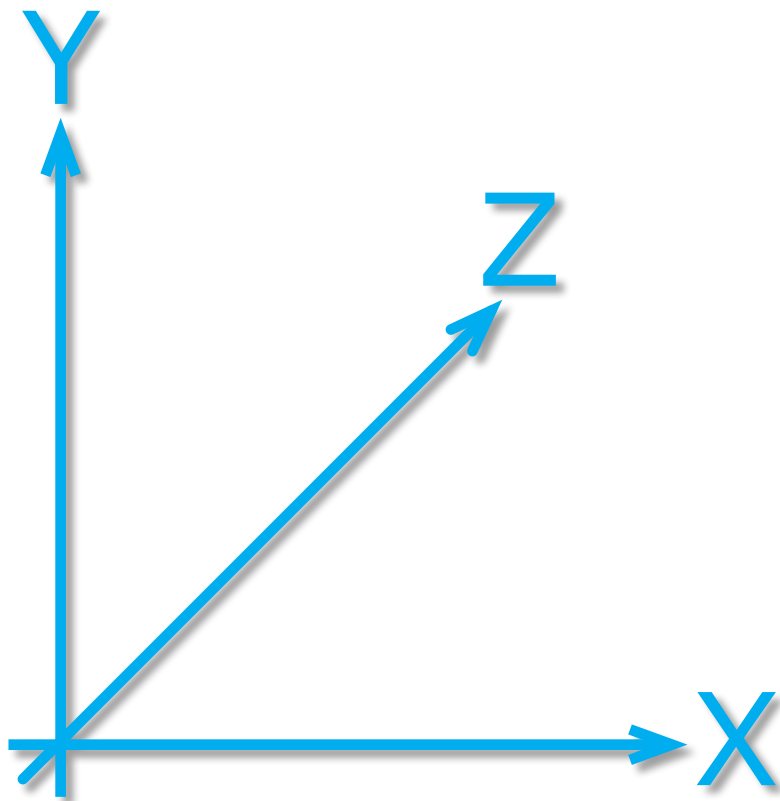


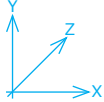
Eagle 3D

paso a paso



paso 1

INTRODUCCIÓN



El creador de Eagle 3D, **Matthias Weißer**, comenta en su web que la idea le vino a raíz de una pregunta en un Foro de discusión sobre Eagle, en el que alguien preguntaba si era posible obtener una vista tridimensional de una PCB a partir de la vista en dos dimensiones que mostraba Eagle. En aquel momento no lo era, así que decidió ponerse manos a la obra y, usando el Lenguaje de Programación de Usuario -ULP-(1), lo hizo posible. Creó lo necesario para obtener un archivo con el formato de Povray, así como las primeras librerías y las definiciones de componentes 3D. Y es mediante el conocido programa de 'trazado de rayos' (Raytracing) Povray con el que podemos obtener una vista en 3D de gran calidad de nuestra PCB.

Algo característico en quien comienza con Eagle 3D es la sensación de que existen muy pocas librerías de componentes y que crear componentes nuevos es muy difícil. Si bien es cierto que faltan bastantes, Eagle 3D trae por defecto la nada despreciable cantidad de 1.532 ya definidos en sus librerías, y la creación de nuevos componentes partiendo de alguno ya existente se verá que no es difícil. Por otro lado, crear 'componentes de la nada' es algo que puede llegar a ser tedioso si el componente es complejo debido a la cantidad de líneas de código que se tendrán que escribir y/o modificar.

Se da por supuesto un cierto conocimiento de Eagle ya que trabajaremos siempre sobre PCB's terminadas. Para abreviar se usará el término PCB para referirnos a la placa de circuito impreso (archivos .brd).

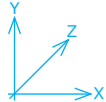
Se verá como crear un encapsulado (package) propio en Eagle y su correspondiente componente en Povray.

Cada vez que se haga referencia a un encapsulado nos estaremos refiriendo a su 'package' en Eagle, y cuando se hable de componente será en referencia a su representación en Povray.

En este manual iremos paso a paso como su nombre indica, y será al final cuando se traten ciertos temas más avanzados.

La información aquí contenida procede en más de una ocasión de diferentes webs. Los enlaces a dichas webs están al final del manual.

(1) En el Manual y en la Guía de Referencia de Eagle encontraréis información sobre ULP, siglas de **U**ser **L**anguage **P**rogramming (Lenguaje de Programación de Usuario).



PROGRAMAS QUE USAREMOS

Eagle 4.16r2 versión LITE -- <http://www.cadsoft.de/freeware.htm>

Eagle 3D 1.05 -- [www.matwei.de/doku.php?id=en:eagle3d:eagle3d\(2\)](http://www.matwei.de/doku.php?id=en:eagle3d:eagle3d(2))

Povray 3.6 -- www.povray.org

Visu 3D -- www.typonrelais.com/visu3D.htm

Notepad++ -- <http://notepad-plus.sourceforge.net/es/site.htm>

ForceVision -- <http://www.forcevision.net/>

Imágenes de muestra -- www.matwei.de/doku.php?id=en:eagle3d:eagle3d

(imágenes en formato .png correspondientes a los encapsulados que incorpora Eagle 3D)

INSTALACIÓN

Una vez hayamos descargado todo lo necesario, procedemos a su instalación.

Comenzaremos instalando Eagle y aceptando aquello que nos propone por defecto, de esta forma se instalará en "c:\Archivos de programa\EAGLE-4.16r2".

En este manual usaremos la versión freeware de Eagle, cuya versión más actual es en este momento la 4.16r2. Si ya existe una instalación previa de Eagle en vuestro PC tenéis que prestar atención a las rutas (path) de instalación a las que se haga referencia posteriormente en el manual.

Instalamos a continuación Eagle 3D. Para ello descomprimos directamente en el subdirectorío de Eagle el archivo en formato zip que previamente descargamos de la web de **Matthias Weißer** (Eagle3d_1_05_27112006 en este momento), de esta forma se creará una carpeta llamada eagle3d con todos los archivos necesarios.

Descomprimos a una carpeta que llamaremos images3d, situada dentro de eagle3d, el contenido del archivo con las imágenes de muestra.

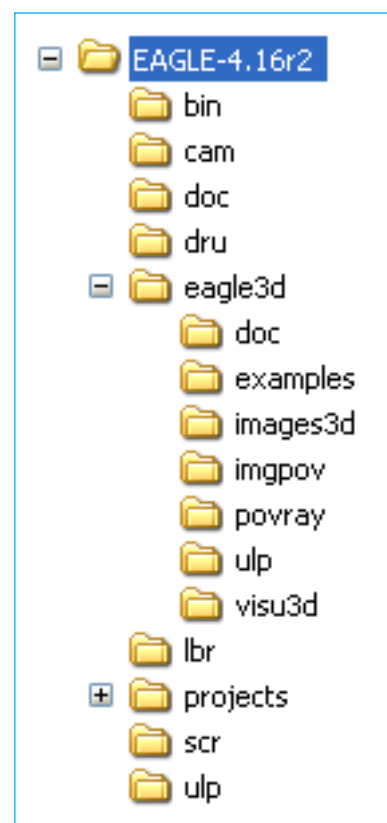
El programa Visu 3D que aprovecharemos también para instalarlo ahora -aunque de momento no lo usaremos-, lo descomprimos a una carpeta llamada visu3d en el subdirectorío eagle3d.

Por último creamos una carpeta de nombre imgpov también en el subdirectorío eagle3d. En la Figura_1.1 podemos ver como deben de quedar nuestras carpetas finalmente.

Procedemos ahora con la instalación de Povray, que por defecto será en "c:\Archivos de programa\POV-Ray for windows v3.6". Una vez instalado, y antes de proseguir, modificaremos con un editor de textos el archivo *POVRAY.INI*, que se encuentra dentro de la carpeta renderer del programa, añadiéndole las siguientes líneas al final del mismo:

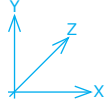
```
Library_Path="C:\Archivos de programa\Eagle-4.16r2\eagle3d\povray"  
Library_Path="C:\Archivos de programa\Eagle-4.16r2\eagle3d\ulp"  
Library_Path="C:\Archivos de programa\Eagle-4.16r2\eagle3d\imgpov"
```

Si vuestra versión de Eagle es otra o el nombre dado a las carpetas difieren de este ejemplo, modificar estas líneas para adaptarlas a vuestro caso concreto.



Figura_1.1

(2) Hay una versión ejecutable (exe) y una versión comprimida (zip) para su descarga, pero la configuración de carpetas y archivos que generan es diferente. En este manual se usa la versión comprimida (zip); si se opta por la ejecutable (exe) las rutas (path) a los archivos no coincidirán.



Povray utiliza la coma (,) como separador entre números y es habitual que la tengamos definida en nuestro sistema de numeración como 'símbolo decimal' y el punto como 'símbolo de separación de miles'. Para evitar que Povray de errores es necesario cambiar el 'símbolo decimal' en nuestro sistema. Para ello, en el Panel de Control de Windows, ejecutamos 'Configuración regional y de idioma' pestaña 'Opciones regionales' botón 'Personalizar...' pestaña 'Números' y cambiamos el 'Símbolo decimal' a un punto (.)

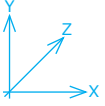
Como editor de textos podéis usar el que más os guste o incluso prescindir de el, ya que los archivos que trataremos se pueden abrir desde el propio Eagle o desde Povray. En el manual usaremos Povray para editar los archivos de librerías y asociación de componentes 3D (.inc y .dat de la carpeta eagle3d/povray) y para editar el archivo *3d41.ulp*, aunque podríamos hacerlo directamente desde Eagle, usaremos en este manual el programa Notepad++

La decisión por mi parte de usar Notepad++ no fue muy difícil: Está en castellano. Me permite abrir todos los tipos de ficheros que vamos a necesitar y lo hace en diferentes pestañas. Dispone de bastantes lenguajes con lo cual puedo ver los listados de código diferenciando colores (a diferencia de Eagle por ejemplo que me permite ver un archivo .ulp pero sin distinciones en el código). Para abrir un archivo basta un click con el botón derecho del ratón y además... es gratuito!

Es muy conveniente tener también un programa que nos permita visualizar bastantes imágenes de cada vez y en diferentes formatos gráficos. De los programas que me parecieron adecuados para la tarea, ForceVision es una de las buenas alternativas. A su favor que también es gratuito y en su contra que es un tanto lento y que lamentablemente, a pesar de disponer de varios idiomas, no está incluido el español entre ellos.

Una de las cuestiones a la que se ha dado importancia a la hora de llevar adelante este manual es que todos los programas a usar fueran gratuitos para uso personal. Pero ello no implica que se usen otros que sean de preferencia para el usuario. Como ejemplo decir que utilizo desde hace años AcdSee y que he buscado y probado alguno que otro gratuito hasta dar con uno que me pareció oportuno, tal es el caso de ForceVision.

Para más información acerca del uso de estos programas dirigirse a sus respectivas webs.



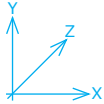
EL PRIMER EJEMPLO

Bien, ya hemos dado todos los pasos necesarios para comenzar con Eagle 3D, pero antes de nada me gustaría llamar la atención sobre algunos detalles que ayudarán a entender mejor el proceso de conversión de Eagle a Povray. Veamos que contiene la carpeta eagle3d y cual es su función:

doc -- archivos de texto que hacen referencia a la web de su creador, Matthias Weißer
examples -- archivos con la extensión .brd que podemos visualizar en Eagle y un ejemplo de nombre MoDsMega.
images3d -- imágenes de muestra de las definiciones de componentes incluídas en el archivo 3dpack.dat.
imgpov -- carpeta que se utilizará para los ejemplos creados en el manual. De momento está vacía
povray -- carpeta que contiene un par de archivos de fuentes (alpha1cd.ttf y eagle3d.ttf), texturas para los condensadores electrolíticos (tex_elko.png) y condensadores electrolíticos axiales (tex_elko_axial.png). Y las librerías con las macros de los componentes (.inc):
cap -- condensadores
capwima -- condensadores WIMA
connector -- conectores
diode -- diodos
ic -- circuitos integrados
qfp -- componentes xQFP
resistor -- resistencias
socket -- zócalos para circuitos integrados
special -- especiales (que no se encuentran en otra parte)
switch -- interruptores
transistor -- transistores
tools -- miscelánea, declaraciones, etc.
user -- de usuario (de momento, en blanco)

ulp:
3d40.ulp -- ULP para versiones de Eagle <= 4.09r2
3d41.ulp -- ULP para versiones de Eagle >= 4.1
3d4(x)func.ulp -- funciones para 3d4(x).ulp
3dcol(_x).dat -- mapas de color en varios idiomas
3dlang(_x).dat -- archivos de idioma.
3dpack.dat -- archivo pack de asociación de componentes
3dconf.dat -- archivo de configuración (de momento, en blanco).
3d_cam.png y 3d_ko.png -- archivos de imagen usadas en el cuadro de diálogo de Eagle 3D.

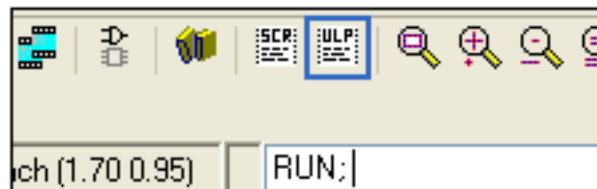
Fijaros que Eagle3d contiene una carpeta llamada ulp al igual que Eagle-4.16r2 y que no hemos de confundir. La carpeta donde se hayan los ulp necesarios para Eagle3d es Eagle-4.16r2/eagle3d/ulp



Ejecutamos Eagle y abrimos desde su Panel de Control el archivo *demo3.brd* que se encuentra en la carpeta *Projects/examples/tutorial*. Si estamos trabajando con la versión que acabamos de instalar nos preguntará si tenemos una licencia u optamos por la versión freeware. En nuestro caso es esta última versión la que usaremos de manera que hacemos un click sobre el botón freeware. Ya no nos volverá a preguntar y la palabra Lite se verá en la barra de Eagle

Una vez abierto, hacemos click sobre el botón **ULP** o usamos el comando **RUN** (run;) en la línea de comandos, tal como se muestra en la Figura_1.2. Nos pide una localización del archivo ULP, que está -recordarlo- en *eagle3d/ulp*.

