

ODROID

Año Seis
Num. #62
Feb 2019

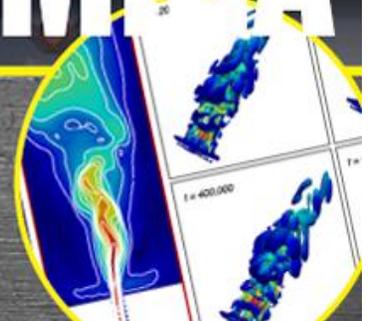
Magazine



Amibian.js EMULACION COMMODORE AMIGA

KERNEL 5.0:
ULTIMA VERSION DEL KERNEL
DE LINUX PARA TU ODROID

CLUSTER MCI:
COMO EJECUTAR PUNTERAS
SIMULACIONES CON ODROIDS

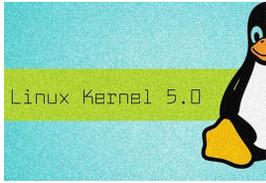




Clon de ODROID-GO

© February 1, 2019

El ODROID-GO es un gran invento de Hardkernel.



Kernel 5.0: La Próxima Versión LTS del Kernel de Linux

© February 15, 2019

¡El Kernel Linux 5.0, la próxima versión LTS del Kernel de Linux, ya está disponible para los ODROIDS!



Cálculos Científicos en Clúster con un ODROID-MC1

© February 1, 2019

Las simulaciones comunes de última generación emplean cientos de miles de núcleos de CPU en sistemas de computación de alto rendimiento (HPC) para resolver grandes desafíos sociales. En este contexto, la escalabilidad y el rendimiento del kernel de simulación son clave para una computación multi-núcleo eficiente



Amibian.js: Emulando un Commodore Amiga en un clúster ODROID-XU4

© February 1, 2019

Amibian es lo todo que necesitas para convertir tu ODROID en un Amiga. Es una imagen de tarjeta SD muy liviana apta para tarjetas SD de 2 GB en adelante. Está diseñada para ofrecerte la mejor experiencia de Amiga que puedes conseguir sin tener un Amiga real. Amibian te va [▶](#)



Campamento de Programación – Parte 11: Controlar el LED desde tu Smartphone a través del WiFi

© February 1, 2019

ESP32, que es usado en ODROID-GO, es compatible con WiFi 802.11b/g/n, de modo que podemos programar servicios WiFi con prácticas librerías en Arduino



OGO-FTPD: Un Servidor FTP para el ODROID-GO

© February 1, 2019

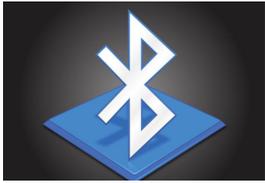
En este artículo, presentaré un servidor FTP para ODROID-GO. Es una implementación más bien minimalista que actualmente no admite la autenticación y el funcionamiento en modo pasivo.



Juegos Android

© February 1, 2019

Rob me preguntó si tenía vicheados algunos juegos que acabase de jugar en mi ODROID con Android y ¡por suerte contaba con algunos! ¿Quién podría imaginarse que juego en mi tiempo libre?



Campamento de Programación – Parte 12: Comunicación Serial a través de Bluetooth

© February 1, 2019

En este artículo, llevaremos a cabo una conexión inalámbrica a nuestro smartphone utilizando el protocolo RFCOMM de Bluetooth.



Pack de Juegos ODROID-GO: Pack de Descarga de Aplicaciones y Juegos de Terceros

© February 1, 2019

Este artículo contiene todas las aplicaciones y juegos de terceros que se han ido lanzando hasta el momento para el ODROID-GO por los diferentes colaboradores



Juegos Linux: PC-Engine/TurboGrafx – Parte 2

© February 1, 2019

Realmente he disfrutado probando todos estos juegos de PC-Engine/TurboGrafx CD



Conociendo un ODROIDian: Chris Lord

© February 1, 2019

oy un ingeniero de software y músico, la mayor parte de mi vida me dedico a lo primero. Actualmente, estoy trabajando en un sistema de captura de movimiento en tiempo real para dispositivos integrados, pero he trabajado en todo tipo de cosas en la

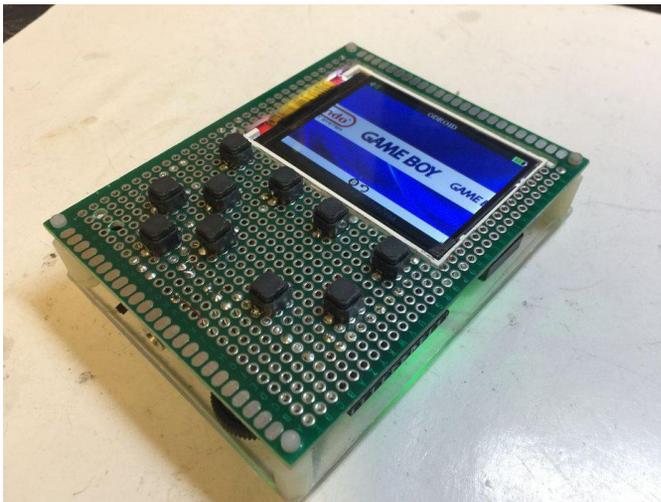
última década.

Clon de ODROID-GO

© February 1, 2019 By @cheungbx ↳ ODROID-GO, Mecaniquero, Tutoriales



El ODROID-GO es un gran invento de Hardkernel. Puede funcionar como una consola de juegos con muchos emuladores. Aunque el número de emuladores es pequeño en comparación con los que son compatibles con RetroPie o Recalbox instalados, descubrí que responde mucho más rápido y suave, además de consumir mucha menos energía. Poder guardar los juegos por defecto también es muy útil.



Puedes encontrar un video de introducción sobre mi ODROID-GO casero en: <https://youtu.be/E3ZfBwI9Wt8>.



Se puede ver una demostración completa de los pasos en: https://youtu.be/sP3x_Fm-htA



No he podido comprar el ODROID-GO a un precio razonable en Hong Kong, debido al alto coste de los envíos de los EE. UU., Por lo que decidí crear uno por mi cuenta. Está muy bien que Hardkernel haya compartido los esquemas de diseño y el firmware en github y lo hagan de código abierto, de modo que los creadores como yo podemos retornar a nosotros mismos para fabricar nuestro propio diseño de ODROID-GO.

Al poder cargar diferentes firmwares, puedes usarlo para ejecutar códigos binarios diseñados para Arduino, códigos micropython y muchos otros tipos de software de terceros. Aunque este artículo te muestra los pasos para hacerlo, no todo el mundo puede desarrollar uno con éxito, ya que requieren ciertas habilidades para hacer soldaduras y retirarlas. También recomiendo a cualquier persona que desee tener un ODROID-GO que lo compre en Hardkernel, ya que el coste de fabricarlo (más todo el envío) puede ser similar o incluso superior al de comprarlo directamente a Hardkernel.

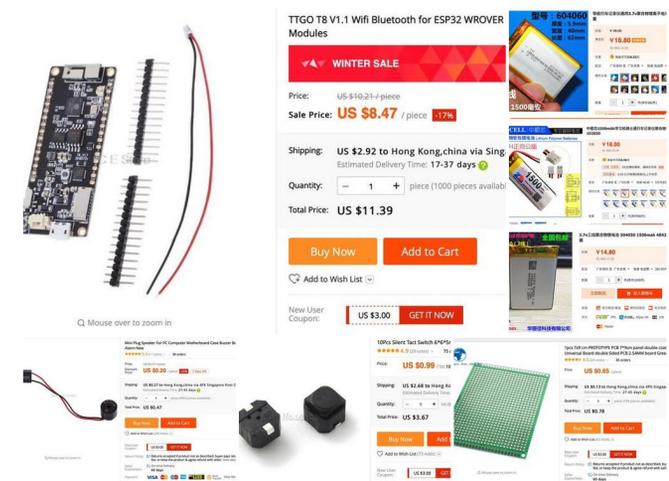
Tengo la intención de montarle una LCD TFT de 2.2" más pequeña en lugar de usar una de 2.4" que es la que viene con los originales. Tampoco necesito una salida de audio tan potente, la salida de los auriculares es lo suficientemente potente para mí, de modo que no necesito el circuito del amplificador. Simplemente conecte el GPIO 26 y la conexión a tierra del ESP32 al conector de los auriculares o mini/altavoz con un control de volumen por hardware. Esto es todo lo que necesito. Conservé el cabezal de extensión de 10 pines, aunque tengo que usar mi propio juego de botones silenciosos.

Está fabricado sobre un prototipo de PCB como estructura básica, sobre la que tallé una ventana rectangular para montar la LCD, de manera que

quedara plana sobre la PCB en su parte delantera. Para la placa mcu, elegí la placa TTGO T8 ESP32 Wrover. Esta placa tiene 4MB PsRam adicionales (RAM pseudo-estática) en comparación con la placa ESP VRoom que solo cuenta con 4MB de RAM flash pero no PsRam. El firmware GO-Play de ODROID-GO que controla toda la emulación de los juegos retro no puede funcionar con solo 4 MB de RAM Flash. El firmware GO-Play de ODROID-GO no se puede cargar en solo 4MB flash, así que actualicé la memoria RAM de 4MB montada a 16MB. Me cargué una placa usando una pistola de aire caliente para retirar la memoria RAM original de 4 MB soldada y accidentalmente reventé algunas resistencias que no logré encontrar. Tuve que pedir una nueva y esta vez recurrí a una técnica más segura. Simplemente corte los pines de la RAM de 4MB, luego coloque la nueva RAM de 16MB en la parte superior y la soldé. Es mucho más sencillo de esta forma.

Por favor, sigue los pasos que se describen a continuación para fabricar uno si se te da bien soldar. Si las soldaduras no es lo tuyo, te recomiendo que compres el original.

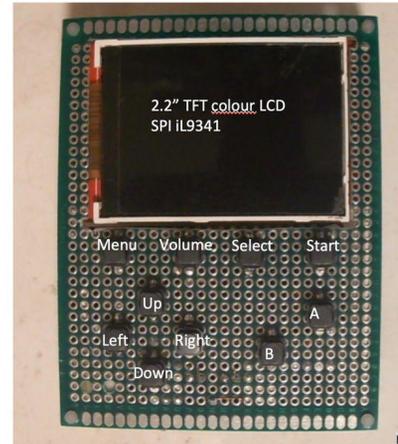
Lista de componentes



La lista de componentes incluye:

- TTGO T8 V1.1 ESP32 Wrover 4Mbyte Flash + 4Mbyte PSRAM
- Memoria flash serie de 16Mb Winbond W25Q128FVSI SOP8 (para actualizar la ESP32 de 4Mb de ram a 16Mb)
- Tarjeta TF de 16G con ranura para tarjeta SD
- TFT LCD SPI i19341 de 2.2"
- Batería LIPO 3.7V 1500MaH

- Mini altavoz
- Toma de auriculares de 3,5 mm con interruptor
- 10 botones silenciosos
- VR 10K para control de volumen
- Cabezal de 10 pines para expansión
- Cabezal de 5 pines y 9 pines para conectar los pines LCD TFT desde las ranuras IC
- Mini interruptor deslizante como botón de encendido
- Prototipo PCB de doble cara de 7cmx9cm
- Placa Arglic de 7cm x 9cm para la contracubierta
- Cuatro tornillos de 3mmx20mm para sujetar la cubierta posterior
- Cable laminado (aislado) de 0.2mm o 0.3mm



La mayoría de todos estos componentes están disponibles en Amazon, AliExpress o TaoBao en China.

Configuración del hardware

Odroid-go pin layout

2.2" TFT

VCC 3V3
 GND GND
 CS- IO5-VSPI CS0
 RESET-RESET
 DC- IO21
 MOSI - IO23-VSPI-MOSI
 SCK - IO18-VSPI-SCK
 LED- IO14
 MISO - IO19-VSPI-MISO

TF Card

CS- IO22 VSPI CS1
 MOSI - IO23 VSPI MOSI
 MISO - IO19 VSPI SCK
 SCK - IO18 VSPI MISO

Audio

Speaker- GND
 Speaker+ - 10K VR- IO26

D-PAD Buttons

tie one end to 3V3
 UP IO35-10K-GND
 Down-10K IO35
 Left IO34-10K-GND
 Right-10K IO34

Other Buttons

tie one end to GND
 Menu IO13
 Volume IO00-10K-3v3
 Select IO27
 Start IO39, VN, -10K-3v3
 B IO33
 A IO32

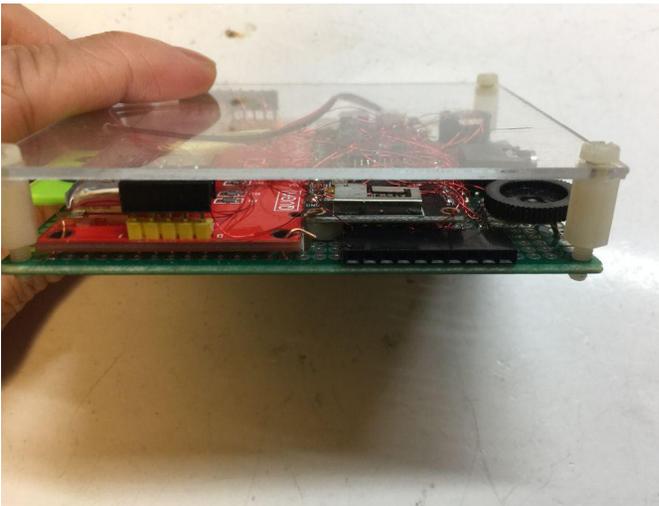
(10 pin extension header)

1 GND
 2 IO18 (VSPI-SCK),
 3 IO12
 4 IO15 (ADC2_Ch3),
 5 IO4 (ADC2_Ch0)
 6 3V3
 7 IO19 (VSPI-MISO),
 8 IO23 (VSPI-MOSI),
 9 N.C.
 10. 5V (only available when USB is plugged in).

Power

GND-100K-
 IO36, VP, -100K-VBat, 3.7V,
 GND-0.1uF-IO36



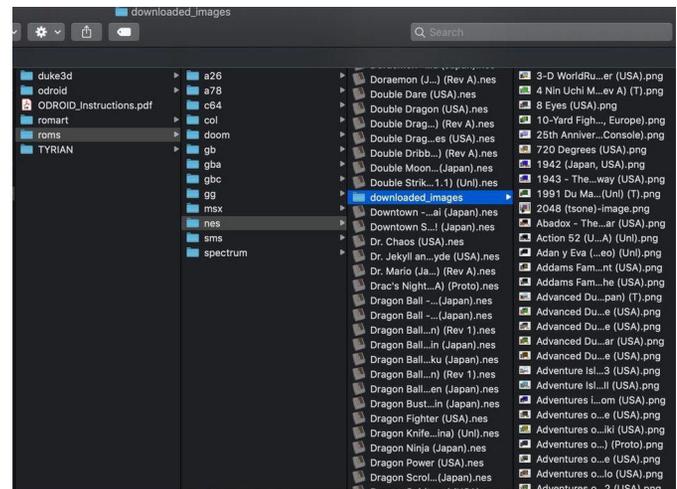


- Soldar la resistencia variable 10K, la toma de auriculares y el altavoz tal como se muestra en el diseño de pin
- Soldar dos resistencias de 100K y una 0.1 uF y conectarla a GPIO36 para medir el nivel de la batería
- Soldar el cabezal de extensión de 10 pin a los pines del ESP32 conforme al diseño de pin
- Soldar un cable a la batería y conectarlo a la entrada de la batería de la placa ESP32
- Cubrir la parte posterior de la PCB con un panel acrílico y asegurarlo con tornillos

Este proyecto utiliza un prototipo de PCB de doble cara de 7x9 cm para la estructura de la consola de juegos. Nos referiremos a esto como "la PCB". Los pasos de configuración incluyen:

- Hacer un agujero en forma de cuadrado en la PCB para la LCD de 2.2"
- Soldar los botones tal y como se muestra en el diseño a la parte frontal de la PCB
- Retirar la memoria RAM de 4Mb soldada existente de la placa TTGO T8 ESP32-Wrover
- Soldar una memoria RAM de 16MB (Winbond W25Q128FVSIQ))
- Montar la placa TTGO T8 ESP32-Wrover en la parte posterior de la PCB
- Montar la pantalla LCD TFT de 2.2" en la parte posterior de la PCB
- Suprimir los 9 pines de una toma ic para añadir los pines LCD de la TFT LCD
- Suprimir los 5 pines de una toma ic para añadir los pines de la tarjeta SD de TFT
- Usando cables laminados (aislados) de 0.2 o 0.3 mm, empezar con los trabajos de soldadura
- Soldar todas las conexiones desde la pantalla LCD TFT a la placa ESP32 siguiendo el diagrama del circuito y el diseño de pin
- Soldar todas las conexiones desde la ranura de la TARJETA SD a la placa ESP32 siguiendo el diagrama del circuito y el diseño de pin
- Soldar un extremo de los botones D-PAD a los 3.3V (es decir, arriba/abajo/izquierda/derecha), y el otro extremo al pin GPIO derecho de la ESP32 siguiendo el diseño de pin
- Soldar un extremo de los otros botones a GND, y el otro extremo al pin GPIO derecho de la ESP32 siguiendo el diseño de pin

Grabar la tarjeta SD con el firmware y las ROMs



Sigue estos pasos:

- Dirígete a https://wiki.odroid.com/odroid_go/make_sd_card
- Sigue las instrucciones para descargar los archivos de la estructura básica de la tarjeta SD (sin las ROM de juegos), luego formatea una tarjeta flash de 16G o 32G como FAT y copia las carpetas y archivos de la estructura básica de la tarjeta SD descargados en el paso 2 al directorio raíz de la tarjeta SD
- Opcionalmente, también puedes descargar y agregar otro software de terceros a la tarjeta SD siguiendo las instrucciones de <https://goo.gl/vr4YdQ>
- Para las ROM de juegos, puedes buscar en los diferentes sitios de alojamiento de roms del juego buscando por el título del juego, el nombre de la consola del juego + rom. Por ejemplo, "pinball nes rom". Una vez descargado, asegúrate de descomprimirlo para disponer del archivo original, por ejemplo, pinball.nes en lugar de pinball.zip
- Copia las Roms del juego en la carpeta roms de la tarjeta SD bajo la carpeta correcta de la consola del juego, por ejemplo, copia pinball.nes en e: oms es

- Al igual que en RetroPie o Recalbox, los archivos png de los íconos de las ROM se almacenan en la carpeta downloaded_images dentro de cada carpeta de consola de juegos
- Si está copiando roms de juegos de RetroPie o Recalbox en ODROID-GO, ten en cuenta que algunos de los nombres de las carpetas son diferentes, por ejemplo: en lugar de "atari2600", se utiliza "a26" y no permite los archivos .zip
- Una vez que todas las ROM de juegos estén copiadas en la tarjeta SD, retira ésta de forma segura de tu ordenador (depende del sistema operativo que uses, windows, linux, mac)
- Inserte la tarjeta SD en el ODROID-GO

Grabar la imagen de arranque en la Esp32



Los pasos a seguir para grabar la imagen de arranque son:

- Asegúrate de que la tarjeta SD con todo el firmware y las roms de juegos que grabaste en el paso anterior esté insertada en el ODROID-GO
- En tu estación de trabajo, dirígete a https://wiki.odroid.com/odroid_go/firmware_update
- Sigue las instrucciones para descargar la herramienta de flash ESP32 y la imagen de arranque del Odroid-go. Las herramientas e instrucciones dependerán de si estás usando un ordenador con Windows, Linux u OSX
- No es necesario borrar la memoria flash por completo al principio. Las correspondientes particiones de flash se borrarán cuando se cargue un archivo de firmware desde la tarjeta SD a la ESP32
- Carga la imagen de arranque del ODROID-GO en la ram SP32 SPI de acuerdo con las instrucciones. Consulta los comandos de ejemplo y los resultados de pantalla que aparecen a continuación. Estos comandos

y resultados de pantalla corresponden a la aplicación terminal de un Mac OSX:

```
# ...
# check the ESP32 SPI serial flash memory size
# ...
$ python3 -m esptool --port
/dev/tty.SLAB_USBtoUART flash_id
esptool.py v2.5.0
Serial port /dev/tty.SLAB_USBtoUART
Connecting....._
Detecting chip type... ESP32
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef
calibration in efuse
MAC: 3c:71:bf:03:50:40
Uploading stub...
Running stub...
Stub running...
Manufacturer: ef
Device: 4018
Detected flash size: 16MB
Hard resetting via RTS pin...
# ...
# burn the ODROID-GO-firmware into ESP 32 EEPROM
# ...
$ python3 -m esptool --chip esp32 --port
/dev/cu.SLAB_USBtoUART --baud 921600
write_flash --flash_mode dio --flash_freq 40m --
flash_size detect 0
ODROID-GO-firmware-20181001.img
esptool.py v2.5.0
Serial port /dev/cu.SLAB_USBtoUART
Connecting....._
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef
calibration in efuse
MAC: 3c:71:bf:03:50:40
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 16MB
Compressed 301920 bytes to 146523...
Wrote 301920 bytes (146523 compressed) at
0x00000000 in 2.3 seconds (effective 1039.5
kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
```

- Una vez completado el proceso, la ESP32 se reiniciará y ejecutará el menú de arranque del ODROID-GO

Ejecuta juegos con Go-Play y los emuladores



Main Menu

Left / Right – select game system

A – show games for selected system

B – return to previously loaded game

Volume + Right – toggle speaker / headphones

Game System Menu

Up / Down – move one step through game list

Left / Right – move one screen through game list

Select / Start – jump to previous/next letter of the alphabet in the game list

A – select game to run **B** – back to main menu

In-game

Menu – save game and return to main menu

Other buttons are described per system overleaf





Puedes seguir estos pasos para jugar a los diferentes juegos:

- Después de arrancar, verás una lista de firmwares. Elige el Go-Play y lánzalo pulsando el botón A
- El firmware de Go-Play se guardará en la memoria flash ESP32. Hay un total de cuatro particiones (0,1,2,3) a grabar. Para cada archivo, el espacio correspondiente (partición) de la memoria flash serie de 16M SPI se borrará en primer lugar y luego se escribirá. Llevará alrededor de un minuto grabar todas las particiones. Por suerte, no tendrás que hacer esto cada vez que juegues a un juego diferente. Mientras estés dentro del firmware de Go-Play, puede cargar y jugar a un juego en pocos segundos
- Se inicia el menú principal de Go-Play verás una lista de emuladores de juegos para elegir. Usa el botón izquierdo y derecho para seleccionar el emulador que desees. por ejemplo, nep. En este menú principal, puedes presionar el botón de volumen para subir o bajar el volumen del audio. O mantener presionado el botón de inicio y presionar el botón arriba/abajo para ajustar el brillo de la pantalla LCD TFT, o presionar B para volver al juego anterior al que habías jugado
- Una vez que selecciones un emulador, verás todas las salas de juegos ordenadas por orden alfabético. Usa el botón arriba/abajo para navegar a través de los juegos, izquierda/derecha para avanzar/retroceder de página.

Presiona Select y Start para saltar a la letra anterior o siguiente de la lista de juegos que está ordenada por orden alfabético. Esto es muy útil ya que, al tener miles de juegos terminas cansándote de presionar tantas veces el botón abajo

- Para iniciar un juego, presiona el botón "A" dos veces. O puedes presionar el botón "B" para volver al menú del emulador del juego y seleccionar un emulador de juegos diferente. Para salir de un juego y volver al menú principal, presiona el botón de menú. Mientras estés dentro del juego, puedes mantener presionado el botón de inicio al mismo tiempo que haces clic en el botón derecho para ajustar el escalado de la pantalla hacia arriba o hacia abajo. Verás cómo los gráficos se amplían o reducen un poco, en lugar de tener que mapear píxel a píxel el juego original. Para poner el ESP32 en modo de suspensión, mantén presionado el botón de menú durante 2 segundos
- Para activar la ESP32, vuelve a pulsar el botón de menú. Para recuperarte de un juego bloqueado, apaga la alimentación (utilizando el interruptor deslizante). Luego mantén presionado el botón de menú antes de encender. Para volver al menú de firmware o para recuperarse de un firmware dañado, apágalo. Luego mantén presionado el botón B antes de encender. Puedes grabar una nueva imagen de arranque en la ESP 32 en cualquier momento sin presionar ningún botón especial. Sin embargo, si no logras hacerlo funcionar, apágalo de nuevo. Después, mantén presionado el botón de volumen antes de encenderlo. La pantalla se iluminará completamente. Luego podrás conectar la unidad esp 32 al puerto usb de tu ordenador para que se actualice con una nueva imagen de arranque

Las instrucciones sobre cómo navegar por el firmware go-play están disponibles en https://wiki.odroid.com/odroid_go/emulator/usage_go_play. Para comentarios, preguntas y sugerencias, visita el artículo original en <https://goo.gl/yAbmbt>.

Kernel 5.0: La Próxima Versión LTS del Kernel de Linux

February 15, 2019 By Marian Mihailescu Linux



Kernel 5.0 Marian Mihailescu

¡El Kernel Linux 5.0, la próxima versión LTS del Kernel de Linux, ya está disponible para los ODROIDs! Incluye soporte para ARM BIG.little, y ha sido diseñado específicamente para trabajar con dispositivos ARM. Puedes encontrar más información en <https://www.xda-developers.com/linux-kernel-5-0-rc1-arm-big-little-eas-support-f2fs-fixes/>.

```
$ uname -a
Linux odroid 5.0.0-rc2 #3 SMP PREEMPT Sat Jan 19
00:08:56 ACDT 2019 armv7l armv7l armv7l GNU/Linux
```

