  
$ cd tatchi  
$ make install add  
$ cd ../../ && rm -r touchegg
```

Un ícono en forma de mano debería aparecer en la esquina superior derecha del panel. Tócalo una vez para cambiar al modo multitouch. El ícono ahora cambiará de un dedo a dos dedos. Esto significa que touchegg está listo para reconocer gestos

multitáctiles. Intenta desplazar el terminal con dos dedos. Toca el botón de la mano una vez más para volver al modo individual. Es tan simple como eso.

Es probable que quieras personalizar la configuración de los gestos multitáctiles. Puedes hacerlo editando `~/.config/touchegg/touchegg.conf`. Visita la wiki de touchegg para obtener más información: <https://github.com/JoseExposito/toucheegg/wiki>.

Configurar el adaptador WiFi Alfa AWUS1900

Ahora que tenemos un XU4 totalmente operativo con la última versión de Kali Linux, es hora de ponernos manos a la obra con el adaptador Alfa AWUS1900, considerado como el mejor adaptador WiFi disponible para las pruebas de penetración. El AWUS1900 permite velocidades de conexión de router de hasta 1900 Mbps, lo cual requiere un puerto USB 3.0. El ODROID-XU4 es uno de los pocos microordenadores que pueden usar una antena cuádruple tan potente. Para que funcione, simplemente necesitamos instalar el driver del chipset Realtek RTL8814U, que es el chipset que usa el adaptador AWUS1900.

```
$ git clone https://github.com/aircrack-ng/rtl8812au -b v5.2.20  
