CCTV Shinobi • Domótica • Gadget HID • Minería GPU • Juegos

ODROID Magazine

Año cinco
Núm. #52
Abr 2018

ESTUDIO DE Sonido PORTATIL

GRABA MUSICA CON UN ODROID-XU4Q CUANDO QUIERAS Y DONDE QUIERAS

NEXTCLOUD SERVER:
CREA UN ALMACENAMIENTO CONECTADO EN RED (NAS) CON UN ODROID-HC2

ARCADE BOX EN ODROID:
SACALE PARTIDO A LA PLACA ARM CON MEJOR GPU PARA EJECUTAR INCREIBLES JUEGOS

Dispositivo Gadget HID Usando el ODROID-C2

April 1, 2018

La siguiente guía describe cómo configurar el ODROID-C2 como dispositivo gadget HID,

Circuito Cerrado de TV Shinobi (CCTV): Creando un Sistema de Vídeo Vigilancia Usando el ODROID-HC2

April 1, 2018

Es un placer para mí compartir mi experiencia sobre cómo montar un sistema CCTV doméstico utilizando el software Shinobi CCTV y el ODROID-HC2. Con algo de suerte, espero ayudar a alguien que esté pensando hacer los mismo. En mi opinión, los tradicionales sistemas de alarma para el hogar generalmente causan molestias ▶

Estudio de Sonido Portátil: Graba Música con un ODROID-XU4Q en Cualquier Momento y en Cualquier Lugar

April 1, 2018

Hay cierta escasez de sistemas de hardware asequibles en el mercado actual que soporten la variedad de grabaciones en directo que abarca la mayor parte de mi negocio. Tengo que desmantelar mi estudio y reajustar mi sistema de megafonía para llevarlo de viaje. Empecé a preguntarme si podría crear una ▶

Juegos Linux: Juegos Saturn – Parte 3

April 1, 2018

Una vez más, volvemos a la temática de este mes sobre los juegos de Sega Saturn para el ODROID-XU3/XU4. La última vez hablé de un montón de shoot'em ups, pero en este artículo hare una buena combinación de diferentes géneros, aunque sí es verdad que he seleccionado bastantes juegos "mecha" ▶

Arcade Box

April 1, 2018

Hemos creado nuestro propio mando arcade con simples botones GPIO y joysticks, lo hemos llamamos ODROID Arcade Box

Servidor Nextcloud: Creando un Almacenamiento Conectado en Red (NAS) con un ODROID-HC2

April 1, 2018

Esta guía te guiará y ayudará con la configuración de un NAS (Almacenamiento conectado en red) en un ordenador de placa reducida ODROID-HC2. La guía está escrita pensando en aquellas personas que no tienen experiencia con Linux, pero sí una mínima experiencia con el montaje de ordenadores. Y su contenido ▶

Controlar Cualquier Dispositivo Eléctrico con un ODROID-C2: Un Proyecto de Muestra

April 1, 2018

Presentaremos una forma de controlar casi cualquier dispositivo eléctrico con un sólo clic desde cualquier otro dispositivo que tenga acceso a la web.

Buscadores, Mineros y 49 – Parte 2: Minería GPU-CPU Dual en el ODROID-XU4/MC1/HC1/HC2

April 1, 2018

Aquí tienes un breve resumen de los kernels y algoritmos de criptografía que fueron modificados para Odroid y los resultados de las pruebas.

Almacenamiento Seguro: Creando un Sistema de Archivos Cifrado en Linux.

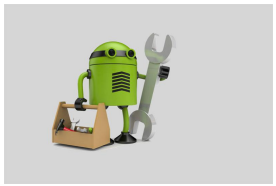
April 1, 2018

Aprende a encriptar su sistema de archivos en un ODROID usando LUKS como configuración de claves.

Tvheadend

April 1, 2018

Las siguientes instrucciones muestran cómo compilar el código de TVH para el ODROID-C2



Desarrollo Android: Así que Quieres Ser un Desarrollador de Aplicaciones

April 1, 2018

Entonces, ¿Quieres ser un desarrollador de aplicaciones? ¡Veamos qué puedo hacer para ayudarte con este tema! Un vistazo a los números atrasados de ODROID Magazine ha puesto de manifiesto que hay un montón de artículos detallados sobre Android en ODROID. Lo que la comunidad ha estado echando en falta es ▶

Configurando tu ODROID: El ODROID-XU4 como un NAS polivalente y versátil

April 1, 2018

Quería que mi ODROID-XU4 hiciera mucho más que lo que puede hacer un simple NAS, y quería seguir usando Ubuntu para beneficiarme de los nuevos paquetes. Todo esto representa una gran oportunidad para adquirir nuevos conocimientos.

Conociendo a un ODROIDian: Ernst Mayer, un Extraordinario Matemático

April 1, 2018

He estado viviendo y trabajando en Silicon Valley durante aproximadamente 20 años, realizando primero trabajos de codificación y algorítmicos para varias empresas tecnológicas y luego para empresas más grandes. La mayor parte de ese trabajo estaba relacionado con software EDA para el diseño de chips. El año pasado, los 4-5 ▶

Dispositivo Gadget HID Usando el ODROID-C2

April 1, 2018 By @Ashren0 Linux, ODROID-C2, Tutoriales

La siguiente guía describe cómo configurar el ODROID-C2 as a HID gadget device, en este caso lo utilizaremos como un teclado o gamepad básico.

Los siguientes pasos pueden adaptarse para cualquier otro dispositivo que

- Soporte el modo USB OTG
- Tenga un kernel Linux superior a la versión 3.19 con el módulo FunctionFS HID compilado, nosotros utilizaremos Linux versión 4.15.

Antes de empezar, debes tener en cuenta que todos los comandos deben ejecutarse como root.

Figura 1 – El ODROID-C2 puede actuar como dispositivo HID usando el modo USB OTG

Preparación

La forma más fácil para ejecutar un kernel reciente, por ahora, es usar la imagen ArchLinuxARM de https://archlinuxarm.org/platforms/armv8/amlogic/odroid-c2. Después de seguir las instrucciones y haber finalizado la instalación por defecto, actualiza el sistema para usar el kernel principal (4.15 en el momento de escribir esta guía):

```
$ sudo pacman -Syu
$ sudo pacman -R uboot-odroid-c2
```

```
$ sudo pacman -S uboot-odroid-c2-mainline
linux-aarch64 dtc
```

Asegúrate de NO reiniciar el dispositivo todavía, si estás arrancando desde la eMMC. La instalación por defecto tiene algunas peculiaridades que hacen que el sistema sea de sólo lectura y esté deshabilitado el módulo OTG. Primero debemos desmontar el Device Tree Blob o DTB, archivo utilizado por el C2 para activar los periféricos integrados en su código fuente desde el DTS

```
$ cd /boot/dtbs/amlogic/
$ sudo cp -p meson-gxbb-odroidc2{,_backup}.dtb
$ sudo dtc -I dtb -O dts meson-gxbb-
odroidc2.dtb > meson-gxbb-odroidc2.dts
```

Una vez hecho esto, edita el código fuente:

```
$ sudo nano meson-gxbb-odroidc2.dts
```

En la sección [mmc@74000], cambia la siguiente línea para volver a activar el acceso de escritura de la eMMC (incluso si no estás usando una eMMC en estos momentos, siempre es bueno cambiarlo):

```
max-frequency = <0xbebc200>;
```

pr:

```
max-frequency = <0x8f0d180>;
```

Y en la sección [usb@c9000000], cambia:

```
dr_mode = "host";
```

por:

```
dr_mode = "peripheral";
```

usb@c9000000 es el periférico OTG que está en modo “host” por defecto. Ten cuidado de no tocar nada relativo al usb@c910000 que es el Hub USB de 4 puertos. Una vez hecho todo esto, podemos volver a compilar el archivo DTB y reiniciar:

```
$ sudo dtc -I dts -O dtb meson-gxbb-
odroidc2.dts > meson-gxbb-odroidc2.dtb
$ sudo reboot
```

Podemos comprobar qué kernel se está ejecutando con el siguiente comando:

```
$ uname -r
```

Debería ser 4.15+ y si todo funcionaba bien, el siguiente comando debería mostrar un enlace simbólico [c9000000.usb]:

```
$ ls /sys/class/udc/
```

Configuración

Utiliza el siguiente script python3 que realiza la configuración y desmonta el dispositivo de forma automática:

```
import sys
import os
import shutil
import pwd
import asyncio
import subprocess
import argparse
import atexit

class HIDReportDescriptorKeyboard(object):
def __len__(self):
return 8

def __bytes__(self):
return bytes([
 0x05, 0x01, # Usage Page (Generic Desktop
Ctrls)
 0x09, 0x06, # Usage (Keyboard)
 0xA1, 0x01, # Collection (Application)
 0x05, 0x07, # Usage Page (Kbrd/Keypad)
 0x19, 0xE0, # Usage Minimum (0xE0)
 0x29, 0xE7, # Usage Maximum (0xE7)
 0x15, 0x00, # Logical Minimum (0)
 0x25, 0x01, # Logical Maximum (1)
 0x75, 0x01, # Report Size (1)
 0x95, 0x08, # Report Count (8)
 0x81, 0x02, # Input (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position)
 0x95, 0x01, # Report Count (1)
 0x75, 0x08, # Report Size (8)
 0x81, 0x03, # Input (Const,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position)
 0x95, 0x05, # Report Count (5)
 0x75, 0x01, # Report Size (1)
 0x05, 0x08, # Usage Page (LEDs)
 0x19, 0x01, # Usage Minimum (Num Lock)
 0x29, 0x05, # Usage Maximum (Kana)
 0x91, 0x02, # Output (Data,Var,Abs)
```

```
 0x95, 0x01, # Report Count (1)
 0x75, 0x03, # Report Size (3)
 0x91, 0x03, # Output (Const,Var,Abs)
 0x95, 0x06, # Report Count (6)
 0x75, 0x08, # Report Size (8)
 0x15, 0x00, # Logical Minimum (0)
 0x25, 0x65, # Logical Maximum (101)
 0x05, 0x07, # Usage Page (Kbrd/Keypad)
 0x19, 0x00, # Usage Minimum (0x00)
 0x29, 0x65, # Usage Maximum (0x65)
 0x81, 0x00, # Input (Data,Array,Abs,No
Wrap,Linear,Preferred State,No Null Position)
 0xC0, # End Collection
])

class HIDReportDescriptorGamepad(object):
def __len__(self):
return 4

def __bytes__(self):
return bytes([
 0x05, 0x01, # USAGE_PAGE (Generic Desktop)
 0x15, 0x00, # LOGICAL_MINIMUM (0)
 0x09, 0x04, # USAGE (Joystick)
 0xa1, 0x01, # COLLECTION (Application)
 0x05, 0x02, # USAGE_PAGE (Simulation
Controls)
 0x09, 0xbb, # USAGE (Throttle)
 0x15, 0x81, # LOGICAL_MINIMUM (-127)
 0x25, 0x7f, # LOGICAL_MAXIMUM (127)
 0x75, 0x08, # REPORT_SIZE (8)
 0x95, 0x01, # REPORT_COUNT (1)
 0x81, 0x02, # INPUT (Data,Var,Abs)
 0x05, 0x01, # USAGE_PAGE (Generic Desktop)
 0x09, 0x01, # USAGE (Pointer)
 0xa1, 0x00, # COLLECTION (Physical)
 0x09, 0x30, # USAGE (X)
 0x09, 0x31, # USAGE (Y)
 0x95, 0x02, # REPORT_COUNT (2)
 0x81, 0x02, # INPUT (Data,Var,Abs)
 0xc0, # END_COLLECTION
 0x09, 0x39, # USAGE (Hat switch)
 0x15, 0x00, # LOGICAL_MINIMUM (0)
 0x25, 0x03, # LOGICAL_MAXIMUM (3)
 0x35, 0x00, # PHYSICAL_MINIMUM (0)
 0x46, 0x0e, 0x01, # PHYSICAL_MAXIMUM (270)
 0x65, 0x14, # UNIT (Eng Rot:Angular Pos)
 0x75, 0x04, # REPORT_SIZE (4)
 0x95, 0x01, # REPORT_COUNT (1)
 0x81, 0x02, # INPUT (Data,Var,Abs)
 0x05, 0x09, # USAGE_PAGE (Button)
 0x19, 0x01, # USAGE_MINIMUM (Button 1)
 0x29, 0x04, # USAGE_MAXIMUM (Button 4)
 0x15, 0x00, # LOGICAL_MINIMUM (0)
 0x25, 0x01, # LOGICAL_MAXIMUM (1)
 0x75, 0x01, # REPORT_SIZE (1)
 0x95, 0x04, # REPORT_COUNT (4)
 0x55, 0x00, # UNIT_EXPONENT (0)
 0x65, 0x00, # UNIT (None)
 0x81, 0x02, # INPUT (Data,Var,Abs)
 0xc0 # END_COLLECTION
])

class HidDaemon(object):
 def __init__(self, vendor_id, product_id,
manufacturer, description, serial_number,
hid_report_class):
 self._descriptor = hid_report_class()
 self._hid_devname = 'odroidc2_hid'
 self._vendor = vendor_id
 self._product = product_id
 self._manufacturer = manufacturer
 self._desc = description
 self._serial = serial_number
 self._libcomposite_already_running =
self.check_libcomposite()
 self._usb_f_hid_already_running =
self.check_usb_f_hid()
 self._loop = asyncio.get_event_loop()
 self._devname = 'hidg0'
 self._devpath = '/dev/%s' % self._devname

def _cleanup(self):
 udc_path =
'/sys/kernel/config/usb_gadget/%s/UDC' %
self._hid_devname
 if os.path.exists(udc_path):
 with open(udc_path, 'w') as fd:
 fd.truncate()
 try:
 shutil.rmtree('/sys/kernel/config/usb_gadget/
%s' % self._hid_devname, ignore_errors=True)
 except:
 pass
 if not self._usb_f_hid_already_running and
self.check_usb_f_hid():
 self.unload_usb_f_hid()
 if not self._libcomposite_already_running and
self.check_libcomposite():
 self.unload_libcomposite()

@staticmethod
def check_libcomposite():
 r = int(subprocess.check_output("lsmod | grep
```

```
'libcomposite' | wc -l", shell=True,
close_fds=True).decode().strip())
 return r != 0

@staticmethod
def load_libcomposite():
 if not HidDaemon.check_libcomposite():
 subprocess.check_call("modprobe
libcomposite", shell=True, close_fds=True)

@staticmethod
def unload_libcomposite():
 if HidDaemon.check_libcomposite():
 subprocess.check_call("rmmod libcomposite",
shell=True, close_fds=True)

@staticmethod
def check_usb_f_hid():
 r = int(
 subprocess.check_output("lsmod | grep
'usb_f_hid' | wc -l", shell=True,
close_fds=True).decode().strip())
 return r != 0

@staticmethod
def load_usb_f_hid():
 if not HidDaemon.check_libcomposite():
 subprocess.check_call("modprobe usb_f_hid",
shell=True, close_fds=True)

@staticmethod
def unload_usb_f_hid():
 if HidDaemon.check_libcomposite():
 subprocess.check_call("rmmod usb_f_hid",
shell=True, close_fds=True)

def _setup(self):
f_dev_name = self._hid_devname
os.makedirs('/sys/kernel/config/usb_gadget/%s/
strings/0x409' % f_dev_name, exist_ok=True)
os.makedirs('/sys/kernel/config/usb_gadget/%s/
configs/c.1/strings/0x409' % f_dev_name,
exist_ok=True)
os.makedirs('/sys/kernel/config/usb_gadget/%s/
functions/hid.usb0' % f_dev_name,
exist_ok=True)
with
open('/sys/kernel/config/usb_gadget/%s/idVendo
r' % f_dev_name, 'w') as fd:
fd.write('0x%04x' % self._vendor)
with
open('/sys/kernel/config/usb_gadget/%s/idProdu
ct' % f_dev_name, 'w') as fd:
fd.write('0x%04x' % self._product)
with
open('/sys/kernel/config/usb_gadget/%s/bcdDevi
ce' % f_dev_name, 'w') as fd:
fd.write('0x0100')
with
open('/sys/kernel/config/usb_gadget/%s/bcdUSB'
% f_dev_name, 'w') as fd:
fd.write('0x0200')

with
open('/sys/kernel/config/usb_gadget/%s/strings
/0x409/serialnumber' % f_dev_name, 'w') as fd:
fd.write(self._serial)
with
open('/sys/kernel/config/usb_gadget/%s/strings
/0x409/manufacturer' % f_dev_name, 'w') as fd:
fd.write(self._manufacturer)
with
open('/sys/kernel/config/usb_gadget/%s/strings
/0x409/product' % f_dev_name, 'w') as fd:
fd.write(self._desc)

with
open('/sys/kernel/config/usb_gadget/%s/configs
/c.1/strings/0x409/configuration' %
f_dev_name, 'w') as fd:
fd.write('Config 1 : %s' % self._desc)
with
open('/sys/kernel/config/usb_gadget/%s/configs
/c.1/MaxPower' % f_dev_name,'w') as fd:
fd.write('250')

with
open('/sys/kernel/config/usb_gadget/%s/functio
ns/hid.usb0/protocol' % f_dev_name, 'w') as
fd:
fd.write('1')
with
open('/sys/kernel/config/usb_gadget/%s/functio
ns/hid.usb0/subclass' % f_dev_name, 'w') as
fd:
fd.write('1')
with
open('/sys/kernel/config/usb_gadget/%s/functio
ns/hid.usb0/report_length' % f_dev_name, 'w')
as fd:
fd.write(str(len(self._descriptor)))
with
open('/sys/kernel/config/usb_gadget/%s/functio
ns/hid.usb0/report_desc' % f_dev_name, 'wb')
```

```
as fd:
fd.write(bytes(self._descriptor))

os.symlink(
'/sys/kernel/config/usb_gadget/%s/functions/hi
d.usb0' % f_dev_name,
'/sys/kernel/config/usb_gadget/%s/configs/c.1/
hid.usb0' % f_dev_name,
target_is_directory=True
)

with
open('/sys/kernel/config/usb_gadget/%s/UDC' %
f_dev_name, 'w') as fd: fd.write('
'.join(os.listdir('/sys/class/udc')))

def run(self):
if not self._libcomposite_already_running:
self.load_libcomposite()
atexit.register(self._cleanup)

# Setup HID gadget (keyboard)
self._setup()

# Use asyncio because we can then do thing on
the side (web ui, polling attached devices
using pyusb ...)
try:
self._loop.run_forever()
except KeyboardInterrupt:
pass

if __name__ == '__main__':
user_root = pwd.getpwuid(0)
user_curr = pwd.getpwuid(os.getuid())
print('Running as <%s>' % user_curr.pw_name)
if os.getuid() != 0:
print('Attempting to run as ')
sys.exit(os.system("/usr/bin/sudo /usr/bin/su
root -c '%s %s'" % (sys.executable, '
'.join(sys.argv))))
parser = argparse.ArgumentParser()
parser.add_argument('hid_type', choices=
['keyboard', 'gamepad'])
args = parser.parse_args()
if args.hid_type == 'keyboard':
print('Emulating: Keyboard')
# Generic keyboard
hid = HidDaemon(0x16c0, 0x0488, 'author',
'ODROID C2 KBD HID', 'fedcba9876543210',
HIDReportDescriptorKeyboard)
hid.run()
elif args.hid_type == 'gamepad':
print('Emulating: Gamepad')
# Teensy FlightSim for the purpose of this
example (and since it's intended for DIY, it
fits ou purpose)
hid = HidDaemon(0x16c0, 0x0488, 'author',
'ODROID C2 GAMEPAD HID', 'fedcba9876543210',
HIDReportDescriptorGamepad)
hid.run()
```

Las clases HIDReportDescriptorKeyboard y HIDReportDescriptorGamepad son donde describimos nuestro dispositivo, su tipo, los botones y la cantidad de ejes. Ten en cuenta que vendorId y productId también son muy importantes puesto que, aunque describas tu dispositivo como un gamepad, si VID/PID son los de un teclado, el sistema operativo probablemente identifique el dispositivo como un teclado.

A continuación, ejecuta el siguiente comando, que requiere privilegios de root, con el fin de crear un dispositivo /dev/hidg0 con el que puedas escribir libremente:

```
$ sudo python3 script.py keyboard
```

o

```
$ sudo python3 script.py gamepad
```

Después podemos probarlo con el argumento del teclado:

```
$ sudo sleep 5 && echo -ne "" > /dev/hidg0 &&
echo -ne "" > /dev/hidg0
```

Este comando escribirá “A” (o “Q” si usas una distribución azerty), tras 5 segundos. Ahora, para probarlo con el argumento del gamepad usa:

```
$ sudo sleep 5 && echo -ne "" > /dev/hidg0
&& echo -ne "" > /dev/hidg0
```

Esto activará el cuarto botón del dispositivo gamepad.

Creando diferentes dispositivos

Los dos ejemplos usan descriptores muy básicos que usan el siguiente formato de bit al escribir en /dev/hidg0:

Teclado (6 rotaciones)

BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8
Modificadores	Reservado	Tecla 1	Tecla 2	Tecla 3	Tecla 4	Tecla 5	Tecla 6

Gamepad

BYTE 1								BYTE 2							
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Acelerador (-127 to 127)								Eje-X (-127 to 127)							

								BYTE 4							
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Eje-Y (-127 to 127)								B4	B3	B2	B1	(1)		(2)	

(1)Cuando es igual a 0b11, los botones HAT están configurados como no activos, que debería ser el valor por defecto. (2)Mascara de bits botones HAT, dado que los bits 5-6 están configurados en 0 (a)0b00 => ARRIBA (b)0b01 => DERECHA (c)0b10 => ABAJO (d)0b11 => IZQUIERDA

Conclusión

Este es un ejemplo muy básico, aunque nos muestra las opciones que tenemos disponibles. El análisis de las especificaciones USB y HID nos da una idea de los muchos posibles usos:

- Simular un dispositivo, como un teclado, gamepad, ratón específicos para fines de desarrollo.
- Hacer que un dispositivo aparezca como otro. Para dispositivos heredados
- Pura depuración/ingeniería inversa USB (proyecto USBProxy en https://github.com/dominicgs/USBProxy).
- Pruebas de penetración.
- Y seguramente muchos otros en los que no he pensado.

Para comentarios, preguntas y sugerencias, visita el hilo original en https://forum.odroid.com/viewtopic.php?f=139&t=30267.

Recursos

Especificaciones del USB HID 1.1 http://www.usb.org/developers/hidpage/HID1_11.pdf

Dónde encontrar los códigos de las teclas para el teclado https://gist.github.com/MightyPork/6da26e382a7ad91b5496ee55fdc73db2

Para entender mejor los descriptores de los informes https://hamaluik.com/posts/making-a-custom-teensy3-hid-joystick/ y http://eleccelerator.com/tutorial-about-usb-hid-report-descriptors/

Herramienta para hacer descripciones (bastante tosca pero oficial) http://www.usb.org/developers/hidpage#HID%20Descriptor%20Tool

Descriptor de informe de teclado http://isticktoit.net/?p=1383

Utilidad para verificar los descriptores de informes http://eleccelerator.com/usbdescreqparser/

Modificación DTB de periférico OTG https://community.nxp.com/thread/383191

Soporte para el kernel principal del C2 (ajustes de frecuencia) https://forum.odroid.com/viewtopic.php?f=135&t=22717

Circuito Cerrado de TV Shinobi (CCTV): Creando un Sistema de Vídeo Vigilancia Usando el ODROID-HC2

April 1, 2018 By Dylan Tutoriales, ODROID-HC2

Es un placer para mí compartir mi experiencia sobre cómo montar un sistema CCTV doméstico utilizando el software Shinobi CCTV y el **ODROID-HC2**. Con algo de suerte, espero ayudar a alguien que esté pensando hacer los mismo.

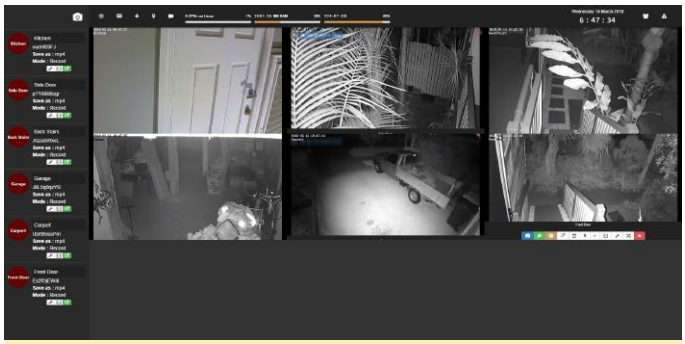


Figura 1 – Panel de instrumentos del Shinobi CCTV

En mi opinión, los tradicionales sistemas de alarma para el hogar generalmente causan molestias a los vecinos, y deben estar activados para ser efectivos. En comparación, un sistema de circuito cerrado de televisión (CCTV) es silencioso y también proporciona registros, bajo mi experiencia es un elemento disuasorio más eficaz para los posibles delincuentes. Un buen sistema de CCTV graba las 24 horas del día, los 7 días de la semana y avisa en caso de detectar movimiento.

Otros beneficios de un sistema de CCTV pasar por su utilidad para indicarte cuándo te han dejado un paquete en la puerta de entrada de casa; saber quién está en la puerta; o vigilar la barbacoa situada en la terraza mientras juegas a tus videojuegos dentro.

Advertencias y Descargo de Responsabilidad

Con el fin de respectar la privacidad de los demás, instala las cámaras únicamente con vistas a tu propiedad, y no a la de los vecinos. También es aconsejable que revises tu normativa local con el objeto de cubrir cualquier requisito adicional en relación a la señalización, las restricciones en la

grabación de audio y demás. Aunque solo recomiendo instalar las cámaras en el exterior, una posible excepción podría ser usar una cámara en el interior en forma de monitor para bebés. Por otro lado, tal y como me comentó uno de mis amigos, ¡Es muy importante asegurarse de que el acceso/seguridad a estas cámaras sea totalmente hermético!

Fundamentos del sistema

El cableado estructurado en forma de cable CAT6 o superior proporcionará comunicaciones robustas y seguras para tu sistema CCTV. Yo instalé 20 tomas Ethernet en mi casa, ocho en el techo para conectar las cámaras. El cableado normalmente es algo que instala un electricista, pero con paciencia y algo de investigación puedes hacerlo tú mismo. Tienes numerosos recursos online disponibles para indagar en este tema.

Figura 2 – Cableado estructurado

Las cámaras son de dos tipos: analógicas o digitales. Las cámaras analógicas necesitan más hardware para operar en un sistema en red. Prefiero las cámaras digitales puesto que se conectan a través de ethernet, ofrecen una configuración más flexible y por lo general cuestan lo mismo o quizás menos. Yo elegí cámaras que se alimentan a través del Ethernet (PoE) lo cual significa que puedo proporcionar energía y comunicaciones a través de CAT6. El estándar abierto para las cámaras de red es ONVIF (https://www.onvif.org). Recomiendo usar cámaras compatibles con ONVIF, como la Foscam FI9853EP, con el fin de mitigar los problemas de compatibilidad.

Figura 3 – Ejemplo de una cámara instalada

Configuración del ODROID-HC2

El ODROID-HC2 es perfecto para mi sistema CCTV. Ofrece velocidad de red gigabit, una interfaz SATA nativa HDD/SDD de 3.5 “o 2.5” y sin pitidos o sirenas innecesarios. Quiero almacenar imágenes durante al menos 30 días utilizando como medio de almacenamiento discos duros de baje coste.

Arranque el ODROID-HC2 utilizando la última Imagen SO Ubuntu minimal para ODROID-XU4. Grabé la imagen en una microSD usando WinDiskImager, luego la inserté en el ODROID junto con una conexión de Ethernet y un disco duro de 3.5 “. Encendí el sistema usando una fuente de alimentación de 12 voltios compatible. El ODROID necesitó unos 2 minutos para iniciarse y en el siguiente arranque se activó el acceso SSH a través de una IP asignada por DHCP.

Aplicamos las actualizaciones disponibles:

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

Montamos el disco duro. El disco duro debe aparecer como /dev/sda, aunque ejecuté el siguiente comando para comprobarlo:

```
$ sudo fdisk -l
```

En el siguiente paso asumimos que el disco duro se muestra como /dev/sda. Creamos una única partición utilizando fdisk. Recurre a la documentación de fdisk para particionar el disco.

```
$ sudo fdisk /dev/sda
```

Una vez que tengas una partición creada, formatéala:

```
$ sudo mkfs.ext4 /dev/sda
```

El ODROID-HC2 solo tiene espacio para un único disco duro. Como tal, simplemente monté la referencia de disco /dev/sda. Si lo prefieres, puedes montarlo usando el UUID. Añade la siguiente línea a fstab:

```
$ sudo nano /etc/fstab
/dev/sda /media/CCTV ext4 defaults 0 2
```

El siguiente script nos ayuda a detener el disco duro cuando apagamos el sistema. Descarga e instala el script:

```
$ wget
https://dn.ODROID.com/5422/script/ODROID.shutd
own
$ sudo install -o root -g root -m 0755
./ODROID.shutdown /lib/systemd/system-
shutdown/ODROID.shutdown
```

Decidí Compartir todo el disco duro usando Samba. Esto me permitía recuperar las grabaciones y enviarlas a un recurso compartido en red.

```
$ sudo apt-get install samba samba-common-bin
```

Luego, configuré Samba:

```
$ sudo nano /etc/samba/smb.conf

#===== Share Definitions =====
[CCTV]
comment = CCTV
path = /media/CCTV/
browsable = yes
writable = yes
guest ok = yes
read only = no
```

Reinicie el dispositivo y me aseguré de que se pudiera leer/escribir en el disco duro.

Configuración de Shinobi

Utiliza el script de instalación para instalar Shinobi CCTV y los correspondientes requisitos previos:

```
$ sudo apt-get install curl

$ bash <(curl -s
https://raw.githubusercontent.com/ShinobiCCTV/
Shinobi-Installer/master/shinobi-install.sh)
```

Instala todas las dependencias. Yo elegí la rama Shinobi Pro con las restantes opciones por defecto. Accede a la vista de administrador de Shinobi colocando el enlace: https://[ip-ODROID-HC2]:8080/super en tu navegador.

A continuación, agrega una nueva cuenta, añade el disco duro a la configuración y guarda los cambios:

```
"addStorage": [
 {
 "name": "second",
 "path": "/media/sda/CCTV"
 }
```

Es posible que también quieras cambiar la configuración del correo electrónico, las cables API y la contraseña del superusuario Shinobi.

