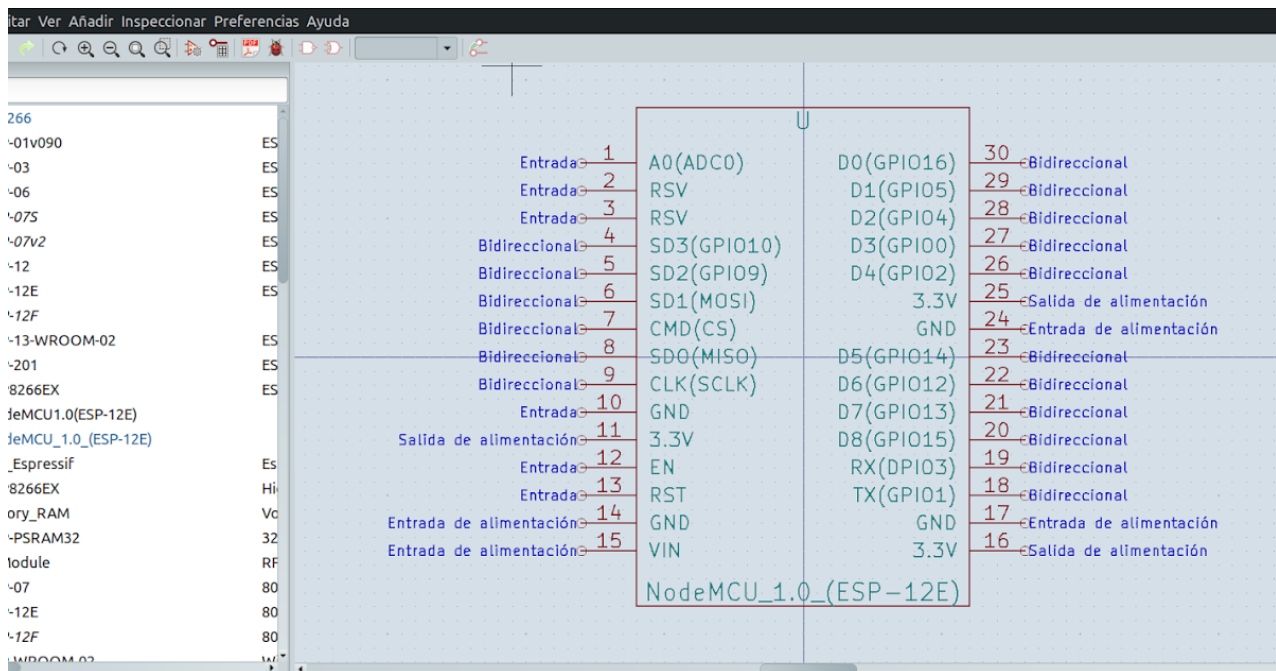


NodeMCU 12_Desarrollo de placas para el Botón del pánico: versión 01. FASE 1

agrpportfolioeducativo.blogspot.com/2020/04/nodemcu-12desarrollo-de-placas-para-el.html



Inicio

Proyecto en github: https://github.com/agrgal/Bot-n_del_p-nico_NODEMCU

Una vez llevado a cabo los programas y testeados los montajes en protoboard, habría que sacar una versión más robusta de los prototipos. Esto pasaría por soldar los componentes a una placa, que previamente, hay que diseñar.

El diseño del circuito impreso se lleva a cabo con el programa kicad. Hacía tiempo que no lo manejaba, así que la primera labor ha consistido en actualizarlo a la versión 5.1.5

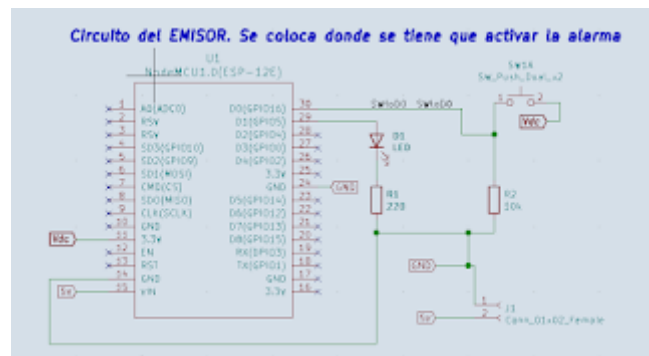
=====

Esquemático

El proceso empieza por diseñar el esquemático. Para ello, descargué la librería de símbolos ([kicad-ESP8266 - a library for KiCAD](#)) que contiene la biblioteca de símbolos ESP8266.lib para el esquemático y los footprints en la carpeta **pretty** correspondiente.

