

Generación automática de archivos de fabricación en KiCad

Salvador E. Tropea
Centro de Micro y Nano Tecnologías
Instituto Nacional de Tecnología Industrial
Buenos Aires, Argentina
stropea@inti.gob.ar

Resumen—KiCad es probablemente la herramienta de desarrollo de circuitos impresos más avanzada en el mundo del software libre.

En este trabajo presentamos un conjunto de herramientas capaces de automatizar la verificación de reglas de diseño y la generación de los archivos de fabricación cuando usamos KiCad.

Estas herramientas son adecuadas para ser usadas localmente o en entornos de integración continua.

Index Terms—KiCad, automatización, fabricación, integración continua

I. INTRODUCCIÓN

A la hora de elegir una herramienta de desarrollo de circuitos impresos que sea de software libre KiCad [1] es probablemente la mejor elección. Esta herramienta ha avanzado mucho en los últimos años, con el apoyo de prestigiosas instituciones como el CERN [2].

Una vez desarrollado el circuito se debe generar toda la documentación necesaria para su fabricación. Esta documentación incluye una amplia variedad de documentos, tales como el listado de materiales, los gerbers [3], archivos de taladrado, archivos de posicionamiento de componentes, etc. Todos acompañados por archivos PDF con el esquemático y PCB. Siendo los gerbers el formato estándar de fabricación de PCBs. La generación de estos archivos es una tarea bastante tediosa ya que es necesario generar cada uno de estos documentos usando opciones diferentes de herramientas diferentes. Esto es peligroso ya que se podrían cometer errores en alguno de los pasos.

Lamentablemente no hay una receta estandarizada que le permita a la herramienta generar esta documentación en una única operación. Esto es así porque cada fabricante, y cada desarrollador, tiene sus propias preferencias. Pero lo que sí sería posible es volcar dichas preferencias en un único archivo y que luego la herramienta se encargara de generar la documentación de acuerdo con esta especificación.

En este trabajo se presenta un conjunto de herramientas capaces de realizar la tarea antes mencionada. Estas herramientas fueron desarrolladas tomando ideas y componentes presentados en la KiCon 2019 [4], junto con herramientas ya usadas por nuestro equipo de trabajo.

II. HERRAMIENTAS

II-A. Núcleo central

El núcleo de este conjunto de herramientas es un derivado de KiPlot [5] [6]. Esta herramienta permite tomar un archivo de configuración en formato YAML [7] y generar en forma automática la documentación especificada. Todas las opciones quedan así concentradas en un único archivo de fácil interpretación.

Originalmente KiPlot se limita a generar los archivos que pueden generarse a través de la interfaz Python de KiCad. Esto incluye los gerbers y todos los formatos disponibles en la opción `plot` del menú y los archivos de taladrado. Pero esto es sólo una parte de los archivos necesarios.

Por esta razón se extendió la aplicación para soportar otros formatos. Por un lado se agregaron los archivos de posicionamiento de componentes y el archivo de integración de los gerbers a la misma herramienta. Por otro lado se agregó la posibilidad de integrar otras tres aplicaciones para complementarla.

La Fig. 1 muestra el flujo de trabajo. En la misma se aprecia como KiPlot tiene un rol central.

II-B. Listado de materiales

Esta compuesto por todos los componentes pasivos, integrados, conectores, etc. presentes en el esquemático. KiCad utiliza un mecanismo de plug-ins para esta tarea, por lo que es relativamente simple utilizar herramientas externas. Para ello se seleccionaron dos aplicaciones muy útiles.

KiBoM [8] [9] es un script que permite generar un listado de materiales ordenado y completo. Se contribuyó a este proyecto varios detalles interesantes, tales como poder reconocer campos con el código de parte en Digi-Key [10] y convertirlos en links al sitio, o generar un listado separado de componentes que son opcionales. Luego se integró con KiPlot.

InteractiveHtmlBom [11] [12] es otro script interesante. El mismo permite generar una página web interactiva donde se puede ver la posición de cada componente en el PCB e ir marcando cuáles ya han sido soldados. También fue integrada a KiPlot.

II-C. Automatización de la interacción con el usuario

KiCad no ofrece mecanismos automáticos para generar las versiones PDF de documentación del esquemático ni del PCB.