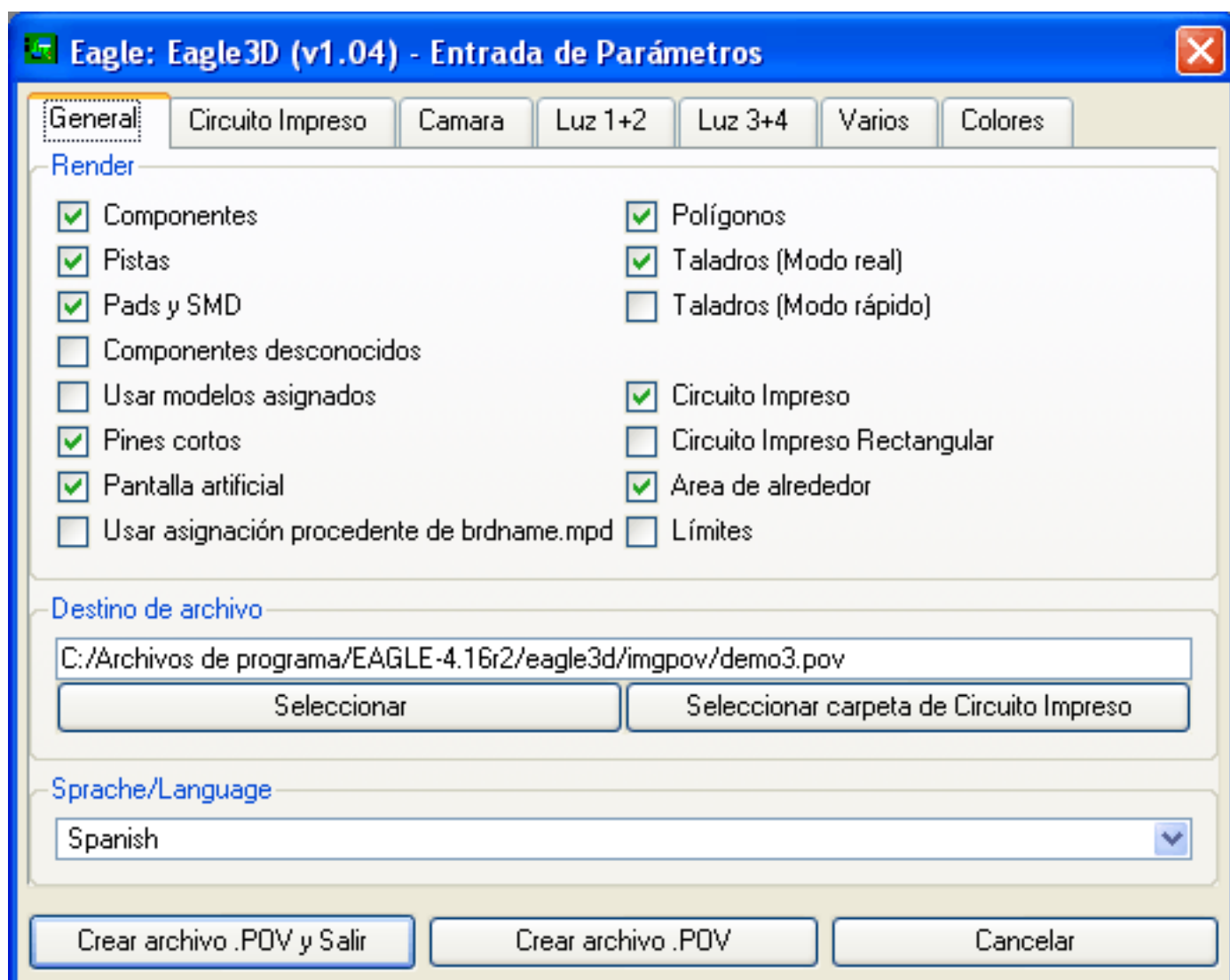
Así, señalamos el archivo *3d41.ulp* y pulsamos el botón Abrir. Ahora tenemos un desplegable para seleccionar el idioma, así que seleccionamos Spanish. Pulsamos el botón Ok y de nuevo Ok para los créditos. A continuación nos pide que seleccionemos una carpeta de salida y seleccionamos la carpeta *imgpov* que creamos anteriormente.



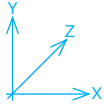
Figura_1.2

Tras seguir estos pasos nos muestra el cuadro de diálogo (Figura_1.3), donde podemos ajustar los parámetros iniciales de Eagle 3D. Fijaros que indica que el archivo de Povray (.pov) resultante se guardará en la carpeta que se seleccionó anteriormente, y toma el nombre del archivo .brd de Eagle al que le añade la extensión .pov. Y ya sin más demora hagamos un click sobre el botón **Crear archivo .POV y Salir...**

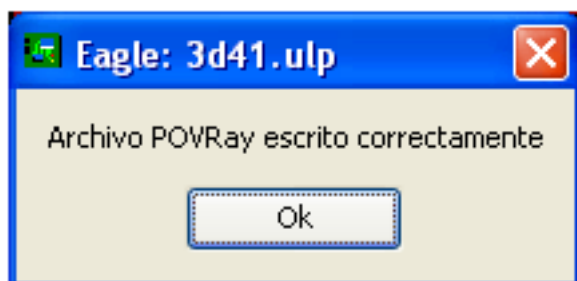
La próxima vez que ejecutemos Eagle 3D se mostrará directamente el cuadro de diálogo



Figura_1.3



La PCB contiene un componente llamado IC1 para el cual nos pregunta si queremos que lleve el Logo de alguno de los 4 fabricantes propuestos o que no lo lleve(3). Después de hacer la selección, y tras una pequeña demora, nos indica que el archivo fue creado. Figuras 1.4 y 1.5

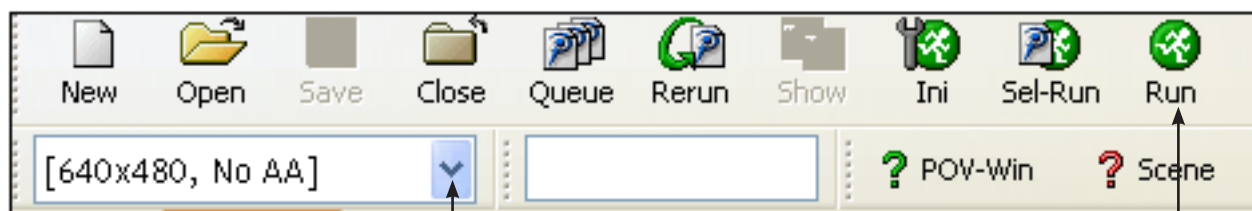


Figura_1.5



Figura_1.4

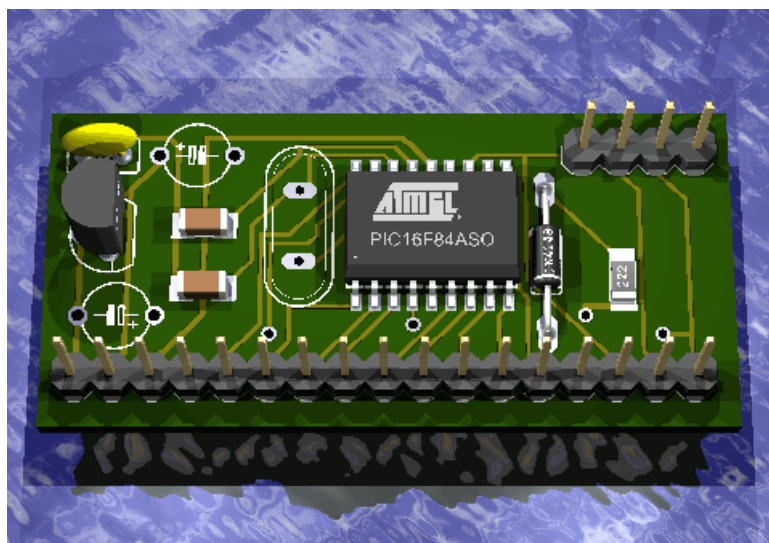
Ahora llegó el momento de ver nuestro PCB, de manera que ejecutamos Povray y abrimos el archivo. Si Povray muestra un cuadro indicando que no se dispone de permisos para acceder al archivo *tools.inc*, pulsar el botón **Aceptar** para otorgárselos. Más adelante trataremos este tema. Una vez cargado el archivo en Povray, seleccionamos la resolución deseada en el desplegable que se encuentra a la izquierda en la barra de herramientas y luego pulsamos sobre el botón **RUN** de esa misma barra, tras unos segundos se mostrará la imagen.



Figura_1.6

1 - Seleccionar Resolución

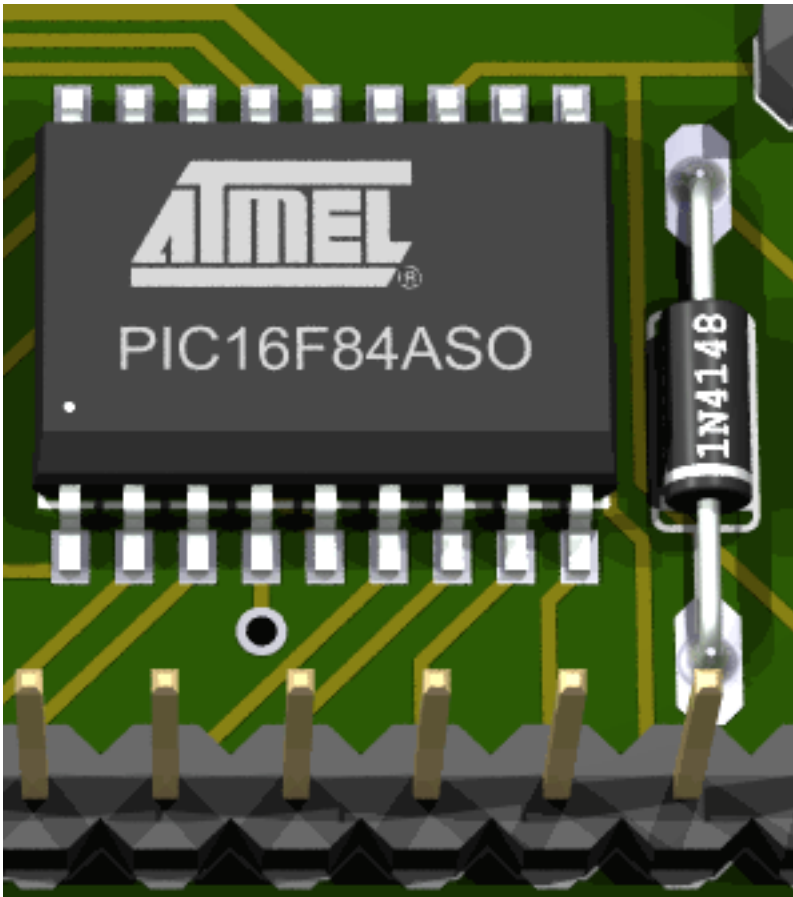
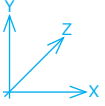
2 - Renderizar



La imagen muestra nuestra PCB en 3 dimensiones pero faltan algunos componentes, y no parece muy lógico que no muestre algo tan habitual como un condensador electrolítico o un cristal de cuarzo ¿Acaso no están estos componentes definidos en las librerías de Eagle 3D? La respuesta es que si lo están, y pronto conoceremos el motivo por el cual no los muestra en esta imagen.

demo3.bmp

(3) En el caso de este ejemplo tan solo se indica si llevará o no Logo la imagen generada. Pero no es el único valor que se puede determinar sino que también podemos decidir el color de un led, si un encapsulado DIP lleva zócalo, si un puente se mostrará abierto o cerrado... etc.



En esta imagen podemos ver un detalle de *demo3.bmp*.

(lógicamente, el pic16f84a es de la casa Microchip, y no de Atmel como aparece en la imagen)

Figura_1.7

Si ahora vemos el contenido de la carpeta ulp encontraremos un par de cambios. El primero es que el archivo *3dconf.dat* que estaba vacío ya no lo está, ahora contiene los parámetros que se establecieron en la instalación, como el idioma o la carpeta de salida para los archivos generados. El segundo cambio es la existencia de un archivo pack que antes no estaba: *3dusrpac.dat*, que de momento está vacío. En este archivo asociaremos nuestros propios componentes, evitando de esta manera modificar el archivo *3dpack.dat*.

Y en la carpeta imgpov tenemos un par de archivos nuevos: el creado por Eagle 3D: *demo3.pov* y *demo3.bmp*, creado por Povray. Como se puede observar, toma por defecto el nombre del archivo cargado en Eagle.

```
C:/Archivos de programa/EAGLE-4.16r2/eagle3d/imgpov/  
C:/Archivos de programa/EAGLE-4.16r2/eagle3d/ulp//3dcol_sp.dat  
C:/Archivos de programa/EAGLE-4.16r2/eagle3d/ulp//3dlang_sp.dat  
21,22,51,52  
21,22,51,52  
  
21,22,23,24,25,26,27,28  
230
```

Archivo *3dconf.dat* con la configuración actual

PROGRAMAS Y ARCHIVOS IMPLICADOS EN EL PROCESO

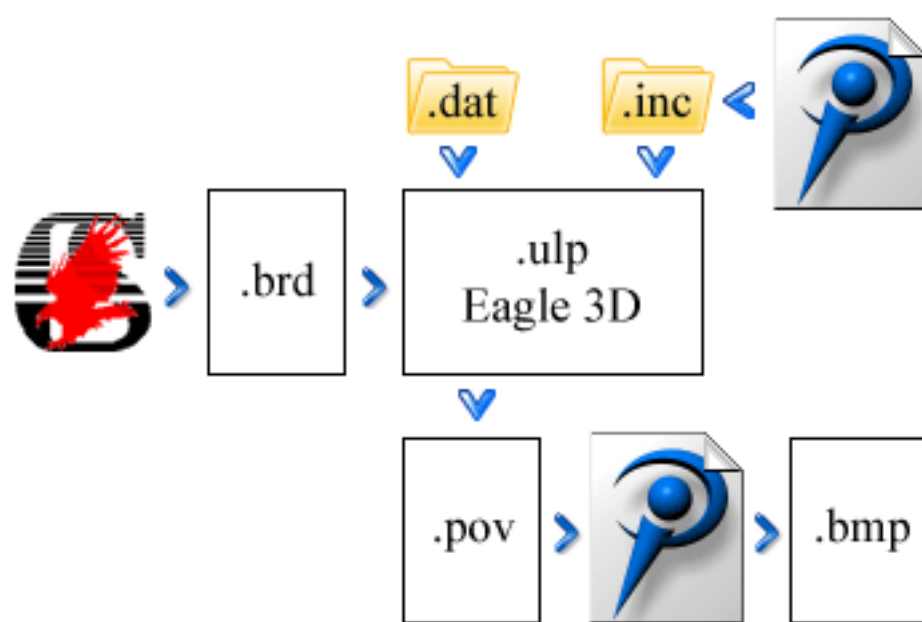


Figura_1.8

En la Figura_1.8 vemos gráficamente los pasos que se han dado en las páginas anteriores.