Empezamos el esquemático de la CENTRAL y de la ESTACIÓN. Es el primer paso. Hay que comprobar que no tienen errores ejecutando ERC (control de reglas eléctricas).

=====



Asignación de footprints

Una vez realizados sin errores, continuamos asignando **footprints** a cada componente. Un **footprint** es el lugar de la placa donde irá soldado. Por ejemplo, una resistencia consistirá en dos contactos separados cierta distancia. Ese dibujo de "huella" en la placa está sistematizado y hay bibliotecas de las mismas; pero claro, tenemos que relacionar primero nuestro esquema con los footprints deseados:

Para el circuito CENTRAL:

Para el circuito EMISOR:

En algunos casos, como el conector de 4 pines para la pantalla OLED o el conector de 2 para el Buzzer se escogen componentes genéricos de conexión porque esos elementos no van a ir soldados en placa. Tampoco probablemente en el diseño definitivo el LED y el pulsador, pero esos componentes tienen claramente un footprint asignado.

Antes de este proceso, la huella del componente NodeMCU V1.0 ESP-12E no está cargada; de la biblioteca correspondiente al mismo en el directorio pretty descargado se incorpora en el editor de huellas (footprint editor).

=====

Fichero netlists

Cuando todos estos procesos estén finalizados se genera un fichero netlist (extensión net) que contiene la información de las conexiones que se han creado. Este fichero formará parte de tu proyecto y conectará con un programa diferente de kicad llamado PCB Layout editor

=====

PCB Layout editor

En esta parte del programa se empieza el verdadero diseño del circuito impreso. Lo primero que tenemos que hacer es especificar las características de la placa (ARCHIVO>>BOARD SETUP).

Yo he elegido dos capas de cobre, e indicando que las pistas de cobre estarían en la parte de atrás (capa B.Cu) y la capa de delante "de cobre" será para hacer puentes soldando algunos cables si es necesario (capa F.Cu) ya que pienso aprovechar placas de baquelita de cobre de una sola capa. Añado en este paso también , aparte del diseño de vías y pistas por defecto, algunas más en **Tracks & Vias**.

Básicamente las pistas y vías por defecto son pequeños. Prefiero tener más zona de contacto, pistas más amplias siempre que pueda permitírmelo según los componentes. Usaré pistas más anchas y vías de 1,5mm.

= = = = =

Colocación de componentes

Realizado este paso, se carga el netlist "Load Netlist" accediendo al fichero que creamos anteriormente. Elige la zona del área de trabajo donde colocarás los componentes y haz click.

Haz click en cada componente. Muévelos pulsando "M" y rótalos con "R". Haz una distribución de los mismos.

Las conexiones entre componentes aparecerán como líneas blancas. Es el denominado "rastnet" o red de rastros. El objetivo será conectar todas las conexiones haciendo aparecer pistas y eliminando todos los rastros.

= = = = =

Ruteo

Para dibujar las pistas uso un procedimiento manual. Al parecer, kicad ha prescindido de sus procedimientos de autorouting. Aunque todavía puedes hacerlo con programas gratuitos. Por ejemplo, freeRouting.jar. A mí no me ha hecho falta.

Posteriormente debes trazar el contorno con la herramienta polígono de la placa, eligiendo la capa (paso 1) **Edge.Cuts** Ese dibujo delimitará la zona de impresión de pistas.

= = = = =

Zonas

Es una buena práctica hacer zonas dentro de la placa. Esto evita tener que retirar demasiado material y, si lo hacemos por ejemplo de tensiones fijas a cero voltios (tierra) podremos minimizar las interferencias.

Bajo el símbolo de vías, encontramos el de zonas. Antes de dibujar el polígono de la zona (ver imagen) aparecerá un menú de selección en el que lo más importante es elegir la capa donde se dibujará la zona, la pista que servirá de relleno, y la distancia de la zona al resto de las pistas. Importante: en el menú de la izquierda, tener escogida la opción "Mostrar áreas rellenas en zonas"

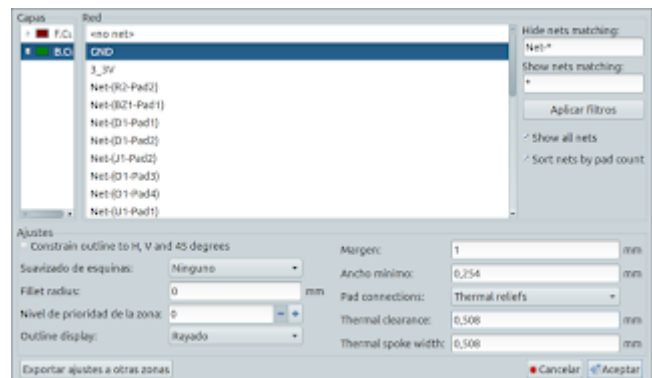
Terminado el proceso, y dibujada la zona, podemos escribir textos en algún lugar donde no afecte a la conectividad.



