

Experiencia de migración hacia herramienta de software libre para el diseño de circuitos impresos

Tropea, S.⁽¹⁾; Brengi, D.⁽¹⁾; Borgna, J. P.⁽¹⁾

⁽¹⁾INTI-Electrónica e Informática

⁽¹⁾Ex Becario INTI-Electrónica e Informática

Introducción

En este trabajo se presenta el proceso de migración hacia la herramienta de diseño de circuitos impresos de software libre llamada KICAD. Se expondrán los motivos por los cuales se ha realizado el cambio, el plan de acción realizado, las adaptaciones realizadas al software y las herramientas complementarias desarrolladas para cubrir las necesidades particulares del laboratorio.

En todo laboratorio electrónico de desarrollo es necesario diseñar circuitos impresos, también llamados PCBs (*Printed Circuit Boards*). Esta tarea se compone de dos etapas principales. La primera es el ingreso a la PC de un circuito esquemático: diagrama eléctrico conceptual que define las conexiones entre los componentes que forman el circuito. En una etapa siguiente se realiza el diseño físico del circuito impreso en base a las conexiones definidas en el esquema anterior.

Existen en el mercado numerosos paquetes de software que cumplen con los requisitos mínimos para realizar esta tarea. Sin embargo ninguno de los utilizados hasta el momento dentro del laboratorio cumplía totalmente con nuestras expectativas:

- Libre uso, copia y redistribución.
- Disponibilidad del código fuente del programa.
- Opción para entornos GNU/Linux.

Identificando ventajas estratégicas en la utilización de herramientas de software libre, y luego de un proceso de investigación y evaluación, se decidió realizar una migración en todo el laboratorio de la herramienta utilizada hasta ese momento para el diseño de circuitos impresos.

Metodología

Dividiremos la metodología de trabajo en cuatro etapas principales: Búsqueda y selección del software más adecuado; evaluación de las capacidades de la herramienta; proceso de migración dentro del laboratorio y herramientas desarrolladas para facilitar nuestra labor.

Búsqueda y selección

Al momento de seleccionar una herramienta de trabajo es importante evaluar, además del costo y las prestaciones, la posibilidad de adaptación y apropiación de una tecnología en particular. Esto se torna de mayor importancia si se pretende promover y fomentar la actividad regional, y especialmente para quien brinda servicios de desarrollo y transferencia tecnológica, ya que los archivos de diseño necesitan del software para ser modificados y adaptados en el futuro.

Por este motivo se consideran altamente beneficiosas todas aquellas herramientas de software que trabajen con estándares abiertos y preferentemente bajo licencias de software abierto o libre ^[1] que permitan el libre uso, modificación y adaptación del programa.

Hace varios años que evaluamos las opciones basadas en software libre para el diseño de circuitos electrónicos. Entre ellas podemos mencionar las siguientes:

- Ingreso de circuitos esquemáticos: Geda Gschem^[2], Xcircuit^[3].
- Ruteo de circuitos impresos: PCB^[4], Freepcb^[5], MUCS-PCB^[6].
- Visualización de archivos Gerber: gervb ^[7].

Estas herramientas han estado disponibles por mucho tiempo para el diseño de circuitos impresos bajo software libre. Sin embargo no cumplían con todos los requerimientos deseados, debido en parte a la escasa

integración entre estos tres diferentes tipos de programas para realizar una sola tarea, por no poseer una interfaz unificada o por la escasa utilización por parte de los usuarios finales.

A mediados del año 2005 y gracias a internet, tomamos conocimiento de la existencia de un software recientemente publicado llamado KICAD para el diseño de circuitos impresos. Esta herramienta poseía varias características atractivas:

—Código fuente disponible y bajo licencia GPL^[8] de software libre que permite el libre uso, redistribución y adaptación del programa.

—Interfaz gráfica unificada y sencilla que permite el ingreso de circuitos esquemáticos, diagramación y ruteo del circuito impreso y manejo de archivos *Gerber*.

—Soporte de plataformas GNU/Linux y MS Windows (actualmente incorpora también Mac OSX).