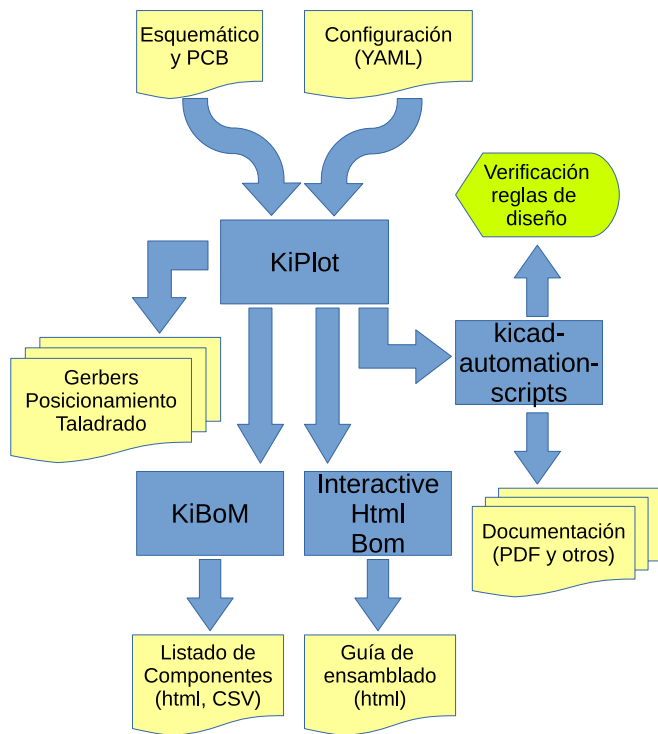


Figura 1. Flujo de datos.

Si bien es posible generar desde Python archivos PDF, estos son más bien unos gerbers en formato PDF, y no los archivos de documentación.

Adicionalmente es imposible automatizar las tareas de verificar las reglas de diseño o actualizar el netlist necesario para KiBoM e InteractiveHtmlBom.

Para automatizar estas tareas es necesario simular las acciones que haría una persona con las herramientas. Es decir que es necesario sintetizar la interacción con la interfaz de usuario. En la KiCon 2019 [13] se presentó un esfuerzo por realizar estas tareas con una herramienta denominada kicad-automation-scripts [14] [15]. La misma estaba incompleta y sólo orientada a entornos de integración continua, siendo complicado usarla de manera local.

Se mejoró y complementó esta herramienta, para luego integrarla a KiPlot. En la actualidad es posible realizar todas las tareas antes mencionadas especificándolas en el archivo de configuración de KiPlot.

II-D. Integración continua

Los entornos de integración continua incluidos en los repositorios populares como GitHub y GitLab permiten realizar tareas automáticas cada vez que un cambio es subido al repositorio. Este tema también fue cubierto por otra charla de la KiCon 2019 [16] y es impulsado por herramientas comerciales [17].

Si bien el objetivo de estas herramientas es generar la documentación final para la fabricación, existen varias razones que hacen su uso atractivo en estos entornos, entre ellas:

- La posibilidad de correr los chequeos de las reglas de diseño.
- La generación de documentación parcial durante el desarrollo. De manera tal que otros equipos involucrados o el mismo cliente puedan acceder a documentación temprana.
- La documentación y verificación de las revisiones posteriores a la fabricación del primer prototipo.

Por estas razones se decidió generar imágenes docker conteniendo KiCad [18] [19] y las herramientas producto de este trabajo [20] [21]. Las mismas se encuentran disponibles en Docker Hub.

Al mismo tiempo, esto permite disponer de entornos completos y controlados en los cuales correr las aplicaciones.

III. EJEMPLOS DE APLICACIÓN

Se crearon tres ejemplos de aplicación. Dos muy básicos, cada uno como ejemplo de uso del entorno GitHub [22] y GitLab [23]. El tercero es un poco más elaborado ya que se trata de un proyecto real compuesto por tres PCBs [24] (Spora [25]). Adicionalmente en este último ejemplo se implementó un sistema de releases automáticas, al realizar un tag en el repositorio se corren las verificaciones de las reglas de diseño, se genera la documentación y se empaqueta todo en una release.

En el ejemplo que usó como base el proyecto Spora se puede observar como las herramientas reportaron errores en las reglas de diseño que luego fueron subsanados, ver la Fig. 2.

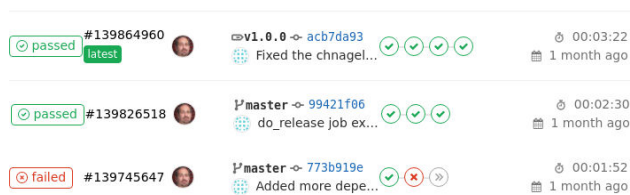


Figura 2. CI/CD pipelines. Arriba ejemplo de release automática, en medio verificación exitosa y abajo falla en las verificaciones

IV. CONCLUSIONES

La integración con los sistemas de CI/CD de GitHub y GitLab mostró ser simple. Permitiendo resultados consistentes y automáticos, aplicados a proyectos reales. Cualquier cambio que viole las reglas de diseño es informado por correo electrónico y es posible ver el estado de cumplimiento en la página del proyecto.