Aquí tienes un ejemplo del registro de arranque:

```
[ 0.000000] Booting Linux on physical CPU 0x100
[ 0.000000] Linux version 5.0.0-rc2
(odroid@odroid) (gcc version 7.3.0 (Ubuntu/Linaro
7.3.0-16ubuntu3)) #3 SMP PREEMPT Sat Jan 19
00:08:56 ACDT 2019
[ 0.000000] CPU: ARMv7 Processor [410fc073]
revision 3 (ARMv7), cr=10c5387d
[ 0.000000] CPU: div instructions available:
```

```
patching division code
[ 0.000000] CPU: PIPT / VIPT nonaliasing data
cache, VIPT aliasing instruction cache
[ 0.000000] OF: fdt: Machine model: Hardkernel
Odroid XU4
[ 0.000000] Memory policy: Data cache
writealloc
[ 0.000000] cma: Reserved 128 MiB at 0xb6800000
[ 0.000000] On node 0 totalpages: 518656
[ 0.000000] Normal zone: 1728 pages used for
memmap
[ 0.000000] Normal zone: 0 pages reserved
[ 0.000000] Normal zone: 196608 pages, LIFO
batch:63
[ 0.000000] HighMem zone: 322048 pages, LIFO
batch:63
[ 0.000000] Running under secure firmware.
[ 0.000000] random: get_random_bytes called
from start_kernel+0xa0/0x498 with crng_init=0
[ 0.000000] percpu: Embedded 17 pages/cpu
@((ptrval) s38348 r8192 d23092 u69632
[ 0.000000] pcpu-alloc: s38348 r8192 d23092
u69632 alloc=17*4096
[ 0.000000] pcpu-alloc: [0] 0 [0] 1 [0] 2 [0] 3
```

```
[0] 4 [0] 5 [0] 6 [0] 7
[ 0.000000] Built 1 zonelists, mobility
grouping on. Total pages: 516928
[ 0.000000] Kernel command line: console=tty1
console=ttySAC2,115200n8 root=UUID=12345678-1234-
5678-9abc-123456789abc rootwait ro fsck.repair=yes
net.ifnames=0 HPD=true vout=hdmi
smc95xx.macaddr=00:1e:06:61:7a:39 false
s5p_mfc.mem=16M
[ 0.000000] hdmi: using HDMI mode
[ 0.000000] Dentry cache hash table entries:
131072 (order: 7, 524288 bytes)
[ 0.000000] Inode-cache hash table entries:
65536 (order: 6, 262144 bytes)
[ 0.000000] Memory: 1901132K/2074624K available
(9216K kernel code, 731K rwdata, 2424K rodata,
1024K init, 318K bss, 42420K reserved, 131072K
cma-reserved, 1157120K highmem)
[ 0.000000] Virtual kernel memory layout:
                vector : 0xffff0000 -
0xffff1000   ( 4 kB)
                fixmap : 0xffc00000 -
0xff000000   (3072 kB)
                vmalloc : 0xf0800000 -
0xff800000   ( 240 MB)
                lowmem  : 0xc0000000 -
0xf0000000   ( 768 MB)
                pkmap   : 0xbfe00000 -
0xc0000000   ( 2 MB)
                modules : 0xbf000000 -
0xbfe00000   ( 14 MB)
                .text   : 0x(ptrval) -
0x(ptrval)   (10208 kB)
                .init   : 0x(ptrval) -
0x(ptrval)   (1024 kB)
                .data   : 0x(ptrval) -
0x(ptrval)   ( 732 kB)
                .bss   : 0x(ptrval) -
0x(ptrval)   ( 319 kB)
[ 0.000000] SLUB: HWalign=64, Order=0-3,
MinObjects=0, CPUs=8, Nodes=1
[ 0.000000] hperf_hmp: fast CPUs mask: 000000F0
[ 0.000000] hperf_hmp: slow CPUs mask: 0000000F
[ 0.000000] rcu: Preemptible hierarchical RCU
implementation.
[ 0.000000] Tasks RCU enabled.
[ 0.000000] rcu: RCU calculated value of
scheduler-enlistment delay is 25 jiffies.
[ 0.000000] NR_IRQS: 16, nr_irqs: 16,
preallocated irq: 16
[ 0.000000] GIC: Using split EOI/Deactivate
mode
[ 0.000000] Switching to timer-based delay
```

```
loop, resolution 41ns
[ 0.000000] clocksource: mct-frc: mask:
0xffffffff max_cycles: 0xffffffff, max_idle_ns:
79635851949 ns
[ 0.000007] sched_clock: 32 bits at 24MHz,
resolution 41ns, wraps every 89478484971ns
[ 0.000029] genirq: irq_chip COMBINER did not
update eff. affinity mask of irq 49
[ 0.001019] Console: colour dummy device 80x30
[ 0.001665] printk: console [tty1] enabled
[ 0.001718] Calibrating delay loop (skipped),
value calculated using timer frequency.. 48.00
BogoMIPS (lpj=96000)
[ 0.001758] pid_max: default: 32768 minimum:
301
[ 0.001975] Mount-cache hash table entries:
2048 (order: 1, 8192 bytes)
[ 0.002013] Mountpoint-cache hash table
entries: 2048 (order: 1, 8192 bytes)
[ 0.002965] CPU: Testing write buffer
coherency: ok
[ 0.003471] CPU0: thread -1, cpu 0, socket 1,
mpidr 80000100
[ 0.024054] Setting up static identity map for
0x40100000 - 0x40100060
[ 0.024382] ARM CCI driver probed
[ 0.024623] Exynos MCPM support installed
[ 0.031995] rcu: Hierarchical SRCU
implementation.
[ 0.042046] soc soc0: Exynos: CPU[EXYNOS5800]
PRO_ID[0xe5422001] REV[0x1] Detected
[ 0.047996] smp: Bringing up secondary CPUs ...
[ 0.080353] CPU1: thread -1, cpu 1, socket 1,
mpidr 80000101
[ 0.104314] CPU2: thread -1, cpu 2, socket 1,
mpidr 80000102
[ 0.136309] CPU3: thread -1, cpu 3, socket 1,
mpidr 80000103
[ 0.168309] CPU4: thread -1, cpu 0, socket 0,
mpidr 80000000
[ 0.168318] CPU4: Spectre v2: using ICIALLU
workaround
[ 0.192297] CPU5: thread -1, cpu 1, socket 0,
mpidr 80000001
[ 0.192304] CPU5: Spectre v2: using ICIALLU
workaround
[ 0.204515] CPU6: thread -1, cpu 2, socket 0,
mpidr 80000002
[ 0.204523] CPU6: Spectre v2: using ICIALLU
workaround
[ 0.216521] CPU7: thread -1, cpu 3, socket 0,
mpidr 80000003
[ 0.216528] CPU7: Spectre v2: using ICIALLU
```

```
workaround
[ 0.216711] smp: Brought up 1 node, 8 CPUs
[ 0.216757] SMP: Total of 8 processors
activated (384.00 BogoMIPS).
[ 0.216781] CPU: WARNING: CPU(s) started in
wrong/inconsistent modes (primary CPU mode 0x1a)
[ 0.216809] CPU: This may indicate a broken
bootloader or firmware.
[ 0.218665] devtmpfs: initialized
[ 0.234381] VFP support v0.3: implementor 41
architecture 4 part 30 variant f rev 0
[ 0.234532] hperf_hmp: registered cpufreq
transition notifier
[ 0.234815] clocksource: jiffies: mask:
0xffffffff max_cycles: 0xffffffff, max_idle_ns:
7645041785100000 ns
[ 0.234851] futex hash table entries: 2048
(order: 5, 131072 bytes)
[ 0.237448] pinctrl core: initialized pinctrl
subsystem
[ 0.239514] NET: Registered protocol family 16
[ 0.241329] DMA: preallocated 256 KiB pool for
atomic coherent allocations
[ 0.242593] audit: initializing netlink subsys
(disabled)
[ 0.242785] audit: type=2000 audit(0.240:1):
state=initialized audit_enabled=0 res=1
[ 0.243262] cpuidle: using governor menu
[ 0.243767] hw-breakpoint: found 5 (+1
reserved) breakpoint and 4 watchpoint registers.
[ 0.243793] hw-breakpoint: maximum watchpoint
size is 8 bytes.
[ 0.287465] EXYNOS5420 PMU initialized
[ 0.383251] usbcore: registered new interface
driver usbfs
[ 0.383323] usbcore: registered new interface
driver hub
[ 0.383490] usbcore: registered new device
driver usb
[ 0.384087] s3c-i2c 12c80000.i2c: slave address
0x00
[ 0.384116] s3c-i2c 12c80000.i2c: bus frequency
set to 65 KHz
[ 0.384402] s3c-i2c 12c80000.i2c: i2c-2: S3C
I2C adapter
[ 0.384831] media: Linux media interface: v0.10
[ 0.384880] videodev: Linux video capture
interface: v2.00
[ 0.385008] pps_core: LinuxPPS API ver. 1
registered
[ 0.385028] pps_core: Software ver. 5.3.6 -
Copyright 2005-2007 Rodolfo Giometti
<giometti@linux.it>
```

```
[ 0.385285] s3c2410-wdt 101d0000.watchdog:
watchdog inactive, reset disabled, irq disabled
[ 0.386077] Advanced Linux Sound Architecture
Driver Initialized.
[ 0.387307] clocksource: Switched to
clocksource mct-frc
[ 0.423321] random: fast init done
[ 1.035504] VFS: Disk quotas dquot_6.6.0
[ 1.035594] VFS: Dquot-cache hash table
entries: 1024 (order 0, 4096 bytes)
[ 1.048841] NET: Registered protocol family 2
[ 1.049470] tcp_listen_portaddr_hash hash table
entries: 512 (order: 0, 6144 bytes)
[ 1.049521] TCP established hash table entries:
8192 (order: 3, 32768 bytes)
[ 1.049611] TCP bind hash table entries: 8192
(order: 4, 65536 bytes)
[ 1.049778] TCP: Hash tables configured
(established 8192 bind 8192)
[ 1.049896] UDP hash table entries: 512 (order:
2, 16384 bytes)
[ 1.049949] UDP-Lite hash table entries: 512
(order: 2, 16384 bytes)
[ 1.050264] NET: Registered protocol family 1
[ 1.050826] RPC: Registered named UNIX socket
transport module.
[ 1.050850] RPC: Registered udp transport
module.
[ 1.050869] RPC: Registered tcp transport
module.
[ 1.050888] RPC: Registered tcp NFSv4.1
backchannel transport module.
[ 1.051098] Trying to unpack rootfs image as
initramfs...
[ 1.520593] Freeing initrd memory: 8212K
[ 1.521801] hw perfevents: enabled with
armv7_cortex_a7 PMU driver, 5 counters available
[ 1.522633] hw perfevents: enabled with
armv7_cortex_a15 PMU driver, 7 counters available
[ 1.527107] Initialise system trusted keyrings
[ 1.527384] workingset: timestamp_bits=14
max_order=19 bucket_order=5
[ 1.536232] squashfs: version 4.0 (2009/01/31)
Phillip Lougher
[ 1.537006] NFS: Registering the id_resolver
key type
[ 1.537044] Key type id_resolver registered
[ 1.537063] Key type id_legacy registered
[ 1.537093] nfs4filelayout_init: NFSv4 File
Layout Driver Registering...
[ 1.537145] romfs: ROMFS MTD (C) 2007 Red Hat,
Inc.
[ 1.641405] Key type asymmetric registered
```

```
[ 1.641431] Asymmetric key parser 'x509'
registered
[ 1.641500] bounce: pool size: 64 pages
[ 1.641560] Block layer SCSI generic (bsg)
driver version 0.4 loaded (major 245)
[ 1.641793] io scheduler mq-deadline registered
[ 1.641816] io scheduler kyber registered
[ 1.642077] io scheduler bfq registered
[ 1.644526] samsung-usb2-phy 12130000.phy:
12130000.phy supply vbus not found, using dummy
regulator
[ 1.644633] samsung-usb2-phy 12130000.phy:
Linked as a consumer to regulator.0
[ 1.645388] exynos5_usb3drd_phy 12100000.phy:
12100000.phy supply vbus not found, using dummy
regulator
[ 1.645482] exynos5_usb3drd_phy 12100000.phy:
Linked as a consumer to regulator.0
[ 1.645516] exynos5_usb3drd_phy 12100000.phy:
12100000.phy supply vbus-boost not found, using
dummy regulator
[ 1.645887] exynos5_usb3drd_phy 12500000.phy:
12500000.phy supply vbus not found, using dummy
regulator
[ 1.645991] exynos5_usb3drd_phy 12500000.phy:
Linked as a consumer to regulator.0
[ 1.646024] exynos5_usb3drd_phy 12500000.phy:
12500000.phy supply vbus-boost not found, using
dummy regulator
[ 1.653795] dma-pl330 121a0000.pdma: Loaded
driver for PL330 DMAC-241330
[ 1.653825] dma-pl330 121a0000.pdma: DBUFF-
32x4bytes Num_Chans-8 Num_Per-32 Num_Events-32
[ 1.656518] dma-pl330 121b0000.pdma: Loaded
driver for PL330 DMAC-241330
[ 1.656546] dma-pl330 121b0000.pdma: DBUFF-
32x4bytes Num_Chans-8 Num_Per-32 Num_Events-32
[ 1.657395] dma-pl330 10800000.mdma: Loaded
driver for PL330 DMAC-241330
[ 1.657422] dma-pl330 10800000.mdma: DBUFF-
64x8bytes Num_Chans-8 Num_Per-1 Num_Events-32
[ 1.719606] Serial: 8250/16550 driver, 4 ports,
IRQ sharing disabled
[ 1.721748] 12c00000.serial: ttySAC0 at MMIO
0x12c00000 (irq = 58, base_baud = 0) is a
S3C6400/10
[ 1.722153] 12c10000.serial: ttySAC1 at MMIO
0x12c10000 (irq = 59, base_baud = 0) is a
S3C6400/10
[ 1.722543] 12c20000.serial: ttySAC2 at MMIO
0x12c20000 (irq = 60, base_baud = 0) is a
S3C6400/10
[ 2.640269] printk: console [ttySAC2] enabled
```

```
[ 2.644919] 12c30000.serial: ttySAC3 at MMIO
0x12c30000 (irq = 61, base_baud = 0) is a
S3C6400/10
[ 2.654959] exynos-trng 10830600.rng: Exynos
True Random Number Generator.
[ 2.661237] iommu: Adding device 14450000.mixer
to group 0
[ 2.666316] exynos-mixer 14450000.mixer: Linked
as a consumer to 14650000.sysmmu
[ 2.674761] exynos-hdmi 14530000.hdmi: Failed
to get supply 'vdd': -517
[ 2.680599] iommu: Adding device 10850000.g2d
to group 1
[ 2.685238] exynos-drm-g2d 10850000.g2d: Linked
as a consumer to 10a60000.sysmmu
[ 2.692602] exynos-drm-g2d 10850000.g2d: Linked
as a consumer to 10a70000.sysmmu
[ 2.705591] mali 11800000.mali: Continuing
without Mali regulator control
[ 2.712268] mali 11800000.mali: GPU identified
as 0x0620 r0p1 status 0
[ 2.717941] mali 11800000.mali: Protected mode
not available
[ 2.723557] devfreq devfreq0: Couldn't update
frequency transition information.
[ 2.731075] mali 11800000.mali: Probed as mali0
[ 2.745078] brd: module loaded
[ 2.747198] libphy: Fixed MDIO Bus: probed
[ 2.751105] usbcore: registered new interface
driver r8152
[ 2.756278] usbcore: registered new interface
driver cdc_ether
[ 2.762073] usbcore: registered new interface
driver cdc_subset
[ 2.769920] ehci_hcd: USB 2.0 'Enhanced' Host
Controller (EHCI) Driver
[ 2.775016] ehci-exynos: EHCI EXYNOS driver
[ 2.779530] exynos-ehci 12110000.usb: EHCI Host
Controller
[ 2.784643] exynos-ehci 12110000.usb: new USB
bus registered, assigned bus number 1
[ 2.792552] exynos-ehci 12110000.usb: irq 85,
io mem 0x12110000
[ 2.811362] exynos-ehci 12110000.usb: USB 2.0
started, EHCI 1.00
[ 2.816228] usb usb1: New USB device found,
idVendor=1d6b, idProduct=0002, bcdDevice= 5.00
[ 2.824173] usb usb1: New USB device strings:
Mfr=3, Product=2, SerialNumber=1
[ 2.831359] usb usb1: Product: EHCI Host
Controller
[ 2.836204] usb usb1: Manufacturer: Linux
5.0.0-rc2 ehci_hcd
```

```
[ 2.841838] usb usb1: SerialNumber:
12110000.usb
[ 2.846993] hub 1-0:1.0: USB hub found
[ 2.850201] hub 1-0:1.0: 3 ports detected
[ 2.855038] ohci_hcd: USB 1.1 'Open' Host
Controller (OHCI) Driver
[ 2.860310] ohci-exynos: OHCI EXYNOS driver
[ 2.864628] exynos-ohci 12120000.usb: USB Host
Controller
[ 2.869844] exynos-ohci 12120000.usb: new USB
bus registered, assigned bus number 2
[ 2.877589] exynos-ohci 12120000.usb: irq 85,
io mem 0x12120000
[ 2.947600] usb usb2: New USB device found,
idVendor=1d6b, idProduct=0001, bcdDevice= 5.00
[ 2.954430] usb usb2: New USB device strings:
Mfr=3, Product=2, SerialNumber=1
[ 2.961616] usb usb2: Product: USB Host
Controller
[ 2.966376] usb usb2: Manufacturer: Linux
5.0.0-rc2 ohci_hcd
[ 2.972009] usb usb2: SerialNumber:
12120000.usb
[ 2.977134] hub 2-0:1.0: USB hub found
[ 2.980371] hub 2-0:1.0: 3 ports detected
[ 2.986369] mousedev: PS/2 mouse device common
for all mice
[ 2.991775] i2c /dev entries driver
[ 3.010215] vdd_ldo9: Bringing 3300000uV into
3000000-3000000uV
[ 3.023733] vddq_mmc2: Bringing 3300000uV into
2800000-2800000uV
[ 3.040091] vdd_sd: Bringing 3300000uV into
2800000-2800000uV
[ 3.187373] usb 1-1: new high-speed USB device
number 2 using exynos-ehci
[ 3.349477] usb 1-1: New USB device found,
idVendor=0bda, idProduct=8176, bcdDevice= 2.00
[ 3.356230] usb 1-1: New USB device strings:
Mfr=1, Product=2, SerialNumber=3
[ 3.363322] usb 1-1: Product: 802.11n WLAN
Adapter
[ 3.368088] usb 1-1: Manufacturer: Realtek
[ 3.372159] usb 1-1: SerialNumber: 00e04c000001
[ 4.109715] s5m-rtc s2mps14-rtc: registered as
rtc0
[ 4.113702] s2mps11-clk s2mps11-clk: DMA mask
not set
[ 4.120973] iommu: Adding device 11f50000.jpeg
to group 2
[ 4.124956] s5p-jpeg 11f50000.jpeg: Linked as a
consumer to 11f10000.sysmmu
[ 4.132244] s5p-jpeg 11f50000.jpeg: encoder
```

```
device registered as /dev/video30
[ 4.139142] s5p-jpeg 11f50000.jpeg: decoder
device registered as /dev/video31
[ 4.146087] s5p-jpeg 11f50000.jpeg: Samsung S5P
JPEG codec
[ 4.151766] iommu: Adding device 11f60000.jpeg
to group 3
[ 4.156925] s5p-jpeg 11f60000.jpeg: Linked as a
consumer to 11f20000.sysmmu
[ 4.164185] s5p-jpeg 11f60000.jpeg: encoder
device registered as /dev/video32
[ 4.171117] s5p-jpeg 11f60000.jpeg: decoder
device registered as /dev/video33
[ 4.178062] s5p-jpeg 11f60000.jpeg: Samsung S5P
JPEG codec
[ 4.184222] iommu: Adding device 11000000.codec
to group 4
[ 4.189433] s5p-mfc 11000000.codec: Linked as a
consumer to 11200000.sysmmu
[ 4.195927] s5p-mfc 11000000.codec: Linked as a
consumer to 11210000.sysmmu
[ 4.220127] s5p-mfc 11000000.codec:
preallocated 16 MiB buffer for the firmware and
context buffers
[ 4.227837] s5p-mfc 11000000.codec: Direct
firmware load for s5p-mfc-v8.fw failed with error
-2
[ 4.236446] s5p_mfc_load_firmware:73: Firmware
is not present in the /lib/firmware directory nor
compiled in kernel
[ 4.246971] s5p-mfc 11000000.codec: decoder
registered as /dev/video10
[ 4.253450] s5p-mfc 11000000.codec: encoder
registered as /dev/video11
[ 4.261770] iommu: Adding device
13e00000.video-scaler to group 5
[ 4.266898] exynos-gsc 13e00000.video-scaler:
Linked as a consumer to 13e80000.sysmmu
[ 4.274956] iommu: Adding device
13e10000.video-scaler to group 6
[ 4.280789] exynos-gsc 13e10000.video-scaler:
Linked as a consumer to 13e90000.sysmmu
[ 4.292670] exynos-tm10060000.tmu: Linked as
a consumer to regulator.7
[ 4.298174] thermal thermal_zone0: failed to
read out thermal zone (-22)
[ 4.305357] exynos-tm10064000.tmu: Linked as
a consumer to regulator.7
[ 4.311504] thermal thermal_zone1: failed to
read out thermal zone (-22)
[ 4.318690] exynos-tm10068000.tmu: Linked as
a consumer to regulator.7
[ 4.324838] thermal thermal_zone2: failed to
```

```
read out thermal zone (-22)
[ 4.332073] exynos-tmu 1006c000.tmu: Linked as
a consumer to regulator.7
[ 4.338176] thermal thermal_zone3: failed to
read out thermal zone (-22)
[ 4.345371] exynos-tmu 100a0000.tmu: Linked as
a consumer to regulator.7
[ 4.351542] thermal thermal_zone4: failed to
read out thermal zone (-22)
[ 4.358617] device-mapper: uevent: version
1.0.3
[ 4.362902] device-mapper: ioctl: 4.39.0-ioctl
(2018-04-03) initialised: dm-devel@redhat.com
[ 4.372008] cpu cpu0: Linked as a consumer to
regulator.44
[ 4.376524] cpu cpu0: Dropping the link to
regulator.44
[ 4.382332] cpu cpu0: Linked as a consumer to
regulator.44
[ 4.393774] cpu cpu4: Linked as a consumer to
regulator.40
[ 4.405406] sdhci: Secure Digital Host
Controller Interface driver
[ 4.410131] sdhci: Copyright(c) Pierre Ossman
[ 4.414633] Synopsys Designware Multimedia Card
Interface Driver
[ 4.420997] dwmmc_exynos 12200000.mmc: IDMAC
supports 32-bit address mode.
[ 4.427409] dwmmc_exynos 12200000.mmc: Using
internal DMA controller.
[ 4.433702] dwmmc_exynos 12200000.mmc: Version
ID is 250a
[ 4.439095] dwmmc_exynos 12200000.mmc: DW MMC
controller at irq 87,64 bit host data width,64
deep fifo
[ 4.448406] dwmmc_exynos 12200000.mmc: Linked
as a consumer to regulator.18
[ 4.455346] dwmmc_exynos 12200000.mmc: Linked
as a consumer to regulator.3
[ 4.462198] dwmmc_exynos 12200000.mmc:
allocated mmc-pwrseq
[ 4.484970] mmc_host mmc0: Bus speed (slot 0) =
50000000Hz (slot req 400000Hz, actual 396825Hz div
= 63)
[ 4.506103] dwmmc_exynos 12220000.mmc: IDMAC
supports 32-bit address mode.
[ 4.511568] dwmmc_exynos 12220000.mmc: Using
internal DMA controller.
[ 4.517956] dwmmc_exynos 12220000.mmc: Version
ID is 250a
[ 4.523340] dwmmc_exynos 12220000.mmc: DW MMC
controller at irq 88,64 bit host data width,64
deep fifo
```

```
[ 4.532656] dwmmc_exynos 12220000.mmc: Linked
as a consumer to regulator.19
[ 4.539582] dwmmc_exynos 12220000.mmc: Linked
as a consumer to regulator.13
[ 4.561916] mmc_host mmc1: Bus speed (slot 0) =
50000000Hz (slot req 400000Hz, actual 396825Hz div
= 63)
[ 4.587457] mmc_host mmc0: Bus speed (slot 0) =
200000000Hz (slot req 200000000Hz, actual
200000000Hz div = 0)
[ 4.596855] mmc0: new HS200 MMC card at address
0001
[ 4.601657] mmcblk0: mmc0:0001 016G92 14.7 GiB
[ 4.606029] mmcblk0boot0: mmc0:0001 016G92
partition 1 4.00 MiB
[ 4.610824] s5p-secss 10830000.sss: s5p-sss
driver registered
[ 4.612116] mmcblk0boot1: mmc0:0001 016G92
partition 2 4.00 MiB
[ 4.617831] hidraw: raw HID events driver (C)
Jiri Kosina
[ 4.623194] mmcblk0rmpb: mmc0:0001 016G92
partition 3 512 KiB, chardev (244:0)
[ 4.631959] exynos-bus soc:bus_wcore: Linked as
a consumer to regulator.41
[ 4.642938] exynos-bus soc:bus_wcore: Dropping
the link to regulator.41
[ 4.647563] mmcblk0: p1 p2
[ 4.658168] exynos-nocp: new NoC Probe device
registered: 10ca1000.nocp
[ 4.663478] exynos-nocp: new NoC Probe device
registered: 10ca1400.nocp
[ 4.670023] exynos-nocp: new NoC Probe device
registered: 10ca1800.nocp
[ 4.676618] exynos-nocp: new NoC Probe device
registered: 10ca1c00.nocp
[ 4.683937] exynos-adc 12d10000.adc: Linked as
a consumer to regulator.4
[ 4.695206] NET: Registered protocol family 17
[ 4.698213] NET: Registered protocol family 15
[ 4.702686] Key type dns_resolver registered
[ 4.707396] Registering SWP/SWPB emulation
handler
[ 4.711640] mmc_host mmc1: Bus speed (slot 0) =
50000000Hz (slot req 50000000Hz, actual 50000000Hz
div = 0)
[ 4.720430] registered taskstats version 1
[ 4.722872] mmc1: new ultra high speed DDR50
SDHC card at address aaaa
[ 4.725410] Loading compiled-in X.509
certificates
[ 4.732815] mmcblk1: mmc1:aaaa SL16G 14.8 GiB
[ 4.744316] mmcblk1: p1 p2
```

```
[ 4.747714] Key type encrypted registered
[ 4.777617] exynos-hdmi 14530000.hdmi: Linked
as a consumer to regulator.6
[ 4.783596] exynos-hdmi 14530000.hdmi: Linked
as a consumer to regulator.7
[ 4.790398] OF: graph: no port node found in
/soc/hdmi@14530000
[ 4.796566] [drm] Exynos DRM: using
14450000.mixer device for DMA mapping operations
[ 4.803543] exynos-drm exynos-drm: bound
14450000.mixer (ops 0xc0a62a84)
[ 4.810165] exynos-drm exynos-drm: bound
14530000.hdmi (ops 0xc0a63128)
[ 4.816826] exynos-drm-g2d 10850000.g2d: The
Exynos G2D (ver 4.1) successfully registered.
[ 4.824975] exynos-drm exynos-drm: bound
10850000.g2d (ops 0xc0a6403c)
[ 4.831469] [drm] Supports vblank timestamp
caching Rev 2 (21.10.2013).
[ 4.838051] [drm] No driver support for vblank
timestamp query.
[ 5.008869] Console: switching to colour frame
buffer device 240x67
[ 5.029973] exynos-drm exynos-drm: fb0: frame
buffer device
[ 5.036099] [drm] Initialized exynos 1.1.0
20180330 for exynos-drm on minor 0
[ 5.043402] exynos-dwc3 soc:usb3-0: Linked as a
consumer to regulator.9
[ 5.049827] exynos-dwc3 soc:usb3-0: Linked as a
consumer to regulator.11
[ 5.056259] dwc3 12000000.dwc3: Failed to get
clk 'ref': -2
[ 5.061838] xhci-hcd xhci-hcd.1.auto: xHCI Host
Controller
[ 5.066982] xhci-hcd xhci-hcd.1.auto: new USB
bus registered, assigned bus number 3
[ 5.074840] xhci-hcd xhci-hcd.1.auto: hcc
params 0x0220f04c hci version 0x100 quirks
0x0000000002010010
[ 5.084012] xhci-hcd xhci-hcd.1.auto: irq 156,
io mem 0x12000000
[ 5.090229] usb usb3: New USB device found,
idVendor=1d6b, idProduct=0002, bcdDevice= 5.00
[ 5.098192] usb usb3: New USB device strings:
Mfr=3, Product=2, SerialNumber=1
[ 5.105375] usb usb3: Product: xHCI Host
Controller
[ 5.110206] usb usb3: Manufacturer: Linux
5.0.0-rc2 xhci-hcd
[ 5.115845] usb usb3: SerialNumber: xhci-
hcd.1.auto
[ 5.121011] hub 3-0:1.0: USB hub found
```

```
[ 5.124435] hub 3-0:1.0: 1 port detected
[ 5.128526] xhci-hcd xhci-hcd.1.auto: xHCI Host
Controller
[ 5.133786] xhci-hcd xhci-hcd.1.auto: new USB
bus registered, assigned bus number 4
[ 5.141424] xhci-hcd xhci-hcd.1.auto: Host
supports USB 3.0 SuperSpeed
[ 5.148056] usb usb4: We don't know the
algorithms for LPM for this host, disabling LPM.
[ 5.156180] usb usb4: New USB device found,
idVendor=1d6b, idProduct=0003, bcdDevice= 5.00
[ 5.164304] usb usb4: New USB device strings:
Mfr=3, Product=2, SerialNumber=1
[ 5.171486] usb usb4: Product: xHCI Host
Controller
[ 5.176321] usb usb4: Manufacturer: Linux
5.0.0-rc2 xhci-hcd
[ 5.181958] usb usb4: SerialNumber: xhci-
hcd.1.auto
[ 5.187110] hub 4-0:1.0: USB hub found
[ 5.190550] hub 4-0:1.0: 1 port detected
[ 5.195388] exynos-dwc3 soc:usb3-1: Linked as a
consumer to regulator.9
[ 5.201521] exynos-dwc3 soc:usb3-1: Linked as a
consumer to regulator.11
[ 5.207974] dwc3 12400000.dwc3: Failed to get
clk 'ref': -2
[ 5.213545] xhci-hcd xhci-hcd.2.auto: xHCI Host
Controller
[ 5.218707] xhci-hcd xhci-hcd.2.auto: new USB
bus registered, assigned bus number 5
[ 5.228478] xhci-hcd xhci-hcd.2.auto: hcc
params 0x0220f04c hci version 0x100 quirks
0x0000000002010010
[ 5.238437] xhci-hcd xhci-hcd.2.auto: irq 157,
io mem 0x12400000
[ 5.245277] usb usb5: New USB device found,
idVendor=1d6b, idProduct=0002, bcdDevice= 5.00
[ 5.254152] usb usb5: New USB device strings:
Mfr=3, Product=2, SerialNumber=1
[ 5.261975] usb usb5: Product: xHCI Host
Controller
[ 5.267438] usb usb5: Manufacturer: Linux
5.0.0-rc2 xhci-hcd
[ 5.273714] usb usb5: SerialNumber: xhci-
hcd.2.auto
[ 5.279523] hub 5-0:1.0: USB hub found
[ 5.283897] hub 5-0:1.0: 1 port detected
[ 5.288609] xhci-hcd xhci-hcd.2.auto: xHCI Host
Controller
[ 5.294680] xhci-hcd xhci-hcd.2.auto: new USB
bus registered, assigned bus number 6
[ 5.302932] xhci-hcd xhci-hcd.2.auto: Host
```

```
supports USB 3.0 SuperSpeed
[ 5.310180] usb usb6: We don't know the
algorithms for LPM for this host, disabling LPM.
[ 5.318965] usb usb6: New USB device found,
idVendor=1d6b, idProduct=0003, bcdDevice= 5.00
[ 5.327840] usb usb6: New USB device strings:
Mfr=3, Product=2, SerialNumber=1
[ 5.335672] usb usb6: Product: xHCI Host
Controller
[ 5.341159] usb usb6: Manufacturer: Linux
5.0.0-rc2 xhci-hcd
[ 5.347424] usb usb6: SerialNumber: xhci-
hcd.2.auto
[ 5.353218] hub 6-0:1.0: USB hub found
[ 5.357605] hub 6-0:1.0: 1 port detected
[ 5.363951] dma-pl330 3880000.adma: Loaded
driver for PL330 DMAC-241330
[ 5.371181] dma-pl330 3880000.adma: DBUFF-
4x8bytes Num_Chans-6 Num_Peris-16 Num_Events-6
[ 5.379932] rtc rtc1: invalid alarm value:
1900-01-11T00:00:00
[ 5.386467] s3c-rtc 101e0000.rtc: registered as
rtc1
[ 5.393351] exynos-bus soc:bus_wcore: Linked as
a consumer to regulator.41
[ 5.400987] exynos-bus: new bus device
registered: soc:bus_wcore ( 84000 KHz ~ 400000
KHz)
[ 5.410331] exynos-bus: new bus device
registered: soc:bus_noc ( 67000 KHz ~ 100000 KHz)
[ 5.419432] exynos-bus: new bus device
registered: soc:bus_fsys_apb (100000 KHz ~ 200000
KHz)
[ 5.428802] exynos-bus: new bus device
registered: soc:bus_fsys (100000 KHz ~ 200000 KHz)
[ 5.437937] exynos-bus: new bus device
registered: soc:bus_fsys2 ( 75000 KHz ~ 150000
KHz)
[ 5.447283] exynos-bus: new bus device
registered: soc:bus_mfc ( 96000 KHz ~ 333000 KHz)
[ 5.456438] exynos-bus: new bus device
registered: soc:bus_gen ( 89000 KHz ~ 267000 KHz)
[ 5.465404] exynos-bus: new bus device
registered: soc:bus_peri ( 67000 KHz ~ 67000 KHz)
[ 5.474626] exynos-bus: new bus device
registered: soc:bus_g2d ( 84000 KHz ~ 333000 KHz)
[ 5.483738] exynos-bus: new bus device
registered: soc:bus_g2d_acp ( 67000 KHz ~ 267000
KHz)
[ 5.493119] exynos-bus: new bus device
registered: soc:bus_jpeg ( 75000 KHz ~ 300000 KHz)
[ 5.502244] exynos-bus: new bus device
registered: soc:bus_jpeg_apb ( 84000 KHz ~ 167000
```

```
KHz)
[ 5.511619] exynos-bus: new bus device
registered: soc:bus_displ1_fimd (120000 KHz ~
200000 KHz)
[ 5.521180] exynos-bus: new bus device
registered: soc:bus_displ1 (120000 KHz ~ 300000
KHz)
[ 5.529874] usb 3-1: new high-speed USB device
number 2 using xhci-hcd
[ 5.537291] exynos-bus: new bus device
registered: soc:bus_gscl_scaler (150000 KHz ~
300000 KHz)
[ 5.547105] exynos-bus: new bus device
registered: soc:bus_mscl ( 84000 KHz ~ 400000 KHz)
[ 5.557596] odroid-audio sound: i2s-hifi <->
samsung-i2s mapping ok
[ 5.564403] dma-pl330 3880000.adma: PM domain
MAU will not be powered off
[ 5.574100] s5m-rtc s2mps14-rtc: setting system
clock to 2019-01-18T13:52:24 UTC (1547819544)
[ 5.607567] ALSA device list:
[ 5.611035] #0: Odroid-XU4
[ 5.615391] Freeing unused kernel memory: 1024K
[ 5.635534] Run /init as init process
[ 5.699868] usb 3-1: New USB device found,
idVendor=05e3, idProduct=0610, bcdDevice=92.22
[ 5.708836] usb 3-1: New USB device strings:
Mfr=1, Product=2, SerialNumber=0
[ 5.716557] usb 3-1: Product: USB2.0 Hub
[ 5.721047] usb 3-1: Manufacturer: GenesysLogic
[ 5.788397] hub 3-1:1.0: USB hub found
[ 5.792998] hub 3-1:1.0: 2 ports detected
[ 6.179574] usb 4-1: new SuperSpeed Gen 1 USB
device number 2 using xhci-hcd
[ 6.210486] usb 4-1: New USB device found,
idVendor=05e3, idProduct=0616, bcdDevice=92.22
[ 6.219540] usb 4-1: New USB device strings:
Mfr=1, Product=2, SerialNumber=0
[ 6.228530] usb 4-1: Product: USB3.0 Hub
[ 6.233813] usb 4-1: Manufacturer: GenesysLogic
[ 6.268690] hub 4-1:1.0: USB hub found
[ 6.273386] hub 4-1:1.0: 2 ports detected
[ 6.315381] usb 3-1.1: new high-speed USB
device number 3 using xhci-hcd
[ 6.428551] usb 3-1.1: New USB device found,
idVendor=05e3, idProduct=0610, bcdDevice=92.12
[ 6.437411] usb 3-1.1: New USB device strings:
Mfr=1, Product=2, SerialNumber=0
[ 6.445230] usb 3-1.1: Product: USB2.0 Hub
[ 6.449791] usb 3-1.1: Manufacturer:
GenesysLogic
[ 6.459404] usb 6-1: new SuperSpeed Gen 1 USB
device number 2 using xhci-hcd
```

```
[ 6.484035] usb 6-1: New USB device found,
idVendor=0bda, idProduct=8153, bcdDevice=30.00
[ 6.492722] usb 6-1: New USB device strings:
Mfr=1, Product=2, SerialNumber=6
[ 6.500461] hub 3-1.1:1.0: USB hub found
[ 6.504914] usb 6-1: Product: USB 10/100/1000
LAN
[ 6.510138] hub 3-1.1:1.0: 4 ports detected
[ 6.514899] usb 6-1: Manufacturer: Realtek
[ 6.519502] usb 6-1: SerialNumber: 000001000000
[ 6.635505] usb 4-1.1: new SuperSpeed Gen 1 USB
device number 3 using xhci-hcd
[ 6.662686] usb 4-1.1: New USB device found,
idVendor=05e3, idProduct=0612, bcdDevice=92.12
[ 6.671521] usb 4-1.1: New USB device strings:
Mfr=1, Product=2, SerialNumber=0
[ 6.679319] usb 4-1.1: Product: USB3.0 Hub
[ 6.683894] usb 4-1.1: Manufacturer:
GenesysLogic
[ 6.716632] hub 4-1.1:1.0: USB hub found
[ 6.721256] usb 6-1: reset SuperSpeed Gen 1 USB
device number 2 using xhci-hcd
[ 6.729009] hub 4-1.1:1.0: 4 ports detected
[ 6.800151] r8152 6-1:1.0 eth0: v1.09.9
[ 6.879387] usb 3-1.1.2: new low-speed USB
device number 4 using xhci-hcd
[ 6.927449] EXT4-fs (mmcblk1p2): mounted
filesystem without journal. Opts: (null)
[ 6.995419] usb 3-1.1.2: New USB device found,
idVendor=0b38, idProduct=0010, bcdDevice= 1.02
[ 7.004483] usb 3-1.1.2: New USB device
strings: Mfr=0, Product=0, SerialNumber=0
[ 7.179338] usb 3-1.1.3: new low-speed USB
device number 5 using xhci-hcd
[ 7.291103] usb 3-1.1.3: New USB device found,
idVendor=093a, idProduct=2510, bcdDevice= 1.00
[ 7.300240] usb 3-1.1.3: New USB device
strings: Mfr=1, Product=2, SerialNumber=0
[ 7.308327] usb 3-1.1.3: Product: USB Optical
Mouse
[ 7.308333] usb 3-1.1.3: Manufacturer: PixArt
[ 7.404722] NET: Registered protocol family 10
[ 7.410590] Segment Routing with IPv6
[ 7.434148] systemd[1]: systemd 237 running in
system mode. (+PAM +AUDIT +SELINUX +IMA +APPARMOR
+SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP +GCRYPT
+GNUTLS +ACL +XZ +LZ4 +SECCOMP +BLKID +ELFUTILS
+KMOD -IDN2 +IDN -PCRE2 default-hierarchy=hybrid)
[ 7.458784] systemd[1]: Detected architecture
arm.
[ 7.488449] systemd[1]: Set hostname to .
[ 7.784617] random: systemd: uninitialized
urandom read (16 bytes read)
```

```
[ 7.792320] systemd[1]: Started Forward
Password Requests to Wall Directory Watch.
[ 7.815444] random: systemd: uninitialized
urandom read (16 bytes read)
[ 7.824240] systemd[1]: Created slice System
Slice.
[ 7.843429] random: systemd: uninitialized
urandom read (16 bytes read)
[ 7.850952] systemd[1]: Listening on udev
Kernel Socket.
[ 7.871646] systemd[1]: Listening on Journal
Socket.
[ 7.894204] systemd[1]: Mounting POSIX Message
Queue File System...
[ 7.920531] systemd[1]: Created slice system-
getty.slice.
[ 7.939629] systemd[1]: Listening on fsck to
fsckd communication Socket.
[ 8.166377] EXT4-fs (mmcblk1p2): re-mounted.
Opts: errors=remount-ro
[ 8.569110] systemd-journald[277]: Received
request to flush runtime journal from PID 1
[ 9.073008] random: crng init done
[ 9.073021] random: 7 urandom warning(s) missed
due to ratelimiting
[ 9.160823] input: gpio_keys as
/devices/platform/gpio_keys/input/input0
[ 9.593187] input: HID 0b38:0010 as
/devices/platform/soc/soc:usb3-
0/12000000.dwc3/xhci-hcd.1.auto/usb3/3-1/3-1.1/3-
1.1.2/3-1.1.2:1.0/0003:0B38:0010.0001/input/input1
[ 9.651666] hid-generic 0003:0B38:0010.0001:
input,hidraw0: USB HID v1.10 Keyboard [HID
0b38:0010] on usb-xhci-hcd.1.auto-1.1.2/input0
[ 9.654992] input: HID 0b38:0010 System Control
as /devices/platform/soc/soc:usb3-
0/12000000.dwc3/xhci-hcd.1.auto/usb3/3-1/3-1.1/3-
1.1.2/3-1.1.2:1.1/0003:0B38:0010.0002/input/input2
[ 9.711625] input: HID 0b38:0010 Consumer
Control as /devices/platform/soc/soc:usb3-
0/12000000.dwc3/xhci-hcd.1.auto/usb3/3-1/3-1.1/3-
1.1.2/3-1.1.2:1.1/0003:0B38:0010.0002/input/input3
[ 9.711862] hid-generic 0003:0B38:0010.0002:
input,hidraw1: USB HID v1.10 Device [HID
0b38:0010] on usb-xhci-hcd.1.auto-1.1.2/input1
[ 9.714123] input: PixArt USB Optical Mouse as
/devices/platform/soc/soc:usb3-
0/12000000.dwc3/xhci-hcd.1.auto/usb3/3-1/3-1.1/3-
1.1.3/3-1.1.3:1.0/0003:093A:2510.0003/input/input4
[ 9.714502] hid-generic 0003:093A:2510.0003:
input,hidraw2: USB HID v1.11 Mouse [PixArt USB
Optical Mouse] on usb-xhci-hcd.1.auto-1.1.3/input0
[ 9.714617] usbcore: registered new interface
```

```
driver usbhid
[ 9.714622] usbhid: USB HID core driver
[ 9.776583] cfg80211: Loading compiled-in X.509
certificates for regulatory database
[ 9.786327] cfg80211: Loaded X.509 cert
'sforshee: 00b28ddf47aef9cea7'
[ 9.888965] rtl8192cu: Chip version 0x10
[ 9.987081] rtl8192cu: Board Type 0
[ 9.987396] rtl_usb: rx_max_size 15360,
rx_urb_num 8, in_ep 1
[ 9.987542] rtl8192cu: Loading firmware
rtlwifi/rtl8192cufw_TMSC.bin
[ 9.988186] ieee80211 phy0: Selected rate
control algorithm 'rtl_rc'
[ 9.989308] usbcore: registered new interface
driver rtl8192cu
[ 9.997638] usbcore: registered new interface
driver rtl8xxxu
[ 10.251057] FAT-fs (mmcblk1p1): Volume was not
properly unmounted. Some data may be corrupt.
Please run fsck.
[ 11.760287] IPv6: ADDRCONF(NETDEV_UP): wlan0:
link is not ready
[ 11.762438] rtl8192cu: MAC auto ON okay!
[ 11.795947] rtl8192cu: Tx queue select: 0x05
[ 11.998081] rtl8192c_common: chksum report
fail! REG_MCUFWDL:0x00030000 .
[ 12.003466] rtl8192c_common: Firmware is not
ready to run!
[ 12.358230] IPv6: ADDRCONF(NETDEV_UP): wlan0:
link is not ready
[ 12.469419] IPv6: ADDRCONF(NETDEV_UP): wlan0:
link is not ready
[ 13.499789] wlan0: authenticate with
2c:0b:e9:be:9c:81
[ 13.511744] wlan0: send auth to
2c:0b:e9:be:9c:81 (try 1/3)
[ 13.515086] wlan0: authenticated
[ 13.519370] wlan0: associate with
2c:0b:e9:be:9c:81 (try 1/3)
[ 13.556477] wlan0: RX AssocResp from
2c:0b:e9:be:9c:81 (capab=0x1401 status=204 aid=0)
[ 13.556484] wlan0: 2c:0b:e9:be:9c:81 denied
```

```
association (code=204)
[ 14.624535] wlan0: authenticate with
2c:0b:e9:be:9c:81
[ 14.648246] wlan0: send auth to
2c:0b:e9:be:9c:81 (try 1/3)
[ 14.751335] wlan0: send auth to
2c:0b:e9:be:9c:81 (try 2/3)
[ 14.803352] wlan0: authenticated
[ 14.807365] wlan0: associate with
2c:0b:e9:be:9c:81 (try 1/3)
[ 14.842205] wlan0: RX AssocResp from
2c:0b:e9:be:9c:81 (capab=0x1401 status=204 aid=0)
[ 14.842211] wlan0: 2c:0b:e9:be:9c:81 denied
association (code=204)
[ 16.316586] wlan0: authenticate with
2c:0b:e9:be:9c:81
[ 16.340248] wlan0: send auth to
2c:0b:e9:be:9c:81 (try 1/3)
[ 16.443345] wlan0: send auth to
2c:0b:e9:be:9c:81 (try 2/3)
[ 16.547327] wlan0: send auth to
2c:0b:e9:be:9c:81 (try 3/3)
[ 16.556576] wlan0: authenticated
[ 16.559338] wlan0: associate with
2c:0b:e9:be:9c:81 (try 1/3)
[ 16.618346] wlan0: RX AssocResp from
2c:0b:e9:be:9c:81 (capab=0x1411 status=0 aid=1)
[ 16.658872] wlan0: associated
[ 16.659055] wlan0: Limiting TX power to 27 (30
- 3) dBm as advertised by 2c:0b:e9:be:9c:81
[ 16.776129] cryptd: max_cpu_qlen set to 1000
[ 16.808727] IPv6: ADDRCONF(NETDEV_CHANGE):
wlan0: link becomes ready
```

