

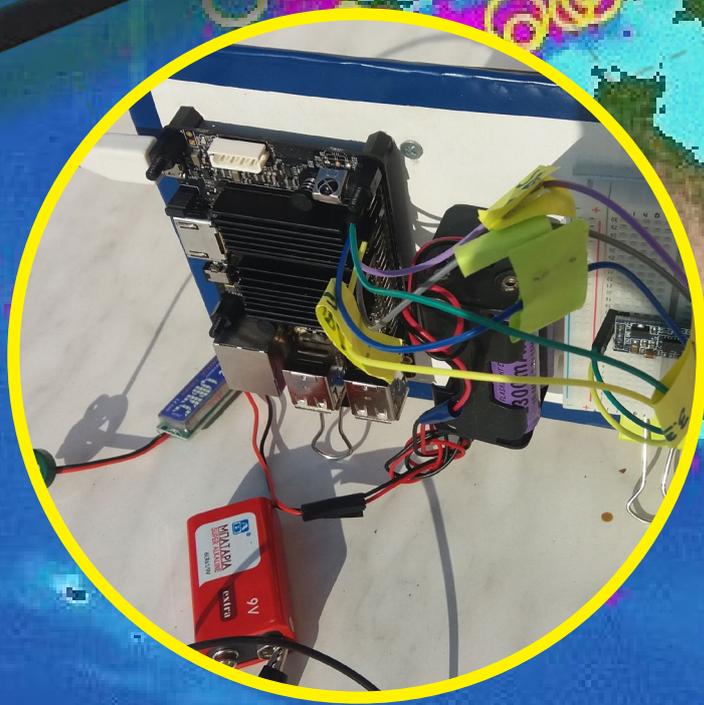
# ODROID

Año Cuatro  
Núm. #43  
Jul 2017

Magazine

Sensor técnico:  
*Sismografo*

# ODROID



Mediando la  
Aceleración Sísmica  
con el ODRROID-C2



Una Vivienda  
Inteligente de la  
Mano de tu ODRROID

Usa tu Propio  
XU4 como un  
Servidor de Mapas



# Qué defendemos...

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

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

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

Simple, moderno y único.

De modo que tienes a tu alcance lo mejor para hacer todo lo que imagines



## HARDKERNEL



Realizamos envíos de ODROID-C2 and ODROID-XU4 a los países de la UE! Ven y visita nuestra tienda online!

**Dirección:** Max-Pollin-Straße 1  
85104 Pförring Alemania

### Teléfono & Fax

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

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





**E**speremos que nunca tengas que presenciar un terremoto de primera mano, pero es bueno saber que un **ODROID** puede ayudarte a detectar la actividad sísmica en tu área. Usando un simple acelerómetro, un Kit Tinkering **C**, una librería **WiringPi**, una plataforma **ThingSpeak** y un script personalizado en **Python**, un **C2** puede convertirse en un laboratorio geográfico en miniatura que registre la magnitud de las ondas que se dan cerca. Nuestro colaborador habitual **Miltiadis** detalla este increíble proyecto para que puedas crear tu propio sismógrafo.

La domótica ya es casi un estándar en las casas de hoy en día, **Adrian** nos muestra cómo acondicionar una casa para conectar más de **650** tipos de componentes con el fin de tener un control totalmente centralizado de los dispositivos inteligentes desde cualquier dispositivo **ODROID**. Combinado con una de las pantallas táctiles de **Hardkernel**, ofrece una alternativa económica a la instalación de un paquete de automatización comercial. **Nanik** continúa su serie sobre **Android Debug Bridge (ADB)**, **José** presenta una guía para usar un **ODROID-XU4** como servidor de mapas, **@jojo** nos describe su contador **Geiger** casero, **@redrocket** nos da una visión general de **Gogs**, una alternativa a **GitHub/GitLab**, y conoceremos los nuevos extensores de cabezales de **Hardkernel**.

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



**HARDKERNEL**



Hundreds of products available online for the professional developer and hobbyist alike



**ODROID-XU4**



**ODROID-C1+**



**ODROID-C0**



**OWEN ROBOT KIT**



**ODROID-C2**



**VU7 TABLET KIT**



## **Rob Roy, Editor Jefe**

Soy un programador informático que vive y trabaja en San Francisco, CA, en el diseño y desarrollo de aplicaciones web para clients locales sobre mi cluster ODROID. Mis principales lenguajes son jQuery, angular JS y HTML5/CSS3.

También desarrollo SO precompilados, Kernels personalizados y aplicaciones optimizadas para ODROID basadas en las versiones oficiales de Hardkernel, por los cuales he ganado varios Premios. Utilizo mi ODROIDS para diversos fines, como centro multimedia, servidor web, desarrollo de aplicaciones, estación de trabajo y como plataforma de juegos. Puedes echar un vistazo a mi colección de 100 GB de software ODROID, kernel precompilados e imágenes en <http://bit.ly/1fsaXQs>.

---



## **Bruno Doiche, Editor Artístico Senior**

Continúa jugando a los juegos de Exagear, pero también siente la tentación de cambiar su playstation 3 por una playstation 4 para disfrutar de la nueva versión del juego que tanto tiempo lleva esperando. ¡Elite Dangerous! El perro está disfrutando de unas estupendas vacaciones en su casa mientras su suegro se recupera de un brazo roto.

---



## **Manuel Adamuz, Editor Español**

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

---



## **Nicole Scott, Editor Artístico**

Nicole es una experta en Producción Transmedia y Estrategia Digital especializa en la optimización online y estrategias de marketing, administración de medios sociales y producción multimedia impresa, web, vídeo y cine. Gestionando múltiples cuentas con agencias y productores de cine, desde Analytics y Adwords a la edición de vídeo y maquetación DVD, Nicole ayuda a sus clientes con todos los aspectos de la visibilidad online. Posee un ODROID-U2, varios ODROID-U3 y Xu4's, y espera poder utilizar las últimas tecnologías tanto para a nivel personal como empresarial. El sitio web de Nicole lo puedes encontrar en <http://www.nicolecscott.com>.

---



## **James LeFevour, Editor Artístico**

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

---



## **Andrew Ruggeri, Editor Adjunto**

Soy un ingeniero de sistemas Biomédicos anclado en Nueva Inglaterra que actualmente trabaja en la industria aeroespacial. Un microcontrolador 68HC11 de 8 bits y el código ensamblador son todo lo que me interesa de los sistemas embebidos. Hoy en día, la mayoría de los proyectos en los que trabajo están en lenguajes C y C ++, o en lenguajes de alto nivel como C# y Java. Para muchos proyectos, utilizo placas ODROID, pero aún sigo intentando utilizar los controladores de 8 bits cada vez que puedo (soy un fan de ATMEL). Aparte de la electrónica, soy un amante de la fotografía analógica y desarrollo la película friki con la que disfruto intentando hablar en idiomas extranjeros.

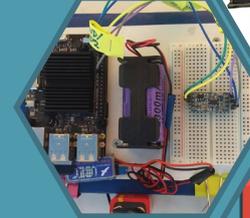
---



## **Venkat Bommakanti, Editor Adjunto**

Soy un apasionado de los ordenadores desde la bahía de San Francisco en California. Procuo incorporar muchos de mis intereses en proyectos con ordenadores de placa reducida, tales como pequeños modificaciones de hardware, carpintería, reutilización de materiales, desarrollo de software y creación de grabaciones musicales de aficionados. Me encanta aprender continuamente cosas nuevas, y trato de compartir mi alegría y entusiasmo con la comunidad.

---

	<b>DOMOTICA - 6</b>
	<b>DESTACANDO LA COMUNIDAD - 12</b>
	<b>EXTENSORES DE CABEZAL XU4 - 13</b>
	<b>DESARROLLO ANDROID - 14</b>
	<b>SISMOGRAFO - 16</b>
	<b>GOGS - 21</b>
	<b>SERVIDOR DE MAPAS - 23</b>
	<b>CONTADOR GEIGER - 25</b>
	<b>LAKKA 2.0 - 26</b>
	<b>CONOCIENDO UN ODROIDIAN - 29</b>

# DOMOTICA CON HOME ASSISTANT

## UNA CASA INTELIGENTE DE LA MANO DE TU ODRROID

por Adrian Popa

Llega un momento en la vida en la que deseamos poner en orden algunas cosas y disponer de un fácil acceso a soluciones complejas. Por ejemplo, quizás tengas varios scripts que te solucionas varios problemas (como encender/apagar un calentador, tomar fotografías con tus cámaras de seguridad, gestionar detectores de presencia, etc.), pero eres el único que puede manejarlos porque requieren de un mantenimiento a través de SSH o a través de alguna rudimentaria página web. Yo también he llegado a ese momento, tengo que buscar una solución “global” que me permita gestionar todas mis automatizaciones y que al mismo tiempo me ofrezca un acceso simple para mi familia.

Pensé en desarrollar un panel de instrumentos web acorde a mis necesidades, pero detesto el desarrollo web. Soy un poco vago y mis sitios no suelen tener muy buena apariencia. Además, necesitaba que funcionase con todo tipo de dispositivos y tamaños de pantalla, y que también estuviera preparado para el futuro. Afortunadamente, pasé bastante tiempo buscando hasta que encontré la solución perfecta - Home Assistant (<http://bit.ly/2h10POE>) - HA para abreviar.

Home Assistant es una plataforma de domótica de código abierto basada en Python 3 que soporta más de 650 componentes, que son módulos que facilitan la interacción con cosas como interruptores “inteligentes”, relés, luces, sensores, dispositivos de red (televisores, routers y cámaras), Software (como Kodi, MPD y Transmission), servicios de red (como el pronóstico del tiempo), pero también te permite añadir tus propios componentes personalizados. Las principales marcas y tecnologías de domótica, como Hue, Nest, IKEA, Vera, ZigBee y MQTT están presentes, puedes encontrar una lista completa de los componentes en <http://bit.ly/2sWJsPy>.

Aparte de los componentes, la plataforma cuenta con una interfaz web a modo de panel de instrumentos y un motor de automatización donde puedes combinar datos de diferentes componentes y generar un evento. Por ejemplo, si de lunes a viernes entre las 8:00 y las 15:00 horas el tiempo es soleado, la temperatura exterior está por encima de los 30 grados, es muy poco probable que llueva y los aspersores exteriores han estado apagados al menos 4 Horas, entonces podemos encenderlos durante 20 minutos. Lo único complicado en esta automatización es buscar la forma de encender y apagar tus aspersores - el resto lo proporciona los componentes existentes y el motor de automatización de Home Assistant. Otros usos pueden ser la posibilidad



¡Ahora tu casa será más inteligente que tus amigos!

de bloquear y desbloquear la puerta cuando una persona específica se conecta al wifi (aunque yo no lo haría personalmente), o iniciar el aire acondicionado automáticamente cuando el sistema detecta que estás llegando a casa del trabajo. Puedes ver más ejemplos de posibles usos en un video de 1 hora en <http://bit.ly/2t0GgCI>. Si está familiarizado con Tasker para Android o IFTTT, Home Assistant es perfecto para tu hogar.

### Instalación

Puede instalar Home Assistant en cualquier dispositivo ODRROID. Dependiendo de la cantidad de automatizaciones que tengas pensado hacer, puedes utilizar un C1 para una instalación de iluminación o un XU4 para grandes casas y complejas reglas que pueden implicar incluso el reconocimiento facial. Yo lo estoy usando en un C2 que me funciona también como reproductor Kodi sin ningún problema.

Vamos a proceder a la instalación de “virtualenv”, lo que significa que se instalarán todos los módulos python necesarios en un directorio específico sin interferir con los módulos del sistema. Además, usaremos un usuario distinto para Home Assistant. También hay imágenes Docker disponibles. Las instrucciones completas con comentarios las tienes disponibles en <http://bit.ly/2t0iaYC>.

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

```

$ sudo apt-get install python-pip python3-dev
$ sudo pip install --upgrade virtualenv
$ sudo adduser --system homeassistant
$ sudo addgroup homeassistant
$ sudo usermod -G dialout -a homeassistant
$ sudo mkdir /srv/homeassistant
$ sudo chown homeassistant:homeassistant /srv/homeassistant
$ sudo su -s /bin/bash homeassistant
$ virtualenv -p python3 /srv/homeassistant
$ source /srv/homeassistant/bin/activate
(homeassistant)$ pip3 install --upgrade homeassistant
$ exit

```

Para iniciar y administrar el proceso, lo mejor es crear un servicio systemd para manejarlo:

```

$ sudo vi /etc/systemd/system/homeassistant.service
[Unit]
Description=Home Assistant
After=network.target time-sync.target
Requires=time-sync.target

[Service]
Type=simple
User=%i
ExecStart=/srv/homeassistant/bin/hass -c "/home/homeassistant/.homeassistant"

[Install]
WantedBy=multi-user.target

```

Para iniciar Home Assistant, simplemente inicia su servicio:

```

$ sudo service homeassistant start
$ sudo service homeassistant enable

```

Ten en cuenta que, si vas a utilizar componentes que necesiten HTTPS, deberás tener configurado correctamente el tiempo de inicio, para que los certificados sean válidos. El inicio del servicio depende de systemd-timesyncd, que a su vez depende de ntp que no está instalado:

```

$ sudo apt-get remove ntp
$ sudo service systemd-timesyncd restart
$ sudo systemctl enable systemd-timesyncd

```

En caso de problemas, podrás revisar los archivos log a través de journalctl:

```

$ sudo journalctl -u homeassistant -f

```

Una vez que se inicie el proceso, podrás conectarse a <http://odroid-ip:8123/>. Ten en cuenta que el primer inicio (o un inicio tras una actualización) puede ser más lento, así que deja que se ejecute durante unos minutos hasta que accedas a la interfaz web. Home Assistant cuenta con una aplicación nativa para iOS (<http://apple.co/2tYi2WI>), mientras que para los clientes Android se puede configurar la página como pantalla de inicio (Chrome -> navega a <http://ip-odroid:8123> -> Menu -> Add to homescreen).

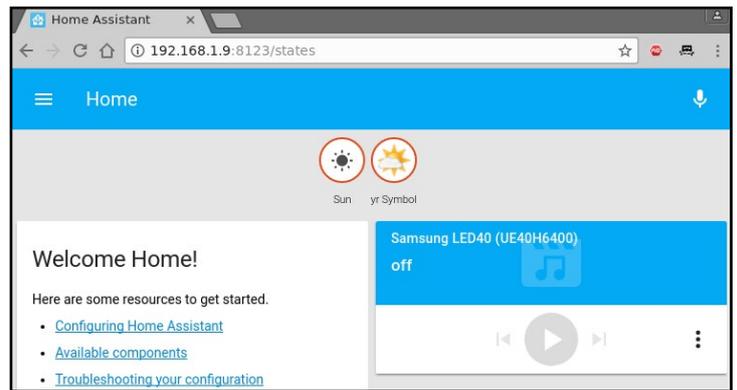


Figura 1 - Página de Inicio de Home Assistant

## El archivo de configuración

Para poner en marcha los componentes y configurar tu instalación, tendrá que trabajar bastante con el (los) archivo(s) de configuración de Home Assistant. Esperemos que en una futura versión sea posible manejar la configuración directamente desde la interfaz web, pero de momento necesitarás un editor de texto. El archivo principal es `/home/homeassistant/.homeassistant/configuration.yaml`. Su formato es YAML - que significa "Another Another Markup Language". Al igual que Python, utiliza espacios en blanco (no tabulaciones) para delimitar las secciones de código. Por defecto, utiliza una sangría de dos espacios para las secciones anidadas. En caso de problemas, recibirás mensajes de error al iniciar el servicio. Puedes validar la sintaxis con un servicio como <http://www.yamllint.com/> que te permitirá saber dónde te has equivocado. También tienes una guía de solución de problemas en <http://bit.ly/2tDHMSa>.