Ahora conéctate a la interfaz principal de CCTV Shinobi con tu navegador en https://[ip-tu-ODROID]:8080 usando la nueva cuenta. Haz clic en el símbolo más [+] con la herramienta 'Add Monitor'. Configura tu cámara de la siguiente forma (tu configuración ideal puede variar):

```
Identity
Mode: Record
Monitor ID: (leave default)
Name: (i.e. Front Door)
Retry Connection: 0
Storage Location: second

Input
Input Type: H264 (or that supported by your
camera)
```

```
Connection Type: RSTP
RTSP Transport: UDP
Username: (Camera login)
Password: (Camera password)
Host: (Camera IP)
Port: (Camera RTSP Port)
Force Port: No
Path: (RSTP Path)
Analyzation Duration: 100000
Probe Size: 100000
Accelerator: No

Stream
Stream Type: FLV
FLV Stream Type: Websocket
Max Latency: 20000
Video Encoder: Copy
Audio Encoder: No Audio
TV Channel: No

Stream Timestamp
Enabled: No

Stream Watermark
Enabled: No

JPEG API Snapshot (cgi-bin)
Enabled: No

Recording
Record File Type: MP4
Video Codec: Copy
Audio Codec: No Audio
Double Quote Directory: No
Recording Segment Interval: 15

Custom
(leave blank)

Logging
Log Level: (set to silent once camera is
stable)
Save Log in SQL: No
```

Tendrás que revisar el manual de configuración de la cámara para determinar las opciones. La API JPEG debería funcionar en la mayoría de las cámaras, pero desafortunadamente no en la mía. Yo prefiero el protocolo UDP para las transmisiones de la cámara. Si tienes pensado transmitir a través de Internet o redireccionar a través de una LAN concurrida, selecciona el protocolo TCP. Se supone que el protocolo HTTP Live Streaming (HLS) es superior a la transmisión Flash Video (FLV), pero he tenido problemas para visualizarlo correctamente.

Existen muchas más personalizaciones para Shinobi. Puedes configurar un detector de movimiento, alertas de correo electrónico y programar clientes, por nombrar algunos. Encontré muy útiles las siguientes páginas sobre Shinobi:

https://goo.gl/47M2Ty
https://shinobi.video/docs/cameras
https://goo.gl/kvKax1

Necesitaras configurar la redirección de puertos u usar otros métodos para ver Shinobi a través de Internet. Podrás ver todas las transmisiones sin dificultad desde la casa de un amigo. Sin embargo, la visualización de las transmisiones en tiempo real desde un teléfono móvil parece dar problemas, aunque la reproducción de las grabaciones es perfecta. Espero que la visualización a través del teléfono móvil mejore con futuras actualizaciones o adaptaciones.

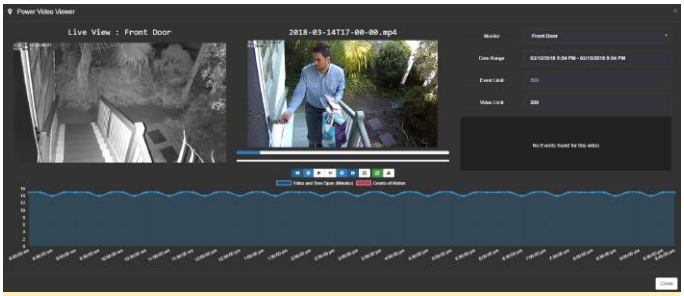


Figura 4 – Potente Visor del CCTV Shinobi

Estoy capturando y grabando imágenes con 6 cámaras a 720p 2M 15 fps. El consumo de la CPU de mi ODROID-HC2 está entre el 1% a 5% y la memoria a un 36%. Hasta ahora, el sistema se ha mantenido estable durante 2 semanas sin signos de degradación. No se bloquea ni se reinicia, ni muestra ningún otro comportamiento anormal. Durante varios días realice pruebas hasta lograr que el sistema funcionase tal y como deseaba.

Para comentarios, preguntas y sugerencias, visita el artículo original en **https://goo.gl/tJprLA**.

Estudio de Sonido Portátil: Graba Música con un ODROID-XU4Q en Cualquier Momento y en Cualquier Lugar

April 1, 2018 By Stephen Baldassarre Linux, ODROID-XU4

Stephen Baldassarre es un batería que trabaja en dos bandas activas que realizan producción de videos y películas. Lo que es más importante aún, también es el ingeniero de Golden Clam Machine Studio en Boise, Idaho. Sus clientes incluyen a Steve Schwarz de la banda de metal Pyrael y la organización sin ánimo de lucro Story Story Night. Stephen recientemente ha publicado un video en Youtube en el que muestra su inusual equipo de grabación, el cual ha llamado la atención d3e ODROID Magazine sobre su configuración.

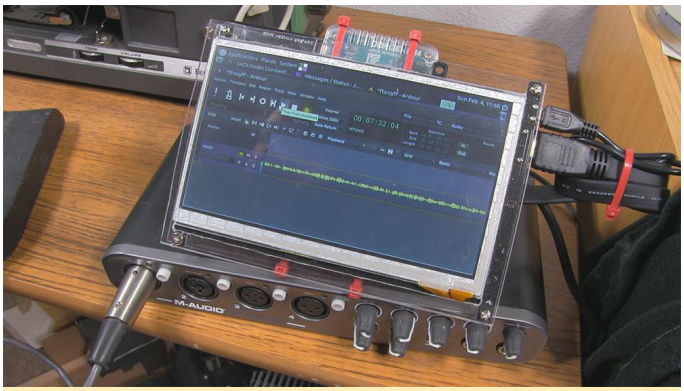


Figura 1 – Un ODROID-XU4Q grabando voz superpuesta a través de la interfaz USB

Requisitos de hardware

- ODROID-XU4Q
- Ventilador de 40mm 5V
- ODROID-VU7
- Fuente de alimentación US 5V/6A
- Carcasa transparente ODROID-XU4
- Unidad de sistema MicroSD de 64GB SanDisk

- Unidad USB de 64 GB SanDisk (para grabar pequeñas pistas)
- Placa de conexión USB3.0 a SATA con SanDisk 240GB SSD (para grabar grandes cantidades de pistas)
- Lámina de plexiglás
- 4 bridas pequeñas para tenerlo todo junto

Requisitos de Software

- Ubuntu (ubuntu-16.04.3-4.14-mate-odroid-xu4-20171212)
- Kit de conexión de audio JACK – Qt GUI V0.4.2: 4/7/2016
- Ardour 4.6.0 "Evening Star"

Stephen, ¿por qué has creado esto?

Utilizo a diario los ordenadores para la producción de audio y video, pero intento valerme de soluciones específicas tanto como me sea posible. Desafortunadamente, hay cierta escasez de sistemas de hardware asequibles en el mercado actual que soporten la variedad de grabaciones en directo que abarca la mayor parte de mi negocio. He estado utilizando M-Audio Fast Track Ultra (**https://goo.gl/bpx3Ly**), una interfaz USB con seis entradas analógicas, con un ordenador portátil ejecutando Sony Vegas para proyectos de grabación en directo más pequeños como 'Story Story Night', que en su mayor parte es dialogo con música en directo entre los narradores de cuentos. Todos los espectáculos se graban, mezclan y se cargan en varios servicios de streaming para que todos los escuchen. No me gusta llevar mi ordenador portátil a las actuaciones porque hay muchas cosas que pueden salir mal, incluidos los posibles robos y los fallos técnicos relacionados con Microsoft Windows. Para espectáculos más exigentes, como un concierto de rock de 3 horas, he estado recorriendo a mi HD24, que es mucho más fiable y tiene más entradas de audio que el Fast Track. Sin embargo, tengo que desmantelar mi estudio y reajustar mi sistema de megafonía para llevarlo de viaje. Empecé a preguntarme si podría crear una especie de sistema modular que cubriera todas mis necesidades de grabación en directo y quizás empezar a dar forma un posible negocio familiar de fabricación/venta. Pensé que algún tipo de ordenador de placa reducida podría ser capaz de gestionar esta tarea. No logré encontrar ningún ejemplo en Internet en el que alguien hubiera hecho algo parecido, así que me aventuré con el ODROID-XU4Q, debido a sus especificaciones superiores.

¿Qué hardware utilizaste?

En cierto modo fui hacia atrás con este tema. Mi colega Steve Schwarz, un incansable fan de GNU/Linux, me ha estado sugiriendo desde hace ya varios años que probara Ardor. Ardor 4 requiere al menos 2 GB de RAM, de modo que descarté la mayoría de SBC. Me gustaba la idea de un disipador de calor pasivo: funcionamiento silencioso, menos potencia, menos posibilidad de fallos. No quería lidiar con el ratón y el teclado, así que la solución más obvia era contar con una pantalla táctil. Elegí el ODROID-VU7 porque quería una pantalla un poco más grande y la resolución no era especialmente importante para mí. No iba a hacer mezclas; simplemente quería grabar directamente en una unidad USB y luego mover los archivos al ordenador de mi estudio o HD24 para empezar a mezclar. Obviamente, con una unidad USB no iba a manejar muchas pistas, así que también me hice con un adaptador USB3 a SATA junto con un SSD de 240GB para grabar las grandes actuaciones.

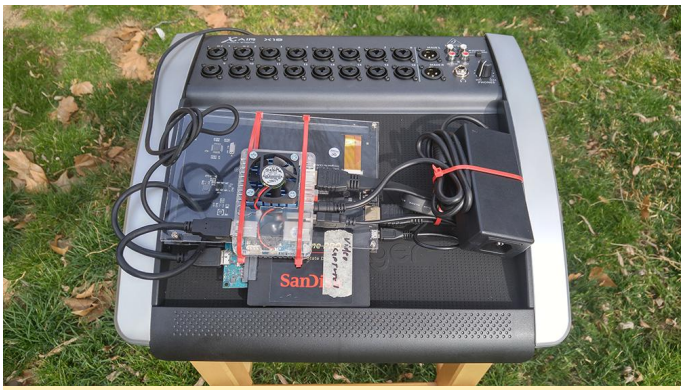


Figura 1 – Parte posterior de XU4Q con ventilador y SSD colocados entre las capas de Plexi

¿Cómo montaste toda la estructura?

No estaba muy seguro del tipo de requisitos de potencia necesitaría, así que opté por una fuente de alimentación de 5V/6A. La mayoría de los usuarios de Ardor parecen usar Ubuntu, de hecho, el sistema operativo fue en realidad la última decisión que tomé. Copié una imagen de Ubuntu 16.04 en una tarjeta

MicroSD que ya tenía. Me quedaba algo de plexiglás de otro proyecto, así que corté un rectángulo del tamaño del VU7 y utilicé los separadores y tornillos suministrados para montar el plexiglás. Hice unos cuantos agujeros en la parte posterior, del mismo tamaño y espacio que la carcasa del ODROID-XU4, y luego coloqué el ODROID-XU4Q en su lugar con un par de bridas. Los canales fueron cortados dentro del Plexiglass con una pequeña lima para evitar que se deslizaran las bridas. El ODROID-XU4Q está cerrado, de modo que actúa a modo de soporte para la pantalla, manteniéndola con un cierto ángulo si la coloco sobre una mesa. Parece un poco raro, pero me permite desmontarlo todo rápidamente si fuera necesario.

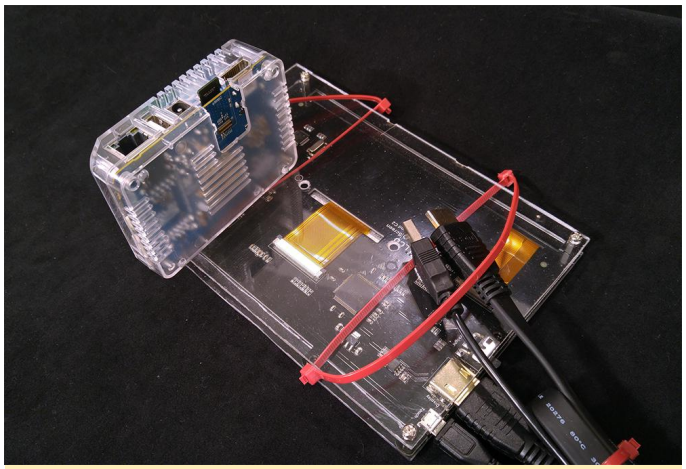


Figura 2 – Parte posterior de VU7 que muestra la placa de soporte improvisada

¿Cómo usas tu estudio portátil?

Inicialmente, el Fast Track ocupaba la parte posterior de mi PA continuamente, con varias salidas desde mi mezclador de audio. En la mayoría de las actuaciones, pongo la voz principal en input-1 y el resto de voces en input-2. Los instrumentos son sub-mezclado en estéreo y se alimentan en 3-4, mientras que los micrófonos del público son 5-6. La mayoría de mis clientes trabajan con presupuestos extremadamente ajustados. Al igual que con SSN, solo dispongo de 2 horas y media para convertir una actuación de 2 horas en dos podcasts de 45 minutos. Me he vuelto bastante hábil a la hora de obtener buenas mezclas sobre la marcha, pero siempre es mejor almacenar las voces por separado. Los micrófonos del público, que son inútiles para el sonido en directo, está conectados directamente a la interfaz.

Figura 3: 400 invitados buscan asiento en "Story Story Night"

Lo que normalmente estoy haciendo al final es usar un mezclador digital Behringer X18 con posibilidad de interfaz USB. Esto me permite grabar submezclas de 6 pistas para trabajos más pequeños, aunque si quiero puedo tener hasta 16 pistas independientes, todo sin reconfigurar nada. También he probado hasta 26 pistas desde un X32, un mezclador digital de 32 canales con resultados muy diversos (no es un juego de palabras). Ni que decir tiene que estoy muy contento de contar con un sistema de grabación de reserva en todas las actuaciones: https://www.youtube.com/watch?v=4TVOxfPE2ps.

The Fabulous Chancellors necesita muchas pistas, pero un único y diminuto ordenador

¿Qué problemas encontraste durante su creación y cómo los solventaste?

No tenía experiencia con los SBCs o Linux con anterioridad a este proyecto, do modo que no lo habría hecho realidad sin la ayuda de Steve Schwarz. Él vive en Nueva York, así que recibir asesoramiento

requería de una cierta paciencia. Incidentalmente, se me ocurrió este sistema de grabación Ubuntu/Ardor por sus sugerencias y porque él puso una consola analógica y HD24 en su estudio por mi culpa. De todos modos, cuando empecé no sabía que todo lo importante pasaba por el terminal comandos. Sin embargo, el único comando que realmente necesitaba conocer era:

```
$ sudo apt-get install ardour
```

Estoy muy contento de que Ardour 4 se encuentre en el repositorio de Ubuntu, simplemente había que descargarlo e instalarlo junto con todas sus dependencias. Hacer que funcionara correctamente me llevó a experimentar un poco. Descubrí que a Ardor no le gusta que sea instalado si actualizas el sistema operativo, de modo que lo dejé con la instalación estándar "ubuntu-16.04.3-4.14-mate-odroid-xu4-20171212".

El ODROID-XU4Q tiene una cierta tendencia a sobrecalentarse y fallar cuando se graban muchas pistas al mismo tiempo. Añadir un ventilador al disipador de calor del ODROID-XU4Q ayuda bastante. Es muy sensible a la electricidad estática, así que necesitaba encontrar algún tipo de revestimiento blindada. Mientras tanto, he adquirido el hábito de colocar siempre una puesta a tierra antes de tocarlo durante las sesiones de grabación.

Con Fast Track, las salidas principales están desactivadas y las salidas alternativas son ahogadas en reverberación. Es posible hacer overdubs (grabar más pistas a posteriori) pero no lo recomiendo. Esto parece ser una barrera creada intencionadamente cuando no se usan los drivers patentados de M-Audio.

JACK es muy importante con Fast Track porque Ardor no se conecta directamente con los drivers ALSA en este caso. JACK debe abrirse e iniciarse para que Ardor se conecte. Ardor no inicia JACK automáticamente, y no se puede configurar en "tiempo real" o Ardor no se abrirá ni creará sesiones. También es importante asegurarse de que las tasas de muestreo coincidan antes de ejecutar Ardor. Si utilizas un mezclador digital, Ardor debe coordinarse con el reloj interno del mezclador antes de conectarse. JACK no es necesario con los mezcladores digitales Behringer, y solo han sido probados Fast Track Ultra, X18 y X32.

Desde que le he puesto un ventilador, he probado el ODROID-XU4Q grabando 16 pistas durante dos horas seguidas y no he tenido ningún problema. Antes de colocar el ventilador, el disipador de calor se calentaba a los cinco minutos. Con el ventilador, el XU4Q se mantiene a baja velocidad aproximadamente 2/3 del tiempo con breves ráfagas de alta velocidad. El uso de la CPU y la memoria son bastante bajos, así que no dudo que pueda grabar 32 pistas al mismo tiempo durante períodos prolongados. De lo único que me arrepiento de este proyecto, es de no haber gastado 25$ adicionales en una pantalla de 8" de alta resolución porque algunas ventanas del programa no pueden asustarse al espacio de 800×480. Es muy probable que pronto me haga con una pantalla más grande.

Juegos Linux: Juegos Saturn – Parte 3

April 1, 2018 By Tobias Schaaf Juegos, ODROID-XU4

Una vez más, volvemos a la temática de este mes sobre los juegos de Sega Saturn para el ODROID-XU3/XU4. La última vez hablé de un montón de shoot'em ups, pero en este artículo hare una buena combinación de diferentes géneros, aunque sí es verdad que he seleccionado bastantes juegos "mecha" con los que disfruto bastante. Como hasta ahora he ido cubriendo la mayoría de los juegos en un único artículo, esta vez me he esforzado en hacer unas descripciones más breves.

Aunque estos juegos han sido probados en el ODROID-XU3/XU4, deberían funcionar igual de bien, si no mejor, en el próximo ODROID-N1. Mis pruebas iniciales con el ODROID-N1 han tenido muy buenos resultados, de modo que no me sorprendería que el N1 soportara la emulación de Sega Saturn con bastante fluidez.

Macross – Do you remember Love

Lo que realmente llama la atención en este juego son las increíbles escenas de anime. Estas no son las típicas escenas al estilo JRPG como en **Popfull Mail**; son directamente de la serie de anime japonesa Macross. Las escenas están increíblemente detalladas, es difícil creer que estás viendo una escena del juego y no del anime real.

Se pueden encontrar escenas similares dentro del propio juego: cuando te encuentras un jefe y cuando la pantalla muestra una conversación al estilo anime entre tu personaje y el jefe. El juego está completamente en japonés, pero esto no interfiere en el disfrute del juego, aunque es posible que te pierdas algunos elementos menores de la trama.

En este juego, vuelas con un avión transformable que puede convertirse en un mecha. Cada etapa tiene sus ventajas y desventajas. El modo avión es muy rápido y fácil de maniobrar, pero tus armas son algo más débiles. Al cambiar a un mecha fortaleces tus

ataques, pero te vuelves más lento, más lento en tus movimientos y más grande, con lo cual es más fácil que te disparen. Luchando a través de diferentes niveles, tu forma puede estar limitada a una o dos de las tres opciones dependiendo del escenario.

Figura 1 – Macross tiene tres escenarios de lucha, cada uno con sus pros y contras

Figura 2 – Macross tiene tres escenarios de lucha, cada uno con sus pros y contras

Figur3 3 – Macross tiene tres escenarios de lucha, cada uno con sus pros y contras

También tienes tres armas diferentes: misiles que apuntan automáticamente y que cargas y disparas manteniendo presionado un botón, una ametralladora rápida y una bomba que cubre la mayor parte de la pantalla, la cual te ayudará si estás rodeado de enemigos. Como los enemigos te atacan desde tres planos (primer plano, segundo plano y fondo), es probable que los misiles que apuntan automáticamente sean tu mejor opción.

El juego te permite guardar tus progresos en la memoria del sistema. Este es uno de los pocos juegos que realmente caben en la memoria del sistema, sin que éste te avise de que no tiene suficiente espacio. El juego es un buen shooter muy divertido y en el que las escenas de anime son simplemente sorprendentes.

Magic Knight Rayearth

Este juego está basado en un anime del mismo nombre en el que tres colegialas son transportadas a un mundo mágico donde se les dice que son las “Magic Knights” y se supone que tienen que salvar el mundo. El juego empieza con una larga introducción que ofrece una buena combinación de escenas de anime, gráficos del juego y muchos diálogos.

Magic Knight Rayearth fue traducido por Working Designs, una compañía que pasó muchos años exportando juegos japoneses al mercado norteamericano y haciendo un trabajo increíble, con

traducciones muy fieles y sin una interpretación de voces monótona. Este fue el último juego que exportaron para Sega y el último lanzamiento oficial en Norte América de Sega Saturn.

El juego viene a ser un RPG de acción similar a The Legend of Zelda o Beyond Oasis. Controlas los movimientos de las tres chicas al mismo tiempo, pero solo una está activa. Hikaru (chica roja) forma parte de un club de kendo y usa una espada que es un poco corta y difícil de manejar. Umi (chica azul) forma parte de un club de esgrima y usa un espadín más largo para sus ataques. Fuu (chica verde) forma parte de un club de tiro con arco y lucha con su arco. También pueden usar magias: Hikaru posee habilidades basadas en el fuego; Umi es muy habilidosa con el agua; y Fuu usa técnicas basadas en el viento y hechizos de curación.

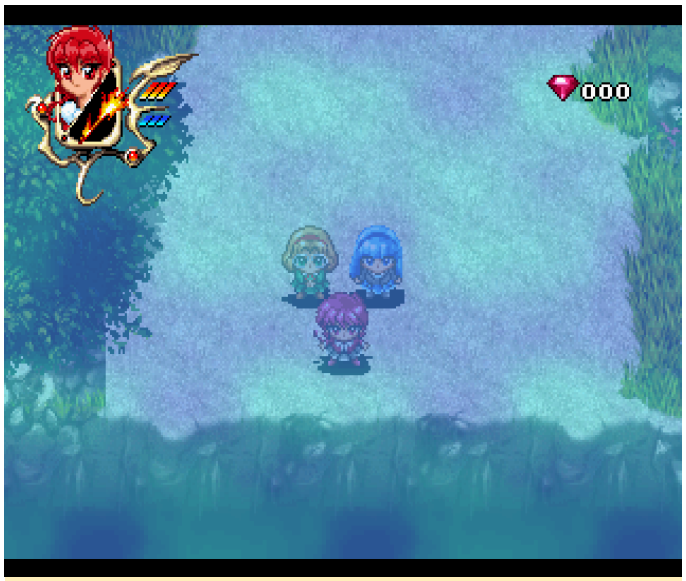


Figura 4 – Los tres personajes principales de Magic Knight Rayearth, comienzan su aventura.

Los diálogos del juego están muy bien escritos, con doblaje en algunas partes del juego. De vez en cuando te encontrarás con nuevas personas o situaciones, que te deleitaran con nuevas y agradable escena de anime. En general, es un juego bastante divertido. Deambulando por diferentes mazmorras, lucharás contra infinidad de monstruos al mismo tiempo que intentas encontrar tesoros ocultos. De vez en cuando localizas elementos que mejoran tu salud o tu magia, pero cada elemento solo funciona una vez, así que piénsate bien qué chica lo va ha recibir.

Me llama la atención los colores brillantes y el sistema de juego se comprende perfectamente. También me gusta que puedas guardar los progresos del juego donde quieras y te recomendo que lo hagas. La traducción está muy bien hecha, haciendo de este juego una obra maestra de Sega Saturn. No he jugado mucho, pero lo que he visto me ha gustado bastante y pienso seguir jugando durante bastante tiempo. Definitivamente deberías probar este juego.

Figura 5 – Llegando a la casa de Presea para conseguir tus primeras armas

Figura 6 – En el camino de regreso de tu primera misión para recoger Escudo

Mega Man 8–Anniversary Edition

No soy un gran fan de la serie Mega Man, pero de todos los juegos de esta serie a los que he jugado, este es posiblemente el que más me gusta. Cuando se trata de gráficos, me gustan los buenos juegos al

estilo cómic. Juegos como Monkey Island 3 o Mega Man 8 no tienen límites en el tiempo debido a su estilo gráfico. Se veían increíbles por aquel entonces y todavía lo son hoy en día. Si te gustan los juegos 2D, esto es de lo mejorcito que puedes encontrar. Aunque el juego también existe para PlayStation One, la versión de Saturn ofrece mejor música (PCM VS Midi), jefes adicionales (Cut Man y Wood Man), ilustraciones extras, etc. La versión PS1 además, presenta algunos fallos menores en cuanto a gráficos.

Figura 7 – El menú de selección de nivel en Mega Man 8

Desde la pantalla de selección de nivel, puedes ir a mundos muy diversos en los que luchas contra diferentes enemigos y jefes. Si lográs derrotarlos consiguiras poderes que puedes utilizar contra otros enemigos, al igual que ocurren en el resto de juegos de Mega Man. Desde la pantalla de selección de nivel, también puede ir a “home”, donde puede canjear los “screws” reunidos por mejoras especiales, como son los diferentes estilos de ataque y otras cosas.

Figura 8 – Pantalla de actualización de Mega Man

Algunas de estas mejoras o actualizaciones marcan una gran diferencia. Un ataque que no sólo golpea a un único enemigo, sino a todos los enemigos, suena a algo que me gustaría tener. Mega Man 8 ofrece niveles muy variados con diferentes enemigos, diferentes estilos gráficos y diferentes desafíos.

Figura 9 – Existen niveles muy diversos en Mega Man 8, como es el cielo

Figura 10 – Existen niveles muy diversos en Mega Man 8, incluido el interior de una mazmorra mecanicada

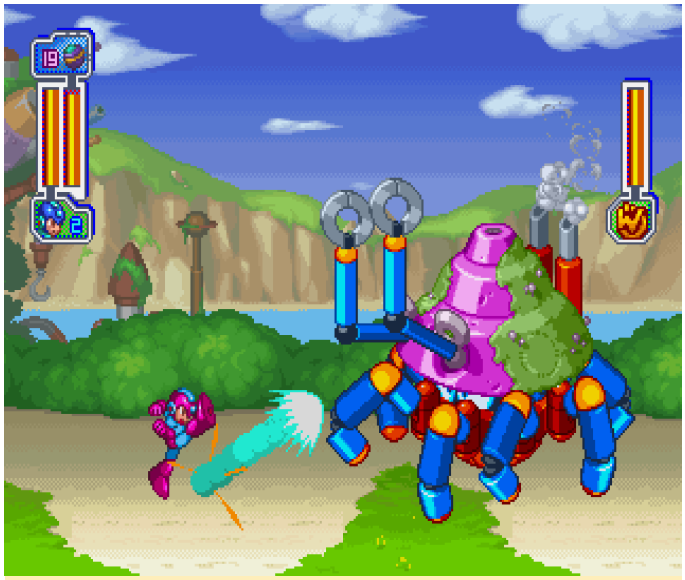


Figura 11 – ¡Ningún juego de Mega Man está completo sin las famosas batallas de los jefes!

En general, este juego es una auténtico festival de colores y gráficos magníficamente dibujados. La increíble banda sonora completa esta maravillosa experiencia. Recomiendo este juego de Sega Saturn para ODROID-XU3/XU4/N1.

Mobile Suit Gundam Side Story I, II, and III

Siempre he dicho que el Sega Saturn no estaba hecha para juegos en 3D y que los desarrolladores deberían haberse quedado sólo con los juegos 2D. ¡Chaval, eso es mentira!

Es cierto que hay muchas consolas con mejor potencial 3D que Sega Saturn, pero Saturn tenía sus joyas: juegos que sacaban lo máximo que podia ofrecer la consola, a la vez que mantenían la velocidad y la jugabilidad. La serie Gundam sin duda se ajusta a esta descripción.

Figura 12 – Los gráficos 3D en Mobile Suit Gundam Side Story son sorprendentemente buenos para Sega Saturn

Mi primer juego Gundam 3D era en realidad de Sega Dreamcast. Al instante me enamoré de él, aunque viendo los juegos de Sega Saturn, estoy seguro de que también los habría disfrutado con mucha emoción. Definitivamente los disfruto ahora y me sorprende lo bien que se ejecutan en mi ODROID. Disfruto mucho de este juego, incluso si a veces resulta difícil encontrar el objetivo, puesto que el juego está completamente en japonés. Aun así, el sistema de juego es fácil de entender y entras en acción casi de inmediato.

Figura 13: apunta a un enemigo y dispara hasta que sea destruido

Utiliza tu Gundam para cazar a los diferentes enemigos usando una ametralladora, tu cañón, tu armamento de zona (como la granada o el misil) o una espada. Tengo que admitir que todavía no he descubierto cómo hacer que la espada funcione. El informe de la misión es un poco difícil de entender, puesto que está en japonés. Está doblado al completo, lo cual es bueno, pero no entiendo ni una palabra.

El primer juego de la serie tiene una interfaz de misión muy minimalista. Tienes un mapa general pero rara vez puedes ves algún punto de la misión, así que es difícil determinar tu objetivo. A partir del segundo juego, la pantalla de información se vuelve poco más elocuente y se puede deducir a partir de los iconos, flechas y objetos que parpadean cuál será tu objetivo.

Figura 14 – Tu mapa de misión en Mobile Suit Gundam 2 y 3, que visualizas antes de cada misión

Cada nivel implica luchar contra diferentes tipos de Mecha/Gundam enemigos y contra un jefe más fuerte de vez en cuando al que debes derrotar o ahuyentar. Estos jefes pueden suponer un auténtico reto ya que se mueven muy rápido y pueden ser difíciles de alcanzar con tu principal cañón. Durante el transcurso del juego, y desde un juego de la serie al siguiente, tu Gundam cambiará, consiguiendo mejores armas que te permitirán hacer más daño y desplazarte más rápido a medida que vas progresando.

Figura 15 – El nuevo jefe acercándose ... todas las escenas están hechas con gráficos del juego

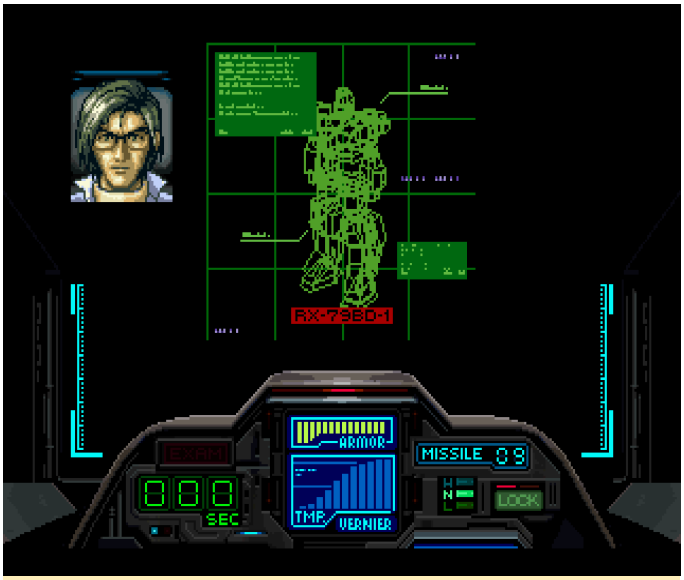


Figura 16 – Información general de tu Gundam y su sistema

Realmente disfruto jugando a este juego en ODROID y se lo recomiendo encarecidamente a cualquiera que le guste los juegos de acción.

Mobile Suit Z Gundam

Sí, este es otro juego de Gundam, y no el último, hay uno más en la sección de menciones honorificas. Sin embargo, este es diferente de los anteriores. No es un juego en 3D, sino 2D, es más del estilo de Macross al que hemos aludido anteriormente.