Se logró la disponibilidad de varios documentos preliminares generados en forma automática con cada cambio introducido. Esto permite que otros grupos del equipo de desarrollo accedan a información actualizada sin interferir con las tareas de desarrollo de hardware.

A futuro se piensa agregar la generación de modelos 3D del proyecto y completar la testeabilidad de los programas usados. Así como también simplificar la configuración.

REFERENCIAS

- [1] J.-P. Charras *et al.*, “KiCad EDA: A cross platform and open source electronics design automation suite;” <http://www.kicad-pcb.org/>.
- [2] (2020, May) About KiCad. [Online]. Available: <https://kicad-pcb.org/about/kicad/>
- [3] Wikipedia. (2020, Jul.) Gerber (formato de archivo). [Online]. Available: [https://es.wikipedia.org/wiki/Gerber_\(formato_de_archivo\)](https://es.wikipedia.org/wiki/Gerber_(formato_de_archivo))
- [4] (2020, May) KiCon 2019 the first and the largest gathering of hardware developer using KiCad. [Online]. Available: <http://kicad-kicon.com/>
- [5] J. J. Beard. (2020, May) KiPlot kicad pcbs plotter. [Online]. Available: <https://github.com/johnbeard/kiplot>
- [6] J. J. Beard and S. E. Tropea. (2020, May) KiPlot kicad pcbs plotter (fork). [Online]. Available: <https://github.com/INTI-CMNB/kiplot>
- [7] (2020, May) YAML ain't markup language. [Online]. Available: <https://yaml.org/>
- [8] (2020, May) KiBoM: Configurable BoM generation tool for KiCad EDA. [Online]. Available: <https://github.com/SchrodingersGat/KiBoM>
- [9] (2020, May) KiBoM: Configurable BoM generation tool for KiCad EDA (fork). [Online]. Available: <https://github.com/INTI-CMNB/KiBoM>
- [10] (2020, May) Digi-key distribuidor de componentes electrónicos. [Online]. Available: <https://www.digikey.com/es>
- [11] (2020, May) Interactive HTML BOM plugin for KiCad. [Online]. Available: <https://github.com/openscopeproject/InteractiveHtmlBom>
- [12] (2020, May) Interactive HTML BOM plugin for KiCad. [Online]. Available: <https://github.com/INTI-CMNB/InteractiveHtmlBom>
- [13] J. Vincent. (2020, May) Automating fab. outputs with kicad and git. [Online]. Available: <https://youtu.be/NZCyk3rmmGQ>
- [14] S. Stas and J. Vincent. (2020, May) Kicad automation scripts: A bunch of scripts to automate kicad processes. [Online]. Available: <https://github.com/obra/kicad-automation-scripts>
- [15] J. V. Seppe Stas and S. E. Tropea. (2020, May) Kicad automation scripts: A bunch of scripts to automate kicad processes (fork). [Online]. Available: <https://github.com/INTI-CMNB/kicad-automation-scripts>
- [16] A. Shmakov. (2020, May) Pcbops: Applying ci/cd to pcb design projects. [Online]. Available: <https://youtu.be/cQ-iFtBBwFc>
- [17] A. Mahpour. (2020, May) Continuous integration and deployment in ecad. [Online]. Available: <https://resources.altium.com/p/continuous-integration-and-deployment-in-ecad>
- [18] S. E. Tropea. (2020, May) Docker image for KiCad on Debian. [Online]. Available: https://hub.docker.com/repository/docker/setsoft/kicad_debian
- [19] ——. (2020, May) Docker image for KiCad on Debian. [Online]. Available: https://github.com/INTI-CMNB/kicad_debian/
- [20] ——. (2020, May) Docker image for KiCad automation. [Online]. Available: https://hub.docker.com/repository/docker/setsoft/kicad_auto
- [21] ——. (2020, May) Docker image for KiCad automation. [Online]. Available: https://github.com/INTI-CMNB/kicad_auto/
- [22] ——. (2020, May) Test for continuous integration using kicad (github). [Online]. Available: https://github.com/INTI-CMNB/kicad_ci_test
- [23] ——. (2020, May) Test for continuous integration using kicad (gitlab). [Online]. Available: <https://gitlab.com/set-soft/kicad-ci-test>
- [24] ——. (2020, May) Test for continuous integration using kicad applied to spora. [Online]. Available: <https://gitlab.com/set-soft/kicad-ci-test-spora>
- [25] (2020, May) Spora, small wearable open hardware design. [Online]. Available: <https://github.com/INTI-CMNB/spora>