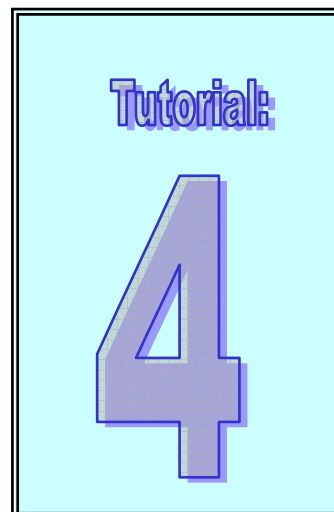


Vistas tridimensionales (3D) de circuitos electrónicos utilizando la herramienta Proteus^(*).



Fº Javier Alexandre.

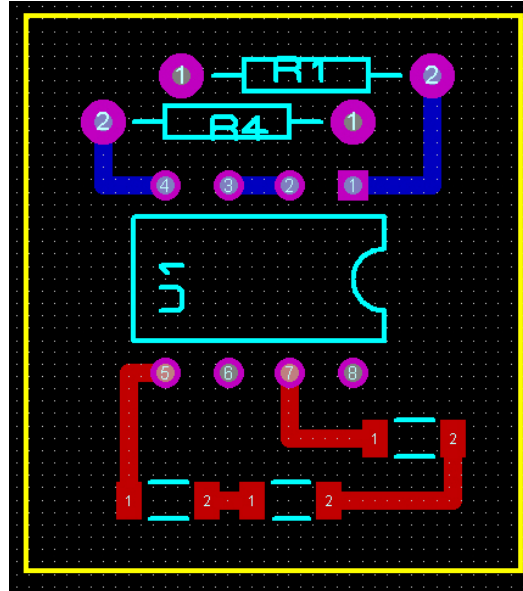
1.- Introducción.

La mayoría de los diseñadores de circuitos electrónicos están acostumbrados a realizar su trabajo en dos dimensiones. Posiblemente esta deformación viene motivada porque, históricamente, cuando se realizaba un circuito impreso, el problema consistía en trazar una serie de pistas en un plano que servían para unir los distintos pines de cada uno de los componentes electrónicos que se iban a utilizar en el diseño. La tercera dimensión sólo se utilizaba cuando nuestra placa podía componerse de más de una capa de cobre (típicamente la capa superior e inferior y en placas más complejas, capas intermedias).

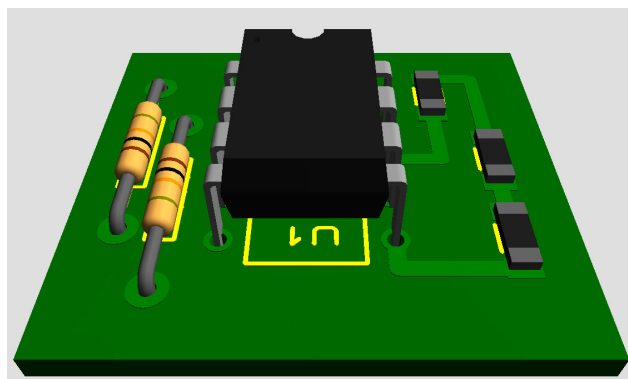
Por este motivo, cuando un ingeniero utiliza un determinado componente electrónico en un diseño, usualmente, sólo piensa en él considerando su contorno sobre la placa (layout) y sus diferentes pines (en el caso de componentes convencionales) o paths (en el caso de elementos SMD).

(*) La posibilidad de obtener y manejar vistas tridimensionales sólo está disponible en las versiones de Ares nivel 2 o superiores.

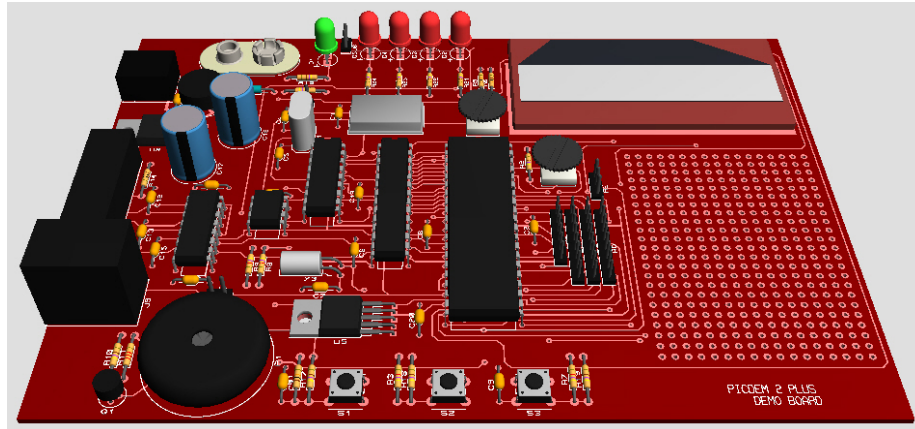
A continuación podemos ver una imagen donde se representa un diseño de una placa muy sencilla, compuesta por componentes tanto convencionales como smd y donde podemos observar que toda la información es "plana".



Sin embargo, en muchas ocasiones puede resultar extremadamente útil disponer de la información tridimensional de nuestro trabajo. Bien porque nos interese conocer la altura real de nuestro diseño para comprobar si va a ser posible utilizarlo en una determinada envolvente, o bien con fines estético-comerciales para adornar la presentación de nuestro trabajo al posible cliente final del producto, a quien la visión anterior no le dice nada. Veamos, como ejemplo, el anterior circuito en una vista 3D.