Una vez realizados los cambios en el archivo de configuración, deberás reiniciar el servicio homeassistant para que tengan

Figura 2 - La configuración por defecto

```

1 homeassistant:
2   # Name of the location where Home Assistant is running
3   name: Home
4   # Location required to calculate the time the sun rises and sets
5   latitude: 44.7
6   longitude: 26.45
7   # Impacts weather/sunrise data (altitude above sea level in meters)
8   elevation: 69
9   # metric for Metric, imperial for Imperial
10  unit_system: metric
11  # Pick yours from here: http://en.wikipedia.org/wiki/List_of_tz_database_time_zones
12  time_zone: Europe/Bucharest
13
14  # Show links to resources in log and frontend
15  introduction:
16
17  # Enables the frontend
18  frontend:
19
20  http:
21    # Uncomment this to add a password (recommended!)
22    # api_password: PASSWORD
23    # Uncomment this if you are using SSL or running in Docker etc
24    # base_url: example.duckdns.org:8123
25
26  # Checks for available updates
27  updater:
28
29  # Discover some devices automatically
30  discovery:
31
32  # Allows you to issue voice commands from the frontend in enabled browsers
33  conversation:
34
35  # Enables support for tracking state changes over time.
36  history:
37
38  # View all events in a logbook
39  logbook:
40
41  # Track the sun
42  sun:
43
44  # Weather Prediction
45  sensor:
46    platform: yr
47
48  # Text to speech
49  tts:
50    platform: google

```

efecto. Puede hacerlo desde el intérprete de comandos con “sudo service homeassistant restart”, o desde la interfaz web de HA, haciendo clic en el ícono superior izquierdo, seleccionando el icono “Configuration” y activando la opción “Restart” desde la sección “Server Management”. El video <http://bit.ly/2sAmD3F> muestra algunos consejos prácticos para editar la configuración.

Si tienes pensado usar HA desde fuera de tu LAN (por ejemplo, desde Internet), tiene varias opciones. Uno de ellas es habilitar el soporte HTTPS y redireccionar el puerto 8123 en tu router. Esto te permite realizar conexiones cifradas, pero expones tu instalación a Internet (podría existir vulnerabilidades que permitirían a los atacantes tomar el control de tu sistema/LAN). Una segunda opción (la que yo prefiero) es configurar una VPN en tu router (o incluso en tu ODROID) que te permite conectarte y acceder a HA (y otros recursos LAN) de forma segura.

Si vas a utilizar HTTPS, para que funcione todos los servicios, necesitaras proporcionar certificados SSL válidos (no auto-firmados). Para obtener certificados válidos necesitarás tener un nombre DNS público (por ejemplo, usando un servicio DNS dinámico como duckdns.org) y recurriendo a letsencrypt.org para configurar un certificado SSL válido para tu instalación. Tienes información detalladas sobre este tema en el video <http://bit.ly/2tY6LGB>. Si tienes que usar certificados autofirmados, existe una guía en <http://bit.ly/2t00bzH>.

Independientemente del método de acceso (http o https), querrás configurar una contraseña. HA no soporta multiples cuentas de usuario, pero puedes definir una contraseña API que necesitaras para iniciar sesión en la interfaz web. La mejor forma de hacer esto es creando un archivo que almacene todos tus datos confidenciales (como contraseñas y URLs), lo llamaremos “secrets.yaml” y lo referenciamos en el archivo configuration.yaml

```
$ cat /home/homeassistant/.homeassistant/secrets.yaml
api_password: odroid
$ cat /home/homeassistant/.homeassistant/configuration.yaml
...
http:
  api_password: !secret api_password
...
```

Ahora, cuando reinicies HA, se te solicitará una contraseña. Tienes más detalles sobre el tema de confidencialidad en <http://bit.ly/2rLGEkV>.

Para familiarizarnos con la configuración de HA, vamos a poner en marca algunos componentes. Vamos a configurar el pronostico del tiempo, algunas cámaras IP, Kodi y MPD, la detección de presencia basada en WiFi y, además, un sensor de temperatura 1-wire conectado al ODROID.

## El tiempo desde Darksky

Hay varios proveedores meteorológicos ya integrado en HA (<http://bit.ly/2t411Rh>), así que puedes elegir el que más te

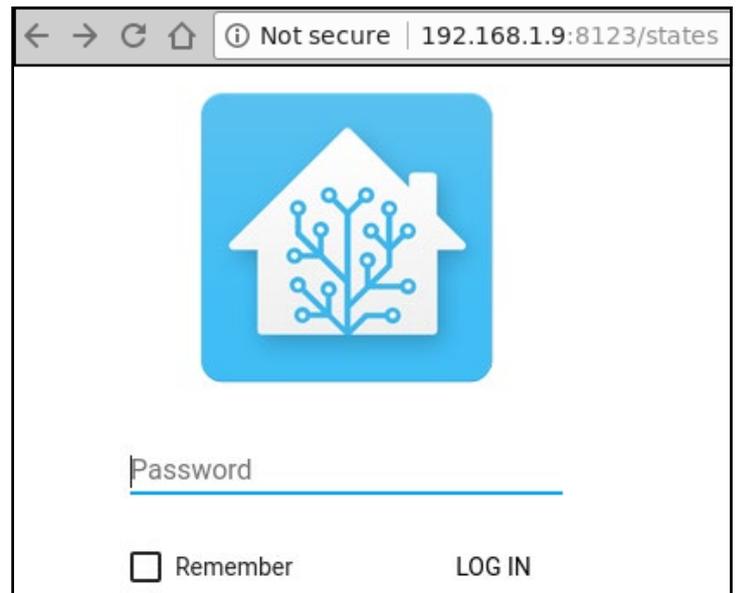


Figura 3 – Autenticación

guste. Yo voy a recurrir a DarkSky (<http://bit.ly/2t4gq0S>), que proporciona pronósticos bastante precisos de mi zona. Deberías consultar la página de ayuda del componente para conocer los detalles de la configuración y las variables que puedes utilizar. Necesitas registrarte en DarkSky y conseguir una clave API que te permitirá hacer unas 1000 consultas por día de forma gratuita. Lo mejor es guardar esta clave API dentro de tu archivo secrets.yaml (Reemplaza tu propia clave)

```
darksky_api_key: 87f15cbb811204412cc75109777ea5cf
```

La configuración tiene varias variables, la mayoría son opcionales, sin embargo, en configuration.yaml, debajo de la sección del sensor deberías tener lo siguiente (no dudes en borrar la entrada “platform: yr”):

```
sensor:
  - platform: darksky
    api_key: !secret darksky_api_key
    name: Dark Sky
    monitored_conditions:
      - summary
      - precip_type
      - precip_probability
      - temperature
      - humidity
      - precip_intensity
      - wind_speed
      - pressure
      - wind_bearing
      - apparent_temperature
      - icon
      - minutely_summary
      - hourly_summary
      - temperature_max
      - temperature_min
    units: si
    update_interval: '00:15'
```

El código es en su mayoría auto-explicativo. Éste configura

una nueva plataforma del tipo “darksky”, con un nombre específico (opcional) y una `api_key` (requerida), y extrae un conjunto de parámetros (condiciones monitorizadas) del proveedor meteorológico cada 15 minutos. Tu localización real se coge de

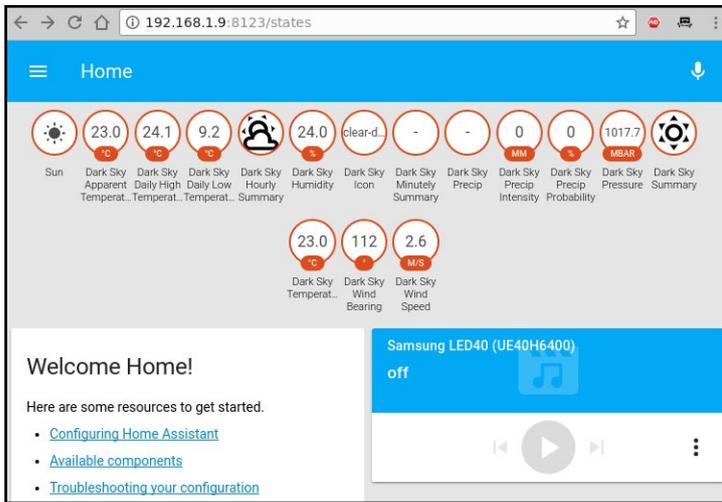


Figura 4 – Datos meteorológicos

los parámetros de latitud/longitud en homeassistant, así que asegúrate de que son correctos. Tras reiniciar el servicio homeassistant, deberías ver las variables monitorizadas en forma de escudos en la parte superior de la ventana. Al hacer clic en un escudo te mostrará cómo ese valor en particular ha cambiado con el tiempo.

## Visionado de cámaras IP

HA soporta una gran cantidad de cámaras (<http://bit.ly/2t4DtsD>), incluyendo la lectura de datos de un archivo que se podría usar para presentar un gráfico, o los datos visuales generados por otras herramientas. Utilizaremos el componente Generic MJPG Camera (<http://bit.ly/2t4tIKM>) y el componente Local File (<http://bit.ly/2s4Y5w4>).

La cámara que queremos monitorear está disponible en <http://bit.ly/2t4cHkc> (es una cámara web pública), que deberíamos añadir al archivo `secrets.yaml`.

```
camera1_stream_url: http://iris.not.iac.es/axis-
cgi/mjpg/video.cgi?resolution=320x240
camera1_still_url: http://iris.not.iac.es/jpg/im-
age.jpg
```

Para las cámaras la configuración dentro de `configuration.yaml` es similar a esto:

```
camera:
- platform: mjpeg
  mjpeg_url: !secret camera1_stream_url
  still_image_url: !secret camera1_still_url
  name: Observatory in Spain
- platform: local_file
  file_path: /tmp/tux.jpg
```

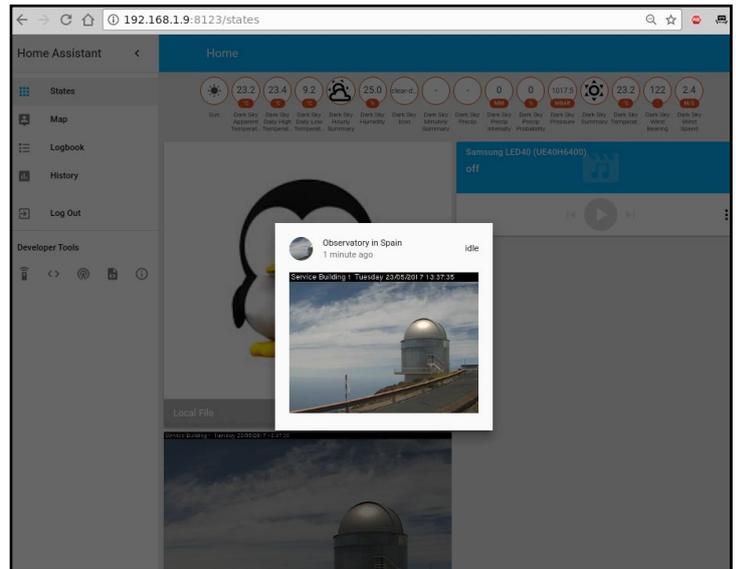


Figura 5 - ¡Webcams!

Como de costumbre, necesitarás reiniciar el servicio HA para volver a leer la configuración (este podría ser un buen momento para comentar el componente “Introducción”). Ten en cuenta que, al hacer clic en una cámara web, verás un canal en vivo, de lo contrario la imagen fija se actualiza cada 10 segundos.

Entonces, ¿Qué se puede hacer con estas webcams configuradas aparte de mirarlas? Bueno, puedes usarlas con otros componentes como OpenCV (<http://bit.ly/2s4UUEJ>) para generar disparadores cuando se vean determinados rostros, o Seven Segments Display (<http://bit.ly/2sAbOP0>) que permite tomar lecturas de varias pantallas digitales.

## Kodi y MPD

Para configurar los reproductores multimedia, puede buscar en la lista de componentes Media Player en <http://bit.ly/2s0IAtQ>. Para configurar Kodi (<http://bit.ly/2sA5qr6>), deberás activar la opción “Allow remote control via HTTP” (<http://bit.ly/2t4cYne>) y configurar primero un nombre de usuario y una contraseña. Para ello, añade el usuario y la contraseña al archivo `secrets.yaml`:

```
kodi_user: kodi
kodi_pass: kodi
```

A continuación, edita `configuration.yaml`:

```
media_player:
- platform: kodi
  host: 192.168.1.140
  name: Kodi Livingroom
  username: !secret kodi_user
  password: !secret kodi_pass
```

Para configurar MPD, suponiendo que ya tiene un servidor MPD en tu red, agrega el componente MPD (<http://bit.ly/2s5sbzE>) y añade la contraseña a `secrets.yaml`:

```
mpd_secret: mpd
```

Y luego, edita configuration.yaml:

```
media_player:
  ...
  - platform: mpd
    host: 192.168.1.140
    name: MPD Living
    password: !secret mpd_secret
```

Tras reiniciar Home Assistant, dispondrás de los dos nuevos reproductores multimedia y podrás ver su estado (reproducir/de-tener), controlar el volumen e incluso cambiar la lista de reproducción actual o utilizar el componente text-to-speech para que el reproductor multimedia “reproduzca” lo que quieras.

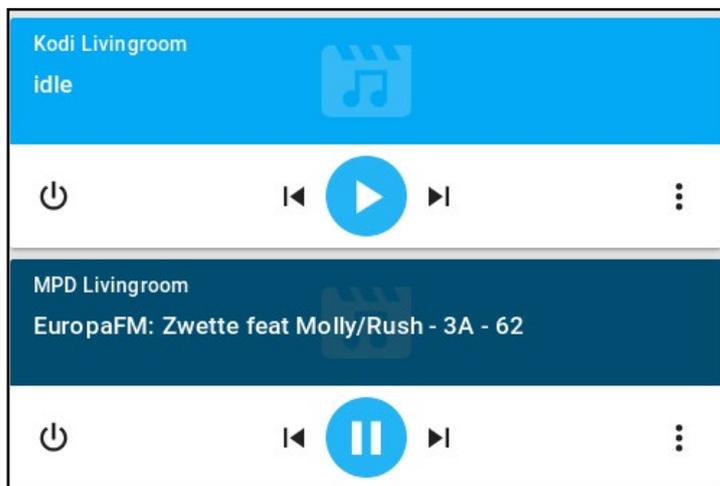


Figura 6 - Reproductores Multimedia

## Detectores de presencia

Los componentes de detección de presencia (<http://bit.ly/2t0Gt8H>) intentan rastrear las ubicaciones de las personas y así poder aplicar reglas de geocaching (por ejemplo, hacer algo si una persona entra o sale de una ubicación). Por lo general, el rastreo se realiza mediante dispositivos de detección conectados a un router (vía wifi), a través de la proximidad del bluetooth (<http://bit.ly/2s0Sqfw>) o utilizando servicios de localización como Owntracks (<http://bit.ly/2rLQdR1>).

Utilizaremos un rastreador que, dependiendo de tu router, se conectará periódicamente a la interfaz de administración de tu router, listará la tabla ARP y descubrirá qué dispositivos están conectados. Son muchos los tipos de router que son compatibles, desde los de proveedores de gama alta como Cisco, a los routers de consumo como Asus, Netgear y TP-Link. Incluso son compatibles los firmwares de código abierto, como OpenWRT, DD-WRT y Tomato.