—Ejemplo de diseño de alta densidad y 4 capas. Se trata de una tarjeta PCI para manejo de señales de video, realizada por el autor del programa.

—Sin costo monetario. El programa y el código fuente se obtienen gratuitamente de Internet.

—Visualización 3D del diseño.

Evaluación

Para evaluar las capacidades del software KICAD, y considerar su adopción dentro del laboratorio para el trabajo cotidiano, se decidió abordar con el mismo un trabajo de complejidad media y realizar todo el proceso desde el diseño hasta el armado físico del circuito. Estas son las características del circuito resultante de la experiencia:

—Circuito impreso doble faz.

—Montaje superficial con encapsulados PQ208, SOIC20, SSOP, etc.

—Área aproximada de 12x10 cm.

—Pistas de 10 mils.

—Espaciado mínimo de 8 mils entre islas o pistas.

Este diseño fue realizado exitosamente. Finalmente se generaron con el programa los archivos *Gerber* que luego se enviaron a la empresa que fabrica a pedido los circuitos impresos. Luego de recibir las plaquetas, se procedió con el montaje de los componentes y la

prueba del circuito sin registrarse problemas de importancia (ver Fig. 1).

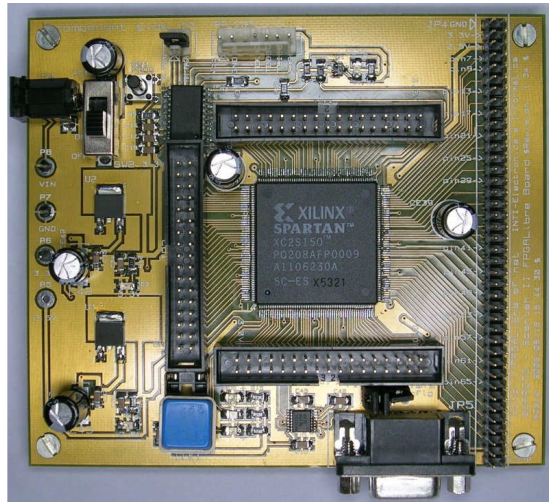


Figura 1: Vista superior del circuito impreso armado y probado

Proceso de migración

Luego del éxito de la experiencia piloto. Se decidió utilizar KICAD para todo trabajo futuro. Esta decisión involucra varios aspectos: Instalación en los puestos de trabajo, capacitación del personal, creación de componentes de librería para los nuevos diseños y considerar los circuitos desarrollados con herramientas previas a KICAD. Veremos estos puntos con más detalle.

Instalación: Dentro del laboratorio todos los puestos de trabajo utilizan exclusivamente el sistema operativo Debian GNU/Linux^[9]. Hacia el año 2005 todavía no existía el paquete de software KICAD para sistemas Debian, motivo por el cual se armó el paquete correspondiente para facilitar la instalación, desinstalación y administración de KICAD en todos los puestos de trabajo. Este paquete se ofreció al público en general a través de la web del proyecto FPGALibre^{[10][11]}, contando hasta el momento con más de 1000 descargas en total. Hoy en día este paquete de software adaptado por nuestro laboratorio específicamente para Debian ha quedado obsoleto ya que Debian ofrece oficialmente KICAD en su distribución.

Capacitación: Pasar de un software a otro generalmente involucra un período de capacitación o aprendizaje por parte de los usuarios. Esto genera normalmente una pérdida de tiempo en las etapas iniciales de migración pero redundará en un beneficio a mediano y largo

plazo. Quienes participaron en la primer experiencia piloto brindaron una capacitación y soporte a los demás miembros del laboratorio. Esto se realizó simplemente compartiendo dentro del grupo los conceptos aprendidos y destacando las principales diferencias con respecto al software anteriormente utilizado.