Mobile Suit Z te permite alternar tu Gundam entre una especie de robot y una aeronave. Similar a Macross, la aeronave se desplaza más rápido y dispara más rápido, pero el Gundam es mucho más

poderoso, con acceso a ataques adicionales y armas como la posibilidad de apuntar a enemigos en primer plano y de fondo, y cuenta un sable muy potente y rápido para matar a tu enemigo.

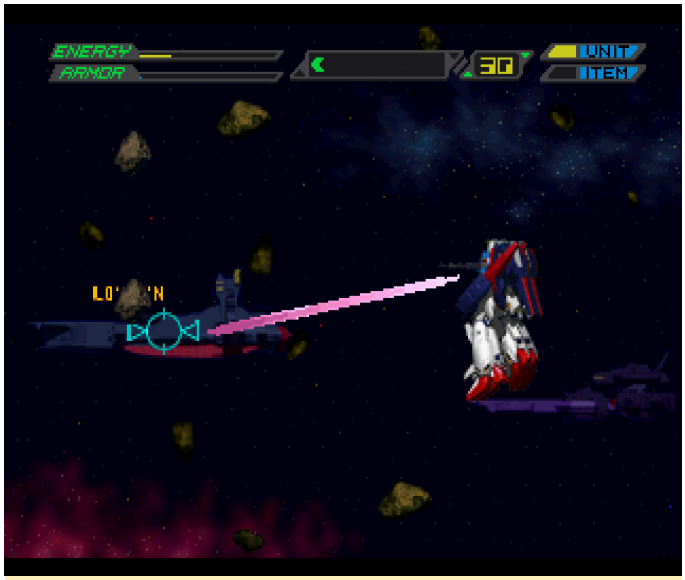


Figura 17 – Apuntar a un enemigo de fondo solo funciona al estilo Gundam

Comparandolo con los juegos de Gundam que hemos mencionado anteriormente, este juego está lleno de escenas de anime. Intros, antes y después de las misiones e incluso a veces durante las misiones, tienes escenas de anime que se ajustan bastante bien al escenario. En comparación con Mobile Suit Z Gundam Zenpen Z no Kodou, esta versión es bastante más fácil de controlar ya que todas tus acciones están asignadas a diferentes botones, lo que facilita cambiar entre atacar o apuntar a enemigos de fondo mientras liquidas enemigos a mitad de suelo.

Figura 18: Derrotar algunos enemigos con tu sable siempre es muy gratificante y divertido

Al final de los niveles, a menudo te encuentras con un jefe que puede hacerte mucho más daño que el resto de enemigos que suelen atacarte todo el tiempo. Me gusta que siempre haya conversaciones entre los personajes lo cual hace que parezca que estás ahí en ese instante.

Figura 19 – Antes de cada pelea con jefes hay una escena en la que apareces tú y tu enemigo

En la esquina superior izquierda de la pantalla hay dos barras: “Energy”, que es tu salud, y “Armor”. Esta última se regenera con el tiempo, de modo que recibir algunos golpes no es tan grave. Después de cada nivel, aparece una pantalla con tus progresos, te muestra cómo de bien lo has hecho y cómo ha mejorado tu personaje y Gundam. Con el tiempo, se volverá mucho más fuerte, podrá recibir más golpes,

se regenerará mucho más rápido y apuntará a más enemigos al mismo tiempo. Como de costumbre, el juego está desarrollado por completo en japonés, pero el sistema de juego es bastante fácil de entender.

Pocket Fighters

Este juego se ejecuta mejor con la expansión de memoria de 4MB. Cuenta con muchos clips animados, y la memoria extra mejora bastante las animaciones. Si comparamos la Sega Saturn con la PS1, las dos versiones de Pocket Fighters son muy similares, aunque la Sega Saturn cuenta con algunas animaciones más, y un golpe "Around the World" que no está presente en la versión para PS1. Sin embargo, la versión para PS1 está disponible en inglés, mientras que Saturn está solo en japonés, aunque esto realmente no afecta el juego. Como la versión de Sega Saturn funciona muy bien en XU3/XU4 (u ODROID-N1), depende de di la versión a la que desees jugar.

Figura 20 – Inicialmente solo ves tres personajes masculinos (mira la imagen del medio) pero cuando te desplazas a la izquierda o la derecha de Ruy o Ken descubrirás dos personajes más "ocultos" para elegir

Figura 21 – Inicialmente solo ves tres personajes masculinos (mira la imagen del medio) pero cuando te desplazas a la izquierda o la derecha de Ruy o Ken descubrirás dos personajes más "ocultos" para elegir

Figura 22 – Inicialmente solo ves tres personajes masculinos (mira la imagen del medio) pero cuando te desplazas a la izquierda o la derecha de Ruy o Ken descubrirás dos personajes más "ocultos" para elegir

A este juego se juega como a la mayoría de los juegos de lucha, pero con un divertido estilo de personajes "chibi". Durante algunos combos, los personajes cambian su vestuario en numerosas ocasiones, haciendo que el juego sea bastante divertido. Definitivamente no es un juego de lucha que pueda tomarse demasiado en serio, ofreciéndote un estilo único.

Figura 23 – Un cofre con gemas dentro aparece entre los luchadores al inicio de cada pelea

Este juego también es conocido como “Super Gem Fighters” en máquinas recreativas. Este nombre hace referencia al hecho de que recoges gemas durante las peleas abriendo un cofre o golpeando al enemigo.

Figura 24 – Las gemas que ganas durante los combates le dan al juego un segundo nombre: Super Gem Fighters

Figura 25 – Las gemas que ganas durante los combates le dan al juego un segundo nombre: Super Gem Fighters

Si eres capaz de alcanzar un combo con éxito, el enemigo soltará un cofre con varias gemas de diferentes tamaños en su interior. Probablemente sea uno de mis juegos de lucha favoritos, independientemente del sistema.

Robo Pit

Este es otro juego 3D de Sega Saturn que no defrauda. Los gráficos no son tan buenos, de hecho, son bastante simples, pero creo que precisamente eso es lo que realmente le salva. Funciona bastante bien, aunque creo que no se ejecuta a toda velocidad, aun así, es muy jugable y posiblemente no notarás que va un poco lento.

En este juego, puedes construir tu propio robot y usarlo para luchar contra otros robots en un escenario (supongo que es por eso por lo que se llama Robo Pit en primer lugar). Tu robot es personalizable con una variedad de diferentes tipos de cuerpo, piernas (para conducir, saltar o incluso volar), brazos (que tienen diferentes armas) e incluso ojos.

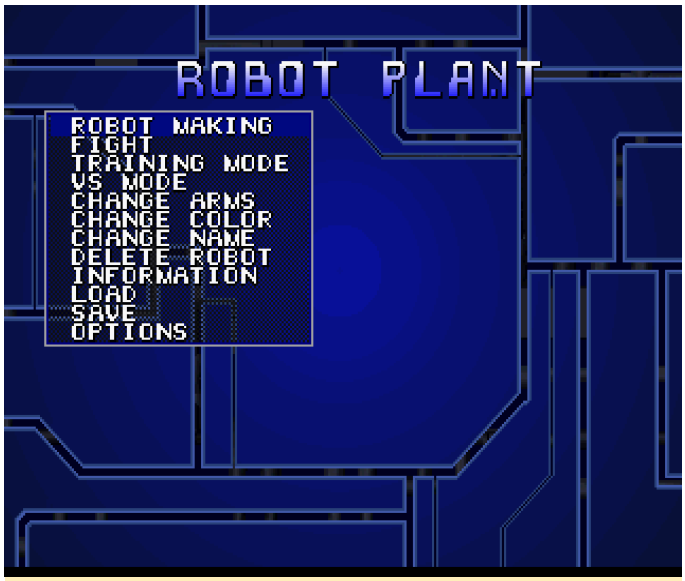


Figura 26 – El menú del juego es muy simple, pero te permite acceder a todos los aspectos de tu robot y tu trayectoria

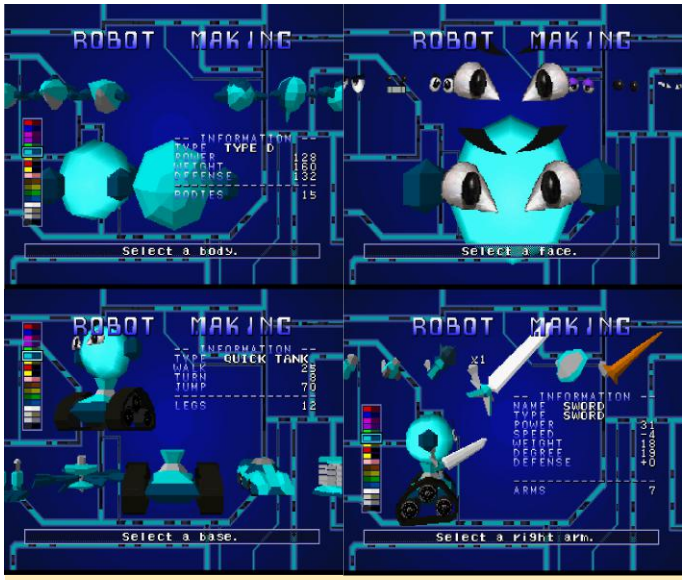


Figura 27: Crea tu propio robot según tus preferencias

Después de crear y asignar un nombre a tu robot, avanzas hacia la batalla. Los combates son bastante simples: aplastar, trocear, disparar o golpear a tu enemigo hasta que su barra de salud llegue a cero, antes de que la tuya descienda. Cada pelea te otorga puntos que aumentan tu rango, lo cual te permite luchar contra enemigos aún más fuertes. Después de cada pelea tus estadísticas aumentan dependiendo de cómo lo hayas hecho en el combate. Si solo usas tu arma izquierda para atacar, tu habilidad con esa arma aumentará mientras que el arma del lado derecho se mantendrá igual. Obtienes puntos con golpes adicionales, experiencia para tu arma izquierda y derecha, y defensa dependiendo de cómo se haya desarrollado el combate.

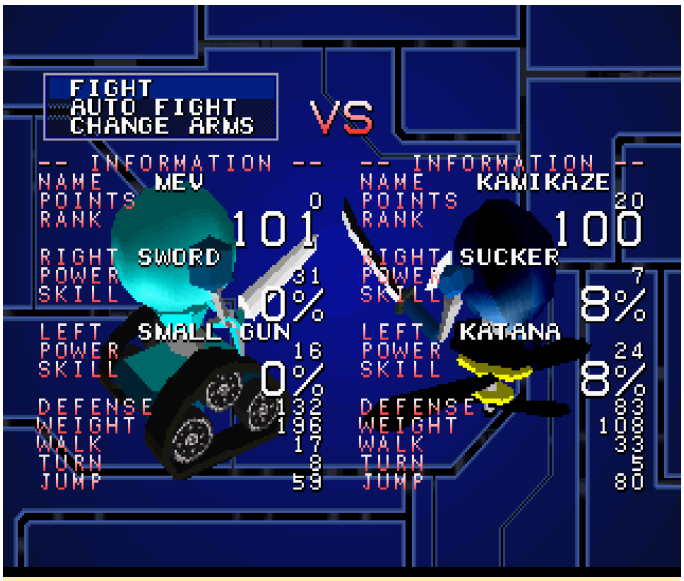


Figura 28 – Seleccionar un enemigo y vencerlo – ¡Tan fácil como eso!

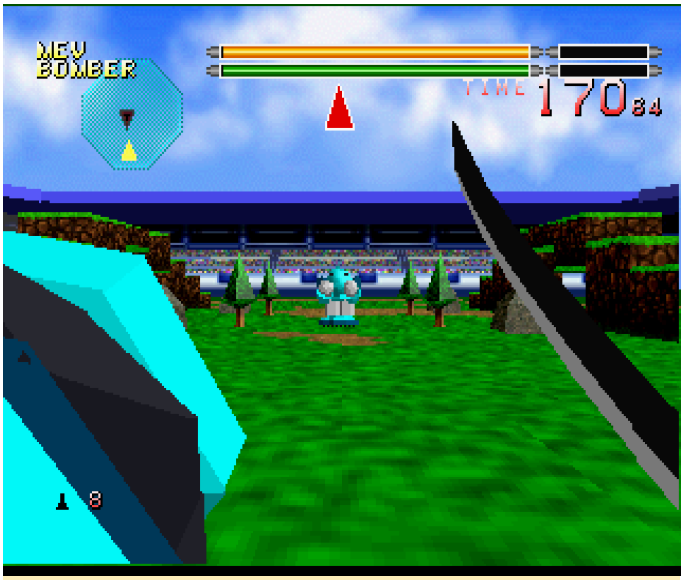


Figura 29 – Los escenarios no tienes demasiados detalles, pero se ajustan al juego

Si vences a un enemigo, cogeras sus armas y podrás usarlas. Si pierdes una pelea, perderás tus armas y tendrás que recurir a algo de tu inventario, de modo que es bueno hacerse con algunos repuestos de los alrededores o obtenerlos de los enemigos más débiles.

Los gráficos son bastante simples, especialmente los escenarios, pero se adaptan perfectamente al juego. Puede cambiar entre una perspectiva en primera persona y dos perspectivas diferentes en tercera persona. Es un pequeño juego bastante divertido con poca historia o estimulo. Simplemente “golpea al

pamplinas de tu enemigo y consigue una bonificación". Es un buen juego para dedicarle media hora más o menos, realmente me gusta ejecutarlo en el ODROID.

Menciones Honoríficas

Marvel Super Heroes

Por lo general no me gustan demasiado los juegos de lucha, aunque este juego es bastante bueno, y simplemente lo ubiqué aquí porque ya tenía demasiados juegos citados en la sección antiror. Honestamente es un juego realmente bueno. Los sprites de los personajes son enormes, con colores muy vibrantes y el sistema de juego es impresionante. Se ejecuta con el paquete de expansión de RAM de 1MB o 4MB para más animaciones.

Mega Man X3

Aunque este no es un mal juego, prefiero mejor Mega Man 8. Mega Man X3 se le ve mucho más serio que el resto de juegos de Mega Man. En el video de presentación, los coches y los edificios de la ciudad están completamente destruidos, lo que le da una sensación más siniestra. Los personajes son un poco más grandes y parecen más experimentados. Siempre he pensado que Mega Man era muy similar a Astro Boy: un niño luchando contra monstruos androides imaginativos. Mega Man X3 parece ser más serio que esto. También tiene algunos fallos gráficos menores al ir andando, pero nada grave. No obstante, es un buen juego.

También está Mega Man X4 pero parece tener problemas con los controles. Usando yabause-qt te te quedas atascado cuando intentas entrar en el juego, aunque con yabause-gtk parece funcionar. En mi opinico, todavía no vale la pena cambiar de emulador.

Metal Black

Realmente es un juego de disparos bastante interesante. Es muy básico: un grupo de enemigos te atacan, los matas y de vez en cuando te encuentras con un jefe. Lo diferente de este juego es que tus "power-ups" simplemente caen del cielo como flores que se lleva el viento y los recoges. Cuantos más recoges, más fuerte se vuelve tu ataque principal (6 niveles). La misma energía también se usa para iniciar un "mega-ataque" que inflige mucho más daño y normalmente mata a todo lo que aparece en pantalla. El único problema que tengo con este juego es que una vez que empiezas tu ataque especial, se agota por completo tu energía. No he encontrado ninguna forma de detenerlo. Esto significa que tras un ataque especial te vuelve muy débil y haces muy poco daño.

Metal Slug – Super Vehicle-001

Este juego es similar a la versión de NeoGeo o PS1, aunque ligeramente superior a la versión de PS1, ya que la RAM extra en el Sega Saturn permitía más sprites en las animaciones. Esta versión funciona bastante bien, aunque he experimentado algunos fallos gráficos esporadicos. Este juego requiere un pack de expansión de memoria de 1MB o 4MB para que funcione.

Mobile Suit Z Gundam Zenpen Z no Kodou

Este juego es similar al Mobile Suit Z Gundam, pero con controles ligeramente diferentes. En lugar de tener asignado cada botón del mando para un ataque, tienes que usar los botones para alternar entre las armas y usar un único botón para atacar. Es un poco incómodo, así que no es tan bueno como Mobile Suit Z Gundam.

Nights Into Dreams...

Probablemetne hay quien me mataría si no mencionara este juego. Incluso algunos podrían matarme por sólo incluirlo en una "nota a pie de página". Es un juego interesante, pero solo lo he jugado un par de minutos una vez. Para mí no es un juego que me llame la atención, ya que rara vez lo ejecuto, aunque puedo ver por qué a algunas personas realmente les gusta. Funciona bien en el XU4, pero se aprecia algo de lentitud. La parte 3D ralentiza el juego, quizás funcione mejor en la N1 o con un sistema 3D autentico

Norse by Norsewest (también conocido como Lost Vikings 2)

Realmente es un buen juego de puzzles, simplemente no soy un gran fan de este tipo de juegos. Aunque es cierto que jugué al primer juego de Lost Vikings en el Amiga CD32 durante bastante tiempo, tampoco es

uno de mis juegos favoritos. En este juego controlas a tres vikingos con diferentes habilidades, guiándolos por diferentes niveles, resolviendo puzzles para vencer al malvado Tomator. Es un pequeño juego divertido, con mucha interpretación de voz y buena música, pero no es para mí.

Panzer Dragoon Series

A mucha gente parece gustarle el Panzer Dragoon, no son malos juegos, simplemente creo que no deberían haber existido para la Sega Saturn. La Saturn era buena con los gráficos 3D. Estos juegos están completamente en 3D, y como tales, parecen ejecutarse bastante mal, al menos en los ODROIDs. También son bastante lentos con el ODROID-XU4. Podrían haber sido buenos juegos en la verdadera Sega Saturn, pero en ODROID, simplemente no son tan buenos. Me gusta Panzer Dragoon Saga (un juego de rol de la serie) pero los gráficos no son muy buenos. Apuesto a que estos juegos habrían sido mucho mejores en el Sega Dreamcast.

El último juego de la serie, Panzer Dragoon Orta, fue lanzado para la Xbox original, y te da una idea de cómo podría haber sido el juego si se hubiera exportado a la Dreamcast. Aún así, el Panzer Dragoon Saga ganó varios premios y es fácil adivinar por qué. Sin embargo, huviera sido mejor si los gráficos estuvieran más pulidos.

Parodius

Parodius en realidad viene con dos juegos: Parodius y Fantastic Journey. Se parece mucho a Gradius y se supone que es una parodia de éste. Tiene un estilo cómico interesante y no suele tomarse demasiado en serio. Tienes muchos personajes diferentes donde elegir, todos tienen su propia arma individual y actualizaciones, de modo que es divertido experimentar con los diferentes personajes.

También viene con un modo automático y semi automático, donde el PC gestiona las actualizaciones de tu arma para que puedas concentrarte en disparar. Esto funciona muy bien, conseguirás un ataque muy fuerte en poco tiempo, ya que los poderes son muy comunes. Sin embargo, como con todos los juegos de estilo Gradius, pierdes todo cuando te destruyen.

Princess Crown

Este es un curioso juego de acción RPG con grandes personajes y un sistema de lucha en tiempo real. Realmente me gusta, pero me gustaría aun más si pudiera entender lo que dicen los personajes, ya que el juego esta completamente en japonés. Es una lastima, ya que de veras me hubiera gustado disfrutarlo al completo.

Purikura Daisakusen

Se trata de un casi prefecto arcade. Si conoces la versión arcade, conoces la versión de Sega Saturn. Probablemente sea un juego muy gracioso, pero en lugar de volar y realizar cualquier desplazamiento lateral, caminas a pie a través de los diferentes niveles. Tienes un pequeño compañero que va evolucionando poco a poco a medida que avanzas, tiene un ataque especial con el que puede golpear a todos los enemigos de la pantalla. Me gusta este juego por los sprites coloridos y los fondos llamativos, pero los constantes efectos de sonido “bang bang” de los disparos pueden ser un poco molestos.

Rabbit

Este es otro juego que coloco aquí simplemente porque ya es suficentemente extensa la sección anterior. Rabbit es otro juego de lucha en el que peleas conta el “espíritu de una bestia”. Tú y tu enemigo teneis un espíritu animal que podeis invocar, lo cual te permite realizar ataques especiales. Una vez que has vencido a un enemigo, recoges su espíritu para usarlo como ataque especial contra tu próximo enemigo. Es un juego de lucha muy interesante, aunque no me gustan todos los sprites de personajes ya que algunos no están muy pulidos. Hay juegos con mejores gráficos, pero al menos el estilo de lucha es único.

Radiant Silvergun

Este es el primer shoot ’em up al que realmente jugué en la Sega Saturn. También lo incluí en mi primer artículo sobre Sega Saturn para ODROID en septiembre de 2016. Lo que más me gusta de este

juego es la disponibilidad de los diferentes ataques. Tienes seis ataques diferentes, desde ataques automáticos hasta fuertes ataques frontales pasando por lo ataques con los que disparas a los enemigos que tienes detrás. Incluso hay chispas eléctricas que se vuelven más fuertes cuanto más atacas. El mando de Sega Saturn tenía seis botones de acción, lo cual significa que cada ataque estaba asignado a un botón diferente. Hay más botones que puedes usar, como son los botones superiores que activan una espada con la que puedes atacar y recoger munición enemiga, lo que la hace perfecta para evitar golpes. También podrías lanzar un ataque aún mayor que cubría casi toda la pantalla, golpeando así a muchos enemigos al mismo tiempo o simplemente causando un daño adicional a los jefes.

Radiant Silvergun fue mi primer shoot 'em up para Sega Saturn y era bastante bueno. A pesar de que está hecho como un juego en 3D, se ejecuta bastante bien, lo cual es algo raro en Saturn. Mi opinión sobre Saturn siempre ha sido que destaca en gráficos 2D y tenia que haberse mantenido alejada del 3D.

Rayman

Sé que Rayman existe para muchas plataformas, y la versión PS1 se acerca mucho a la versión de Saturn, aunque la versión de Satur esta considerada ligeramente superior. Tiene animaciones adicionales entre las pantallas de carga, entre los niveles, y en ciertos jefes. También se dice que el sonido y la música son mejores en Saturn. Aparte de eso, el juego en sí mismo es prácticamente el mismo en todos los sistemas.

Arcade Box

April 1, 2018 By Brian Kim, Charles Park, and John Lee Juegos, ODROID-XU4, Mecaniqueo

Figura 1 – ODROID Arcade Box

Los ODROID tienen mejor rendimiento que las placas de la competencia, especialmente en lo que respecta al procesamiento de video, lo que significa que las placas ODROID son idóneas para ejecutar juegos, que es lo que hacen muchos usuarios de ODROID. Ya existen varios sistemas operativos de plataformas de juegos disponibles, como Lakka (**http://bit.ly/1NO8BBC**) y ODROID GameStation Turbo (**http://bit.ly/1ASFO5O**). Para poder disfrutar aún más de nuestras sesiones de juegos, hemos creado nuestro propio mando arcade con simples botones GPIO y joysticks, lo hemos llamamos ODROID Arcade Box. Hemos elegido un **ODROID-XU4** para este proyecto porque es el que tiene mejor rendimiento GPU de todos los dispositivos ODROID actuales. Este artículo describe cómo volver a crear el ODROID Arcade Box.

Figura 2 – Nuestro primer y simple prototipo

Requisitos

Figura 3 – Herramientas y componentes

Decidimos hacer el ODROID Arcade Box utilizando MDF (tablero de fibra de densidad media). El XU4 Shifter Shield también nos es útil para utilizar los pines de expansión del ODROID-XU4. Los Joysticks, botones y cables son los componentes de entrada, y se utilizó un SMPS (fuente de alimentación conmutada) como fuente de alimentación. Las herramientas usadas y la lista de componentes se enumeran a continuación:

- Panel MDF 12T
- 2EA 600×220
- 2EA 600×75
- 2EA 220×75
- Taladro
- Plegadora
- Desmontadores
- Cinta métrica
- Navaja multiuso
- Alicates de punta larga
- ODROID-XU4
- XU4 Shifter shield
- SMPS
- Extensores Ethernet, USB, HDMI
- Toma de corriente & Switch
- Bisagras 2EA
- Receptor de puerta
- Bases de goma 4EA
- Tornillos
- Botones 19EA
- Joystick 2EA
- Cables
- Terminales

ODROID Arcade Box necesita un total de 27 entradas (19 entradas para botones y 8 entradas para joysticks). Las 24 entradas digitales GPIO del ODROID-XU4 no son suficientes para cubrir las 27 entradas, de modo que creamos dos puertos ADC adicionales para los tres botones adicionales. Los valores de entrada ADC se basan en el voltaje de entrada. Los valores de entrada digital y analógica son procesados por el demonio de teclas GPIO, que se describen a continuación.

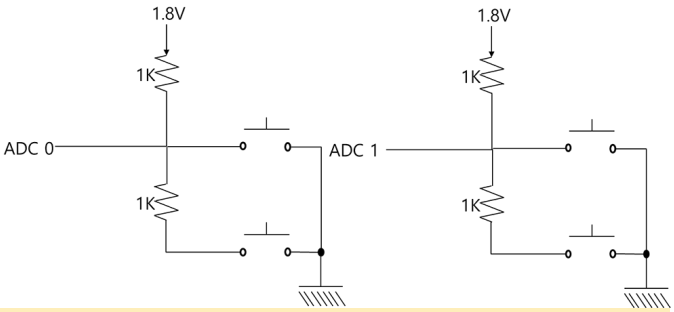


Figura 4 – Esquema de los puertos de expansión

Diseño y montaje

Los paneles del ODROID Arcade Box se deben diseñar y fabricar de forma que los botones y los joysticks estén bien ubicados. Elegimos MDF 12T considerado el precio y la durabilidad. El diseño se puede hacer con cualquier herramienta CAD que conozcas, como Google Sketch o SolidWorks. Aunque hay disponibles

muchas plantillas de diseño para paneles joypad, optamos por un diseño arcade japonés estándar.

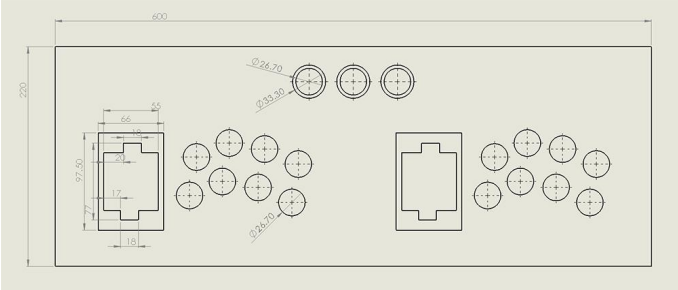


Figura 5 – Patrón de diseño del Joypad

El primer paso del montaje es fijar la lámina al panel MDF. Este paso era fácil, pero nos llevó algo más de tiempo que el resto. Después, insertamos los joysticks, la toma de corriente, el interruptor y los botones en el panel superior de MDF. Los extensores HDMI, Ethernet y USB se colocaron en la parte posterior del panel MDF. El siguiente paso fue montar cada panel MDF usando el taladro para hacer los agujeros, luego usamos los tornillos para sujetar los paneles.

Figura 6 – Diseño del ODROID Arcade Box montado

El último paso para montar el ODROID Arcade Box es cablear los pines de expansión del ODROID-XU4 hasta los componentes de entrada. En este proyecto diseñamos las entradas GPIO externas, tal y como se muestra en la Figura 5. Los botones Select y Temp están conectados a los puertos de expansión ADC, como se muestra en la Figura 3.

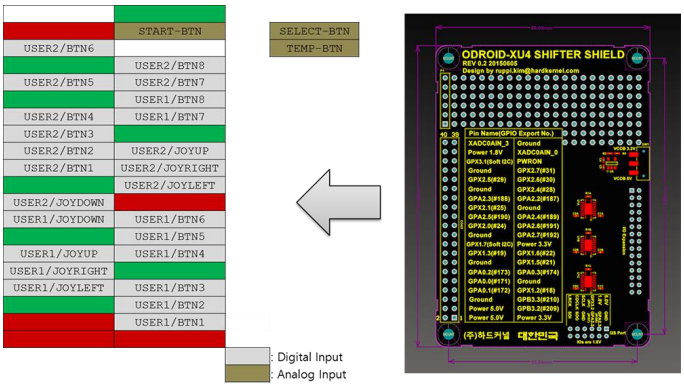


Figura 7: Asignaciones externas de GPIO para los botones y joysticks

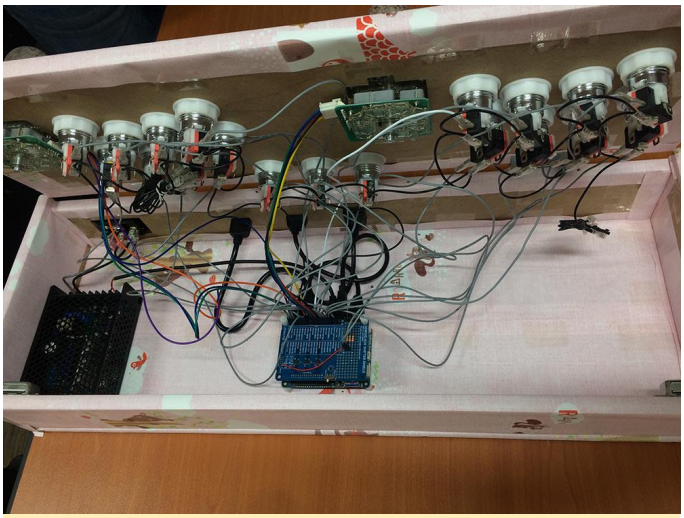


Figura 8 – Cableado del ODROID Arcade Box

Configuración del software

Desarrollamos un nuevo demonio de teclas GPIO llamado gpio_keyd (http://bit.ly/2ljOZKg). El demonio gpio_keyd puede asignar entradas GPIO y eventos de tecla usando uinput y wiringPi, que es una librería de acceso GPIO basada en pin. Está diseñado para que le sea familiar a las personas que han usado el sistema de cableado de Arduino. Aunque la librería wiringPi sólo admite Raspberry Pi, Hardkernel ofrece una versión de wiringPi para ODROID en su repositorio GitHub (http://bit.ly/1Eq3UpF). El módulo uinput es un módulo del kernel Linux que maneja el subsistema de entrada desde el área del usuario. Se puede usar para crear y gestionar dispositivos de entrada desde una aplicación.