Para ver el código fuente, contribuir al proyecto y realizar pull requests, visita el repositorio de GitHub en https://github.com/mihailescu2m/linux/tree/odr_oidxu4-5.0.y. Para comentarios, preguntas y sugerencias, visita el hilo del Foro ODROID en <https://forum.odroid.com/viewtopic.php?f=95&t=33510>.

Cálculos Científicos en Clúster con un ODROID-MC1

© February 1, 2019 By Andreas Lintermann Linux, ODROID-MC1, Tutoriales



El ODROID-MC1 se ha convertido en un interesante sistema clúster para experimentar, por ejemplo, con implementaciones de swarm Docker [1] y con la minería de criptomonedas [2-4]. Tienes una breve introducción sobre el ODROID-MC1 en la referencia [5]. Un sólo ODROID-MC1 consta de cuatro nodos ODROID-XU4 reducidos, cada uno equipado con un Samsung Exynos 5 Octa (5422). El Exynos 5 Octa es un sistema ARM Big.LITTLE de dos zócalos compuesto por las CPUs Cortex-A15 y Cortex-A7 de cuatro núcleos, con una velocidad de reloj de 2 GHz y 1.4 GHz respectivamente. Las CPU cuentan con multiprocesamiento heterogéneo (HMP). Cada nodo está equipado con 2 GB de LPDDR3 RAM y con una GPU Mali-T628 MP6, que admite OpenGL ES 3.1/2.0/1.1 y el perfil completo OpenCL 1.2. Además, las placas cuentan con Ethernet Gigabit y refrigeración activa por ventilador.

En la referencia [6] tienes una presentación general de la computación con clúster utilizando varios

ODROID-XU4s, donde se describe brevemente el concepto de topologías MapReduce. El debate [7] presenta algunos ejemplos de computación paralela de un conjunto Mandelbrot con la librería de paso de mensajes MPJ Express que está implementada en JAVA. En términos de computación científica, JAVA no es la opción más óptima debido a sus limitaciones de rendimiento. En aplicaciones de física e ingeniería, C++, C o FORTRAN siguen siendo los lenguajes dominantes para escribir código científico [8].

Las simulaciones comunes de última generación emplean cientos de miles de núcleos de CPU en sistemas de computación de alto rendimiento (HPC) [8,9] para resolver grandes desafíos sociales. En este contexto, la escalabilidad y el rendimiento del kernel de simulación son clave para una computación multi-núcleo eficiente, es decir, no solo es esencial contar con un código altamente optimizado que se ejecute de manera eficiente en un único núcleo, sino que también sea posible aumentar la eficiencia

informática con un aumento del número de núcleos computacionales. Mientras que el rendimiento de un único núcleo se mejora en general con las técnicas de ajuste del kernel, como la vectorización de bucle, la omisión de errores en la línea de caché y la programación inteligente, la eficiencia paralela se puede medir mediante robustos experimentos de escalado. En estos experimentos, el número de núcleos se duplica continuamente para un problema dado. En el mejor de los casos, el tiempo de resolución se divide con cada duplicado. En este tipo de computación paralela, cada uno de los procesos resuelve un subconjunto del problema original. Con un mayor número de núcleos, también aumenta la sobrecarga de las comunicaciones de red, lo que conduce a una disminución de la eficiencia. Los mejores códigos de simulación son capaces de escalar desde una pequeña cantidad de núcleos hasta cientos de miles de procesos [8-10].

El ODROID-MC1 puede considerarse como un pequeño sistema HPC y además, es ideal para la simulación de problemas científicos de pequeña y mediana escala. En especial su bajo consumo de energía y su bajo precio lo hacen ideal para el desarrollo de códigos paralelos y para ser adquiridos en departamentos o empresas más pequeñas, o en universidades con fines docentes.

En el siguiente texto se describe cómo configurar un sistema de clúster con un sistema de archivos compartido mediante el Sistema de archivos en red (NFS) y la Interfaz de transmisión de mensajes (MPI) junto con el programador de tareas de clúster SLURM. Se proporcionan ejemplos de cómo ejecutar cálculos paralelos en este sistema. La Mecánica de Fluidos Computacional (CFD) corrobora la aplicabilidad del ODROID-MC1 para resolver problemas científicos. Los pasos que se presentan son los detalles técnicos que están detrás de las simulaciones y los análisis tratados[10].

Configurar el sistema del clúster

El sistema del clúster consta de un nodo frontal, que es un único ODROID-XU4 equipado con un módulo eMMC 5.0 de 16 GB, el ODROID-MC1, un Rackstation RS816 de Synology de 4 bahías, un Switch Ethernet Gigabit de 50 puertos gestionado via Web GSSAE

ProSAFE y una puerta de enlace a internet. La Figura 1 muestra una foto de la configuración actual montada en un rack de 19".

El RS816 hace de servidor DHCP y NFS y está conectado al switch con una configuración de conexión con doble puerto. Obviamente, el RS816 y el GS750E pueden ser reemplazados por cualquier otro servidor y switch que tengas la misma funcionalidad, por ejemplo, el propio nodo central (con interfaz de usuario) puede actuar como servidor DHCP y NFS. Para la siguiente explicación, se supone que el acceso a Internet es a través de la puerta de enlace con la IP local 192.168.1.1, el servidor denominado "FS" está instalado y funcionando y éste tiene la dirección IP 192.168.1.2. En este ejemplo, el servidor exporta los tres directorios a través de NFS:

- /homes/ (almacenará los directorios home de los usuarios)
- /netopt/ (contendrá software compartido)
- /work/ (será utilizado como espacio de trabajo para realizar los cálculos)

El nodo central se identificará con el nombre "FE" y se le asignará la IP 192.168.1.100. El resto de nodos del clúster se llamarán CL [1 ... 4] y tendrán las direcciones IP 192.168.1.101 hasta la 192.168.1.104. Primero, vamos a tratar la configuración del nodo central antes abordar la instalación del software genérico y la instalación de los nodos del clúster.

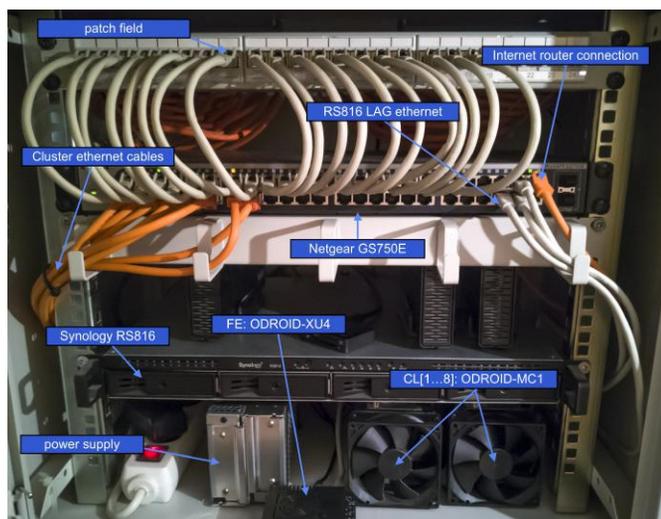


Figura 1 - Imagen de la configuración del ODROID-MC1 (aquí, son visibles dos sistemas MC1)

Instalación del nodo central

En primer lugar, Debemos instalar la imagen de Ubuntu Linux en el módulo eMMC del nodo central ODROID-XU4 o en una tarjeta SD. Puedes encontrar una buena explicación paso a paso de este proceso online en [11]. Para el resto de nodos del cluster es suficiente con instalar la imagen mínima de Ubuntu. Ten en cuenta que la instalación del clúster ha sido probada con Ubuntu Linux 16.04 Xenial. Los detalles de la instalación del sistema clúster con Ubuntu Linux 18.04 Bionic la puedes encontrar en la Sec. 5. Para instalar y configurar el nodo central, las tareas que se detallan a continuación deben realizarse como superusuario. Lo primero que debe hacer tras iniciar sesión es fijar una nueva contraseña y generar una clave para iniciar sesión con facilidad

```
$ passwd
$ ssh-keygen -t rsa
```

Esto instalará la clave ssh para root en /root/.ssh/id_rsa. Esta clave será copiada en un paso posterior a los nodos del clúster, permitiendo así una fácil administración de los mismos.

Configuración de la dirección IP y del nombre de host

En el sistema FE, será asignada la IP fija 192.168.1.100 a través de la actualización del archivo /etc/network/interfaces:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static address 192.168.1.100
gateway 192.168.1.1
netmask 255.255.255.0
dns-nameservers 192.168.1.1
```

El nombre del sistema se puede actualizar añadiendo

```
FE 192.168.1.100
```

a /etc/hosts y reemplazando odroid en /etc/hostname por FE. Además, para que el sistema sea consciente de la existencia del resto de nodos del clúster, es decir, si tu servidor DHCP no asigna los nombres correctos a los nodos, hay que añadirlos, FS y FE a /etc/hosts también.

```
CL1 192.168.1.101
CL2 192.168.1.102
CL3 192.168.1.103
CL4 192.168.1.104
FS 192.168.1.2
```

Reiniciar el sistema tiene mucho sentido en esta fase de la configuración.

Instalación y configuración del montador automático

Para montar automáticamente los recursos compartidos del servidor NFS, es necesario instalar el montador automático a través de:

```
$ apt-get update
$ apt-get upgrade
$ apt-get install autofs
```

Luego, modifica el archivo /etc/auto/master y añade lo siguiente al final del archivo:

```
/nfs_mounts /etc/auto.nfs
```

Acto seguido, crea el archivo /etc/auto.nfs con el siguiente contenido:

```
netopt -fstype=nfs,rw,soft,tcp,noexec,uid=user
FS:/volume1/shares/netopt
homes -fstype=nfs,rw,soft,tcp,noexec,uid=user
FS:/volume1/shares/homes
work -fstype=nfs,rw,soft,tcp,noexec,uid=user
FS:/volume1/shares/work
```

Esto monta los recursos NFS desde el servidor NFS con el nombre de usuario user ubicado en /volume1/shares en el servidor. Ten en cuenta que el nombre de usuario debe existir en ambos sistemas FS y FE. Además, crea los correspondientes directorios:

```
$ mkdir /nfs_mounts
$ mkdir /netopt
$ cd /netopt
$ ln -s /nfs_mounts/netopt
$ cd /home
$ ln -s /nfs_mounts/homes/user
$ cd /home/user
$ ln -s /nfs_mounts/work
```

Ten en cuenta que las carpetas /nfs_mounts/netopt, /nfs_mounts/homes y /nfs_mounts/work no existen

en este paso, sin embargo, estarán disponibles al iniciar el montador automático. Por lo tanto, ejecuta:

```
$ service autofs restart
```

También tiene sentido que para user y root se agreguen las rutas del software que se instalará en la siguiente sección para las variables de entorno del directorio de búsqueda de rutas. Por lo tanto, añada al archivo `~/.profile`.

```
$ export PATH=$PATH:/netopt/mpich/bin
$ export PATH=/netopt/slurm/bin:$PATH
$ export PATH=/netopt/munge/bin:$PATH
```

Instalación del software para la computación en paralelo del cluster.

Todo el software compartido se instalará en el recurso NFS compartido `/netopt`. Todas las fuentes se descargarán y configurarán en el subdirectorio `/netopt/install`. El siguiente paso se realiza en el FE central y asume que hay disponible un compilador como `llvm` o el compilador GNU.

Instalación de MPICH

Para permitir el desarrollo de software paralelo, es necesario instalar una librería de comunicación paralela. En este ejemplo, instalaremos la librería MPICH versión 3.2.1, que está disponible en www.mpich.org, con los siguientes comandos:

```
$ cd /netopt/install
$ mkdir mpich
$ cd mpich
$ tar -xvf mpich-3.2.1

$ cd mpich-3.2.1
$ ./configure --enable-mpi-cxx --
prefix=/netopt/mpich-3.2.1
$ make -j 4
$ make install

$ cd /netopt
$ ln -s mpich-3.2.1 mpich
```

Instalación y configuración de MUNGE

Para instalar el programador automático de tareas SLURM, debemos instalar los servicios MUNGE (MUNGE 0.5.13; disponibles en <https://dun.github.io/munge/>). MUNGE es un servicio

de autenticación para crear y validar credenciales que es necesario para la programación autenticada. Para instalar MUNGE primero escribe

```
$ apt-get install munge
```

Esto permite tener a mano todos los scripts de inicio y de ejecución de servicios necesarios. Para instalar, sin embargo, la última versión de MUNGE, el código fuente que hemos mencionado anteriormente se descarga y almacena en `/netopt/install`. Para compilar MUNGE e instalarlo ejecuta:

```
$ cd /netopt/install
$ mkdir munge
$ cd munge
$ tar -xvf munge-0.5.13.tar.gz

$ cd munge-munge-0.5.13 .
$ ./configure --prefix=/netopt/munge-0.5.13
$ make -j 4
$ make install

$ cd /netopt
$ ln -s munge-0.5.13 munge

$ cd munge
$ mv etc etc.old
$ mv var var.old
$ ln -s /etc
$ ln -s /var
```

Tenga en cuenta que los registros log de MUNGE se escribirán en el sistema de archivos local `/var` y la configuración se realizará en `/etc`. Para configurar MUNGE, debe generarse una clave secreta MUNGE:

```
$ dd if=/dev/random bs=1 count=1024 >
/etc/munge/munge.key
```

Ten en cuenta que en un paso posterior (consulta la Sección 2.3.2), el archivo `/etc/munge/munge.key` se copiará al resto de nodos del clúster. Además, dado que la instalación compilada de MUNGE reemplaza la versión previamente instalada, el enlace al ejecutable de MUNGE debe actualizarse:

```
$ cd /usr/sbin
$ mv munged munged.old
$ ln -s /netopt/munge/sbin/munged
```

Instalación de PMIX

Otra herramienta que debe instalarse es PMIX (en este caso PMIX 2.1.0; disponible en <https://github.com/pmix/pmix/releases>):

```
$ cd /netopt/install
$ mkdir pmix
$ cd pmix
$ tar -xvf pmix-2.1.0.tar.gz

$ cd pmix-2.1.0
$ ./configure --prefix=/netopt/pmix-2.1.0
$ make -j 4
$ make install

$ cd /netopt
$ ln -s pmix-2.1.0 pmix
```

Instalación y configuración de SLURM

Finalmente, instalaremos el programador SLURM (SLURM 17.11.3-2; disponible en <https://slurm.schedmd.com>). Similar a MUNGE, primero instalaremos Ubuntu SLURM con

```
$ apt-get slurm-llnl libslurm-dev
```

Después, instalaremos la última versión en /netopt con

```
$ cd /netopt/install
$ mkdir slurm
$ cd slurm
$ tar -xvf slurm-17.11.3-2.tar.gz

$ cd slurm-17.11.3-2
$ ./configure --prefix=/netopt/slurm-17.11.3-2 --
sysconfdir=/etc/slurm-llnl --with-
munge=/netopt/munge
--with-pmix=/netopt/pmix
$ make -j 4
$ make install

$ cd /netopt
$ ln -s slurm-17.11.3-2 slurm
```

Para configurar SLURM, el archivo /etc/slurm-llnl/slurm.conf debe modificarse con

```
# GENERAL
ControlMachine=FE
AuthType=auth/munge
CryptoType=crypto/munge
MpiDefault=none
```

```
ProctrackType=proctrack/pgid
ReturnToService=1
SlurmctldPidFile=/var/run/slurm-llnl/slurmctld.pid
SlurmdPidFile=/var/run/slurm-llnl/slurmd.pid
SlurmdSpoolDir=/var/spool/slurmd
SlurmUser=slurm
StateSaveLocation=/var/spool/slurmctld
SwitchType=switch/none
TaskPlugin=task/affinity
TaskPluginParam=sched

# SCHEDULING
FastSchedule=1
SchedulerType=sched/backfill
SelectType=select/cons_res
SelectTypeParameters=CR_Core

# LOGGING AND ACCOUNTING
AccountingStorageType=accounting_storage/none
ClusterName=odroid
JobAcctGatherType=jobacct_gather/none
SlurmctldDebug=verbose
SlurmctldLogFile=/var/log/slurmctld.log
SlurmdDebug=verbose
SlurmdLogFile=/var/log/slurmd.log

# COMPUTE NODES
NodeName=CL[1-4] CPUs=8 RealMemory=1994
State=UNKNOWN
PartitionName=batch Nodes=CL[1-4]
OverSubscribe=EXCLUSIVE Default=YES
MaxTime=INFINITE State=UP
```

En /usr/sbin, actualiza los siguientes enlaces:

```
$ cd /usr/sbin
$ mv slurmctld slurmctld.old
$ mv slurmd slurmd.old
$ mv slurmstepd slurmstepd.old

$ ln -s /netopt/slurm/sbin/slurmctld
$ ln -s /netopt/slurm/sbin/slurmd
$ ln -s /netopt/slurm/sbin/slurmstepd
```

Además, asegúrate de añadir la siguiente carpeta y cambiar los permisos de la siguiente forma

```
$ mkdir /var/spool/slurmctld
$ chown slurm:slurm /var/spool/slurmctld
```

Instalación de los nodos del clúster

Los nodos del cluster usan la imagen mínima de Ubuntu Linux. La siguiente configuración se muestra

a modo de ejemplo para el primer nodo de clúster CL1 con IP 192.168.1.101 y ésta debe aplicarse a todos los nodos de clúster.

Configuración general del nodo del clúster

Tras la instalación de la tarjeta SD, asegúrate de actualizar el sistema:

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

Asegúrate también de copiar la carpeta /root/.ssh de FE a CL1, es decir, en FE ejecuta lo siguiente (asegúrate de que rsync esté instalado):

```
$ rsync -av /root/.ssh CL1:/root/
```

Luego, sigue los pasos de la sec. 2.1.1 y sec. 2.1.2 para tener la dirección IP correcta (192.168.1.101), el nombre de host (CL1) y el montador automático activado.

Integración del sistema cluster

Instala todos los paquetes necesarios en CL1:

```
$ apt-get install munge slurm-llnl libslurm-dev
```

Después, la clave MUNGE generada en la sec. 2.2.2 y que se encuentra dentro de /etc/munge/munge.key en FE y el archivo de configuración SLURM que se encuentra en /etc/slurm-llnl/slurm.conf deben transferirse a CL1 ejecutando:

```
$ rsync -av /etc/munge/munge.key CL1:/etc/munge/
$ rsync -av /etc/slurm-llnl/slurm.conf
CL1:/etc/slurm-llnl/
```

en FE. En este paso tiene sentido reiniciar el nodo del clúster. Después de instalar cada nodo, el sistema está casi listo para realizar cálculos.

Administración del clúster

Para que el planificador de tareas se ejecute, debemos ejecutar los siguientes comandos en los nodos

```
$ sudo service munge start
$ sudo service slurmd start
```

y en FE:

```
$ sudo service munge start
$ sudo service slurmd start
```

Se puede comprobar el estado del nodo con:

```
$ scontrol show nodes
```

o con:

```
$ sinfo -N --long
```

Si uno de los nodos se encuentra en estado DOWN o UNKNOWN, se puede reanudar mediante

```
$ scontrol update NodeName=NAME State=RESUME
```

donde NAME es el nombre del nodo, por ejemplo, CL1.

Envío de tareas

Ahora que el clúster es completamente funcional, podemos enviar tareas al programador, el cual necesita un archivo de trabajo como:

```
#!/bin/bash -x
#SBATCH --nodes=4           // allocates 4 nodes
for the job
#SBATCH --ntasks-per-node=2 // starts 2 MPI
ranks per node
#SBATCH --cpus-per-task=4   // for each MPI
rank per node 4 OpenMP threads are reserved
#SBATCH --output=mpi-out.%j // location of
the output file
#SBATCH --error=mpi-err.%j  // location of
the error file
#SBATCH --time=00:20:00     // wall time of the
job
#SBATCH --partition=batch   // the name of
the partition

export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}
// information for OpenMP

srun --mpi=pmi2 COMMAND    // runs the command
COMMAND in parallel
```

Las tareas se pueden programar para los nodos de diferentes formas, es decir, se puede usar para el cálculo los núcleos de forma individual, los núcleos Cortex-A7 (lentos) o los Cortex-A15 (rápidos), o ambos. Todo esto se configura con el comando srun en el script de trabajo:

```

srun --cpu-bind=verbose,mask_cpu:ABxCD --mpi=pmi2
COMMAND // uses mask ABxCD for scheduling

```

La opción `mask_cpu` permite especificar la máscara para la ejecución. Las máscaras para usar un único sistema Cortex o los dos se muestran en las Tablas 1 y 2.

Bitsets for CPU masking								
fast cores				slow cores				
7	6	5	4	3	2	1	0	HEX code
0	0	0	0	0	0	0	1	0x1
0	0	0	0	0	0	1	0	0x2
0	0	0	0	0	1	0	0	0x4
0	0	0	0	1	0	0	0	0x8
0	0	0	1	0	0	0	0	0x10
0	0	1	0	0	0	0	0	0x20
0	1	0	0	0	0	0	0	0x40
1	0	0	0	0	0	0	0	0x80

Tabla 1: Máscara CPU para núcleos individuales.

Bitsets for CPU masking								
fast cores				slow cores				
7	6	5	4	3	2	1	0	HEX code
0	0	0	0	1	1	1	1	0xf
1	1	1	1	0	0	0	0	0xf0
1	1	1	1	1	1	1	1	0xff

Tabla 2: Máscara CPU para un único Cortex o ambos.

Ejemplo: Simulación de flujo en el ODROID-MC1

Para demostrar que el sistema ODROID-MC1 se puede utilizar para llevar a cabo simulaciones científicas, a continuación, se muestra un ejemplo de

simulación de flujo en una geometría de quemado [10]. Ver la Fig. 2; Lado izquierdo. La simulación utiliza un código de red-Boltzmann [12], que resuelve las ecuaciones de control de la mecánica de fluidos sobre una malla cartesiana que discrimina el espacio, es decir, la ecuación de Boltzmann se resuelve para todas las ubicaciones espaciales dentro de esta malla en el tiempo.

En cada paso de tiempo, en cada ubicación de la malla, se calcula un vector de velocidad y densidad mediante un algoritmo de dos etapas que simula localmente la colisión de partículas en un volumen finito y transporta la información de colisión a las ubicaciones vecinas. La malla es generada por un generador de malla paralelo [13] que se muestra en la Fig. 2 en el lado derecho. Especialmente proximo a las paredes y en la región del chorro de combustión, la malla se refina localmente para tener una resolución suficiente como para capturar las principales características de flujo. La Figura 3 muestra los resultados del cálculo, generado utilizando únicamente los núcleos más rápidos Cortex-A15 del ODROID-MC1. La simulación se ejecuta durante 24 horas. Obviamente, un chorro avanza por la región de la ranura que llega hasta la cámara de combustión. A la izquierda se muestra una sección transversal en la región de la ranura con contornos de la magnitud de la velocidad. El lado derecho muestra contornos tridimensionales de las estructuras voraginosas generadas en diferentes pasos de tiempo de la simulación. Como he mencionado en la introducción, la escalabilidad es un aspecto importante en las simulaciones HPC. Por ello, se suelen realizar intensas mediciones de escalabilidad para todo el sistema.

La Figura 4 muestra la escalabilidad de la simulación utilizando diferentes estrategias de paralelización, es decir, utilizando una simple MPI y estrategias de paralelización híbridas MP/OpenMP, esta última con diferentes opciones de programación OpenMP para bucles paralelos. Las líneas negras representan el comportamiento de la escala ideal. En la Figura 4 se puede ver que, entre todos los casos, la MPI/OpenMP híbrida que utiliza las estrategias de paralelización de bucle guiado es la que mejor funciona y, por lo tanto,

es la técnica elegida para la simulación. El efecto de la sobrecarga de comunicación ya es visible desde la discrepancia hasta la línea negra ideal. Sin embargo, esta configuración no supera el rendimiento de solo usar los núcleos rápidos Cortex-A15 (tal y como se usa para la simulación de quemado). Para más detalles, el lector interesado puede dirigirse a la referencia [10] de donde se han tomado los resultados y que, además, analice el consumo de energía del ODROID-MC1 y compare su rendimiento con los sistemas HPC alemanes de última generación.

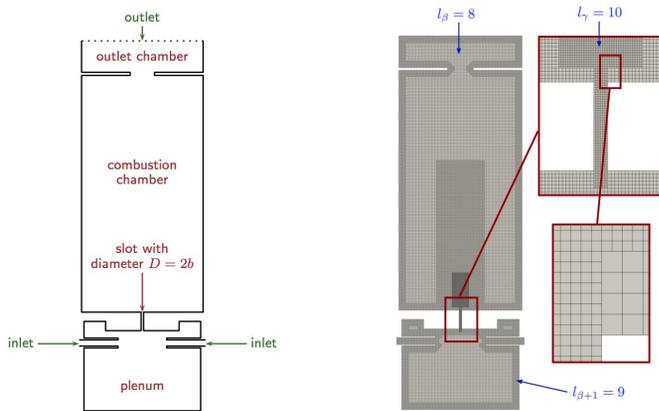


Figura 2: Vista esquemática y malla computacional de una configuración de quemado [10] [10]

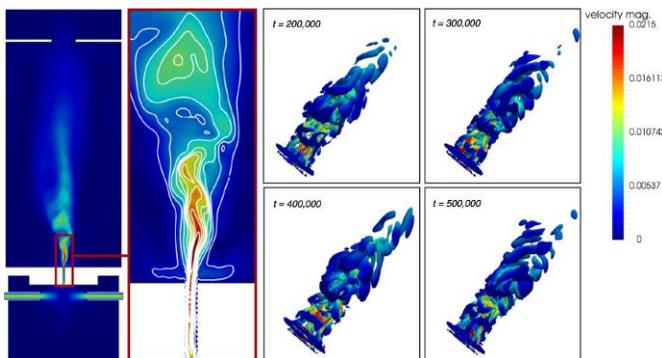


Figura 3: Resultados de la simulación de flujo en una geometría de quemado [10] [10]

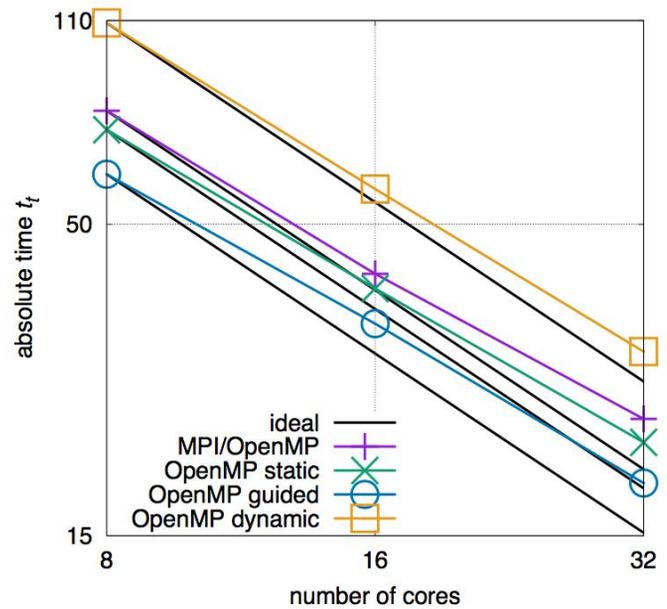


Figura 4: Gran escalabilidad del código lattice-Boltzmann en todo el clúster ODROID-MC1 utilizando diferentes estrategias de paralelización [10] [10]

Resumen y conclusión

El ODROID-MC1 es un sistema prometedor para el funcionamiento de un clúster y para la simulación de pequeños y medianos problemas científicos. La correspondiente instalación del software es sencilla. Este artículo nos ha proporcionado un manual paso a paso sobre cómo configurar el sistema de clúster para llevar a cabo computación paralela utilizando MPI con MPICH y PMIX. El programador automático de tareas SLURM utiliza MUNGE para la autenticación y permite fijar tareas para ejecutar simple trabajos MPI y MPI/OpenMP híbrido. Un ejemplo de simulación de flujo en una configuración de quemado ha demostrado que el ODROID-MC1 es un sistema adecuado para la simulación de tipo de problemas. La escalabilidad del software de simulación en el sistema es mas que suficiente para calcular soluciones en un tiempo razonable. Es decir, para los pequeños departamentos o grupos de investigación el ODROID-MC1 es una alternativa rentable frente a los sistemas HPC basados en x86 si las simulaciones a gran escala no son el objetivo principal.

Comentarios adicionales

En lugar de instalar cada nodo del clúster de forma individual, también es posible instalar el arranque PXE y hacer que cada nodo arranque a través de la red

desde TFTP. El sistema de archivos root se importa a través de NFS desde un servidor de archivos. Puedes encontrar una guía detallada sobre cómo configurar el arranque PXE en ODROID-XU4 en las páginas de la Wiki de ODROID [14]. Usando la última versión de Linux Ubuntu 18.04 Bionic, son necesarios realizar algunos cambios en el proceso de instalación. En primer lugar, la configuración de la red ha cambiado de una configuración en `/etc/network/interfaces` a una configuración a través de `netplan`. Es decir, en lugar de modificar `/etc/network/interfaces`, debe crearse el archivo `/etc/netplan/01-networkd.yaml` con el siguiente contenido (ejemplo para CL1)

```
network:
  ethernets:
    eth0:
      addresses: [192.168.1.101/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [192.168.1.1]
      dhcp4: no
  version: 2
```

Asegúrate de no usar tabulaciones en el archivo para el sangrado. Después de esto puedes ejecutar

```
$ netplan apply
$ netplan --debug apply
```

que debería cambiar tu IP de inmediato. Además, la versión SLURM en el repositorio de Ubuntu en Bionic es diferente, la cual necesitas instalar