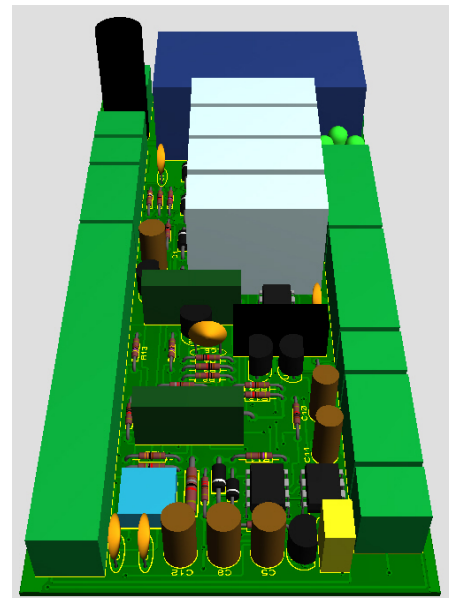
La diferencia es obvia. Además, en función de nuestras necesidades y del tiempo que estemos dispuestos a emplear, el resultado puede ser tan espectacular como el que se muestra en la imagen siguiente.



Precisamente, porque no siempre nuestras necesidades son las mismas, hemos dividido este tutorial en tres partes diferentes.

En un primer apartado describiremos cómo podemos obtener, visualizar y guardar, utilizando Proteus, una vista tridimensional de nuestro diseño sin entrar en más detalles sobre la calidad obtenida en nuestro trabajo.

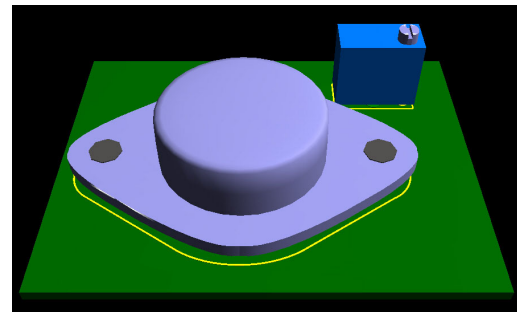
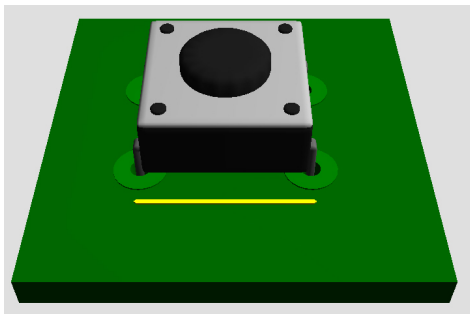
La clave de la calidad obtenida en nuestra vista tridimensional pasa por la definición de cada componente utilizado en nuestro circuito. Proteus dispone de unas sencillas herramientas para definir la imagen tridimensional de cada componente. En este caso la imagen obtenida no será de gran calidad pero sí suficiente como para conocer la "altura" de nuestro diseño. Un ejemplo puede verse en la imagen de la derecha.



Como podemos comprobar, con muy poco trabajo, es fácil hacerse una idea de la altura de los componentes utilizados y saber cuáles de ellos son los que más sobresalen y, por lo tanto, conocer la altura máxima de nuestro diseño.

El segundo apartado de este tutorial nos presentará las herramientas que Proteus nos ofrece para definir la imagen tridimensional de nuestros componentes.

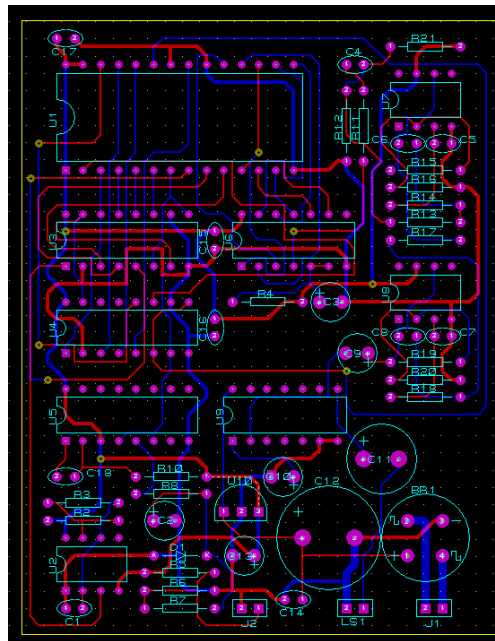
En el caso de que necesitemos una mayor calidad, Proteus permite asociar un fichero con la imagen tridimensional del componente al modelo de dicho componente. El fichero debe cumplir con el formato .3ds ampliamente utilizado por las principales herramientas de diseño tridimensional presentes en el mercado. En las siguientes imágenes podemos comprobar un ejemplo de varios componentes realizados utilizando ficheros en formato .3ds.