Elegimos ODROID GameStation Turbo (http://bit.ly/1ASFO5O) como plataforma de software para nuestro ODROID Arcade Box, que tiene el uinput integrado. Debes asegurarte de que el archivo del

dispositivo uinput exista en el sistema operativo elegido, ya que algunos no tienen dispositivos uinput.

```
$ ls /dev/uinput
```

Si tu sistema operativo no tiene un archivo de dispositivo /dev/uinput, entonces será necesario volver a compilar e instalar un nuevo kernel con la opción de configuración INPUT_UINPUT definida. La página de la Wiki http://bit.ly/1YIToBI describe cómo compilar e instalar la imagen del kernel desde el código fuente.

```
$ make menuconfig

 Device Drivers -> Input device support
 -> Generic input layer
 -> Miscellaneous device
 -> User level driver support <*>
```

Ten en cuenta que wiringPi debe estar instalado antes de instalar gpio_keyd. En la imagen ODROID GameStation, los comandos sudo deben ejecutarse como root, porque la cuenta “odroid” no está designada como usuario sudo.

```
$ git clone
https://github.com/hardkernel/wiringPi.git
$ cd wiringPi
$ sudo ./build
```

Descarga el código fuente de gpio_keyd, que está disponible desde nuestro repositorio GitHub. Los métodos de compilación e instalación de gpio_keyd son muy simples:

```
$ git clone
https://github.com/bkrepo/gpio_keyd.git
$ cd gpio_keyd
$ make
$ sudo make install
```

El script gpio_keyd hace referencia a /etc/gpio_keyd.conf como valor por defecto para GPIO y la información de asignación de teclas. El archivo de configuración ha sido modificado para las 27 entradas del ODROID Arcade Box. Algunas teclas ya se utilizan en el emulador del juego, así que tuvimos que cambiar la configuración de teclas del emulador para evitar conflictos entre el emulador y las teclas de entrada GPIO. Ten en cuenta que el campo en el archivo de configuración hace referencia al número WiringPi, no al GPIO ni al número de pin (**http://bit.ly/2lbzPIB**).

Ejemplo de archivo de configuración para las 27 entradas: /etc/gpio_keyd.conf

```
# Digital input
#
# User 1
KEY_LEFT digital 15 0
KEY_RIGHT digital 1 0
KEY_UP digital 4 0
KEY_DOWN digital 16 0
KEY_A digital 2 0
KEY_S digital 3 0
KEY_D digital 30 0
KEY_F digital 21 0
KEY_Z digital 8 0
KEY_X digital 9 0
KEY_C digital 7 0
KEY_V digital 0 0
# User 2
KEY_BACKSLASH digital 12 0
KEY_SLASH digital 13 0
KEY_SEMICOLON digital 14 0
KEY_LEFTBRACE digital 5 0
KEY_Y digital 26 0
KEY_U digital 27 0
KEY_I digital 22 0
KEY_O digital 23 0
KEY_H digital 6 0
KEY_J digital 10 0
KEY_K digital 11 0
KEY_L digital 31 0

# Analog input
#
KEY_B analog 0 0
KEY_N analog 0 2045
KEY_M analog 1 2045
```

Es conveniente ejecutar el demonio gpio_keyd en cada arranque para ODROID Arcade Box.

```
/etc/init.d/gpio_keyd
#! /bin/sh
### BEGIN INIT INFO
# Provides: gpio_keyd
# Required-Start: $all
# Required-Stop:
```

```
# Default-Start: 2 3 4 5
# Default-Stop:
# Short-Description: Run /usr/bin/gpio_keyd if
it exist
### END INIT INFO

PATH=/sbin:/usr/sbin:/bin:/usr/bin

. /lib/init/vars.sh
. /lib/lsb/init-functions

do_start() {
 if [ -x /usr/bin/gpio_keyd ]; then
 /usr/bin/gpio_keyd -d
 ES=$?
 [ "$VERBOSE" != no ] && log_end_msg $ES
 return $ES
 fi
}

case "$1" in
start)
do_start
;;
restart|reload|force-reload)
echo "Error: argument '$1' not supported" >&2
exit 3
;;
stop)
killall gpio_keyd
exit 0
;;
*)
echo "Usage: $0 start|stop" >&2
exit 3
;;
Esac
$ sudo chmod +x /etc/init.d/gpio_keyd
$ sudo update-rc.d gpio_keyd defaults
$ sudo reboot
```

En los comandos anteriores, el script gpio_keyd se ejecuta como demonio utilizando la opción “-d”. El uso de gpio_keyd se puede comprobar con la opción “-h”. Vuelve a verificar las teclas utilizadas por el juego o el emulador, luego configura correctamente gpio_keyd. Ahora es cuando estás listo para jugar y disfrutar de tus juegos con tu nuevo ODROID Arcade Box.

Figura 9 – El Rey de los Luchadores 98, John vs. Brian

Servidor Nextcloud: Creando un Almacenamiento Conectado en Red (NAS) con un ODROID-HC2

April 1, 2018 By Autonomous Linux, Tutoriales, ODROID-HC2

Esta guía te enseñará a configurar un NAS (Almacenamiento conectado en red) en un ordenador de placa reducida ODROID-HC2. No hay ninguna razón por la cual no pueda funcionar en un HC1 o XU4, sin embargo, el XU4 usa USB3 en lugar de SATA. La guía está escrita pensando en aquellas personas que no tienen experiencia con Linux, pero sí una mínima experiencia con el montaje de ordenadores. La he escrito porque mi experiencia con Linux era mínima y me costó bastante trabajo averiguar cómo configurarlo todo, e incluso tuve que leerme unas cuantas docenas de guías.

Listado de componentes

- ODROID-HC2 : Home Cloud Two
- Fuente de alimentación 12V/2A con clavija US para el ODROID-HC2
- Kit módulo ODROID-USB-UART
- Tarjeta de memoria de 32GB clase 10 SDHC UHS-I SanDisk, hasta los 80 MB, gris/negro (SDSDUNC-032G-GN6IN)
- Cable de conexión Ethernet RJ45 Cat-6 – 3 pies (0,9 metros)
- Carcasas ODROID-HC2 (negra)
- Unidad de disco duro WD Red NAS de 2TB – 5400 RPM Clase SATA 6 Gb/s 64MB Cache 3.5 Inch – WD20EFRX

El HC2 es superior al Raspberry Pi 3 ya que tiene conexión ethernet gigabit, una conexión de disco duro SATA, una CPU más rápida de 2 Ghz y 2 GB de memoria. Está diseñado específicamente para ser un dispositivo de almacenamiento conectado en red o un pequeño servidor. La Raspberry Pi 3 es superior en cuanto a la cantidad de soporte disponible de los fabricantes y de la comunidad. Opté por el HC2 sobre el HC1 porque las unidades de 3,5 pulgadas de alta capacidad son más económicas que las unidades de

2,5 pulgadas. El HC1 es más eficiente en cuanto a consumo, si esto es una cuestión que sueles tener en cuenta. El HC2 no tiene salida HDMI, de modo que todo debe configurarse antes de que la imagen se grabe en la tarjeta MicroSD, a través de una conexión remota VNC o a través de una conexión en serie con líneas de comandos. En esta guía realizaremos la configuración con una conexión en serie, pero vale la pena configurar un servidor VNC, ya que Linux es más fácil de usar con una GUI si no conoces muy bien los comandos de consola.

El HC2 necesita una fuente de alimentación de 12 voltios/2 amperios, una tarjeta MicroSD y un disco duro SATA de 3.5 pulgadas. Usé un disco duro Toshiba de 3TB que tenía por ahí. El HC2 con la tapa solo admite discos duros de hasta 27 mm de altura.

Además, si los tornillos incluidos son demasiado cortos, es posible que necesite otros tornillos para fijar el disco duro a la estructura de aluminio del HC2.

También necesitarás un cable ethernet para conectarte a tu router y a Internet. Además, necesitará un ODROID-USB-UART para conectarte al puerto serie del HC2.

Hazte con una carcasa HC2 por 5$ para cerrarlo todo una vez hayas terminado.

Montaje

El montaje es extremadamente simple. Coloca el disco duro en la parte inferior del HC2 y lo deslizas hacia el puerto SATA. Sujetando el disco duro y la estructura de aluminio, le das la vuelta a las dos piezas para asegurarte de que el disco duro no haga peso sobre el conector SATA. Una vez que lo tengas boca abajo, usa los tornillos para fijar la parte inferior del disco duro al HC2. Dale la vuelta de nuevo y conecta el cable ethernet desde HC2 a tu router. Conecta el USB-UART al puerto serie del HC2. Conecta el extremo USB a tu ordenador. No conectes la fuente de alimentación todavía.

Software

Descargue la versión ubuntu-16.04.3-4.9-mate-odroid-xu4-20171025.img.xz de ODROID desde **https://odroid.in/ubuntu_16.04lts**. Ubuntu es una distribución Linux gratuita y muy estable. Mientras se estaba descargando, descargue también Etcher: **https://etcher.io/**. Etcher es un software con el que grabas la imagen de Ubuntu en su tarjeta MicroSD. Coloca tu tarjeta MicroSD en el adaptador MicroSD de tu ordenador e inicia Etcher. Selecciona la imagen de Ubuntu y la tarjeta MicroSD y luego haga clic en "Flash". Le llevará unos minutos hasta que finalice el proceso. Cuando termine, Windows mostrará un mensaje para que formatees la unidad antes de usarla. Haga clic en "Cancelar".

Coge tu tarjeta MicroSD grabada e insértala en la ranura MicroSD del HC2.

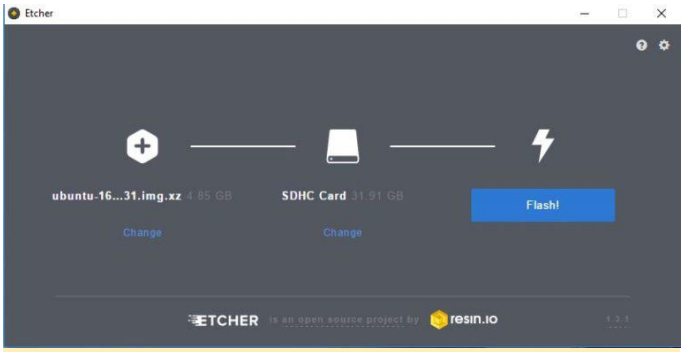


Figura 1 – Grabado en una unidad con Etcher

Descarga los drivers para USB-UART de aquí: **https://goo.gl/opafDn**

Extrae e instala los drivers. Una vez que los controladores estén instalados, descubrirás que aparece como un puerto COM dentro del administrador de dispositivos.

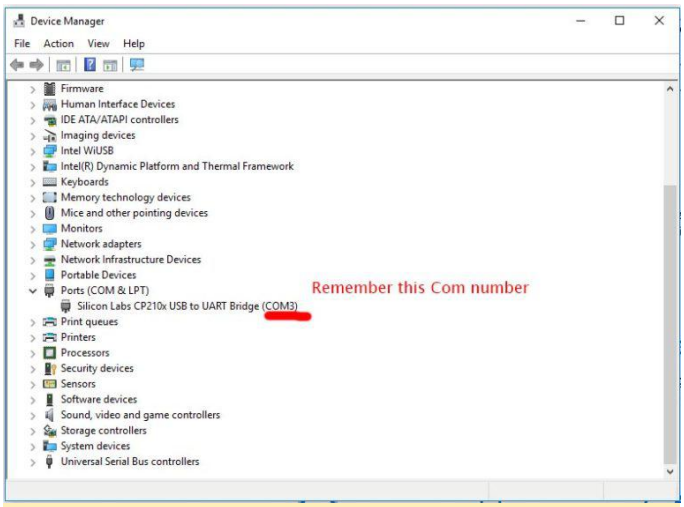


Figura 2 – Toma nota del puerto COM del dispositivo USB-UART

A continuación, descarga e instala PuTTY desde **http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe**. PuTTY es un programa gratuito de consola

que utilizaremos para enviar comandos al HC2. PuTTY te solicitará la configuración de la conexión. Introduce el número de COM que aparecía en el administrador de dispositivos, 115200 en la velocidad y comprueba que "Serial" es el tipo de conexión seleccionada. Guarda tu configuración por si necesitas usarla en el futuro.

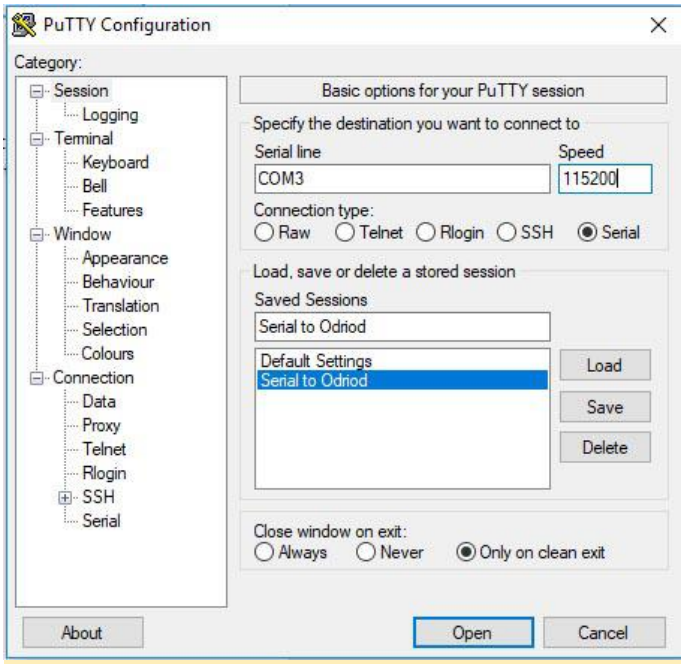


Figura 3 – Configuración de PuTTY

Configuración

Haga clic en "Open" en PuTTY y conecta la fuente de alimentación al HC2. La ventana de la consola mostrará todos los mensajes del arranque y finalmente, aparecerá un aviso de inicio de sesión. La información de inicio de sesión por defecto es:

Usuario: odroid Contraseña: odroid

Credenciales de acceso root:

Ususario: root Contraseña: odroid

Vamos a iniciar sesión e instalar primero las actualizaciones. El comando sudo es el comando de "Super Usuario" para ejecutar comandos como Admin o Root. Escribe los siguientes comandos:

```
$ odroid login: odroid
$ Password: odroid

$ sudo apt update
$ sudo apt upgrade
$ sudo apt dist-upgrade
$ sudo apt install linux-image-xu3
```

Cuando instales linux-image-xu3, aparecerá un aviso indicándote que la actualización del kernel es peligrosa y si quieres abortar la actualización. Selecciona "NO". A veces aparecerán errores de los archivos que están siendo utilizados en el arranque. Si lo hace, introduce el siguiente comando y el proceso continuará desde donde lo dejó:

```
$ sudo reboot
```

Particionar el disco duro

Regresa y escribe:

```
$ sudo apt-get install lshw
$ sudo lshw -C disk
```

Esto mostrará una lista de todas las unidades de almacenamiento conectadas

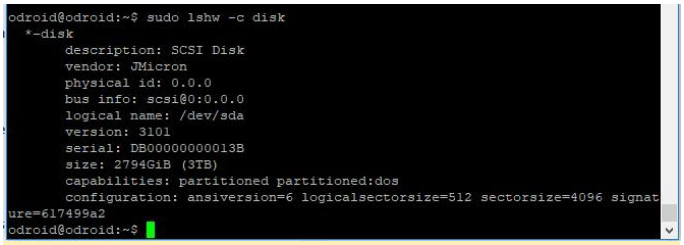


Figura 4 – Listado de unidades de almacenamiento conectadas

En este caso, necesitarás el nombre lógico del disco duro "/dev/sda". A continuación, peticionaremos el disco:

```
$ sudo fdisk /dev/sda
```

Crea un nuevo tipo de partición presionando "n" para la nueva partición, "p" para la partición primaria, luego "1". Para el inicio y final de la partición, presiona INTRO dos veces para usar los valores por defecto de la unidad completa, luego introduce "w" para escribir la tabla de particiones y salir de fdisk.

Formatear el disco duro

En esta sección, vamos a formatear el disco duro como ext4. Si alguna vez tienes que recuperar los datos de la unidad, debes saber que ext4 no es compatible de forma nativa con Windows. Esto no es un problema importante ya que Ubuntu/Nextcloud es capaz de gestionar todas las operaciones de lectura/escritura. Si necesitas que el disco sea directamente compatible con Windows o Mac,

formatéalo como vfat (fat32). Ten en cuenta que vfat solo puede manejar archivos de hasta 4GB y particiones de hasta 2TB. Crea varias particiones que trabajen en torno a ese límite de tamaño.

Escriba el siguiente comando para formatear la unidad:

```
$ sudo mkfs -t ext4 /dev/sda1
```

Cuando finalice el formateo, configuraremos el disco duro para que se monte en el arranque. Vamos a crear un directorio en Ubuntu para montar el disco duro. Yo he nombrado el mío como "SATAHD" pero puedes usar cualquier otro nombre:

```
$ sudo mkdir /media/SATAHD
```

A continuación, necesitamos el UUID del disco duro. Cuando un disco duro está particionado, se genera un UUID y se asigna a la partición. Nuestro disco duro se identifica físicamente como "/dev/sda1", siendo "sda" la unidad y "1" la primera partición. Si el disco duro fuera de intercambió, /dev/sda1 sería la primera partición de la nueva unidad. Sin embargo, el UUID de la partición es único para esta partición. Si quieres que Ubuntu monte automáticamente cualquier unidad que incluiste como SATAHD, puede usar /dev/sda1 en lugar del UUID:

```
$ sudo blkid
```

Busca el resultado de "/dev/sda1 uuid =", luego copia el UUID en el bloc de notas o en un papel.

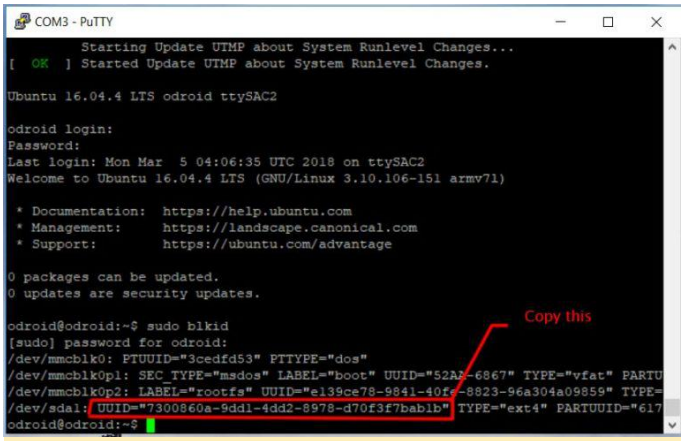


Figura 5 – Toma nota del UUID de la unidad

A continuación, vamos a configurar la unidad para que se monte automáticamente en el arranque. El comando "sudo nano" es un editor de texto basado en terminal:

```
$ sudo nano -Bw /etc/fstab
```

Usa las teclas de flecha para ir al final del archivo y añadir la siguiente línea:

```
UUID=7300860a-9dd1-4dd2-8978-d70f3f7bab1b
/media/SATAHD EXT4 defaults 0 2
```

Usa el UUID que copiaste en el anterior paso en lugar del que yo usé. "/Media/SATAHD" es el punto de montaje y el directorio que hemos creado anteriormente. ext4 es el sistema de archivos que usamos para formatear la unidad. Si usaste algo que no fuera ext4, reemplázalo por sistema de archivos correcto.

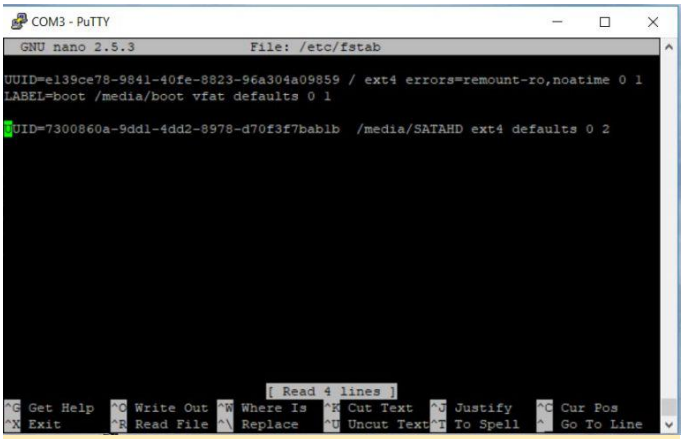


Figura 6 – Editando /etc/fstab para montar automáticamente la unidad durante el arranque

Mantén presionado CTRL + X para salir. Presiona "Y" para guardar los cambios y Enter para guardar el archivo anterior, luego reinicia el ordenador:

```
$ sudo reboot
```

Instalar Nextcloud

Vamos a instalar Nextcloud con un paquete snap. Si quieres instalar y configurar NextCloud manualmente, tienes disponible la documentación en http://goo.gl/DS5ZjY. Escriba el siguiente comando para instalarlo:

```
$ sudo snap install nextcloud
```

Nextcloud tiene una lista de dominios o direcciones autorizadas. Por defecto, la primera dirección IP con la que acceda a Nextcloud se agregará a esa lista.

Después de esto, debes agregar las direcciones manualmente. Prefiero que mi router automáticamente proporcione la misma dirección IP al HC2 utilizando la dirección MAC. Si tu router no admite la asignación de direcciones IP de este modo, puedes configurar tu dirección IP para que sea estática y se asigne automáticamente en lugar de usar DHCP.

Para ver la dirección IP eth0 de tu HC2, escribe el siguiente comando:

```
$ ifconfig -a
```

Abre tu navegador y escribe la dirección IP de tu HC2. Serás recibido con la pantalla “Make a new admin account”. Introduce un nombre de usuario y una contraseña. En la siguiente pantalla, cierre la ventana emergente “Welcome to Nextcloud”. Haz clic en el icono en forma de engranaje situado en la esquina superior derecha. Selecciona “apps”. Desplázate hacia abajo en la lista hasta que encuentre “External storage support”. En el lado derecho, haga clic en el botón “Enable” para activarlo.

Haga clic de nuevo en el icono de engranaje en la esquina superior derecha. Selecciona “Admin.” En la barra lateral izquierda, selecciona “External Storage”. Haz clic en la casilla desplegable “Add Storage” y selecciona “Local”. Completa la ubicación como “/ media / SATAHD” y pincha en la casilla de verificación de la derecha para guardar los cambios.

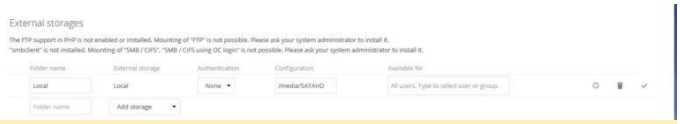


Figura 7 – Añadiendo el almacenamiento en Nextcloud

Tu disco duro aparecerá ahora en la pantalla de inicio como “Local”. Si necesitaras modificar el archivo config.php del servidor, escribe:

```
$ sudo nano
/var/snap/nextcloud/current/nextcloud/config/c
onfig.php
```

La configuración con la que probablemente tendrás que lidiar es con la de los dominios de confianza, en el caso de que cambie tu dirección IP. Para mayor seguridad, deberías cambiar la contraseña de la cuenta ODROID y root. Escribe:

```
$ sudo passwd
$ and
$ passwd
```

Puedes habilitar HTTPS, pero a los navegadores web no les gustarán los certificados autofirmados. Hazlo bien y usa Let’s Encrypt. También puede configurar tu Nextcloud para que sea accesible a través de un DNS dinámico, pero ese será otro tutorial. Para comentarios, preguntas y sugerencias, visite el artículo original en **http://autonomousdev.net/index.php/2018/03/06/odroid-hc1-2-ubuntu-nas-with-nextcloud-setup-guide/**.

Lecturas recomendadas

https://help.nextcloud.com/t/adding-a-new-trusted-domain/26 https://github.com/nextcloud/nextcloud-snap
https://docs.nextcloud.com/server/13/admin_manual/

Controlar Cualquier Dispositivo Eléctrico con un ODROID-C2: Un Proyecto de Muestra

April 1, 2018 By Miltiadis Melissas (miltos) ODROID-C2, Mecaniqueo

Siempre ha sido un sueño del siglo 20 que existiera una era en la que cualquier aparato eléctrico del hogar pudiera controlarse con un simple clic desde cualquier dispositivo conectado a internet, como un PC, tablet o TV inteligente, y desde cualquier lugar. Esta era ha llegado y hoy te presentaremos una forma de controlar cualquier dispositivo eléctrico desde cualquier otro dispositivo que tenga acceso a internet.

Nosotros usaremos una lámpara como ejemplo, pero ésta podría ser fácilmente sustituido por un frigorífico, una lavadora o una cafetera eléctrica, por ejemplo. No obstante, tuvimos que hacer una simplificación, que es electrificar la lámpara con una corriente de 12V en lugar de 220V, principalmente por razones de seguridad. Advertimos a los usuarios de esta guía que vayas hacer lo mismo: ¡Es muy fácil exponerse a descargas eléctricas peligrosas!

El módulo relé que utilizamos con el **ODROID-C2** en este proyecto se puede conectar fácilmente a una fuente de alimentación de 220V, que controla cualquier dispositivo eléctrico (hasta 10A). Los usuarios experimentados pueden intentar trabajar con este voltaje, asegurándose siempre de tomar todas las precauciones en términos de seguridad. Vamos a seguir profundizando en el ilimitado potencial del ODROID-C2.

Requisitos de hardware

- ODROID-C2 (http://bit.ly/1oTJBya)
- Módulo placa relé de 5V con canales 1/2/4/8 ARM AVR DSP PIC (http://ebay.eu/2ncLWD8)
- Lampara JM/84211 3W 3000K 12V o cualquier otra de similares caracterisiticas
- Cables Dupon hembra a hembra, macho a macho (http://ebay.eu/2mDWf6Q)

- 4 pilas Li-ion 3.7V 2300mAh LS 14500 (http://ebay.eu/2m0F7EI)

Requisitos de Software

- Ubuntu 16.04 v2.0 de Hardkernel (http://bit.ly/2cBibbk)
- Python 2.3 o 3.3 preinstalado con Ubuntu
- Librería WiringPi para controlar los pines GPIO del ODROID-C2. Puede aprender cómo instalar esto en http://bit.ly/2ba6h8o
- Editor de HTML gratuito CoffeeCup (http://bit.ly/2lCxgB8)
- PuTTY * – Vamos a necesitar conectarnos a nuestro ODROID-C2 a través de SSH, PuTTY es el cliente perfecto para hacer esto (http://bit.ly/2kFVngX)
- FileZilla – Vamos a necesitar una forma de transferir archivos al ODROID-C2 usando SFTP, que es FTP sobre SSH (http://bit.ly/1gEw9op)

Conectándolo todo

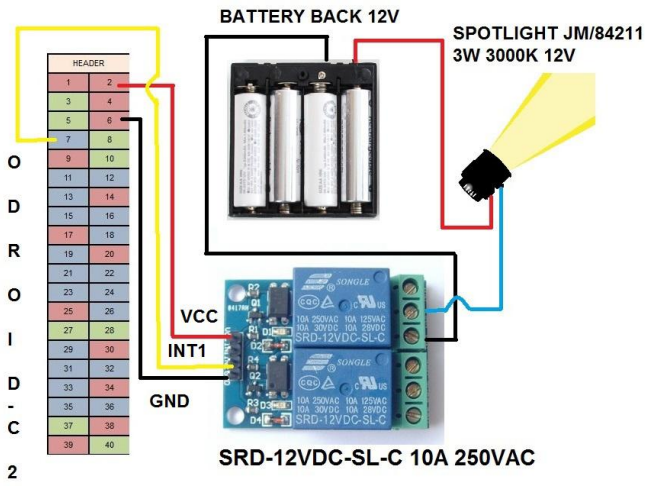


Figura 1 – Diagrama por bloques

El diseño de este proyecto es muy simple, y el relé Songle juega un papel esencial. Hemos conectado el GND del relé Songle al pin6 del ODROID-C2 (GND). El pin VCC del relé está conectado directamente al pin2 de ODROID-C2, que proporciona 5V a este circuito y electrifica la resistencia eléctrica del relé. Finalmente, el pin INT1 del relé está conectado al pin7 de nuestro ODROID, que es el pin que realmente controla el relé, que básicamente enciende (ON) y apaga (OFF) el dispositivo. En el otro lado del relé Songle, hay un simple interruptor que hemos conectado a la lámpara a través de la batería o la red eléctrica. Echa un vistazo al esquema de la Figura 1 para que te hagas una idea de cómo está montado todo el circuito. Como referencia técnica de los pines ODROID-C2, hemos utilizado el excelente Mapa de PIN proporcionado por Hardkernel en http://bit.ly/2aXAlmt. De acuerdo con este mapa, pin2 = 5V, pin6 = GND y pin7 = GPIOX.BIT21 (Pin de entrada/salida de aplicación general). Todas las conexiones se realizaron usando cables Dupon hembra-hembra, macho-macho o macho-hembra. Ahora que nuestro hardware está listo, veamos cómo compilar el software y ponerlo a funcionar todo.

Diseñando una simple página web

Usamos el editor gratuito CoffeeCup HTML para diseñar una simple página web HTML para controlar la lámpara. En esta página, añadiremos las imágenes de dos botones para poder controlar la lámpara, los botones de ENCENDIDO y APAGADO. Consulta la Figura 2 para ver esta página. Todo el proyecto se controla utilizando una IU web. Para lograr esto, hemos hipervinculados esos botones a los correspondientes scripts Python songleon.py y songleoff.py que controlan el ENCENDIDO y APAGADO de la lámpara. Las instrucciones sobre cómo escribir esos programas en Python están detalladas en una sección que se muestra más adelante.

Figura 2 – Página web

Cuando finalmente diseñes tu sitio web, asegúrate de que tu página de inicio se llame index.php y no index.html, simplemente para hacer las cosas más uniformes. Sin embargo, solo vamos a usar dos scripts PHP, songleon.php y songleoff.php, para controlar la lámpara. El código Python y PHP que necesitamos escribir es muy simple y está muy bien documentado.

Instalando el servidor

Para utilizar el ODROID-C2 como servidor web en este proyecto, tenemos que instalar todos los componentes de software de servidor necesarios.

Además, puesto que queremos un simple servidor HTML, instalaremos Apache con soporte PHP (lenguaje de programación del servidor) en el ODROID-C2. Los siguientes pasos se pueden realizar con PuTTY. El acceso al ODROID-C2 con este cliente SSH está bien documentado. Todo lo que necesitas saber es la dirección IP del ODROID-C2.

El servidor Apache es el servidor web más utilizado hoy día. Aquí tienes cómo instalar Apache con soporte PHP:

```
odroid@odroid:~# sudo apt-get install apache2
php libapache2-mod-php
```

Cuando se te solicite que continúes, introduce “y” para “sí”. Luego, activa e inicia Apache:

```
odroid@odroid:~# systemctl enable apache2
 odroid@odroid:~# systemctl start apache2
 odroid@odroid:~# systemctl status apache2
```

Probando Apache

Abre tu navegador web y navega hasta http://localhost/ o http:///. Esta es la dirección de tu ODROID-C2 en tu red local. Puedes localizarla simplemente escribiendo:

```
odroid@odroid:~# ifconfig
```

Verás una página similar a la que se muestra en la Figura 3.