$ cd rtl8812au  
$ make ARCH=arm  
$ make install  
$ cd .. && rm -r rtl8812au
```

La instalación del driver ha finalizado. Conecta tu AWUS1900 y se encenderá un led azul.

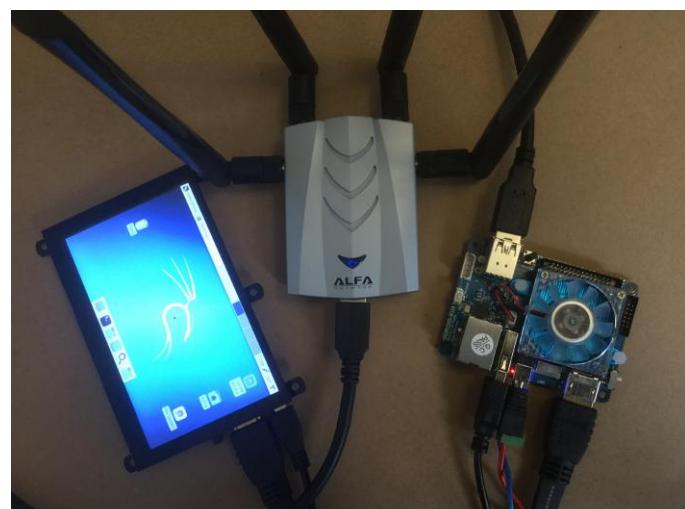


Figura 2 – Indicador Led azul en el adaptador WiFi Alfa AWUS1900

Ahora usaremos ifconfig con nuestro AWUS1900:

```
$ apt install net-tools  
$ ifconfig
```

Ahora podrás ver wlan0, siempre que estés utilizando un cable Ethernet para Internet. Para poner AWUS1900 en modo monitor manualmente, ejecuta los siguientes comandos:

```
$ airmon-ng check kill  
$ ip link set wlan0 down  
$ iw dev wlan0 set type monitor  
$ ip link set wlan0 up  
$ service NetworkManager restart
```

Pruébalo realizando un búsqueda de puntos de acceso:

```
airodump-ng wlan0
```

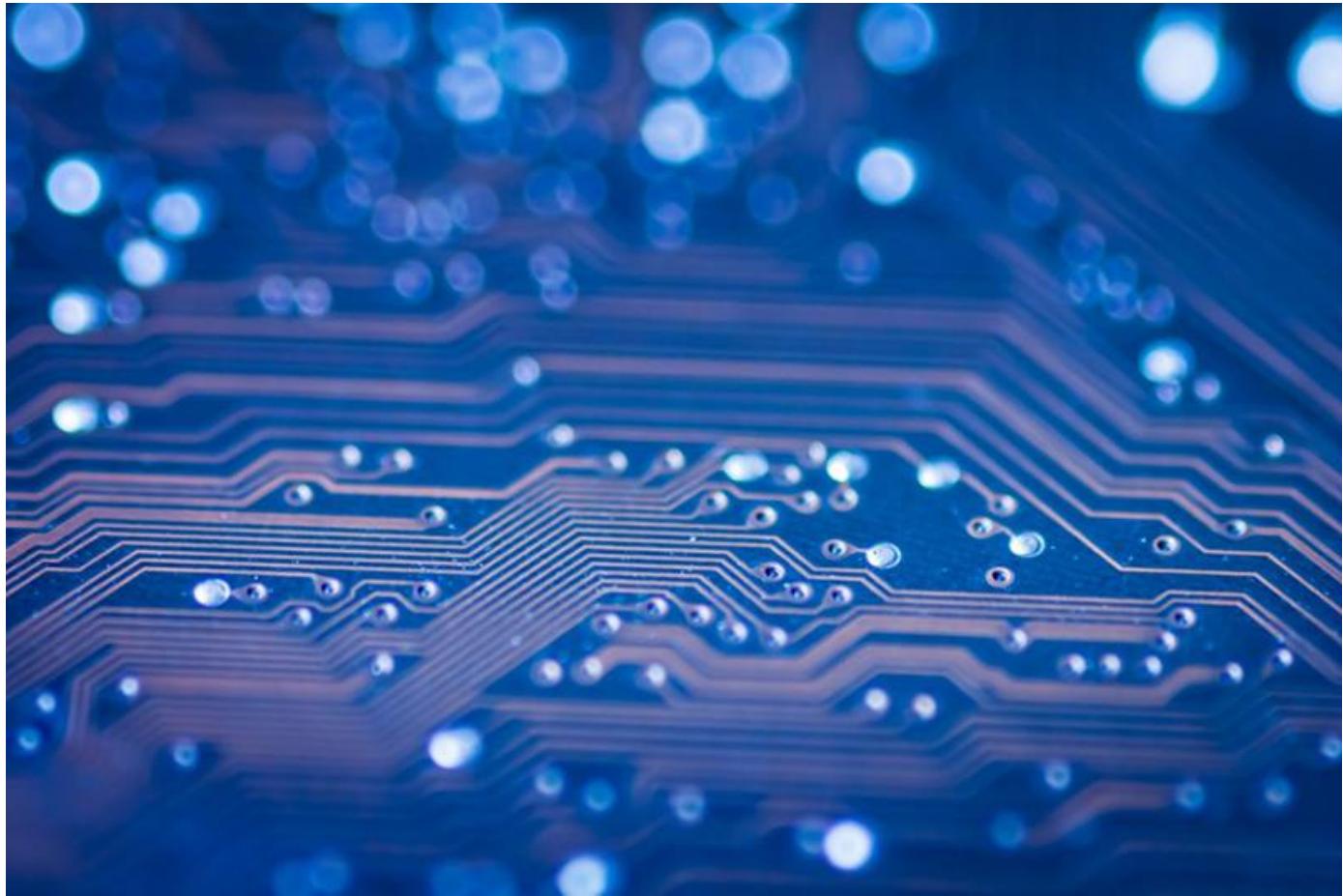
Debería ser capaz de detectar puntos de acceso cercanos. Pulsa ^C (Control + C) para detener el escaneo.

Ahora contamos con uno de los microordenadores más potentes que ejecuta Kali Linux 2018.2 y que funciona con una pantalla táctil ODROID y un adaptador wifi Alfa AWUS1900. Le doy las gracias a @odroid, @robroy y @mad_ady por ayudarme a encontrar soluciones a ciertas cuestiones tratadas en este artículo.

Para comentarios, preguntas y sugerencias, visita el hilo del foro ODROID en <https://forum.odroid.com/viewtopic.php?f=96&t=31737>.