Nosotros utilizaremos un router Asus con SSH activado, de modo que necesitamos el componente ASUSWRT: <http://bit.ly/2s4T32Q>. Puedes elegir iniciar sesión con nombre de usuario/contraseña o configurar una clave SSH e iniciar sesión con ella. Ten en cuenta que determinadas versiones de firmware tienen activadas medidas de seguridad que limitan el número de conex-

iones SSH, llegando a incluir tu IP en una lista negra si se inician demasiadas conexiones.

Como de costumbre, configuraremos los datos privados (la ruta a la clave o la contraseña ssh) en el archivo secrets.yaml:

```
router_user: admin
router_password: my_secret_password
```

Dentro de configuration.yaml añade la siguiente sección:

```
device_tracker:
  - platform: asuswrt
    host: 192.168.1.1
    username: !secret router_user
    password: !secret router_password
    interval_seconds: 120
    consider_home: 300
    track_new_devices: yes
```

La página de configuración del rastreador de dispositivos (<http://bit.ly/2s4WPcA>) proporciona más detalles sobre las opciones que puedes utilizar. La opción interval\_seconds es el tiempo que transcurre entre las exploraciones (2 minutos) y la opción consider\_home te mantiene “en casa” incluso si tus dispositivos no te ven durante 300 segundos.

Una vez que reinicies HA y tras haber realizado un barrido inicial, se creará un nuevo archivo denominado known\_devices.yaml. Aquí podrás asignar un nombre amigable e incluso una imagen a un dispositivo específico, o hacer que otros dispositivos sean ignorados.

Una entrada en known\_devices.yaml podría ser esta:

```
aldebaran:
  hide_if_away: false
  mac: 00:1E:06:31:8C:5B
  name: aldebaran
  picture: /local/aldebaran.png
  track: true
  vendor: WIBRAIN
```

Ten en cuenta que he añadido una ruta a la imagen local que se almacena en /home/homeassistant/.homeassistant/www/aldebaran.png. Puedes crear la carpeta “www” con el comando:

```
$ sudo mkdir /home/homeassistant/.homeassistant/
www
```



Figura 7 - Descubrimiento inicial / Entradas personalizadas



Si hay dispositivos que no quieres monitorizar, puedes definir “track: false” en el archivo `known_devices.yaml`.

## Medición de la temperatura

Una característica muy potente de Home Assistant es la posibilidad de rastrear todo tipo de sensores (<http://bit.ly/2cNb4gJ>). Queremos monitorear un sensor de temperatura basado en el protocolo 1-wire, conectado localmente al ODROID (<http://bit.ly/2s12ZPx>). Antes de añadir el sensor a HA, asegúrate de que es legible desde la línea de comandos. Puedes seguir la guía de configuración de la wiki <http://bit.ly/2s0zbTp>.

Necesitarás conocer el ID del sensor para añadirlo a HA:

```
$ ls /sys/bus/w1/devices/
28-0516866e14ff w1_bus_master1
$ cat /sys/bus/w1/devices/28-0516866e14ff/w1_
slave
92 01 4b 46 7f ff 0c 10 b5 : crc=b5 YES
92 01 4b 46 7f ff 0c 10 b5 t=25125
```

A continuación, puede realizar los siguientes cambios en `configuration.yaml` y sondear el sensor cada 5 minutos:

```
sensor:
  ...
  - platform: onewire
    names:
      28-0516866e14ff: Living room
      scan_interval: '00:05'
```

Después de reiniciar HA, la nueva lectura será visible en la interfaz web a modo de escudo en la parte superior de la página.

## Organizando las vistas

Notarás que a medida que vas agregando componentes, la interfaz web empieza a desordenarse con gran cantidad de elementos dispersos por todas partes. Puedes utilizar grupos y vistas para limpiar la interfaz y colocar elementos relacionados en una determinada pestaña. Para entender lo que hay que hacer, vamos a aclarar el vocabulario.

Las “entities” son variables que proporcionan datos, como un sensor o un interruptor. Las plataformas (como `dark_sky`) normalmente proporcionan acceso a múltiples “entities” (temperaturas mínimas/máximas o pronóstico). Puedes ver una lista de las “entities”, sus nombres y sus valores si navegas por la interfaz web bajo Developer tools -> States (<>) -> Entities.

Un “group” es simplemente un objeto que contiene una lista de “entities”. Visualmente, un “group” se representa como un panel o una ficha. Por defecto, el group “group.all\_devices” existe y almacena los elementos descubiertos por una plataforma de seguimiento de dispositivos. Los “groups” normalmente contienen una lista de “entities”

Una “view” se presenta como una pestaña separada den-

tro de Home Assistant. Las “view” son en realidad grupos de “groups” y difieren de los grupos normales por tener la propiedad “view: yes”. También puedes agregar “entities” individuales, así como grupos a una “view”

Nosotros agruparemos nuestros sensores actuales en las siguientes categorías:

La primera pestaña se llamará Home y contiene los siguientes grupos (internamente se llamará `default_view`, para que se muestre al iniciar sesión):

- Datos Meteorológicos
- Información de presencia
- Información del sistema (para mostrarte si hay actualizaciones disponibles)

La segunda pestaña se llamará Media y contendrá los siguientes grupos:

- Reproductores multimedia
- La pestaña final se llamará Imágenes y contiene:
- Webcams

La configuración tendrá un aspecto similar a este:

```
group:
  default_view:
    view: yes
    entities:
      - group.weather
      - group.presence
      - group.systeminfo
  media:
    view: yes
    entities:
      - group.mediaplayers
  images:
    view: yes
    entities:
      - camera.observatory_in_spain
      - camera.local_file
  weather:
    name: Weather
    entities:
      - sensor.dark_sky_apparent_temperature
      - sensor.dark_sky_daily_high_temperature
      - sensor.dark_sky_daily_low_temperature
      - sensor.dark_sky_hourly_summary
      - sensor.living_room
  presence:
    name: Presence
    entities:
      - device_tracker.aldebaran
```

# NUEVO LOGO DE HARDKERNEL UN NUEVO Y RELUCIENTE LOGO DE LA COMUNIDAD PARA NUESTRO FORO

editado Rob Roy

**P**or casualidad no has notado algo diferente en los Foros ODROID. Hay un nuevo logotipo de la comunidad ODROID que ha sido diseñado y donado por Daniel Mehrwald (@AreaScout), compuesto por la palabra “ODROID”, una imagen de nuestro hardware y una imagen del kernel.



## LISTA DE COMPATIBILIDAD DE WINDOWS X86 PARA ODROID XU3/XU4 EJECUTANDO EXAGEAR 2.1 DESKTOP

editado por Bruno Doiche

**Q**uieres conocer lo que estamos ejecutando últimamente en Exagear sobre nuestro XU3 / XU4. ¡No busque más! ¡Aquí tienes nuestra lista más actualizada de los clásicos juegos de PC que harán que tu inversión valga cada céntimo!

**7th Legion - Alien Nations - Age of Wonders - Alpha Centauri + Alien Crossfire - Anno 1602 - Arcanum: Of Steamworks and Magick Obscura - Balls of Steel - Caesar III - Dune 2000 - Earth 2140 - Homeworld: Cataclysm - KKND2 Krossfire - Larry 7: Love for Sail**

¡Cada uno de los juegos incluidos en la lista funcionará perfectamente en tu ODROID! Disfruta, y para más información visita:

<https://oph.mdrjr.net/meveric/other/ExaGear/>

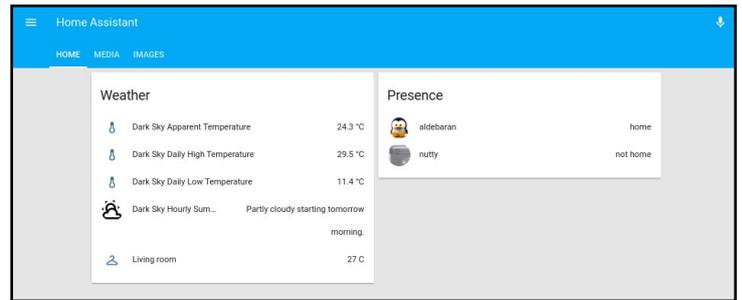


Figura 8 - Una interfaz limpia con “view” y “groups”

```
- device_tracker.nutty
systeminfo:
  name: System Info
  entities:
    - updater.updater
mediaplayers:
  name: Media Players
  entities:
    - media_player.mpd_livingroom
    - media_player.kodi_livingroom
```

Tienes más detalles sobre los grupos y el diseño en el video <http://bit.ly/2s5d6xT>.

## Actualizaciones

Como Home Assistant no se ha instalado a través de apt-get, deberás gestionar las actualizaciones manualmente. Antes de actualizar, lo mejor es leer las notas de la versión y comprobar que la actualización no vaya a estropear ninguna configuración anterior, ya que a veces se rediseña la configuración de nuevos componentes, lo que significa que tendrá que volver a montarla. Para recibir notificaciones de nuevas versiones puedes utilizar la entrada updater.updater que comprueba periódicamente las versiones más recientes y las muestra dentro de Home Assistant. Las actualizaciones son bastante frecuentes, y suele aparecer una versión importante cada 2-3 semanas. El procedimiento de actualización es simple y los detalles los puedes encontrar en <http://bit.ly/2s0Kn24>.

```
$ sudo service homeassistant stop
$ sudo su -s /bin/bash homeassistant
$ source /srv/homeassistant/bin/activate
(homeassistant)$ pip3 install --upgrade homeassis-
tant
(homeassistant)$ exit
$ sudo service homeassistant start
```

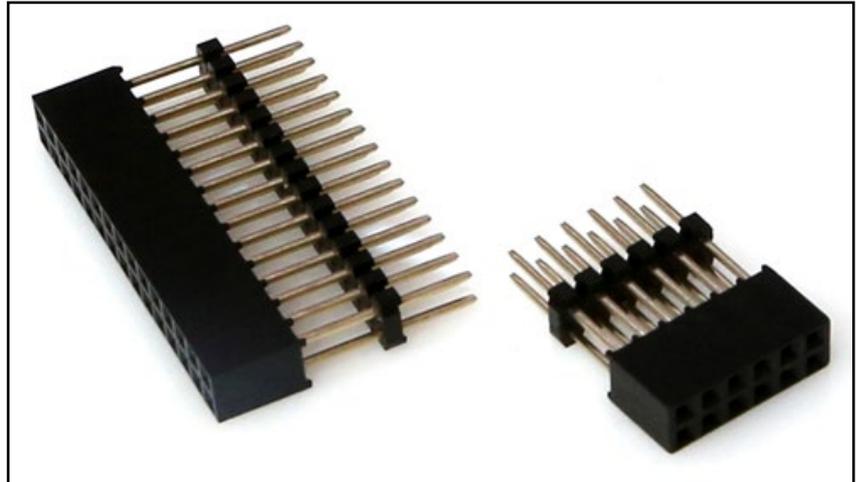
En futuros artículos, veremos cómo configurar componentes más complejos como un relé remoto o una unidad de aire acondicionado, configurando las automatizaciones e instalando un panel de instrumentos. Para comentarios, preguntas y sugerencias, visita el hilo de soporte en <http://bit.ly/2s13GbB>.

# EXTENSORES DE CABEZALES PARA ODROID-XU4

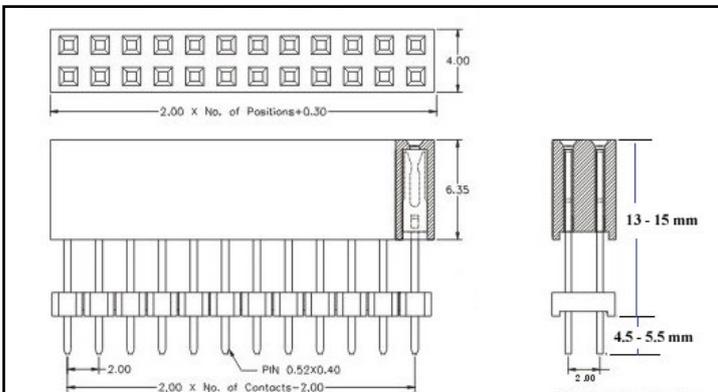
## CONEXIONES MAS ELEVADAS PARA TUS PROYECTOS

editado por Rob Roy

**R**ecientemente, dos miembros de los foros ODROID solicitaron un extensor de cabezal para aumentar la altura de las conexiones de 30 y 12 pines en los ODROID-XU4 y ODROID-XU4Q. Hardkernel respondió creando un nuevo producto, disponible en <http://bit.ly/2t4Qouj>, que permite apilar varios shields por encima del ODROID-XU4, como el Shifter Shield (<http://bit.ly/2t4huSj>).



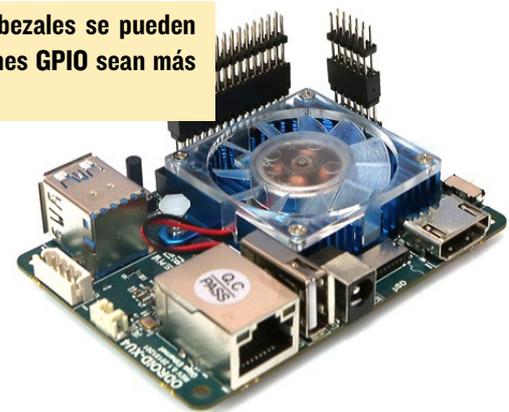
Extensores de los cabezales de 30 y 12 pines



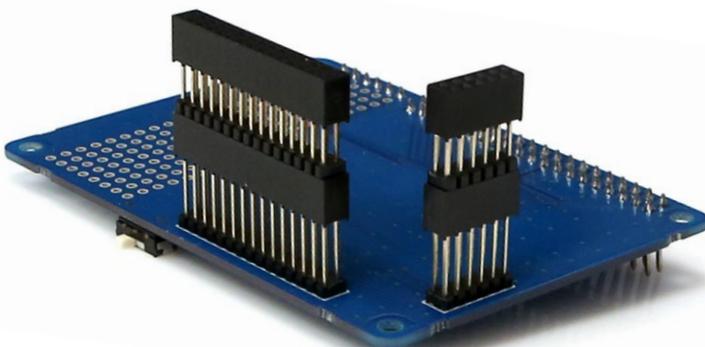
Especificaciones técnicas de los extensores de cabezales



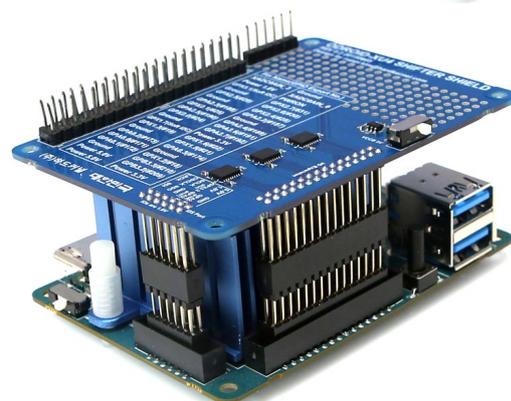
Los extensores de cabezales se pueden apilar para que los pines GPIO sean más accesibles



El ODROIDX-U4Q con extensores de cabezales y Shifter Shield instalado encima del disipador de calor azul



Las conexiones son útiles cuando quieres que el Shifter Shield esté más alto



# DESARROLLO ANDROID

## UN INTENSO VIAJE POR DENTRO DE ADB

### PARTE 2

por Nanik Tolaram

Esta es la segunda entrega de una serie de dos partes sobre Android ADB, en la que analizamos en profundidad su código fuente y cómo funciona internamente. Veremos los 2 componentes de adb que se inician en el escritorio cuando se ejecuta el comando “adb”.