```
$ apt-get install slurm-wlm
```

en lugar del paquete `slurm-llnl`. Este todavía proporciona la misma estructura de directorios que `slurm-llnl`, así que no hay que hacer más cambios.

Referencias

- [1] A. Yuen, ODROID-MC1 Docker Swarm: Getting Started Guide, Odroid Magazine (46)(2017)
- [2] E. Kisiel, Prospectors, Miners, and 49er's: Dual GPU-CPU Mining on the ODROID-XU4/MC1/HC1/HC2, Odroid Magazine (51)(2018).
- [3] E. Kisiel, Prospectors, Miners, and 49er's – Part 2: Dual GPU-CPU Mining on the ODROID-

XU4/MC1/HC1/HC2, Odroid Magazine (52)(2018).

[4] E. Kisiel, Prospectors, Miners, and 49er's – Part 3: Operation and Maintenance of Crypto-Currency Mining Systems, Odroid Magazine (53)(2018).

[5] R. Roy, ODROID-HC1 and ODROID-MC1: Affordable High-Performance And Cloud Computing At Home, Odroid Magazine (45)(2017).

[6] M. Kamprath, ODROID-XU4 Cluster, Odroid Magazine (53)(2018).

[7] A. Yuen, ODROID-MC1 Parallel Programming: Getting Started, Odroid Magazine (46)(2017).

[8] D. Brömmel, W. Frings, B. J. N. Wylie, B. Mohr, P. Gibbon, T. Lippert, The High-Q Club: Experience Extreme-scaling Application Codes. *Supercomputing Frontiers and Innovations*, 5(1), 59–78 (2018). [doi:10.14529/jsfi180104](https://doi.org/10.14529/jsfi180104)

[9] A. Pogorelov, M. Meinke, W. Schröder, Cut-cell method based large-eddy simulation of tip-leakage flow. *Physics of Fluids*, 27(7), 075106 (2015). [doi:10.1063/1.4926515](https://doi.org/10.1063/1.4926515)

[10] A. Lintermann, D. Pleiter, W. Schröder, Performance of ODROID-MC1 for scientific flow problems, *Future Generation Computer Systems* (in press, first online: Jan. 04, 2019). [doi:10.1016/j.future.2018.12.059](https://doi.org/10.1016/j.future.2018.12.059)

[11] Odroid Wiki https://wiki.odroid.com/troubleshooting/odroid_flashing_tools

[12] R.K. Freitas, A. Henze, M. Meinke, W. Schröder, Analysis of Lattice-Boltzmann methods for internal flows. *Computers & Fluids*, 47(1), 115–121 (2011). [doi:10.1016/j.compfluid.2011.02.019](https://doi.org/10.1016/j.compfluid.2011.02.019)

[13] A. Lintermann, S. Schlimpert, J. H. Grimmen, C. Günther, M. Meinke, W. Schröder, W. Massively parallel grid generation on HPC systems, *Computer Methods in Applied Mechanics and Engineering* 277, 131–153 (2014). [doi:10.1016/j.cma.2014.04.009](https://doi.org/10.1016/j.cma.2014.04.009)

[14] PXE boot on ODROID-XU4/MC1/HC1, https://wiki.odroid.com/odroid-xu4/application_note/software/pxe_boot

Amibian.js: Emulando un Commodore Amiga en un clúster ODRROID-XU4

© February 1, 2019 By Jon L. Aasenden ODRROID-XU4, Tutoriales



Amibian es lo todo que necesitas para convertir tu ODRROID en un Amiga. Es una imagen de tarjeta SD muy liviana apta para tarjetas SD de 2 GB en adelante. Está diseñada para ofrecerte la mejor experiencia de Amiga que puedes conseguir sin tener un Amiga real. Amibian te va a permitir recordar, revivir y redescubrir el placer de tener un Amiga fácilmente con un hardware barato y un esfuerzo mínimo.

A principios de este mes, anuncié que el hardware oficial estaría basado en los SBCs (ordenadores de placa reducida) de la línea ODRROID-XU4 de Hardkernel que ofrecen buen rendimiento, estabilidad excepcional y bajo consumo de energía a un precio bastante razonable. También comenté que la primera configuración de Amibian.js consistiría en cinco placas ODRROID-XU4 funcionando conjuntamente en un clúster, lo que significa que la carga de trabajo queda dividida entre estas cinco

placas para lograr una alta eficiencia y que se compartan los recursos disponibles.

Amibian.js está ganando cada vez más adeptos a medida que más y más desarrolladores, arquitectos de sistemas integrados, jugadores y aficionados a los ordenadores retro descubren el proyecto. ¡Tengo que admitir que estoy muy contento con lo que estamos desarrollando!

Sin embargo, al igual que ocurre con cualquier nueva tecnología o invento, existen dos trampas muy comunes en las que la gente suele caer: La primera es subestimar seriamente una tecnología. Sin duda, JavaScript incita a ello, porque hace solo una década este lenguaje era poco más que un juguete. Desde entonces, JavaScript ha evolucionado hasta convertirse en el lenguaje de programación más ampliamente usado en el todo el mundo, y los motores de tiempo de ejecución como el V8 de Google ejecutan el JavaScript casi tan rápido como el

código binario compilado. “Nativo” viene a ser un código máquina como el producido por un compilador C / C ++, compilador Pascal o cualquier otra cosa que produzca programas que se ejecuten bajo Linux o Windows.

Es necesario llevar a cabo algunos ajustes, especialmente por los programadores tradicionales que no han prestado demasiada atención hacia donde van los navegadores, pero hace mucho que está entre nosotros el JavaScript interpretado. El JavaScript moderno primero se analiza, se tokeniza y se compila en bytecodes. Luego, estos bytecodes se compilan en JIT (“justo a tiempo”, lo que significa que la compilación se realiza dentro del navegador) sobre un código de máquina real utilizando técnicas de vanguardia (LLVM). Así que el JavaScript del 2018 no se parece para nada al JavaScript del 2008.

La segunda trampa en la que puedes caer es en exagerar lo que puede hacer una nueva tecnología y añadir posibilidades y expectativas a un producto que simplemente no se puede presentar. Para mí es muy importante que la gente no caiga en ninguna de las dos trampas, y que todo el mundo conozca lo que realmente es Amibian.js y lo que puede llegar a ofrecer, así como lo que no. Roma no se construyó en un día, y es de sabios estudiar todos los factores antes de emitir un juicio.

Me siento muy afortunado de ver cómo las personas han apoyado financieramente el proyecto a través de Patreon y, como tal, siento que es mi deber documentar y explicar todo cuanto me sea posible. Soy programador y a menudo olvido que no todo el mundo entiende de lo que estoy hablando. Todos somos humanos y cometemos errores. Esperemos que este artículo dibuje una imagen más clara de lo que es Amibian.js y de lo que estamos desarrollando. El proyecto está dividido en dos fases: una primera, donde terminaremos Amibian.js; y una segunda, donde escribiremos un clon de Visual Studio que se ejecute totalmente en el navegador.

¿Qué diablos es Amibian.js?

Amibian.js es un conjunto de servicios y librerías que combinados crean un sistema operativo portátil que se procesa en HTML5. Un sistema que ha sido

escrito utilizando la tecnología web fácilmente disponible, y fue diseñado para ofrecer funcionalidades de escritorio avanzadas con aplicaciones web.

Los servicios que conforman Amibian.js se diseñaron para ser usados sobre una delgada capa de Linux, donde Linux se ocupa del hardware, los controladores y los aspectos básicos que damos por sentado. No tiene sentido intentar escribir un kernel mejor en 2018, porque nunca vas a alcanzar a Linus Torvalds. Es mucho más interesante llevar la tecnología web moderna a los límites absolutos, y desarrollar un sistema que sea verdaderamente portátil y distribuido.



Figura 1: Amibian.js está creado en Smart Pascal y compilado con JavaScript

La capa de servicio está escrita completamente en node.js (JavaScript), lo cual garantiza el mismo comportamiento independientemente de la plataforma del host. Uno de los beneficios de usar la tecnología web es que puedes copiar físicamente todo el sistema de una máquina a otra sin necesidad de tener que realizar cambios. Por lo tanto, si tienes un sistema Amibian.js ejecutándose en tu PC x86 y copias todos los archivos en un ordenador ARM, ni siquiera tiene que volver a compilar el sistema. Simplemente arranca los servicios y estarás de vuelta en la partida.

Ahora, antes de descartar esto como “otro simple modelo web”, recuerda lo que he dicho sobre el JavaScript: el JavaScript del 2018 no es el JavaScript del 2008. Ningún otro lenguaje en el planeta ha visto tanto desarrollo como el JavaScript, y ha evolucionado desde un “juguete de navegador” hasta convertirse en el lenguaje de programación más importante de nuestra era.

Así que Amibian.js no es un simple modelo superficial de un ordenador de escritorio (el señor sabe que hay muchos de esto online). Ejecuta tecnologías avanzadas como el mapeo de sistemas de archivos remotos, un protocolo de mensajes orientado a objetos (Ragnarok), RPCS (pila de invocación de llamada a procedimiento remoto), funciones de códec de video y mucho más, todo ello hecho con JavaScript.

De hecho, una de las demos con las que se envía Amibian.js incluye el Quake III, compilado nuevamente en JavaScript. Ofrece 120 fps sin fallos (el navegador está limitado a 60 fps) y hace uso completo de las tecnologías estándar de navegador (WebGL). En efecto, el JavaScript del que estamos hablando es bastante puntero. La mayor parte de Amibian.js está compilado como "Asm.js", lo que significa que el tiempo de ejecución del V8, el código que ejecuta JavaScript dentro del navegador o como un programa bajo node.js, compilará JIT en un código de máquina altamente eficiente, que es por ello que Amibian.js puede hacer cosas que la gente ni si quiera imagina.

¿De qué se compone Amibian.js?

Amibian.js tiene muchos componentes, pero podemos dividirlo en dos grandes categorías:

- Un cliente de escritorio HTML5
- Un servidor de sistema y varios procesos secundarios

Estas dos categorías guardan exactamente la misma relación que el escritorio X y el kernel de Linux. El cliente se conecta al servidor, activa determinados procedimientos para realizar alguna tarea y luego representa visualmente la respuesta. Es idéntico a cómo el escritorio X llama a las funciones en el kernel o a una de las librerías de Linux. La diferencia entre un sistema operativo tradicional, basado en código máquina y nuestra variante web, es que nuestra versión no tiene que preocuparse por el hardware. También podemos asignar diferentes roles a Amibian.js. Trataremos esta cuestión más adelante.

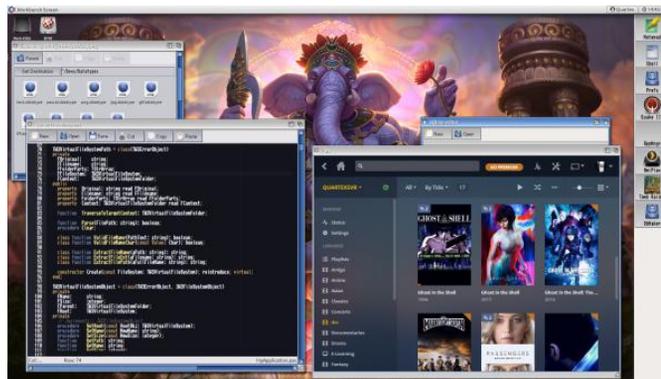


Figura 2: Disfrutar de aplicaciones en la nube es fácil con Amibian.js. Aquí tienes Plex, un sistema basado en el mismo concepto que Amibian.js

Para que así conste: estoy intentando evitar un simple sistema operativo, de lo contrario habría escrito el sistema usando un lenguaje de programación nativo como C u Object Pascal. Así que no uso JavaScript porque me falta habilidad en los idiomas nativos, estoy usando JavaScript porque el código nativo no es aplicable a las tareas que resuelve Amibian.js. Si usara un back-end nativo, podría haber terminado el proyecto en un par de meses, pero un servidor nativo no podría replicarse entre las instancias de la nube porque el chipset y la CPU serían factores determinantes.

El servidor Amibian.js no es un programa único. El back-end para Amibian.js se compone de varias aplicaciones de servicio (demonios en Linux) que ofrecen características específicas. La funcionalidad combinada de estos servicios conforma "el kernel Amibian" en nuestra analogía con Linux. Puede pensar en estos servicios como archivos de librerías en un sistema tradicional, y los programas que están escritos para Amibian.js pueden utilizarlos para una amplia gama de tareas. Puede ser tan simple como leer un archivo, o tan complejo como registrar un nuevo usuario o solicitar derechos de administrador.

La mayor fortaleza de Amibian.js es que está diseñado para ejecutarse en un clúster, utilizando tantos núcleos de CPU como le sea posible. También está diseñado para escalar, lo que significa que se replicará y dividirá el trabajo entre las diferentes instancias. Aquí es donde las cosas se ponen interesantes, ya que un clúster Amibian.js no necesita el hardware más moderno e increíble que exista para ofrecer un buen rendimiento. Puedes montar un

clúster con los viejos PCs de tu oficina o con un puñado de placas integradas. Un ODROID-XU4, una Raspberry Pi o una Tinker Board son candidatos perfectos.

¿Por qué no seguir adelante con Linux?

Es una pregunta aceptable, y aquí es donde entran en juego los roles que mencioné anteriormente. Como desarrollador de software, muchos de mis clientes trabajan con dispositivos integrados y sistemas Kiosco. Tenemos empresas que fabrican routers y decodificadores, cajas NAS de diversa complejidad, sistemas de ticket para trenes y autobuses; Y todas ellas terminan teniendo que resolver las mismas necesidades.

Lo que todos y cada uno de estos fabricantes tiene en común es la necesidad de un sistema de escritorio web que pueda adaptarse a un programa específico. Cualquiera puede escribir una aplicación web, pero cuando necesitas un acceso seguro al sistema de archivos, APIs unificadas que puedan delegar señales a Amazon, Azure o al servidor de tu empresa, las cosas se complican de repente. Incluso cuando tiene todo eso, aún necesitas un modelo de aplicación sólido y apto para la informática distribuida. Es posible que tengan 1 stand de ticket, o 10.000 repartidos por todo el país. No hay sistemas disponibles que estén diseñados para lidiar con la tecnología web a esa escala. Todavía.

Veamos un par de escenarios de la vida real que he encontrado. Estoy seguro de que identificarás una necesidad común. Existen algunos roles que Amibian.js puede asumir para ayudar a presentar una solución con rapidez. También te dará algunas ideas de las posibilidades económicas.

Ten en cuenta que estamos hablando de JavaScript, no de código nativo. Existen muchas soluciones nativas por ahí, pero aquí la cuestión principal es olvidarse de la CPU, los chips, y el objetivo es tener un sistema flotante independientemente de lo que haya debajo.

- Cuando quieres cambiar algunas configuraciones en tu router, inicias sesión en tu router. Éste contiene un pequeño servidor Apache (o algo similar) que te permite realizar todo su mantenimiento a través de

una interfaz web. La interfaz web suele ser bastante superficial, es un fastidio trabajar con ella, y le suponen auténticos quebraderos de cabeza a los desarrolladores actualizarla, ya que está conectada a un módulo Apache nativo que depende 100% del firmware. Cada proveedor termina reinventando la rueda una y otra vez..

- Cuando visitas un gran museo observas las pantallas. Un museo necesita mostrar contenido multimedia, preferiblemente en dispositivos con capacidad táctil, a lo largo de diferentes objetos expuestos. El coste de hacer que un desarrollador cree aplicaciones nativas que muestren contenidos multimedia, reproduzca películas y ofrezca retroalimentación visual es enorme. Esta es la razón por la que la mayoría de los museos adoptan la tecnología web para gestionar la interacción y la presentación de los contenidos multimedia, una vez más reinventando la rueda con diferentes niveles de éxito.
- Los hoteles también tienen más o menos exactamente la misma necesidad, pero a menor escala, especialmente los hoteles más grandes donde el vestíbulo cuenta con puestos de información, y cada habitación muestra una interfaz web a través del televisor.
- Los centros comerciales se enfrentan al mismo desafío y, dependiendo del tamaño, pueden necesitar cualquier cosa, desde uno hasta cien nodos.
- Las escuelas gastan millones en software de formación y lenguajes de programación cada año. Amibian.js puede ofrecer ambos, lo cual permite a las escuelas pagar solo por el mantenimiento y la adaptación; el producto en sí es gratuito. ¡Los niños obtienen el beneficio de aprender lenguajes tradicionales y disfrutar del feedback visual instantáneo! Pueden aprender Basic, Pascal, JavaScript y C. Creo firmemente que los lenguajes clásicos les ayudarán a ser mejores programadores a medida que vayan evolucionando.

Probablemente estés empezando a ver el denominador común que existe: todos necesitan un sistema de escritorio basado en web, que pueda ejecutar aplicaciones multimedia complejas basadas en HTML5 y que proporcione la misma exhaustividad que un sistema operativo nativo, lo cual es bastante difícil de lograr con solo JavaScript.

Amibian.js proporciona una base rica de más de 4000 clases que los desarrolladores pueden usar para escribir aplicaciones voluminosas, complejas y ricas en contenido multimedia (consulta Smart Mobile

Studio más abajo). Al igual que Linux y Windows que proporcionan una gran cantidad de librerías y funciones para el desarrollo de aplicaciones nativas, Amibian.js aspira a proporcionar lo mismo para la nube y los sistemas integrados. Como su nombre lo indica, Amibian.js tiene sus orígenes en el pasado en la máquina que definió el contenido multimedia, el Commodore Amiga. El parentesco es más que simplemente visual: Amibian.js usa la misma arquitectura de sistema porque creemos que es uno de los mejores sistemas jamás diseñado.

Si JavaScript es tan pobre, ¿por qué deberíamos confiar en ti para que nos ofrezcas tanto?

En primer lugar, no estoy vendiendo nada. No es que este proyecto sea algo que me vaya a aportar un montón de dinero. Solicito soporte durante el período de desarrollo porque quiero destinar el suficiente y adecuado tiempo para ello, pero cuando termine Amibian.js será gratis para todo el mundo (LGPL). Y también lo estoy escribiendo porque es algo que necesito y que no he visto en ningún otro sitio. Creo que tienes que escribir software para ti mismo, de lo contrario la calidad se verá mermada.

En segundo lugar, escribir Amibian.js en JavaScript puro y duro con la misma cantidad de funciones y exhaustividad llevaría años de trabajo. La razón por la que puedo ofrecer tanta funcionalidad en tan poco tiempo es porque uso un sistema de compilación llamado Smart Mobile Studio. Éste me ahorra meses y años de desarrollo, pudiendo usar todos los beneficios de la OOP (programación orientada a objetos).

Antes de empezar el proyecto Amibian.js, pasé aproximadamente unos 9 años creando Smart Mobile Studio. Smart no es un proyecto en solitario, muchas personas han participado en el mismo. El producto proporciona un compilador, IDE (editor y herramientas) y una extensa librería en tiempo de ejecución de clases precompiladas (aproximadamente 4000 clases listas para usar, o bloques de compilación).

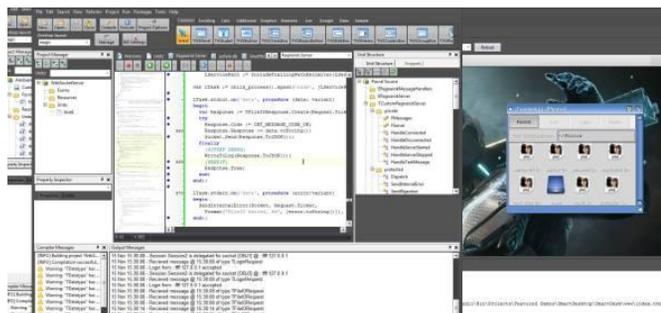


Figura 3 - ¡Escribir servicios node.js a gran escala en Smart es fácil, divertido y muy eficiente!

A diferencia de otros sistemas de desarrollo, Smart Mobile Studio compila en JavaScript en lugar de código máquina. Hemos dedicado mucho tiempo para asegurarnos de usar el OOP más adecuado, y hemos pasado más de tres años perfeccionando un entorno de trabajo de aplicación visual con la misma exhaustividad que VCL o FMX (los principales entornos de trabajo visuales para C ++ Builder y Delphi).

El resultado es que puedo derribar en un solo día una gran aplicación en la que un programador de JavaScript normal a pasado semanas trabajando.

Smart Mobile Studio utiliza el lenguaje Object Pascal, un dialecto que es aproximadamente un 70% compatible con Delphi. Delphi es excepcionalmente adecuado para escribir grandes aplicaciones de manejo de datos. También es apto para sistemas integrados y servicios de sistemas de bajo nivel. En pocas palabras, es mucho más fácil mantener 50.000 líneas de código pascal de objetos, que 500.000 líneas de código JavaScript.

En Amibian.js, tanto la capa de servicio como la aplicación cliente visual HTML5, están escritas totalmente usando Smart Mobile Studio. Esto me proporciona, como desarrollador principal de ambos sistemas, una gran ventaja: ¿quién lo conoce mejor que el propio diseñador, verdad? Además, puedo escribir código que verdaderamente sea OOP (clases, herencia, interfaces, métodos virtuales y abstractos, clases parciales, etc.), porque nuestro compilador crea algo llamado VMT (tabla de métodos virtuales) en JavaScript.

El JavaScript tradicional no tiene OOP, tiene algo llamado prototipos. Con Smart Pascal puedo importar código de la comunidad de Object Pascal,

componentes y librerías escritas en Delphi o FreePascal, que varían en cientos de miles. Delphi solo tiene una enorme librería de códigos para elegir. Ha sido un conjunto de herramientas muy popular por el tiempo (C es tres años más antiguo que pascal).

Pero, ¿Cómo usaría Amibian.js?

Amibian.js se puede configurar y utilizar de cuatro formas diferentes:

- Como un autentico escritorio, arrancar directamente en Amibian.js en pantalla completa.
- Como un servicio en la nube, al que se accede a través de cualquier navegador moderno.
- Como un front-end de Kiosko o NAS.
- Como un sistema local sobre tu sistema operativo existente. Un script batch lo iniciará y podrás acceder a él introduciendo en tu navegador <https://127.0.0.1:8090>.

De modo que la respuesta breve es sí, lo puedes instalar. Pero es lo mismo que instalar Chrome OS. No es como una aplicación que acaba de instalar en tu maquina Linux, Windows o OSX. El objetivo principal de Amibian.js es tener un sistema agnóstico que no dependa de la plataforma. Algo de lo que no tienes que preocuparte si estás utilizando ARM, x86, PPC o Mips como CPU de preferencia. Los desarrolladores sin duda lo instalarán en las máquinas que tienen. Amibian.js no es intrusivo y no afecta ni toca archivos fuera de su propio ecosistema. Sin embargo, el típico no-programador probablemente lo instalará en una máquina dedicada (o varias) o simplemente la implementará en su NAS doméstico.

La primera forma de disfrutar Amibian.js es instalarlo en un PC o dispositivo ARM. Se ofrecerá una imagen de disco a los fanáticos para que lo puedan poner en funcionamiento lo antes posible. Esta imagen de disco estará basada en una configuración muy liviana de Linux, la suficiente para que todos los controladores funcionen (pero no el escritorio X). Arrancará todos los servicios de node.js y finalmente iniciará una página web a pantalla completa (basada en Chromium) que mostrará el escritorio. Este es el método que la mayoría de los usuarios preferirán para trabajar con Amibian.js.

La segunda forma es usarlo como un servicio en la nube. Instala Amibian.js como he mencionado anteriormente, pero hazlo en Amazon o Azure. De esta manera, puede iniciar sesión en su escritorio utilizando únicamente un navegador web. Esta es una forma muy rentable de disfrutar Amibian.js ya que el alquiler de una instancia virtual es bastante asequible y el almacenamiento del que se dispone es abundante.

La tercera opción es para desarrolladores. Amibian.js es un sistema de escritorio, lo que significa que está diseñado para alojar aplicaciones más complejas. Normalmente tú simplemente insertarías un sitio web externo en un IFrame, pero Amibian.js no es tan primitivo. El alojamiento de aplicaciones externas requiere que escribas un archivo de seguridad expresamente, aunque lo más importante: la aplicación debe interactuar con el escritorio a través del puerto de mensajes de la ventana. Este es un objeto especial que se envía a la aplicación para establecer la comunicación entre ambos, la única forma de que la aplicación acceda a cosas como el sistema de archivos y la funcionalidad del servidor, es a través de este puerto de mensajes.

La llamada a las funciones a nivel de “kernel” desde una aplicación alojada se realiza únicamente a través del puerto de mensajes que hemos mencionado anteriormente. Los datos reales del mensaje están en JSON y deben cumplir con la especificación del protocolo del cliente Ragnarok. Esto no es tan difícil como parece, pero Amibian.js se toma muy en serio la seguridad, de modo que las aplicaciones que intenten causar daños se cerrarán rápidamente.

Has mencionado aplicaciones alojadas, ¿te refiere a sitios web?

Tanto sí como no: Amibian.js admite 3 tipos de aplicaciones:

- Aplicaciones comunes basadas en HTML5/JS, o “sitios web”, como muchos las llamarían. Pero como he dicho anteriormente, tienen que establecer un diálogo con el escritorio antes de que puedan hacer algo útil.
- Aplicaciones híbridas donde la mitad se instalan como un servicio de node.js, y la otra actúan como una aplicación HTML5 normal. Este es el mejor modelo de programa, los desarrolladores básicamente escriben

ambos: un servidor y un cliente y luego lo implementan como un paquete único.

- Aplicaciones bytecode compiladas en LDEF, un lenguaje ensamblador inspirado en 68k que esta compilado en JIT por el navegador (comúnmente llamado "asm.js") y se ejecuta extremadamente rápido. La máquina virtual LDEF es un subproyecto en Amibian.js

La última opción, bytecodes, es un poco como Java. Una parte del proyecto Amibian.js es un compilador y un sistema en tiempo de ejecución llamado LDEF.

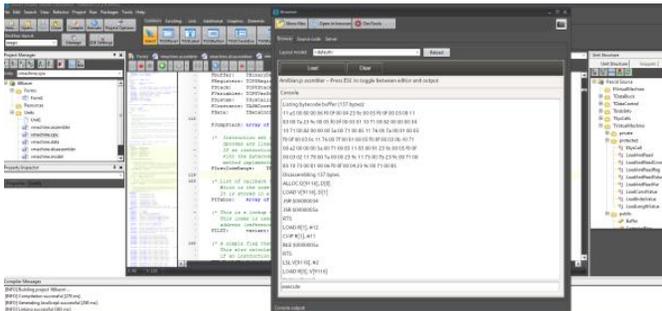


Figura 4 – El ensamblador LDEF de Amibian.js, en este caso lista opcodes y desmonta un método

La primera parte del proyecto Amibian.js es implementar el escritorio y los servicios back-end. La segunda parte del proyecto es crear la primera plataforma de desarrollo basada en la nube del mundo. Un completo clon de Visual Studio, si quieres, que le permita a cualquiera persona escribir aplicaciones nativas, móviles y en la nube directamente desde el navegador.

LDEF admite varios lenguajes, puedes escribir programas en Object Pascal, Basic y C. El dialecto Basic es especialmente divertido para trabajar, ya que es una reimplementación de BlitzBasic, con muchos extras adicionales. Los desarrolladores de Amiga sin duda recordarán BlitzBasic, ya fue utilizado para crear algunos excelentes juegos en los años 80 y 90. Es ideal para juegos y programación multimedia y, sobre todo, es muy fácil de aprender.

Los desarrolladores más avanzados pueden disfrutar de Object Pascal (leer: Delphi) o un subconjunto de C/C++. Ten en cuenta: este IDE está diseñado para aplicaciones a gran escala, no simples fragmentos de código. El objetivo final de Amibian.js es mover todo el ciclo de desarrollo a la nube y alejarlo del escritorio. Con Amibian.js puede escribir una buena "app" en

BlitzBasic, ejecutarla directamente en el navegador o compilarla en el servidor e implementarla en tu teléfono Android como una aplicación real compilada de forma nativa. Cualquier idea sobre un "escritorio simulado para HTML" debe dejarse firmemente a un lado. No estoy jugando con este producto y las apuestas son muy reales.

Pero, ¿por qué no usar ChromeOS?

Hay muchas razones, aunque la más importante es la independencia del conjunto de chips. Chrome OS es un sistema nativo, lo que significa que sus servicios centrales están escritos en C / C++ y están compilados en código máquina. El principio fundamental de Amibian.js es ser 100% independiente de la plataforma, y "no se permite código nativo". Es por esto que todo el back-end y la capa de servicio está orientado a node.js. Esto garantiza el mismo comportamiento independientemente del procesador o sistema host (siendo Linux el host por defecto).

Node.js tiene la ventaja de ser 100% independiente de la plataforma. Descubrirás que node.js funciona para ARM, x86, Mips y PPC. Esto significa que puedes aprovechar cualquier hardware disponible. Incluso puede reutilizar ordenadores antiguos que ya no tiene soporte y usarlos para ejecutar Amibian.js.

Una segunda razón es la siguiente: Chrome OS puede ser gratuito, pero esta tan abierto como Google quiere que lo esté. Chrome OS no es simplemente algo que coges y empiezas a modificar. La dependencia de los lenguajes de programación nativos, las cadenas de herramientas del compilador y un gran conjunto de librerías lo hacen extremadamente específico. Además, tiene totalmente blindadas las partes más interesantes, en concertó, los servicios back-end. Es francamente aburrido y presenta muchas limitaciones para cualquier uso práctico, excepto para Google y sus socios tecnológicos, claro está.

Quería un sistema por el que pudiera moverme, que pudiera ejecutarse en la nube sobre económicos SBCs. Un sistema que permitiese escalar de 10 a 1000 usuarios, un sistema que soporte clústeres y se pueda instalar en múltiples máquinas dentro de un swarm.

Un sistema que cualquier persona con conocimientos de JavaScript pueda usar para crear nuevos y brillantes sistemas, que se pueda expandir fácilmente y servir como base para sustanciosas aplicaciones multimedia.

¿Qué es esto de Amiga, no es una máquina antigua?

En términos de computación sí, pero también lo es Unix. Viejo no significa que automáticamente sea malo, en realidad significa que ha sabido adaptarse y ha sobrevivido a los desafíos más allá de su diseño inicial. Aunque la mayoría de nosotros recordamos a Amiga por sus juegos, yo en particular la recuerdo por su elegante y potente sistema operativo. Un sistema tan flexible que aún continúa utilizándose en todo el mundo, 33 años después de que la máquina llegase al mercado. Esto es todo un logro.



Figura 5: el sistema operativo original de Amiga, ¡no está nada mal para ser un sistema operativo de 33 años! Fue y sigue estando muy por delante de todos los demás. Un testimonio de la creatividad de sus autores.

Amibian.js, tal y como su nombre indica, toma prestados elementos en masa de la arquitectura del sistema operativo Amiga. Sencillamente, porque la forma en que se organiza el sistema operativo Amiga y la forma en que se aborda la informática en el Amiga es brillante. Amiga OS es mucho más intuitivo y fácil de entender que Linux y Windows. Es un sistema que puedes aprender a usar completamente con solo un par de días explorando y sin necesidad de recurrir a manuales.

Pero las similitudes no son solo visuales o a nivel de arquitectura. ¿Recuerdas que comenté que las

aplicaciones alojadas pueden acceder y usar los servicios de Amibian.js? Estos servicios implementan la mayor cantidad posible de funciones originales del Kernel ROM. Naturalmente, no puedo exportarlo todo, porque realmente no es necesario para Amibian.js. Cosas como los controladores del dispositivo no sirven para Amibian.js porque Amibian.js habla con node.js, y node habla con el sistema real, que a su vez gestiona los dispositivos de hardware. Pero la forma en las que crean ventanas, los controles visuales, los eventos de enlace y la creación de aplicaciones modernas, basada en eventos las he conservado lo mejor que he podido.

¿Cómo arranca esta cosa?

Si has configurado una máquina dedicada con Amibian.js, la secuencia de inicio es la misma que la de Linux, excepto que los servicios de node.js se ejecutan como procesos en segundo plano (demonios o servicios que son llamados), el servidor central se inicia y luego se configura una vista HTML5 en pantalla completa que muestra el escritorio.

Pero esto es solo para iniciar el sistema. Tu secuencia de inicio personal se ocupa de tu cuenta, tus preferencias y adaptaciones, que se inician cuando te conectas al sistema.

Cuando inicias sesión en tu cuenta de Amibian.js, no importa si solo lo haces localmente en un PC, un clúster distribuido, o a través del navegador en tu cuenta de la nube, suceden varias cosas:

- El cliente (página web si lo deseas) se conecta al servidor mediante WebSocket.
- El inicio de sesión es validado por el servidor.
- El cliente empieza a cargar los archivos de preferencias a través del sistema de archivos asignado y luego los aplica al escritorio.
- Un archivo con un script de secuencia de inicio se carga desde tu cuenta y luego se ejecuta. El motor de tiempo de ejecución de shell-script está integrado en el cliente, al igual que la ejecución del REXX.
- El script de inicio pondrá en marcha las configuraciones, creará enlaces simbólicos (cesiones), montará dispositivos externos (Dropbox, Google drive, ubicaciones de FTP, etc.)
- Cuando finalice, los programas que haya dentro de la carpeta `~/WbStartup` se iniciaran. Estos pueden ser

tanto visuales como no visuales.

Como puedes ver, Amibian.js no es un modelo o escritorio "falso". Implementa todas las funciones avanzadas que esperas de un escritorio "real". La asignación del sistema de archivos es especialmente avanzada, donde los datos del archivo se cargan a través de controladores especiales; controladores que actúan como un puente entre un servicio de almacenamiento (un disco duro, un recurso compartido de red, un host FTP, Dropbox o lo que sea) y el escritorio. Los desarrolladores pueden agregar tantos controladores como quieran. Si tienen tu propio sistema de almacenamiento casero en tus servidores, pueden implementar un controlador para él. Esto garantiza que Amibian.js pueda acceder a cualquier dispositivo de almacenamiento, siempre que la unidad cumpla con el estándar del controlador. En resumen, puede crear, eliminar, mover y copiar archivos entre dispositivos tal como lo haces en el escritorio de Windows, OSX o Linux. Y las aplicaciones alojadas que se ejecutan dentro de tu propia ventana también pueden solicitar acceso a estos controladores y trabajar con el sistema de archivos (¡y mucho más!).

¿Puede Amibian.js realmente ejecutar programas reales?

Amibian.js cuenta con una versión JavaScript de UAE (Unix Amiga Emulator). Se trata de una bifurcación de SAE (scripted Amiga Emulator) que ha sido altamente optimizada para la web. No solo está escrita en JavaScript sino que funciona de una manera brillante y, por lo tanto, nos permite iniciar un sistema Amiga real. Así que, si tienes algunas imágenes de disquete con juegos que te gusta, estos se ejecutarán perfectamente en el navegador. Incluso yo he llegado a arrancar una imagen de disco duro de 2 gigabytes.

Aunque la emulación de Amiga es solo el comienzo. Más y más emuladores han sido exportados a JavaScript; tienes NES, SNES, N64, PSX I y II, Sega Megadrive e incluso una versión de NEO GEO. ¡Jugar a tus juegos de consola favoritos directamente en el navegador es bastante sencillo!

Pero la parte más interesante será probablemente QEmu. Éste te permite ejecutar instancias x86

directamente en el navegador. Puedes arrancar Windows 7 o Ubuntu dentro de una ventana de Amibian.js si quieres. Quizás no sea práctico llegar a ese punto, pero pone de manifiesto el gran potencial del sistema.

He estado experimentando con un sistema de emulación distribuida, donde la emulación se ejecuta en el servidor, y solo los gráficos y el sonido se transmiten al cliente Amibian.js en tiempo real. Esto ha sido posible durante años a través de Apache Guacamole, pero hacerlo en puro JS encaja más con nuestra filosofía: ¡Sin código nativo!

¿He oído algo de clustering?

¿Recuerdas lo que he escrito sobre los servicios que tiene Amibian.js? ¿Aquellos que actúan casi como librerías en un ordenador físico? Bueno, estos servicios no tienen por que estar en la misma máquina, puedes ubicarlos en máquinas separadas y, por lo tanto, pueden trabajar más rápido.

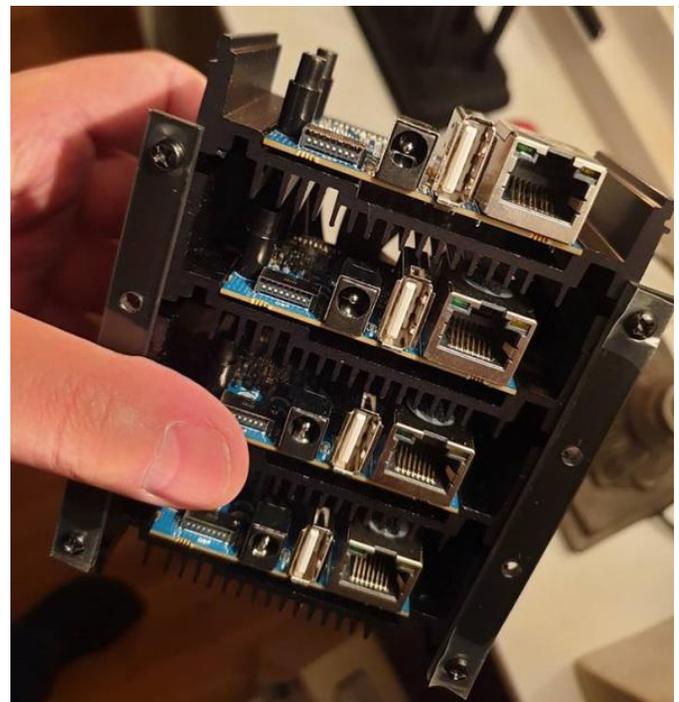


Figura 6 - El clúster oficial Amibian.js, 4 x SBCs ODRROID-XU4 en un micro-rack

Un clúster suele ser varios ordenadores conectados entre sí, con el único propósito de tener más núcleos CPU para dividir el trabajo. Lo bueno de Amibian.js es que no le importa la CPU que hay de fondo. Siempre y cuando nodos.js esté disponible, gustosamente ejecutará cualquier servicio que quieras con el mismo comportamiento y resultado.

El clúster oficial de Amibian.js consta de cinco SBCs ODROID-XU4 (ordenadores de placa reducida). Cuatro de ellas son los llamados ordenadores "secundarios", lo que significa que no tienen un puerto HDMI y están diseñados para iniciar sesión y configurarles el software a través de SSH o herramientas similares. La última máquina es un ODROID-XU4 con un puerto de salida HDMI, que actúa de "maestro" o "central".

La arquitectura es bastante simple: asignamos un SBC completo a un único servicio y permitimos que el servicio se replique para usar todos los núcleos de CPU disponibles; cada SBC tiene ocho núcleos CPU. Con esta arquitectura, la máquina que gestiona los clientes de escritorio no tiene que hacer todo el trabajo. Aceptará tareas del usuario y las aplicaciones alojadas, y luego delegará las tareas entre las otras cuatro máquinas.

Ten en cuenta que el número de SBCs no es fijo. Dependiendo de tu uso, es posible que no necesite más de un único SBC en la configuración de tu hogar, o quizás dos. Yo he empezado con cinco porque quiero que cada componente de la arquitectura tenga la mayor cantidad de potencia de CPU posible. Así que la primera configuración "oficial" de Amibian.js es un monstruo de 40 núcleos que se envía por alrededor de 250\$.

Pero como he dicho, no tienes que comprar todo esto para usar Amibian.js. Puede instalarlo en una PC X86 que tengas de sobra, o conectar en cadena un par de PC antiguos a un switch para lograr el mismo resultado.

¿Por qué SBC secundarios?

Los SBCs secundarios sin pantalla en el diseño inicial tienen una GPU (unidad de procesamiento gráfico) así como capacidad para reproducir audio. Lo que les falta son pines GPIO y 3 puertos USB adicionales. Por lo tanto, cada uno de los nodos de nuestro clúster puede manejar gráficos a gran velocidad, pero a fin de cuentas no es su tarea. Sirven más como módulos informáticos a los que se asignarán tareas para que las lleven a cabo con rapidez, mientras que la máquina principal se ocupa de los usuarios, las sesiones, el tráfico y la seguridad.

El clúster de 40 núcleos que uso tiene más potencia de cálculo que el que tenía el norte de Europa a principios de los 80. Eso es algo que te da para pensar. ¡Y su precio es de menos de 300\$! No sé tú, pero yo siempre he querido tener mi propio superordenador, una plataforma informática distribuida en la que pudiera iniciar sesión y realizar grandes tareas mientras hago cualquier otra cosa. Esto es lo más cercano a un presupuesto limitado, ¡Aún así las limitaciones me parecen muy emocionantes y divertidas!

Parte de la razón por la que opté por un diseño con cluter tiene que ver con el desarrollo futuro. Mientras que UAE.js es brillante para emular Amiga directamente en el navegador, un diseño más interesante es desacoplar la emulación de la salida. En otras palabras, ejecutar la emulación a toda velocidad en el servidor y simplemente transmita la imagen y los sonidos a la pantalla de Amibian.js. Esto nos aseguraría que la emulación de cualquier plataforma se ejecute lo más rápido posible, haciendo uso del multi-procesamiento (lectura: subprocesos múltiples) y de todo el ancho de banda de la red dentro del diseño (el clúster se ejecuta en su propio switch, separado de la world-wide-web exterior).

También estoy muy interesado en la informática distribuida, donde dividimos un programa y ejecutamos cada parte en diferentes núcleos. Este es un tema que quiero investigar más a fondo cuando se complete Amibian.js. Sin duda, requerirá un nuevo diseño del sistema de bytecode LDEF, aunque esto es algo que investigaremos más adelante.

¿Amibian.js reemplazará mi máquina Windows?

Eso depende completamente de para qué uses Windows. El objetivo es crear un sistema autosuficiente. Para la informática retro, la emulación y el desarrollo de increíbles aplicaciones, Amibian.js será increíble. Pero Roma no se construyó en un día, de modo que es aconsejable ser paciente y tratar Amibian.js como lo harías con Chrome OS. Algunas tareas son más adecuadas para sistemas nativos como Linux, aunque cada vez más habrá tareas que se ejecutarán en un escritorio en la nube como Amibian.js.

Una vez que el IDE y los compiladores estén en su lugar después de la fase dos, el sistema se parecerá más a un sistema operativo integrado. Cuando el compilador LDEF y el IDE estén listos, la gente empezará a usarlo en masa y desarrollaran aplicaciones para él. Siempre supone un poco de esfuerzo llegar a ese punto y crear una masa crítica.

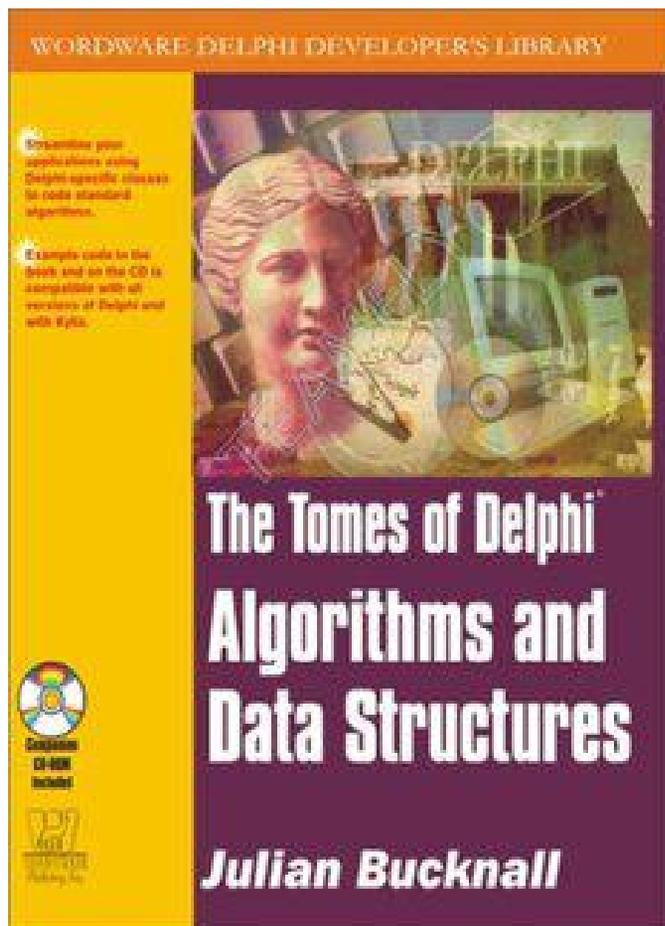


Figura 7: Object Pascal es increíble, aunque los modernos sistemas de desarrollo nativos son bastante exigentes

Mis necesidades personales tienen que ver con el desarrollo. Algunos de los lenguajes que utilizo instalan gigabytes de datos en mi PC y necesitas un completo ordenador portátil para acceder a ellos. Me encanta Amibian.js porque podré trabajar en cualquier parte del mundo, siempre que tenga disponible un navegador y una línea de Internet normal. En mi caso, puedo instalar un compilador nativo en uno de los nodos del clúster y hacer que LDEF emita un código compatible. Voila, puedes crear

aplicaciones listas para ser incluidas en app-store desde un entorno de navegador.

También me encanta poder configurar una plataforma dedicada que ejecute aplicaciones y juegos heredados, y poder escribir nuevas aplicaciones y servicios utilizando lenguajes modernos y comerciales. Si un nodo del clúster se estropea, puedo copiar todo el sistema a un nuevo y económico SBC y continuar mi trabajo. No es necesario comprar un costoso hardware, ni pagar absurdas tarifas de alojamiento y, por último, contamos un sistema que todos podemos configurar y utilizar en una gran cantidad de sistemas. Desde un completo escritorio hasta un super avanzado NAS o router que utiliza Amibian.js para proporcionar a los usuarios una experiencia fantástica.

Y sí, puedo recrear la maravillosa realidad de Amiga OS sin el egoísmo absurdo que domina a los propietarios de Amiga hasta hoy día. Ni siquiera sé por dónde empezar con los actuales titulares de licencias, y estoy harto de que mi drama parezca el único camino razonable para avanzar.

Espero que esto te ayude a aclarar cualquier idea falsa sobre Amibian.js, y que lo encuentres tan interesante como lo hago yo. A medida que más y más servicios pasen a la nube, más relevante se volverá Amibian.js. Es perfecto como base para aplicaciones a gran escala, sistemas integrados y, de hecho, como una plataforma individual que se ejecuta en dispositivos integrados. ¡No puedo esperar a terminar los servicios y poner en práctica este sistema en el rack ODR0ID!

Si encuentras este proyecto interesante, diríjase a mi sitio web de Patreon en <http://www.patreon.com/quartexNow> e involúcrate! Realmente me resultaría muy útil tu apoyo, incluso con solo un "apoyo" de 5\$. Para comentarios, preguntas y sugerencias, visita el artículo original en <https://jonlennartaasenden.wordpress.com/2018/12/05/amibian-js-under-the-hood/>.

Campamento de Programación – Parte 11: Controlar el LED desde tu Smartphone a través del WiFi

© February 1, 2019 By Justin Lee ODRROID-GO, Tutoriales



Este artículo te mostrará cómo montar un servidor web a modo de punto de acceso WiFi, el cual te permitirá controlar un LED desde tu navegador web de forma remota.

Antes de empezar, hay dos cosas importantes que deberías leer primero:

- Consulta los documentos oficiales de Arduino. Estos incluyen las funciones más comunes con excelentes instrucciones, disponibles en: <https://www.arduino.cc/reference/en/>
- Consulta la guía de programación oficial ESP32. La mayoría de las funciones específicas de ESP32 están presentes en: <https://esp-idf.readthedocs.io/en/v3.0/>



Figura 1 – Ejemplo de LED controlado por wifi

WiFi en modo AP

ESP32, que es usado en ODRROID-GO, es compatible con WiFi 802.11b/g/n, de modo que podemos programar servicios WiFi con prácticas librerías en Arduino. En esta guía, vamos a utilizar la librería `wifi.h`:

```
#include  
  
const char *apSsid = "ODRROID_GO_AP";
```

```

const char *apPasswd = "12345678";

WiFiServer server(80);

void setup() {
  IPAddress gateway(192, 168, 4, 1);
  IPAddress subnet(255, 255, 255, 0);

  if (WiFi.softAP(apSsid, apPasswd)) {
    server.begin();
  }
}

void loop() {
}

```

Este es el código básico para el WiFi en modo AP. Como ya sabes lo que supone el modo AP, este código hace que ODROID-GO genere una señal WiFi y tú puedas acceder a este AP a través de tu dispositivo con conexión WiFi. Para hacerlo, hemos definido la información AP del SSID y la contraseña, incluida la dirección IP de la puerta de enlace y la máscara de subred. Estas direcciones IP deben crearse como una instancia de la clase IPAddress. De esta forma, con este código, nos encontraremos en el rango de direcciones IP 192.168.4.x, y puedes conectarse al AP accediendo a 192.168.4.1. Este AP se activa con la función WiFi.softAP().

Para ofrecer una página web, define una instancia de WiFiServer como una variable global. Esto empieza cuando el WiFi se activa con éxito en el modo AP. A continuación, añade el código para configurar el LED en estado azul y aparecerán algunos mensajes de depuración que se muestran en el monitor. Esto nos resultará muy útil para ver cómo fluye el código.