- 1 - Ejecutamos Eagle.
- 2 - Abrimos la PCB (.brd) del circuito a renderizar, en este caso *demo3d.brd*.
- 3 - Cargamos (comando **run**; o botón **ulp**) Eagle 3D (*3d41.ulp*).
- 4 - Pulsamos botón **Crear archivo POV y Salir...** para obtener el archivo .pov (en este caso *demo3.pov*).

- 5 - Ejecutamos Povray.
- 6 - Cargamos el archivo .pov a renderizar (*demo3.pov* en este caso).
- 7 - Establecemos la resolución de salida y pulsamos sobre el botón **RUN**.
- 8 - Povray guarda el archivo *demo3.bmp*(4) y muestra la imagen renderizada en pantalla.



Figura_1.9

Y en la Figura_1.9 lo vemos con algo más de detalle.

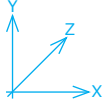
Eagle 3D (*3d41.ulp*) hace una llamada a *3dfunc41.ulp*, *tools.inc* y *user.inc* mediante (#include) y a los archivos de configuración *3dconf.dat* y de asociación de componentes. Estándar: *3dpack.dat*, y de usuario: *3dusrpac.dat*.

Establece los parámetros por defecto para el archivo .pov basándose en el panel de control, los datos que obtiene de la PCB y las librerías con sus respectivos nombres de encapsulado.

Una vez creado el archivo .pov y abierto en Povray solo se mostrarán al renderizar aquellos encapsulados cuyo nombre en las librerías de Eagle se corresponda con el nombre dado en cualquiera de los archivos pack de asociación: *3dpack.dat* o *3dusrpac.dat* y su librería se haya definido en alguno de los archivos de macro(.inc). Se volverá sobre estos archivos más adelante.

No se hizo uso para el proceso de otros programas aparte de Eagle, Eagle 3D y Povray. Los demás programas aquí propuestos no son imprescindibles, pero si que son extremadamente útiles como posteriormente se verá.

(4) El formato de archivo establecido por defecto es .bmp, sin embargo Povray puede usar también los formatos .tga y .png



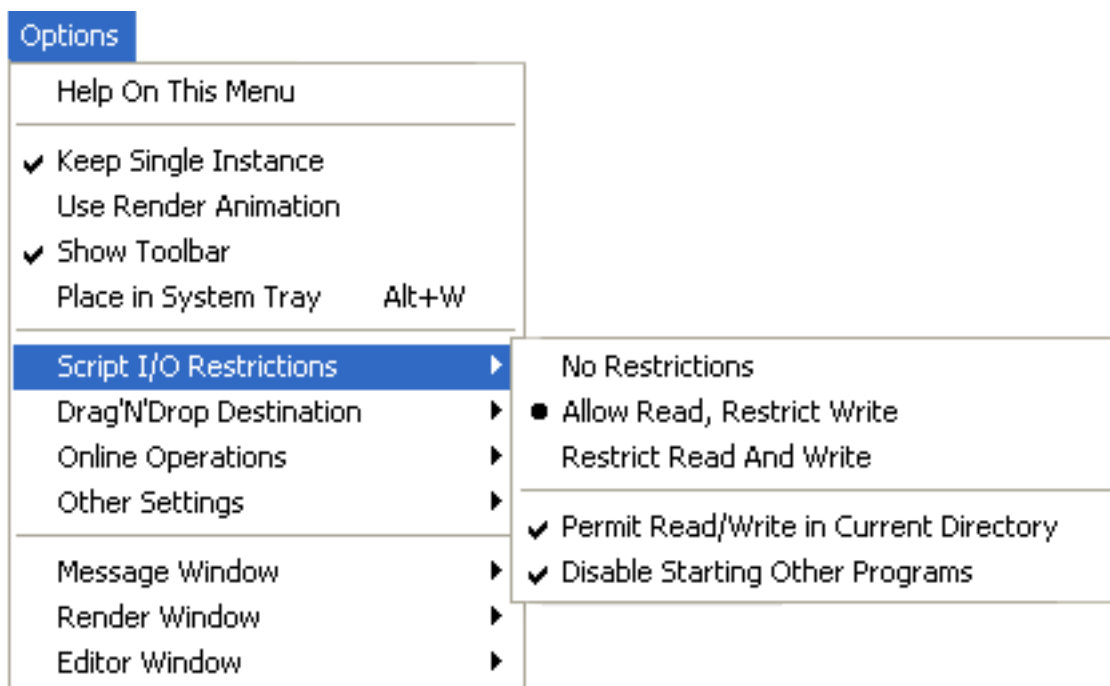
SOLUCIONAR ERROR DE PERMISOS EN POVRAY

Cuando tratamos de renderizar nuestro archivo .pov en Povray nos muestra un cuadro de diálogo indicando que no se dispone de permisos de lectura para *tools.inc* invitándonos a cancelar el proceso o a permitir que Povray establezca esos permisos temporalmente pulsando el botón **Aceptar**.

Para evitar que esta situación se repita continuamente a lo largo del manual, podemos optar por, en principio, dos soluciones:

1 - Seleccionar como carpeta de destino para nuestros archivos .pov la carpeta eagle3d/povray(5) en vez de la carpeta eagle3d/imgpov. La carpeta povray tiene ya los permisos establecidos y por eso los archivos que estén en su interior compartirán esos permisos. Esta opción tiene en su contra a mi entender que tendremos mezclados nuestros trabajos con los archivos de instalación de Eagle 3D.

2 - En el menú **Options** de Povray seleccionar **Scrip I/O Restrictions** y ahí **Allow Read, Restrict Write** y **Permit Read/Write in Current Directory**. Tal como podemos ver en la Figura_1.10. De esta forma tendremos más organizado nuestro trabajo, ya que podemos llegar a tener un número considerable de archivos tanto en formato .pov como .bmp y dejamos la carpeta povray para macros y texturas.



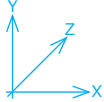
Figura_1.10

Volveremos sobre este tema porque existe una tercera opción que es la de definir los permisos 'a mano' tanto de escritura como de lectura, y que es además la más adecuada.

(5) Será suficiente con editar el archivo *3dconf.dat* y cambiar la ruta de destino.

paso 2

ENCAPSULADOS Y COMPONENTES



En la imagen creada por Povray en el capítulo anterior se apreciaba la falta de dos condensadores y un cristal de cuarzo, y se decía que estos componentes sí están definidos en el pack de asociación (*3dpack.dat*) y en las librerías de componentes de Eagle 3D. La pregunta que nos hacíamos era ¿porqué no los muestra entonces?

Bien, en esta versión de Eagle 3D se dispone de 1.532 componentes 'asociados' al correspondiente encapsulado (package) de Eagle, pero la cantidad de encapsulados en el mismo Eagle es mucho mayor, y además determinado componente en Eagle puede llegar a tener más de un encapsulado y en diferentes librerías.

Veámoslo en imágenes. Abrimos el archivo *demo3.brd* en Eagle y solicitamos información (botón **Info**) sobre los componentes que faltan. Tenemos dos condensadores, uno -c4- de 47u/25v y otro -c5- de 47u, ambos con encapsulado TAP5-45 y en la librería rcl (Figura_2.1). Vemos que el cristal tiene un encapsulado QS y se haya en la librería special (Figura_2.2).

```
Element: C4 (0.175 0.2
Value = 47u/25V
Rotation = R180
Package = TAP5-45
Library = rcl
Smashed

ELECTROLYTIC CAPACITOR
```

Figura_2.1

```
Element: Q1 (0.65 0.52
Value =
Rotation = R90
Package = QS
Library = special
Smashed

CRYSTAL
```

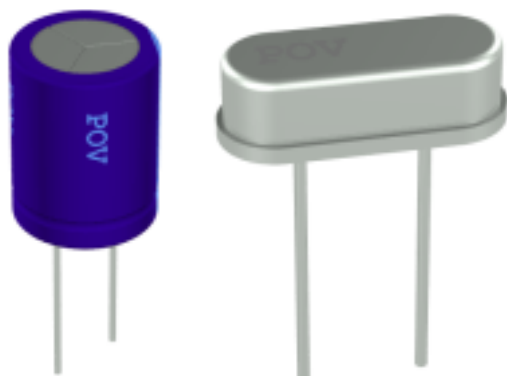
Figura_2.2

Ahora abrimos el archivo *demo3.pov* en Povray y nos desplazamos hasta el final del listado para ver las siguientes líneas:

```
//Parts not found in 3dpack.dat or 3dusrpac.dat are:
//C4 47u/25V TAP5-45
//C5 47u TAP5-45
//Q1 QS
```

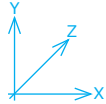
Figura_2.3

Fijándonos en la primera: No encontrados en *3dpack.dat* o *3dusrpac.dat*, que son los archivos de asociación de componentes. La Figura_2.4 muestra los dos componentes en la carpeta de imágenes de ejemplo (images3d) que son, como recordaréis, las imágenes de muestra de los componentes que contiene *3dpack.dat*. Básicamente no los muestra porque el nombre dado al encapsulado en Eagle no coincide con el nombre dado en Eagle 3D. Es decir, en el archivo *3dpack.dat*.



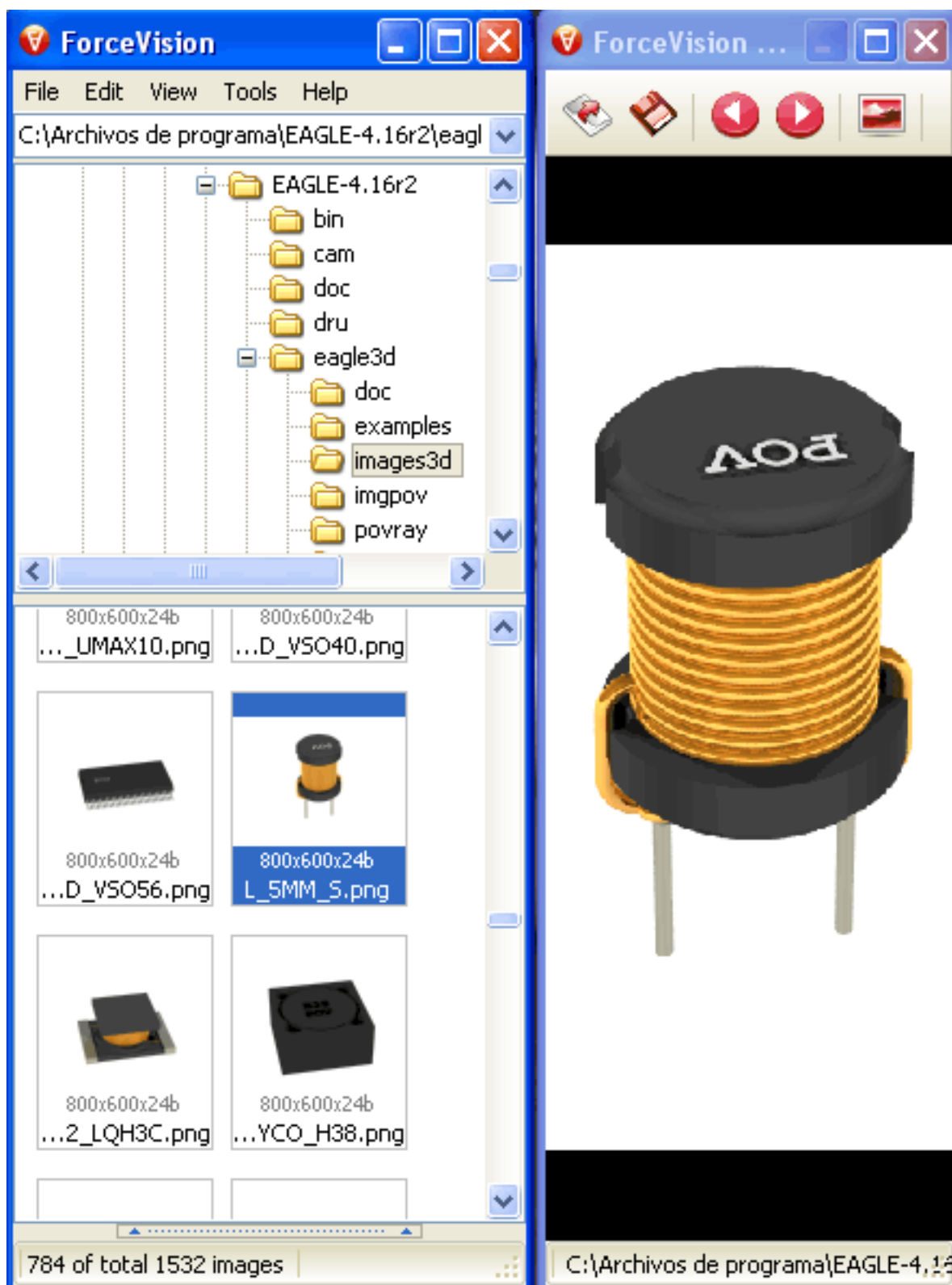
Figura_2.4

Veremos a continuación como resolverlo. Y dado que usaremos archivos 'sensibles' al buen funcionamiento de Eagle 3D no está de más realizar una copia de seguridad de la carpeta eagle3d en otra localización de nuestro disco duro por si tenemos que recuperar algún archivo. Haciendo especial hincapié en los .dat, .inc y .ulp.

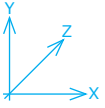


ForceVision

Recurrimos a su ayuda para localizar los componentes de forma más sencilla. Cuando lo ejecutamos por primera vez muestra una separación de dos ventanas en que podemos ver a la izquierda la barra de navegación y en su parte inferior el contenido de la carpeta seleccionada, y en la ventana de la derecha el editor gráfico propiamente dicho. En el se mostrará ampliada la imagen que tengamos seleccionada en la ventana izquierda. Figura_2.6 para más detalles.