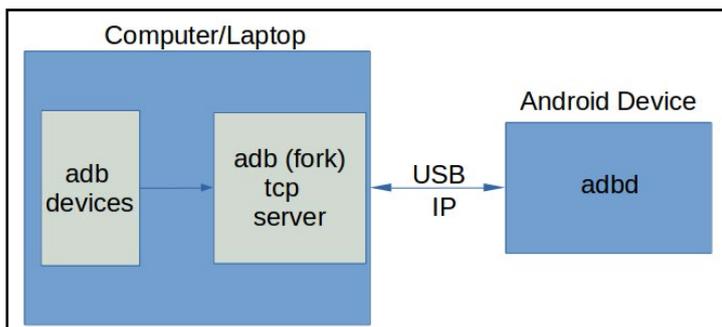


Figura 1 - Arquitectura de ADB

En la parte 1, vimos el diagrama que se muestra en la Figura 1 y ahora vamos a ver con más detalle cómo funciona adb dentro de tu ordenador cuando conectas un dispositivo Android a tu PC/Portátil. Este artículo está basado en el adb de Android 5.1.1 y por lo general, la lógica debe ser similar para las nuevas versiones de Android con ligeras diferencias en el comportamiento y el código.

### Arquitectura más profunda

Como puede ver en la Figura 2, cuando ejecutas adb lo primero que hará es configurar un servidor local en el puerto 5037.

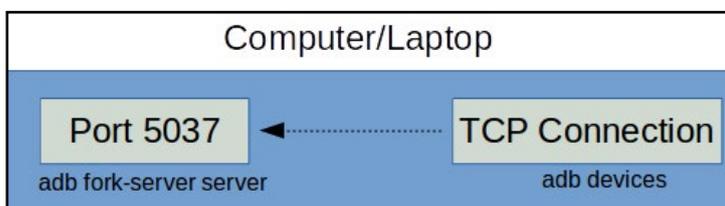
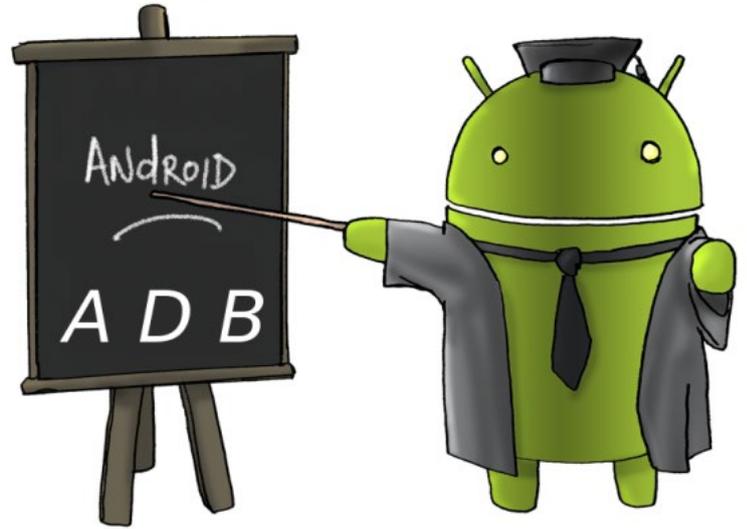


Figura 2 - Servidor local ADB 5037



El puerto 5037 será utilizado internamente por el código que ejecutará el comando real. Echemos un vistazo a un ejemplo cuando ejecutamos un comando específico en adb:

```
$ adb devices
```

Cuando adb se ejecuta por primera vez, se inicia otro proceso adb con el siguiente parámetro adicional:

```
'fork-server server'
```

La Figura 3 muestra el código que hace exactamente esto.

El proceso generado se ejecutará y actuará como servidor escuchando el puerto local 5037, mientras que el proceso origi-

```
char path(PATH_MAX);
int fd[2];

// set up a pipe so the child can tell us when it is ready.
// fd[0] will be parent's end, and fd[1] will get mapped to stderr in the child.
if (pipe(fd)) {
    fprintf(stderr, "pipe failed in launch_server, errno: %d\n", errno);
    return -1;
}
get_my_path(path, PATH_MAX);
pid_t pid = fork();
if(pid < 0) return -1;

if (pid == 0) {
    // child side of the fork

    // redirect stdout to the pipe
    adb_close(fd[0]);
    dup2(fd[1], STDOUT_FILENO);
    adb_close(fd[1]);

    char str_port[30];
    snprintf(str_port, sizeof(str_port), "%d", server_port);
    fprintf(stderr, "***RUNNING ADB using execl()");
    // child process
    int result = execl(path, "adb", "-P", str_port, "fork-server", "server", NULL);
    // this should not return
    fprintf(stderr, "OOPS! execl returned %d, errno: %d\n", result, errno);
} else {
    // parent side of the fork

    char temp[3];

    temp[0] = 'A'; temp[1] = 'B'; temp[2] = 'C';
    // wait for the "OK\n" message
    adb_close(fd[1]);
}
```

Figura 3 - Proceso de inicio del servidor fork de ADB

nal esperará y con el tiempo se conectará al puerto 5037 enviando el comando “devices”.

## Servidor ADB

Ahora que sabemos que adb generará otro proceso adb y lo ejecutará como servidor, vamos a echar un vistazo a lo que realmente sucede cuando se ejecuta como servidor. La figura 4 muestra el flujo del código.

El fdevent\_loop que se muestra en la Figura 5 es un bucle continuo a la espera de que llegue un

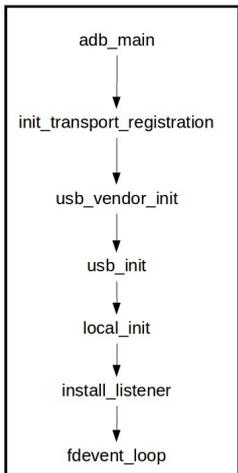


Figura 4 - Flujo del código que se ejecuta como servidor

nuevo evento.

La gestión del evento utiliza un objeto interno denominado fdevent que se utiliza junto con el multiplexado de E/S síncrono que forma parte de la API de Linux. El proceso de registro fdevent utiliza el descriptor de archivo que el código escucha y cuando se reciben datos para ese descriptor en particular, el bucle llamará a la función registrada dentro de

la función fdevent\_call\_fdfunc ().

```

if (!is_daemon)
{
    // Inform our parent that we are up and running.
    #ifdef HAVE_WINS32_PROC
    DWORD count;
    WriteFile(GetStdHandle(STD_OUTPUT_HANDLE), "OK\n", 3, &count, NULL);
    #elif defined(HAVE_FORNEXEC)
    fprintf(stdout, "OK\n");
    fflush(stdout);
    #endif
    //start_logging();
    D("Event loop starting\n");
    fdevent_loop();
    usb_cleanup();
    return 0;
}

void fdevent_loop()
{
    fdevent *fde;
    fdevent_subproc_setup();
    for(;;) {
        D("--- waiting for events\n");
        fdevent_process();
        while((fde = fdevent_plist_dequeue())) {
            fdevent_call_fdfunc(fde);
        }
    }
}
    
```

Figura 5 - fdevent\_loop

## Cliente ADB

Una vez realizado el paso anterior de ejecutar adb como servidor, el código continuará su ejecución conectándose al puerto local 5037 y retransmitirá nuestro comando de ejemplo ‘devices’. El código esperará la respuesta del servidor y mostrará el resultado una vez que haya completado el proceso. La Tabla 1 muestra algunas de las funciones más utilizadas relacionadas con fdevent relativas a adb.

## Registro del transporte

ADB soporta conexiones vía USB y TCP, lo cual facilita la vida a los desarrolladores. Cuando se ejecuta el código del servidor, hay un subproceso de fondo que escucha las conexiones USB entrantes. Una vez que se detecta un nuevo dispositivo USB, utiliza fdevent\_install para instalar una nueva función

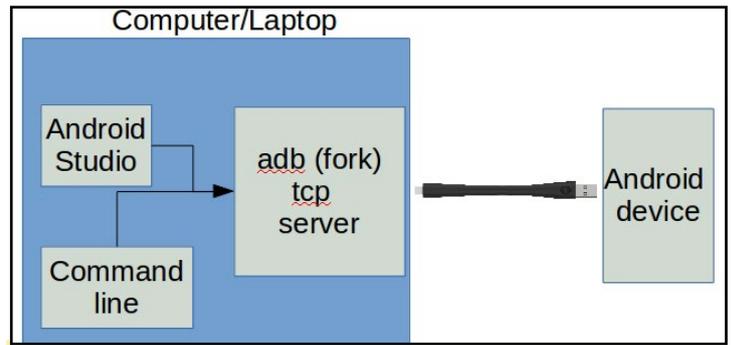


Figura 6 - Conexión USB para dispositivos Android

para el descriptor de archivo con el objeto de establecer la comunicación entre el dispositivo y el servidor local.

La Figura 7 describe el proceso de registro dentro de ADB

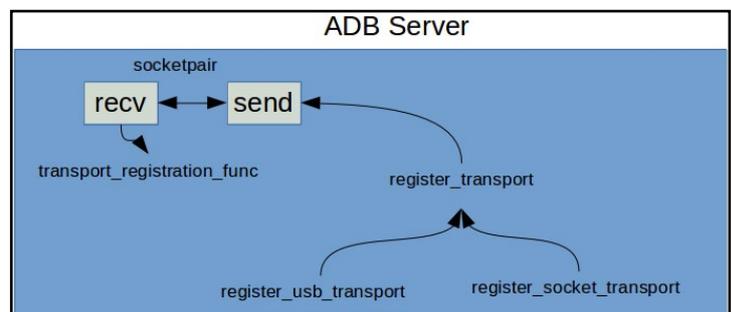


Figura 7 - Registro de transporte

para USB y TCP. Internamente, hay un “socketpair” que se utiliza para comunicarse cuando se detecta un nuevo dispositivo Android. Esto es importante para que el ADB pueda gestionar varios dispositivos Android al mismo tiempo. La función register\_usb\_transport es responsable de las conexiones USB, mientras register\_socket\_transport es responsable de

fdevent_install	Register function as a callback for a particular file descriptor
fdevent_set	Flag the file descriptor to wait for reading, writing or error handling
fdevent_add	Add flag to the file descriptor
fdevent_del	Delete flag from the file descriptor
fdevent_loop	Loop to process all incoming events

Tabla 1 - Algunas de las funciones fdevent disponibles

las conexiones de red (TCP). Una vez que la conexión es detectada y gestionada por adb, éste crea un hilo de ejecución independiente para manejar la comunicación entre sí mismo y el dispositivo.



# DETECTOR SISMOGRAFO DE TERREMOTOS

## MIDIENDO LA ACELERACION SISMICA USANDO EL ODROID-C2

por Miltiadis Melissas

Los terremotos son el resultado de la liberación repentina de energía sobre la litosfera de la Tierra que crea ondas sísmicas (Figura 1). Los terremotos varían en magnitud desde los que son tan débiles que no se perciben, a los suficientemente violentos como para echar a la gente a la calle y destruir ciudades enteras. La sismología o actividad sísmica hace referencia a la frecuencia, el tipo y el tamaño de los terremotos que tienen lugar durante un período de tiempo en un área determinada (<http://bit.ly/1WBSHEL>).

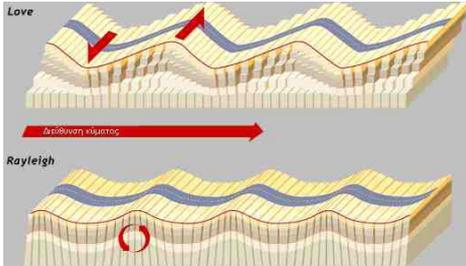


Figura 1 - Diagrama de un terremoto

En este proyecto, crearemos un dispositivo sismográfico detector de terremotos utilizando el ODROID-C2 junto con un acelerómetro (MMA7455) para detectar y medir la aceleración de la gravedad causada por las ondas sísmicas. Nuestro sismógrafo será capaz de:

- Medir la magnitud de un terremoto en gramos (g)
- Detectar y medir la aceleración de un sismo sobre tres ejes diferentes (x, y, z)
- Registrar el tiempo y la duración de un terremoto

- Representar los valores anteriores con gráficos en tiempo real.
- Representar las coordenadas GPS en Google Maps
- Publicar en Twitter cuando la aceleración sísmica alcance un cierto límite - normalmente 0.3g - avisar a los ciudadanos del momento y de la magnitud del terremoto.

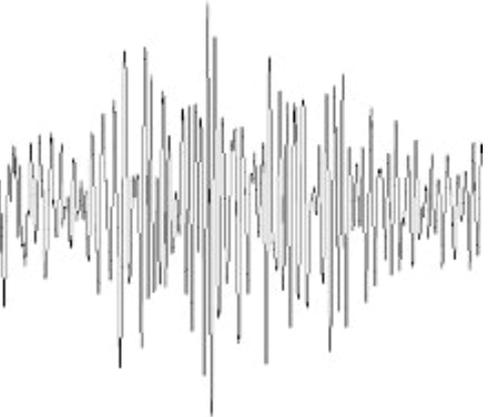
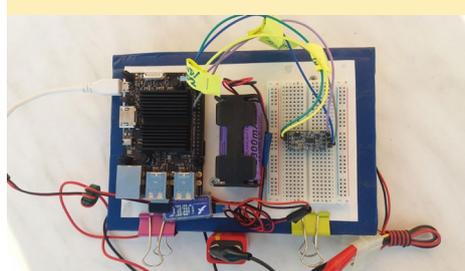
Este dispositivo ofrece numerosas ventajas, tales como:

- No hay componentes mecánicos, lo cual reduce la fricción y aumenta la precisión
- Disponibilidad de código abierto
- Económico, ideal para desarrollar de una red de sismógrafos
- Puede ser programado y controlado remotamente vía web
- Los informes del sismógrafo son accesibles en todo el mundo, desde cualquier dispositivo con una conexión a Internet

### Hardware

Necesitarás todos los accesorios ha-

Foto 1 - ODROID-C2 como sismógrafo



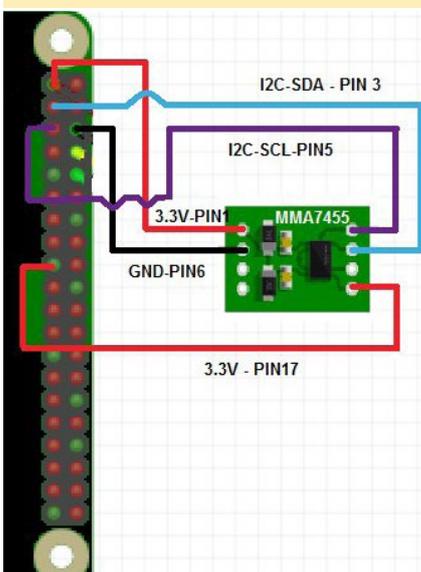
bituales del ODROID-C2:

- ODROID-C2
- Tarjeta MicroSD con la última versión de Ubuntu 16.04 proporcionada por Hardkernel (<http://bit.ly/2rDOCfn>)
- Librería WiringPi para controlar los GPIOs del ODROID-C2 que ejecuta Ubuntu 16.04. Las instrucciones de Hardkernel sobre cómo instalar la librería las puedes encontrar en <http://bit.ly/1NsrIU9>
- Teclado
- Pantalla
- Cable HDMI
- Alimentación por Micro USB o mejor aún, una fuente de alimentación proporcionada por Hardkernel (<http://bit.ly/1X0bgdt>)
- Opcional: Banco de energía con UBEC (3A max, 5V) si quieres utilizar el dispositivo de forma autónoma (ver Foto 1). Hardkernel ofrece una solución mejor con el UPS3 específicamente diseñado para ODROID-C2. Puede comprar este sistema de alimentación desde su tienda en este enlace: <http://bit.ly/211rE25>. El UPS3 es una buena opción, ya que proporciona al detector la capacidad de operar de forma autónoma con mayor estabilidad y duración.