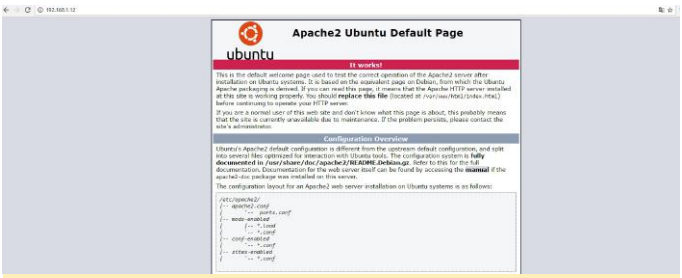


Figura 3 – Prueba de Apache

Probando PHP

Para probar PHP, crea un archivo testphp.php de muestra en la carpeta raíz de documento de Apache:

```
odroid@odroid:~# sudo nano
/var/www/html/testphp.php
```

Agrega las siguientes líneas y guarda el archivo:

Reinicia el servicio Apache.

```
$ sudo systemctl restart apache2
```

Navega a http: //dirección-ip-servidor/testphp.php. Se mostrarán todos los detalles de PHP, como la versión, la fecha de compilación y los comandos, por nombrar algunos. Tal y como puede observar en la Figura 4.

Figura 4 – Información de PHP

Instalando FTP

Instala un servidor FTP como vsftpd usando el siguiente comando:

```
$ sudo apt-get install vsftpd
```

Edita el archivo de configuración FTP escribiendo:

```
$ sudo nano /etc/vsftpd.conf
```

y realiza los siguientes cambios:

- Presiona ctrl+W y busca anonymous_enable=YES, cámbialo por anonymous_enable=NO
- Elimina el signo # que hay delante de local_enable=YES
- Elimina el signo # que hay delante de write_enable=YES
- Vete al final del archivo y añade force_dot_files=YES
- Presiona ctrl+X para salir, introduce “y” para guardar y presiona INTRO para confirmar

Luego, reinicia vsftpd:

```
$ sudo service vsftpd restart
```

Publicando en la web

De momento, deberías tener un sitio web con el que poder transferir datos al ODROID-C2. Una vez que hayas realizado todos los pasos anteriores y hayas comprobado que puedes ver tu sitio web en otro ordenador, podemos continuar para que la página

web encienda nuestro foco utilizando el Módulo placa relé de 5V ARM AVR DSP PIC.

Dentro del directorio de tu sitio web, crea un nuevo archivo PHP llamado songleon.php que contenga el siguiente fragmento de código, luego guarda el archivo:

A continuación, crea una carpeta en el directorio del sitio web llamado “scripts”, luego cree una subcarpeta dentro de ésta llamada “lights”, y dentro crea un nuevo archivo llamado songleon.py. Este será el script python que encenderá nuestra lámpara. Dentro de este archivo añade el siguiente código y luego guarda el archivo:

```
import wiringpi2 as odroid
 odroid.wiringPiSetup()
 odroid.pinMode(7,1)
 odroid.digitalWrite(7,0)
```

Regresa a tu página web en modo de diseño/edición y asegúrate de que el hipervínculo para tu botón “ON” está vinculado con songleon.php. Ahora, cuando hagas clic en el botón, el script songleon.php ejecutará el script python songleon.py, lo que tendrá como resultado que la lámpara se encienda. Por último, haremos lo contrario, apagar la lámpara.

Dentro del directorio del sitio web, crea un nuevo archivo llamado songleoff.php. Dentro de este archivo, introduce el siguiente fragmento de código y guárdalo:

```
 <!--?php system("echo odroid | sudo -S python
/var/www/html/scripts/lights/songleoff.py");
header( 'Location: 'index.php' ) ; ?-->
```

Nuevamente, asegúrate de que la ruta del archivo sea la misma, para que todo funcione. Además, configura tus reglas de redireccionamiento para redirigirlas a la página que quieres. Luego, crea un nuevo archivo en la carpeta scriptslights llamado sognleoff.py. Dentro de este, introduce el siguiente código, después guarda el archivo:

```
import wiringpi2 as odroid
odroid.wiringPiSetup()
odroid.pinMode(7,1)
odroid.digitalWrite(7,1)
```

Añade un hipervínculo a songleoff.php para tu botón “OFF”, el cual debería hacer que tu lámpara se apague. ¡Ahora tiene un sitio web con el que puedes controlar tus luces!

Transfiriendo archivos a Apache

Es muy fácil iniciar sesión en tu servidor web Apache ODROID-C2 con Filezilla tan simple como saber la dirección IP del ODROID-C2. Si ya ha iniciado sesión con SSH en tu ODROID-C2 con PuTTY, puedes localizarla escribiendo:

```
odroid@odroid:~# ifconfig
```

Debes introducir el nombre de usuario y contraseña “odroid” y “odroid”. Serás redirigido inmediatamente al sistema de archivos raíz del ODROID-C2. Desde allí, dirigiere a la carpeta /var/www/html/, dentro de este directorio copia los archivos de tu disco local. Aquí tienes todos los archivos y carpetas que debes copiar desde ese directorio local:

- index.php
- songleon.php
- songleoff.php y finalmente la carpeta /scripts/lights/ con
- songleon.py y songleoff.py

Ahora has terminado con la parte principal del proyecto. Un consejo final: para hacer posible la ejecución de los scripts que controlan la lámpara (songleon.py y songleoff.py), debes cambiar los permisos/derechos de todos los archivos y carpetas que hemos mencionado anteriormente. Recomendamos para este proyecto en concreto proporcionarles acceso completo con privilegios de lectura, escritura y ejecución para root:

```
$ sudo chown 755 /var/www/html/index/php
```

Además, debes modificar el archivo sudoers con el editor nano:

```
$ sudo nano /etc/sudoers
```

Introduce tu contraseña para cualquier modificación y agrega la siguiente línea:

```
www-data ALL=(ALL) NOPASSWD ALL after this
line: %sudo ALL=(ALL:ALL) ALL
```

Probando las aplicaciones

Veamos si todo funciona correctamente. Desde tu PC de escritorio, ordenador portátil o tablet, navega hasta la dirección IP de tu ODROID-C2 con el navegador y haz clic en el botón “ON”. ¿La lámpara ilumina tu habitación? Ahora sitúate sobre el botón “OFF”. Haz clic y verás como la lámpara se apaga. ¡Lo hemos conseguido!

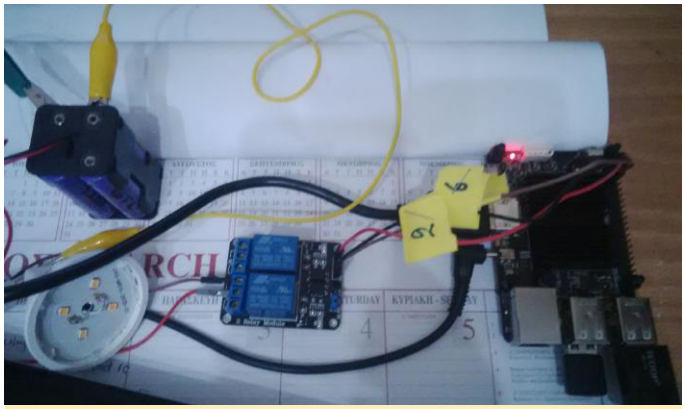


Figura 5 – Instalación del Hardware

Notas finales

Podríamos continuar asesorándote sobre cómo controlar remotamente cualquier dispositivo eléctrico desde la oficina, durante un viaje o si te encuentra con una emergencia. Esta cuestión no es nada difícil ahora que tienes el circuito básico funcionando en su red local, pero ten en cuenta que dicho paso conlleva un cierto riesgo de seguridad. Los hackers pueden estar interesados en controlar tu servidor aprovechando una contraseña débil, un puerto que es usado con frecuencia o configuraciones del router incorrectas. Por esta razón, te recomendamos que hagas segura tu red e incluso cambies la contraseña del usuario ODROID-C2 a una más robusta. Ahora tienes los conocimientos que necesitas para crear algo innovador e interesante para ti y tus colegas.

Buscadores, Mineros y 49 – Parte 2: Minería GPU-CPU Dual en el ODROID-XU4/MC1/HC1/HC2

April 1, 2018 By Edward Kisiel (@hominoid) Linux, Tutoriales

En un artículo del mes pasado pusimos en práctica la doble minería GPU-CPU en el Odroid XU4/MC1/HC1/HC2. Este mes pondremos al corriente a la comunidad de cómo van las mejoras en este tema y discutiremos algunos ajustes básicos de la GPU. Se ha completado la eliminación de todas las dependencias AMD OpenCL y el ensamblador INTEL para los kernels OpenCL y algoritmos de criptografía para sgminer-arm. Genesis Mining también ha lanzado recientemente una nueva versión de sgminer-gm 5.5.6. Estos cambios han sido incorporados al sgminer 5.5.6-ARM-RC1 recientemente lanzado. Aquí tienes un breve resumen de los kernels y algoritmos de criptografía que han sido modificados para Odroid y los resultados de las pruebas.

ocl/build_kernel.c

algorithm/cryptonight.c

– Optimizaciones del ensamblador INTEL

algorithm/neoscrypt.c

– Optimizaciones de la arquitectura AMD

kernel/cryptonight.cl

– Extensiones AMD OpenCL

kernel/equihash.cl

– Extensiones AMD OpenCL y optimizaciones de la arquitectura AMD

kernel/ethash.cl

– Extensiones AMD OpenCL

kernel/ethash-genoil.cl

– Optimizaciones de la arquitectura AMD

kernel/ethash-new.cl

– Optimizaciones de la arquitectura AMD

kernel/lyra2re.cl

– Extensiones AMD OpenCL

kernel/lyra2rev2.cl

– Extensiones AMD OpenCL

kernel/whirlpoolx.cl

– Optimizaciones de la arquitectura AMD

kernel/wolf-aes.cl

– Extensiones AMD OpenCL

kernel/wolf-skein.cl

– Extensiones AMD OpenCL

Debían tomarse decisiones sobre algoritmos de monedas muy concretas y kernels OpenCL que tenían configuraciones específicas de arquitectura (no extensiones AMD) como se indica. El 70% de los núcleos OpenCL comparten una o más de las mismas extensiones AMD OpenCL, que fueron modificadas y probadas con el kernel cryptonight, que también usa 2 kernels auxiliares OpenCL (wolf-aes.cl y wolf-skein.cl). Parece que ethash-new.cl no se usa para ninguna moneda dejando 2 sin probar, en cualquier caso, whirlpoolx.cl y ethash-genoil.cl. El resto solo tenían optimizaciones de arquitectura AMD y/o Nvidia que se eliminaron. El planteamiento más conservador ha sido usado en determinadas modificaciones para que se ejecutaran en una amplia gama de GPU actuales y futuras, pero siempre hay espacio para errores técnicos y humanos. La implementación de sgminer-arm debe ser independiente de la CPU y GPU, lo que aumenta la posibilidad de añadir algunas optimizaciones de ARM-Mali basadas en arquitecturas específicas (ARMv7, ARMv8, Mali-T628, Mali-T860) en el futuro.

Afinando la GPU

Cuando intentas averiguar por primera vez cuáles son las configuraciones para una moneda que no has extraído, sueles empezar de un modo muy conservador con todas las configuraciones y vas aumentado usando el método de prueba y error hasta que comienzan a aparecer fallos o el rendimiento empieza a caer. Aquí tienes algunos ajustes que son un buen punto de partida para empezar:

```
./sgminer -k algorithm -o
stratum+tcp://your.pool.com:3333 -u user -p
password -I 3 -w 32 -d 0,1 --thread-
concurrency 8192
```

Ten en cuenta que en todos los SOC de ARM-Mali, la GPU comparte la memoria principal con la CPU, de modo que también existe una dinámica entre las dos cuando realices minería dual. Es por eso que por lo general se pierde algo de rendimiento CPU/GPU dual en comparación con la minería únicamente sobre la CPU.

Cuando inicias sgminer-arm y la configuración es errónea o excesiva para la GPU, puede manifestarse de muchas y diferentes formas. El kernel OpenCL puede bloquearse, colgarse o indicar diversos mensajes de error. A continuación, se muestra el típico error de un problema de reajuste de la GPU. No es posible compilar el kernel OpenCL.

```
[19:28:21] Error -6: Creating Kernel from
program.  (clCreateKernel)
[19:28:21] Failed to init GPU thread 1,
disabling device 1
```

Otro mensaje de error muy común es que el kernel OpenCL esté intentando asignar más memoria de la que tiene disponible. Ambos indican que tiene que reducirse una o más configuraciones de la GPU (Intensidad, Tamaño de trabajo, Número de subprocesos o Coherencia de los subprocesos).

```
[19:28:16] Maximum buffer memory device 0
supports says 522586112
[19:28:16] Your settings come to 536870912
```

En Mineria Dual, se reajusta la GPU y se ejecuta por sí sola y luego se reajusta la CPU y se ejecuta por si sola. Después se intenta ejecutar ambas al mismo tiempo, aunque esperando a que se reajusten de nuevo en consecuencia (generalmente la CPU). La mayoría del software de minería de CPU intentará usar todos los recursos del sistema que pueda. Puede que tengas que configurar manualmente los parámetros del minado por CPU en lugar de dejar que se seleccionen automáticamente. Del mismo modo, hay situaciones

en las que existe un uso tan intensivo de la memoria que cualquier otro proceso normal que intente iniciarse puede llegar a causar un problema en el sistema (bloqueo, errores, etc.). En la minería por CPU, ninguna de las versiones de la Imagen HK usa memoria de intercambio por defecto para que así Hugh Pages no pueda activarse. De modo que, por lo general, debería existir un rendimiento similar independientemente del software de minería por CPU que se utilice.

Refrigeración, consumo de energía y monitorización del sistema

Tienes que controlar las temperaturas de la CPU mientras estés reajustando hasta que sepas qué configuración de minería es capaz de utilizar los algoritmos criptográficos que sueles usas. ¡El sistema puede dañase y es probable que ocurra sin una refrigeración adecuada durante la minería! En términos generales, la refrigeración OEM no es suficiente sin reducir significativamente la frecuencia de la CPU. Es una de las muchas razones por las que un sistema puede bloquearse o reiniciarse durante un minado simple o dual. Incluso los pequeños cambios de temperatura del ambiente pueden afectar significativamente a tu sistema y causar daños. Controla las temperaturas del ambiente y del sistema de forma regular durante el minado. Use watchtemp.sh si no cuentas con otros medios.

```
watchtemp.sh
#!/bin/bash
z=0
echo "T, Freq4,   Freq5,   Freq6,   Freq7,
T4, T5, T6, T7, TGPU"

while true :
do
     fa=`cat
/sys/devices/system/cpu/cpu4/cpufreq/scaling_c
ur_freq`
     fb=`cat
/sys/devices/system/cpu/cpu5/cpufreq/scaling_c
ur_freq`
     fc=`cat
/sys/devices/system/cpu/cpu6/cpufreq/scaling_c
ur_freq`
     fd=`cat
/sys/devices/system/cpu/cpu7/cpufreq/scaling_c
ur_freq`
     s1=`cat
/sys/devices/virtual/thermal/thermal_zone0/tem
p`
     s1t=$(($s1/1000))
     s2=`cat
/sys/devices/virtual/thermal/thermal_zone1/tem
p`
     s2t=$(($s2/1000))
     s3=`cat
/sys/devices/virtual/thermal/thermal_zone2/tem
p`
     s3t=$(($s3/1000))
     s4=`cat
/sys/devices/virtual/thermal/thermal_zone3/tem
p`
     s4t=$(($s4/1000))
         g1=`cat
/sys/devices/virtual/thermal/thermal_zone4/tem
p`
     g1t=$(($g1/1000))

     echo $z, $fa, $fb, $fc, $fd, $s1t, $s2t,
$s3t, $s4t, $g1t
     sleep 2
     (( z += 2 ))
done
```

No se han llegado a realizar pruebas de uso de energía, solo pruebas térmicas. Echa un vistazo a un artículo de los últimos meses en el que se detalla una prueba térmica preliminar. Se usan muchos recursos simultáneamente durante la minería dual y algunos algoritmos criptográficos usan mucha más energía que otros. Por ejemplo, el minado scrypt2 (VRM) utiliza aproximadamente un 20-25% más de energía que el algoritmo cryptonight (Monero). Monitoriza o realiza un estudio de consumo de energía para comprender mejor el uso de energía de la ARM-Mali antes o durante la minería dual para los diferentes algoritmos criptográficos.

Cuando la minería dual sea conservadora, permitirá que el resto del sistema operativo funcione, controle la temperatura y tenga en cuenta el uso de energía hasta que pruebes bien las configuraciones de la CPU y la GPU, y entonces será cuando podrás apoyarte más en ella. La minería dual lleva a estos sistemas al límite. Esta es una nueva frontera para los SBCs ARM,

así que ten en cuenta que estás al borde de la utilización extrema del sistema.

Probando con Crypotnight (Monero Coin) con sólo una GPU Odroid-XU4:

```
sgminer 5.5.6-ARM-RC1 - Started: [2018-03-13
03:06:15] - [0 days 12:38:48]
------------------------------------------------
---------------------------------
(5s):22.52 (avg):23.85h/s | A:700000  R:10000
HW:48  WU:0.187/m
ST: 1  SS: 0  NB: 421  LW: 48993  GF: 21  RF:
0
Connected to pool.supportxmr.com (stratum)
diff 5K as user
49cbPdjG8RUFjWau2aR9gR1bU6fsP7eGBfaXVsQuFtLrPr
ZkGpC4AuCEJsuKX
Block: e368fdd9...  Diff:905.5  Started:
[15:44:30]  Best share: 325K
------------------------------------------------
---------------------------------
[P]ool management [G]PU management [S]ettings
[D]isplay options [Q]uit
GPU 0:                |  13.41/ 13.40h/s | R:
2.6% HW:24 WU:0.103/m I: 7
GPU 1:                |  10.45/ 10.45h/s | R:
0.0% HW:24 WU:0.084/m I: 7
------------------------------------------------
---------------------------------
[13:41:07] Accepted 035e18c0 Diff 19.5K/5K GPU
0
[13:54:00] Accepted 058d63f3 Diff 11.8K/5K GPU
0
[13:55:45] Accepted 0b3c0178 Diff 5.83K/5K GPU
0
[14:04:37] Accepted 054a51d3 Diff 12.4K/5K GPU
1
[14:15:02] Accepted 014287d0 Diff 52K/5K GPU 1
[14:19:56] Accepted 018036d4 Diff 43.7K/5K GPU
1
[14:20:16] pool.supportxmr.com stale share
detected, submitting (user)
[14:20:16] Accepted 052596c8 Diff 12.7K/5K GPU
0
[14:36:38] Stratum connection to
pool.supportxmr.com interrupted
[14:40:09] Stratum connection to
pool.supportxmr.com interrupted
[14:42:19] Accepted 0b9ad8d2 Diff 5.65K/5K GPU
0
[14:44:23] Accepted 04b3a1c8 Diff 13.9K/5K GPU
0
[14:44:43] Accepted 011e41a0 Diff 58.6K/5K GPU
0
[14:54:26] pool.supportxmr.com stale share
detected, submitting (user)
[14:54:26] Accepted 060914e6 Diff 10.9K/5K GPU
0
[14:57:14] pool.supportxmr.com stale share
detected, submitting (user)
[14:57:14] Accepted 0a2b9f80 Diff 6.44K/5K GPU
1
[15:04:20] pool.supportxmr.com stale share
detected, submitting (user)
[15:04:20] Accepted 076a4aeb Diff 8.84K/5K GPU
0
[15:05:37] Accepted 09d5465e Diff 6.66K/5K GPU
0
[15:10:32] Accepted 0a066760 Diff 6.54K/5K GPU
1
[15:16:15] pool.supportxmr.com stale share
detected, submitting (user)
[15:16:16] Accepted 06082f75 Diff 10.9K/5K GPU
1
[15:18:06] pool.supportxmr.com stale share
detected, submitting (user)
[15:18:06] Accepted 0ce5243a Diff 5.08K/5K GPU
1
[15:18:30] Accepted ccf47695 Diff 81.9K/5K GPU
1
[15:30:46] Accepted 0857db6a Diff 7.86K/5K GPU
0
[15:30:57] Accepted 090f5ea6 Diff 7.23K/5K GPU
1
[15:31:34] Accepted 071e4b0f Diff 9.21K/5K GPU
1
[15:38:34] Accepted 0a0c5007 Diff 6.52K/5K GPU
0
[15:44:56] Accepted 88acc80f Diff 123K/5K GPU
0
```

La prueba se ejecutó durante más de 12 horas y todo transcurrió sin problemas. El resumen muestra 2 shares reales rechazados. Tengo una conexión a Internet relativamente lenta, de modo que algún servidor Stratum se desconectaría, lo cual significa que los shares obsoletos no son tan inusuales.

```
Summary of runtime statistics:

[15:46:02] Started at [2018-03-13 03:06:15]
[15:46:02] Pool:
stratum+tcp://pool.supportxmr.com:3333
```

```
[15:46:02] Runtime: 12 hrs : 39 mins : 47 secs
[15:46:02] Average hashrate: 0.0 Kilohash/s
[15:46:02] Solved blocks: 0
[15:46:02] Best share difficulty: 325K
[15:46:02] Share submissions: 142
[15:46:02] Accepted shares: 140
[15:46:02] Rejected shares: 2
[15:46:02] Accepted difficulty shares: 700000
[15:46:02] Rejected difficulty shares: 10000
[15:46:02] Reject ratio: 1.4%
[15:46:02] Hardware errors: 48
[15:46:02] Utility (accepted shares / min):
0.18/min
[15:46:02] Work Utility (diff1 shares solved /
min): 0.19/min

[15:46:02] Stale submissions discarded due to
new blocks: 0
[15:46:02] Unable to get work from server
occasions: 21
[15:46:02] Work items generated locally: 49055
[15:46:02] Submitting work remotely delay
occasions: 0
[15:46:02] New blocks detected on network: 421

[15:46:02] Summary of per device statistics:

[15:46:02] GPU0                  | (5s):13.48
(avg):13.40h/s | A:380000 R:10000 HW:24
WU:0.103/m
[15:46:02] GPU1                  | (5s):10.45
(avg):10.45h/s | A:320000 R:0 HW:24 WU:0.084/m
[15:46:02]
```

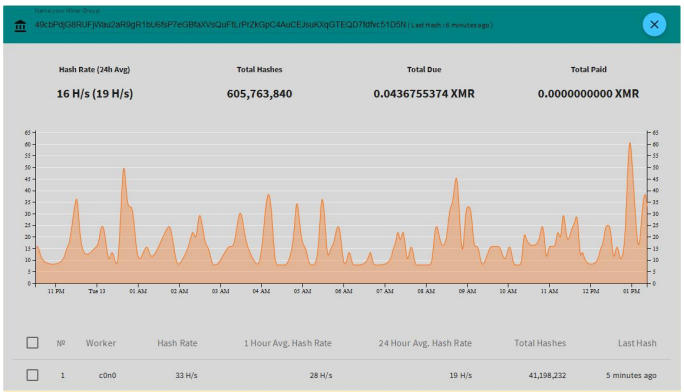


Figura 1 – Los resultados de la "pool" confirman los resultados del resumen

Con la modificación de sgminer se ha configurado un git para facilitar su uso y futuras modificaciones. Asimismo, el proceso de instalación ha cambiado y ya no requiere AMD_SDK, solamente la librería ARM Computer Vision and Machine Learning. A continuación, se muestra el nuevo procedimiento.

Descarga e instala la última Librería ARM Computer Vision and Machine Learning desde **https://github.com/ARM-software/ComputeLibrary/releases**. Ten en cuenta que han separado las librerías de Linux y Android para que ahora quepan en una tarjeta SD de 8GB. Usa el siguiente comando para extraer los archivos:

```
$ tar -xvzf `filename to extract`
```

A continuación, instala las dependencias y copie las cabeceras OpenCL:

```
$ apt-get install automake autoconf pkg-config
libcurl4-openssl-dev libjansson-dev libssl-dev
libgmp-dev make g++ git libgmp-dev
libncurses5-dev libtool opencl-headers mali-
fbdev
$ cp ./arm_compute-v18.03-bin-
linux/include/CL/* /usr/include/CL/
```

Descarga sgminer-5.5.6-ARM-RC1 con el siguiente comando:

```
$ git clone
https://github.com/hominoids/sgminer-arm
```

Luego, compila el código fuente:

```
$ cd sgminer-arm
$ git submodule init
$ git submodule update
$ autoreconf -fi
$ CFLAGS="-Os -Wall -march=native -std=gnu99 -
mfpu=neon" ./configure --disable-git-version -
-disable-adl --disable-adl-checks
```

Opcionalmente, puedes utilizar el siguiente comando para que sea más explícito en cuanto a concer dónde se ubica la librería y las cabeceras:

```
$ CFLAGS="-Os -Wall -march=native -std=gnu99 -
mfpu=neon -I/opt/arm_compute-v18.03-bin-
linux/include/CL" LDFLAGS="-L/opt/arm_compute-
v18.03-bin-linux/lib/linux-armv7a-neon-cl"
./configure --disable-git-version --disable-
adl --disable-adl-checks

$ make -j5
```

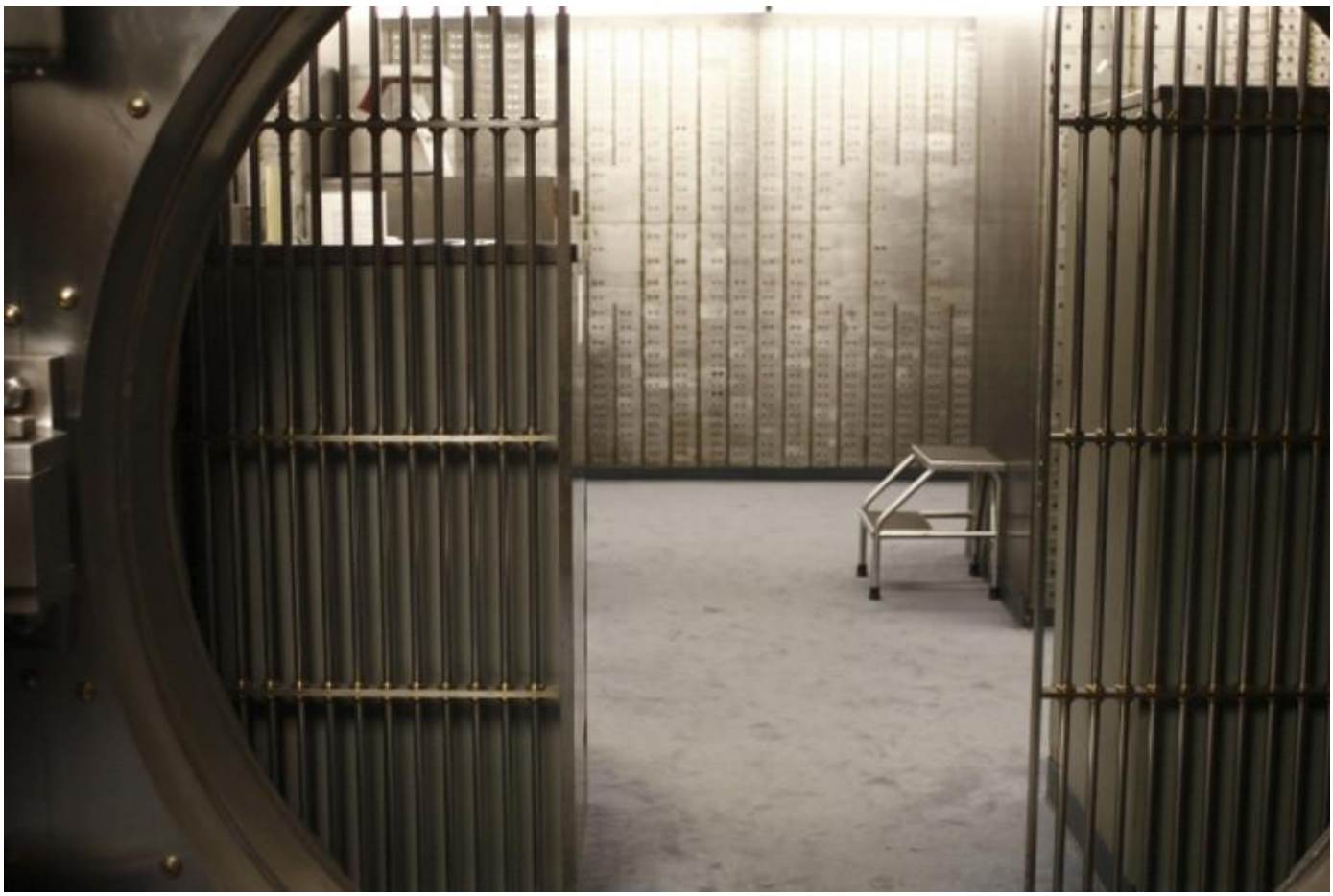
Aquí tienes el script y la configuración utilizada para las pruebas de la moneda XMR-Monero utilizando el algoritmo cryptonight:

```
#!/bin/bash

export GPU_FORCE_64BIT_PTR=1
export GPU_USE_SYNC_OBJECTS=1
export GPU_MAX_ALLOC_PERCENT=100
export GPU_SINGLE_ALLOC_PERCENT=100
export GPU_MAX_HEAP_SIZE=100

./sgminer -k cryptonight -o
stratum+tcp://pool.supportxmr.com:3333 -u
username -p password -I 7 -w 32 -d 0,1 --
thread-concurrency 8192 --monero --pool-no-
keepalive
```

¡La Comunidad ODROID ahora cuenta con el único sistema de minado GPU OpenCL ARM-Mali Linux multi-algoritmo del que tengo constancia de la comunidad crypto! Recuerda consultar el foro para obtener más información y actualizaciones en **https://forum.odroid.com/viewtopic.php?f=98&t=29571**.

Almacenamiento Seguro: Creando un Sistema de Archivos Cifrado en Linux.

April 1, 2018 By @Anand.moon ODROID-HC2



En Linux, el cifrado se realiza a través de dm-crypt usando LUKS igual que la configuración de la clave utilizando la API crypto del Kernel. Esta característica forma parte de Kernel Linux 4.14.18-106 o superior, adicionalmente necesitamos el driver Exynos5422 Slim SSS (Subsistema de seguridad) que admite encriptaciones AES, SHA-1, SHA-256, HMAC-SHA-1 y HMAC-SHA -256. El objetivo de device-mapper es proporcionar un cifrado transparente de los dispositivos de bloque utilizando API crypto del kernel.