Usando ODROID en Informática de Alto Rendimiento (HPC) – ARM Frente a la Competencia

© September 1, 2018 By Kurt Keville, MIT Sin Categoría



Un centro de datos moderno usa demasiada electricidad y aire acondicionado para funcionar de manera eficiente. Un proveedor de servicios de Internet con tecnología ARM puede llegar a ofrecer páginas web con mucha menos energía que las arquitecturas convencionales

(<http://tinyurl.com/ApacheOnARM>). Al mismo tiempo, los núcleos ARM están siendo rápidamente utilizados por la comunidad científica de Datacenter porque éstos son más rápidos con operaciones intensivas en matemáticas de punto flotante, por una décima parte de los costes de energía, y su hoja de ruta muestra que estar por llegar una arquitectura con un mayor rendimiento por vatio. También existe un creciente interés en HPC (Informática de Alto Rendimiento) y sus usos en dominios más amplios de la comunidad de desarrolladores de ARM. En el entorno académico, también se habla de la denominada Informática Extrema de Alto

Rendimiento (<http://www.ieee-hpec.org/>) y de la Supercomputación IEEE, que fueron probadas en la reciente conferencia SC13 en Denver, Colorado. Además, el clúster IEEE, con el que se contó en Indiana en 2013, ha propiciado un notable aumento de las publicaciones centradas en ARM.

La mayoría de los modernos centros de supercomputación tienen miles o decenas de miles de núcleos dedicados a sus necesidades particulares de procesamiento. Cada vez que un programador quiere ejecutar una aplicación con una relación de rendimiento mejorada (por vatio, dólar o metro cuadrado) supone una gran ventaja para las partes implicadas del Datacenter, así como para los clientes de dominio informático-científico que soporte. Este es el caso en el que cada vez son más frecuentes las situaciones en las que las aplicaciones deben ejecutarse varias veces y en las que múltiples

aplicaciones comparten recursos en HPC, de forma muy similar a la nube volviéndose al uso compartido de recursos al viejo estilo.

¿Por qué ARM?

La arquitectura ARMv7 ha demostrado estar a la altura de los desafíos del HPC de diversas formas en las que anteriores arquitecturas ARM no lo consiguieron. Uno podría usar ARM al margen de la eficiencia energética, ya que esta tecnología representa una ruta de crecimiento muy rápida en lo que respecta a la informática integrada. Cuando un Datacenter se compone de más de 10.000 núcleos, se consiguen considerables ventajas a través de mejoras graduales. Estos pequeños cambios pueden sumar ahorros significativos en cuanto a espacio, potencia y refrigeración. Cuando la memoria se comparte entre la CPU y la GPU en SoCs ARM (Sistema en un Chip), se duplica las extensiones SIMD (Single Instruction, Multiple Data) en la GPU NEX Cortex-A15 y se cuenta con un potencial acceso a la memoria considerablemente mayor, se obtienen beneficios en el lugar donde más los necesitamos; donde la aplicación y los conjuntos de datos se entrecruzan. Con la creciente aceptación de GP-GPU (GPU de propósito general) y la expansión de aplicaciones de tipo HPC basadas en aplicaciones de big data, los rápidos modos de computación ARM son más relevantes que nunca, con una ruta tecnológica hacia una participación del HPC en constante expansión.

¿Por qué ODROID?

Hoy en día, ODROID tiene un procesador de la familia Exynos con al menos 4 núcleos ARM. La próxima serie Exynos5 tiene 8 núcleos, 4 de los cuales son ARM Cortex-A15. Hardkernel, al igual que el consorcio de investigación de RunTime Computing Solutions, han demostrado mejoras sustanciales de potencia y rendimiento del XU en comparación con otras arquitecturas modernas. Con el ambicioso calendario

de lanzamiento de nuevas tecnologías de Hardkernel, los usuarios de esta plataforma siguen la ley de Moore y el lanzamiento de ARM de manera eficiente, permitiéndoles unirse a la ola de sistemas novedosos, mejores, de bajo coste y con mayor rendimiento a medida que aparecen con una significativa compatibilidad ascendente.