- Cable Ethernet o dongle wifi usb y router Internet
- Un ventilador de PC de 90 mm, modificado para usarlo como placa de regresión
- El Kit Tinkering C sobre Ubuntu, que se puede comprar en Hardkernel (<http://bit.ly/1NsrlU9>)
- Módulo acelerómetro MMA7455, que se puede encontrar en varios sitios, incluyendo eBay.

Para el cableado, por favor sigue el esquema de la figura 2. Necesitarás 3.3V para el componente digital y analógico del acelerómetro, así que conecta el GPIO\_1 al Pin 1 del acelerómetro, que proporcionará los 3.3V. Tu cable de puesta a tierra debe conectar el Pin 6 del GPIO al Pin 2 del acelerómetro. El Pin 7, el ChipSelect (CS), conectado al Pin 17 de GPIO del ODROID-C2, añadiendo así otros 3.3V necesarios (activa low para SPI de lo contrario, como en este caso, I2C). Hay dos cables importantes para la comunicación: el SDA que proporciona los datos en serie I2C y el SCL que proporciona el reloj en serie I2C. El SDA está en el Pin 13 del acelerómetro y está conectado al Pin 3 GPIO del ODROID. Por último, pero no menos importante, el SCL está en el Pin 14 y está conectado al pin 5 GPIO del ODROID. Consulta de nuevo el esquema (Figura 2) y el excelente diseño de

Figura 2 - Esquema del cableado



40 pines de Hardkernel para ODROID-C2 (<http://bit.ly/2aXAlmt>) con el fin de colocar el cableado correctamente.

Ahora que tenemos nuestro hardware listo, vamos a ver cómo podemos crear una comunicación entre el ODROID-C2 y el acelerómetro MMA7455 utilizando el protocolo I2C.

## Comunicación I2C

Todos los comandos son introducidos en una ventana de terminal o vía SSH. Primero, tendrás que actualizar el C2 para asegurarte de que tienes instalados los paquetes más recientes:

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

Luego tendrás que reiniciar el ODROID-C2:

```
$ sudo reboot
```

Necesitarás instalar SMBus y I2C-Tools ya que el acelerómetro MMA7455 utiliza este protocolo para comunicarse con el ODROID-C2. El bus de Administración del Sistema (SMBus) es un simple bus de dos hilos para las comunicaciones livianas. Es muy común encontrarlo en las placas base de los ordenadores para comunicarse con la fuente de alimentación (<http://bit.ly/2rAWHuU>). El protocolo I2C es un bus serie de intercambio de paquetes, multi-maestro, multi-esclavo, inventado por Philips Semiconductor (ahora NXP Semiconductors). Normalmente se utiliza para conectar ICs periféricos de baja veloci-

dad a procesadores y microcontroladores en comunicaciones intra-placa de corta distancia (<http://bit.ly/2qGiYP4>).

Volviendo a nuestra guía, una vez hayas iniciado sesión en tu C2 desde la línea de comandos, ejecuta los siguientes comandos para instalar Python-SMBus y I2C-Tools:

```
$ sudo apt-get install python-smbus
$ sudo apt-get install i2c tools
```

Configura el ODROID-C2 para cargar el driver I2C:

```
$ modprobe ami-i2c
```

Configura el ODROID-C2 para iniciar I2C automáticamente al arrancar editando `/etc/modules`:

```
$ sudo nano /etc/modules
```

Usa las teclas del cursor para moverte hasta la última línea. Añade una nueva línea y escribe:

```
$ i2c-dev
```

Pulse Intro y luego añade:

```
$ am1_i2c
```

Guarda tus cambios y salte del editor nano. Para evitar tener que ejecutar I2C tools como root, añade el usuario "ODROID" al grupo I2C:

```
$ sudo adduser Odroid i2c
```

Después reinicia el ODROID-C2:

Figura 3 - Ejemplo del Resultado de un acelerómetro

```
odroid@odroid64: ~
File Edit View Search Terminal Help
odroid@odroid64:~$ i2cdetect -y -r 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

```
$ sudo reboot
```

Una vez que tu ODROID-C2 se haya reiniciado, tendrás soporte para I2C. Puedes comprobar si hay dispositivos I2C conectados con el siguiente comando:

```
$ sudo i2cdetect -y -r 1
```

Si aparece "1d" en la línea 10 bajo la columna D, significa que el acelerómetro se está comunicando con el ODROIDC2 y funciona correctamente (Figura 3). Puedes encontrar más detalles <http://bit.ly/2qCQM1s>.

## Software Python

Para facilitar la comprensión, vamos a presentar el código por trozos. Primero, importaremos los módulos necesarios:

```
import wiringpi2 as odroid
import time
from import sleep
```

Luego, configura la librería WiringPI:

```
odroid.wiringPiSetup()
DEBUG = 1
```

Configura la API y el retardo en el envío de datos para la plataforma ThingSpeak ([https:// thingspeak.com](https://thingspeak.com)). Consulta la siguiente sección del artículo para saber cómo registrarse y utilizar la plataforma ThingSpeak.

```
myAPI = "XXXXXXXXXXXXXXXX" # This
is the API key provided by Thing-
Speak
myDelay = 1 # How many seconds
between posting data
```

El siguiente código permite controlar el SMBus para comunicar el acelerómetro con el C2 a través del protocolo I2C:

```
class Accel():
myBus = 1
b = smbus.SMBus(myBus)
```

Define una clase llamada Accel para calibrar el sensor y leer valores de aceleración sobre los tres ejes (x, y, z). Esos valores representan las mediciones de la aceleración de la gravedad cuando las ondas sísmicas "golpean" el dispositivo:

```
def setUp(self):
    self.b.write_byte_
data(0x1D,0x16,0x55) # Set up the
Mode (sensitivity=2g)
    self.b.write_byte_
data(0x1D,0x10,0) # Calibrate
sensor
    self.b.write_byte_
data(0x1D,0x11,0) # Calibrate
sensor
    self.b.write_byte_
data(0x1D,0x12,0) # Calibrate
sensor
    self.b.write_byte_
data(0x1D,0x13,0) # Calibrate
sensor
    self.b.write_byte_
data(0x1D,0x14,0) # Calibrate
sensor
    self.b.write_byte_
data(0x1D,0x15,0) # Calibrate
sensor
def getValueX(self):
return self.b.read_byte_
data(0x1D,0x06)
def getValueY(self):
return self.b.read_byte_
data(0x1D,0x07)
def getValueZ(self):
return self.b.read_byte_
data(0x1D,0x08)
```

Crea el objeto MMA7455:

```
MMA7455 = Accel()
MMA7455.setUp()
```

Para el siguiente fragmento de código, necesitarás usar un poco las matemáticas para transformar los valores brutos del acelerómetro en algo con más sentido, utilizando el acelerómetro de un smartphone para la calibración. Puede que tengas que trabajar un poco con

tu teléfono, en tanto que las condiciones de trabajo de tu dispositivo para la calibración de estas fórmulas pueden variar el resultado.

```
def get SensorData():
    x = MMA7455.getValueX()
    y = MMA7455.getValueY()
    z = MMA7455.getValueZ()
    x2=((x-115)%256)-(128)/
float(100) # Calibrate values x
    y2=((y-226+128)%256)-128)/
float(100) # Calibrate values y
    z2=((z-126+128)%256)-128)/
float(100) # Calibrate values z
mag=(math.
sqrt(x2*x2+y2*y2+z2*z2))
time.sleep(01) - Sampling every
1s
return (str(x2), str(y2),
str(z2), str(mag))
```

Después, encontrarás la función main que te permite publicar los datos en la plataforma ThingSpeak y visualizar los datos a medida que las ondas sísmicas "golpean" el dispositivo. Recuerda que el detector de terremotos es capaz de registrar vibraciones sobre los ejes X, Y, Z.

```
# main() function
def main():
```

Finalmente, imprime los resultados:

```
print "starting..."

baseURL = "https://api.thing-
speak.com/update?api_key=%s" %
myAPI

print (baseURL)

while True:
try:

x2, y2, z2, mag = getSensorData()

$ f = urllib2.urlopen(baseURL +
"&field1=%s&field2=%s&field3=%s&field
4=%s" % (x2, y2, z2, mag))
```

El resto es puro código funcional:

```
$ sleep(int(myDelay))
    $ except:
    $ print "exiting."
    $ break

$ # call main
$ if_name_=="_main_":
    $ main()
```

## ThingSpeak

Según sus desarrolladores en <http://bit.ly/2qrCMoE>, ThingSpeak es una API y una aplicación de código abierto del Internet de las cosas (IoT) para almacenar y recuperar datos de cosas que usan el protocolo HTTP vía Internet o a través de una red de área local. ThingSpeak permite crear aplicaciones de registro de sensores, aplicaciones de seguimiento de localizaciones y una red social de cosas con actualizaciones de estado.

Tras inscribirte en el servicio (el procedimiento es autoexplicativo), inicia sesión en la plataforma ThingSpeak con tus credenciales. Tendrás que crear un "New Channel" para visualizar tus datos sismográficos. Desde la página de inicio, haga clic en "My Channels" y luego en "New Channel". Consulta la Figura 4 para conocer cómo configurar tu canal.

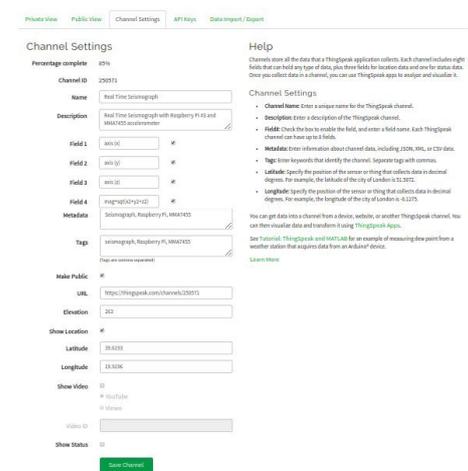


Figura 4 - Cómo configurar tu canal

El "Name" y la "Description" del canal son obvios. Rellena estos campos con algo descriptivo. Lo que más importa son los cuatro campos que definimos

(es decir, los campos 1, 2, 3 y 4). Naturalmente, estos campos representen la gravedad de aceleración en los tres ejes (x, y, z) más uno, el campo 4 que representa la magnitud de la aceleración sísmica. Rellena los respectivos campos con las descripciones apropiadas. Por último, inserta la "Latitude", "Longitude" y la "Elevation" de tu dispositivo para una representación adecuada del lugar en Google Maps (quinto gráfico en la parte inferior). ¡Se trata de una función pseudo GPS, pero es mejor que nada!

Tan pronto como guardes tu Canal, se le asignará un nuevo "ID Channel". Este "ID Channel" también forma parte de tu dirección web. Por ejemplo, nuestro "ID Channel" es "25071", de modo que nuestros datos serán accesibles a través de esta dirección de Internet: <http://www.thingspeak.com/25071>.

Por supuesto, esta dirección es accesible desde cualquier ordenador, tablet, portátil o teléfono móvil, una característica muy potente de este dispositivo.

Además, necesitarás una clave API para acceder a la plataforma a través de nuestro código. A la(s) clave(s) API se puede acceder a través de la pestaña "API Keys" en la parte superior de la página.

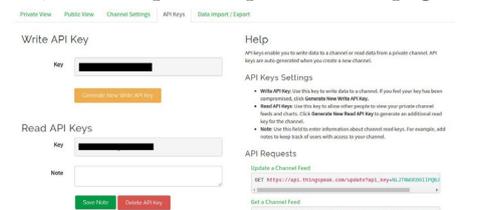


Figura 5 - Interfaz web de las claves API

Tan pronto como hagas clic en ella aparecerá la siguiente página (Figura 5).

El campo más importante es "Write API Key", ya que éste escribirá o publicará datos en la plataforma ThingSpeak. Recupera las líneas de tu código python en "seismos.py", líneas que hacen uso de esta API Write key:

```
$ myAPI = "XXXXXXXXXXXXXXXXXX" # This
is the API key provided by the
ThingSpeak platform.
$ ---
```

```
$ baseURL = "https://api.thingspeak.
com/update?api_key=%s" % myAPI
```

Esto es todo lo que necesitará en relación a Thing-Speak y a la configuración de la plataforma para plasmar tus datos en el dispositivo.

## Simulando ondas sísmicas

¿Cómo puedes probar tu detector de terremotos? ¡Por supuesto, no te va a quedar sentado esperando a que tenga lugar un terremoto! Puedes crear un mecanismo de regresión para producir vibraciones en el dispositivo deseado. Esta denominada "placa de regresión" se puede crear fácilmente usando un ventilador de refrigeración de PC al que hay que realizar unas pequeñas modificaciones. En nuestro caso, cogemos un ventilador de PC de 90mm al que cortaremos tres de sus siete aspas giratorias. Al

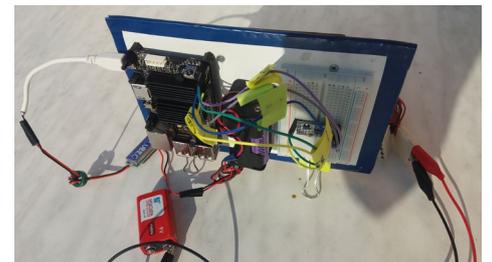


Foto 2 - Vista lateral del sismógrafo

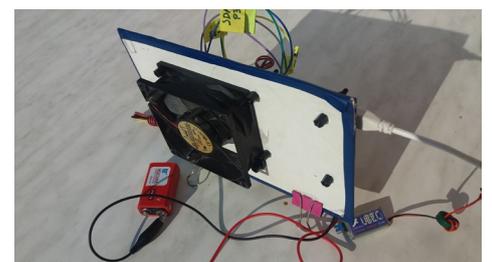


Foto 3 - Vista lateral opuesta

hacer esto, creamos una "desalineación", un mecanismo de rotación que vibre (Foto 3). Como paso final, colocaremos este improvisado mecanismo debajo de la placa del detector de terremotos.

## Probando el detector

Escribe lo siguiente en una ventana de Terminal para probar el sismógrafo:

```

odroid@odroid64: ~/Downloads
File Edit View Search Terminal Help
odroid@odroid64:~/Downloads$ sudo python seismos.py
starting...
https://api.thingspeak.com/update?api_key=NLJTNWUCO6IIPQB3
34067
-0.02 -0.01 0.1 0.10246950766
34068
0.01 0.0 0.1 0.100498756211
34069
-0.01 0.0 0.1 0.100498756211
34070
-0.02 0.0 0.1 0.101980390272
34071
-0.02 -0.01 0.1 0.10246950766
34072
0.0 -0.02 0.1 0.101980390272
34073
-0.01 0.0 0.1 0.100498756211
^Cexiting.
odroid@odroid64:~/Downloads$ scrot
    