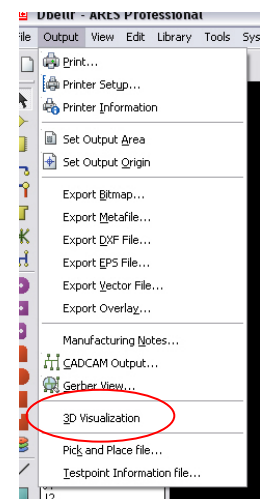
El tercer apartado de este tutorial está dedicado a la forma en que podemos utilizar ficheros en formato .3ds para definir la imagen tridimensional de los componentes utilizados en nuestro diseño.

2.-Obtención de vistas tridimensionales de nuestro proyecto.

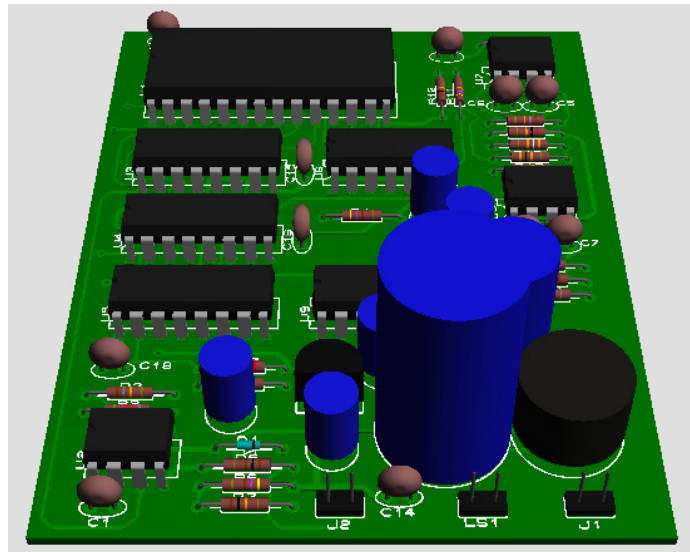
El procedimiento para obtener la imagen tridimensional de nuestro circuito es una tarea muy sencilla en Proteus. Supongamos que partimos del siguiente diseño (archivo dbellr.lyt situado en el directorio Schematic & PCB layout de la carpeta de ejemplos suministrada con Proteus).



Como se puede observar es un diseño nada especial y muy parecido a cualquiera de los trabajos habituales de un diseñador de circuitos electrónicos. Este podría ser el final del trabajo en la mayoría de los casos. Pero dar un paso más, y obtener una vista tridimensional de nuestro diseño es una tarea muy simple y que no nos llevará mucho trabajo en Proteus. Simplemente debemos invocar la opción de menú Output->3D visualization, como se muestra en la imagen de la derecha, y obtendremos la vista tridimensional de nuestro diseño que se ve



en la imagen siguiente.

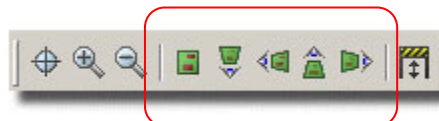


Evidentemente, el resultado no puede ser más satisfactorio para el tiempo invertido en obtenerlo.

En la nueva ventana obtenida, Proteus nos facilita una serie de controles, situados en la zona inferior izquierda de nuestra pantalla, útiles para navegar por nuestra imagen.



Como se puede observar la barra con los controles está dividida en tres zonas, separadas por un línea vertical. El segundo apartado, formado por cinco controles, sirve para seleccionar de forma rápida entre cinco vistas pre-configuradas. Vista desde arriba, vista desde el frente, vista desde el lateral izquierdo, vista desde atrás y vista desde el lateral derecho.



También podemos obtener estas vistas pre-configuradas desde el menú View, o con las teclas F8 a F12.

Otra vista muy útil, la podemos obtener desde el menú View, usando la opción Auto Spin. Con ello lograremos que de forma automática Proteus nos presente una vista tridimensional girando continuamente, lo que nos permite ver el resultado de nuestro diseño por todos los lados con un efecto realmente profesional para su utilización en tareas comerciales.

En el primer apartado de los controles, podemos encontrar también las herramientas que nos permiten modificar el zoom para acercar nuestro diseño y ver los detalles. Siguiendo los estándares habituales en Windows estos botones tienen la forma de una lupa con un + o un - según deseemos acercar o alejar nuestro diseño.



También se puede obtener este mismo efecto con la rueda central del ratón, con las teclas F6 y F7 o desde el menú View.

Se puede modificar el ángulo de situación de la cámara desde 30° a 60° con el menú Setting->Camera.

Aunque las opciones vistas hasta ahora son útiles en la mayoría de los casos, cuando realmente se disfruta de la potencia de las capacidades de renderización de Proteus es cuando utilizamos la opción de navegación. Esta opción la podemos invocar desde el primer control de la barra de controles (una especie de diana), desde el menú View->Navigate o pulsando la tecla F5.



Cuando comenzamos la navegación el cursor se sitúa en el centro de nuestro diseño. A partir de ahí con el ratón podemos empezar a navegar

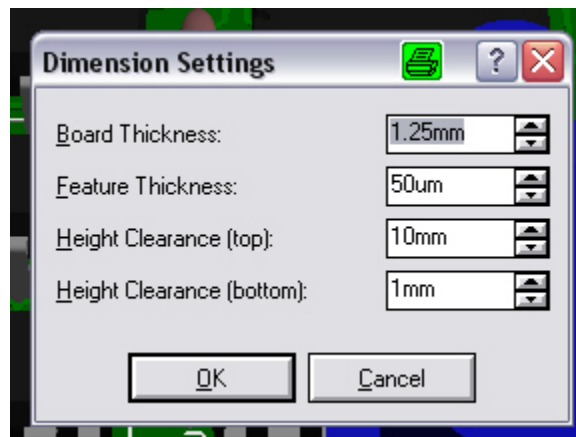
por nuestro diseño. Moviendo el ratón podemos "sobrevolar" sobre nuestro diseño. Pulsando sobre el botón izquierdo del ratón "soltaremos" o volveremos a iniciar nuestro vuelo.