Además, el Cortex-A15 gana en la mayoría de pruebas de rendimiento ARM-HPC (por ejemplo, las NAS Parallel Benchmarks en <http://tinyurl.com/ODROID-HPC>). El equipo de RunTime Computing Solutions ha demostrado recientemente ventajas muy significantes del A15 frente al A9 en el HPCC Challenge, el estándar de evaluación preferido para HPC (<http://hpcchallenge.org/>). Esta prueba sólo utiliza los núcleos A15 en el Exynos 5410 y los mantiene a la máxima capacidad durante toda la prueba (<http://tinyurl.com/ODROID-LINPACK>) lo cual no es una comparación completamente equitativa. No obstante, se puede extraer cierta información significativa de ella; XUJessie es doblemente mejor que U2Whisper en G-HPLINPACK, la primera prueba. Los autores mostraron el SOX BOX en SC13 incluyendo muchas modificaciones de hardware que mejoraron el rendimiento. Echa un vistazo a nuestros sitios para más información.

Conclusión

Actualmente podemos ejecutar muchas aplicaciones de HPC en ODROID, y como las demandas al alza de la eficiencia energética provocan que los profesionales de la industria reconsideren el diseño de los Datacenter, progresivamente más centros adoptarán estas arquitecturas. El futuro es brillante para los diseñadores, ya que la tendencia hacia la computación escalable marca el inicio de un nuevo y emocionante tópico en las tecnologías SoC integradas.

Conociendo un ODROIDian: Andrew Armstrong

© September 1, 2018 By Rob Roy Conociendo un ODROIDian



Por favor, háblanos un poco sobre ti. Soy el CTO y cofundador de Cadonix, la primera solución de CAD eléctrica en la nube del mundo totalmente basada en el navegador. Tengo una amplia variedad de conocimientos que van desde la investigación científica, sistemas integrados, diseño electrónico, ingeniería de software y fabricación. Soy muy afortunado ya que todo esto me ha permitido trabajar con algunas de las compañías más interesantes del mundo en sus nuevas y emocionantes tecnologías. De igual manera, disfruto mucho de la incipiente comunidad de entusiastas de la electrónica, la informática y los juegos que encuentro a través de mi canal de YouTube, donde trabajo en muchos proyectos de electrónica. Vivo en el Oxford rural del Reino Unido. Es una excelente ubicación para una empresa de tecnología ya que no solo está cerca de Londres, sino que también es uno de los centros científicos clave del Reino Unido, especialmente por nuestra emergente industria de

vehículos eléctricos y espaciales. ¡Hay mucha gente inteligente por aquí si es lo que vas buscando!



Figura 1 – Stuart Ashen (también conocido como Ashens <http://www.youtube.com/ashens>) es un famoso Youtuber, jugador y productor/actor de películas. Esto fue en PLAY EXPO London 2018. Estuvimos charlando sobre ODROID-GO puesto que a Stuart le encanta testear el hardware de los sistemas de juegos y es bastante conocido por sus análisis y película sobre "Gamechild")

En la mayoría de las materias, fui un estudiante de secundaria bastante mediocre (probablemente por

debajo del promedio) que no llegó a conseguir las calificaciones suficientes para la universidad deseada. Afortunadamente, me armé de coraje para visitarlos en persona y finalicé un curso de Ingeniería de Software que completé con un título de Honores de 1ra Clase. Sin embargo, la cosa no terminó ahí, ya que la Universidad amablemente me patrocinó su programa de Doctorado, y unos años más tarde terminé con un doctorado, centrado concretamente en el campo de la compresión de la imagen. Luego pasé a una beca de investigación posdoctoral de la Universidad de Loughborough para trabajar en la compresión de video H.264. Unos 5 años más tarde, fui acreditado como Ingeniero Colegiado. Siempre animo a la gente a que siga intentando y a que encuentre aquello que realmente le apasiona. ¡Para mí fue, y sigue siendo, conocimiento!



Figura 2 – Heyford Air Base 10K 2018 – El Dr. A y el entrenador Ali Gilbrath destrozando sus mejores marcas personales en los 10k



Figura 3 – Kids Heyford Air Base 10K 2018 – Mis dos hijos pequeños Alex y Theo también participaron y ganaron medallas en su categoría, ¡Fue el día en el que me sentí más orgulloso de mi vida!