```

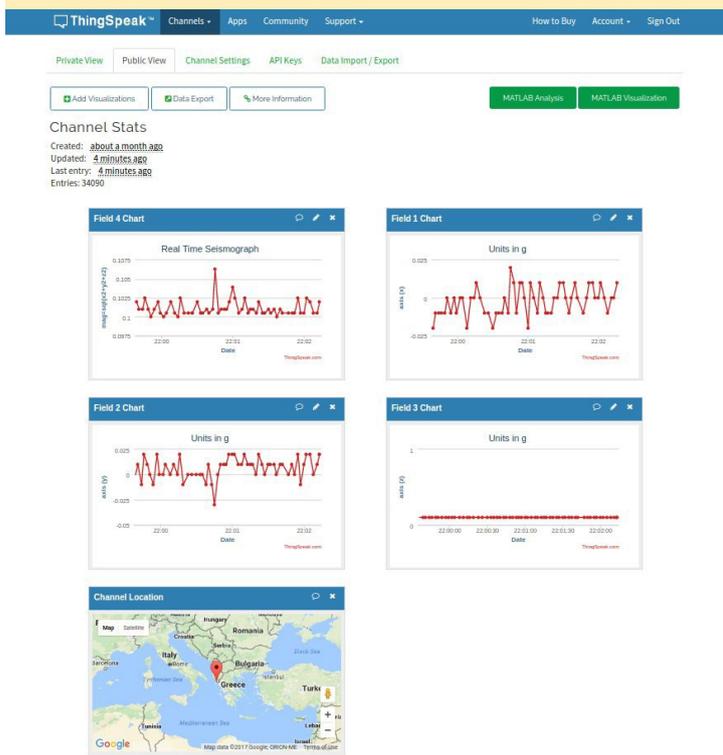
**Figura 6 - Resultado de Seismos.py para ThingSpeak**

```
$ sudo python seismo.py
```

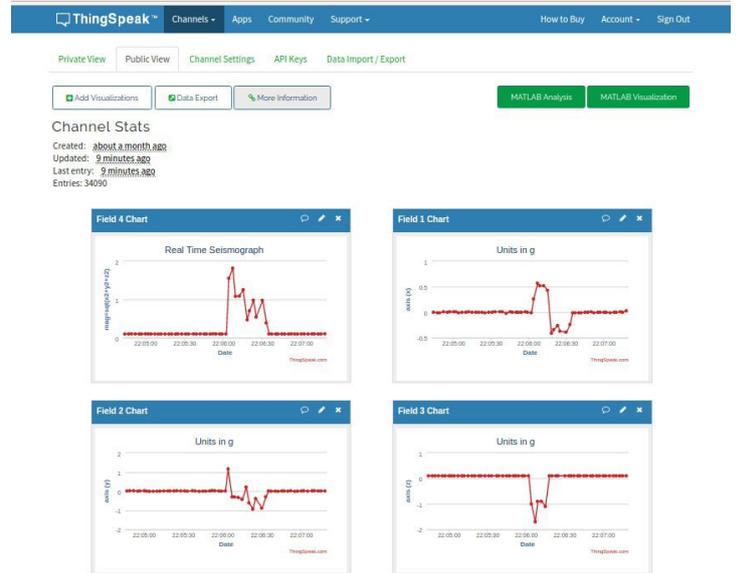
Seismos.py envía inmediatamente los datos a ThingSpeak, y además reporta los datos localmente en el terminal (Figura 6).

Los valores de la gravedad de la aceleración son reportados cada segundo y luego se suben a la plataforma ThingSpeak, que transforma esos valores en gráficos. En total, hay cuatro gráficos, el primero es el más importante ya que representa la aceleración sísmica con los valores del resto de gráficos (tres en total). En otras palabras, esta es la magnitud de la aceleración sísmica (Mg) como resultado de esta fórmula matemática ( $Mg = \dots$ ). Lo mejor es que estos gráficos son accesibles a través de ThingSpeak (en <http://bit.ly/2s0mDqM> en nuestro caso) desde cualquier dispositivo conectado a la red, ordenadores, tablets, portátiles y teléfonos móviles. Puede que notes vibraciones menores que van desde -0,3g a + 0,3g antes de empezar a zandar el dispositivo usando tu ventilador de regresión, pero estas vibraciones son simplemente ruido de tu entorno (Figura 7).

**Figura 7 - Gráficos de vibraciones del sismógrafo**



El último gráfico representa las coordenadas GPS del dispositivo en Google Maps. Este no es el epicentro del terremoto, sino el lugar donde se encuentra tu dispositivo sismógrafo. Se necesita al menos tres de estos dispositivos para determinar el epicentro del terremoto con cierta precisión, pero esto está fuera del alcance de este proyecto. Por ahora zandaremos el detector usando el ventilador de regresión conectado a una



**Figura 8 - Ejemplo del resultado al hacer vibrar el detector**

batería. ¿Puedes notar la diferencia (Figura 8)?

Verás que la magnitud de la aceleración sísmica muestra valores más altos que antes, mucho más altos, a medida que la simulación de las ondas de terremoto incide en el detector.

## Tweet, tweet, tweet

Ahora que tenemos instalado el detector de terremotos, vamos a ver cómo usar el detector para informar a la gente de cuándo y si ha ocurrido un suceso catastrófico (esperamos que no). Esta parte es bastante fácil una vez que decides el valor de referencia que debe activar los tweets que se enviarán.

En la plataforma ThingSpeak, en el menú de la parte superior de la página, selecciona "Apps" y, a continuación, "Actions", y finalmente "Reacts". Quieres que el dispositivo reaccione cada vez que la aceleración sísmica se encuentre por encima

**Figura 9 - Configuración de los parámetros para el detector**

**Analytics**



**MATLAB Analysis**  
Explore and transform data.



**MATLAB Visualizations**  
Visualize data in MATLAB plots.



**Plugins**  
Display data in gauges, charts, or custom plugins.

**React**  
React when channel data meets certain conditions.

**TalkBack**  
Queue up commands for your device.

**ThingHTTP**  
Simplify device communication with web services and APIs.

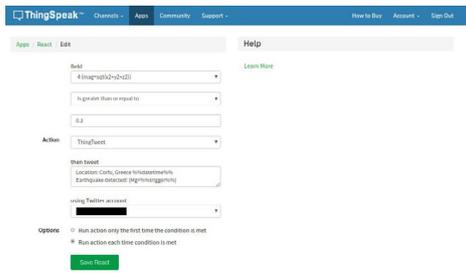


Figura 10

del límite que has elegido. Haga clic en “New React” y ajusta los parámetros para este evento, véase la Figura 9.

Estas cosas son más o menos autoexplicativas, pero, en caso de que necesites alguna aclaración, preste atención al campo “Condition”. Esta es la parte más crítica de la aplicación, ya que aquí es

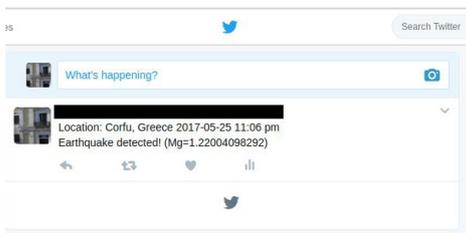


Figura 11

donde fijas tu límite de referencia. Por ejemplo, nosotros fijaremos la aceleración sísmica (fórmula) mayor o igual a 0,3 (nuestro límite). Puesto que esta fórmula está representada por el gráfico 4 en el campo 4, lo procesa así (Figura 10).

En “Action”, elije “ThingTweet” y rellena el campo “then tweet” con el texto que quieres que aparezca en todos tus tweets (por ejemplo, Ubicación: Corfú, Grecia %% datetime %%, %% trigger %%). Finalmente, selección la opción “Run action each time condition is met”, ya que quieres que tu “React” se active cada vez que se cumpla la condición. Los resultados se mostrarán en nuestro flujo de Twitter (Figura 11).

## Conclusión

Tienes total libertad para hacer cualquier modificación o añadir funciones adicionales a nuestro detector de terremotos, ya que este dispositivo sin duda podrá tener alguna utilizada en el campo de la ciencia.

# GOGS

## UNA ALTERNATIVA A GITHUB/GITLAB

por @redrocket



**G**ogs es un servicio Git auto-distribuido. El objetivo del proyecto es el de configurar de la forma más fácil, rápida y eficiente un servicio Git auto-distribuido. Con Go, esto se puede hacer con una distribución binaria independiente en todas las plataformas que soporten Go, incluyendo Linux, Mac OS X, Windows y ARM. En este artículo se describen los pasos necesarios para instalar Gogs en un ODROID-C2 ejecutando Ubuntu Mate, Apache2, MySQL y PHP 7.0. ¡Asegúrate de hacer una copia de seguridad de todos los archivos antes de intentar instalar Gogs!

## Compilación

Tenemos que compilar la fuente nosotros mismos, porque yo no fui capaz de ejecutar las imágenes de linux armv5\_x64 y RasPi armv6 que ofrece el sitio web de Gogs. Primero, crea una carpeta en tu directorio de inicio donde vamos a descargar las fuentes, luego navega hasta esa carpeta:

```
$ mkdir gogs_sources
$ cd gogs_sources
```

Luego, instala el lenguaje GO y Git:

```
$ sudo apt-get install git -y
$ sudo apt-get install golang -y
```

Usando Go, descarga Gogs junto con sus dependencias:

```
$ go get -u github.com/gogits/gogs
```

Copia y pega cada línea de abajo en una ventana de Terminal, donde \$USER es tu nombre de usuario en Linux:

```
$ echo `export GOPATH=/
home/$USER/go` >> $HOME/.bashrc
$ echo `export PATH=$PATH:GOPATH/
bin` >> $HOME/.bashrc
$ source $HOME/.bashrc
```

Compila el código fuente:

```
$ cd src/github.com/gogits/gogs
$ go build
```

Si no aparecen errores durante la compilación, ejecuta el siguiente comando para testear el resultado:

```
$ ./gogs web
```

Si no se muestran errores, presiona Ctrl + C para detener el servidor Gogs. Mueve toda su carpeta:

```
$ cp gogs ~/gogs/
$ cp -a templates/ ~/gogs/
$ cp -a scripts/ ~/gogs/
$ cp -a public/ ~/gogs/
```

Según la documentación, la versión 0.6.0 y superior te permite crear un archivo de configuración app.ini, que debería estar ubicado en ~/gogs/custom/conf/app.ini:

```
$ pico ~/gogs/custom/conf/app.ini
```

Pega el siguiente fragmento de configuración en el archivo:

```
[repository]
ROOT = /path/to/repos-folder

[database]
PASSWD = `root`
```

No te preocupes por la línea PASSWD. Por ahora, introduce la contraseña de root y más adelante, cuando instales gogs y crees un usuario/pass específicos, esa línea cambiará. Guarda y cierre el archivo presionando Ctrl + X -> Y -> Intro.

## Preparando MySQL

Para preparar MySQL para la instalación de Gogs, utilicé PHPMyAdmin para acceder a MySQL y ejecuté el archivo /scripts/mysql.sql. Para ello, inicie sesión a través de PHPMyAdmin, luego pinche en “Importar” y seleccione el archivo /home/\$USER/gogs/scripts/mysql.sql. Te sugiero que crees un usuario específico para la base de datos de Gogs, y luego uses ese nombre de usuario/contraseña para finalizar la instalación de Gogs usando la opción DB -> Privileges -> Add user account.

## Instalación

En una ventana Terminal, navega hasta la carpeta Gogs y ejecuta:

```
$ cd ~/gogs
$ ./gogs web
```

Abre un navegador y navega hasta <http://localhost:3000>. Sigue las instrucciones de la pantalla para finalizar la instalación. En la pantalla de instalación, presta especial atención al campo user. Por defecto es “git”, pero si añadiste un usuario diferente, tienes que cambiarlo. Llegados a este punto deberías tener un servidor git funcionando en tu red interna. Si quieres que sea accesible desde fuera de tu red, continúa leyendo.

Si está intentando acceder desde fuera de tu red, no olvide cambiar la URL de la aplicación a una específica, como <http://git.domain.com>. Antes comenté que mi ODDROID-C2 tiene instalado Apache, que utilizo para que mis clientes puedan ver mi trabajo en curso. Por lo tanto, si queremos que nuestro servidor Gogs sea accesible desde Internet, debemos combinar Apache con un módulo proxy.

## Preparando Apache

Para configurar Apache y poder reenviar las peticiones desde Internet hacia el servidor Gogs, primero debemos comprobar si tenemos módulos proxy\* instalados en nuestro servidor Apache:

```
$ apache2ctl -M
```

Al finalizar este comando, aparecerá una pregunta sobre qué módulos deseas instalar y entre paréntesis, se indicará que los comodines están permitidos. Simplemente escribe lo siguiente y presionar la tecla Intro:

```
proxy*
```

Después, crea el archivo git.config:

```
$ sudo pico /etc/apache2/\
sites-available/git.conf
```

Pega el siguiente fragmento de configuración en el archivo, asegúrate de reemplazar \$USER por tu nombre de usuario en Linux:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /home/$USER/gogs/
    public
    ProxyPreserveHost On
    ProxyPass / http://local-
host:3000/
    ProxyPassReverse / http://loc-
alhost:3000/
    ServerName git.domain.com
    ServerAlias git.domain.com
</VirtualHost>
```

Guarda y cierre el archivo presionando Ctrl + X -> Y -> Intro.

A continuación, activa la nueva configuración y reinicia el servicio Apache para que la nueva configuración tenga efecto:

```
$ sudo a2ensite git.conf
$ sudo service apache2 restart
```

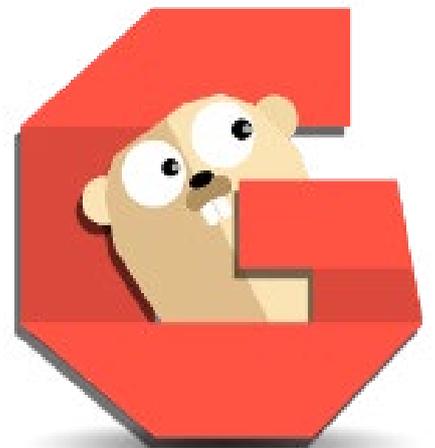
Reza una oración, y cuando abras <http://git.domain.com>, deberías ver la pantalla de inicio de Gogs. Si ve un error 503 o algo más, deberías comprobar si el servidor Gogs se está ejecutando. Si no es así, inícialo desde la consola:

```
$ cd ~/gogs
$ ./gogs web
```

Finalmente, activa gogs.service para que cada vez que reinicies tu C2, Gogs se ejecute automáticamente:

```
$ sudo cp ~/gogs/script/systemd/
gogs.service /etc/systemd/system/
$ sudo systemctl enable gogs
$ sudo systemctl start gogs
```

Para comentarios, preguntas y sugerencias, visita el hilo original en <http://bit.ly/2sw41jz>.



# ODROID-XU4 COMO SERVIDOR DE MAPAS

## UNA GUIA PARA USAR GEOSERVER

por José Cerrejón



Voy a empezar un nuevo proyecto para servir capas de mapas a algunos de mis clientes agrícolas. Estos agricultores nos van a dejar estudiar su tierra y almacenar esa información en la nube, para que esté siempre online y lista para su acceso.

Es un proyecto fascinante, inicialmente pensamos en usar un Raspberry Pi pero su Ethernet 10/100 y 1GB de RAM eran insuficientes para nuestros objetivos, así que nos decantamos por una solución basada en el ODROID-XU4 que cubría perfectamente nuestras necesidades. Teníamos el hardware ya de por sí, aunque olvidado en un cajón. En este artículo, te voy a hablar de mis andanzas con la configuración de un ODROID-XU4 usando DietPi como sistema operativo y Geo Server como entorno de desarrollo.