```

#include

#define PIN_STATUS_LED 2

const char *apSsid = "ODROID_GO_AP";
const char *apPasswd = "12345678";

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
}

```

```

pinMode(PIN_STATUS_LED, OUTPUT);
digitalWrite(PIN_STATUS_LED, LOW);

IPAddress gateway(192, 168, 4, 1);
IPAddress subnet(255, 255, 255, 0);

if (WiFi.softAP(apSsid, apPasswd)) {
  Serial.println("WiFi AP established.");
  Serial.print("WiFi AP IP: ");
  Serial.println(WiFi.softAPIP());
  Serial.print("AP SSID: ");
  Serial.println(apSsid);
  Serial.print("AP Password: ");
  Serial.println(apPasswd);

  server.begin();
} else {
  Serial.println("WiFi AP establishing failed.");
}

void loop() {
}

```

Finalmente, crea un “cliente escuchador” en la función loop(). Este código de escucha se reproduce en bucle y sólo responderá cuando accede el cliente. No hemos facilitado una descripción de este código web, porque lo importante es que tú puedas responder como un paquete de datos que contenga el contenido deseado:

```

#include

#define PIN_STATUS_LED 2

const char *apSsid = "ODROID_GO_AP";
const char *apPasswd = "12345678";

WiFiServer server(80);

void setup() {
  Serial.begin(115200);

  pinMode(PIN_STATUS_LED, OUTPUT);
  digitalWrite(PIN_STATUS_LED, LOW);

  IPAddress gateway(192, 168, 4, 1);
  IPAddress subnet(255, 255, 255, 0);

  if (WiFi.softAP(apSsid, apPasswd)) {

```




The image shows a terminal window titled "/dev/ttyUSB1" with a "Send" button in the top right corner. The terminal displays the following text:

```
AP Password: 12345678
New Client.
GET / HTTP/1.1
Host: 192.168.4.1
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/w
DNT: 1
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
New Client.
```

At the bottom of the terminal window, there are four controls: a checked checkbox labeled "Autoscroll", a dropdown menu set to "No line ending", a dropdown menu set to "115200 baud", and a button labeled "Clear output".

Figura 5 – Presentación de la información de depuración de la página web

El ejemplo completo está disponible para importarlo haciendo clic en Files → Examples → ODROID-GO → WiFi_AP, luego presiona CTRL-U para compilar/cargar. Esta guía ha sido cogida de la wiki de ODROID que está disponible en https://wiki.odroid.com/odroid_go/arduino/08_wifi_ap.

OGO-FTPD: Un Servidor FTP para el ODROID-GO

© February 1, 2019 By @Paspartout ↗ ODROID-GO, Tutoriales



En este artículo, presentaré un servidor FTP para ODROID-GO. Es una implementación más bien minimalista que actualmente no admite la autenticación y el funcionamiento en modo pasivo. En este caso vamos a intentar transferir y administrar tus archivos ODROID en una red WiFi fiable, para la cual ya funciona bastante bien.

La razón por la he desarrollado esta aplicación ha sido que recientemente me hice con un ODROID-GO, y no lograba encontrar mi lector de tarjetas SD. Aunque podía flashear el ODROID-GO utilizando el cable USB, pensé que sería una buena idea implementar un servidor FTP para el ODROID-GO. Fue una experiencia muy enriquecedora, aunque me llevo bastante más tiempo que comprar un nuevo lector de tarjetas, lo cual hice más tarde. Una vez que empecé a trabajar en el servidor, quise terminar una versión que fuera aprovechable. Más tarde, leí un post en el foro de sugerencias de proyectos (<https://goo.gl/6wzxFv>) que a alguien realmente le

gustaría usar el odroid como un servidor FTP. Así que espero que éste le sea de ayuda.

La velocidad máxima que alcance fue de alrededor de unos 500 KB/s, pero no estoy seguro de si la tarjeta SD o la conexión WiFi hacían de cuello de botella. Las pruebas se llevaron a cabo en Linux utilizando FileZilla (<https://filezilla-project.org>) como cliente. Me encantaría escuchar tus impresiones e informes de errores al respecto.

Puede descargar la aplicación en <https://github.com/Paspartout/ogo-ftpd/releases>, y el código fuente está disponible en <https://github.com/Paspartout/ogo-ftpd>. Es un problema conocido que a veces tengo que reiniciar la aplicación para que se conecte a mi WiFi.

Para configurar el punto de acceso WiFi que debe usar el ODROID, tienes colocar un archivo llamado wifi.json en la carpeta raíz de su tarjeta SD. El contenido debe seguir este patrón:

```
{"networks": [{"ssid": "YOUR_SSID", "password": "YOUR_PASSWOD", "authmode": "YOUR_AUTHMODE"}]}
```

YOUR_AUTHMODE puede ser uno de los siguientes:

- open
- wep
- wpa-psk
- wpa2-psk
- wpa/wpa2-psk

También puedes añadir varias redes agregándolas a la matriz json. Puedo añadir la capacidad de configurar el wifi mediante un teclado en pantalla como el `odroid-go-launcher` (<https://github.com/jkent/odroid-go-launcher>).

Descubrí que el `odroid-go-launcher` es un proyecto

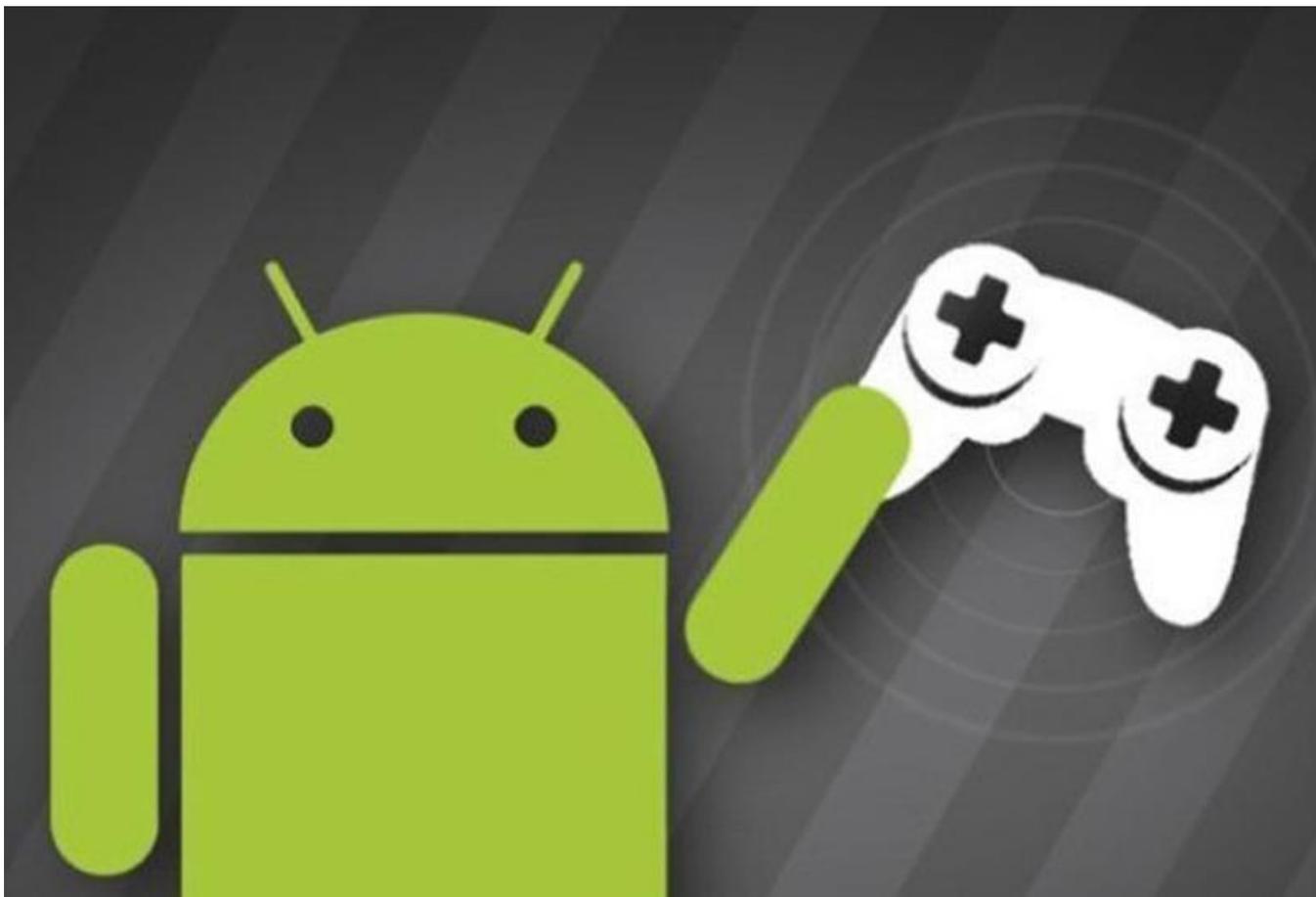
interesante, especialmente por la posibilidad de instalar y usar múltiples aplicaciones sin tener que volver a editarlas todo el tiempo. El servidor usa un código de Jeff Kent, quien creó el lanzador, así que le doy las gracias por ello.

Para transferir de forma segura los archivos a través de Internet, se necesitarán más cosas como el cifrado. El modo pasivo y la autenticación no deberían ser difíciles de implementar, y tal vez incluso FTPS (<https://en.wikipedia.org/wiki/FTPS>) podría ser fácil de añadir, puesto que `esp-idf` ya proporciona una implementación TLS. Las peticiones son bienvenidas.

Para comentarios, preguntas y sugerencias, visita el post original en <https://forum.odroid.com/viewtopic.php?f=158&t=33275>.

Juegos Android

February 1, 2019 By Bruno Doiche Android, Juegos



Hola mis queridos lectores, hace mucho que no nos vemos, ¿verdad? Bueno, eso es así si has pasado por alto nuestra última edición en la que justamente hice una retrospectiva de nuestros 5 años con ODROID Magazine, o como lo llama con cariño: ODROIDMAG. En ese tipo de artículos, solemos bromear hablamos de juegos, y eso es cierto. Así que Rob me preguntó en seguida si tenía vicheados algunos juegos que acabase de jugar en mi ODROID con Android y ¡por suerte contaba con algunos! ¿Quién podría imaginarse que juego en mi tiempo libre?

Así que, sin más rodemos, veamos qué os puedo mostrar:

Sonic Dash

Download Mi hija continúa jugando sin parar con su teléfono móvil a los juegos similares a Temple Run, y siendo un tipo de persona (un tanto celosa), busqué algo parecido, ¿quién se habría preguntado si Sega haría un “runner interminable” con Sonic?

Personalmente estoy enganchado a este tipo de juegos desde el clásico de flash player: canabalt. Son fáciles de jugar y difíciles de dominar. Si de alguna vez has pasado por alto la diversión que supone jugar a un “runner interminable”, y por casualidad jugaste a un juego de Sonic durante tus años de juventud (o aún está viviendo tus años de juventud), descárgate este, seguramente lo adorarás.



Pictura 1 - Sonic Dash

CrossFire: Legends

Download Tengo a Justin Lee a quien culpar por jugar a este juego, el año pasado hizo un artículo sobre cómo jugar a PUBG en el XU4 y mientras buscaba la página de los Juegos de Tencent, me topé con éste. Con un cambio inesperado de modo PVE, es un compañero perfecto para esos momentos en los que aún quieres jugar a un shooter moderno y perfeccionar tus habilidades de PUBG aunque tiene un modo más introspectivo. Por lo general, me pongo de fondo una buena lista de reproducción de synthwave y suelo chillar durante un buen domingo por la tarde.



Picture 2 - Crossfire: Legends

Lamplight

Download Tras dos juegos alocados, te presento Lamplight, un juego basado en turnos. De hecho, es un juego que se desarrolló durante un game jam y es bastante llamativo. Lo mejor de estos juegos independientes es que obtienes una experiencia de juego pura y simple, y por las prisas en el desarrollo consigues un montón de cosas que se pueden interpretar como bonos. Tales como: juego totalmente desbloqueado, sin anuncios, y no se necesita internet. Así que adelante, explora las habitaciones oscuras y derrota a los enemigos usando las habilidades de tu lámpara mágica. Elige tu camino a través de las profundidades y encuentra enemigos únicos en entornos muy variados. Encuentra artefactos que te ayudarán a derrotar a los jefes para robar sus poderes y mejorar tu lámpara. Toma decisiones difíciles mientras te enfrentas a enemigos cada vez más numerosos. ¿Podrás resistir hasta el final?

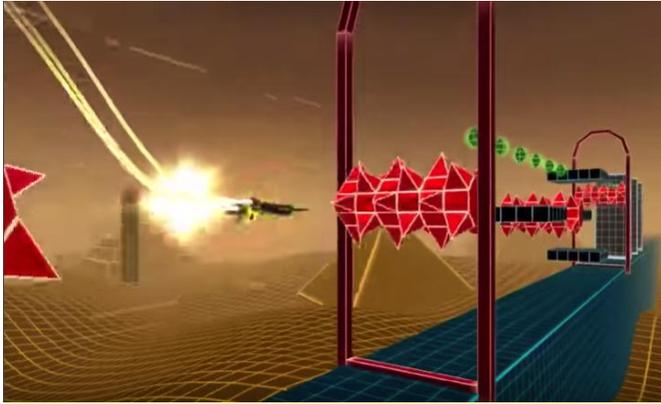


Pictura 3 - Lamplight

OutRush

Download ¿Espera un segundo? ¿Leí en la parte CrossFire: Legends de este artículo sobre chillaxing (chill + relaxing) al sonar canciones geniales de Synthwave? Bueno, no hay nada más justo que

indicar OutRush, que junto con las bandas sonoras de synthwave, viene con el paquete completo de imágenes nostálgicas de los 80 que son tan atractivas. Me recuerda a una mezcla de REZ en mi vieja PS2 y una perspectiva de Neo Geo.

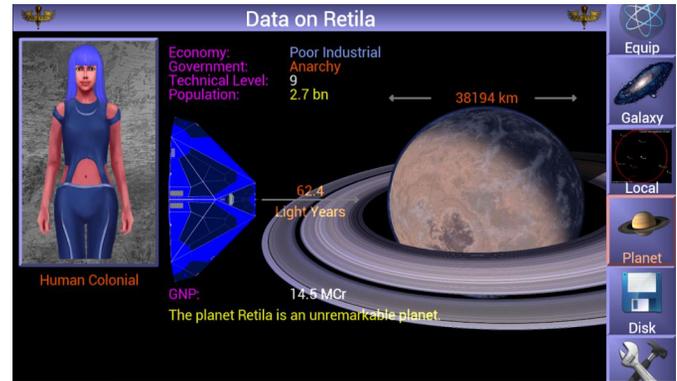


Pictura 4 - OutRush

Alite

Download Un clon muy competente del clásico juego Elite, no te vas a equivocar con este juego. Si no tienes Elite: Dangerous instalado en tu PC / Xbox / PS4 pero

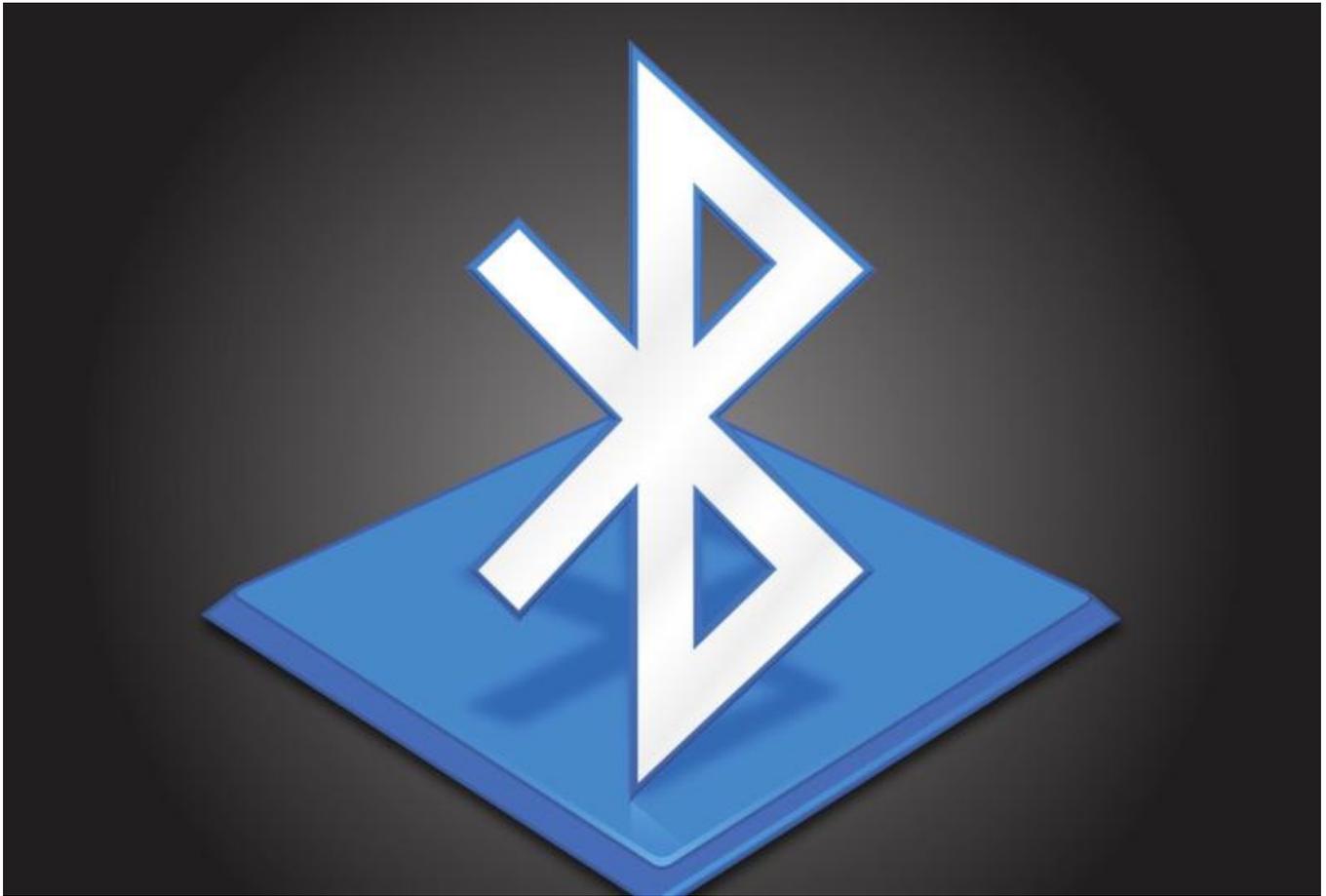
si aun así no quieres perderte una galaxia generada de manera procesal en la que eres un operador espacial que tiene que lidiar con piratas, este juego es para ti. ¡Coge el bloc de notas, empieza a trazar tus mapas de operaciones e ve al vacío, llamadas espaciales!



Pictura 5 - Alite

Campamento de Programación – Parte 12: Comunicación Serial a través de Bluetooth

© February 1, 2019 👤 By Justin Lee ➦ ODRROID-GO, Tutoriales



En este artículo, llevaremos a cabo una conexión inalámbrica a nuestro smartphone utilizando el protocolo RFCOMM de Bluetooth.

(Figura 1 – Enviar mensaje a ODRROID-GO) (Figura 2 – Recibir mensaje desde ODRROID-GO)

Antes de empezar, asegúrate de haber seguido las guías de [Primeros Pasos con Arduino](#) and [Arduino para ODRROID-GO – Hello World](#). Además, siempre puedes consultar la documentación oficial de Arduino. Esta nos proporciona muchas funciones comunes que son muy útiles con excelentes instrucciones

en <https://www.arduino.cc/reference/en/>. La guía de programación oficial de ESP32 también es un buen sitio para obtener información, la mayoría de las funciones específicas de ESP32 las puedes encontrar en <https://esp-idf.readthedocs.io/en/v3.0/>. Por último, el original de esta guía la tienes en la página wiki de ODRROID-GO

en https://wiki.odroid.com/odroid_go/arduino/07_bluetooth_serial.

Bluetooth Serial

El ESP32, que se usa en ODRROID-GO, es compatible con Bluetooth 4.2, por lo que podemos programar las funciones de Bluetooth con las tan útiles librerías de Arduino. En esta guía, vamos a utilizar la librería BluetoothSerial.h:

```
#include "BluetoothSerial.h"

BluetoothSerial serialBT;

void setup() {
  Serial.begin(115200);

  serialBT.begin("ODROID-GO");
  Serial.println("The device started, now you can
  pair it with bluetooth!");
}
```