Vivo con mis dos hijos pequeños Alex y Theo, mi esposa Sarah y nuestro gato Ziggy. Sarah trabaja en Oxford Brooks University ayudando a las jóvenes mentes a elegir el camino formativo adecuado. A veces puede resultar un poco ajetreado porque tengo muchos viajes internacionales como parte de mi función, pero solemos encontrar las formas de hacerlo funcionar con una familia razonablemente cercana que nos respalda.

¿Cómo empezaste con los ordenadores? Me presentaron por primera vez a los ordenadores a los 3 años. Era un TRS-80 que mi padre, que trabajaba para ICL en ese momento (la versión de IBM del Reino Unido), había comprado a un amigo. Recuerdo ver pacientemente el asterisco parpadeando en la esquina de la pantalla mientras esperaba a que un reproductor de cassettes cargara mi pasatiempo favorito, Dancing Demon. Después de esto, fue un continuo venir de ordenadores y consolas de 8 y 16 bits a lo largo de los años, desde BBC Micro, Commodore 64, Atari ST, PC, Apple Macs, etc.



Figura 4 – Neil de retromancave es otro Youtuber de viejos ordenadores y dispositivos portátiles. Tenemos un podcast llamado Retro Island Diskettes



Figura 5 – Jarad y las Back to the future Sneaker – Jarad es un amigo creador, estamos posando junto a una nueva versión de las legendarias zapatillas Nike Back to the Future

Probablemente las máquinas más comunes de nuestros hogares fueron los diversos PC que siempre existieron en mi entorno, ya sea la humilde IBM AT o la exótica Compaq Portable III, que posiblemente fueron la base de mis conocimientos informáticos. Puedo afirmar que aprendía a programar con cada máquina que llegaba a tener.

¿Qué te atrajo a la plataforma ODROID? Me avergüenza un poco decir que prácticamente desconocía el hardware ODROID hasta hace poco, recurriendo a sus primos omnipresentes en mis proyectos. Sin embargo, estoy realmente interesado utilizar el ODROID-XU4 como base para futuros proyectos. Las he pasado canutas con muchos ordenadores de placa reducida, de modo que es importante encontrar el

equilibrio adecuado entre potencia y soporte, mi reciente participación en la comunidad ODROID me ha dado confianza en la plataforma.



Figura 6 – Design Stream Rig – Esta es la plataforma que configuré para grabar los diversos videos de ODROID-GO donde diseñamos y construimos los primeros prototipos de cascos de audio

¿Cómo usas tus ODROIDs? ¡Uso mis ODROID para jugar! ¿A quién no le gusta echar a andar viejos emuladores y jugar a algunos de los viejos títulos de su infancia? Es fácil ver esto como un uso bastante trivial de la tecnología sofisticada, pero creo que es un caso de uso bastante sorprendente, ya que ha propiciado que muchas personas empiecen a interesarse por este hardware e incluso lleguen a desarrollar sus propias consolas arcade.



Figura 7 – Odroid Audio Hard – Parte del primer lote de unidades que ensamblé yo mismo



Figura 8 – Odroid Audio Hard montados – Detalle de las placas v1 ensambladas a mano

Desde un punto de vista más formal, desarrollo una gran cantidad de componentes electrónicos integrados, especialmente en el área de la monitorización, adquisición de datos y control. Muchos de ellos se encuentran en la industria del automóvil, donde el rendimiento es clave, especialmente en el desarrollo de los vehículos más innovadores y avanzados. Estoy realmente interesado en explorar cómo aprovechar el hardware para esto. Incluso estoy desarrollando una interfaz de adquisición de datos para el ODROID-GO, puesto que estamos ante una excelente unidad portátil.



Figura 10 – Mi primer ordenador TRS-80 – Mi primer ordenador, recuperado recientemente de un ático y en muy mal estado, ¡He cambiado los chips de memoria, pero necesito un poco más de TLC!