### DietPi

Usando Etcher, instalé la última versión de DietPi, que descargué de su sitio web. El proceso de instalación fue sencillo, no tuvimos ningún problema en absoluto. El sistema operativo detectó que estaba usando un XU4 y fijó el Kernel al 4.9 con un ID único para esta placa. Las opciones que he usado son:

#### DietPi-Config:

- Performance Options > CPU Throttle Up: 85%
- Performance Options > CPU Governor: Elegí 'interactive' aunque, 'performance' debería ser la opción más adecuada.
- Advanced Options > Swap File: 1 (Auto size)
- Language and Regional Options > Timezone y keyboard
- Security Options > Cambiar la contraseña de root y nombre de host (fue necesario reiniciar)

#### Software:

- Fail2Ban
- Tomcat8

```
DietPi | 10:12 | Thu 18/05/17
-----
v150 | oDroid XU3/4 (armv7l)
-----
IP Address | 192.168.1.9
-----

Created by : Daniel Knight
Web       : http://DietPi.com
Twitter  : http://twitter.com/dietpi_
Donate   : http://goo.gl/pzISt9

Device image possible thanks to: Meveric
DietPi's web hosting is powered by: MyVirtualServer.com

dietpi-launcher = All the DietPi programs in one place.
dietpi-config   = Feature rich configuration tool for your device.
dietpi-software = Select optimized software for installation.
htop            = Resource monitor.
cpu             = Shows CPU information and stats.

root@geomapas:~# uname -a
Linux geomapas 3.10.104+ #1 SMP PREEMPT Tue Feb 21 14:20:54 CET 2017 armv7l GNU/Linux
root@geomapas:~# free -h
              total        used        free      shared    buffers     cached
Mem:           1.9G         316M         1.6G          8.0M          6.1M          66M
-/+ buffers/cache: 243M         1.7G
Swap:          0B           0B           0B
root@geomapas:~# netstat -plntu | grep 8080
tcp6          0          0 :::8080          :::*             LISTEN         1049/java
```

Figura 1 – DietPi en funcionamiento

- Midnight Commander
- Build-Essentials
- Git Client

Puedes elegir SD/EMMC (no recomendado) o un disco duro externo para la ubicación de los datos de usuario. Deje que todo se instale y me tome un café mientras espera. Tras el reinicio, comprobé que el servidor SSH funcionaba y que Tomcat se estaba ejecutando.

### GeoServer

Contamos con dos métodos para instalar GeoServer: archivos binarios o un archivo .war para Tomcat. Yo elegí la segunda opción. Puedes visitar la página oficial para instalar la última versión. En el momento de la publicación de este artículo, la última versión era la 2.11.0.

```
$ wget \
http://sourceforge.net/projects/geoserver/\
files/GeoServer/2.11.0/geoserver-2.11.0-war.zip
```

```
root@geopma:~# wget http://sourceforge.net/projects/geoserver/files/GeoServer/2.11.0/geoserver-2.11.0-war.zip
--2017-05-18 10:18:25-- http://sourceforge.net/projects/geoserver/files/GeoServer/2.11.0/geoserver-2.11.0-war.zip
Resolving sourceforge.net [sourceforge.net]: 216.34.181.60... connected.
HTTP request sent, awaiting response... 302 Moved Permanently
Location: https://sourceforge.net/projects/geoserver/files/GeoServer/2.11.0/geoserver-2.11.0-war.zip [following]
--2017-05-18 10:18:25-- https://sourceforge.net/projects/geoserver/files/GeoServer/2.11.0/geoserver-2.11.0-war.zip
Connecting to sourceforge.net [sourceforge.net]:216.34.181.60:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://sourceforge.net/projects/geoserver/files/GeoServer/2.11.0/geoserver-2.11.0-war.zip/download [following]
--2017-05-18 10:18:25-- https://sourceforge.net/projects/geoserver/files/GeoServer/2.11.0/geoserver-2.11.0-war.zip/download
Connecting to sourceforge.net [sourceforge.net]:216.34.181.60:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/geoserver/GeoServer/2.11.0/geoserver-2.11.0-war.zip?r=4959550t&use_mirror=kent [following]
--2017-05-18 10:18:27-- https://downloads.sourceforge.net/project/geoserver/GeoServer/2.11.0/geoserver-2.11.0-war.zip?r=4959550t&use_mirror=kent
Resolving downloads.sourceforge.net [downloads.sourceforge.net]: 216.34.181.59... connected.
Connecting to downloads.sourceforge.net [downloads.sourceforge.net]:216.34.181.59:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://kent.di.sourceforge.net/project/geoserver/GeoServer/2.11.0/geoserver-2.11.0-war.zip [following]
--2017-05-18 10:18:27-- https://kent.di.sourceforge.net/project/geoserver/GeoServer/2.11.0/geoserver-2.11.0-war.zip
Connecting to kent.di.sourceforge.net [kent.di.sourceforge.net]: 213.219.86.485... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71590590 (68M) [application/octet-stream]
Saving to: 'geoserver-2.11.0-war.zip'

geoserver-2.11.0-war.zip 100% [#####] 68.27M 6.240M/s in 12s

2017-05-18 10:18:41 (5.50 MB/s) = 'geoserver-2.11.0-war.zip' saved [71590590/71590590]
root@geopma:~# unzip geoserver*-war.zip
Archive:  geoserver-2.11.0-war.zip
  inflating: geoserver-war
  inflating: geoserver-war
  inflating: GDAL.txt
  inflating: LICENCE.txt
  creating target/
  inflating: target/VERSION.txt
root@geopma:~# cp geoserver-war /var/lib/tomcat8/webapps
```

Figura 2 - Configurando GeoServer

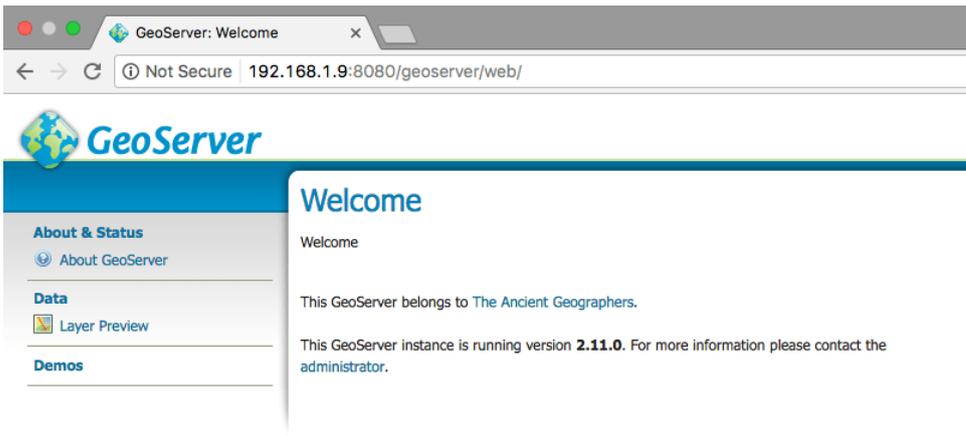


Figura 3 - GeoServer ejecutándose

```
$ unzip geoserver*-war.zip
$ cp geoserver-war \
/var/lib/tomcat8/webapps
```

Ahora el archivo \*.war\* está descomprimido en el sistema de archivos. ¡Puede tardar un par de minutos, después tu servidor de mapas estará listo!

Navega hasta la parte superior derecha de la interfaz web para iniciar sesión en GeoServer. Las credenciales de administrador por defecto son las siguientes:

User name: admin  
Password: geoserver

## GDAL

GeoServer soporta gran variedad de formatos raster, aunque a veces aparecen formatos no estándar y GDAL. Necesitamos instalarlo escribiendo el siguiente comando en una ventana de terminal:

```
$ sudo apt-get install gdal-bin
```

Ahora necesitas instalar una extensión en Tomcat. Dirígete a la página de descarga de GeoServer, haga clic en extensión, descarga la librería correcta y cópiala en el directorio /var/lib/tomcat8/webapps/geoserver/WEB-INF/lib/. Tienes un buen compañero en la herramienta gdal. El comando gdalinfo te ayudará a obtener información sobre los archivos ráster. Esto es muy útil, ¡créeme!

## Cargando algunos archivos

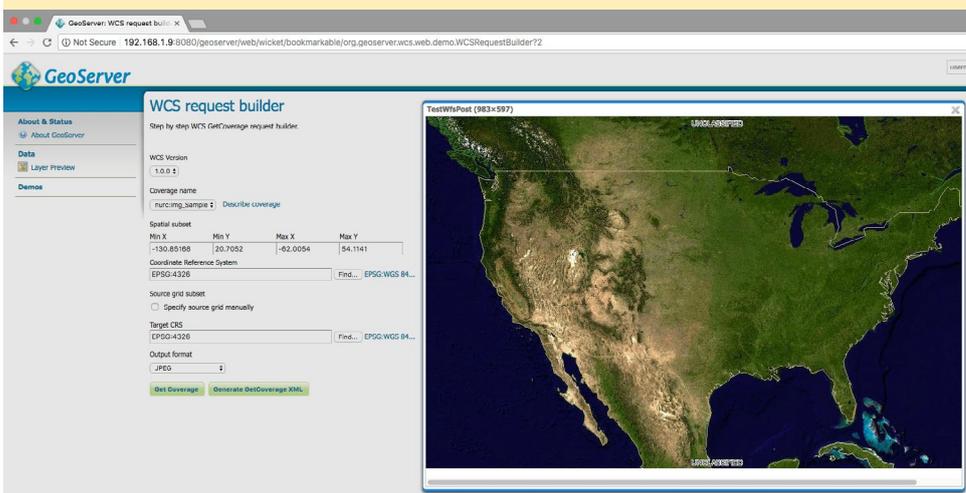
Podemos seleccionar una demo en el menú izquierdo para comprobar si funciona, tal y como se muestra en la Figura 4. En la Vista previa de capas, puedes ver algunos ejemplos. Simplemente haga clic en Openlayers para verlo en acción.

## Mejoras

Todavía tenemos mucho trabajo por hacer en el sistema:

- Asegurar las conexiones SSL
- Hacer cumplir la seguridad
- Configurar Map Server para guardar y utilizar archivos desde un almacenamiento externo
- Un visor de mapas como Leaflet
- Un panel de administración con Laravel, queremos montar un panel fácil de usar que permita controlar el sistema y sus recursos
- Panel para usuario y agricultores para que puedan ver las estadísticas y añadir o quitar capas

Figura 4 - GeoServer con mapas cargados



## Notas finales

Existe mucha documentación que nos puede ayudar a que un servidor de mapas se ejecute apenas sin esfuerzo. Algunas pruebas de rendimiento nos muestran que un ODROID-XU4 tiene la suficiente capacidad para servir mapas y rasters. Puede que sea un poco más lento de lo deseado en producción, pero por ahora, es una perfecta plataforma para realizar pruebas.

# CONTADOR GEIGER

## DETECTANDO RADIACION CON UN ODROID-C2

por @jojo

**E**n este proyecto de registro de datos de radiación, utilicé un ODROIDC2 y una placa detectora Geiger. Esta reseña inicial únicamente cubre la configuración básica del hardware y una mínima programación. En un futuro artículo, me gustaría presentar gráficos de visualización de datos en un sitio web, localizables por fechas y horas. Si alguien está interesado en contribuir en esta iniciativa, puede ponerse en contacto conmigo.

Por ahora, simplemente he importado los archivos log en LibreOffice Calc para crear gráficos manualmente. La figura 1 muestra mi configuración.

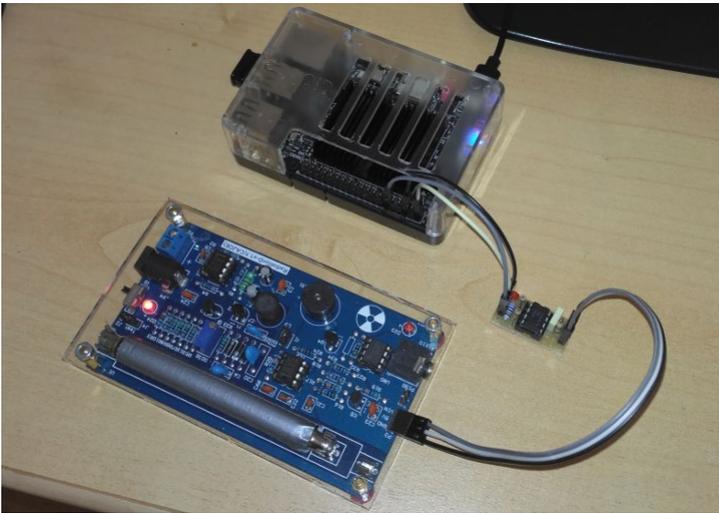


Figura 1 - El contador Geiger conectado a la ODROID-C2

### Hardware

La barata placa detectora que utilicé se puede conseguir en varias tiendas online como Aliexpress. Sin embargo, tuve que hacer frente a dos problemas a la hora de utilizarla:

1. El tubo Geiger (J305 beta) parecía ser muy sensible a la luz. No sé si esto era normal, pero era realmente extraño ver que los niveles de radiación aumentar periódicamente cada día. Observé que sucedía cuando el sol incidía directamente sobre mi escritorio, así que decidí



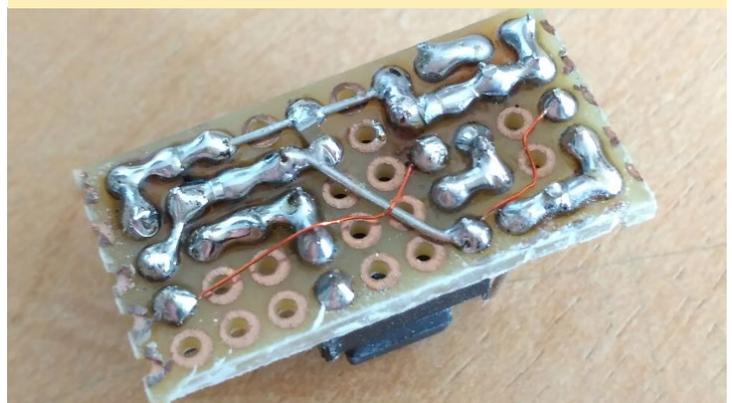
pintar el tubo de cristal de negro y además, lo cubrí con una cinta no transparente. Esto redujo drásticamente la sensibilidad a la luz.

2. El “paso conductor” en la placa detectora era extremadamente débil. La placa no era capaz de mover el pin del puerto de entrada del C2 a un nivel definido, de modo que tuve que añadir otro paso conductor. Utilicé una LMC555 (variante CMOS de la NE555) en modo de operación monoestable. Cuando es activada por la placa detectora, la LMC555 proporciona un pulso de 500  $\mu$ s de longitud, lo cual hacía frente a la señal tan débil y a veces “sucia” de la placa detectora.

Figura 2 - Parte superior de la placa conductora



Figura 3 - Parte inferior de la placa conductora



Tanto la placa detectora como la placa conductora se alimentan directamente por el cabezal GPIO del C2, y no necesitan ninguna fuente de alimentación adicional.

## Software

Escribí un pequeño programa C, inspirándome principalmente en este proyecto, <http://bit.ly/2tmPv0N>. Añadí las funciones de filtrado de datos y actualizaciones de datos a la consola y a un archivo. El programa creará un nuevo archivo en el momento en el que se inicie por primera vez. Si continúa ejecutándose hasta que empiece un nuevo día, se creará un nuevo archivo log cada día. El nombre del archivo log incluirá la fecha y la hora para facilitar su clasificación en un explorador de archivos. El programa no es muy bonito y aún no está terminado, pero por ahora, funciona tal como está diseñado.