```

void loop() {
  if (Serial.available()) {
    serialBT.write(Serial.read());
  }
  if (serialBT.available()) {
    Serial.write(serialBT.read());
  }
  delay(20);
}

```

Define una instancia para BluetoothSerial como una variable global llamada serialBT. Para configurarla, usamos la función begin() que tiene un parámetro que indica el nombre del dispositivo Bluetooth para los escáner Bluetooth y lo llamamos ODROID-GO. En la función loop(), si hay un mensaje del monitor Serie disponible, envía ese mensaje al dispositivo conectado a través del Bluetooth Serial. Por otro lado, si hay un mensaje del Bluetooth serie disponible, lo envía al host.

Monitor Serial

Puedes usar un monitor serie para ver los mensajes de depuración desde el puerto serie, que se pueden asignar con el menú Tools → Serial Monitor. Otra posibilidad es presionar CTRL-SHIFT-M para abrirlo rápidamente. Para mostrar el mensaje correctamente, debe fijar el ancho de banda en 115200 baudios. Esta podría ser una herramienta de depuración muy útil. En esta guía, estamos usando esta herramienta para comunicarnos con el dispositivo conectado.



Figura 3- El Monitor Serial

Conectar y comunicar

Probamos con la aplicación Serial Bluetooth Terminal de Android Playstore, en un Galaxy Note 8. Ésta encuentra un dispositivo Bluetooth llamado ODROID-GO y puede conectarse a este.

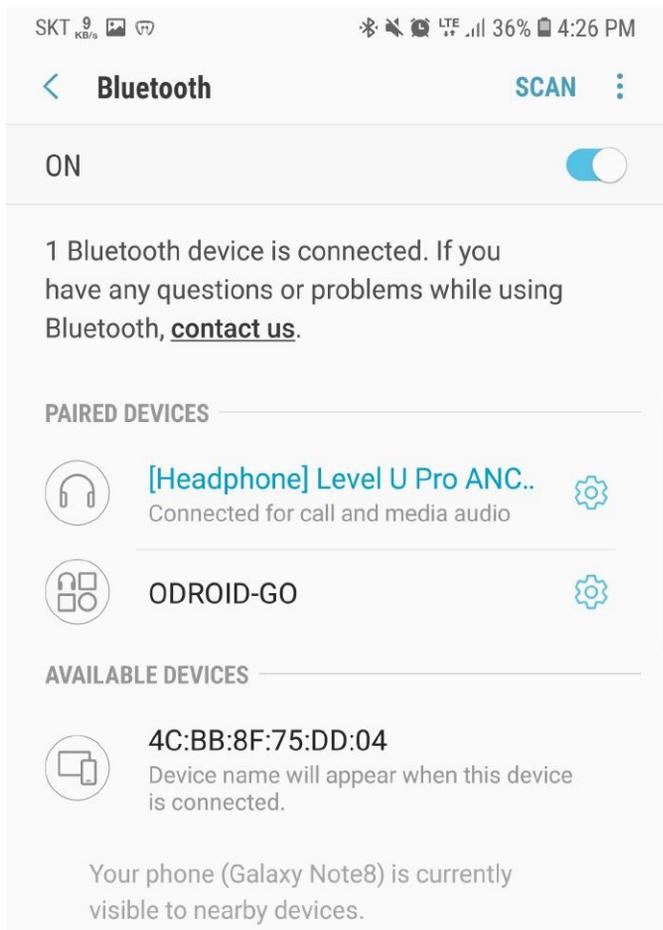


Figura 4 - Emparejamiento Bluetooth con un teléfono inteligente

Y en la aplicación, seleccionamos ODROID-GO en la pestaña de Dispositivos.

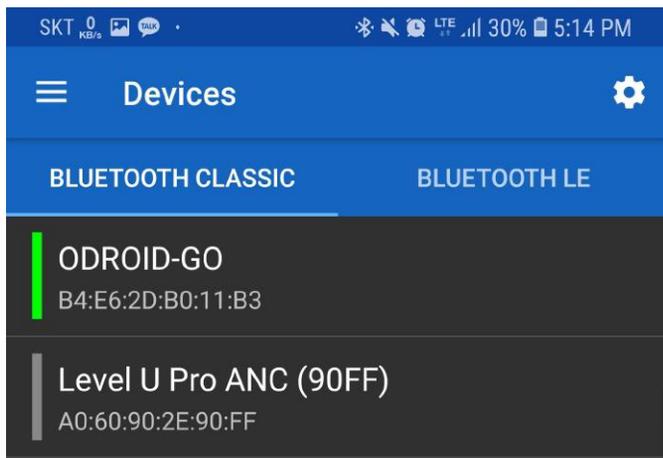


Figura 5 - Emparejamiento de la aplicación Bluetooth Serial

Presiona el botón de conexión de la parte superior de la pantalla en la pestaña Terminal, de esta forma podrán comunicarse entre sí.

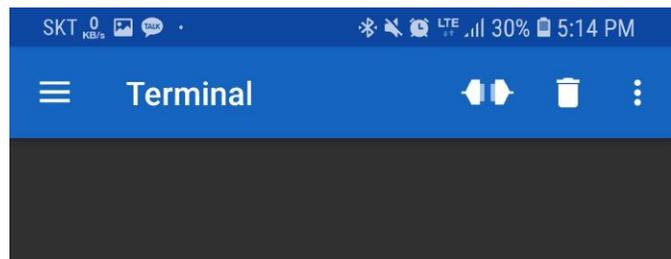


Figura 6 - Aplicación Serial tras el emparejamiento

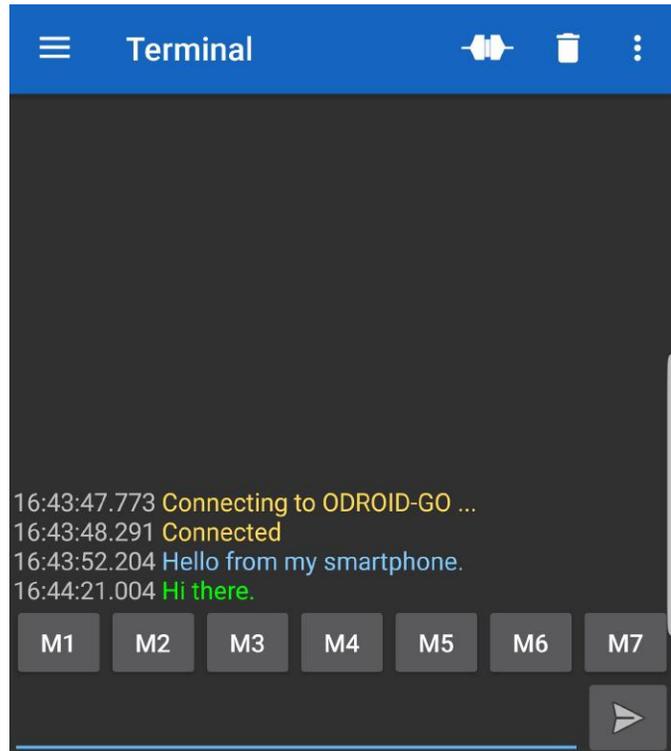


Figura 7 - Aplicación Serial que muestra el texto hacia y desde el ODROID-GO

Por supuesto, también puedes hacer esto utilizando iOS o un PC/Portátil, si disponen de función Bluetooth.

Un ejemplo completo

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

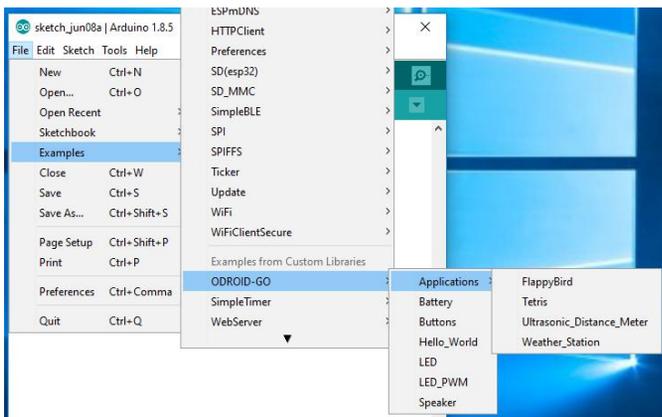
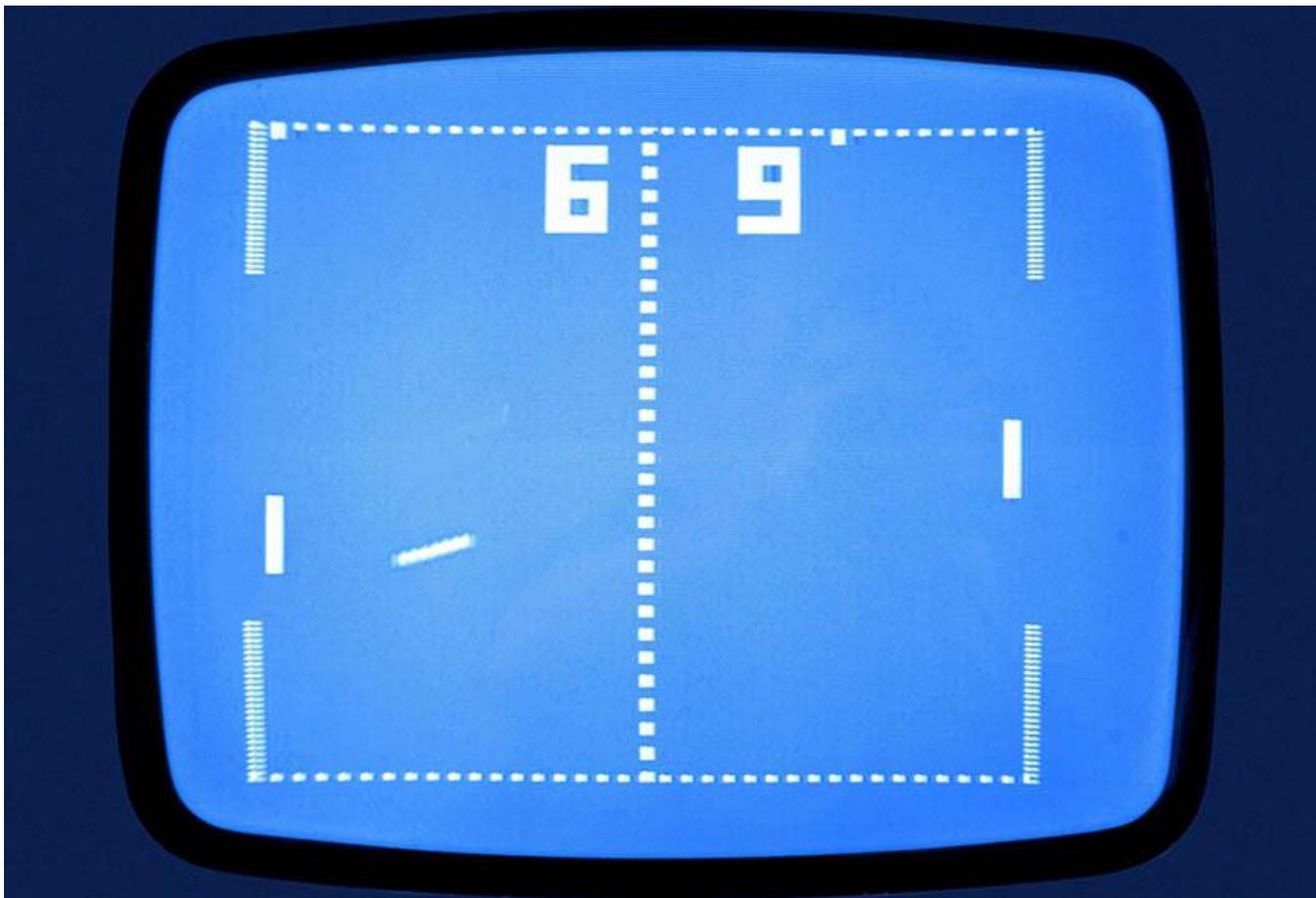


Figura 8 – Cargando el ejemplo de Bluetooth

Pack de Juegos ODROID-GO: Pack de Descarga de Aplicaciones y Juegos de Terceros

© February 1, 2019 By @jutleys Juegos, ODROID-GO, Tutoriales



Este artículo contiene todas las aplicaciones y juegos de terceros que se han ido lanzando hasta el momento para el ODROID-GO por diferentes colaboradores. Para mantenerte al día de los últimos lanzamientos, consulta el artículo original en <https://forum.odroid.com/viewtopic.php?f=159&t=31716>. ¡Muchas Gracias a todos los desarrolladores y colaboradores por las aplicaciones y los juegos!

Instrucciones

Copia el contenido del pack en el directorio raíz de tu tarjeta SD. En cada carpeta encontraras las correspondientes instrucciones. Los archivos básicos están disponibles en https://drive.google.com/file/d/11zbZmq3gUEuFcx1BODcXB_lardaje9a4/view?usp=sharing, y los juegos y aplicaciones de terceros están disponibles en <https://drive.google.com/file/d/1kDO1lmb9x0S0T>

[22JeTcHCl0Lt2s_0gWC/view?usp=sharing](https://github.com/chrisdiana/awesome-odroid-go) (actualizado a 16 de enero de 2019). Para más juegos y recursos, visita <https://github.com/chrisdiana/awesome-odroid-go>, que esta mantenido por @inflam52.

- Copia la carpeta "Romart" en la raíz de tu tarjeta SD
- Coloca los archivos .fw en la carpeta ODROID/firmware
- Apaga el ODROID-GO, mantén presionado el botón B, luego enciéndelo y espera, deja presionado B hasta que aparezca el menú



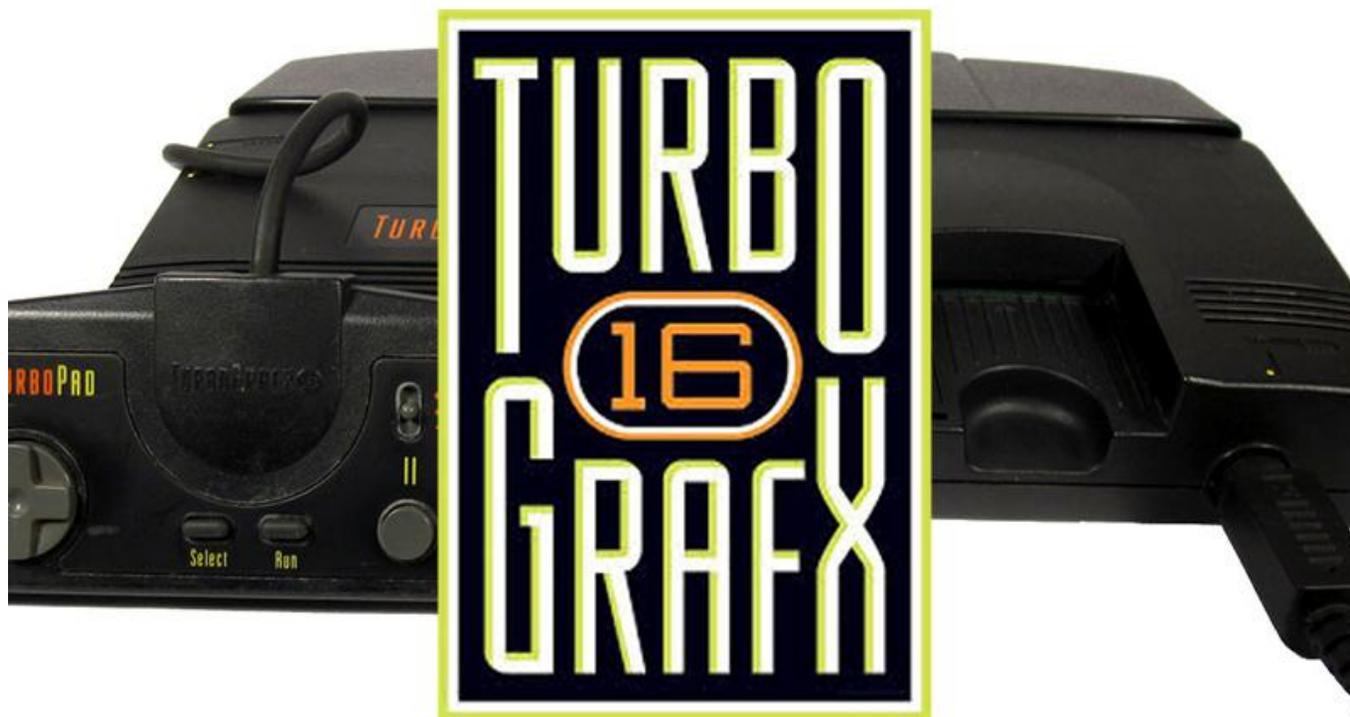
Figura 1 – Coloca los archivos .fw en ODROID/firmware, luego apaga, presiona B, enciende y deja presionando B hasta que aparezca el menú

Lista de Juegos

- Go-Play.fw latest release:
<https://github.com/OtherCrashOverride/go-play/releases>
- Keyboard (turns the go into a BT remote):
<https://github.com/OtherCrashOverride/bt-keyboard-go/releases>
- Wolfenstein 3D Full v1.4 GT and Spear of Destiny:
<https://github.com/jkirsons/wolf4sdl/tree/master/release>
- Doom.fw release: <https://github.com/mad-ady/doom-odroid-go/releases/tag/20180816>
- Doom Latest release with sound:
<https://github.com/mad-ady/doom-ng-odroid-go/releases/tag/20181213>
- Duke Nukem 3D:
<https://github.com/jkirsons/Duke3D/tree/master/release>
- <https://youtu.be/S-DgYw0V4NQ>
- OpenTyrian on ODROID-GO:
<https://github.com/jkirsons/OpenTyrian/tree/master/release>
- Pong Game by metagod194041:
<https://github.com/khuenqdev/goduino/tree/master/pong>
- C64-go Schuemi: <https://github.com/Schuemi/c64-go/blob/master/README.md>
- (New build by Nemo1984
<https://forum.odroid.com/download/file.php?id=8003>)
- C64 frodo-go crashoverride:
<https://github.com/OtherCrashOverride/frodo-go/releases/tag/20181022>
- MicroPython.fw latest release:
https://github.com/OtherCrashOverride/MicroPython_ESP32_psRAM_LoBo-odroid-go
- Stella.fw latest release:
<https://github.com/OtherCrashOverride/stella-odroid-go/releases/tag/20180801>
- Prosystem.fw latest release:
<https://github.com/OtherCrashOverride/prosystem-odroid-go/releases/tag/20180803>
- ZX_Spectrum.fw latest release:
<https://bitbucket.org/DavidKnight247/odroid-go-spectrum-emulator>
- fMSX-go.fw latest release:
<https://github.com/Schuemi/fMSX-go/releases/tag/20180816>
- ODROID-GO compatible M5STACK Library latest release: <https://yadi.sk/d/0wo7ympO3Zc6RT>
- ROM artwork:
https://dn.odroid.com/ODROID_GO/romart-20180810.tgz

Juegos Linux: PC-Engine/TurboGrafx – Parte 2

© February 1, 2019 By Tobias Schaaf ↗ Juegos, ODR0ID-XU4



Realmente he disfrutado probando todos estos juegos de PC-Engine/TurboGrafx CD. Incluso he jugado a algunos de ellos. Estoy deseando probar muchos más. En este artículo, probaré los juegos como lo hice la última vez: jugando con todos y cada uno de ellos durante un rato y luego decidiendo si me han gustado o no. Por ahora, solo me concentraré en los juegos en CD.

Juegos que me han gustado

Doraemon – Nobita no Dorabian Night

Aunque este juego empieza siendo bastante infantil, resultó ser un juego de plataformas muy bueno. Viajas a través del tiempo para salvar a tus amigos y recoger diferentes armas y potenciadores a lo largo de cada nivel, donde puedes seleccionar y cambiar de juego usando el botón de selección para acceder al menú. Los gráficos están bien, aunque los personajes principales son bastante simples. Supongo que esto tiene que ver con el estilo de la serie de televisión en

la cual se basa el juego. En general, disfruté jugando al juego y aunque está completamente en japonés, las cosas se pueden resolver bastante rápido.

A medida que viajas por los diferentes escenarios, te espera una puerta al final de cada nivel que te lleva al siguiente. Dentro de los escenarios puedes recoger artículos de bonificación, que a menudo se encuentran tras las puertas o dentro de las cuevas. Ocasionalmente, encontrarás minis juegos donde puedes conseguir uno de los cuatro elementos disponibles, incluida una vida adicional que es bastante útil.

Double Dragon II – The Revenge

Este juego realmente me recuerda a los viejos tiempos. Me solían gustar los juegos de Double Dragon a los que jugaba todo el tiempo. Realmente me gusta éste en concreto. El juego tiene buenos gráficos, admite el juego cooperativo con dos jugadores, y la música y los efectos de sonido

también están bien. Entre los niveles y antes de que empiece el juego, aparecen buenas secuencias de vídeo y aunque están completamente en japonés, deberías entender lo que sucede. Estas secuencias de vídeo le dan al juego un toque bastante agradable. Tras reproducirse, volverás a dar patadas y puñetazos.



Figura 1 - El bueno y viejo brawler Double Dragon en plena acción



Figura 2: Al final de cada nivel te espera un jefe

Download 2



Figura 3 y 4: Download 2 tiene muchos y diferentes niveles que nunca se parecen



Download 2 es el primer shooter de estos sistemas al que realmente he dedicado tiempo. Los gráficos son agradables con bastantes desplazamientos de paralaje en todos los niveles, e incluso escenas entre niveles. El juego está completamente en japonés, lo cual significa que no entiendo ni una sola palabra de lo que se dice en las secuencias de vídeo, aunque esto no te impide disfrutar del juego en general. El sonido es bueno; me gusta que la música vaya acorde al juego de disparos (aunque no sea innovadora). Cuenta con un total de 4 armas diferentes: un lanzador de guisantes muy rápido que se amplía con un disparo de tres direcciones cuando los actualizas; un arma tipo láser que permite atravesar todos los objetos y que varía de una a cinco líneas láser, dependiendo de tus actualizaciones; Los "misiles" caseros se parecen a bolas, así que no estoy seguro

de si son realmente misiles, que pueden ir en cualquier dirección y atacar automáticamente a todos los enemigos de la pantalla (cuanto más potenciadores tengas, más fuego dispararás a la vez), y por último una descarga eléctrica muy corta pero extremadamente potente, que es la que causa mayor daño, aunque su alcance es tan corto que a menudo te mata si intentas usarla. Sin embargo, puede ser muy útil en los combates contra jefes.

Aparte de las mejoras de las armas, puedes recoger aceleradores que te permiten desplazar tu nave más rápido. Esto también tiene su lado negativo, ya que algunos niveles requieren que te muevas con mucha precisión. Ser demasiado rápido puede hacer que golpees objetos por accidente. También hay satélites que capturan los disparos y te protegen de recibir golpes. Puedes recoger un escudo que te permite absorber diferentes tipos de balas de tus enemigos. El juego es muy bueno, y te recomiendo que lo pruebes si te gustan los shooters.

Dragon Ball Z - Idainaru Son Gokuu Densetsu

Tengo que admitir que soy un gran fan de Dragon Ball Z y probablemente conozca todos los episodios de Dragon Ball Z de memoria. ¡Cuando inicié este juego por primera vez me sorprendió al instante! Este juego tiene algunos de los mejores gráficos que he visto en un juego de Dragon Ball de la época. ¡Las secuencias de video son increíbles! Incluso han llegado a incluir la intro completa de la serie dentro de los gráficos de PC Engine, que está MUY, MUY cerca de la intro original, a pesar de que solo se trata de imágenes fijas y sprites animados. Realmente sientes que los desarrolladores han puesto mucho esfuerzo en este juego.

Algunas escenas están tan bien hechas que creerás que estás viendo la serie de televisión. Por supuesto, todo está en japonés, pero esta vez mi pasión por la propia serie era todo lo que necesitaba para saber exactamente lo que sucedía y que tenía que hacer. Al instante reconoces a todos los personajes, y creo que incluso han llegado a usar todos los actores de doblaje de la serie original japonesa.

La música es la que esperarías que fuera, las mismas canciones que conoces de la serie de televisión. Las

secuencias de video son el aspecto más destacado del juego y se esfuerzan por reproducir las series de TV lo más fielmente posible sin llegar a ser un video en movimiento totalmente.

Me llevo un tiempo hacerme con el sistema de juego. Fue bastante frustrante al principio. Una vez que descubrí lo básico, todo se volvió mucho más sencillo, aunque todavía no domino totalmente el juego. Puesto que me llevo algo de tiempo descubrir lo que encontré, me gustaría compartirlo, ya que creo que es un juego que realmente vale la pena probar.



Figura 5 - Campo de batalla principal en Dragon Ball Z - Idainaru Son Gokuu Densetsu



Figura 6 - Oportunidad de realizar ataques especiales en la segunda fase del combate.

Si echas un vistazo a la primera imagen, verás el típico campo de batalla del juego. Tu personaje puede moverse en tres planos de acá para allá durante el combate. Una pelea se puede prolongar durante

mucho tiempo, aunque al principio no será así, ya que es probable que te derroten bastante rápido y con bastante frecuencia, aunque tú y tu enemigo tenéis mucha salud, la cual tardará un poco en agotarse.

Hablemos de las diferentes barras e iconos:

1. El indicador naranja y azul que parpadea en la parte inferior: ésta es tu barra de ventaja. Se va rellenando a un lado o al otro dependiendo de quién lleva más ventaja en una pelea. Si el indicador se llena por completo a uno lado o al otro, el correspondiente jugador tiene una oportunidad de ataque, como puedes ver en la segunda imagen.
2. Encima del indicador de ventaja tienes tu barra de salud. Por ahora sólo verás bloques amarillos en la imagen. Si te golpean, se volverán rojas y después de rojo pasarán a negras. Es cuando en el juego se gana o se pierde. Como he dicho, esto llevará un tiempo.
3. Por encima de ésta, encontrarás un indicador azul que es tu medidor de potencia. Representa la cantidad de Ki que tienes actualmente. Esta barra tiende a disminuir durante los combates y al realizar ataques especiales, aunque se puede rellenar manteniendo presionados ambos botones del mando.
4. Debajo de tu imagen hay un icono en color. Puede tener 3 colores: verde, rojo y negro, que tienen diferentes propiedades y que puedes cambiar presionando en el mando los botones hacia arriba o hacia abajo. A. El Rojo es tu postura de ataque. Si estás en rojo y atacas al enemigo, es probable que el indicador de ventaja se rellene más a tu favor. B. El verde es tu postura de recarga. Si está en verde y presiona ambos botones para recargar tu Ki, se regenerará MUCHO más rápido que estando el icono en rojo o negro. Si peleas en esta posición, no obtendrás ninguna ventaja sobre tu enemigo. De hecho, como puedes ver en la imagen, más bien será al revés, ya que tener un Ki casi lleno usando el icono en verde en esta situación no ayuda nada. C. Todavía no he averiguado qué significa el negro, aunque estoy seguro de que lo descubriré con el tiempo.

Cuando consigues que el indicador de ventaja se rellene completamente a tu favor, consigues una pantalla similar a la de la segunda imagen, que es tu oportunidad para atacar. Estás en el lado izquierdo de la pantalla y el enemigo está en el lado derecho (no importa quién tenga la ventaja).

Si tienes la ventaja, puedes seleccionar un símbolo japonés que representa un tipo de ataque diferente. Luego, ajusta la barra de la parte inferior que indica la cantidad de Ki que quieres usar para el ataque. La combinación de estos tres elementos creará diferentes tipos de ataques que causarán diversos daños.

Si eres el que está siendo atacado, únicamente puedes ajustar tu barra de Ki, la cual usas para contrarrestar el ataque de tu enemigo. Si usas más energía que tu enemigo y él te ataca, existe la posibilidad de que puedas contrarrestar el ataque o simplemente evitarlo. Un ataque exitoso reduce la barra de salud, decidiendo el resultado de la partida. Es un poco complicado, y en comparación con la serie *Bodukai*, realmente no cuenta como un “combate real”, pero es bastante divertido una vez que le coges el tranquillo.

Dragon Slayer – The Legend of Heroes

Otro de los pocos juegos de rol del sistema que se parecen a los antiguos juegos de *Dragon Quest*. Está completamente en inglés, y viene con interpretaciones de voces un poco cutres. Aunque los gráficos no son tan impactantes como en otros juegos, la jugabilidad y la música lo compensa totalmente. Requiere mucha implicación por tu parte, aunque afortunadamente el juego cuenta con una función de batalla automática que acelera los combates y así poder hacer otras cosas como escribir artículos para *ODROID Magazine*. En algún momento, el juego incluso te dice cuánto tienes que debastar indicándote el nivel que necesitas para que cumplas una determinada misión. Las subidas de niveles también son muy buenas: no solo obtienes mejores estadísticas, sino que también completar tu HP y MP cada vez que subes de nivel. La cantidad de monstruos que te atacan se ajusta a tu fuerza y al tamaño de tu grupo. Esto es bastante bueno, ya que también aumenta la cantidad de Oro y Exp que consigues durante los combates. Mientras tanto, la fuerza del enemigo no aumenta demasiado. Personalmente, he descubierto que tener una buena armadura es casi, si no más importante, que tener una buena arma, ya que reduce en gran medida la cantidad de daño que recibes de los enemigos.

Preferiría golpear a un enemigo tres veces para matarlo y recibir 1 golpe por cada ronda que matar a un enemigo con uno o dos golpes, y recibir 20 golpes a cambio.

Todos los miembros de tu grupo cuentan desde el principio con un hechizo que facilita aún más el proceso de derribo. Tienes un número limitado de espacios para hechizos, de modo que es posible que tengas que intercambiar los hechizos más adelante dentro del juego. Si un enemigo se vuelve más fuerte y tu grupo muere en el campo de batalla, el juego no termina; Puedes reiniciar el combate o regresar a la última ciudad que visitaste. Sugiero reiniciar el combate ya que esto también te permite huir de una pelea y reponer tu salud al estado anterior al combate. Volver a la ciudad te proporcionará un grupo con 1 HP por personaje y no te garantiza que encuentras una posada para pasar toda la noche. He disfrutado jugando al juego hasta ahora y ya he pasado varias horas trabajando duro, obteniendo un gran equipamiento para mi grupo, todo mientras escribía este artículo.

The Dynastic Hero

Se trata de una versión de un juego conocido como "Wonder Boy in Monster World". Es muy parecido a otros juegos que hay para consolas bajo este nombre. De hecho, parece casi idéntico a la versión Genesis/Mega Drive, con algunos cambios menores.

El personaje principal cuenta con un pequeño rediseño y los colores son un poco más oscuros en PC Engine en comparación con la Sega Genesis/Mega Drive. También he notado una ligera diferencia en la relación de aspecto. La PC Engine tiene gráficos ligeramente más grandes que la versión de Sega. También he visto algunos desplazamientos de paralaje en la versión Genesis/Mega Drive, que no parece estar incluidos en la versión PC Engine. Todos estos son solo cambios menores, el juego en sí es bastante similar en ambos sistemas. Se ve muy bien; en mi opinión, uno de los mejores juegos de plataformas de acción para PC Engine.



Figura 7 y 8 - Comparando Dynastic Hero sobre PC Engine (izquierda) y Wonder Boy in Monster World sobre Mega Drive (derecha)



Figura 7 y 8 - Comparando Dynastic Hero sobre PC Engine (izquierda) y Wonder Boy in Monster World sobre Mega Drive (derecha)

La versión de PC Engine viene con una buena intro y bastante música con calidad de CD dentro del juego. Depende de ti elegir uno u otro: son un tanto diferentes, mucha gente se decanta más por los toscos sonidos de Mega Drive en lugar de otras consolas. Simplemente es una recomendación mía. Si deseas comparar los dos juegos, te recomiendo que veas el video de YouTube <https://www.youtube.com/watch?v=R7D2bYgA5IA>.

Exile

Cuando probé por primera vez este RPG de acción no me llegó a impresionar demasiado. Tras una larga y en su mayor parte buena presentación con voces en

inglés, el juego empieza con una vista desde arriba de tu personaje en una ciudad. Los gráficos están lejos de ser impactantes, aunque no son tan malos como en otros juegos.



Figura 9 - Vista panorámica desde arriba del mapa donde hablas con la gente y caminas

Una vez que reuní a mi grupo y fui a un Oasis, me encontré con una trampa que cambió el estilo de juego totalmente, pasando de un juego RPG a un juego de plataformas de acción con gráficos significativamente mejores. Al instante, observé que el juego era mucho mejor que antes. Lamentablemente, esto no duró mucho, ya que pronto se volvió un tanto pobre, los enemigos necesitaban muchos golpes para eliminarlos y se te acercaban muy, muy rápido. Incluso cuando me las ingeniaba para matarlos, había lugares donde los enemigos seguían engendrándose y apenas había terminaba una pelea cuando me veía inverso en otra. Un simple golpe prematuro con mi espada me hacía daño. Tras unos minutos jugando, me mataron y finalicé la partida.



Figura 10: La parte de acción del juego es bastante divertida, una vez que alcanzas un cierto nivel

Aquí fue donde quería dejar de intentarlo en mi primer empeño. Sin embargo, para este análisis, me vi obligado a continuar con el juego. Haciendo uso de las opciones de guardado y carga de partida, logré pasar la primera mazmorra con mucha frustración. Después, volví a bajar, miré un poco a mi alrededor y adquirí un mejor equipo para mi personaje. Esto mejoró un poco las cosas, pero no demasiado. La siguiente sección también le empecé con dificultad, los enemigos necesitaban entre cuatro y ocho golpes para eliminarlos, mientras que yo continuaba cayendo con sólo tres o cuatro golpes. Mientras luchaba y saltaba a lo largo del nivel, de repente adquirí la experiencia suficiente para subir de nivel, y todo cambió drásticamente. Donde al principio necesitaba cuatro a ocho golpes, ahora con tan solo uno o dos golpes eliminaba a un enemigo. También contaba con más HP y podía aguantar muchos más golpes. Como los puntos de experiencia llegaban rápido, pronto subí otro nivel y todo volvió a mejorar, permitiéndome matar a todos los enemigos que encontraba con solo un golpe. Descubrí un curioso lugar donde los enemigos aparecían continuamente y logré subir otro nivel o dos.

Pronto me sentí muy confiado y con ganas de aventura. El segundo jefe con el que me encontré fue fácil comparado con el de la primera mazmorra. El primer jefe que tuve que matar necesitó saltos constantes y golpes precisos. Después de unos 25 golpes o algo así -recuerda que con tres o cuatro

golpes te eliminaban, resultaba bastante difícil- el enemigo por fin caía. El segundo jefe era mucho más fácil. Cuando puedes soportar fácilmente 15 o 20 golpes, luchar contra un jefe que cae tras 10 golpes más o menos, el juego se vuelve mucho más interesante y divertido. Los enemigos posteriores presentaban más de un desafío. Finalmente, llegué a disfrutar bastante del juego y estaba deseando probar el segundo juego de la serie Exile. Dentro del juego, una chica anunciaba la secuela, diciendo que saldría en formato Super CD para PC Engine.

Exile II - Wicked Phenomenon

Aunque se anunciaba como Exile II - The Revenge como un "huevo de Pascua" en el primer juego, Exile II fue finalmente lanzado con un nombre diferente. "¿Tiene algo más de diferente?", quizás te estés preguntando. He investigado sobre ello.

En primer lugar, los gráficos habían mejorado bastante, tanto en la vista vertical del juego como en las escenas de acción de desplazamiento lateral, que ahora ofrecen un poco de paralaje, lo cual es bastante bueno. La jugabilidad es similar al primer juego, aunque presenta algunos cambios.

Viajar entre lugares siempre te colocará en la parte de acción de desplazamiento lateral, donde en el pasado era más bien como un sistema de mazmorras donde el desplazamiento lateral solo tenía lugar cuando entrabas en ciertos edificios o lugares. El juego ahora empieza siendo incluso más difícil que el primero. Dos golpes es todo lo que necesitas para que te maten.

A raíz de mi experiencia con Exile, me tomé mi tiempo para asegurarme de subir de nivel y, por supuesto, las cosas mejoraron aunque no en la misma medida. Podía aguantar más golpes, pero el daño que podía causar todavía seguía siendo bastante bajo en este punto. Ahora puedes ver cuánto HP tiene el enemigo, lo cual te permite hacerte una idea de cuánto tiempo necesitas para golpear a un enemigo antes de que caiga, lo que es especialmente útil cuando luchas contra jefes. Sin embargo, este juego es bastante más difícil, ya que a pesar de que las subidas de nivel son aceptables, no son tan potentes como antes y no

existe una forma fácil de luchar, ya que todos los monstruos son peligrosos de un modo u otro.



Figura 11 y 12 - Gráficos mejorados para Exile 2 tanto en la vista vertical como en la vista de desplazamiento lateral

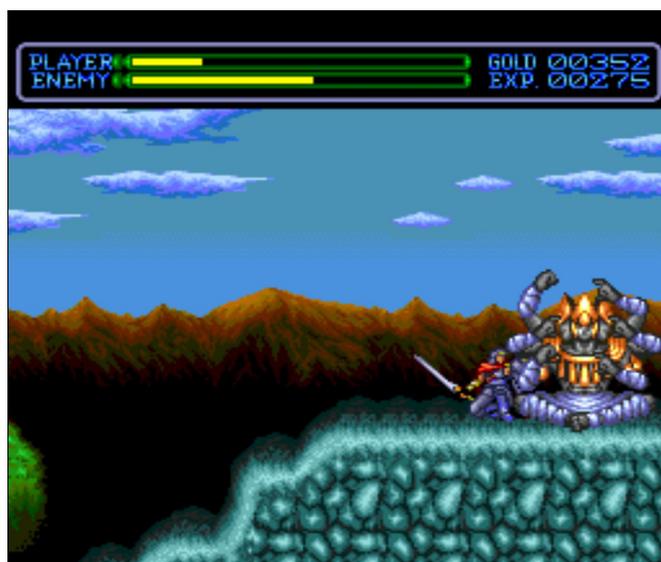


Figura 11 y 12 - Gráficos mejorados para Exile 2 tanto en la vista vertical como en la vista de desplazamiento lateral

Un cambio muy interesante es que ahora puedes controlar los cuatro personajes (que también aparecían en el primer juego) en la parte de acción del desplazamiento lateral del juego. Cada personaje tiene sus propias ventajas y desventajas. Rumi es muy rápido y ágil y tiene un ataque de cierto alcance con el que lanza dagas. Aunque no es muy fuerte. Kindin es un hombre enorme, y lucha con sus puños. Es muy fuerte, aunque sus ataques tienen muy poco alcance, de modo que los enemigos pueden golpearlo con facilidad. Fakhyle usa magia a distancia para atacar. Tanto él como sus ataques son bastante lentos, pero

su objetivo automático permite golpear al enemigo incluso si está encima o debajo. Esto te proporciona algunas opciones tácticas interesantes durante los combates, pero también significa que necesitas cuatro veces los elementos para equipar a tu grupo. Afortunadamente, todos comparten la misma exp, por lo que no es necesario nivelarlos por separado. Aunque encontré Exile II bastante más difícil que Exile, recomiendo probar los dos.

Fantasy Star Soldier / Star Parodier