- Cifrado AES con soporte aes-cbc/aes-ctr
- Cifrado por bloques (CBC)
- Contador (CTR) conocido como modo contador entero (ICM) y modo contador de enteros segmentados (SIC)
- ESSIV (“Encrypted salt-sector initialization vector”) permite al sistema crear IVs basados en un hash que incluye el número del sector y la clave de cifrado
- SHA-256 es el algoritmo hash utilizado para la derivación de claves
- XTS es el modo de encadenamiento orientado al contador
- PLAIN64 es un mecanismo de generación IV que simplemente pasa directamente el índice del sector de 64 bits al algoritmo de encadenamiento como IV

```
Parameters: < cipher > < key > < iv_offset > < device path > < offset > [< #opt_params > < opt_params >]
< cipher >
Encryption cipher, encryption mode and Initial Vector (IV) generator.
The cipher specifications format is:
cipher[ :keycount ]-chainmode-ivmode[ :ivopts ]
Examples:
aes-cbc-essiv:sha256
aes-xts-plain64
Serpent-xts-plain64
```

```
Cipher format also supports direct
specification with kernel crypt API
format (selected by capi: prefix). The IV
specification is the same
as for the first format type.
This format is mainly used for specification
of authenticated modes.
The crypto API cipher specifications format
is:
Capi:cipher_api_spec-ivmode[ :ivopts ]
Examples:
capi:cbc(aes)-essiv:sha256
capi:xts(aes)-plain64
Examples of authenticated modes:
capi:gcm(aes)-random
capi:authenc(hmac(sha256),xts(aes))-random
capi:rfc7539(chacha20,poly1305)-random
```

Prueba de rendimiento Cryptsetup con DRAM

```
# root@odroid:~# cryptsetup benchmark

# Tests are approximate using memory only (no
storage IO).
PBKDF2-sha1 297552 iterations per second
PBKDF2-sha256 195338 iterations per second
PBKDF2-sha512 125068 iterations per second
PBKDF2-ripemd160 247305 iterations per second
PBKDF2-whirlpool 27935 iterations per second
# Algorithm | Key | Encryption | Decryption
aes-cbc 128b 73.8 MiB/s 97.7 MiB/s
serpent-cbc 128b 40.9 MiB/s 42.9 MiB/s
twofish-cbc 128b 58.0 MiB/s 62.0 MiB/s
aes-cbc 256b 59.8 MiB/s 74.0 MiB/s
serpent-cbc 256b 41.5 MiB/s 42.7 MiB/s
twofish-cbc 256b 59.1 MiB/s 62.0 MiB/s
aes-xts 256b 110.6 MiB/s 95.2 MiB/s
serpent-xts 256b 41.6 MiB/s 42.2 MiB/s
twofish-xts 256b 59.7 MiB/s 61.9 MiB/s
aes-xts 512b 86.1 MiB/s 72.5 MiB/s
serpent-xts 512b 42.1 MiB/s 42.4 MiB/s
twofish-xts 512b 60.7 MiB/s 61.6 MiB/s
```

Prueba de rendimiento Cryptsetup con HDD

La Figura 1 muestra los resultados de las pruebas de un ODROID-HC2 utilizando un HDD NAS WD 4TB a 5400RPM, que pueden variar según el tipo de disco duro utilizado:

```
$ iozone -e -I -a -s 100M -r 4k -r 16k -r 512k
-r 1024k -r 16384k -i 0 -i 1 -i 2
```

Encryption cipher	DD result (MB/sec)		Iozone result (kB/sec)					
	Write speed	Read speed	write	rewrite	read	reread	random read	random write
aes-cbc-essiv:sha256 (128 bit key)	69.2	75.2	39662	46940	94300	112724	97757	61227
aes-cbc-essiv:sha256 (192 bit key)	82.2	70.4	39674	48221	104894	115422	97860	52920
aes-cbc-essiv:sha256 (256 bit key)	80.7	64.4	38797	47699	99172	108811	101844	56758
aes-ctr-plain (128 bit key)	76.6	99.6	40956	49572	129875	160192	141948	61756
aes-xts-plain64 (256 bit key)	86.5	87.6	41149	47190	105411	131470	127655	66905
aes-xts-plain64 (512 bit key)	81.6	69.4	37643	39412	104134	109492	97255	57694
twofish-cbc-essiv:sha256 (128 bit key)	77.1	69.3	44008	48753	107413	128105	100541	66445
twofish-cbc-essiv:sha256 (256 bit key)	76.9	68.0	42128	50422	109972	120566	107068	58512
twofish-xts-plain64 (256 bit key)	80.2	55.9	40206	44363	97249	108703	88713	53685
twofish-xts-plain64 (512 bit key)	74.4	57.0	39311	43484	97468	109982	112207	57647
serpent-cbc-essiv:sha256 (128 bit key)	80.1	55.9	39309	41447	87069	106448	111989	52004
serpent-cbc-essiv:sha256 (256 bit key)	80.1	56.4	38312	41841	88125	104045	102048	61575
serpent-xts-plain64 (256 bit key)	71.9	48.1	36857	44134	86668	92474	92731	51055
serpent-xts-plain64 (512 bit key)	72.3	48.0	37968	41449	82182	90441	85489	55216

Figura 1 – Resultados de la prueba de rendimiento de Cryptsetup para un ODROID-HC2

Encriptar el disco duro usando cryptsetup

Instala cryptsetup y como no necesitamos reiniciar, inicia los módulos dm-crypt.

```
$ sudo apt-get install cryptsetup
$ sudo modprobe dm-crypt sha256 aes
```

Prueba a verificar que cryptsetup y dm-crypt estén funcionando:

```
$ fallocate -l 128MiB /tmp/test.bin
$ dd if=/dev/urandom of=/tmp/testkey.key
bs=128 count=1
$ sync
$ cryptsetup luksFormat --debug -q -d
/tmp/testkey.key --cipher aes-cbc-essiv:sha256
-h sha256 -s 128 /tmp/test.bin

$ fallocate -l 128MiB /tmp/test.bin
$ dd if=/dev/urandom of=/tmp/testkey.key
bs=128 count=1
$ sync
$ cryptsetup luksFormat --debug -q -d
/tmp/testkey.key --cipher aes-ctr-plain -h
sha256 -s 128 /tmp/test.bin
```

Una vez que compruebes que cryptsetup funciona bien, puede empezar a encriptar el disco. Ten en cuenta que se trata de un cifrado de todo el disco, de modo que el disco debe formatearse.

```
$ sudo wipefs -a /dev/sda1
/dev/sda1: 6 bytes were erased at offset
0x00000000 (crypto_LUKS): 4c 55 4b 53 ba be
```

Crea una clave para desbloquear el volumen

La encriptación Luks admite múltiples claves. Estas claves pueden ser contraseñas introducidas de forma interactiva, o archivos clave pasados como argumento para desbloquear la partición cifrada.

```
$ sudo dd if=/dev/urandom of=/root/keyfile
bs=1024 count=4
$ sudo chmod 400 /root/keyfile
```

Para crear la partición cifrada en /dev/sda1, se utiliza luks. El cifrado de la partición será gestionado usando el comando cryptsetup.

```
$ sudo cryptsetup --verify-passphrase
luksFormat /dev/sda1 -c aes-cbc-essiv:sha256 -
h sha256 -s 128
```

Esto te solicitará una contraseña que debería ser larga (más de 8 caracteres), lo cual debería tenerse en cuenta. Los siguientes pasos muestran cómo abrir la unidad encriptada y asignar la dp-crypt al sistema de archivos. A continuación, desbloquea la unidad usando la contraseña que acabamos de proporcionar, luego crea un sistema de archivos en el dispositivo.

```
$ sudo cryptsetup luksOpen /dev/sda1
securebackup
```

Formatea la partición:

```
$ sudo mkfs -t ext4 -m 1
/dev/mapper/securebackup
```

Agrega una clave luks que permite el montaje automático durante el arranque:

```
$ sudo cryptsetup -v luksClose securebackup
$ sudo cryptsetup luksAddKey /dev/sda1
/root/keyfile
```

Actualiza el archivo /etc/crypttab para hacer referencia al archivo de claves:

```
$ cat /etc/crypttab
# < target name > < source device > < key file
> < options >
securebackup /dev/sda1 /root/keyfile luks
```

Necesitamos decirle al subsistema dm-crypt que este dispositivo debe montarse antes de la partición HDD encriptada. Para hacerlo, abre el archivo /etc/default/cryptdisks y busca la línea CRYPTDISKS_MOUNT = “”:

```
$ cat /etc/default/cryptdisks
# Run cryptdisks initscripts at startup?
Default is Yes.
CRYPTDISKS_ENABLE=Yes

# Mountpoints to mount, before cryptsetup is
invoked at initscripts. Takes
# mountpoins which are configured in
/etc/fstab as arguments. Separate
# mountpoints by space.
# This is useful for keyfiles on removable
media. Default is unset.
CRYPTDISKS_MOUNT="/root/keyfile"

# Default check script. Takes effect, if the
'check' option is set in crypttab
# without a value.
CRYPTDISKS_CHECK=blkid

# Default precheck script. Takes effect, if
the 'precheck' option is set in
# crypttab without a value.
# Default is 'un_blkid' for plain dm-crypt
devices if unset here.
CRYPTDISKS_PRECHECK=
```

Comprueba que la unidad esté asignada al dispositivo de cifrado:

```
$ sudo cryptsetup luksOpen /dev/sda1
securebackup
$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
mmcblk1 179:0 0 29.8G 0 disk
|-mmcblk1p1 179:1 0 128M 0 part /media/boot
`-mmcblk1p2 179:2 0 29.7G 0 part /
sda 8:0 0 149G 0 disk
`-sda1 8:1 0 149G 0 part
`-securebackup 254:0 0 149G 0 crypt
```

Para montar automáticamente el disco en el siguiente reinicio, debes actualizar la entrada /etc/fstab:

```
$ mkdir -p /media/secure
$ sudo cat /etc/fstab
UUID=e139ce78-9841-40fe-8823-96a304a09859 /
ext4 errors=remount-ro,noatime 0 1
LABEL=boot /media/boot vfat defaults 0 1
```

```
/dev/mapper/securebackup /media/secure ext4
defaults,rw 0 2
```

Podrás montar manualmente el disco si los pasos anteriores han dado resultado:

```
$ mount /dev/mapper/securebackup /media/secure
```

Rendimiento de CIFS/Samba en un HDD encriptado

Las Figuras 2 – 7 muestran el rendimiento utilizando la siguiente configuración de hardware:

- HC2 + ubuntu-16.04.3-4.14-minimal-odroid-xu4-20171213.img.xz with updated 4.14.18-106 kernel
- Tarjeta MicroSD de 8GB
- HDD Seagate 8TB (ST8000AS0002)
- Encriptación: aes-xts-plain64 (256 bit key, SHA256 hash)

Samba usa la siguiente configuración:

```
[HDD INTERNAL]
comment = NAS
path = /media/internal
valid users = odroid
writable = yes
create mask = 0775
directory mask = 0775
# Tweaks
write cache size = 524288
getwd cache = yes
use sendfile = yes
min receivefile size = 16384
socket options = TCP_NODELAY IPTOS_LOWDELAY
```

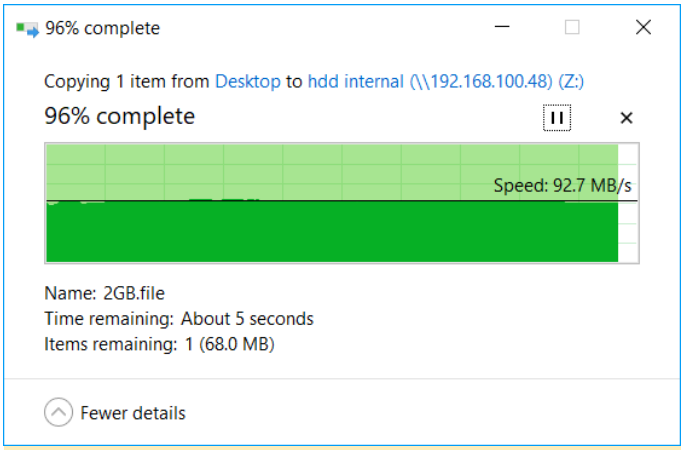


Figura 2 – Antes de la prueba de cifrado 1

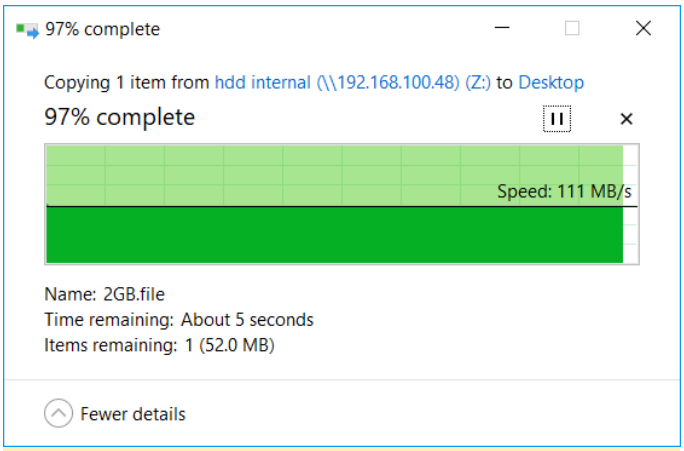


Figura 3 – Antes de la prueba de cifrado 2

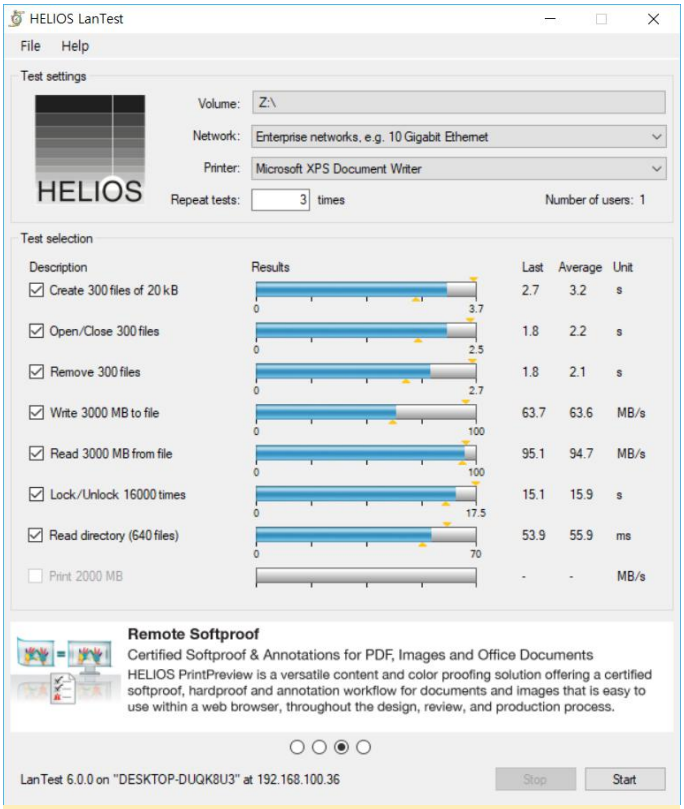


Figura 4 – HELIOS LanTest antes del cifrado

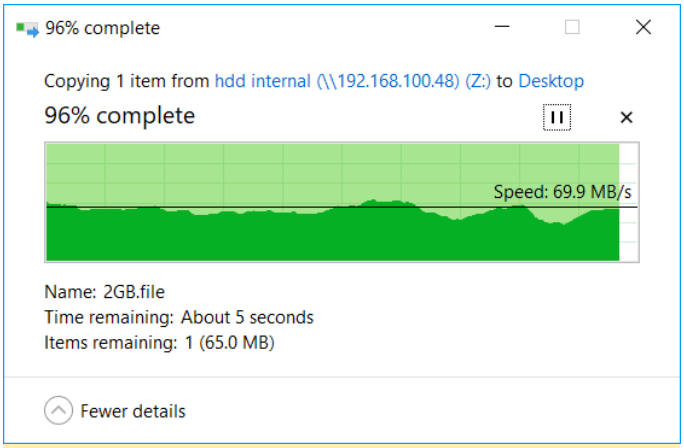


Figura 5 – Después de la prueba de cifrado 1

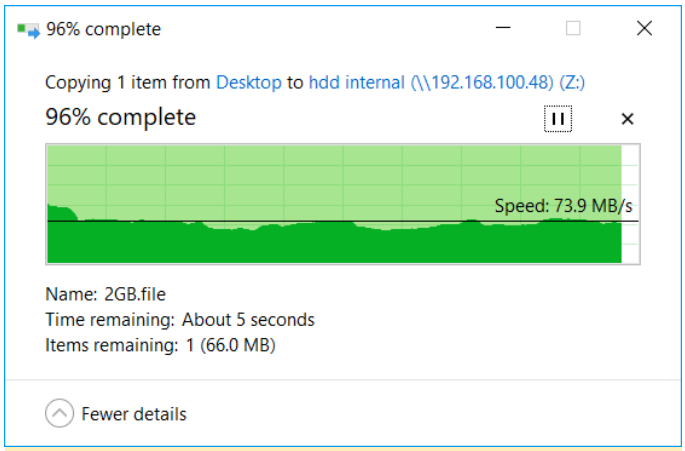


Figura 6 – Después de la prueba de cifrado 2

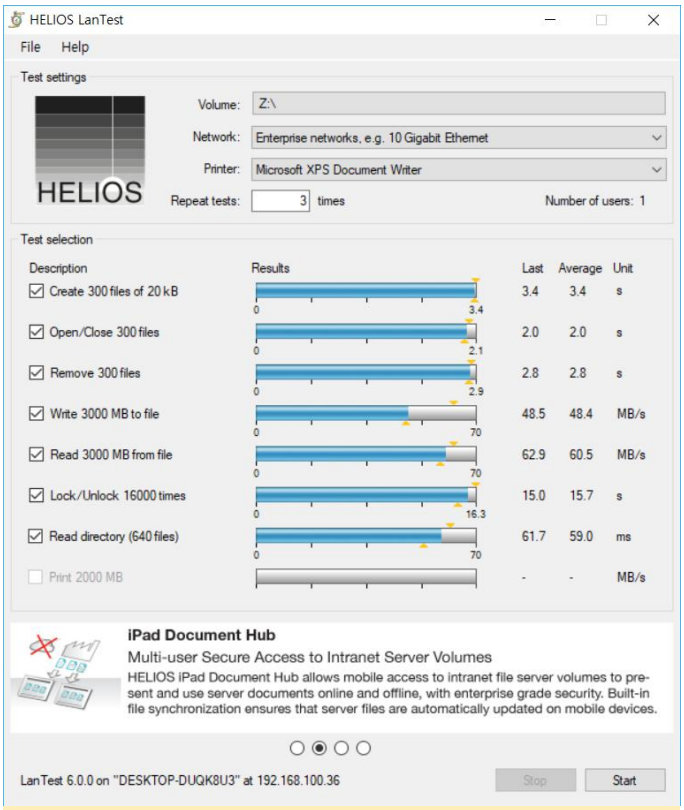


Figura 7 – HELIOS LanTest después del cifrado

Para el ver el artículo original, consulta la entrada de la Wiki en **https://wiki.odroid.com/odroid-xu4/software/disk_encryption**.

Tvheadend

April 1, 2018 By @Pepes ODROID-C2, Tutoriales

Tvheadend (TVH) es un software de servidor que puede leer secuencias de video desde fuentes de LinuxTV y publicarlos como transmisiones de video visibles en dispositivos como televisores inteligentes a través de una Internet IP. Se puede usar también como grabadora. Las fuentes de entrada pueden incluir: DVB-S, DVB-C / T, ATSC, IPTV y SAT> IP, por nombrar algunas. Se pueden combinar múltiples servidores TVH para formar una red. Las siguientes instrucciones muestran cómo compilar el código TVH para el ODROID-C2.

Instalación

Instala los componentes de software necesarios usando los siguientes comandos:

```
$ sudo apt-get install cmake git libssl-dev
libdvbcsa-dev ffmpeg liburiparser-dev openssl
libavahi-client-dev zlib1g-dev libavcodec-dev
libavutil-dev libavformat-dev libswscale-dev
libavresample-dev dvb-apps libiconv-hook-dev
```

Ahora, necesitamos descargar el código fuente:

```
$ wget
https://github.com/tvheadend/tvheadend/archive
/master.zip
```

Muévelo a un directorio de trabajo y luego extráelo:

```
$ cd ~
$ unzip master.zip -d tvheadend
```

Introduce el correspondiente directorio y compila:

```
$ cd tvheadend/tvheadend-master
$ ./configure
$ make
```

Deberías observar errores de compilación en este punto.

Para solucionar esto, descarga los siguientes archivos a ~/tvheadend/tvheadend-master desde el repositorio git:

- config.guess (**https://goo.gl/3wJNcP**)
- config.sub (**https://goo.gl/NKzVes**)

Reemplaza los archivos originales por estas versiones:

```
$ cp config.guess build.linux/ffmpeg/libtheora
$ cp config.sub build.linux/ffmpeg/libtheora
```

Ahora podemos repetir los pasos de compilación:

```
$ make
$ sudo make install
```

La primera vez que ejecutes Tvheadend, necesitarás ejecutarlo con la opción -C:

```
$ tvheadend -C
```

Después, deberás definir la configuración a través de la IU (IP: 9981) y finalizarla presionando CTRL + C. Luego podrás ejecutarlo con otras opciones como:

```
$ screen -m -s tvheadend
```

or:

```
$ tvheadend
```

Por último, debes tener PuTTY conectado todo el tiempo; de lo contrario, finalizará.

Desarrollo Android: Así que Quieres Ser un Desarrollador de Aplicaciones

April 1, 2018 By Randy Hall Android, Tutoriales

Entonces, ¿Quieres ser un desarrollador de aplicaciones? ¡Veamos qué puedo hacer para ayudarte con este tema! Un vistazo a los números atrasados de ODROID Magazine ha puesto de manifiesto que hay un montón de artículos detallados sobre Android en ODROID. Lo que la comunidad ha estado echando en falta es una serie de artículos sobre "Entonces, quieres empezar con esto". Seguro que hay multitud de videos en YouTube con cierta calidad de audio y video cuestionable, así como blogs y tutoriales online que puedes o no ser capaz de encontrar con una búsqueda, pero en lugar de intentar sacar algo en claro de todo este bullicio, pensé que no estaría de más aportar mi granito de arena con todo este tema.

Además de ser ordenadores Linux de placa reducida, los modelos ODROID como C1+, C2 e incluso XU4 son perfecto para el desarrollo de aplicaciones Android, ya sea para un uso portátil, en tablet o en automóviles. Poseo un ODROID-C1 muy querido que compré hace un par de años y aunque está un poco anticuado con respecto a la última generación de hardware disponible, su celebro es el mismo que tiene el ODROID-C0 y C1 + y continúa siendo una plataforma totalmente válida para hacer pruebas y testear aplicaciones Android.

Camina antes de correr

Antes de meternos de lleno en el uso de las placas ODROID para el desarrollo de aplicaciones, debemos tratar algunos aspectos básicos. En primer lugar, necesitarás una forma de crear el software que ejecutarás en el sistema operativo Android. Para ello, necesitarás un entorno de desarrollo, que básicamente es un conjunto de herramientas que trabajan conjuntamente para crear la aplicación. Android tiene algunas opciones para desarrollar

programas en un entorno de desarrollo integrado (IDE). Nos centraremos en el "oficial", Android Studio.

Segundo, necesitarás tener un lugar seguro para almacenar tu trabajo. Aunque personalmente soy un fanático del auto-hospedaje, reconozco que cualquier estrategia de copias de seguridad que implique almacenar una copia de tus datos en otro lugar es una muy buena idea. Así que asegúrate de suscribirte a un repositorio de código fuente online basado en git en un proveedor git como Github, Gitlab o Bitbucket.

Finalmente, nos pararemos un momento para describir cómo debería ser el proceso de desarrollo del software, tanto de forma general como específica para Android:

- Configurar el entorno de desarrollo
- Configurar Android en ODROID
- Crear la app Hello World
- Implementar la app en Android/ODROID
- ¡Repetir!

Configurando el entorno de desarrollo

Cuando configuras un entorno de desarrollo, lo más probable es que lo hagas en uno de los tres sistemas operativos: Linux, Windows o macOS. En este caso instalaremos Android Studio en un Mac, aunque descubrirás que el proceso es sustancialmente similar, independientemente de la plataforma. Comienza visitando la página de descargas de Android Studio en **https://developer.android.com/studio**. La página generalmente selecciona automáticamente la descarga correcta para tu sistema operativo (SO), aunque siempre puedes descargar cualquiera de los instaladores o paquetes para el sistema operativo que quieras..

Cuando escribí este artículo, las instrucciones de instalación de Android Studio eran muy sencillas e incluían pasos específicos para cada sistema operativo compatible, las cuales están disponibles en **https://developer.android.com/studio/install.html**. La instalación en mi Mac fue minuciosa y notablemente indolora. El instalador descarga una serie de librerías Android necesarias para el desarrollo y proporciona el hipervisor Intel HAXM que ayudar con la emulación de Android en el futuro. En solo unos minutos, el trabajo estaba hecho, tal y como se muestra en la Figura 1.

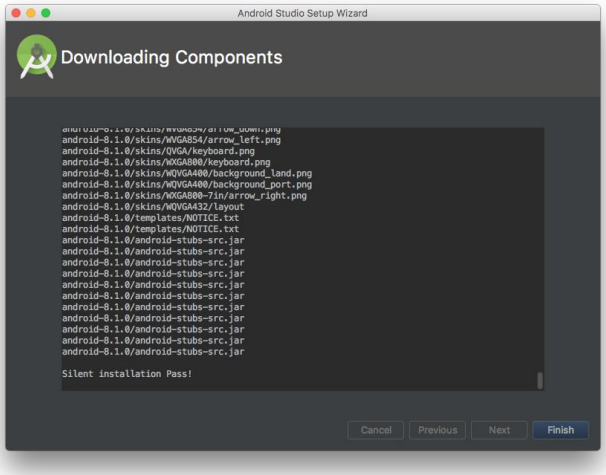


Figura 1 – Descargando componentes

Una vez que hagas clic en "Finish", se iniciará la aplicación y te encontrará con una lista de opciones. Por ahora, selecciona "Start a new Android Studio project" y echa un vistazo. Por supuesto, te sumergirás de inmediato en el mundo de la programación Java con la pantalla "Create Android Project", ya que se te pedirá que crees un "company domain", que explicaré en la siguiente sección.

¿Qué dominio?

En Java, todas las clases de objetos se dividen en espacios de nombres, que son básicamente agrupaciones en las que puede crear clases de Java de forma segura, que pueden nombrarse como quieras y que no afectarán a las definiciones de clase de Java de otra persona. Es similar en concepto a los nombres de dominio en la web, que es por lo que Android Studio se refiere a él como un "company domain", tal y como muestra la Figura 2. Por ahora, puedes poner cualquier dominio que quieras, pero definitivamente no uses un dominio que no poseas ni controles.

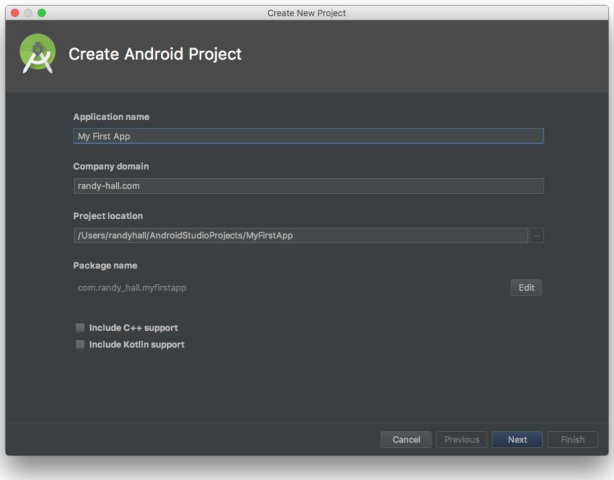


Figura 2 – Creando un proyecto de Android

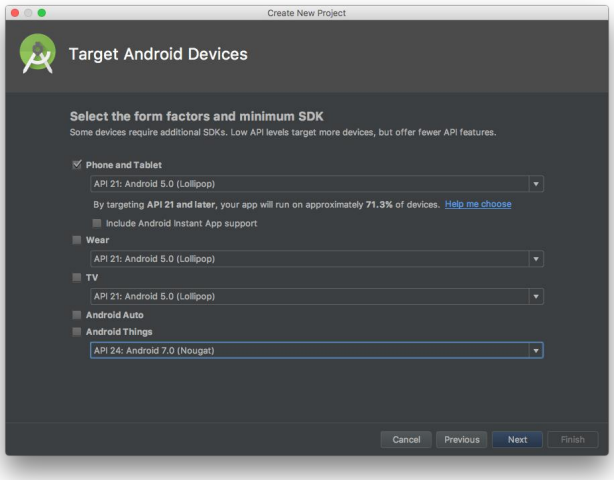


Figura 3 – Dispositivos Android de destino

Puede ver el espacio de nombres resultante para este nuevo proyecto mirando el campo "Package Name" en esa pantalla, que en mi caso muestra "com.randy_hall.myfirstapp". Por ahora, sólo tienes que entender que así es como funcionan Java y por extensión, Android.

¿Qué dispositivo?

A continuación, vamos a ver qué nivel de API Android queremos usar para este proyecto. Esta es una ventaja increíble de Android y al mismo tiempo uno de sus defectos más frustrantes. Existe una variedad tan amplia de versiones del software Android en el mundo que cuando eliges compatibilidad para un nivel de API sobre otro, estás excluyendo intrínsecamente que un determinado número de dispositivos Android usen tu aplicación. Por ahora, seremos razonables y eligiéremos "API 21: Android 5.0 (Lollipop)" tal y como se muestra en la Figura 3, lo que, según Android Studio, significa que nuestra aplicación se ejecutará en aproximadamente el 71.3% de los dispositivos Android a partir de este momento. A medida que se introducen nuevos dispositivos Android y se actualizan o retiran teléfonos viejos, este número en general aumentará con el tiempo.

Añadiendo la actividad

Android Studio puede ayudarte a despegar algo más rápido al proporcionarte algunos pilares, que son plantillas de patrones de aplicaciones comunes que te proporcionaran un punto de partida para empezar tu trabajo. Hay varias para elegir, de las que hablaremos más en el futuro, de momento vamos a elegir "Empty Activity", tal y como se muestra en la Figura 4.

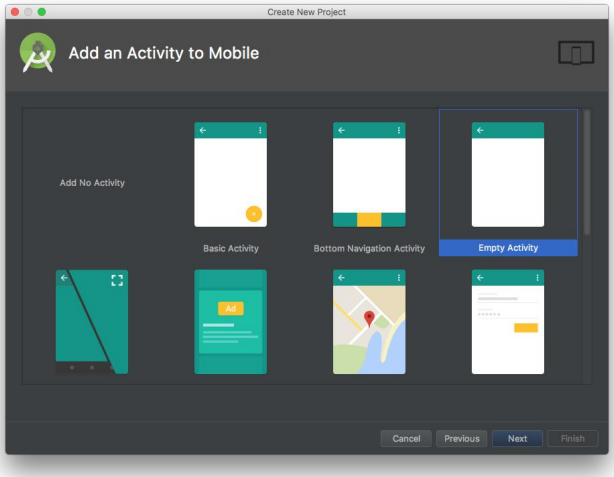


Figura 4 – Añadiendo una actividad

Se te pedirá que configures la actividad vacía proporcionándole un nombre y un nombre de diseño. Ambos se usarán para generar el código del proyecto, de modo que puedes asignarle el nombre que quieras. Me quedaré con los valores por defecto de "MainActivity" y "activity_main", pero siempre puedes cambiarlos más adelante en el código.

