

Originalmente las leyes de copyright se aplicaron a trabajos literarios y artísticos. Hoy también se aplica a programas de computadora porque, así como la literatura y el arte, el código es una forma de expresión. De la misma forma que se puede registrar una pintura del Gran Cañón del Colorado, pero no al mismo Gran Cañón del Colorado, uno puede registrar una implementación de Quicksort en “C”, pero no a la idea de **Quicksort**. Para asegurarse de que el código fuente pueda ser compartido libremente, la comunidad de software creó una licencia de código abierto. Estas licencias iban desde versiones *copyleft* (algo abierto genera algo abierto) como la GNU Public License (**GPL**) a otras más permisivas del estilo “agradeceme, no me demandes, para todo el resto hacé lo que quieras”, como la licencia Berkeley Software Distribution (**BSD**).

Los planos de hardware también se pueden registrar, pero son funcionales, así que definir “hardware abierto” es un poco más complejo. Antes, casi todo el hardware se entregaba con un diagrama esquemático. Sin embargo, en algún punto se volvió imposible para los usuarios hacer mantenimiento al hardware sin romper la garantía. Los dispositivos ahora están llenos de secretos comerciales. Esta tendencia creó una distinción artificial entre hardware cerrado y abierto. Digo “artificial” porque mientras el software puede ser cifrado de manera tan fuerte que probablemente se necesite una computadora del tamaño del planeta para poder romper el cifrado, uno puede relevar cualquier diseño de hardware y armar el esquemático, siempre y cuando se tenga un microscopio lo suficientemente poderoso y un programa para pegar y procesar las imágenes.

Internet está plagada de intentos bien intencionados pero equivocados, pues pretenden aplicar una licencia de copyright abierta de software al hardware. Usar una licencia de software en un pedazo de hardware es como llenar un formulario de casamiento para fusionar dos empresas, aunque el formulario expresa la intención de los autores, puede que no sea aplicable. Por ejemplo, el texto de la GPL no utiliza la palabra hardware ni una sola vez, y esto puede producir que un tribunal falle que la GPL no aplica legalmente al hardware.

Se han creado algunas licencias abiertas específicas para hardware a fin de paliar esta situación (el **CERN OHL** es una licencia decente para hardware estilo *copyleft*), pero la comunidad está dividida sobre cuánto del proceso de creación debería estar abierto para considerar que una pieza de hardware es abierta. Por ejemplo, si comparto los esquemáticos para un placa que diseñé con una herramienta de código cerrado, muchos me dirían que el diseño no es de código abierto. Es más, supongamos que diseñé la placa usando una herramienta de captura que es software libre y abierto (**F/OSS**), ¿Qué pasaría con el diseño de los chips de silicio o los pedacitos de firmware que están impresos en el silicio? ¿Necesitamos también ver los planos de los aceleradores de partículas usados para dopar el silicio? ¿Y la máquina que se usa para grabar las máscaras para la producción del silicio? **Abajo hay tortugas y más tortugas**. El hardware no puede ser de código abierto puro, porque en algún punto las ideas se transforman en sustancia, y acceder a los objetos que transforman y moldean la materia no es algo que esté disponible para todos.

Hay, sin embargo, posiciones mucho más pragmáticas frente al hardware abierto que frente a la microscopía electrónica o exigir fundiciones de silicio abiertas. Simplemente compartir planos en algún nivel de abstracción toma menos esfuerzo, es más intuitivo y aun así tiene un efecto positivo. El estilo de código abierto del mercado gris de shanzhai, que nombré antes como el *gongkai*, recoge los beneficios de esta forma de compartir. En China los planos se comparten públicamente, bajo condiciones poco claras. La mayoría de los diseños aún utilizan sellos de propiedad como “confidencial” o “privado”, y el shanzhai usa copias pirateadas de programas profesionales de código cerrado para el diseño de sus trabajos que son modificaciones de los originales. Sin embargo, al fin y al cabo, esta forma laxa de apertura crea un ecosistema donde cientos de pequeñas compañías se ganan la vida reparando y construyendo celulares. Caminar en los mercados de *Shenzhen* me permitió darme cuenta de que construir un celular no es ni difícil, ni da miedo. Las comunidades por fuera de shanzhai no se sienten con el derecho de mirar adentro de la caja opaca, debido a las leyes de propiedad intelectual que resultan tan restrictivas.