Si movemos el ratón mientras mantenemos pulsado el botón izquierdo navegaremos en los tres ejes posibles de movimiento.

En cualquier momento podemos suspender nuestra navegación pulsando sobre el botón derecho del ratón.

Una vez vistas las opciones de navegación, pasaremos ahora a conocer las posibilidades que nos brinda Proteus para particularizar la vista de nuestro circuito a nuestro gusto personal. Toda esta particularización se realiza desde el menú Settings.

En la opción de menú Settings->Dimensions, obtenemos una ventana donde podemos configurar diversos parámetros relativos a la forma de nuestro PCB.



En la opción Board Thickness, seleccionamos el grosor de nuestra tarjeta. En la opción Feature Thickness, seleccionamos el ancho del relieve con el que se representarán las pistas y gráficos sobre la tarjeta.

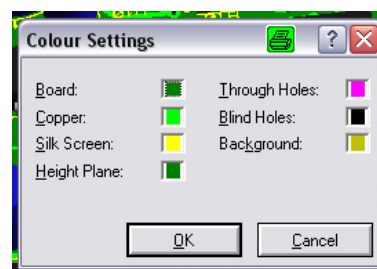
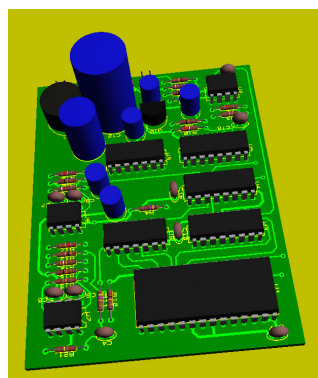
Proteus nos permite controlar también los colores de la representación tridimensional de la tarjeta de nuestro diseño para hacer que sea lo más similar posible a la realidad.

En el menú Settings->Colours, obtenemos una ventana donde podemos configurar diversos parámetros referentes a los colores de nuestra representación tridimensional.



En la casilla Board, seleccionamos el color que se utilizará para la base de nuestra placa. En la casilla Cooper, el color con el que se representarán las pistas de cobre. En la casilla Silk Screen, el color con el que se dibujarán las máscaras y diseños gráficos que aparezcan en la tarjeta. En la casilla Through Holes, el color para los taladros que traspasan la placa. En la casilla Blind Holes, el color para los taladros que no traspasen la placa de parte a parte. Por último, en la casilla Background, seleccionaremos el color de fondo.

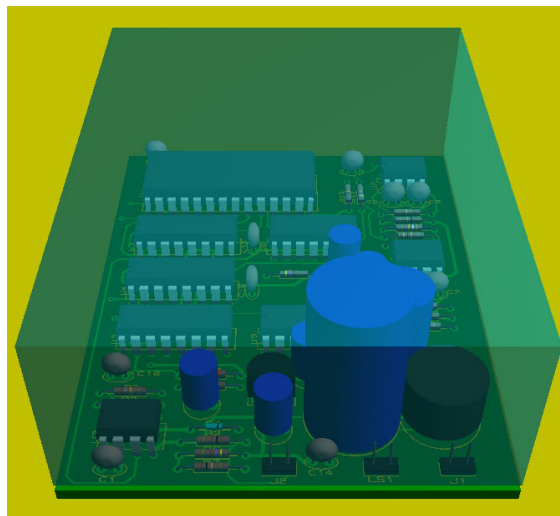
En la imagen siguiente se recoge la vista tridimensional de nuestro circuito ejemplo con la configuración de colores cambiada y el resultado. Hemos seleccionado unos colores especialmente llamativos, para que se vea más claro los efectos de cada parámetro sobre el resultado final.



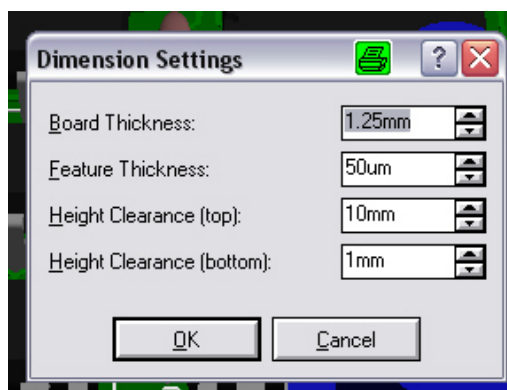
Proteus también nos permite simular la apariencia de la envolvente que servirá para contener nuestro diseño. Para ello podemos utilizar el control situado en la tercera división de nuestra caja de controles o la opción de menú View->Height Bounds.



En la siguiente imagen se ve un ejemplo del resultado.



En la opción de menú Settings->Dimensions, obtenemos una ventana donde podemos configurar la altura de la "caja" por encima de nuestra placa (Height clearance top) y por debajo (Height clearance bottom).



Y en el menú Settings->Colours, obtenemos una ventana donde podemos configurar el color de nuestra caja (Height Plane).