Componentes de librería: Todos los programas para diseño de circuitos impresos poseen bibliotecas de dibujos esquemáticos (símbolos) y bibliotecas de encapsulados de componentes (*footprints* o módulos), permitiendo además incorporar nuevas bibliotecas (*libraries*) particulares o personalizadas. Al comenzar a utilizar un software para diseñar circuitos impresos siempre será necesario crear algunos nuevos componentes de librería adicionales.

Existen algunos *scripts* o programas realizados por los usuarios de KICAD para convertir bibliotecas o componentes desde otros formatos (como Orcad o Eagle) al formato de KICAD. No se han evaluado estas herramientas.

Por otra parte KICAD posee una gran variedad de componentes y actualmente los usuarios contribuyen rápidamente nuevas bibliotecas. En el caso de no existir el componente, KICAD posee un editor de símbolos y módulos para ingresarlo.

Circuitos previos: Uno de los problemas que puede presentarse en el proceso de migración es la necesidad de utilizar o consultar circuitos previamente desarrollados. Para estos casos puede utilizarse la antigua herramienta y generar archivos de documentación que no requieran nuevamente del programa, como por ejemplo formatos *Postscript* o *pdf* para esquemáticos o circuitos impresos y archivos de texto para el listado de componentes.

Otra situación más complicada es cuando se desea reutilizar un viejo diseño y modificarlo para realizar uno nuevo. Como se mencionó, existen algunos programas para convertir otros formatos de componentes a formato KICAD, pero esto resuelve sólo una mínima parte del problema.

En estos casos son pocas las opciones. Nuestra política adoptada es volver a ingresar el viejo diseño copiando al anterior.

Herramientas Desarrolladas

El software KICAD está compuesto oficialmente por varias partes:

—Administrador de proyectos: Permite manejar todos los archivos involucrados en forma centralizada y recorrer organizadamente las etapas de un diseño.

—Eeschema: El editor de esquemáticos para ingresar nuestro diseño.

—Cvpcb: El editor de asociaciones para vincular los símbolos del esquemático con los módulos o *footprints*.

—Pcbnew: El editor de PCBs para realizar el "*place and route*" del impreso. Permite la visualización en 3D del circuito impreso poblado con los componentes electrónicos (*ver Fig. 2*).

—Gerbview: Visualizador de *Gerbers* para verificar visualmente nuestro diseño antes de enviarlo a los servicios de confección.

—Editor de librería: permite crear y modificar los símbolos y los módulos para el diseño.

—Documentación: Instructivo de uso en varios idiomas, incluido el español.

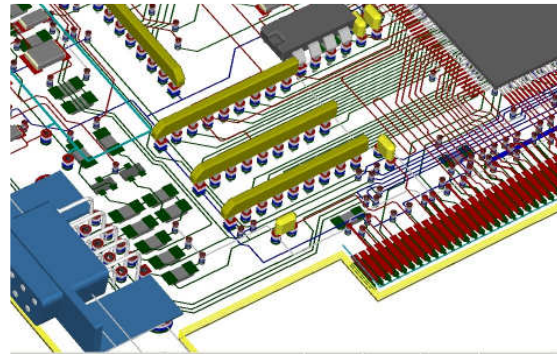


Figura 2: Visualización 3D de la herramienta KICAD.

Por tratarse de un software con el código disponible y que utiliza formatos abiertos y documentados, muchos programadores han desarrollado herramientas adicionales que facilitan las tareas asociadas al diseño. Podemos mencionar rutinas de software que generan automáticamente componentes *DIP Dual Inline Package*, páginas web que permiten diseñar fácilmente símbolos estándar para esquemáticos, conversores para exportar e importar entre formatos de otros programas de diseño, etc.

Para facilitar nuestra tarea en la utilización del software de diseño y acelerar los tiempos necesarios para realizar un trabajo determinado, desarrollamos dos programas complementarios al software KICAD:

—**Kicadlib:** Permite administrar las bibliotecas de KICAD, quitando, agregando y listando los componentes. Este programa se desarrolló en lenguaje C y actualmente se encuentra publicado en internet disponible para cualquiera que desee utilizarlo.