Puedes descargar el archivo fuente (geiger\_counter\_v6.c) desde el post original en <http://bit.ly/2s0plwC>.

Una vez completada la descarga, compila el código:

```
$ gcc -o <your-download-directory>/\
geiger_counter_v6.c \
-lwiringPi -lpthread
```

Después, simplemente inicia el programa:

```
$ sudo Documents/Geiger_Counter
```

Este proyecto “no es un misterio”, es más un proyecto para principiantes. Espero que le sirva de inspiración a todo el mundo para que experimenten.

# LAKKA 2.0

## UN EXCELENTE SISTEMA OPERATIVO DE EMULACION RECIBE UNA IMPORTANTE ACTUALIZACION

por @synportack24



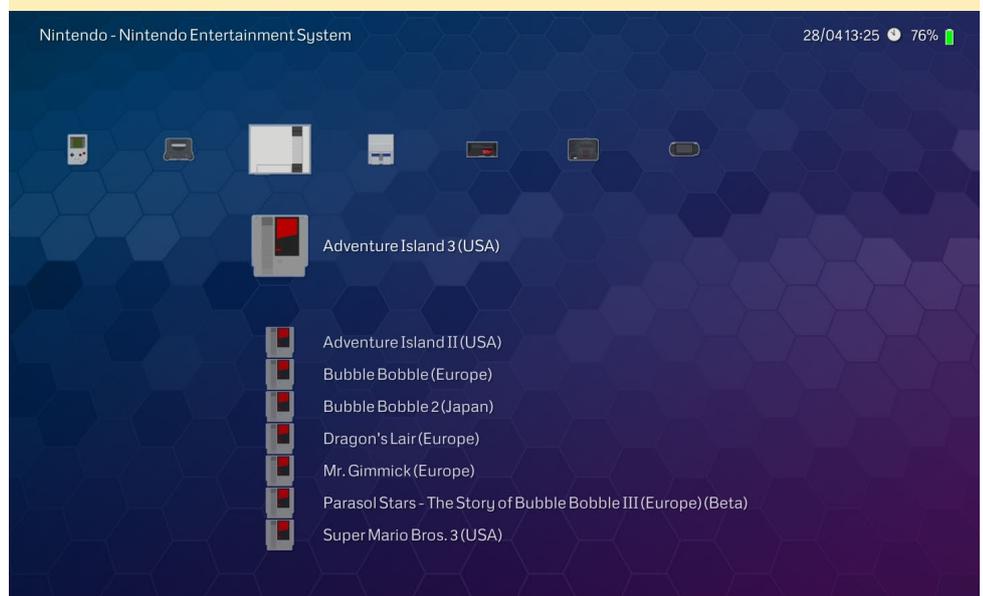
Lakka es un sistema operativo de emulación de juegos que adornó las páginas de ODROID magazine en 2015 con soporte para el ODROIDC1. Lakka ha recorrido un largo camino en este tiempo, con una

larga lista de características y “cores” que se han ido añadiendo con cada lanzamiento. Ahora, con la versión 2.0, Lakka tiene soporte completo para todos los dispositivos ODROID de Hardkernel. Como muchos jugadores retro apasiona-

### Important changes

- LibreELEC 8.0 stable rebase
- RetroArch 1.5.0
  - Simplified menu
  - Intuitive netplay: you create or join netplay rooms directly from the menu.
  - Ability to change the icon set on the fly
  - Revamped virtual keyboard
  - Korean language support
- New server for downloads and updates
- Almost all libreto cores are now enabled on every image

Figura 1 - Ejecutando juegos NES sobre Lakka 2.0



dos ya conocen, no hay escasez de Sistemas Operativos de emulación para cada plataforma ODROID.

Una característica que siempre me ha hecho volver a Lakka ha sido su interfaz limpia y su sistema de actualización fácil de usar, Lakka 2.0 hace bien en mantener esa tradición. Desde el principio, serás recibido con la familiar y simple interfaz lineal, con la excepción de un fondo más animado. Para mí, la mejor característica que ha traído consigo esta actualización es la posibilidad de conectarse a una red inalámbrica directamente desde la GUI.

Realice algunas pruebas con la nueva versión de Lakka en un ODROID-C2 y ODROID-XU4 con un mando XBOX360 por cable. Todos los drivers necesarios para el mando XBOX y para muchos otros vienen preinstalados, así que simplemente lo enchufé y listo para jugar. Tras conectarme a mi red WiFi, me dirigí a la configuración y habilité Samba para copiar rápidamente un par de juegos de prueba. No tuve problemas ni con el XU4 ni con el C2 a la hora de ejecutar juegos de SNES o Sega Genesis. El XU4 comprensiblemente le era más fácil ejecutar los juegos de N64. Lakka soporta muchos emuladores, o “cores”, como se les conoce. Un solo sistema pu-

ede tener múltiples “cores” que puedes usar para ejecutar un juego.

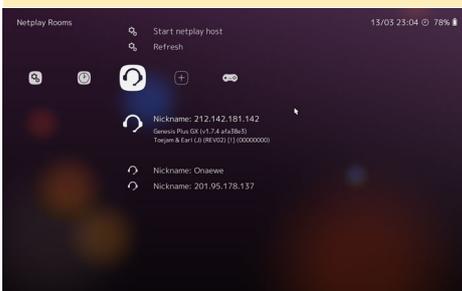
## Novedades

Estas son algunas de las novedades más destacadas que hay disponibles en Lakka 2.0:

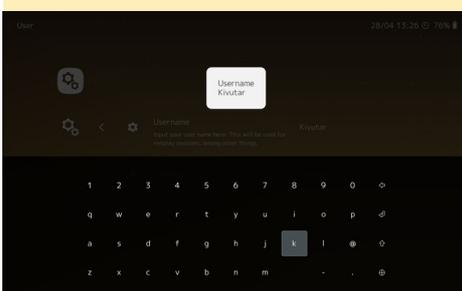
- Un nuevo modo netplay muy intuitivo, lo que significa que, si tienes amigos en diferentes lugares y aún quieres jugar con ellos, puedes hacerlo.
- Basado en LibreELEC actualizado a la versión 8 y RetroPi 1.5
- Han sido añadidos nuevos cores y juegos, incluyendo Easy RPG, UAE4ARM (emulador Amiga), VICE (emulador Commodor 64), PocketCDG (jugador karaoke) y Mr.Boom (clon de bomberman)
- Soporte para todos los mandos y adaptadores de mandos más conocidos (como el Wii U GameCube Adapter)
- Soporte para perfiles

Echa un vistazo a Lakka y descarga una imagen pre-compilada para el ODROID-C2 o ODROID-XU4 en [www.lakka.tv](http://www.lakka.tv).

**Figura 2 - Lakka 2.0 soporta rom de Netplay**



**Figura 3 - Creando un nombre de usuario**



# ODROID Magazine esta en Reddit!

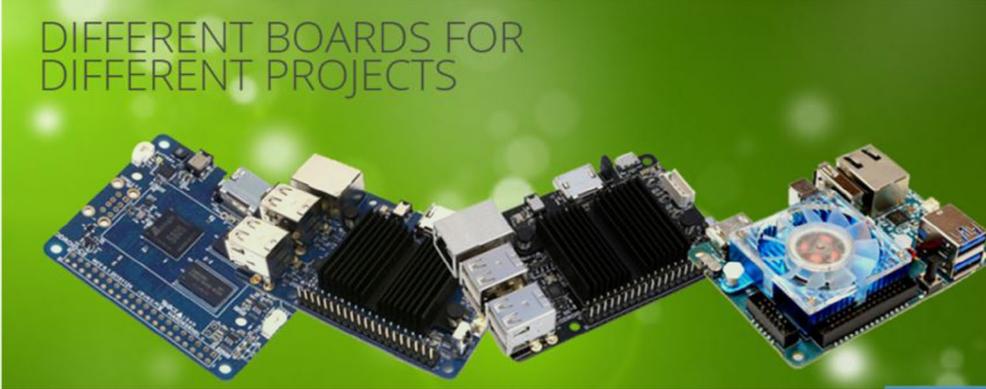


**ODROID Talk  
Subreddit**  
[www.reddit.com/r/odroid](http://www.reddit.com/r/odroid)



# ODROID

DIFFERENT BOARDS FOR  
DIFFERENT PROJECTS



Single Board Computers

Add-On Boards

Cables

Cameras

Cases

Connectivity

Connectors

Coolers

Development

Display

Memory

Power & Battery

Sensors

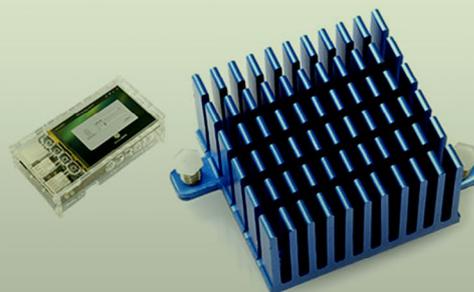
Sound

**¡Visita la  
NUEVA  
tienda  
para  
todas tus  
demandas  
ODROID!**

**odroidinc.com**

## MANY KINDS OF ACCESSORIES

Multiple accessory options gives users the ability to create all-in-one, integrated projects such as: tablets, game consoles, infotainment systems, and embedded systems.



# CONOCIENDO UN ODROIDIAN

MICHEL CATUDAL (@MINOU666)

editado por Rob Roy (@robroy)

*Por favor, háganos un poco sobre ti.*

Tengo 66 años y vivo en White Pigeon Michigan. Tengo una esposa llamada Betty, que trabaja en casa y un hijo de 32 años llamado Pierre, que es el responsable de los envíos en ACTIA Corporation..

Nací en Rochebaucourt, Abitibi, una ciudad del norte de Quebec, a unos 550 kilómetros al noreste de Montreal, donde hace mucho frío en invierno. Con el tiempo nos mudamos al sur cerca de Montreal, me fascinaban los inviernos tan cálidos del sur de Québec. Éramos extremadamente pobres y mis padres tenían 14 hermanos. Nuestra casa tenía muchas ratas, así que mi amor por los gatos se volvió muy fuerte. Más tarde nos mudamos a una casa nueva (sin ratas) a pocos kilómetros de distancia después de que mis hermanas se hicieran mayores, mamá y papá encontraron trabajo en la fábrica textil local.

En 1969, fui a la universidad para convertirme en un experto en tecnología. Las ayudas económicas no eran muy buenas, así que al siguiente año dejé la universidad para trabajar en Montreal. Después de que el ejército canadiense ocupara Montreal, decidí aprender el idioma del enemigo, así que me mudé a Toronto. No hablaba mucho inglés por aquel entonces, así que me fue complicado encontrar trabajo, pero gracias a unos cuantos canadienses franceses locales, encontré trabajo. Nos reuníamos cerca del Ejército de Salvación de Cabbagetown, donde había gente que cogía a trabajadores todas las mañanas para realizar trabajos agrícolas y otros tipos de trabajo. Finalmente regresé a Québec para terminar mi grado de tecnología y luego me convertí en ingeniero.

Cada verano desde 1969 a 1979, trabajaba en granjas en



**Figura 1 - Michel en su oficina en casa con uno de sus gatos, Bella**

Ontario. A menudo era un poco violento debido al odio que nos tenían los Anglos. Para nosotros, Ontario era territorio ajeno y los lugareños eran muy racistas con nosotros, pero los granjeros nos querían porque estábamos dispuestos a hacer cualquier trabajo por muy duro que fuese, sin hacer preguntas. Podía vivir todo un invierno en Québec con los ingresos del verano. Hoy día, los mexicanos son lo que están haciendo ese trabajo tan duro que nosotros haciendo por aquel entonces, y es por ello que les tengo un gran respeto.

Un día, conocí a mi futura esposa en Missouri. Según su madre, ella tiene orígenes Cherokee, Amish e irlandés. Yo ten-



**Figura 2 - Los gatos de Michel son los verdaderos jefazos de la casa**

go parte de Mohawk, Abenakis y francés (Québec, Acadie y Louisiane). Según un investigador de mi familia, tengo algunas raíces inglesas de un esclavo británico que se casó con algún miembro de la familia.

Me gradué en la universidad con un grado en tecnología electrónica e ingeniería eléctrica en Trois-Rivières, Quebec.

Era difícil encontrar trabajo, así que acepté uno en Reed City, Michigan. Mientras estaba en Reed City, hice varios diseños electrónicos desde reconocimiento de voz a sistemas táctiles capacitivos y protección contra ventanas y techos solares. Trabajé en Reed city durante 3 años a principios de los años 80 y 7 en los años 90. Trabajé en Syracuse en 1985 y 1986 en un sistema de correo para Merrill Lynch para el World Trade Center. También diseñé un contador de molino para Acurite después de que la Compañía en Syracuse fuera vendida. Luego me mudé a Buffalo a Cambridge Instruments, sólo para perder ese trabajo más tarde, cuando la empresa fue vendida a Leica. Después de eso, me fui a trabajar en Montreal para un contrato de 3 años, donde diseñé los controles del sistema de iluminación para el nuevo foro de hockey y el casino.

Cuando terminé el contrato, trabajé brevemente para Corexco. Estaba muy disgustado por el hecho de que el salario prometido se redujo drásticamente una vez que firmé para trabajar con ellos. Entonces me fui y volví a Reed City para trabajar en un nuevo diseño de protección para Ford. Cuando mi jefe perdió los casos judiciales contra GM, ST y Amway me dejaron marchar. He estado trabajando en ACTIA Corp en Indiana desde entonces.

Durante los últimos 15 años, he estado haciendo software integrado para la industria del automóvil, marina y militar, desde calibradores inteligentes a clúster. Desde el año pasado, solo desarrollo Linux embebido en placas de



**Figura 3 - El hijo de Michel, Pierre, trabaja en la misma compañía que su padre**

visualización LCD en color con un procesador TI DRA726.

*¿Cómo empezaste con los ordenadores?*

Desarrolle mi primer ordenador (Cosmac) en 1976. Trabajé más tarde con un ordenador ZCPR y con un PC IBM. ZCPR era la versión hacker del CPM/80é.

*¿Qué te atrajo a la plataforma ODROID?*

No sabía nada de los ODROIDS hasta que alguien me pidió que ayudara a Lázaro a trabajar con hardfloat en el ODROID-U2.

*¿Cómo usas tus ODROIDS?*

Mi interés en los ODROIDS viene dado estrictamente por el reto de hacer algo que muy pocas personas pueden hacer y estar familiarizado con Linux embebido. Estoy diseñando algunas placas para el ODROID-C2, como un dispositivo CAN de Microchip. A diferencia de la Beaglebone, que tiene una CAN integrada.

*¿Cuál es tu ODROID favorito y por qué?*

Mi favorito es el XU3, que a la vez es potente y rápido. Incluso sin soporte completo de GPU, la velocidad y la potencia de los núcleos lo compensa. Probablemente me haré con XU4 una vez que termine mi trabajo con ACTIA.

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

Me gustaría ver mejor soporte para GPU en Linux, y una placa con un dispositivo que tenga CAN.

*¿Qué aficiones e intereses tienes aparte de los ordenadores?*

Los ordenadores Linux son mi afición. Recientemente Linux embebido forma parte de mi trabajo. Mi trabajo en Linux embebido es una oportunidad para hacer algo drásti-



**Figura 4 - Michel tiene una impresionante colección de ordenadores de placa reducida**

camente diferente a mi trabajo con Windows en ACTIA Corp.

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

¡Qué no tengas miedo a los retos!