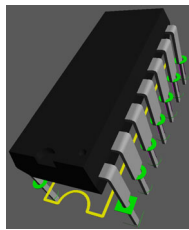
Por último, Proteus nos permite guardar el resultado de nuestra representación tridimensional en un fichero formato *.3ds (un estándar entre las herramientas de diseño 3D), utilizando la opción del menú File->Export 3ds. También podemos guardar el resultado en un fichero formato *.DXF (un estándar entre las herramientas de diseño 2D), utilizando la opción de menú File->Export dxf.

Finalmente podemos imprimir el resultado de nuestro trabajo con la opción de menú File->Print.

3.-Visualizaciones tridimensionales.

En el apartado anterior hemos visto lo fácil que puede ser obtener una vista tridimensional de nuestro diseño y las grandes posibilidades que nos brinda Proteus de trabajo con ella. Sin embargo la calidad del resultado final de nuestro esfuerzo viene determinada por la definición de la imagen tridimensional que hayamos asociado a cada componente utilizado en nuestro circuito.

La librería de encapsulados que se entrega con Proteus incluye imágenes tridimensionales para la mayoría de sus elementos. Por ejemplo, si definimos que un determinado componente va tener un encapsulado estándar tipo DIL14, Proteus, por defecto nos ofrece la siguiente imagen tridimensional asociada con él.

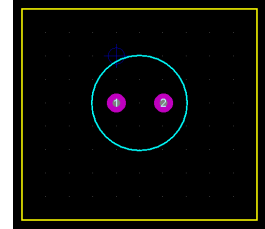


En muchos casos, las imágenes tridimensionales serán suficientes para nuestro trabajo. Pero cuando necesitemos crear un modelo nuevo para un componente no incluido en la biblioteca estándar, será necesario que nosotros definamos la imagen tridimensional que se le debe asociar. Para realizar esta tarea Proteus nos ofrece unas herramientas muy sencillas de definición de la vista tridimensional.

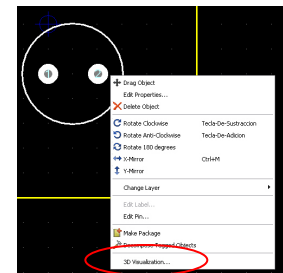
La filosofía de trabajo consiste en asociar al elemento en cuestión uno de los cuatro modelos tridimensionales pre-definidos en Proteus (extruded, none, cuboid, axialcylinder). A continuación, podemos, mediante la utilización de los parámetros asociados a ese modelo, ajustarlo a nuestras necesidades.

El primer modelo -extruded- es posiblemente el más sencillo pero a la vez el más utilizado. Como su nombre indica, únicamente realizamos una extrusión de la forma de nuestro componente realizada en dos dimensiones (layout).

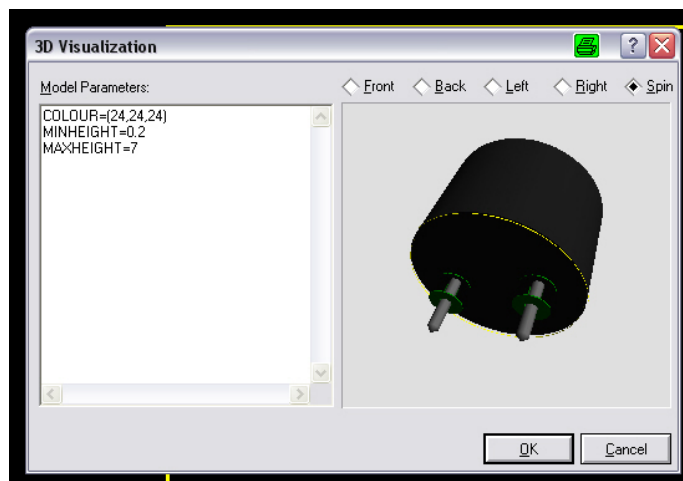
Veamos un ejemplo que nos ayude a entender la filosofía de trabajo y el funcionamiento de este primer modelo. En primer lugar coloquemos un encapsulado tipo CAP-RAD20 en nuestro diseño. La imagen debe ser similar a la siguiente.



A continuación, situemos el ratón encima y pulsemos el botón derecho. En el menú contextual que nos aparece, debemos seleccionar la opción "3D visualization".