¿Cuál es tu ODROID favorito y por qué? ¡Realmente me gusta el ODROID-GO! Desde que lo vi por primera vez, simplemente tenía que tener uno, así que lo pedí de inmediato. Tuve mucha suerte ya que recibí mi

unidad al día siguiente y la monté con mucho entusiasmo. Hay dos razones por las que realmente me gusta la plataforma: 1) puedes ejecutar juegos, el primer uso que le di fue el de jugar mientras haces escala de aeropuerto en aeropuerto, y 2) puedes modificar el hardware. Tengo un video muy popular sobre cómo montar una toma de auriculares analógica para el ODROID-GO, para aquellos de nosotros que queremos usarlo en lugares públicos. Recientemente he empezado a seguir los foros, y en una serie de Live YouTube, la comunidad de backoffice diseñó una mejora sin soldadura para agregar esta capacidad al GO. La comunidad que trabaja en los firmwares ODROID-GO está haciendo grandes cosas, especialmente gente como crashoverride que trabaja incesantemente para mejorar la plataforma. Claramente el hecho de poder modificar el hardware y encontrar soluciones con amigos por Internet es muy divertido y supone un gran atractivo para la plataforma.



Figura 11: un insólito vistazo a algunas de las máquinas que se esconden en el Backoffice: ¿cuántas puedes ver?

¿Qué innovaciones te gustaría ver en futuros productos de Hardkernel? Es complicado, ya que Hardkernel parece estar innovando como un loco, y tengo problemas para seguirle el ritmo. Sin embargo, me encantaría ver una placa interfaz para automóviles y una carcasa realmente bonita. Conozco a mucha gente que está intentado desarrollar (yo hacerme que construya!) herramientas de diagnóstico de vehículos utilizando el bus CAN y el bus LIN, o personas que están tratando de crear sistemas de entretenimiento o de control para vehículos. Existe un gran mercado

para las personas que quieran modificar sus vehículos, y una plataforma suficientemente testeada sería realmente de gran ayuda.



Figura 12 – Me gusta modificar viejos ordenadores y ver si puedo desarrollar interfaces interesantes para ellos. Usaré mi ODROID-GO para transferir datos hacia y desde este Amstrad CPC

¿Qué pasatiempos e intereses tienes aparte de los ordenadores? Cualquiera que vea mi canal de YouTube o esté en mi grupo de Discord perderá el interés fácilmente, y los pasatiempos parecen ir y venir mensualmente. Los que más me han durado son el motociclismo, trabajar en mis coches y el running. Este último está relaciona con el científico que llevo dentro, ya que me encantan analizar los datos que se obtienen al utilizar dispositivos para el running, desde cuánto peso pones en tu pierna izquierda hasta como de fuerte late tu corazón.



Figura 13 – Uno de los proyectos de verano fue aprender a hacer camisetas, lo cual es muy divertido y un proceso bastante técnico que necesitas conocer para saber exponer la pantalla a la luz ultravioleta y así poder plasmar tu diseño.

¿Qué consejo le darías a alguien que quiere aprender más sobre programación? Si alguien llegara a mí sin conocimientos de programación, les diría que realmente lo importante es aprender los aspectos básicos. Para ser realmente un gran ingeniero de software, lo que necesitas es comprender cómo funciona el hardware. Puedes decidir qué vas a ser un programador web, pero todos estos fundamentos que te permiten ejecutar tu código en dispositivos básicos te ayudarán a lo largo de tu carrera, ¡así que empieza con una calculadora! Si pudieran hacerse con un viejo micro-ordenador como BBC Micro, Commodore 64 o Amstrad CPC, y trabajar con guías de programación básica y avanzada, creo que les sería muy útil para moverse entre los diferentes

lenguajes que existen. Exceptuando esto, yo recomendaría descargar Lua o uno de los entornos empaquetados. Éste ha sido mi lenguaje favorito de todos los tiempos en términos de facilidad de uso y aprendizaje. Tiene la simplicidad de BASIC, pero ofrece potencial casi ilimitado en las manos correctas; simplemente mira lo que las personas están haciendo con PICO-8.



Figura 14 – ¡Nunca dejo pasar la oportunidad de poner una cara de tonto en las redes sociales!