Una vez que se haya generado MainActivity, ¡las cosas empiezan a mejorar! Hemos cliqueado mucho, y

Android Studio ha realizado algunos trabajos de fondo, pero ahora estamos en el IDE, listos para empezar. Puedes observar en seguida que la cantidad de código generado es relativamente pequeña. Esto es debido en gran parte a que hemos elegido una "Empty Activity", que está notablemente vacía. Si hubiéramos elegido otra actividad, nos habríamos encontrado con una cantidad de código considerablemente mayor.

Durante el camino, recibí un error "Gradle Sync", tal y como se muestra en la Figura 5, lo que significa que no tenía instalada la API de Android más reciente. Hice clic en "Install missing platform(s)" y actualicé, el error volvió a repetirse para las herramientas de compilación. La instalación de estas dos piezas (si no estaban ya allí) me ayudó a superar esta barrera y situándome en el espacio de trabajo, estaba listo para la codificación.

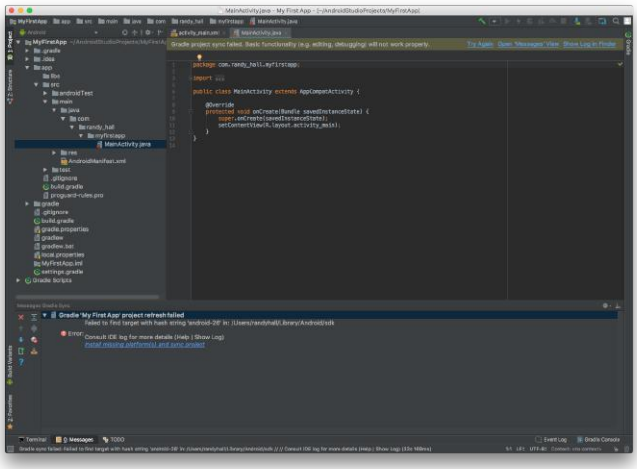


Figura 5 – Error Gradle Sync

Usando Git

Ahora estamos listos para pasar al siguiente paso, que es crear una cuenta git para almacenar y quizás compartir nuestro trabajo. Para esto, utilizaremos uno de los proveedores git algo menos conocidos, Github (**https://www.github.com**).

Los proveedores de git de alojamiento como Github, Gitlab y Bitbucket se basan en el ampliamente utilizado software de control de versiones de código abierto. Podríamos profundizar en la filosofía que hay detrás de git en otro artículo, pero por ahora es suficiente saber que el método que estamos utilizando para respaldar nuestros proyectos está muy bien sustentado en el mundo del desarrollo de aplicaciones.

Crear una nueva cuenta en Github es fácil. Debe elegir un nombre de usuario original, proporcionar una cuenta de correo electrónico válida para activar la cuenta de Github y seleccionar una contraseña. Una vez que hayas hecho esto, Github te ofrecerá suscribirte a servicios adicionales, pero puedes elegir el plan gratuito por defecto, lo que significa que tus repositorios estarán públicamente visibles. Si esto es un problema para ti, siempre puede optar por Gitlab, que proporciona una opción gratuita de repositorio git privado, u optar por una suscripción Github con el privilegio de mantener repositorios privados.

¿Listo para empezar?

Una vez que te hayas registrado, estás a punto de conectar Android Studio a tu cuenta de Github. Esto lo recogeremos en la próxima entrega y entonces será cuando creemos la más típica de todas las aplicaciones: la aplicación "Hello World". ¡El mundo del desarrollo de app Android ya está abierto!

Al margen de lo que pueda pensar, no escribo estos artículos pensando en mi propia comodidad. Me encantaría saber el tipo de desarrollo de aplicaciones en el que estás interesado. Tus comentarios y sugerencias me servirán de apoyo e influirán en cómo evolucionarán estos artículos con el tiempo. Si estás interesado en participar, puedes dejar tus comentarios en la versión interactiva de ODROID Magazine o visitar el foro de ODROID Magazine en **https://forum.odroid.com**.

Configurando tu ODROID: El ODROID-XU4 como un NAS polivalente y versátil

April 1, 2018 By Adrian Popa Linux, ODROID-XU4, Tutoriales

Compré mi ODROID-XU4 con la intención de convertirlo en un NAS. Sin embargo, no quería decidirme por una distribución NAS especializada como OpenMediaVault porque quería que mi ODROID-XU4 hiciera más cosas aparte de ser un simple NAS. Por ejemplo, tenía pensado transcodificar programas de TV grabados desde mi televisor al estándar H264, usando la codificación por hardware del ODROID-XU4 (tal y como se describe en **http://bit.ly/2jnv4Za**), y también hacer uso de los pines GPIO más adelante. Un problema añadido que tiene OpenMediaVault es que se ejecuta sobre Debian, y yo quería seguir usando Ubuntu para beneficiarse de los nuevos paquetes.

Pierdo muchas de las ventajas de utilizar una distribución especializada y, en consecuencia, tengo que descubrir formas alternativas de hacer las cosas de una manera sencilla y amigable. Todo esto representa una gran oportunidad para adquirir nuevos conocimientos.

Estos son los pasos que debemos seguir:

- Instalar Webmin (**http://bit.ly/J5Wtfl**) para facilitar la administración
- Montar los discos
- Configurar recursos compartidos en red (Samba/NFS)
- Instalar Owncloud
- Asegurar y optimizar el sistema operativo

Estas instrucciones dan por sentado que cuentas con una experiencia del sistema de nivel medio o superior.

Webmin

Todo NAS necesita una buena GUI web. Desafortunadamente, la GUI de OpenMediaVault no es una opción, y después de buscar alternativas

durante bastante tiempo, decidí usar Webmin. Webmin ha existido desde 1997 y tiene un sólido soporte para tareas generales de mantenimiento de servidores. Tiene la ventaja de que incluso los usuarios inexpertos pueden desenvolverse con la ayuda integrada para configurar y administrar todo tipo de servidores como Apache, MySQL, Mail, DNS y más. Tiene un soporte muy bueno para la administración de RAID y LVM, y también admite el uso compartido de archivos por Samba y NFS. Desafortunadamente, carece de soporte para servicios más recientes como Transmission o Owncloud, pero siempre puedo configurarlos manualmente.

Para instalarlo, sigue los siguientes pasos:

```
$ echo "deb
http://download.webmin.com/download/repository
sarge contrib" | sudo tee
/etc/apt/sources.list.d/webmin.list
$ wget http://www.webmin.com/jcameron-key.asc
$ sudo apt-key add jcameron-key.asc
$ rm jcameron-key.asc
$ sudo apt-get update
$ sudo apt-get install libapt-pkg-perl libnet-
ssleay-perl libauthen-pam-perl libio-pty-perl
apt-show-versions apt-transport-https
$ sudo apt-get install webmin
$ sudo systemctl enable webmin
$ sudo systemctl start webmin
```

Puedes iniciar sesión con la dirección IP de tu dispositivo en el puerto 10000 y así poder usar la interfaz web: https://ip-odroid: 10000. Sin embargo, después de iniciar sesión (con cualquier usuario del sistema con acceso sudo) es posibles que no quedes muy impresionado con la interfaz por defecto. Parece que ha salido de la década de los 90.

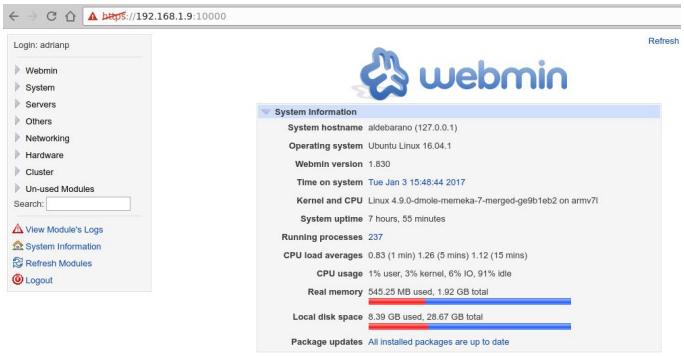


Figura 1 – Interfaz estándar de Webmin

Lo primero que debemos hacer es mejorar su aspecto con ayuda de un tema. El tema más atractivo se llama "Authentic Theme", que ofrece muchas funciones, incluso permite el uso amigable de dispositivos móviles. Puede obtener la última versión desde **http://bit.ly/2jf468e** e instalarla usando el siguiente comando:

```
$ wget https://github.com/qooob/authentic-
theme/releases/download/19.12/authentic-theme-
19.12.wbt.gz
```

Navega dentro de Webmin a "Webmin Configuration -> Webmin Themes -> Install themes -> From uploaded file" y selecciona el tema descargado recientemente. Tras una breve espera y actualizar, aparecerá la siguiente página:

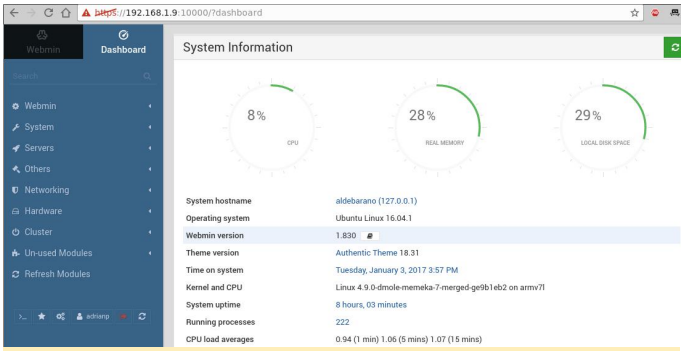


Figura 2 – Webmin con Authentic theme

Puedes explorar las características de Webmin utilizando la herramienta de búsqueda de la interfaz. Tenga en cuenta que puede instalar módulos de terceros disponibles en **http://bit.ly/2jf6KLd**.

Montando los discos

Primero, deberá decidir si va a utilizar RAID o LVM con tus discos y qué sistema de archivos usarás. No voy a entrar en detalles sobre la configuración RAID/LVM porque el tema ya ha sido discutido en anterios artículos. Sin embargo, incluso sin tener mucha experiencia, puedes usar Webmin para que éste haga el trabajo pesado por ti y usar la ayuda integrada para obtener más información. Webmin te solicitará que instales las dependencias que falten. Una vez que tengas listas tus particiones, puedes empezar a montarlas.

El método tradicional para montar el disco es usar /etc/fstab (y Webmin tiene un completo módulo para gestionar también esto), pero puedes tener

problemas si inicias tu sistema sin el disco conectado (al sistema le gusta esperar al disco). Prefiero usar autofs, que monta los discos (locales o en red) a demanda y los desmonta cuando no están en uso. Desafortunadamente, webmin no lo gestiona, de modo que necesitarás tirar del intérprete de comandos:

```
$ sudo apt-get install autofs
```

Tendrás que editar /etc/auto.master y añadir una entrada para montar tu disco, especificando el directorio base y su archivo de configuración:

```
$ sudo vi /etc/auto.master
# add at the end your mountpoint
/media/yourdisk /etc/auto.yourdisk --timeout
20
```

En el comando anterior, reemplaza tu disco por la ruta que desea usar. Luego, edita este archivo de configuración y agrega tus particiones y sus parámetros de montaje, usando el comando "blkid" para localizar el UUID correcto del disco:

```
$ sudo blkid
$ sudo vi /etc/auto.yourdisk
xfs-partition   -
fstype=xfs,dev,exec,suid,noatime
:UUID=9d2d675d-cb08-45b2-b222-c981a8d00c06
```

Reinicia autofs, y cuando accedas a /media/yourdisk/xfs-partition, tu partición se montará automáticamente:

```
$ sudo service autofs restart
```

Tendrás que encargarete de los parámetros de montaje porque cada sistema de archivos tiene sus propios parámetros y pueden afectar al rendimiento. Por ejemplo, sino activas el parámetro big_writes en NTFS, tendrás un rendimiento muy bajo. En caso de duda, puedes ser un poco chapucero y usar Webmin para crear entradas en /etc/fstab, probarlas para asegurarte de que los parámetros son correctos y migrarlas al diseño de autofs más tarde (eso es lo que yo hice). Para forzar el desmontaje de los discos, simplemente puede reiniciar el servicio autofs.

Configurar recursos compartidos

Para configurar recursos compartidos por Samba (y también para instalar Transmission para poder descargar torrents), puede seguir la guía "Diseñando su propio seedbox" que aparece en Odroid Magazine http://bit.ly/2j3xpaK. También puede experimentar con la interfaz de Webmin y crear fácilmente recursos compartidos y usuarios con unos cuantos clics. Por ejemplo, la Figura 3 muestra el cuadro de diálogo "Crear recurso compartido NFS". Al hacer clic en los elementos del formulario, se muestra un menú de ayuda contextual que explica muy bien qué hace ese elemento en concreto. Esto puede ayudarte con ciertas cosas con las que quizás no estés muy familiarizado.

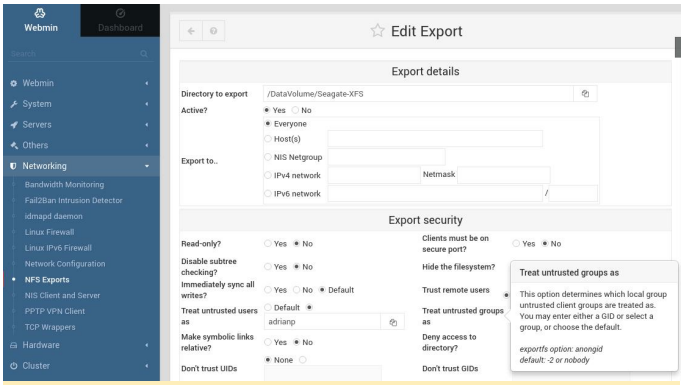


Figura 3 – Creando un recurso NFS

A la hora de crear recursos compartidos Samba/NFS, ten en cuenta la seguridad desde el principio. Samba se autentifica mediante un ID del usuario y una contraseña, pero NFS autentifica a los usuarios solo por IP. Si conoces qué dispositivos de tu red pueden tener acceso a los recursos compartidos específicos, detállalos en la configuración. Por ejemplo, un recurso compartido NFS podría exportarse a "Todos", pero el acceso aún puede estar limitado con iptables o /etc/hosts.allow y /etc/hosts.deny (que son utilizados por el módulo Wrappers TCP de Webmin).

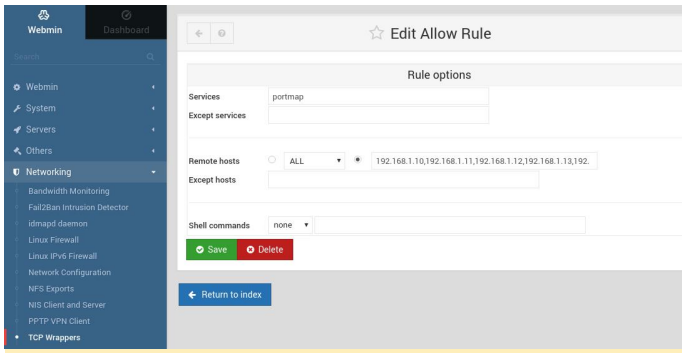


Figura 4 – Configuración de ./etc/hosts.allow para NFS con el fin de limitar el acceso a unos cuantos dispositivos

Utilizado con su configuración por defecto, Samba te dará un rendimiento decente, pero con los ajustes que detallo a continuación, extraídos de los foros ODROID, deberías obtener menos "pausas" en las transferencias de archivos de gran tamaño. Añade las siguientes líneas a la sección [global] de tu /etc/samba/smb.conf:

```
write cache size = 524288
getwd cache = yes
use sendfile = yes
min receivefile size = 16384
```

Instalar Owncloud

Owncloud es un servicio personal "en la nube" que te permite compartir archivos con personas a través de Internet. No voy a entrar en los detalles de la instalación, porque ya han sido tratados en un artículo anterior de la revista **http://bit.ly/2kgVZpn**, aunque hay algunas cuestiones que me gustaría puntualizar.

Lo primero de todo, la instalación es bastante simple en Ubuntu 16.04. Utilicé la guía **http://do.co/2bzxhxG** y logré ponerlo en funcionamiento en 10 minutos. Si tienes un nombre DNS (por ejemplo, DNS dinámico para tu hogar), debes dedicar un tiempo a obtener un certificado SSL válido desde Let's Encrypt (**http://bit.ly/1qmIXIY**) siguiendo los pasos que aparecen en **http://do.co/2bQpv4M**.

Básicamente debes instalar los siguientes requisitos previos antes de instalar OwnCloud

```
$ sudo apt-get install php
  libapache2-mod-php php-mcrypt
  php-mysql php-bz2 php-curl
  php-gd php-imagick php-intl
  php-mbstring php-xml php-zip
  mysql-server apache2
```

A continuación, instala el repositorio de OwnCloud para Ubuntu y actualiza los paquetes disponibles:

```
$ sudo curl
https://download.owncloud.org/download/
repositories/stable/Ubuntu_16.04/Release.key
| sudo apt-key add -
$ echo 'deb
https://download.owncloud.org/download/
repositories/stable/Ubuntu_16.04/ /'
| sudo tee
/etc/apt/sources.list.d/owncloud.list
$ sudo apt-get update
```

Por último, procede a instalar OwnCloud:

```
$ sudo apt-get install owncloud
$ sudo systemctl reload apache2
```

También necesitarás crear un usuario de base de datos para OwnCloud:

```
$ sudo mysql -u root
> CREATE DATABASE owncloud;
> GRANT ALL ON owncloud.* to
'owncloud'@'localhost' IDENTIFIED BY
'databasePassword';
> FLUSH PRIVILEGES;
> exit
```

Después de todo este trabajo, podrás iniciar sesión a través de la interfaz web https:///owncloud y finalizar la instalación. Puesto que OwnCloud debe ser accesible a través de Internet, debes tomarte un tiempo para reforzar tu instalación, tal y como se describe en **http://bit.ly/2jOTe1F**. En mi caso, quería ejecutar el servicio OwnCloud en un puerto diferente (para que los usuarios externos no tengan acceso a mis sitios internos), definir reglas iptables para permitir el acceso solo desde mi país (en base a datos de geo-ip) y configurar fail2ban para protegerme del cálculo automático de contraseñas.

Para ejecutar el host virtual OwnCloud en un puerto diferente, debes realizar algunos ajustes en tu configuración de Apache:

```
$ sudo cp /etc/apache2/sites-
available/default-ssl.conf
  /etc/apache2/sites-available/owncloud.conf
$ cd /etc/apache2/sites-available
$ sudo ln -s ../sites-available/owncloud.conf
  020-owncloud.conf
```

Después, edita /etc/apache2/sites-available/owncloud.conf y realiza los siguientes cambios:

Añade “Listen 8443” como primera fila. Cambia la definición de VirtualHost para usar el puerto 8443 en lugar del 443 (). Cambia DocumentRoot para que apunte a tu instalación de owncloud “DocumentRoot /var/www/owncloud”.

Cuando termines, puedes reiniciar el demonio de Apache y solo podrá acceder a tu instalación de OwnCloud desde https://: 8443/.

Para empezar con las reglas de firewall GeoIP, necesitarás tener las fuentes del kernel (o las cabeceras del kernel) disponibles. A continuación, procede a instalar los módulos adicionales de iptables con el siguiente comando:

```
$ sudo apt-get install
  xtables-addons-dkms
  xtables-addons-common
  xtables-addons-source
```

El paquete dkms no se instalará limpiamente porque algunos de los módulos no compilan con el kernel 4.9/4.14. Puedes deshabilitar los módulos que fallan y volver a compilar el resto fijando los siguientes parámetros con “n” en lugar de “m” en el archivo /var/lib/dkms/xtables-addons/2.10/build/mconfig:

```
$ sudo vi /var/lib/dkms/xtables-
addons/2.10/build/mconfig
build_ACCOUNT=n
build_LOGMARK=n
build_SYSRQ=n
build_pknock=n
build_psd=n
```

Después, tendrás que compilar manualmente el resto:

```
$ cd /var/lib/dkms/xtables-addons/2.10/build/
$ sudo autoconf
$ sudo ./configure
$ sudo make
$ sudo make install
```

Antes de usar el módulo geoip, necesitarás iniciar la base de datos geoip (el prefijo para la asignación del país). Es posible que tengas que repetir este paso de vez en cuando para beneficiarte de los últimos datos:

```
$ sudo apt-get install libtext-csv-xs-perl
$ sudo mkdir /usr/share/xt_geoip
$ sudo /usr/lib/xtables-addons/xt_geoip_dl
$ sudo /usr/lib/xtables-addons/xt_geoip_build
-D /usr/share/xt_geoip
/root/GeoIPCountryWhois.csv
```

Todo lo que queda por hacer ahora es crear y probar las reglas iptables para permitir unicamente el tráfico que deseas que llegue a tu configuración de owncloud. Una posible regla podría ser:

```
$ sudo iptables -N geo-owncloud
$ sudo iptables -A INPUT -p tcp -m tcp --dport
8443 -j geo-owncloud
$ sudo iptables -A geo-owncloud -s
192.168.1.0/24 -j ACCEPT
$ sudo iptables -A geo-owncloud -m geoip ! --
src-cc RO -j DROP
```

No olvides guardar tus reglas y aplicarlas al inicio (ya sea con iptables-save o con webmin). Puedes encontrar más detalles sobre geoip en **http://bit.ly/2jnwUJD**.

Configurar fail2ban no es muy complicado una vez que sigues el tutorial **http://bit.ly/2kipXxn**. Recuerda instalar fail2ban primero (y probarlo con algunas credenciales falsas):

```
$ sudo apt-get install fail2ban
```

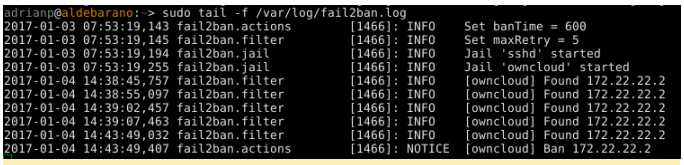


Figura 5 – Fail2Ban haciendo su trabajo con intentos de inicio de sesión fallidos

Puesto que hemos añadido un puerto especial para owncloud, tendremos que retocar la configuración de fail2ban para dar cuenta de ello. Edite

/etc/fail2ban/jail.local y agrega “port 8443” a la línea del puerto y reinicia fail2ban:

```
$ sudo vi /etc/fail2ban/jail.local
port = http,https,8443
$ sudo service fail2ban restart
```

Para levantar manualmente la prohibición de una dirección IP incluida en la lista negra, puedes ejecutar el siguiente comando:

```
$ sudo fail2ban-client set owncloud unbanip
172.22.22.2
```

Asignar tareas a CPU específicas

El ODROID-XU4 viene con dos tipos de núcleos de CPU: 4 núcleos pequeños de baja potencia que son más adecuados para tareas en segundo plano y 4 núcleos grandes que están diseñados para soportar tareas más pesadas. El kernel oficial viene con un programador de tareas “mágico” de Samsung que conoce la verdadera potencia del procesador, y puede cambiar tareas de los núcleos pequeños a los núcleos grandes cuando la carga de trabajo es alta. Puede haber casos especiales en los que desee ejecutar tareas específicas exclusivamente en los núcleos grandes o pequeños, ya sea para maximizar el rendimiento o para minimizar la temperatura. Podemos usar el uso de cgroups como se indica en http://bit.ly/2jP6KlU .

“Cgroups” es una característica de los kernels modernos que permite la asignación de recursos para varios procesos. En nuestro caso, necesitaremos el cgrupo “cpuset” para crear un grupo “littlecores” y un grupo “bigcores”. Cada grupo obligará a los procesos a ejecutarse en núcleos específicos determinando la afinidad. De modo que, los littlecores tendrán las cpus 0-3 y bigcores las 4-7. Afortunadamente, crear los cgroups es muy fácil:

```
# mkdir -p /sys/fs/cgroup/cpuset/littlecores
  /sys/fs/cgroup/cpuset/bigcores
# echo "0-3" > /sys/fs/cgroup/cpuset/
littlecores/cpuset.cpus
# echo "0"> /sys/fs/cgroup/cpuset/
littlecores/cpuset.mems
# chmod -R 777 /sys/fs/cgroup/cpuset/
littlecores
# echo "4-7"> /sys/fs/cgroup/cpuset/
bigcores/cpuset.cpus
# echo "0"> /sys/fs/cgroup/cpuset/
bigcores/cpuset.mems
# chmod -R 777 /sys/fs/cgroup/cpuset/
bigcores
```

Lamentablemente, los comandos solo durarán hasta el próximo reinicio. Así que creamos un servicio para configurarlos en el arranque tan pronto como sea posible:

```
$ sudo wget -O
/etc/systemd/system/cpuset.service
https://raw.githubusercontent.com/mad-ady/
odroid-ODROID-XU4-
optimizations/master/cpuset.service
$ sudo systemctl enable cpuset
$ sudo systemctl start cpuset
```

Llegados a este putno, los cgroups están creados, pero nadie los usa activamente. Para iniciar manualmente un proceso en un cgroup específico, puede usar cgexec:

```
$ sudo apt-get install cgroup-tools
$ cgexec -g cpuset:bigcores sysbench --
test=cpu
--cpu-max-prime=100000 --num-threads=8 run
```

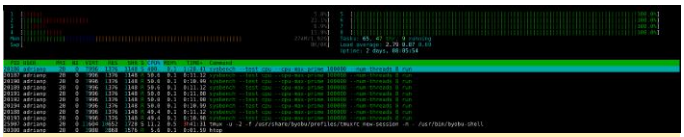


Figura 6: 8 subprocesos sysbench son forzados a ejecutarse en 4 núcleos específicos

Nos encontramos a mitad de camino. Ahora necesitamos indicar qué procesos específicos se ejecutarán en los núcleos pequeños y el qué procesos en los núcleos grandes. Aquí es donde necesitas hacer una lista y decidir lo que quieres. Comienza con una lista de servicios activos de webmin (System -> Bootup and Shutdown) y desactiva todo lo que no vayas a utilizar. En mi caso, desactivé los siguientes servicios: ModemManager, NetworkManager-wait-online, NetworkManager, accounts-daemon, alsa-restore, alsa-state, apport, apport-forward.socket, bluetooth, cups-browsed, cups.path, cups.service, cups.socket, lightdm, lxc-net, lxc, lxcfs, plymouth *, rsync, saned, speech-dispatcher y whoopsie.

Tendrás que editar los scripts de inicio para los servicios que desees y hacer que agreguen su PID al cgroup correcto. Una vez que el proceso principal (y sus hijos) forman parte del cgroup correcto, cualquier hijo nuevo heredará el cgroup. Mi plan era agregar cosas como MySQL, Apache, Samba, NFS e incluso Webmin al grupo grande y cosas como SSH (y toda mi actividad del intérprete de comandos), cron, Munin y Transmission al grupo pequeño. Esto permite que los procesos que están involucrados en la operatividad del NAS sean más ágiles, mientras que otras tareas pueden ejecutarse adecuadamente en los núcleos pequeños. Si también está utilizando la GUI X11, es posible que también quieras agregar lightdm al grupo "bigcore".

Hay dos tipos de scripts de inicio: scripts nativos systemd y sys-v heredado (/etc/init.d/). Cuando edites un script systemd (por ejemplo, nfs-mountd.service), deberás añadir algo como esto a la sección [Servicio]:

```
ExecStartPost=-/bin/sh -c 'echo $MAINPID | tee -a /sys/fs/cgroup/cpuset/bigcores/tasks'
```

Cuando editamos un script sys-v, el tema se complica algo más. Necesitaras encontrar la función de inicio, extraer el PID (s) del proceso recién iniciado y agregarlo a la lista de tareas. A continuación, se muestra un ejemplo donde se cambia el script de inicio de Apache:

```
pidof apache2 | tr " " ""| xargs -0 -n1 | sudo tee -a /sys/fs/cgroup/cpuset/bigcores/tasks
```

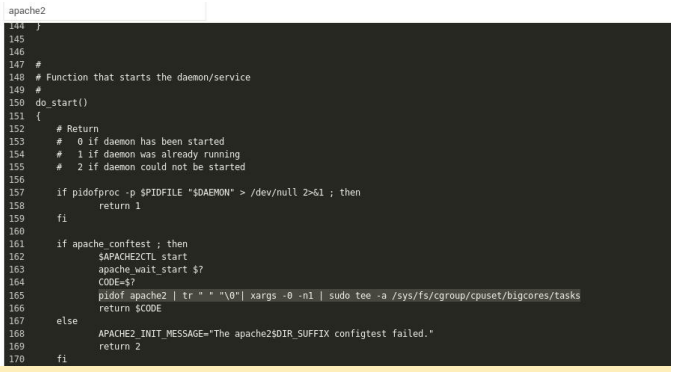


Figura 7 – Cambiando la configuración de inicio de Apache

Asegúrate de reiniciar cada servicio después de cambiarlo y verifica que el PID del proceso esté en el archivo de tareas cpuset correcto. Haz un reinicio completo del sistema y vuelve a comprobarlo después de reiniciar. Si esto te parece demasiado complicado y estás usando el kernel 4.9, hay una forma de hacer "trampa" y ejecutar todas las tareas en los núcleos grandes. Simplemente puedes establecer la afinidad de systemd y todos sus procesos hijos lo heredarán. La afinidad puede controlarse mediante el parámetro CPUAffinity en /etc/systemd/system.conf, pero ten en cuenta que esto te llevará a desperdiciar ciertos núcleos de la CPU.

Duración del disco

Para prolongar la vida útil de tu (s) disco (s), es posible que desees detenerlos tras un período de inactividad. Si está utilizando SSD, puede saltarte esta sección porque solo es aplicable a discos mecánicos antiguos. Los discos pueden recibir un comando "stop" para detener su giro desde un controlador interno, desde el puente USB-SATA o directamente desde el sistema operativo. Sin embargo, a veces los controladores no están sincronizados correctamente provocando que el comando "stop" nunca llegue. Esto hace que el disco siga girando, lo cual genera mucho calor y puede hacer que el disco falle antes de lo normal.

La forma habitual de gestionar esto es decirle al disco que se detenga después de un período de inactividad, lo cual se puede hacer con hdparm:

```
$ sudo apt-get install sdparm hdparm
```

Para configurar manualmente el disco para que se detenga tras 10 minutos de inactividad, puede ejecutar el siguiente comando:

```
$ sudo hdparm -S 120 /dev/sda
```

Si recibes errores (como "bad/missing sense data"), es posible que hdparm no le pueda ayudar con este disco.