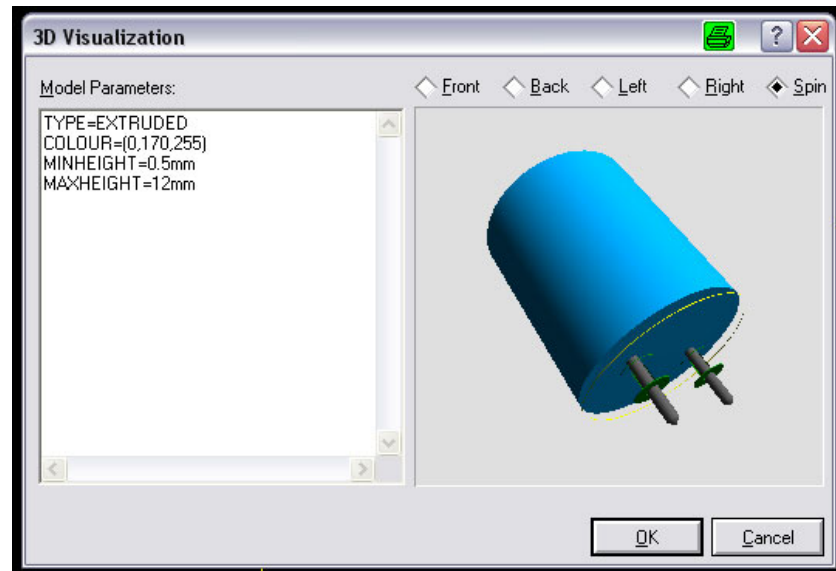


Nos aparece una ventana dividida en dos apartados. En la zona derecha una vista de lo que será nuestra imagen en tres dimensiones. En la parte izquierda un editor de texto donde podremos definir los parámetros de nuestro modelo de visualización tridimensional.



El modelo por defecto es, precisamente, el extruded. Por eso vemos que la imagen es, simplemente un cubo (el resultado de extruir un circulo en el eje z). Proteus, entiende que cuando no se le indica el modelo

que deseamos utilizar, usará el extruded. Sin embargo, con una intención pedagógica, vamos a modificar en la ventana los parámetros de forma que aparezcan todas las posibilidades completas de este modelo. Escriba lo necesario hasta obtener el resultado que aparece en la siguiente imagen.

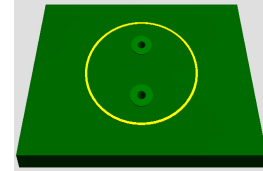
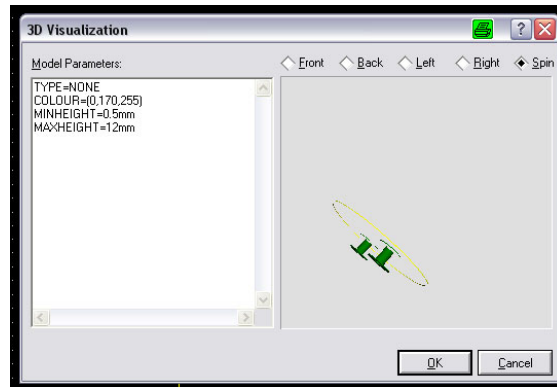


La primera línea (parámetro TYPE) indica cuál de los cuatro modelos disponibles estamos utilizando. Recuerde que si no indica ningún modelo, por defecto Proteus utilizará el modelo extruded. La segunda línea (parámetro COLOUR) indica el color utilizado para la extrusión en formato estándar RGB. El tercer parámetro (MINHEIGHT) indica la separación que existe entre la placa y nuestro modelo. Por último, el parámetro MAXHEIGHT indica la altura de nuestro componente.

Es evidente, que esta es una manera muy cómoda de obtener sencillos modelos tridimensionales asociados a nuestros componentes.

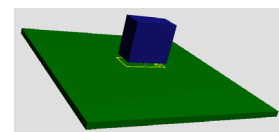
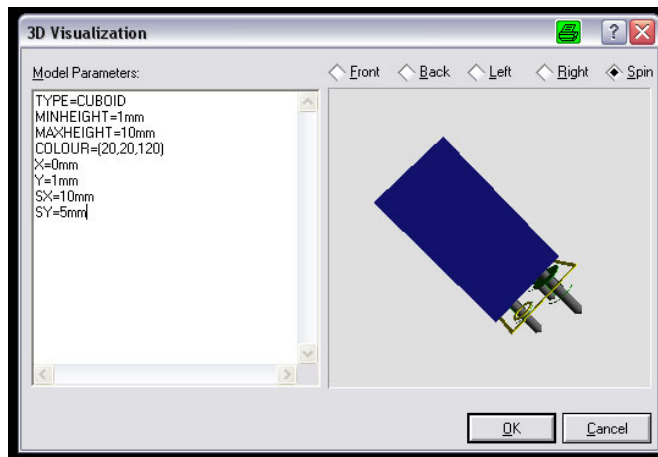
El segundo modelo disponible se llama NONE. Con este modelo le indicamos a Proteus que deseamos borrar el cuerpo de nuestro componente y representar únicamente los pines.

En la siguiente imagen se representa el resultado obtenido si en el modelo anterior sustituimos el valor EXTRUDED del parámetro TYPE por el valor NONE.



El tercer modelo disponible es el denominado Cuboid. Este modelo es adecuado para representar componentes con el cuerpo en forma de un cubo.

Para ver cómo funciona este modelo, coloque en su circuito un encapsulado tipo PRE-SQ2 de la librería estándar de Proteus y asígnele los parámetros según figuran en la imagen siguiente. El resultado que obtendremos también se puede observar en esta misma imagen.