El ecosistema *gongkai*, que exploré en la parte 2, valora la propiedad intelectual y física de manera similar. Los planos de los circuitos sin una cadena de proveedores son inútiles: no podés llamar por celular con los planos de un teléfono. De la misma forma, los fabricantes de chips no hacen negocios si no hay productos que usen sus chips. El resultado es que los creadores de hardware tienen un aliciente natural para compartir información, particularmente la necesaria para diseñar un sistema más grande en base a un chip o a un módulo. Hacer que el cliente adopte un diseño específico para ese chip garantiza que comprará el mismo chip cuando estén listos para salir a producir a gran escala. Este balance entre propiedad intelectual y la cadena de proveedores ha sido difícil de alcanzar en el ecosistema occidental orientado a la propiedad Intelectual, donde las ideas tienen más valor que las fábricas. Esto puede explicar por qué tantos puestos de trabajo en manufactura han emigrado a China, un ecosistema que valora de manera similar la producción y las ideas detrás de los productos.

Confío en que con un esfuerzo consistente, una creciente concientización pública y las condiciones económicas adecuadas, eventualmente el ecosistema de hardware mundial generará una fundición de silicio abierta. Hasta que esto suceda el “hardware abierto” tiene que ser un concepto más pragmático que está limitado a vivir dentro de ciertos niveles de abstracción. Después de todo, solo compartir planos (aun cuando las licencias no sean perfectas y los formatos no sean fácilmente modificables) afecta la innovación de manera dramática. El shanzhai es una prueba viviente de esto.

No importa si es *gongkai* o código abierto. El hardware abierto no es una disputa legal, es empoderar a los usuarios y que ellos sean expertos de su propia tecnología. La libertad de aprender, modificar y mejorar la tecnología está tan enraizada en mi persona que yo la veo como un derecho humano básico. La libertad se atrofia si no se ejercita, y esta es la razón por la cual la defiendo activamente. Yo comparto mi trabajo de manera abierta, con la esperanza de empoderar a otros y hacer saber a todos que se pueden apropiarse de la

tecnología. No somos esclavos de nuestras computadoras o de las corporaciones que las fabrican.

Desafío también a los intentos legislativos y legales que buscan restringir nuestras libertades. Nací en un mundo libre de DMCA, y me gustaría abandonarlo en el mismo estado, logrando establecer que todos tengan el derecho a comprender, reparar y modificar las cosas que poseen. Esto es hoy más importante que nunca, ya que cada vez somos más dependientes de la tecnología. Si permitimos que la tecnología se convierta en una caja negra, someteremos nuestra capacidad de acción a las compañías y gobiernos que las producen y regulan.

Esta parte del libro describe cómo construí tres plataformas de hardware abierto: **Chumby**, **Novena** y **Chibitronics**. Espero que la lectura de mis relatos les haga comprender que se puede entender el hardware y que este conocimiento nos empodera.

Traducción de Nicolás Wolovick

Permiso de uso

bunnie <bunnie@bunniestudios.com>

Jun 29 (1 day ago) ☆

to Nicolás

hi Nicolas,

I think that should be fine. thanks for asking.

thanks,

-b.

--

...

On June 29, 2017 3:22:45 AM GMT+08:00, "Nicolás Wolovick" <nicolasw@famaf.unc.edu.ar> wrote:

Hi. I wonder if it is OK to translate to spanish and distribute internally the Part 4 introduction of "The Hardware Hacker" book. The target is a group of ~50 elementary and high-school teachers doing a [postgraduate course on "Teaching Programming"](#). I wanted some text on open source software and I found an article by RMS about why schools should use F/OSS. And for open hardware I've recently read your latest book and the intro of Part 3 is perfect.

TIA and saludos, Nico