—**Pinarray:** Permite crear símbolos y módulos PCB para arreglos de pines. Se utiliza para definir fácilmente conectores y áreas de prototipado. Este programa se desarrolló en lenguaje Perl y también se publicó a través de internet.

Estas dos herramientas desarrolladas facilitan la creación y administración de componentes de librería y por lo tanto reducen los tiempos involucrados en los nuevos desarrollos de hardware.

Resultados

Se realizó una experiencia piloto exitosa en la utilización del software KICAD para el desarrollo de circuitos impresos prototipos, recorriendo todo el camino desde el diseño hasta la construcción física y prueba de un circuito impreso doble faz. Apoyados en esta experiencia se realizó una migración completa de todos los puestos de trabajo acompañado por un proceso de capacitación interna para abordar los nuevos desarrollos utilizando el nuevo software. Además, gracias a las especificaciones abiertas, se pudieron desarrollar herramientas de software complementarias que reducen los tiempos de desarrollo involucrados en el diseño de circuitos impresos.

Conclusiones

Trabajando con este nuevo programa de software libre se tiene actualmente las siguientes ventajas:

—La posibilidad de inspección y modificación del código fuente para adaptar el software a futuras necesidades.

—Formato interno de los archivos bien documentado. Esto permite crear nuevas herramientas de software.

—Se permite redistribuir el software. Esto es de mucha utilidad al realizar un diseño para un cliente, ya que se le puede entregar el diseño y el software necesario para utilizar y adaptar su diseño en el futuro.

—Se realizó un importante ahorro en el costo de licencias de software.

Basados en esta experiencia de migración y otras similares realizadas dentro del laboratorio, y observando también otros grupos de trabajo bajo los mismos conceptos, consideramos de gran importancia que las empresas e instituciones dedicadas al desarrollo electrónico tomen conciencia de las ventajas que aporta el software libre y las tecnologías abiertas en general. Creemos que estas ventajas son de vital importancia para asignar adecuadamente los recursos para desarrollo cuando los mismos son limitados y para una verdadera evolución tecnológica regional con empresas de nuestro país. Existe software libre de alta calidad técnica, para muchas ramas específicas de la ingeniería y la ciencia en general. Identificar, utilizar, aprender y adaptar el software libre existente es un esfuerzo extra que seguramente brindará sus beneficios.

Referencias

- [1] Free Software Foundation, Inc. "The Free Software Definition". <http://www.gnu.org/philosophy/free-sw.html>.
- [2] Ales Hvezda. "gEDA Schematic Capture". <http://www.geda.seul.org/tools/gschem/>.
- [3] Timothy Edwards. "XCircuit : Schematic drawing and capture program". <http://opencircuitdesign.com/xcircuit/>.
- [4] Thomas Nau, Harry Eaton, Bill Wilson, Dan McMahill, and DJ Delorie. "PCB, an interactive printed circuit board editor". <http://pcb.sourceforge.net/>.
- [5] Allan Wright. "FreePCB an open-source PCB layout editor for Windows". <http://www.freepcb.com/>.
- [6] The University of Manchester, School Of Computer Science. "MUCS-PCB design software". <http://intranet.cs.man.ac.uk/apt/projects/tools/mucs-pcb/>.
- [7] Stefan Petersen et al. "gerbv - A Free Gerber Viewer". <http://gerbv.sourceforge.net/>.
- [8] Free Software Foundation, Inc. "GNU General Public License". <http://www.gnu.org/copyleft/gpl.html>.
- [9] Debian. "Sistema operativo Debian GNU/Linux". <http://www.debian.org>.
- [10] Salvador Eduardo Tropea, Diego Javier Brengi, and Juan Pablo Daniel Borgna. "FPGAlibre: Herramientas de software libre para diseño con FPGAs". In *FPGA Based Systems*, pages 173–180, Mar del Plata, 2006. Surlabs Project, II SPL.
- [11] INTI Electrónica e Informática et al. Proyecto FPGA Libre. <http://fpgalibre.sourceforge.net/>.

Para mayor información contactarse con:

Diego J. Brengi - brenji@inti.gov.ar