Para gestionar la movilidad del disco, sería mejor dejar que udev ejecute el comando después de que se haya conectado un disco. Dado que los discos pueden tener diferentes roles, es posible que desees distintos temporizadores de suspensión (por ejemplo, un disco para copias de seguridad debería suspenderse/detenerse antes, otro que está activo

debería suspenderse/deternerse más tarde). Yo decidí configurar la regla UDEV en función del número de serie del disco. Puede obtener este número de serie buscando en dmesg cuando conectes un disco:

```
[1885221.800435] usb 4-1.3: Product: My
Passport 0730
[1885221.800436] usb 4-1.3: Manufacturer:
Western Digital
[1885221.800437] usb 4-1.3: SerialNumber:
57584431414130563632393 7
```

Para configurar la regla, crea un archivo como este y vuelve a cargar udev:

```
$ sudo vi /etc/udev/rules.d/90-disk.rules
ACTION=="add", ENV{DEVNAME}=="/dev/sd?",
SUBSYSTEM=="block",
ENV{ID_SERIAL_SHORT}=="5758443141413056363239 3
7", RUN+="/sbin/hdparm -S 120 $env{DEVNAME}"
$ sudo udevadm control -R
```

Si hdparm no puede poner su disco en modo suspensión, intentalo con otras alternativas como sdparm, que puede enviar un comando SCSI a tu disco, como el que ordena que se detenga en ese preciso instante:

```
$ sudo sdparm -C stop /dev/sda
```

Hay herramientas como hd-idle (**http://bit.ly/2j3zWSk**) o scripts periódicos que puedes ejecutar para poner tu disco en modo de suspensión. En mi caso, no funcionaron, pero asegúrate de probarlos manualmente antes de decantarte por una u otra solución. Aquí tienes un script manual que comprueba la actividad de un disco (identificado por el UUID de una partición) en una ventana de 10 segundos, y si no hay actividad (datos transferidos), usa sdparm para detener el disco. Puedes ejecutarlo a través de cron:

```
$ sudo wget -O /usr/local/bin/hdd-idle.sh
http://bit.ly/2k6LK7Y
$ sudo chmod a+x /usr/local/bin/hdd-idle.sh
$ sudo /usr/local/bin/hdd-idle.sh "4283-E975"
```

Debe tener en cuenta que existen herramientas y servicios que activarán tu disco periódicamente, incluso si no se transfieren datos. Herramientas como smartctl (de smartmontools) y smartd. El servicio smartd comprueba periódicamente el estado del disco, y si no está configurado correctamente, puede mantener activo tu disco innecesariamente. Puede consultar el hilo **http://bit.ly/2kh6b17** sino logras encontrar que es lo que mantiene tu disco activado. Deberías poder dedudir el estado del disco a partir de este comando: $ sudo smartctl -d sat -n standby -a /dev/sda

Si sale con un error, tu disco todavía está en modo de espera y debería haber sido desactivado.

Rendimiento del disco flash

Una cosa más que hay que tener en cuenta a la hora de usar el almacenamiento flash (eMMC o SSD) es que éste necesita un nivelado periódico para mantener su velocidad. Básicamente, para escribir en un bloque de almacenamiento, primero debe borrarlo y esto le lleva más tiempo que escribir en él. Los sistemas de archivos normales no hacen esto cuando se eliminan datos, de modo que después de un tiempo el rendimiento del disco disminuye significativamente. Para "reavivar" el disco, la operación de nivelado informa al controlador del disco que borre todos los bloques vacíos, restaurando así las velocidades de escritura. La operación de nivelado debe ser compatible con el sistema de archivos y el controlador de disco. De nuevo, usar cron una vez a la semana para ejecutar fstrim puede evitarte ralentizaciones a largo plazo:

```
$ sudo crontab -e
#trim the eMMC once a week
15 0 0 * *      /sbin/fstrim / >/dev/null 2>&1
```

Regulador

El rendimiento y el calor también dependen directamente del regulador que estés utilizando para la CPU. Usar y Mantener la opción "performance" te proporciona un rendimiento superior, pero también genera mucho calor. En mis pruebas, la mejor combinación fue un regulador "ondemand" modificado basado en las recomendaciones de http://bit.ly/2jfaDjw. Para activarlo, asegúrate de seleccionar governor = ondemand en

/media/boot/boot.ini, y configura el resto de parámetros dentro de /etc/rc.local (prueba los comandos antes). Los siguientes comandos funcionan para un kernel 4.9/4.14 y pueden variar para el kernel 3.10:

```
$ sudo vi /etc/rc.local
echo 1 >
/sys/devices/system/cpu/cpufreq/ondemand/io_is
_busy
echo 10 >
/sys/devices/system/cpu/cpufreq/ondemand/sampl
ing_down_factor
echo 80 >
/sys/devices/system/cpu/cpufreq/ondemand/up_th
reshold
```

Con la configuración anterior, la CPU incrementará la frecuencia más rápidamente y se tendrá en cuenta el uso de E/S, haciendo que las tareas intensivas de E/S influyan en la frecuencia de la CPU. Esto te permite tener un gran rendimiento cuando sea necesario y bajar la temperatura de la CPU cuando está inactiva. Con mi uso, los pequeños núcleos se mantienen inactivos alrededor de los 300MHz mientras que los núcleos grandes permanecen inactivos a 200MHz.

Rendimiento de la red – MTU

Si tiene una red Gigabit con el cableado adecuado, puedes aumentar la MTU (Unidad máxima de transmisión) en la red integrada del ODROID-XU4. Esto te permitirá enviar paquetes más grandes con menos sobrecarga y generar menos interrupciones en el receptor. Sin embargo, para beneficiarte de ello, deberás contar con dispositivos de red (switches/routers) y dispositivos finales compatibles con frames Jumbo. En el mejor de los casos, los frames Jumbo deberían estar habilitados en todos los dispositivos de red de tu LAN; de lo contrario, es posible que veas que el tráfico desciende o incluso que los dispositivos no puedan enviar grandes cantidades de tráfico entre ellos. Por ejemplo, SSH funciona porque usa paquetes pequeños, pero transferir una página web o un archivo detiene la conexión. Si decides habilitar los frames jumbo, el valor MTU mágico del ODROID-XU4 es 6975 (**http://bit.ly/2jP9zDl**). Puedes habilitarlos en el ODROID-XU4 dentro de /etc/rc.local:

```
$ sudo vi /etc/rc.local
#MTU
/sbin/ifconfig eth0 mtu 6975 up
```

Transferencias más rápidas sobre sshfs/scp/sftp

Como SSH es un protocolo muy flexible que permite hacer túnel y transferir archivos, sería muy acertado usarlo a toda velocidad. Si intentas hacer una transferencia de copia segura (scp) en un ODROID-XU4 con el proceso sshd vinculado a los pequeños núcleos, lograrás una velocidad máxima de aproximadamente 15MB/s. Si vincula el proceso sshd a los núcleos grandes llegaras a los 40 MB/s. Si eres atrevido y no te importa sacrificar algo de seguridad, puedes alcanzar los 50MB/s bajando el algoritmo de encriptación utilizado. Yo lo hice iniciando una instancia sshd diferente (en el puerto 2222) con diferentes configuraciones:

```
$ sudo wget -O /etc/systemd/system/ssh-
big.service
https://raw.githubusercontent.com/mad-ady/
odroid-xu4-optimizations/master/ssh-
big.service
$ sudo wget -O /etc/ssh/sshd_config_big
https://raw.githubusercontent.com/mad-ady/
odroid-xu4-
optimizations/master/sshd_config_big
$ sudo systemctl enable ssh-big
$ sudo systemctl start ssh-big
```

Para montar o transferir un archivo utilizando este nuevo servicio ssh, necesitaras especificar explícitamente el cifrado (esto está desactivado por defecto porque está considerado poco seguro). Puede hacerlo con una entrada en ~/.ssh/config en el cliente:

```
Host odroid-big
Hostname odroid-ip
Port 2222
Ciphers arcfour
Compression no
```

Para transferir archivos, simplemente puede usar el siguiente comando:

```
$ scp bigfile odroid-big:/remote/path
```

Ajustar los tiempos de espera de systemd

Puede ser bastante irritante el tener que esperar impacientemente a que systemd finalice, sobretodo si es algo que nunca terminará. Puede modificar los tiempos de espera de systemd variando la configuración global de los tiempos de espera en /etc/systemd/system.conf:

```
DefaultTimeoutStartSec=20s
DefaultTimeoutStopSec=10s
```

Ten en cuenta que algunos servicios (como el de redes) establecen tiempos de espera explícitos que deberás cambiarlos también:

```
$ sudo vi /etc/systemd/system/network-
online.target.wants/
networking.service
TimeoutStartSec=30sec
```

Rendimiento

Estas son algunas métricas de rendimiento que puedes esperar con los ajustes anteriores y una red Gigabit. El cliente es un ODROID-C2 que ejecuta Ubuntu 16.04, mientras que el servidor es el ODROID-XU4. Las instrucciones de descarga y subida son relativas al ODROID-XU4. El disco conectado al ODROID-XU4 tiene una velocidad de escritura de 110 MB/s. Las transferencias de archivos se realizo con un archivo de 8GB lleno de ceros (dd if = /dev/zero of=zero bs=1M count=8000 conv=fsync). Ten en cuenta que parte del rendimiento también depende de tu cliente. Pudes lograr un rendimiento mejor con un PC Linux mas que con un ODROID-C2 como cliente.

Conociendo a un ODROIDian: Ernst Mayer, un Extraordinario Matemático

April 1, 2018 By Rob Roy Conociendo un ODROIDian

Por favor, háblanos un poco sobre ti. He estado viviendo y trabajando en Silicon Valley durante aproximadamente 20 años, realizando primero trabajos de codificación y algorítmicos para varias empresas tecnológicas y luego para empresas más grandes. La mayor parte de ese trabajo estaba relacionado con software EDA para el diseño de chips. En realidad, estoy medio retirado (en el sentido de que todavía trabajo, pero principalmente en mis propios proyectos de investigación, no por un salario) desde que me despidieron de mi última empresa hace 6 años. Actualmente vivo en Cupertino, el corazón del país de Apple, aunque nunca he trabajo allí. Es agradable estar cerca de las colinas costeras, aunque los precios de los inmuebles y de los alquileres son realmente altos. Mi hermana y su familia (marido e sus hijos gemelos de 9 años) viven en North Bay, así que los veo con bastante frecuencia. En realidad, vengo de ciencias, pero no de ciencias de la computación: mis títulos de posgrado de la Universidad de Michigan son de Ingeniería Aeroespacial y Matemáticas. Mi tesis doctoral se centraba en la mecánica teórica de fluidos, específicamente en los flujos de vórtice. Muchas ecuaciones diferenciales, matemáticas aplicadas y análisis numérico. Mis conocimientos en codificación que salieron de la universidad eran de informática científica, es decir, Fortran. Aprendí por mi cuenta los principios básicos de C y C ++ después de mudarme a Silicon Valley, y adquirí en el trabajo la mayor parte del resto de conocimientos que necesitaba saber sobre estos lenguajes y algoritmos de estilo CS y estructuras de datos.

Figura 1 – Ernst visitando las Montañas Rocosas canadienses en 1986

¿Cómo empezaste con los ordenadores? En un contexto de trabajo con algoritmos y programación, terminé haciendo ambas cosas durante mi trayectoria profesional y con una continua investigación. Es importante tener en cuenta que, recordando mis primerias experiencias de la universidad, era uno de los frikis de código menos prometedores de la historia. Era un estudiante novato de primer año de la Universidad de Michigan en el otoño de 1981, y como tal, pertenezco a una de las últimas graduaciones en ingeniería de programas que sufrió la programación de Fortran para cursos de ingenieros, tal y como estaba constituido por aquel entonces. El centro de cálculo estaba ubicado en una mediocre estructura que se encontraba literalmente debajo de un paso elevado de peatones, y que formalmente fue nombrado Edificio de la Universidad del Norte, pero conocido por todos por sus siglas, NUBS. Todo estaba basado en una configuración de tiempo compartido con coste escalonado basada en un macro-ordenador estándar, que era un sistema Amdahl. Estoy seguro que fue creado como una alternativa de coste más razonable a la serie IBM 360 líder en el mercado. El verdadero problema, como suele pasar en estos casos, reside en el software: un compilador de sistema no IBM significaba precisamente tener un compilador que no era de IBM, y aunque desconozco qué compiladores alternativos podían haber tenido Amdahl Corp. para ofrecer, sé que los usuarios del sistema tenían que cargar con un compilador un tanto experimental de Waterloo U. en Canadá. El problema era que los mensajes de error del mismo eran tan crípticos (a menudo simplemente códigos de error hexadecimales bastante confusos, incluso sin un número de línea de programa), que para nosotros los novatos era un sistema de mensajería de error de sintaxis binaria: 1 = ” hay> = 1 errores de sintaxis en tu código, no me preguntes dónde, así que buena suerte con la búsqueda”, y 0 = “¡felicitaciones! hay 0 errores de sintaxis en tu código, aquí están tus resultados (probablemente incorrectos) “. Incluso esto hubiera sido pesado pero factible, pero el segundo gran problema era que había muy pocos terminales de línea de impresión y aún menos terminales interactivos disponibles, todos estaban ocupados al 100% por estudiantes de posgrado en ciencias de la computación, así que estábamos limitados a las máquinas de tarjetas perforadas, para las cuales también había colas de espera. Así que una vez que finalmente conseguías un asiento en uno y transcribías todos tus esfuerzos iniciales escritos a mano en un programa para su asignación en tu pequeño pack de tarjetas de papel en forma de baraja, tenías que abandonar tu asiento, llevarte el pack de tarjetas a la máquina lector más cercana, quizás tenías que espera un poco en la cola que se formaba allí, luego te dirigías a la ventana del dispensador de papel impreso para recuperar tu listado de programas y, si tenías suerte, obtenías tus resultados. ¿Tienes un cripto-mensaje de error de sintaxis? Pasas un tiempo analizando tu lista de programas, identificas los posibles errores, después vuelves a la cola de espera de la sala de máquinas de la tarjeta perforada. ¿Sin errores de sintaxis, pero con errores en tus resultados? Más de lo mismo. El insulto final era que a los novatos se les asignaba una cantidad ridículamente pequeña de créditos en cuenta para el tiempo compartido. Creo recordar que eran 2$ de crédito para los proyectos de todo un semestre, sin opción de añadir más. Dado el sistema de precios que estaba vigente, la única forma de convertir dicha cantidad en un número remotamente razonable de ciclos de depuración de los que hemos descrito anteriormente era usar las instalaciones por la noche, cuando los precios estaban en su nivel más bajo. El resultado era que incluso la asignación de programación de 50 líneas más simple casi

invariablemente se convertía en una horrorosa sesión de trabajo durante toda la noche.

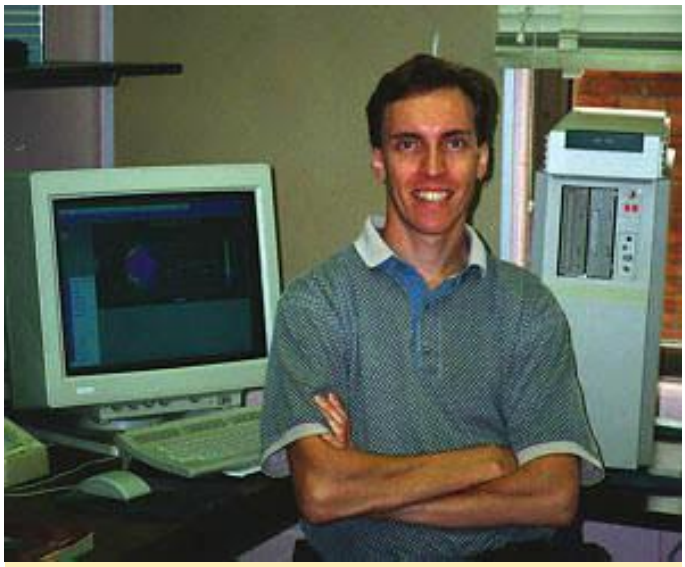


Figura 2 – Ernst en su oficina en la Case Western Reserve University en 1997, frente a una estación de trabajo DEC Alpha, la primera y verdadera arquitectura RISC de 64 bits, que era un excelente sistema en su día, aunque su ODROID-C2 tiene diez veces más de potencia de calculo

Como consecuencia, durante el resto de mis días como estudiante universitario, el único tipo de codificación que llegue a hacer fue en mi fiel calculadora programable HP-41C. Si por aquel entonces alguien del futuro hubiera llegado y me hubiera dicho que terminaría escribiendo (casi completamente desde cero) y manteniendo un programa que consta de aproximadamente medio millón de líneas de código, le habría dicho que estaba loco. Por supuesto, el destino, como lo hace a menudo, tenía otros planes. Mientras estaba en la facultad, me saqué un dinero extra trabajando aproximadamente unas 20 horas a la semana realizando trabajos de carpintería y mantenimiento para un casero local, y pasé tantas horas como me quedaban disponibles realizando actividades al aire libre: Escalada en rocas y viajes de montañismo en verano, ciclismo, artes marciales.

En el verano de 1987, después de haber finalizado mi licenciatura estaba preparando para empezar un programa de doctorado en dinámica de fluidos experimental cuando durante una sesión de ciclismo de montaña sufrí un desagradable derrame celebrar y terminé con el cuello roto y una parálisis desde el pecho hacia abajo. Así que me hice con un laboratorio experimental lleno de equipos; mi trabajo con las matemáticas y la computación yacía dentro. Afortunadamente, los laboratorios de computación de la universidad se habían trasladado a estaciones de trabajo y PC, así que pude hacer la mayoría de la codificación de mi investigación de posgrado en estaciones de trabajo DEC Vax, con software de calidad, una experiencia muy diferente a la que tenía como estudiante de primer año.

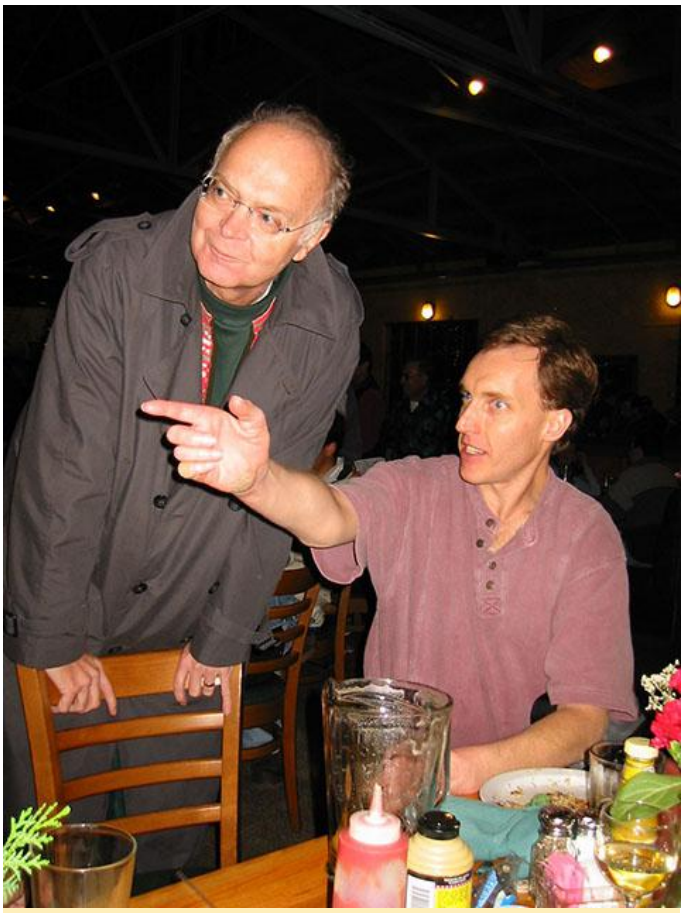


Figura 3 – Ernst con Stanford Donald Knuth y un grupo de compañeros Mersenners en Mountain View Tied House para celebrar el descubrimiento de la 39ª Mersenne prime, M (13466917), diciembre de 2001

¿Qué te atrajo a la plataforma ODROID? Como tome nota del artículo de números primos del mes pasado en ODROID Magazine, después de pasar gran parte de los últimos cinco años escribiendo código ensamblador para los diversos conjuntos de instrucciones aritméticas vectoriales SIMD x86 (SSE2, AVX, AVX2 + FMA3, AVX512), el último año también estaba considerando realizar una primera incursión para añadir soporte SIMD a una familia de procesadores que no fueran x86, y la compatibilidad con vectores de 128 bits de la familia de CPU ARMv8

encajaba perfectamente. Tras realizar algunas tareas con una plataforma de desarrollo de bajo coste, pero razonablemente de alto rendimiento para esta tarea de codificación, el ODROID-C2 surgió como la mejor opción. Obtuve un 50% más de rendimiento con mi código en el C2 que en el Raspberry Pi3. También he recibido los resultados de las pruebas de rendimiento de una versión preliminar del ODROID N1 gracias a un usuario del foro ODROID que fue seleccionado como uno de los analistas beta para el N1, y parece muy prometedor, utilizando las dos CPU del N1 (dual-core Cortex a72 y quad-core Cortex a53), se consigue más o menos el mismo rendimiento que el "pequeño" socket a53 de un C2 (que está basado en la misma CPU quad-core), y el 'big' CPU dual-core a72 ofrece aproximadamente 1.5 veces ese rendimiento. La ejecución de los trabajos en ambos sockets se reduce simultáneamente aproximadamente un 10% en cada uno de esos socket, por lo que no conseguimos 2,5 veces el rendimiento total del C2, pero sigue siendo más del doble que el del C2. De modo que, el C2 definitivamente fue una buena opción como mi primer ODROID.

¿Cómo usas tus ODROID? El año pasado, los 4-5 meses posteriores a la compra de mi C2 me dedique a la codificación del ensamblaje online para tareas pesadas, la depuración y a reajustar el rendimiento. Desde que liberé el código ARMv8 he usado mi C2 más o menos de la misma forma que espero que lo hagan los usuarios de mi código, indagando primero en el proyecto de computación distribuida GIMPS. (En caso de que alguien se lo esté preguntando, no elegí el acrónimo del proyecto ni cometí ninguna falta). También es muy útil tener una máquina con una versión diferente de Linux y GCC instalada ya sea mi Macbook o mi gran sistema Intel quad-core, en caso de que necesite localizar un problema de compilación que parece estar relacionado con el compilador o la versión del sistema operativo.

¿Cuál es tu ODROID favorito y por qué? El ODROID-C2 por supuesto, al menos hasta que el N1 salga a la venta.

¿Qué innovaciones te gustaría ver en futuros productos Hardkernel? Soy un gran consumidor, así que supongo que mi respuesta se reduce a "¡más sockets!". En particular, estoy interesado en cuántas placas ODROID necesitaría para competir con un sistema quad Intel de gama alta, y cómo los respectivos precios de esas opciones de rendimiento similar se compararían. Basándome en el rendimiento relativo de mi Intel quad Haswell y ODROID-C2 y N1, necesitaríamos unos 20 C2 para poder competir con el Haswell y alrededor de 10 N1. Una vez que llegásemos a "alrededor de 10", este tipo de paquete compacto multi-placa ya sería un poco excesivo. El precio al consumidor estimado para el N1 sigue siendo un poco más alto de lo que a uno le gustaría, pero no mucho. De todos modos, estos son los típicos sueños que tiene este ODROIDer. También creo que los micro PC Linux son una excelente manera de que los niños se interesen por los ordenadores de una forma que evite el efecto "Recinto cerrado" que tiene los PC con sistemas operativos propietarios; Pienso que tal vez algún tipo de iniciativa a nivel de educación que pusiera estos sistemas en manos de niños con bajos recursos escolares valdría la pena que la gente de Hardkernel analiza. Este es el tipo de cosas que podría atraer subvenciones del gobierno o de fundaciones privada para su patrocinio.

¿Qué hobbies e intereses tienes aparte de los ordenadores? Siempre he sido una persona a la que le gusta trabajar no solo con la cabeza sino también con las manos. Una cosa que le falta a las matemáticas y a la codificación es la satisfacción tangible que conlleva la construcción física de algo, de modo que siempre me gusta tener algún tipo de proyecto manual que satisfaga esa necesidad. Hace un par de años construí un banco de trabajo realmente resistente con madera de salvamento, en su gran mayoría pallets ya desechados que eran usados a nivel industrial para el envío de equipos informáticos. El proyecto de este invierno fue construir un soporte de exposición para un gran meteorito de hierro (45 kg) que compré hace algunos años, a partir de una base de piedra caliza travertino cubierta con un bloque de piedra arenisca natural perforada para sostener tres barras de acero que actuasen a modo de trípode para sujetar el meteorito. La perforación resultó ser la parte más difícil: uno espera que la arenisca sea bastante blanda

y fácil de trabajar, pero este bloque era arena que aparentemente se había erosionado a partir de algún tipo de mineral duro, lo conseguí gracias a una broca de punta hueca recubierta de diamantes y varias horas de continua y fuerte presión con un taladro y agua para lubricar las partes. ¡Las pase canutas tomando mucho ibuprofeno esa semana!

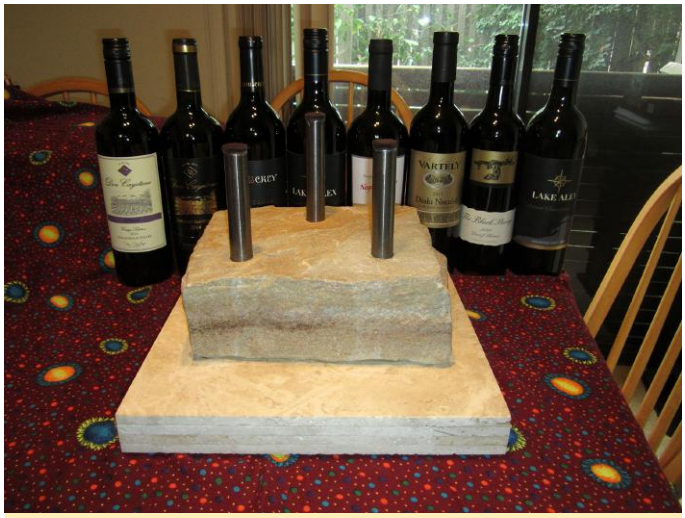
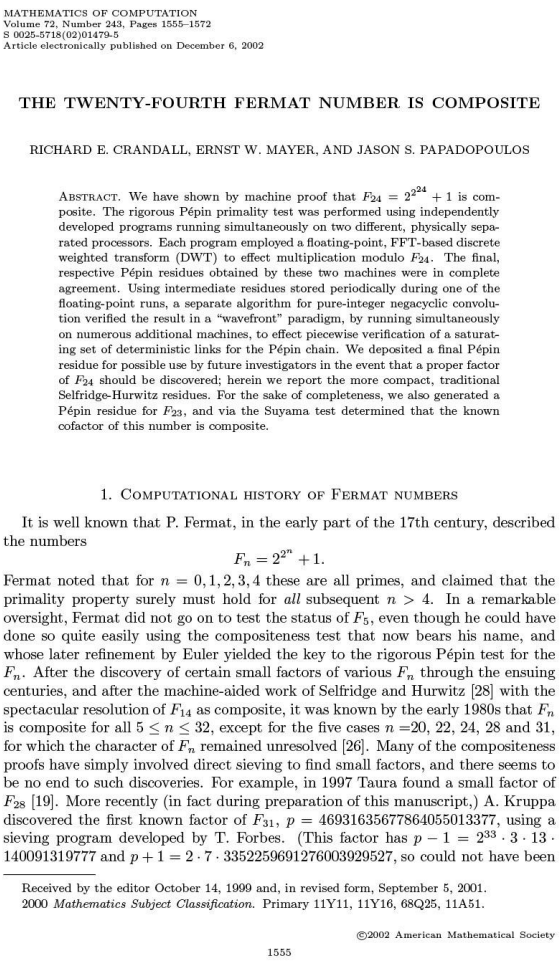


Figura 4 – Ernst actualmente está construyendo un soporte de exposición para su meteorito de hierro

¿Qué consejo le darías a alguien que quiere aprender más sobre programación y matemáticas? Buscar un problema que realmente le interese y que le pueda servir como medio de aprendizaje para estos temas. He tenido varios de estos en mi carrera, ya que mi investigación de doctorado tenía aspectos de ecuaciones diferenciales, análisis asintótico, teoría de perturbaciones, álgebra lineal y sistemas propios. Mi trabajo con números primos implica aritmética de grandes números enteros y algoritmos de procesamiento de señales, código ensamblador aritmético vectorial, además de una fascinante y rica historia sobre algunos de los lumbreras más brillantes de la historia de las matemáticas. El mundo de la ciencia está lleno de problemas así de interesantes; Créeme, sabrás cuando algo así te atrape. Tener tiempo en nuestro moderno mundo lleno de distracciones y gobernado por el dinero para conseguirlo, este es quizás el problema más complicado.



MATHEMATICS OF COMPUTATION
Volume 72, Number 243, Pages 1555–1572
S 0025-5718(02)01479-5
Article electronically published on December 6, 2002

THE TWENTY-FOURTH FERMAT NUMBER IS COMPOSITE

RICHARD E. CRANDALL, ERNST W. MAYER, AND JASON S. PAPADOPOULOS

ABSTRACT. We have shown by machine proof that $F_{24} = 2^{2^{24}} + 1$ is composite. The rigorous Pépin primality test was performed using independently developed programs running simultaneously on two different, physically separated processors. Each program employed a floating-point, FFT-based discrete weighted transform (DWT) to effect multiplication modulo F_{24}. The final, respective Pépin residues obtained by these two machines were in complete agreement. Using intermediate residues stored periodically during one of the floating-point runs, a separate algorithm for pure-integer negacyclic convolution verified the result in a "wavefront" paradigm, by running simultaneously on numerous additional machines, to effect piecewise verification of a saturating set of deterministic links for the Pépin chain. We deposited a final Pépin residue for possible use by future investigators in the event that a proper factor of F_{24} should be discovered; herein we report the more compact, traditional Selfridge-Hurwitz residues. For the sake of completeness, we also generated a Pépin residue for F_{23}, and via the Suyama test determined that the known cofactor of this number is composite.

1. COMPUTATIONAL HISTORY OF FERMAT NUMBERS

It is well known that P. Fermat, in the early part of the 17th century, described the numbers

$$F_n = 2^{2^n} + 1.$$

Fermat noted that for $n = 0, 1, 2, 3, 4$ these are all primes, and claimed that the primality property surely must hold for *all* subsequent $n > 4$. In a remarkable oversight, Fermat did not go on to test the status of F_5, even though he could have done so quite easily using the compositeness test that now bears his name, and whose later refinement by Euler yielded the key to the rigorous Pépin test for the F_n. After the discovery of certain small factors of various F_n through the ensuing centuries, and after the machine-aided work of Selfridge and Hurwitz [28] with the spectacular resolution of F_{14} as composite, it was known by the early 1980s that F_n is composite for all $5 \le n \le 32$, except for the five cases n =20, 22, 24, 28 and 31, for which the character of F_n remained unresolved [26]. Many of the compositeness proofs have simply involved direct sieving to find small factors, and there seems to be no end to such discoveries. For example, in 1997 Taura found a small factor of F_{28} [19]. More recently (in fact during preparation of this manuscript,) A. Kruppa discovered the first known factor of F_{31}, $p = 46931635677864055013377$, using a sieving program developed by T. Forbes. (This factor has $p - 1 = 2^{33} \cdot 3 \cdot 13 \cdot 140091319777$ and $p + 1 = 2 \cdot 7 \cdot 3352259691276003929527$, so could not have been

Received by the editor October 14, 1999 and, in revised form, September 5, 2001.
2000 *Mathematics Subject Classification.* Primary 11Y11, 11Y16, 68Q25, 11A51.

©2002 American Mathematical Society

1555

Figura 5 – Primera contribución de Ernst al campo de la teoría de números computacionales en 2002