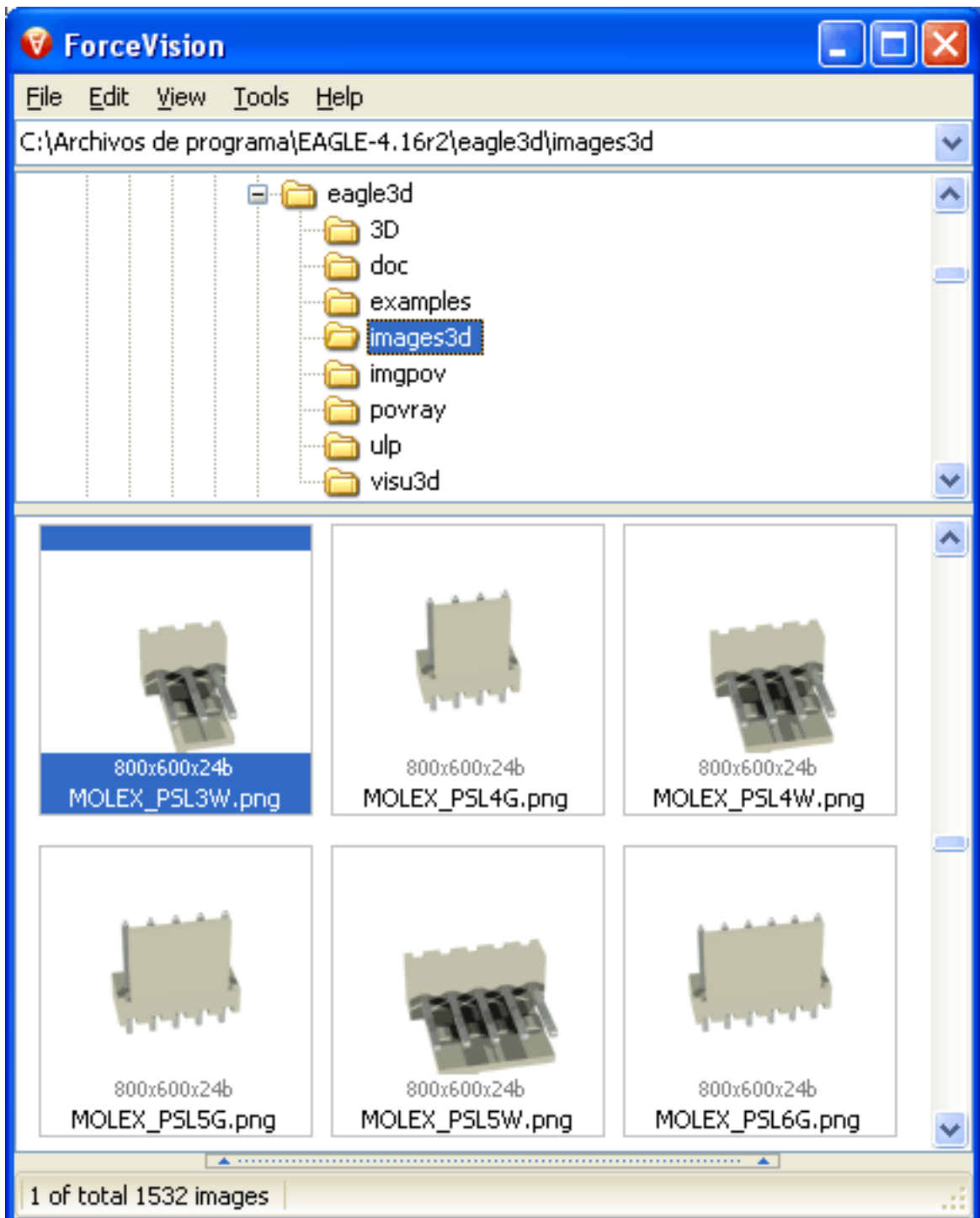
Figura_2.6



Por defecto la parte inferior muestra los nombres pero no las imágenes, para cambiar entre vistas pulsamos **F8** para las imágenes (Thumbnails en el programa) o **F11** para la lista de nombres.

Como me interesa ver el mayor número de imágenes posibles, maximizo el tamaño de la ventana izquierda (quedando la otra de fondo) y en el menú **Tools** opción **Options..** pestaña **Thumbnails** selecciono un tamaño (size) para las imágenes que me permita leer su nombre con facilidad en mi monitor y si lo quiero ver ampliado en tamaño, restauro la ventana para que se muestre la otra. El programa tiene otras características que no se contemplan en este manual.

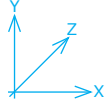
Notaréis una pequeña demora debido a la cantidad de imágenes que contiene la carpeta images3d.



Figura_2.7

paso 3

EL PANEL DE CONTROL



El Panel de Control de Eagle 3D está formado por 7 pestañas con un buen número de opciones, desde las cuales podemos establecer los parámetros básicos que le indican a Povray como debe generar la escena. Estos parámetros establecidos por defecto los toma del archivo *3d41.ulp* cuando se ejecuta, y pueden ser cambiados a otros de nuestra elección.

Demos antes un repaso por las características, y una pequeña descripción extraída del manual de Eagle 3D en la página de su creador:

- Selección y posicionamiento de componentes conocidos
- Pistas (Tracks)
- Agujeros (Holes)
- Vías
- Polígonos
- Serigrafía (Silk screen)
- Etiquetado de componentes
- Selección de color para LEDS
- Asociación manual de encapsulados
- PCBs con contornos cuadrados
- PCBs con contornos redondos
- Recortes de PCB de cualquier tipo
- Zócalos DIP y PLCC
- PCBs multicapa (las capas internas son visibles)
- Archivo de respaldo para asociaciones manuales
- Archivo de asociación auxiliar
- Selección de idioma
- Combinar varias PCBs

PCB

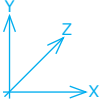
La PCB será generada como un prisma (por defecto) o como una caja (box). Los contornos dibujados en la capa 20 (Dimensions) deberían ser polígonos cerrados. Los contornos de PCB muy raros podrían dar lugar a problemas de visualización, y de ser así se debería seleccionar la opción Circuito Impreso Rectangular en el cuadro de diálogo del Panel de Control. Se pueden usar PCBs cuyos contornos estén completamente dibujados con líneas (wires) o círculos. El uso de arcos sólo está permitido a partir de Eagle 4.1.

RESISTENCIAS: BANDAS DE COLOR Y ETIQUETAS

Las resistencias con un valor especificado por el usuario se dibujarán automáticamente con los colores de resistencia correctos. Sólo se dibujarán 4 bandas de color, con una tolerancia fija (5%, banda color oro). También las resistencias de SMD serán etiquetadas por la combinación correcta de tres números. El valor del componente debe tener un cierto formato. Los formatos permitidos incluyen:

- 1000 corresponde a 1 kohm
- 1k corresponde a 1 kohm
- 4,7k corresponde a 4700 ohmios
- 3M corresponde a 3 Mohm
- 1R2 corresponde a 1,2 ohmios

Pueden tener un número variable de dígitos, como '12k' o '470'. Si este formato no coincide exactamente, no se puede prever el resultado de la rutina que calcula (estima) los colores/números correspondientes. Otra información, que no será evaluada, podría ser añadida después del valor de la resistencia. Esta información no debería estar separada de los caracteres de código de resistencia únicos ('R'; 'k'; 'M'; ',' y '.').



PCBs MULTICAPA

La estructura interna de una PCB multicapa será visible, si la PCB está deshabilitada y su grosor y el cobre es aumentado considerablemente. Un grosor de aproximadamente 30mm para la PCB y 1 mm para el cobre es recomendado para una configuración de 4 capas. Si la PCB está habilitada, todas las capas de cobre interiores no son mostradas porque no se pueden ver. De esta forma, no serán renderizadas, y se conseguirá un ligero aumento de velocidad en el proceso.

POLÍGONOS

Para mostrar un polígono correctamente se debe calcular con RATSNEST antes de ejecutar Eagle 3D. Los polígonos se construyen usando muchas pequeñas cajas en Povray. Su ancho y su número se corresponde al 'width' usado por el polígono en Eagle. ¡Valores muy pequeños pueden conducir a archivos de Povray enormes (imás de 100Mb!). Para tratar tal archivo, Povray necesita mucha memoria y cantidad (y realmente quiero decir cantidad) de tiempo. Hay que evitar usar un ancho de línea demasiado fina. Por regla general: No más finas que 0.1 mm. De hecho, incluso un ancho grueso puede causar tiempos enormes de ejecución. Como regla empírica podemos asumir que aproximadamente 500 a 1000 objetos requerirán 1 MB de memoria.

COMPONENTES DESCONOCIDOS - MODELOS DEFINIDOS POR EL USUARIO

Puede ser muy útil asociar componentes a mano. A menudo hay un modelo en la biblioteca, pero ninguna asociación correspondiente. Podemos asociar un modelo seleccionable a un dispositivo de Eagle. Será guardado en un archivo llamado *boardname.mpd*, que puede ser usado en futuras ocasiones.

Si queremos cambiar una asociación debemos corregir el archivo *3dusrpac.dat*, como ya se ha visto anteriormente. Se volverá a este tema con un ejemplo concreto unas páginas más adelante.

AGUJEROS (HOLES)

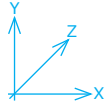
Los taladros (para vías, pads, y agujeros) serán mostrados como cilindros negros por defecto. Para una descripción general no hay casi ninguna diferencia con verdaderos agujeros. Esta opción define un diámetro de taladro inicial, a partir del cual todos los agujeros serán dibujados como taladros reales. Por defecto, un valor muy utilizado son 2 mm. En caso de una vía, puede resultar que las pistas parezcan 'caer' en el agujero. En la mayor parte de situaciones esto no sucede, pero en algunas condiciones si que puede pasar sin embargo. Si un polígono está dibujado sobre un vía, quedará igualmente tapado.

SERIGRAFÍA (SILK SCREEN)

Las capas que son usadas para la serigrafía pueden ser configuradas en la pestaña **Varios** del Panel de Control. Si el número de la capa es impar el objeto será colocado en la capa fondo de la PCB. Y si el número de la capa es par lo será en la capa top.

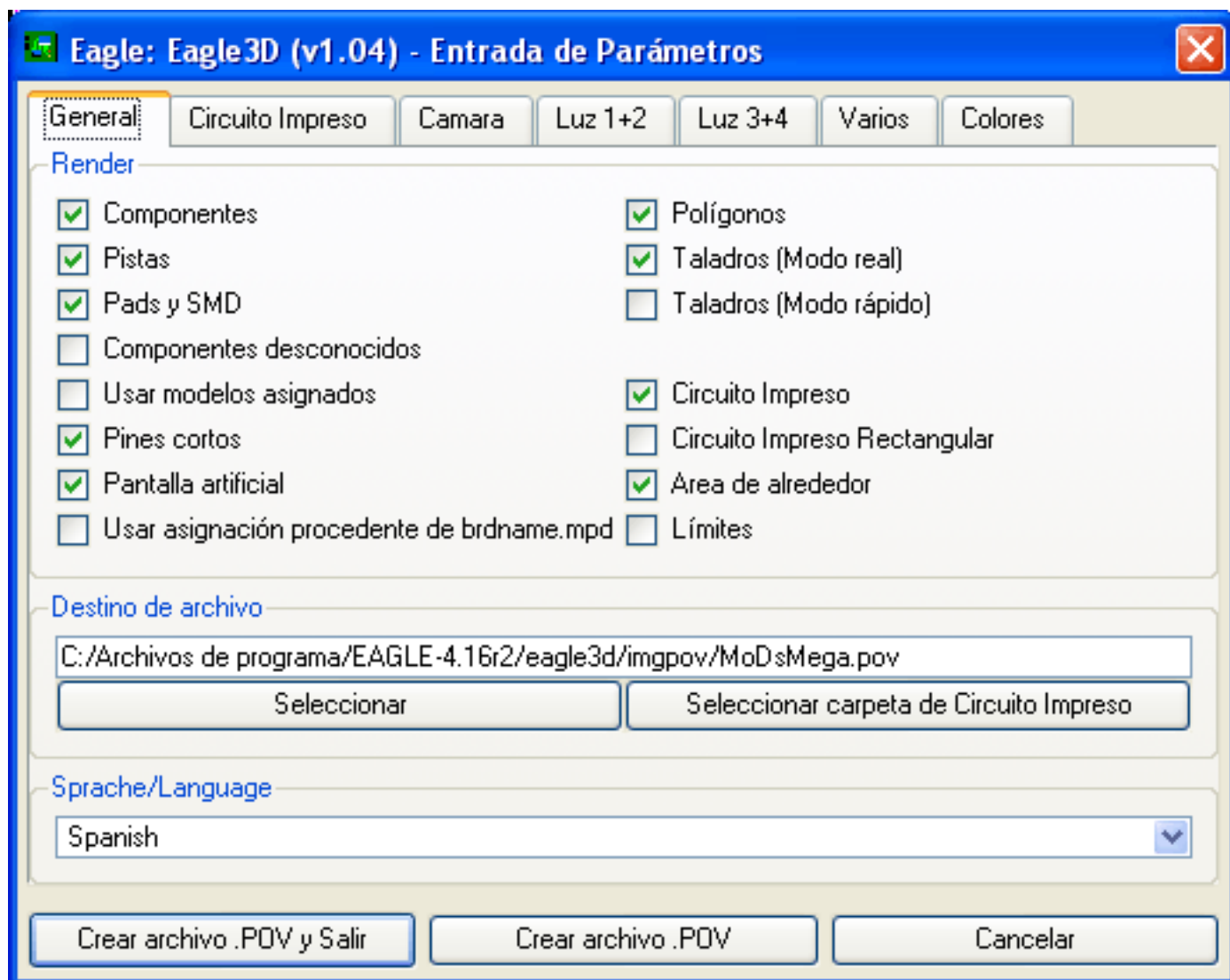
PREFIJOS

Algunos encapsulados pueden ser usados por más de un tipo de componente. Un ejemplo típico son los encapsulados SMD como 0805 y 1206. Por este motivo, los encapsulados (1210, 1206, 0805, 0603 y 0402) toman como prefijo el primer carácter del nombre de dicho componente y luego se realiza la asociación. De esta forma, un componente llamado R12 y con el encapsulado 0805 se renderiza como una resistencia R0805. Por otra parte un componente llamado C43 y con el encapsulado 0805 será renderizado como un condensador C0805.



PARÁMETROS DE ENTRADA

Veamos de nuevo el cuadro de diálogo que se muestra al iniciar Eagle 3D porque trabajaremos con el en las próximas páginas:



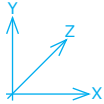
Muestra en un principio la pestaña **General** que está dividida en 4 partes: Render, Destino de archivo, Sprache/Language (parece que no todo ha sido traducido..) y tres botones en el margen inferior.