Este clásico cute-em-up es uno de los muchos shooters para PC Engine. Disfruté bastante de los gráficos tan intensos, con desplazamiento de paralaje y todo tipo de efectos. Viene con tres personajes jugables, uno de los cuales es Bomberman y otro el PC Engine en sí. Los desarrolladores claramente se divirtieron haciendo este juego. Fantasy Star Soldier es la versión americana de Turbo Graphix y Star Parodier es la versión japonesa. Aunque básicamente es el mismo juego, la versión japonesa tiene una intro que fue eliminada de la versión americana, así que, si quieres tener la experiencia completa, te sugiero que pruebes Star Parodier, ya que el juego realmente no necesita ninguna explicación. No es tan difícil como un shooter y si estás buscando algo divertido y amigable para los niños, este es uno de los mejores juegos que existen. Te lo recomiendo.

Faussete Amour

Este juego de plataformas de acción japonés tiene por protagonista a un personaje femenino con una armadura rosa que sale a defender la tierra y a la gente del mal. Para esto tiene algún tipo de mayal que puede usar para atacar en diferentes direcciones, así como una cuerda para llegar a lugares más altos. Cuando saltas, si presionas hacia abajo y el botón de ataque, tu personaje balancea su arma formando un círculo alrededor de ella. También es la única forma de lanzar los tres ataques especiales que puedes recabar con los orbes verdes, rojos y azules lanzados por las hadas, que te proporcionan todos tus artículos dentro del juego.



Figura 13 - Uno de los monstruos jefes del juego. A la mayoría solo se les puede golpear con ataques especiales.

El orbe verde disparará tres burbujas simultáneas en forma de arco, asestando golpes con un amplio ángulo. El orbe rojo disparará tres ráfagas hacia el frente desde donde inicias el ataque. El orbe azul generará tres explosiones de gran amplitud desde el suelo que se extienden hacia arriba cubriendo toda la pantalla. Los gráficos son agradables, la música es buena, los controles son los adecuados y los combates son divertidos. Los desarrolladores fueron un poco pícaros, ya que el personaje principal pierde su armadura cuando es golpeado (como en Ghouls and Ghosts o Ghosts and Goblins), y cuando es golpeada de nuevo, queda completamente desnuda. Entonces ella muere. En general, el juego es divertido y, si eres un fanático de los juegos de plataformas de acción y no suelen ofenderte algunas imágenes, probablemente este sea un buen título para añadir a tu biblioteca.

Fray CD Xak Gaiden



Figura 14: Los adorables gráficos de estilo cómic en Fray CD Xak Gaiden realmente se ajustan a la configuración del juego

Este juego salió en 1994, demasiado tarde en la vida útil del sistema, y esta es probablemente la razón por la que se vea tan bien. Este juego de acción presenta a Fray, que va luchando a través de los niveles con su arma mágica, matando monstruos, recolectando oro para comprar mejores equipos y descubriendo objetos en los cofres del tesoro. Lamentablemente, está completamente en japonés nuevamente, aunque siempre logré encontrar la forma de seguir adelante incluso sin entender lo que se decía. La presentación es agradable y hasta un poco graciosa, ya que el personaje principal es una chica de anime que pone caras divertidas de vez en cuando. Las secuencias de video se muestran bastante bien y, en general, el sonido, los gráficos y la música son excelentes. El juego es bastante divertido. Solo desearía poder entender lo suficiente como para pillar todos los chistes del juego.

Juegos que están bien

Davis Cup Tennis

Este simulador de tenis está bien; Los gráficos son agradables, admiten hasta cuatro jugadores, las voces del juego son excelentes, pero para mi gusto es demasiado difícil. Me frustré tras unos 10 a 15 minutos de juego, ya que solo ganaba una partida cuando el contringante golpeaba la red con cada saque. Supongo que no es el peor juego de deportes al que he jugado, pero personalmente prefiero Virtual Tennis de Dreamcast a esta versión.

Dekoboko Densetsu – Hashiru Wagamanma

Este juego de carreras es bastante interesante y soporta hasta cinco jugadores. Es una especie de juego de coches de choques que es bastante difícil, especialmente si juegas contra el PC. Si presionas seleccionar en la pantalla de inicio, puedes cambiar la dificultad del juego. Te sugiero empezar con el nivel para principiantes, ya que este juego puede llegar a ser bastante complicado.

Golpeando a un enemigo por detrás cogerás un punto de golpe del enemigo. Pero si golpeas a un enemigo en el ángulo equivocado, serás tú el que perderás un punto de golpe. Puedes intentar empujarlos para que se salgan de la pista o hacia los obstáculos que son abundantes. Se parece un poco a los micro machines. Lamentablemente, el enemigo es bastante bueno, así que a los principiantes les resultará un poco difícil acostumbrarse. Creo que este podría ser un juego muy bueno para compartir con tus amigos.

Downtown Nekketsu Monogatari

También conocido como River City Ransom es un juego de lucha muy interesante similar a Double Dragon. A mi gusta especialmente porque comparte el estilo del personaje de mi juego favorito de NES “Nintendo World Cup” (también conocido como Nekketsu Koukou Dodgeball Bu – Soccer Hen). Sólo tiene un grave inconveniente. Está completamente en japonés y por lo tanto, no entiendo lo que hago. Hay cosas que puedo comprar y usar, pero desconozco para qué sirven, así que es un fastidio. Tengo la sensación de que estoy desperdiciando mi dinero que tanto me cuenta ganar en cosas que no necesito. El juego en sí es divertido. Los controles están bien, con un botón para dar patadas, otro para dar puñetazos y ambos botones para saltar. Sin embargo, no es nada fácil y puedes caer más rápido de lo que imaginas si no estás atento. Aún así sigue siendo divertido, solo desearía saber lo que realmente hago.

Dungeon Explorer II

Este juego empieza con una muy buena intro animada y una voz en inglés que te explica la historia de fondo y donde se muestran algunas escenas del primer juego.

Los gráficos de la intro están bastante bien y presenta cierto detalle, aunque dentro del juego no están tan detallados. *Dungeon Explorer II* es un buscador de mazmorras al estilo *Gauntlet*. Lo más interesante es que puedes jugar con hasta cinco jugadores en total. Es un RPG de acción con buena música, gracias al formato CD. Aparte de eso no puedo decir mucho más. Me resultó divertido durante un rato, pero probablemente me divertiría más si lo compartiese con amigos.

Dungeon Master – Theron's Quest

Este juego es el típico buscador de mazmorras en primera persona como *Eye of the Beholder*, *Elvira Mistress of Darkness*, y similares. Por lo general no soy un gran fanático de estos juegos, ya que a menudo requieren retroceder/explorar y un buen cuaderno para anotar todas las pistas y referencias que vas encontrando, puesto que tu inventario tiende a ser bastante pequeño y la información que encuentras no suele usarse hasta transcurrida una o dos horas. No es un mal juego y funciona bien, simplemente no es mi tipo. Tiene muchos sonidos ambientales, pero no hay música.

F1 Circus Special – Pole to Win

Este juego de carreras empieza con una nota interesante y algunas fotografías digitalizadas de la historia de las carreras de Fórmula 1. Es genial verlo, aunque no entiendo nada de lo que se dice, ya que estaba todo en japonés.

El juego en sí no es malo a simple vista, pero tampoco impresiona demasiado. ¡Una cosa que sí es segura es que se trata de un juego muy RÁPIDO! Tal vez esté un poco anticuado o realmente nunca he sido de este tipo de juegos, pero es demasiado rápido para mí. Constantemente me salía de la pista o me golpeaba con cosas. Excesivamente complicado de principio a fin. El juego también tiene un toque de simulación en el que puedes modificar tu coche con diferentes mejoras al igual que en la Fórmula 1 real. Es un juego agradable, pero simplemente no es mi tipo. De hecho, hay una variante 3-D de este juego disponible directamente para ODROID llamado *F1-Spirit* (<https://www.youtube.com/watch?v=M7I4K3-DIW34>)

que al parecer he olvidado publicar y que quizás lo hago en el futuro próximo.

Fiend Hunter

Cuando inicias el juego por primera vez, te encuentras con una intro muy larga y en bucle con muchos desnudos y gráficos detallados. Cuando presionas el botón de inicio y empiezas un nuevo juego, aparece una segunda intro, con gráficos del juego que muestra mucha gente hablando contigo y entre sí, algunos de los cuales se expresan bastante bien.

(Figura 15 – Una de los primeros combates en *Fiend Hunter*. Este tipo puede ser bastante difícil, pero una vez que descubres su patrón, la pelea se vuelve bastante fácil)

Sólo hay un problema con este juego: todo está en japonés, y no tengo ni idea de lo que sucede o de lo que se supone que debo hacer. Tras unos quince minutos jugando, me encontré completamente perdido, así que decidí ir directamente a un extremo de la ciudad para ver qué ocurría. En efecto, abandoné la ciudad y me encontré con un juego de plataformas de acción 2D con muy buen aspecto. No tengo ni idea de si alguna vez debía volver a la ciudad o de lo que todas estas personas me querían decir, pero poco a poco estoy descubriendo el componente de acción del juego. Descubrí que puedes caminar, trepar por salientes, saltar entre huecos y cosas por el estilo, pero no podía hacer mucho más. En un determinado momento llegué a un área y de repente comencé un combate. Ahora contaba con una espada y un pequeño compañero que volaba a mi alrededor todo el tiempo y fui capaz de atacar al enemigo. Estaba seguro de que iba a recibir muchos golpes, así que fui a por él y lo eliminé, perdiendo una cuarta parte de mi HP en el proceso. No está mal, supongo.

Después de matar a un enemigo, éste deja caer un cristal. Este cristal puede usarse para aumentar tus estadísticas, vitalidad, capacidad mental, potencia de tu psico cuchilla, psico flechas y psico chispa. Los dos últimos supongo que aparecerán más adelante en el juego, ya que todavía no los he visto. También puedes mejorar a "Exy the Photon Fiend", tu pequeño compañero que te sigue a todas partes y ataca al

enemigo. Exy también tiene habilidades que puedes usar. Puede volar a tu alrededor y usarlo como fuente de luz en las mazmorras más oscuras. Las peleas no son nada fáciles y admito que utilicé bastante la función de guardardo y carga, aunque cada enemigo tiene un patrón que te permite defenderte de sus ataques y esperar tu turno para contraatacar. Exy es muy útil en este punto, ya que también atacará al enemigo, lo cual a menudo puede salvarte la vida. Probablemente lo hubiera puesto en la sección "Juegos que me han gustado", pero como no entiendo lo que realmente sucede, pienso que me estoy perdiendo algunas cosas importantes.

Final Zone II

Esta es una especie de mezcla de juego de plataformas de correr y abrirte paso a tiros, como los viejos juegos Commando o Ikari Warriors, y un shooter vertical. Las secuencias de vídeo están completamente en inglés y, aunque están bien trazadas, diría que no son muy "buenas". De hecho, son bastante cursis y las interpretaciones de las voces son de lo peor. No tienen mucho sentido, tampoco desarrollan ninguna historia o personaje. Sin embargo, la acción de disparo es bastante divertida y me recuerda a los primeros shooter como Ikari Warriors. Cada uno de los diferentes personajes con los que puedes jugar tiene la misma arma principal, una pistola vulcan, y un arma secundaria distinta. Empiezas como un tipo que lleva un bazoka que hace poco daño. El segundo personaje tiene algún tipo de láser en su lugar. El daño que provoca no es excesivo, pero puede disparar a través de objetos. El tercero en las escenas de tiros horizontales hace volar un helicóptero con "misiles" que causan mucho daño, pero son de corto alcance. Tal vez más adelante aparezcan otros personajes y armas, pero no estoy seguro. Estoy deseando descubrirlo.

Algunos enemigos dejan caer potenciadores, ya sea "H" para salud que la restaura, o una "B" para el arma secundaria. Esto solo restaura tu arma; no te la mejora. Los niveles posteriores te dan la oportunidad de obtener una "P" que aumentará tu salud máxima o una "S" que restaurará tu salud al máximo. El juego solo tiene siete niveles y lo finalizas bastante rápido. En los niveles posteriores puedes elegir qué

personaje quieres llevar a la batalla, lo cual es bastante bueno. Por lo general, el juego es normalito en el mejor de los casos.

Flash Hiders

Este juego es bastante extraño, empieza con una intro que tiene mucha voz narrativa pero poca animación y gráficos. Algunos gráficos son buenos, pero muchos otros son bastante simples. Cuando presionas inicio, aparece una pantalla negra durante varios segundos, que aparentemente es solo una pantalla de carga muy larga. Esto no era nada nuevo que no hubiera encontrado antes en el sistema. El menú te ofrece tres opciones:

1. El modo de batalla cara a cara te permite jugar contra otro jugador o un ordenador, o te permite ver una demo donde el ordenador lucha contra otro ordenador. Antes de empezar, puedes asignar algunos atributos como ataque, defensa y velocidad, y luego comenzar.
2. También hay un modo "Escenario", que es el modo historia del juego, y maldita sea, se pasaron con tanta historia. Antes de cada pelea, tienes una larga escena con personas que hablan entre sí, lo cual parece divertido, pero como está completamente en japonés, no entiendes ni una palabra. Dado que la mayoría son imágenes estáticas con poca o ninguna animación, ni siquiera puedes adivinar de qué se trata. Terminé saltándolas todas. En el modo Escenario, tienes una serie de combates y puedes distribuir tus puntos en ofensiva, defensa y velocidad. Aunque después de un par de peleas, tu personaje subirá de nivel y conseguirás mayores estadísticas. Ya empiezas bastante alto, en el rango de nivel 8 o 9, así que probablemente no verás mucha diferencia.
3. El último modo que tienes es el modo "Progreso". Empiezas con un personaje de nivel 1 que hayas elegido, selecciona las batallas donde quieres combatir, y gana dinero y experiencia. Cuando tienes suficiente dinero, puedes comprar artículos de una tienda para aumentar aún más tu personaje. Este modo lo encontré más interesante y fue al que jugué más tiempo.

Una opción bastante extraña de este juego de lucha es la opción de elegir entre los modos Automático y Manual dentro de los modos Progreso y Escenario. Esto significa que el ordenador toma el control de tu personaje y lucha contra otro oponente del

ordenador. Raro, pero funciona, así que puedes mirar en lugar de luchar contra ti mismo si eso es lo que quieres. Los gráficos y animaciones dentro de los combates son bastante buenos, con un poco de desplazamiento de paralaje en el campo de batalla.

Forgotten Worlds

Se trata un juego de acción de desplazamiento lateral mediocre con muy poca variedad, deficientes voces y mucha redundancia. Tiene una función de disparo automático, de modo que solo necesita "colocar" tu arma en la dirección que deseas disparar. Hay una tienda donde puedes ir para comprar mejoras como salud, armadura o armas, se puede decir que es la parte más emocionante del juego. Los gráficos están bien. No hay desplazamiento de paralaje y en algunas áreas ni siquiera un fondo. Puede resultar divertido durante un rato.

Juegos que no me han gustado

Daisenpuu Custom

Este Shooter en realidad no es uno de mis favoritos. De hecho, creo que es bastante buggy. Hay algunos camiones que dejan caer mejoras de armas o ataques especiales cuando les disparas, pero lamentablemente la mitad del tiempo no es así. El segundo nivel estaba completamente vacío y simplemente volabas. El juego no es muy divertido. No me gustaba el hecho de poder mejorar las armas,

pero no usar armas diferentes. Es un poco tosco y no muy entretenido.

Faceball

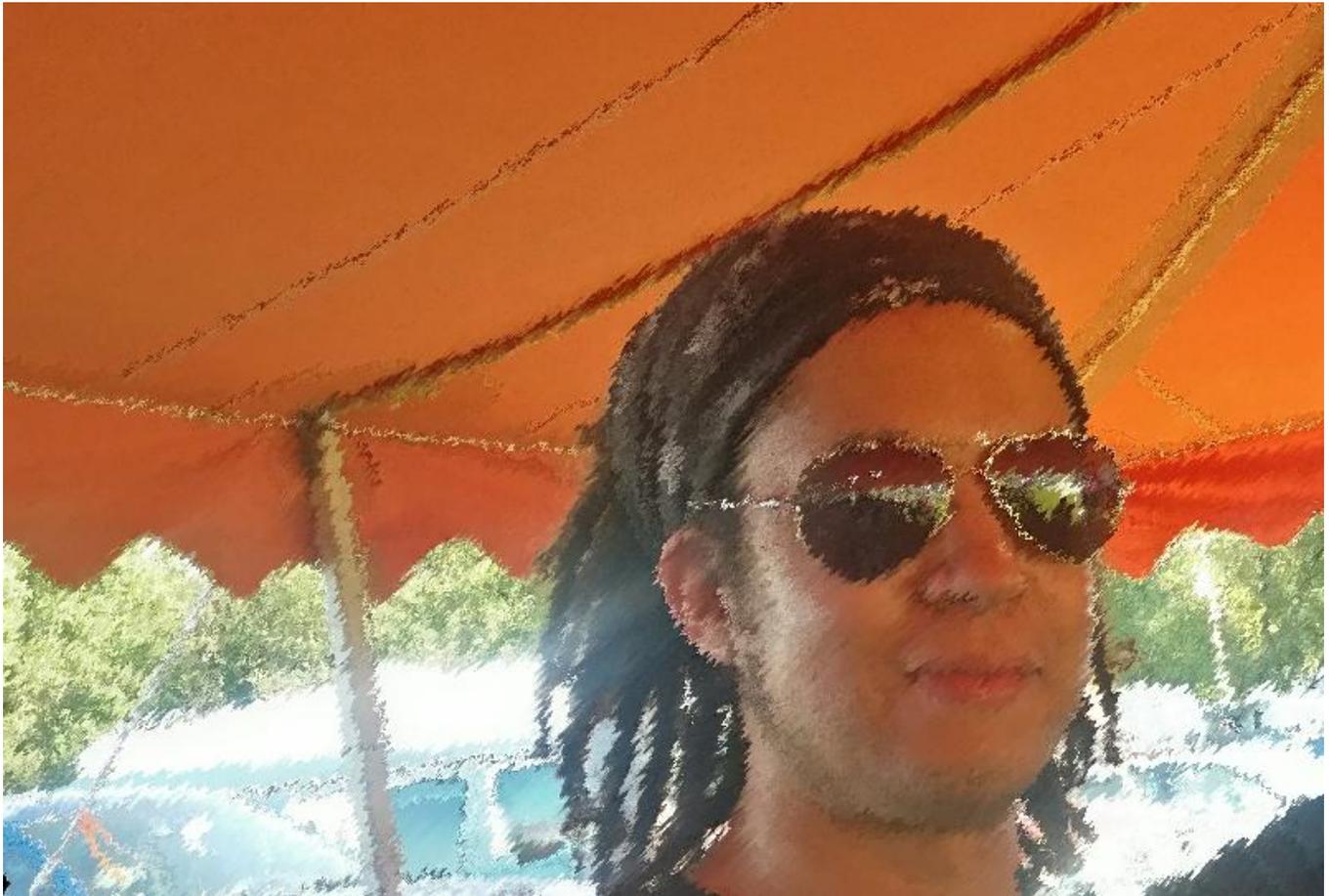
Resulta un tanto complicado describir este juego. Eres una especie de bola con rostro (de ahí el nombre), y te lanzan a un laberinto 3D donde tienes que cazar a otras bolas (modo Batalla) o recolectar huevos (sin saber de dónde vienen) y llevarlos a un muro que parpadea para conseguir puntos (modo Carrera). En mi opinión, ninguno de los dos modos es divertido, y aunque la música es agradable y ver gráficos en 3D en el PC Engine resulta interesante, este juego no me estretuvo lo suficiente como para continuar jugando.

Fighting Street

¿Alguna vez te has preguntado por qué la serie de Street Fighter empezó con "Street Fighter 2"? ¿Qué pasó con "Street Fighter 1"? Realmente existió, lo creas o no, Fighting Street para PC Engine es prácticamente una versión exclusiva de este juego para la consola doméstica. Lamentablemente, el juego en sí nunca fue realmente bueno. Los controles son horribles y las interpretaciones de las voces se copiaron de la versión arcade del juego, lo que era aún peor. La banda sonora fue actualizada con calidad CD, pero esto no remediaba un juego que por lo general no era muy bueno. Si alguna vez quieres saber cómo era el Street Fighter original, puedes probarlo, pero hay juegos de lucha mucho mejores.

Conociendo un ODRROIDian: Chris Lord

February 1, 2019 By Rob Roy Conociendo un ODRROIDian



Por favor h́ablanos un poco sobre ti. Soy un ingeniero de software y ḿusico, la mayor parte de mi vida me dedico a lo primero. Actualmente, estoy trabajando en un sistema de captura de movimiento en tiempo real para dispositivos integrados (<https://github.com/glimpse-project/glimpse>), pero he trabajado en todo tipo de cosas en la última d́cada. Mis áreas de interés han sido, sin ningún orden en particular: sistemas integrados, interfaces de usuario aceleradas por hardware, back-ends de navegador web y, más recientemente, aprendizaje automático. Vivo en el sur de Londres, donde he vivido toda mi vida, aunque estudié en la Universidad de Southampton, donde cursé una licenciatura de informática de primera calidad.



Figura 1 - Encuentro con Ralph Stanley II en el Vine Grove Bluegrass Festival en Kentucky

¿Cómo empezaste con los ordenadores? Desde muy temprana edad, empecé a interesarme por los ordenadores y los aparatos electrónicos, ya que mi

padre era programador informático, y mi madre también pasaba mucho tiempo haciendo contabilidad y escribiendo COBOL para ganarse la vida. Recuerdo tener ordenadores toda mi vida, empezando con un Epson QX-10, antes de pasar a una máquina Amstrad 8086 y luego diversos PCs a lo largo de los años. Siempre he mostrado mucho interés por los ordenadores y mi padre tenía muchas ganas de satisfacer mis curiosidades, así que tuve la ventaja de que empezaron a enseñarme a programar a los 4 años, comenzando con MF-BASIC, antes de continuar con C y adentrarnos en el ensamblador Z80.

Durante mi infancia, mi madre también trabajó en lo que por aquel entonces se llamaba Cable (luego United Artists, luego Telewest, luego BlueYonder y ahora Virgin Media) y esto nos permitió tener Internet a bajo coste, que aproveché al máximo. Esto era bastante inusual por aquel entonces, tener acceso a internet y poder usarlo durante largos períodos de tiempo era algo bastante raro a principios de los 90 en Inglaterra. Aunque pasé mucho tiempo jugando a juegos, también pasé bastante tiempo programándolos, y ampliando mis conocimientos sobre programación en general. No pasó mucho tiempo hasta darme cuenta de que disfrutaba bastante más de todo lo que rodeaba a la programación de los juegos que de la programación de los juegos en sí, así que nunca me sentí atraído por esta trayectoria profesional.

¿Qué te atrajo a la plataforma ODROID? Había estado investigando algunos dispositivos de juegos portátiles basados en la Pi, pero siempre me había molestado el hecho de que usar una Pi significara, por lo general, usar Linux y lidiar con tiempos de arranque prolongados. Que yo sepa, la Pi no admite ningún estado de suspensión, lo cual lo convierte en una opción bastante mala para este caso en concreto. Me encontraba leyendo un artículo en Engadget sobre el kit portátil de juego retro basado en Pi, y alguien en la sección de comentarios mencionó el ODROID-GO. Casi inmediatamente, supe que este dispositivo era prácticamente o estaba muy cerca de lo que estaba buscando; un dispositivo que pueda iniciarse instantáneamente en una aplicación y que tiene el potencial suficiente para emular los sistemas de los

80s. Sin mencionar la insana duración de la batería. Además de disfrutar de todos los kits que se le pueden acoplar, no hace mucho tiempo desarrolle un kit TTL gigatron, un proceso que me pareció muy divertido y que me aportó muchos conocimientos. Estoy ansioso por modificar mi ODROID-GO con una conexión para auriculares, y quién sabe qué otras actualizaciones se le pueda ocurrir a la gente en el futuro.

¿Cómo usas tus ODROIDS? Principalmente utilizo mi ODROID-GO para jugar a juegos de NES durante mis viajes de trabajo o mientras veo la televisión. Especialmente, soy un gran fanático de la versión NES de Tetris, y después de pasar un tiempo mejorando el código de actualización de la pantalla, puedo decir que se trata de un excelente dispositivo para practicar. Probablemente haya pasado más tiempo escribiendo código para este ODROID que realmente jugando en él, aunque es de esperar que la diferencia no sea demasiado notable. Disponer de pines GPIO perfectamente accesibles en el exterior del dispositivo es un buen acierto que sería divertido aprovechar en algún momento, tal vez para sacarle partido a la funcionalidad del cable de conexión de la Gameboy.

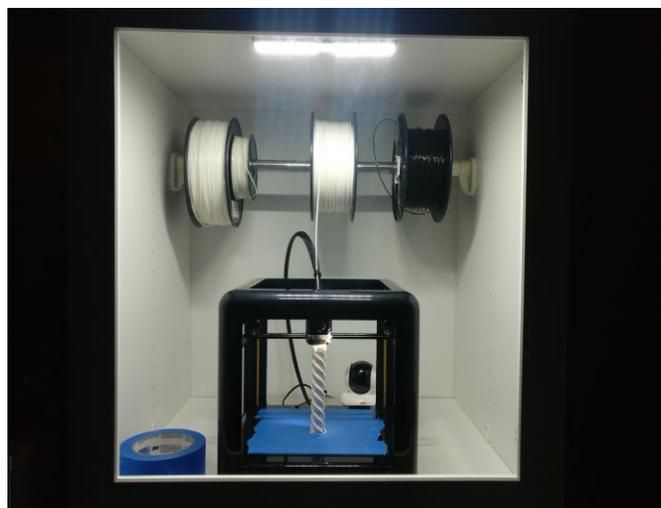


Figura 2: Configuración de la impresión 3D mejorada por IKEA, con un M3D Pro

¿Cuál es tu ODROID favorito y por qué? El ODROID-GO es mi favorito, ya que es el único ODROID que tengo 😊 Sin embargo, mirando el resto de la gama, el H2 parece interesante. Disponer de ese potencial y de la arquitectura x86 supone una gran ventaja para muchos posibles usos. Me interesaría conocer su

consumo de energía en comparación con los modernos SoCs Pi.

¿Qué innovaciones te gustaría ver en futuros productos de Hardkernel? Me encantaría ver más diseños ultraportátiles, como el GO. Los dispositivos de juegos portátiles personalizables me atraen bastante, especialmente por el tipo de perfil ultra reducido del GO. En algunas cuestiones, el GO está un poco limitado, me encantaría ver en un modelo posterior (¿celebrando el undécimo aniversario, tal vez?) SPI para la salida de pantalla. O al menos, usar SPI para la salida de pantalla junto con una pantalla que requiera datos de píxeles de al menos 16 bits. Simplemente no cuenta con suficiente ancho de banda para una salida de 60Hz sin tener que realizar algunos trucos inteligentes (que es en lo que he estado trabajando, con cierto éxito). Si tuviera integrado un controlador LCD que pudiera hacer actualizaciones de 8 bits a base de paletas de colores, o un controlador que pudiera llegar a los 80 MHz completos del bus, no tendría ese problema. Contar con un conjunto de chips que permitiese controlar con mayor precisión la velocidad del bus SPI también ayudaría a resolver este problema, actualmente el ESP32 ofrece 40Mhz o 80Mhz, los informes técnicos muestran que el controlador LCD elegido alcanza los 70Mhz. Realmente se trata de una limitación del chip ESP32, pero creo que podría solucionarse si se pensara en algo.

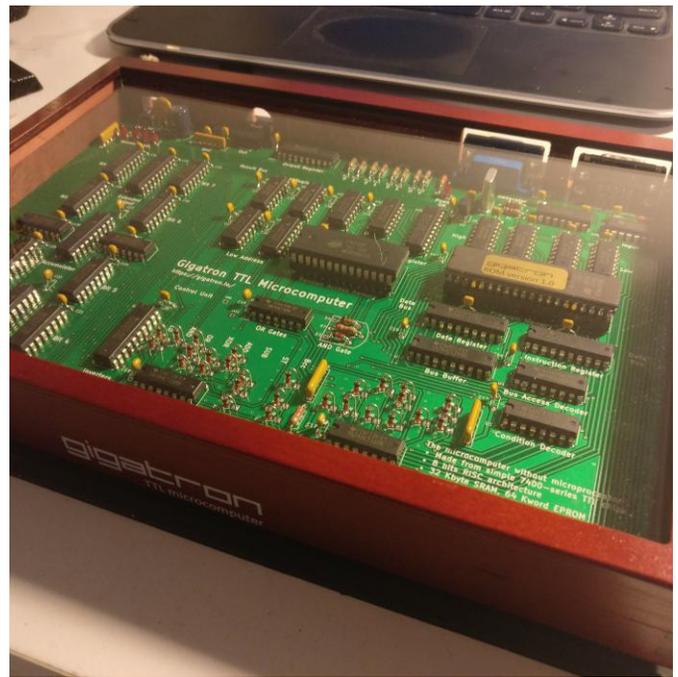


Figura 3 - Una versión terminada del microcomputador Gigatron TTL

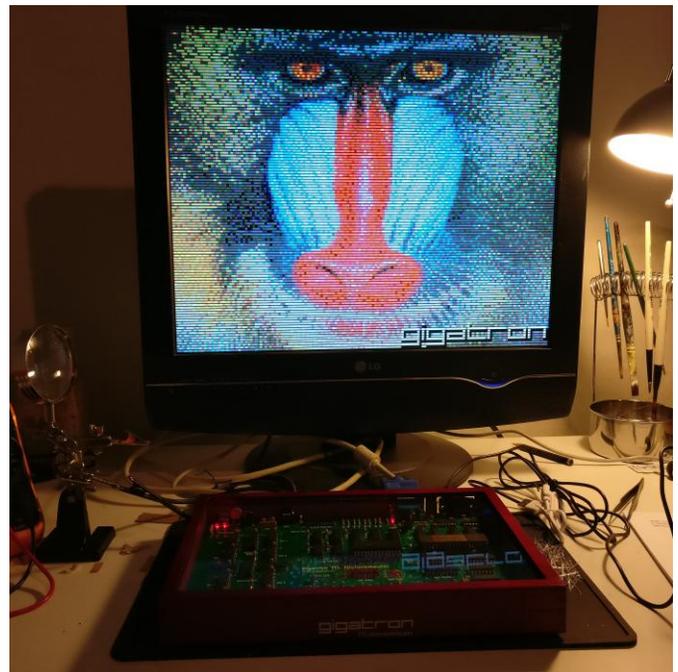


Figura 4 - El microordenador Gigatron TTL mostrando una imagen demo

¿Qué aficiones y hobbies tienes aparte de los ordenadores? Cuando no estoy trabajando con los ordenadores, paso la mayor parte de mi tiempo tocando música. Toco muy bien el banjo y pertenezco en una banda llamada The Vanguard (http://thevanguards.uk/). Cuando no estamos de conciertos, me gusta tocar en sesiones abiertas por Londres, en los últimos años he hecho dos peregrinaciones musicales a los estados del sur. Especialmente me gusta tocar bluegrass, que gusta

bastante en este país. He tocado música durante la mayor parte de mi vida, empezando por el piano cuando tenía 6 años, pasando por el euphonium, trombón, bajo, guitarra eléctrica, batería, banjo, guitarra acústica y contrabajo, con diferentes niveles de dominio. El banjo es mi instrumento principal en este momento, pero todavía disfruto tocando el piano y podría enganchar la guitarra rítmica si me viera en un apuro. Para mí, la música es una forma de evadirme de los ordenadores, que dominan la mayor parte del resto de mi vida. Me gusta mantenerlos separados lo máximo posible.



Figura 5 - La banda de Chris The Vanguard después de grabar un video promocional

¿Qué consejo le darías a alguien que quiera aprender más sobre programación? He tenido el privilegio de estar en situaciones que me han permitido entrevistar a otros programadores durante mi carrera, y algo que creo que realmente separa a unas personas de otras es el conocimiento básico que tienen sobre cómo funcionan verdaderamente los ordenadores. Es demasiado esperar que la gente entienda el funcionamiento de las modernas CPU con sus diversas micro-arquitecturas y extensas y ridículas líneas de datos, aunque indudablemente los

conceptos básicos todavía pueden aplicarse con gran acierto a la programación en la actualidad. Puedes notar la diferencia entre alguien que empezó en JavaScript o Python y nunca fue más allá, y alguien que ha programado sistemas en C o Assembler, incluso para trabajos de alto nivel. Sugeriría a las personas que intentases alcanzar el nivel más bajo con el que se sientan más cómodos. Para mí, ese era la programación Assembler en una calculadora gráfica TI-83, y luego lo mismo para la Gameboy. Pienso que la plataforma Arduino ahora puede enseñar lecciones similares y nunca ha sido tan fácil ni tan barato sumergirse en este mundo.



Figura 6: Logrando de un nuevo record en Tetris NES con el ODROID-GO