== == == == == == == ==

Hasta ahora, y con la placa diseñada, es conveniente ejecutar el control de diseño (DRC - símbolo del insecto) para comprobar que no existen errores de conexión.

== == == == == == == ==



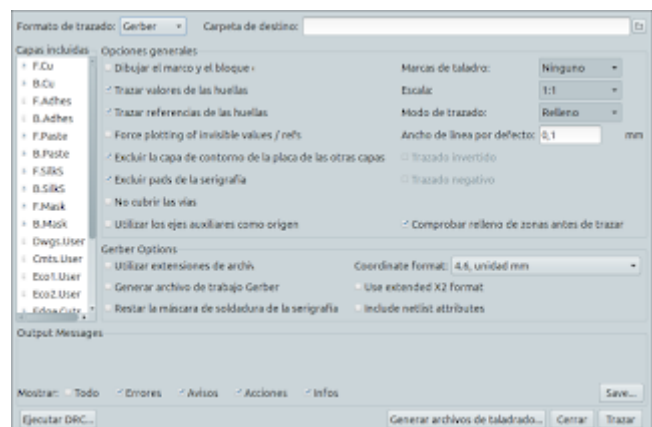
Ficheros Gerber

Aunque todavía no me he puesto a ello (Espero que con este proyecto sí) los ficheros Gerber son necesarios para las CNC, con los que estas máquinas consiguen el taladrado y recorte de las pistas en la placa de baquelita.

En Kicad se consigue con la opción PLOT (trazar), al lado de imprimir. Los valores por defecto se consiguen como en la imagen. Sobre todo, escoger las capas de cobre (B.Cu y F.Cu y la Edges.Cut) aunque es mejor dejar las opciones por defecto.

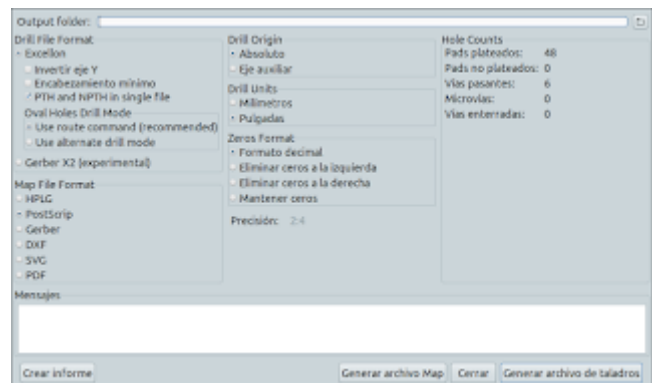
Pulsar en "**Generar archivo de taladrado**". Marcar la opción "PTH y NPTH in single file".

Al volver a la ventana, por fin, pulsar el botón de **Trazar** para generar los distintos ficheros Gerber al proyecto. Al cerrar PCB LAYout Editor y volver a la ventana de proyecto, podremos comprobar como se han generado los ficheros de taladrado (*.drl) y un fichero Gerber por cada capa (*.gbr). Las capas importantes para una placa de una sola cara serán la B.Cu y Edges.Cut. La F.Cu hay que tenerla en cuenta para hacer puentes. las otras contienen las diferentes serigrafías.



Todos estos ficheros pueden visualizarse con la herramienta "Gerber View".

= = = = =



Otros

Se pueden obtener otros ficheros, como los BOM (bills of materials - listados de materiales). También hay otras funcionalidades que me he saltado, básicamente porque las desconozco o se salen del propósito de este pequeño manual.

También se puede usar en PCB Layout Editor un visualizador en 3D de la placa. Hay que buscarse un fichero STEP o STL de la placa NodeMCU V1.0 y exportarlo al formato VRLM 2.0 (*.vrl). Seguramente tendrás que hacer ajustes de escalado y posición.

= = = = =

Enlaces:

1. <https://github.com/jdunmire/kicad-ESP8266>
2. Fichero en 3D del NodeMCU V1.0: <https://grabcad.com/library/nodemcu-2>
3. Ficheros en 3d para kicad:
<https://github.com/Obijuan/videoblog/wiki/Cap%C3%ADtulo-21:-Modelos-3D-en-Freecad-para-Kicad>
4. Freerouting with kicad: <https://freerouting.org/freerouting/using-with-kicad>
5. Manual y documentación de kicad: <https://docs.kicad-pcb.org/>