En Render decidimos como queremos que se genere nuestra imagen en Povray mediante la selección o deselección de ciertas opciones.

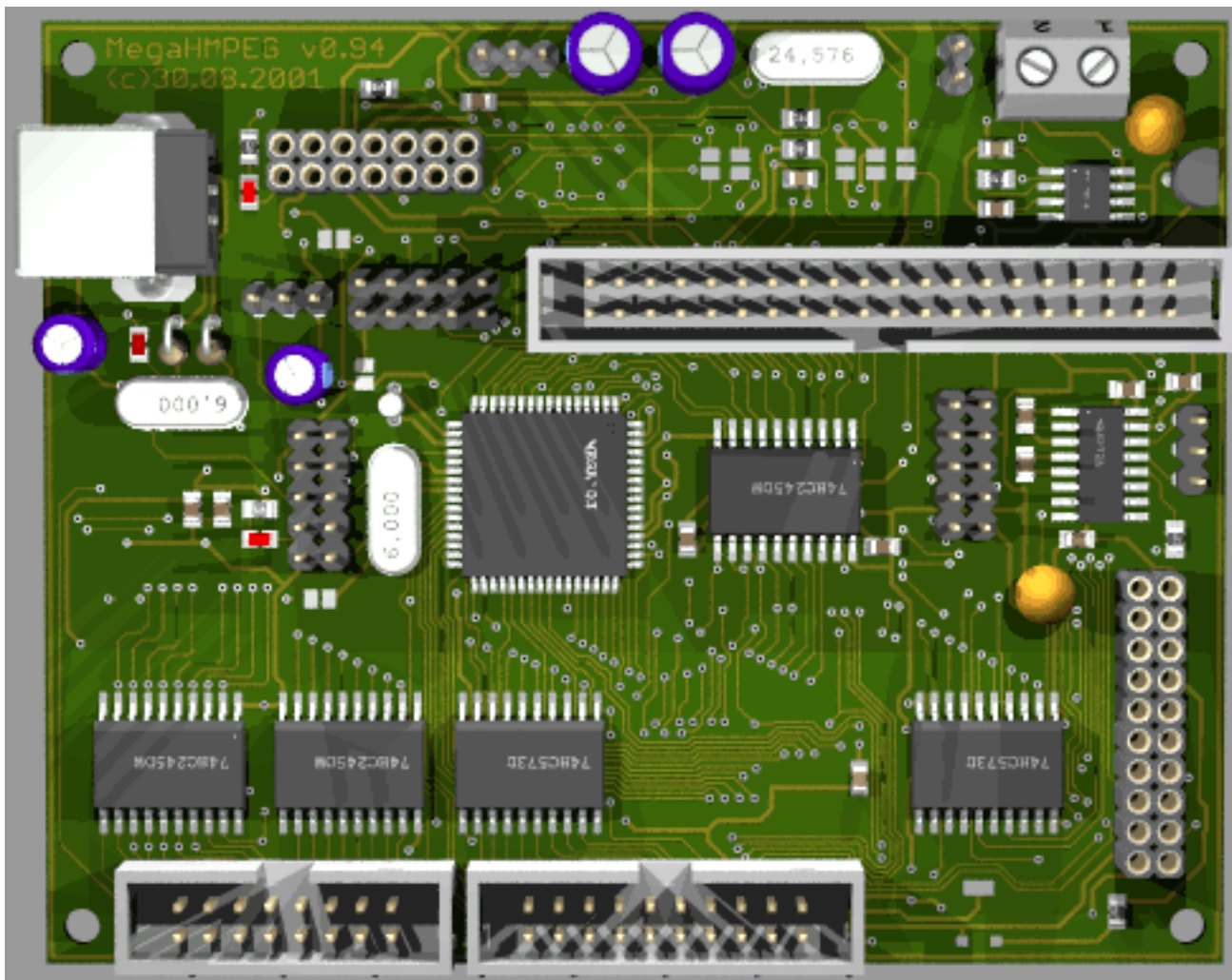
En Destino de archivo mostrará por defecto el nombre del archivo .brd de Eagle y la carpeta establecida en la instalación. Pulsando el botón **Seleccionar** podemos cambiar su ruta y su nombre, y pulsando el botón **Seleccionar carpeta de Circuito Impreso** hará justamente eso, establecerá la carpeta desde la cual se abrió el archivo .brd como carpeta de destino.

En Language seleccionamos el lenguaje que queremos para Eagle 3D.

Y los botones inferiores: **Crear archivo .POV y Salir** crea el archivo y sale de Eagle 3D. **Crear archivo .POV** crear el archivo sin abandonar Eagle 3D y **Cancelar** cancela la creación del archivo y abandona Eagle 3D

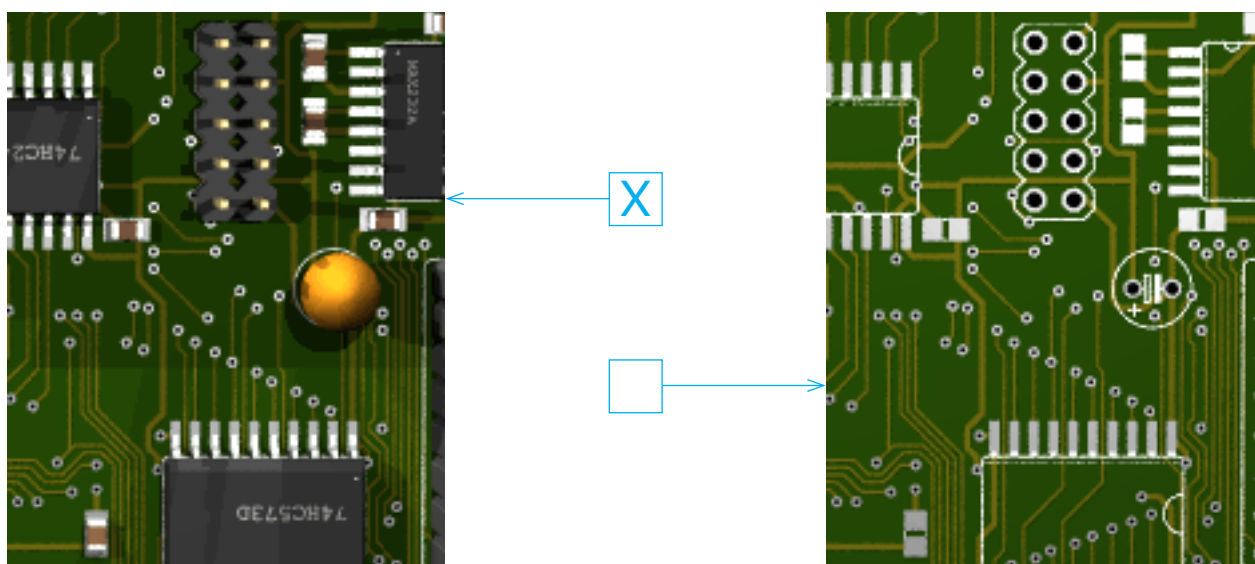


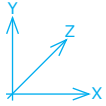
Para los ejemplos trabajaremos con un archivo que nos proporciona Eagle 3D, que se haya en la carpeta eagle3d/examples y cuyo nombre es *MoDsMega.brd*. Como se hizo anteriormente, se abrirá el archivo desde Eagle (Menú File/Open/Board...) y se verá de que forma influyen unas y otras opciones. Esta es la imagen completa generada en Povray, renderizada con todas las opciones por defecto:



Para mayor claridad, en los ejemplos se mostrarán determinadas partes de la imagen nada más. Por el mismo motivo, se ha decidido prescindir del 'decorado' (el símil marino característico de Eagle 3D) y se han rotado las imágenes -30° en el eje x.

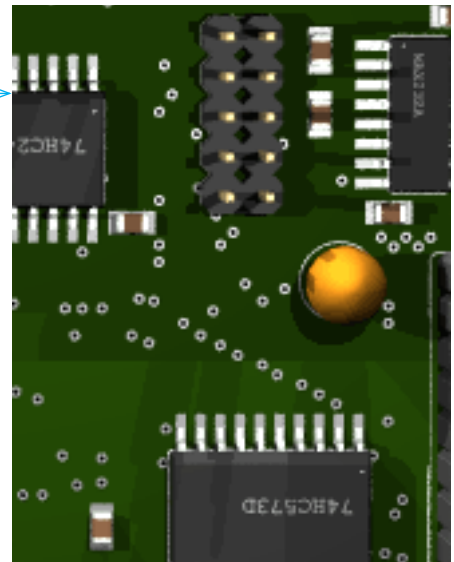
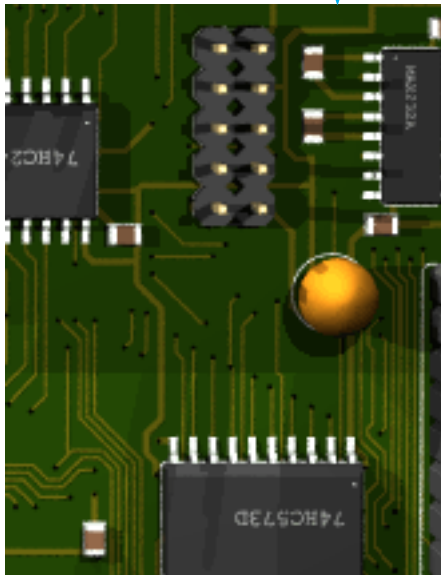
En estas imágenes podemos ver la opción Componentes activada y desactivada.





También podemos ocultar las Pistas

Y los Pads

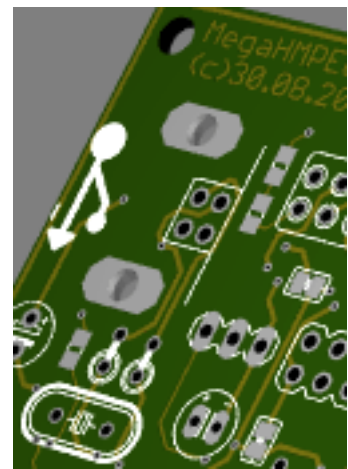
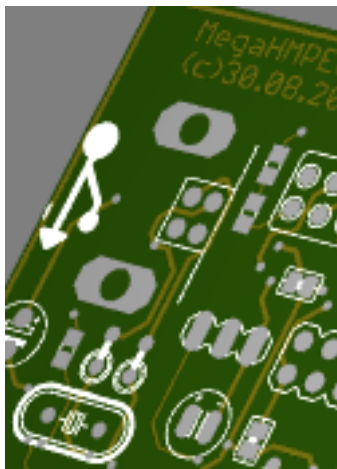


Los Agujeros (Holes) tienen tres opciones de representación:

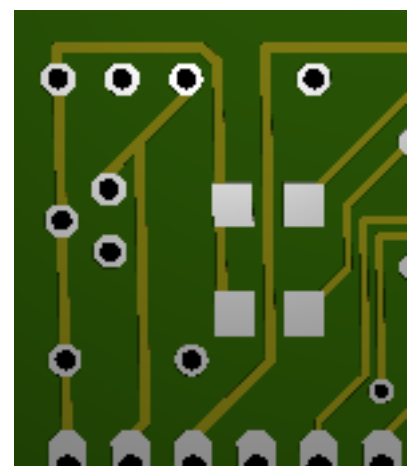
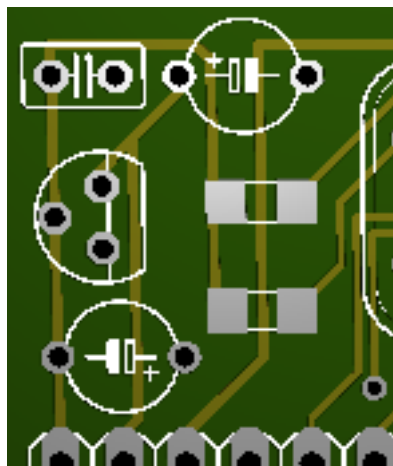
Ninguno

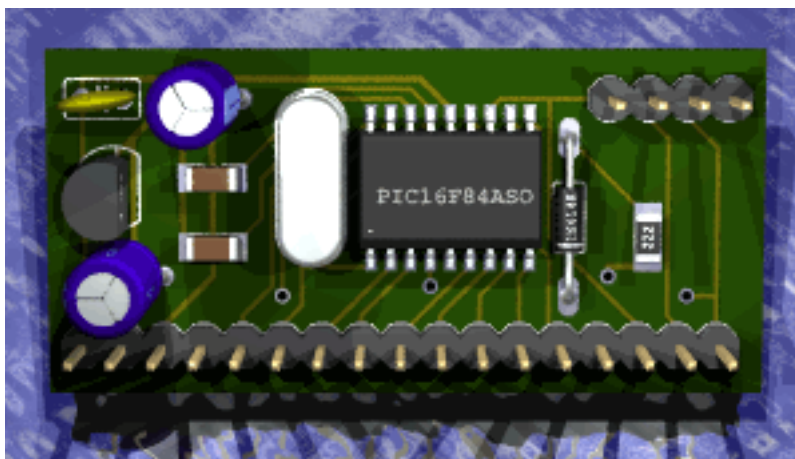
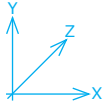
Modo Rápido

Modo Real

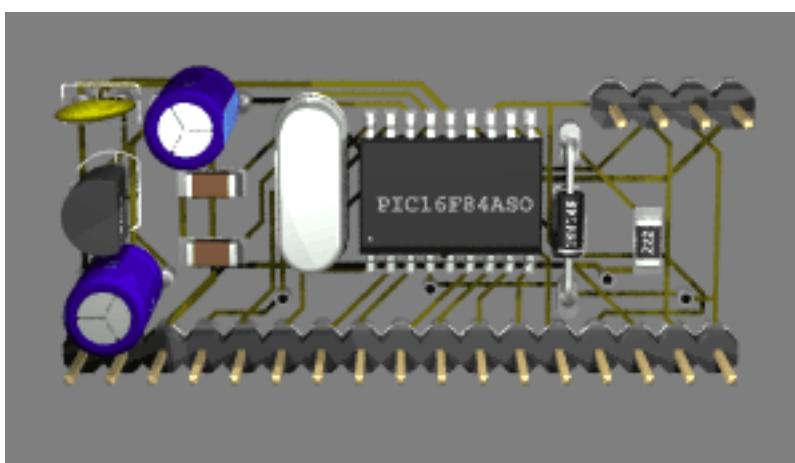


Mostrar u ocultar la Serigrafía (Pantalla artificial o Silk Screen) también es posible, tal como podemos ver en las siguientes imágenes de la PCB *demo3.brd*, ya tratada anteriormente.



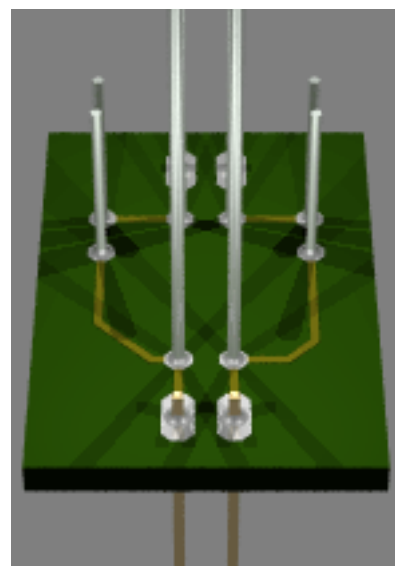
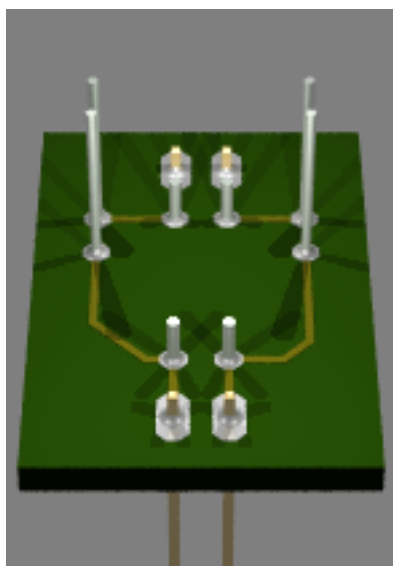
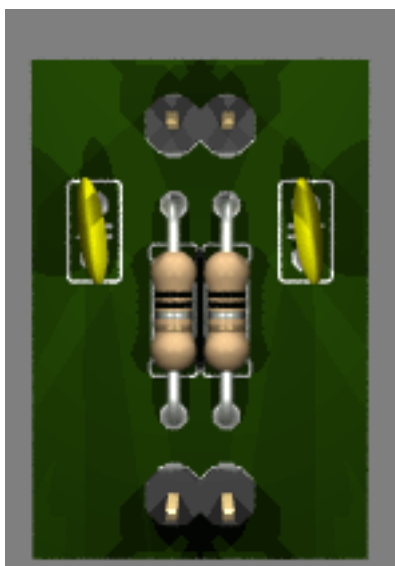


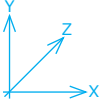
Usamos esa misma PCB para mostrar la pantalla de 'adorno' (Área de alrededor), que es una imagen a modo de plano simulando un mar, y sobre la que se eleva nuestra PCB. Tener esta opción activada supone un mayor tiempo de renderizado.



Y ahora desactivamos la opción Circuito Impreso para que éste no se muestre.

Construí un pequeño circuito para ver la opción de Pines Cortos, que no deja de tener su sorpresa. En la imagen de la izquierda tenemos una vista superior con Pines Cortos habilitado, en el centro la misma imagen vista desde abajo y a la derecha con la opción deshabilitada. Como véis, así como las resistencias 'reaccionan' a la opción, los condensadores permanecen igual y no muestran cambios al habilitar o deshabilitar la opción de Pines Cortos. El motivo es que están contruídos de forma diferente, las resistencias hacen uso de una macro que a su vez llama a un objeto del archivo de macros *tools.inc* que contempla la variable `pin_short = on` del elemento Pines Cortos. Y para el condensador, su autor creó una macro que no contempla esta posibilidad. Se verá con más detalle en la parte avanzada del manual.





Recordaréis que el archivo .pov generado desde Eagle 3D mostraba en sus últimas líneas aquellos encapsulados de Eagle para los que no encontró la asociación correspondiente a su componente de Eagle 3D, en los archivos *3dpack.dat* y *3dusrpac.dat*. Y tratábamos de resolverlo mediante el uso de estos dos archivos digamos de una forma un tanto artesanal y dada a errores debido entre otras cosas a la extensión del archivo *3dpack.dat* y a su nada despreciable número de campos. Eagle 3D nos permite solucionar en gran parte este problema ya en el momento de generar el archivo .pov, y lo hace mediante el uso de una opción del Panel llamada Usar modelos asignados, que está estrechamente relacionada con la opción de Componentes desconocidos.

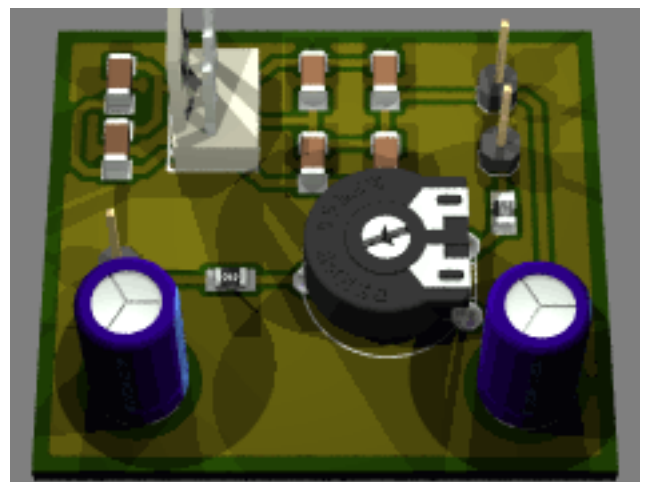
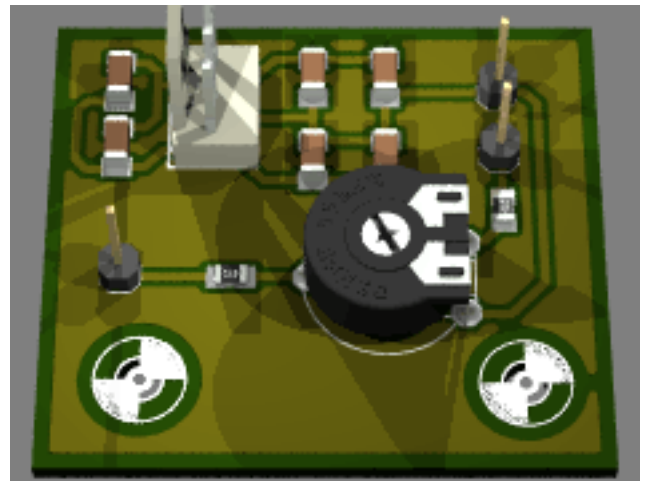
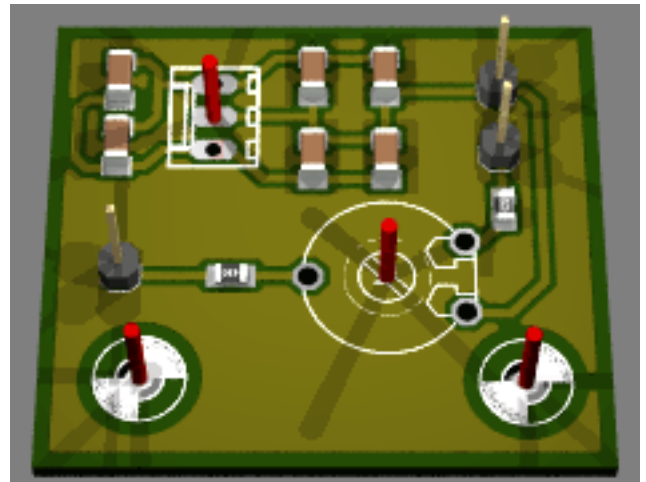
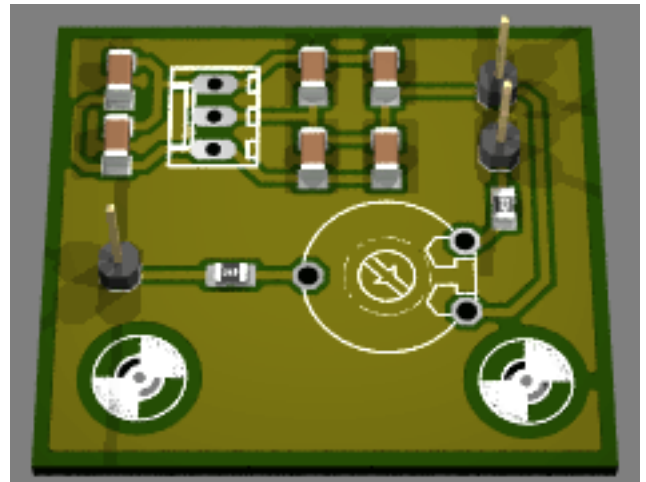
En estas cuatro imágenes se puede ver un circuito para el que se seleccionaron encapsulados a sabiendas de que no estaban asociados en *3dpack.dat* y por tanto no se mostrarían. La imagen superior es con las opciones por defecto, la segunda es con la opción de Componentes desconocidos habilitada. Como véis, muestra unos cilindros de color rojo sobre cada componente no encontrado.

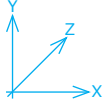
La tercera imagen es después de habilitar la opción Usar modelos asignados (al habilitar esta opción se habilita también automáticamente la opción Componentes desconocidos), y realizadas las debidas asociaciones.

Y la cuarta es consecuencia de cometer un error en el proceso anterior..

Tal como hicimos en su momento para saber que asociación de encapsulados-componentes necesitábamos hacer, lo primero es pedir la 'información' sobre el encapsulado en Eagle. En este ejemplo, los encapsulados cuyos componentes no se muestran son: un potenciómetro de ajuste con encapsulado LI10 de la librería pot.lib, un conector 22-27-2031-03 con encapsulado 6410-03 de la librería con-molex.lib y dos taladros (Holes) MOUNT-HOLE2.8 con encapsulado 2,8 y en la librería holes.lib.

Ahora buscamos con ForceVision entre las imágenes de muestra para saber que nombre tienen estos componentes en Eagle 3D y nos encontramos que para el conector tenemos MOLEX-PSL3G y para el potenciómetro nos servirá RES_DIS_TRIM_PT10_H_10. Lógicamente no encontraremos un componente que sea un 'taladro' entre las imágenes.

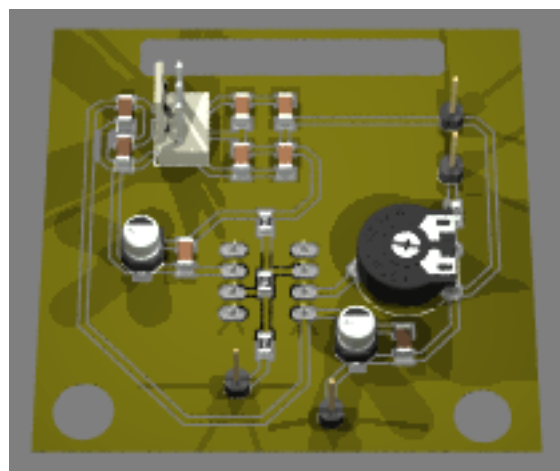
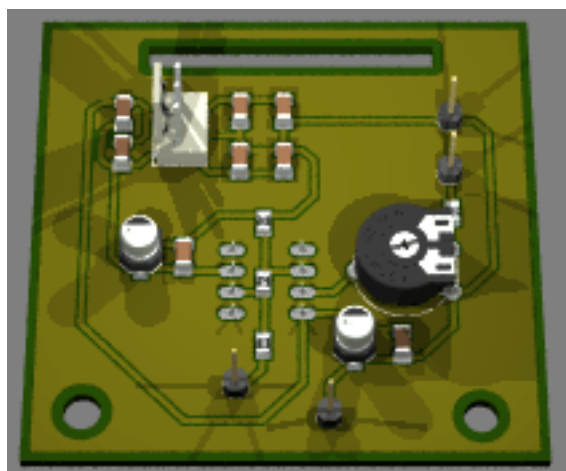




Otra opción que tenemos en el Panel de Control es la de forzar a que nuestra PCB se genere en un formato rectangular. Ello se debe a que formas muy extrañas de PCB pueden llevar a tiempos de renderizado muy grandes. En la imagen de la izquierda la PCB original, y a su derecha con la opción Circuito Impreso Rectangular habilitada.



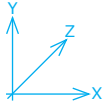
La opción Límites permite mostrar el 'Fresado' (capa 46 Miling) de la PCB en Eagle 3D, pero para que pueda mostrarlo se ha de copiar a la capa 20 (Dimensions). En las imágenes de ejemplo se dibujó un rectángulo con la herramienta Wire y dos círculos con la herramienta Circle. Para la imagen de la izquierda se seleccionó la opción Normal PCB a la hora de crear el archivo .pov, y en la segunda se optó por Ronda PCB, que como se puede ver sólo muestra el cobre.



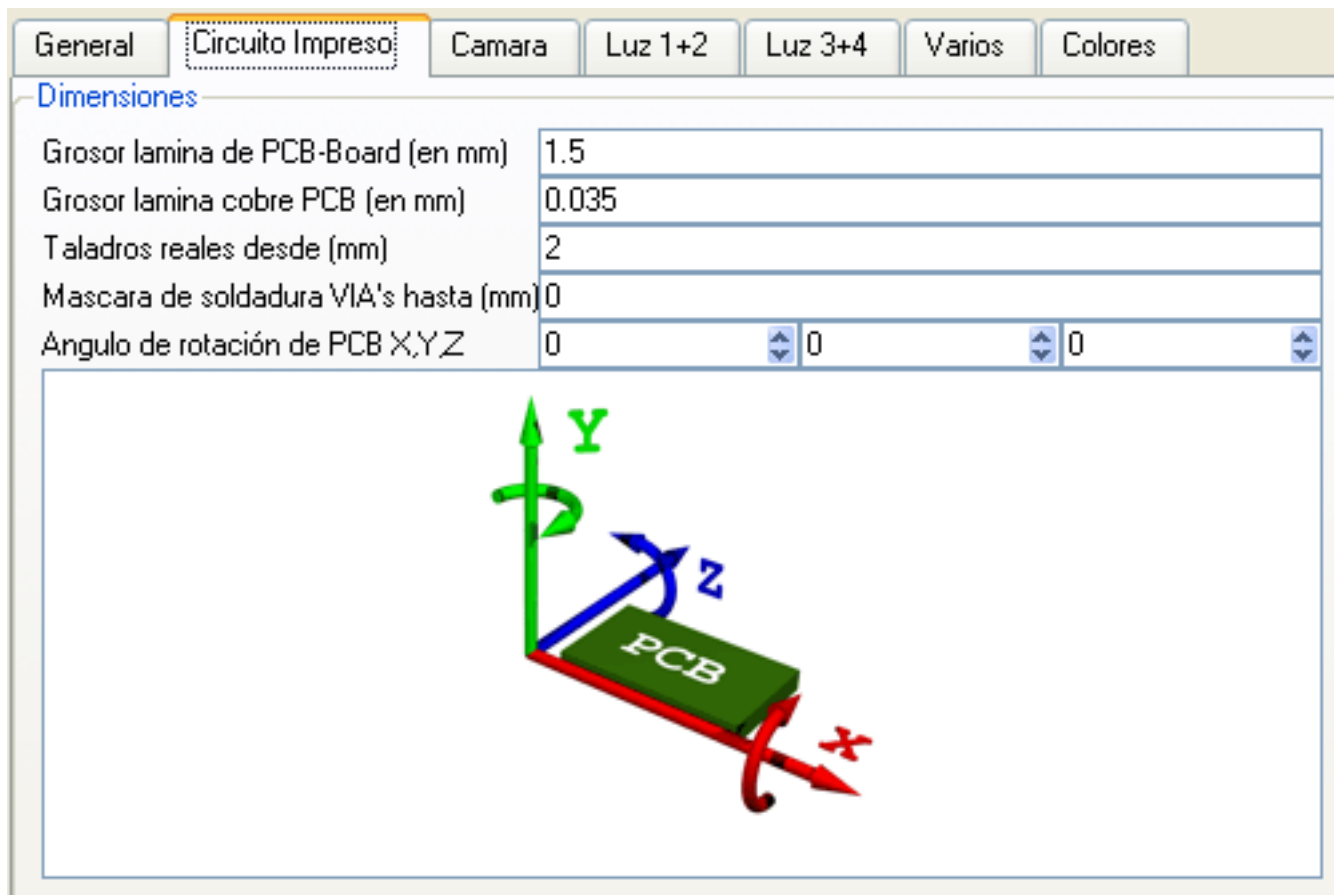
Para que la opción Usar asignación procedente de brdname.mpd funcione tiene que haberse generado el archivo .pov al menos una vez con la opción Componentes habilitada. Cuando al principio de manual se trabajó con el archivo *demo3.brd* recordaréis que el .pov resultante se guardaba en la carpeta que teníamos definida como destino: eagle3d/imgpov, y que por defecto tomaba el nombre del archivo .brd. Bien, pues además en la carpeta de origen de nuestro archivo .brd (projects/example/tutorial) se creaba con esta acción de renderizado un archivo con el mismo nombre pero con la extensión .mpd.

Este archivo .mpd contiene todos los datos de los componentes que Povray necesita para su representación, y que seleccionando esta casilla automáticamente formarán parte del archivo .pov. Podemos ver que la opción Componentes se deshabilita, cosa que por otra parte nos ahorrará unos cuantos clicks en más de una ocasión, pero por el contrario siempre se creará la imagen con los parámetros que contenga el archivo .mpd (logos...etc).

Con el uso de esta opción se puede llegar a ganar bastante tiempo.



La pestaña **Circuito Impreso** nos muestra las opciones siguientes:



Dimensiones:

Grosor de la placa en mm = 1.5

Grosor de la lámina de cobre en mm = 0.035

Taladro reales se muestran a partir de los 2 mm

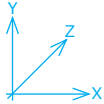
Mínimo tamaño en mm para las Vías = 0 mm

Establecer Ángulo de rotación en los ejes X, Y, Z

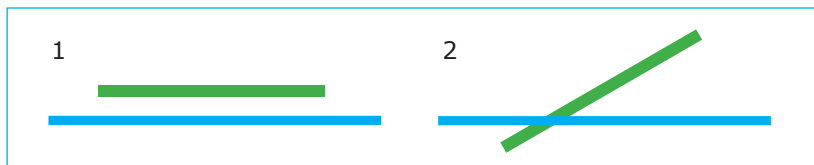
El Ángulo de rotación se basa en la conocida Regla de la mano izquierda y nos será útil saber un poco acerca de como funciona. Si cerramos los dedos de nuestra mano izquierda y estiramos el pulgar hacia un eje determinado, la dirección de nuestros dedos indicará los valores positivos en ese eje. Viendo la siguiente figura extraída del manual de Povray podemos apreciarlo mejor.



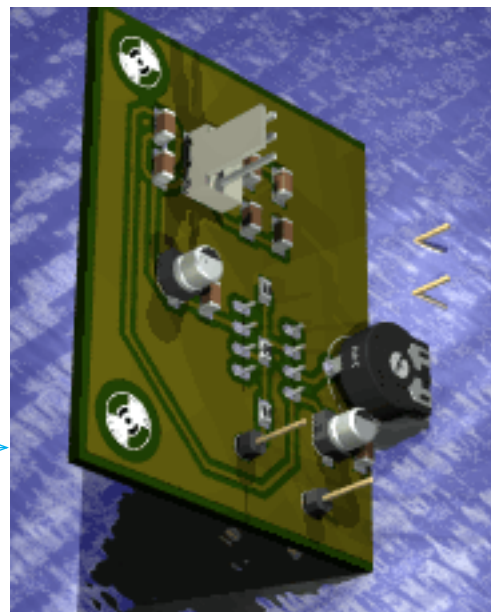
En este caso la mano agarra un flecha de color verde que está orientada hacia el eje X, los valores positivos concuerdan con el sentido de giro de la flecha roja. Las coordenadas en Povray parten de un hipotético punto en el centro del monitor, hacia la derecha tenemos valores positivos del eje X y hacia la izquierda los negativos. Hacia arriba positivos en el eje Y y hacia abajo los negativos. El eje Z parte de este punto hacia el 'interior' del monitor con valores positivos y hacia nosotros (el 'exterior') con valores negativos. Volveremos sobre el tema.



Con el último archivo que estamos usando a modo de ejemplo en este manual, y usando las opciones por defecto a la hora de crear el archivo .pov, obteníamos una PCB 'flotando' sobre un plano que evoca un mar. Es decir, ambas imágenes están desplazadas en el eje Y(1). ¿Pero que sucede si desplazamos ahora nuestra PCB en el eje X como muestra la figura de la derecha?(2)



En este caso nuestra PCB 'se hundirá' en ese mar y el resultado será este que vemos aquí (para mayor claridad, la imagen se desplazó en el eje Z)



En la pestaña **Cámara** del Panel de Control podemos ajustar su Posición, su Objetivo (hacia donde se dirige o punto de vista) y el Ángulo de extensión (a mayor valor mayor distancia, funciona de forma similar a un objetivo fotográfico).

General Circuito Impreso **Camara** Luz 1+2 Luz 3+4 Varios Colores

Posición

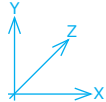
Posición de Camara:	X: 0	Y: 147	Z: -51
Objetivo Camera:	X: 0	Y: -2	Z: 0
Angulo lente cónica:	20		

Camera

Image Plane

Angle

Point of View



Las pestañas correspondientes a la configuración de las luces **Luz 1+2** y **Luz 3+4** nos permiten establecer ciertos parámetros comunes a todas ellas aunque establecidos de forma independiente para cada una.

Ya que son iguales, se detalla únicamente una.

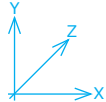
- Activar - Activa o desactiva esa luz en el proceso
- Color - Permite seleccionar un color para esa luz
- Posición - Posición de esa luz
- Objetivo - Punto de destino de esa luz
- Luz lunar - No se trata de una luz intensa sino de un punto de luz dirigida a un cierto objetivo
- Radio - Ángulo de apertura del cono de luz
- Caída - Este ángulo tiene que ser mayor que el Radio, de esta forma la intensidad decrece gradualmente
- Sin sombras - La fuente de luz no produce sombras, pero ilumina el entorno.

La opción Sin sombras disminuye el tiempo de renderizado aunque lógicamente nuestra escena quedará muy 'plana'. Puede ser buena idea no obstante trabajar con ella hasta tener el trabajo terminado y pasar luego a activarlo para generar la imagen final, así como renderizar a pequeños tamaños en Povray hasta que todo esté a nuestro gusto y entonces utilizar una resolución mayor.

The screenshot shows the 'Luz 1+2' tab selected in the control panel. It displays the configuration for two lights, Luz 1 and Luz 2. Both lights are set to 'White' color and have an intensity of 0.714606. Luz 1 is positioned at X: 17, Y: 26, Z: 11, with a target at X: 0, Y: 0, Z: 0. Luz 2 is positioned at X: -17, Y: 26, Z: 11, with a target at X: 0, Y: 0, Z: 0. Both lights have a radius of 10 and a fall-off of 12. The 'Luz lunar' and 'Sin sombras' options are unchecked for both.

Light	Activar	Color	Posición X	Posición Y	Posición Z	Objetivo X	Objetivo Y	Objetivo Z	Luz lunar	Radio	Caída	Sin sombras	Intensidad
Luz 1	<input checked="" type="checkbox"/>	White	17	26	11	0	0	0	<input type="checkbox"/>	10	12	<input type="checkbox"/>	0.714606
Luz 2	<input checked="" type="checkbox"/>	White	-17	26	11	0	0	0	<input type="checkbox"/>	10	12	<input type="checkbox"/>	0.714606

La configuración de luces que se tiene por defecto no es muy afortunada, se explicará y se entenderá (cuando nos adentremos en como construye las cosas Povray) mejor el tema, pero en esta parte básica del manual lo dejamos así.



Al comienzo de este manual, cuando se dieron los primeros pasos en la instalación, recordaréis que se generó un archivo llamado *3dconf.dat* con los datos básicos de directorio de trabajo, archivos de color y lenguaje..., y puede que os llamara la atención las series de números que aparecen en el mismo (página 9), que como vemos ahora son los números de capa usados para la Pantalla artificial (Serigrafía diría que es más correcto. En inglés Silk Screen) y para la Animación.

General	Circuito Impreso	Camara	Luz 1+2	Luz 3+4	Varios	Colores
Estas capas son usadas para Pantalla artificial						
Pista de paquete	21,22,51,52					
Pista de PCB	21,22,51,52					
Texto de paquete						
Texto de PCB	21,22,23,24,25,26,27,28					
Animation						
Eagle3D Layer	230					
Numero de Imágenes	20					
Creación de paquete de capas						
<input type="text"/>						

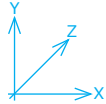
Se corresponden con las capas usadas en Eagle tal como se puede ver en la imagen y se detallan así:

- Pista de paquete - capas usadas para la serigrafía de encapsulados.
- Pista de PCB - capas usadas para la serigrafía de la PCB
- Texto de paquete - capas usadas para los textos de los encapsulados
- Texto de PCB - capas usadas para los textos de la PCB

La parte de Animation no será tratada en este manual básico y quedará para otro en el que se tratarán temas más avanzados.

La opción Creación de paquete de capas permite añadir nuevas capas a aquellas que trae establecidas por defecto. Se guardarán en el archivo *3dconf.dat*

21	tPlace
22	bPlace
23	tOrigins
24	bOrigins
25	tNames
26	bNames
27	tValues
28	bValues
51	tDocu
52	bDocu



La pestaña **Colores** podemos decir que está dividida en tres modos de configuración.

Configurar color - con un selector desplegable para seleccionar entre cinco opciones predefinidas

Selección manual - para una selección personalizada de los colores de nuestra imagen. Ha de estar seleccionada la opción Definido por usuario en el desplegable de la opción Configurar color para que tenga efecto.

Nuevo color - permite añadir un nuevo color a las diferentes opciones de la Selección manual.

Configurar color	
Seleccionar color para PCB	FR4 sin mascara de soldadura (Muecke)

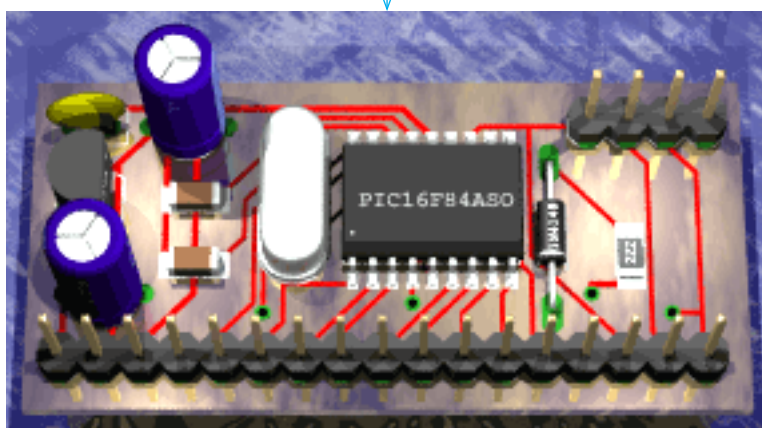
Selección manual	
Circuito Impreso	Verde oscuro
Pistas	Verde forestal
Pads y SMD	Plata
Taladros	Negro
Fondo	Gray50
Pantalla artificial	Blanco
Taladros metalizados	Verde forestal

Nuevo Color	
Rojo	0
Verde	0
Azul	0
Etiqueta	

Añadir Color

Para este ejemplo se ha buscado una combinación de Circuito impreso en color oro y las Pistas de color rojo dejando lo demás sin alterar.

El Panel de Control está íntimamente ligado al archivo *3d41.ulp*, se verá en el próximo paso del manual como localizar y modificar ciertos aspectos.

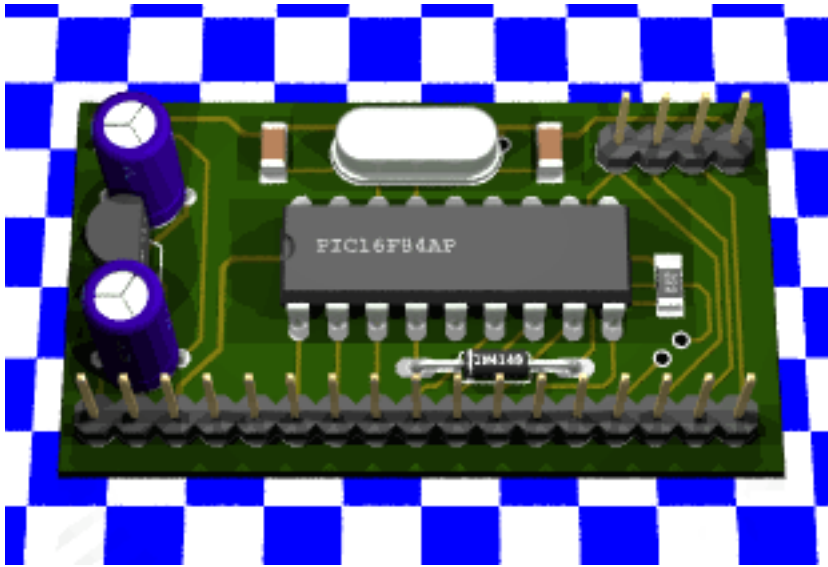


paso 4

TÉCNICAS (+) AVANZADAS

Empezaba este manual básico sobre Eagle 3D haciendo referencia a ULP(7) y terminamos el paso previo diciendo que podíamos modificar las propiedades por defecto del Panel de Control. Bien, ciertamente son muchas las propiedades y no solo las que hacen referencia al Panel de Control las que podemos modificar en el archivo *3d41.ulp* para que el archivo .pov resultante sea de nuestro agrado. En algunos casos más sencillos tan solo dándole el valor '1' o '0' a una variable, por ejemplo para que se muestre o no el Área de alrededor, o las Sombras o etcétera, etcétera porque todo el Panel de Control se puede establecer por defecto.

Y otros más complejos como pueden ser mostrar un Área de alrededor diferente al estilo marino que trae por defecto, y que puede verse de forma exagerada en la figura_4.1.

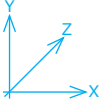


Está lejos de la intención de este manual básico adentrarnos en el ULP pero os invito a que, tras haber realizado una copia del archivo *3d41.ulp* procedáis a editarlo con Notepad++ o similar. Se pueden ver todas las partes que entran en juego a la hora de crear el archivo .pov. Tenemos que nuestro famoso archivo *3d41.ulp* lo que hace es generar un archivo con la extensión .pov y en un formato compatible con el Lenguaje Descriptor de Escenas de Povray.

En esta pequeña muestra del archivo *3d41.ulp* se puede ver la pantalla de inicio del Panel de Control. Podemos ver una de las ventajas de usar Notepad++ haciendo caso a cadsoft que dice que ULP es similar a C, se optó por este lenguaje en el Menú Lenguaje y de esta forma nos muestra el código en colores.

```
//Das hier sind die beim Start eingestellten Optionen
//Options seleccionadas al inicio del ULP
int opt_bau = 1; //Bauteile//Componentes
int opt_lei = 1; //Leiterbahnen//Vías
int opt_pad = 1; //Lötaugen//PAD's y SMD's
int opt_pol = 1; //Polygone//Polígonos
int opt_boh = 1; //Bohrungen echte Löcher in der Platine/Leiterbahnen//Agujeros en
                modo real
int opt_bohf = 0; //Bohrungen angedeutet durch schwarze Zylinder//Agujeros en modo
                rápido
int opt_obj = 1; //Wenn aktiviert werden weitere Informationen in die POVRay-Datei
                geschrieben -> weiterer Konverter
int opt_mup = 0; //Unbekannte Bauteile markieren//Componentes desconocidos
int opt_spn = 1; //Kurze Pins//Pines cortos
int opt_pcb = 1; //Platine einblenden//Muestra PCB
int opt_man = 0; //Bauteile manuell zuordnen//Usar modelos asignados
int opt_opcb = 0; //Platine wird nicht aus Polygonen erzeugt//Circuito impreso rectangular
int opt_bsd = 1; //Bestückungsdruck (Layer21/22)//Pantalla artificial o Serigrafía
int opt_amb = 1; //Sorgt für eine nicht allzu langweilige Umgebung//Área de alrededor
int opt_umpd = 0; //Manuelle Zuordnung aus vorhergehendem Lauf benutzen//Usar
                asignación procedente de brdname.mpd
int opt_lang = 0; //Sprachauswahl//Selección de idioma
int opt_dur = 0; //Durchbrüche erzeugen//Límites
//End//Fin
```

(7) En el Manual y en la Guía de Referencia de Eagle encontraréis información sobre ULP, siglas de **U**ser **L**anguage **P**rogramming (Lenguaje de Programación de Usuario).



No cambiaremos nada del archivo de momento, tan solo se trata de curiosear ya que se trata de un archivo altamente 'sensible' y de su buen funcionamiento depende todo. Además veremos que aparte de ciertas cuestiones repetitivas o que queremos establecer según gustos propios como posición de cámara, luces... lo más lógico es hacerlo desde Povray, en cuyo archivo .pov generado desde *3d41.ulp* veréis tantas concordancias. Con todo, si queréis verlo por vosotros mismos y para ir ahorrando en tiempos de render, probar a cambiar:

```
int opt_amb = 1;
por
int opt_amb = 0;
sin olvidarse del punto y coma (;) final
```

No se gana tiempo en la creación del archivo .pov sino a la hora de renderizar la imagen en Povray. Lo que se hace es darle un valor que posteriormente interroga:

```
#if(environment=on)
```

Y se actúa en consecuencia a ese valor. Además se establecen los parámetros para el dibujo de una superficie de un color o textura determinada que sirva como imagen de fondo a nuestra PCB. He aquí unos ejemplos de lo que encontramos ya al principio del ULP:

```
//Includes und Standardeinstellungen --> //Includes y ajustes estándar
//String für die Umgebung --> //Parámetros para el Área de alrededor (environment)
//used in the logo assignment --> //Asignación de Logotipo
//Variablendeklaration --> //Declaración de Variables
```

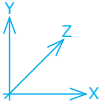
Si cargamos en Povray cualquiera de los ejemplos que se han visto anteriormente podemos ver que todo esto está también en el archivo .pov generado.

Vamos a ir un paso más allá con la intención de que quede más claro el proceso. Por seguridad se hizo una copia del archivo *3d41.ulp* en otra carpeta para restaurarlo después. Y estos son los pasos que se siguieron:

- se cambió la variable `opt_amb` a 0 --> `int opt_amb = 0;` (Área de alrededor)
- se guardó el archivo *3d41.ulp*
- se abrió el archivo *demo3.brd* en Eagle y se ejecutó Eagle 3D (ULP)
- se confirmó que la opción 'Área de alrededor' del Panel de Control estaba deshabilitada
- se creó el archivo .pov
- se renderizó la imagen en Povray para confirmar la ausencia de 'un entorno marino'
- se buscó en el archivo .pov la declaración `#declare environment = off;` y
- se cambió su valor a `#declare environment = on;`
- por último se procedió a renderizar de nuevo la imagen

Como se puede ver, a pesar de haber sido creada con la opción de Área de alrededor deshabilitada, es posible su cambio posterior en Povray. Y este es un motivo más para no tocar *3d41.ulp*.

La invitación es ahora a un primer contacto con el Lenguaje Descriptor de Escenas de Povray, tal como se hizo en el caso del archivo .ulp, naveguemos por sus líneas, en las que se pueden apreciar en bloques separados y comunes a todos los archivos .pov la definición de la cámara (una única cámara por escena), la descripción de las luces (puede haber múltiples y diferentes) y el objeto u objetos que conforman la escena.



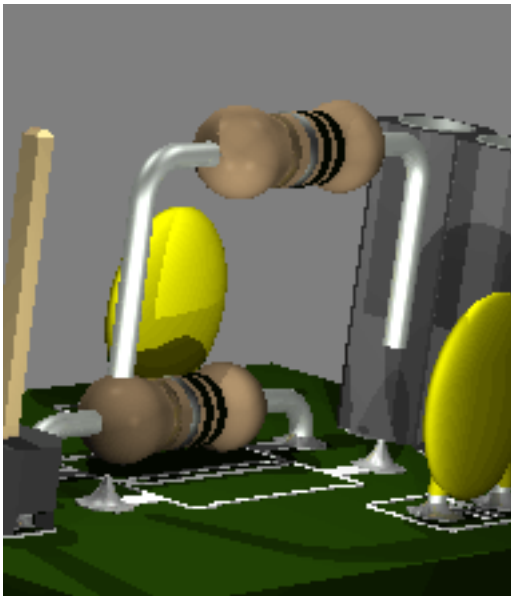
Copiamos las tres líneas de la macro al archivo *user.inc* y lo cambiamos así:

```
#macro BL_1X2()
object{BL_GRND(2,1)}
#end
```

Guardamos por fin los archivos *3dusrpac.dat* y *user.inc* si no lo hicimos antes.

En este caso, ni los componente existían en Eagle 3D ni los encapsulados en Eagle, pero dado que los uso con frecuencia en mis proyectos procedí a crear los correspondientes encapsulados que faltaban en la librería *con-1sta.lbr* de Eagle y a copiar y modificar sus respectivas macros en el archivo *user.inc* y asociaciones en *3dusrpac.dat*.

Hasta ahora no hemos entrado en la creación de componentes en Eagle 3D, y no lo haremos en este manual básico si no en el avanzado porque hay una serie de cuestiones que se deben conocer a priori y unas reglas a la hora de crearlos. De momento esto es todo en cuanto a las macros se refiere... pero volveremos sobre ellas porque pueden llegar a ser muy diferentes unas de otras. Si recordamos en la página 18 se desplazó un condensador en el eje Y, y en la página 25 se veía la diferencia entre 'pines cortos' en un condensador y una resistencia, que en el caso del condensador no existía tal diferencia debido precisamente a como fue definido en su macro. Ahora veamos que sucede cuando tenemos esa resistencia con 'pines cortos' y la desplazamos en su eje Y:

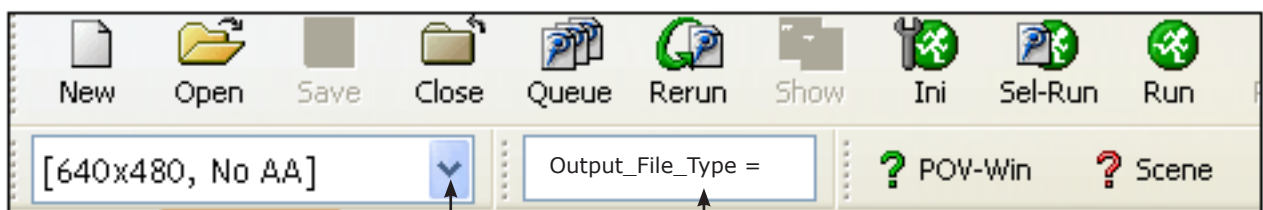


Bien, sucede que por defecto el largo para los 'pines cortos' es de 2.5 y si se supera ese valor, a diferencia del condensador, la resistencia abandona su conexión.

Tenía la intención de dedicar un [paso](#) exclusivamente a Visu 3D en esta parte básica pero está dando errores con ciertas macros de Eagle 3D que en versiones anteriores no recuerdo que existieran. Volveré a tratarlo más adelante, en la parte avanzada.

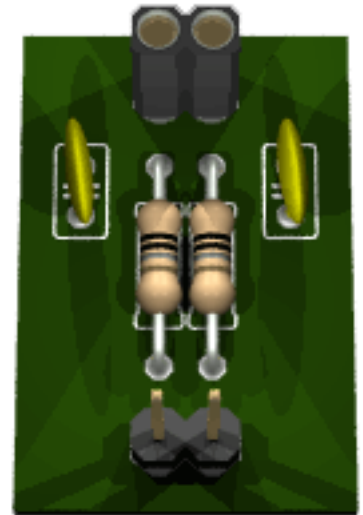
Ya para terminar, si queremos establecer otro formato de imagen diferente de .bmp podemos hacerlo directamente en la Línea de comandos de Povray, escribiendo: **Output_File_Type =**

- S - BMP
- N - PNG
- C - compressed TGA
- T - uncompressed TGA

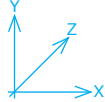


1 - Seleccionar Resolución

2 - Archivo de imagen



los próximos pasos
Eagle 3D Avanzado



Hemos visto lo básico pero quedan bastantes cosas por ver. En la parte avanzada se tratará no solamente una ampliación a ciertas técnicas aquí ya descritas sino muchas otras como la Animación, Creación de componentes a partir de su hoja de datos (datasheet), trabajar con Módulos (varias PCBs a la vez, como podría ser el conocido caso de una Placa base de ordenador y las correspondientes tarjetas en sus slots), un conocimiento mayor de Povray y su gestión de luces tan potente. Veremos también los archivos de configuración que hacen nuestro trabajo más productivo y, como no, volveremos una y otra vez sobre las macros.

Los programas usados en este manual y sus respectivas webs son:

Eagle 4.16r2 versión LITE -- <http://www.cadsoft.de/freeware.htm>

Eagle 3D 1.05 -- [www.matwei.de/doku.php?id=en:eagle3d:eagle3d\(2\)](http://www.matwei.de/doku.php?id=en:eagle3d:eagle3d(2))

Povray 3.6 -- www.povray.org

Notepad++ -- <http://notepad-plus.sourceforge.net/es/site.htm>

ForceVision -- <http://www.forcevision.net/>

Podemos obtener más información acerca de ULP en la web de Eagle:

Manual Eagle 4.1 -- <ftp://ftp.cadsoft.de/eagle/program/4.16r2/manual-eng.pdf>

Referencia Eagle 4.1 -- ftp://ftp.cadsoft.de/eagle/userfiles/doc/eagle416r2_help_en.pdf

Webs que considero interesantes para el aprendizaje de Eagle 3D:

<http://www.matwei.de/doku.php?id=en:eagle3d:eagle3d> - Su creador, **Matthias Weißer**.

<http://perso.numericable.fr/pboucheny/eagle3d/index.htm> - Web en francés (actualmente en proceso de traducción al inglés) con extensa y elaborada información. Lo único que se de su creador es que aparentemente se llama Philippe.

<http://felixchenier.homelinux.com/doku.php?id=pcb:eagle3dnewpart> - Mini-tutorial to design new parts in Eagle3D. Su autor es Félix Chénier. No siempre funciona el enlace

Y, por supuesto:

www.povray.org - Página de Povray

Los programas pueden tener restricciones en cuanto a su uso. Por favor dirigirse a sus respectivas webs para una mayor información

Este manual se terminó el día 16 de febrero de 2008, todas las versiones son las actuales y todos los enlaces funcionan a día de hoy.

Su uso es libre a nivel particular. Para añadir el documento a otra web que no sea la propia del autor, se agradece la notificación por correo a: basic3d@eagle.pcbpics.es y a esta misma dirección también agradezco se me envíe cualquier tipo de error que podáis encontrar.

Gracias a quienes me dieron los ánimos necesarios para terminarlo y a quienes se mostraron interesados en que se hiciera realidad. Y en especial a mis amigos y compañeros de:

RaDiKaLDeSIG - <http://radikaldesig.blogspot.com/>

Foro Todopic - <http://www.todopic.net/foros/>