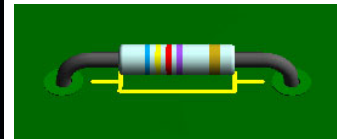
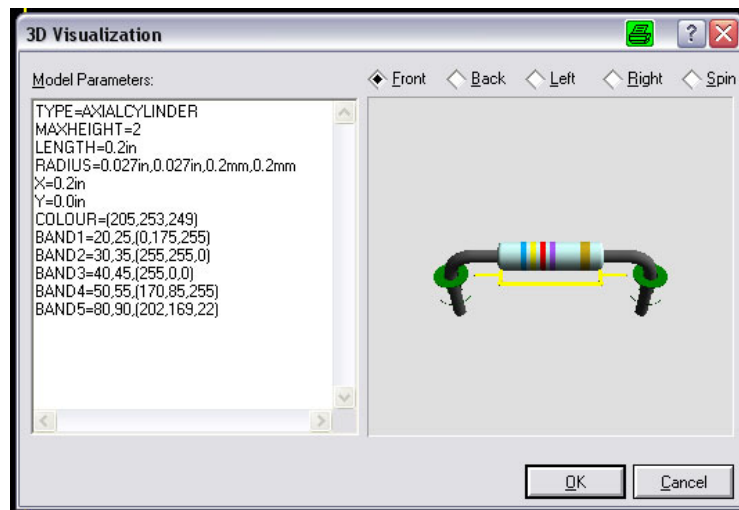


Los parámetros disponibles para este modelo son, además de los ya vistos para el caso del modelo extruded, los siguientes. El parámetro X la

distancia desde el origen del cuerpo al origen de la máscara del PCB en el eje X. El parámetro Y lo mismo respecto al eje Y. El parámetro SX define el ancho del cubo y el parámetro SY el ancho del cubo.

El último modelo disponible es el AXIAL-CYLINDER. Este modelo nos permite definir la vista tridimensional de componentes cilíndricos axiales. El caso más típico es una resistencia convencional.

Como siempre, explicaremos los parámetros disponibles en este modelo con un ejemplo. Sitúe en su circuito el encapsulado RES40 de la librería estándar de Proteus y asígnele los valores indicados en la siguiente figura a los diferentes parámetros disponibles. Vea el resultado en esta misma imagen.



El parámetro MAXHEIGHT define la altura del cuerpo cilíndrico con respecto al circuito impreso. El parámetro MINHEIGHT que no aparece utilizado en este ejemplo representa, como en los modelos anteriores la altura respecto al circuito impreso. El parámetro LENGTH especifica la longitud del cuerpo cilíndrico.

El parámetro RADIUS controla el radio del cuerpo cilíndrico. Se compone de cuatro valores (P1, P2, P3 y P4) separados por comas. El primero define el radio vertical del cilindro y el segundo el radio

horizontal. Con estos dos parámetros se permite la creación de cuerpos de forma elíptica.

El tercer parámetro define la profundidad del radio hacia el eje del cilindro y el cuarto parámetro define la profundidad del radio a lo largo del eje del cilindro. Con estos dos parámetros se posibilita la creación de cuerpos tipo torpedo.

Realmente es mucho más fácil de utilizar estos parámetros que explicarlos. Por eso, es especialmente útil en este caso ir haciendo pruebas variando los valores y observando el resultado obtenido.

Los parámetros X e Y tienen el mismo significado que el que ya vimos en el caso del modelo CUBIC.

El parámetro COLOUR sirve para definir el color del cuerpo cilíndrico en formato RGB.

A continuación viene una serie de parámetros BAND<n>, donde <n> puede tomar los valores de 1 a 9. Con estos parámetros podemos definir hasta nueve bandas de diferentes colores y anchos que se representarán en el cuerpo cilíndrico. Su utilización más común es la representación del código de colores utilizados habitualmente en los componentes electrónicos.

Cada parámetro BAND tiene el siguiente formato:

$BAND\langle n \rangle = P1, P2, (r,g,b).$

Donde P1 y P2 representan el comienzo y el final de la banda expresado en tantos por ciento respecto al total del cuerpo cilíndrico. Así en nuestro ejemplo la primera banda está definida como:

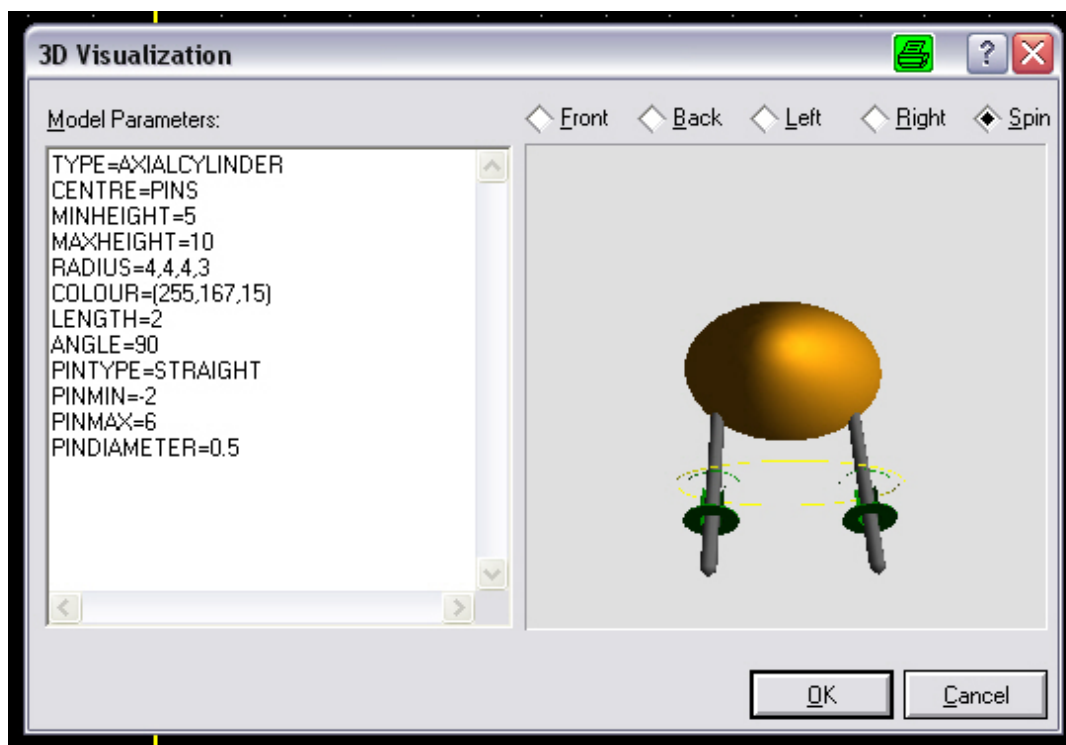
$BAND1 = 20, 25, (0, 175, 255)$

Lo que significa que la primera banda empieza en el 20% del cilindro contando desde la izquierda hasta el 25% y es de color azul claro.

También en nuestro ejemplo hemos querido simbolizar la banda que representa la tolerancia de nuestra resistencia más ancha que las otras cuatro y de color dorado. Para ello hemos usado el parámetro:

BAND5 = 80, 90, (202,169,22)

Por último vamos a presentar otro ejemplo de uso de este modelo para definir un condensador cerámico tipo lenteja.



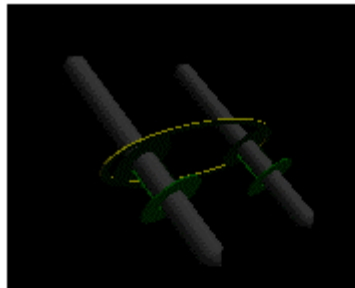
Observe la utilización del parámetro **CENTRE=PINS** para indicarle que en este caso el cuerpo sobresale respecto a los pines y no queda entre ellos como en el caso de las resistencias visto anteriormente.

Por último, aprovechando que en este ejemplo los cuatro últimos parámetros hacen referencia a los pines de nuestro condensador, vamos a

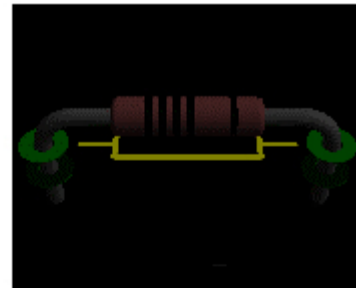
explicar los diferentes tipos de pines que podemos utilizar en la definición de la imagen tridimensional de nuestros componentes.

Existen siete tipos de pines disponibles: NONE, STRAIGHT, BENTWIRE, SMTZ, SMTJ, SMTB y CUBOID. En la figura siguiente vemos un ejemplo de los seis últimos (el primero sirve para indicar que no representaremos los pines).

Straight



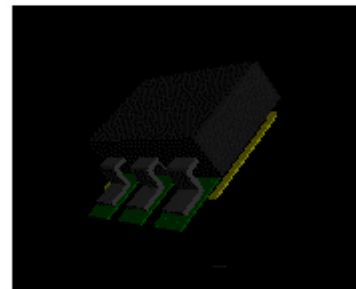
Bentwire



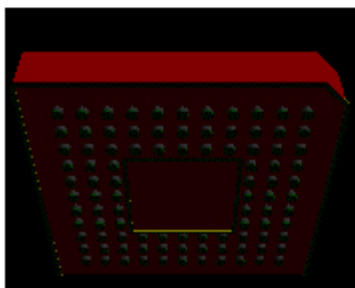
SMTJ



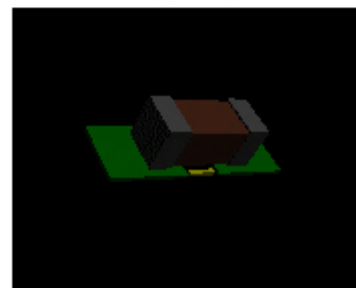
SMTZ



SMTB



CUBOID



Cuando definimos el layout de un componente, podemos definir que tipo de PAD vamos a utilizar para soldar el componente al PCB. Cada uno

de los tipos de pines disponibles está asociado por defecto a un determinado tipo de PAD.

Para los "pads" tipo ahujero pasante, el pin por defecto será el BENTWIRE, excepto si los pines están bajo el cuerpo del encapsulado o no hay un encapsulado asociado con este componente, en cuyo caso el pin por defecto será STRAIGHT.

Para los "pads" tipo smd, el pin por defecto defecto será el SMTZ, excepto si los pines están bajo el cuerpo del encapsulado o no hay un encapsulado asociado con este componente, en cuyo caso el pin por defecto será SMTJ.

Si deseamos aplicar un pin diferente al ofrecido por defecto el parámetro que debemos utilizar es:

PINTYPE=XXXXXX

El resto de parámetros utilizables para los pines son:

PINCOLOUR = define el color del pin.

PINMAX = máxima altura del pin.

PINMIN = separación del pin respecto al PCB. En algunos casos el valor puede ser negativo y en el caso de los encapsulados SMD debe ser 0.

PINDIAMETER = en el caso de los pines BENTWIRE y STRAIGHT define el diámetro del pin.

PINLENGTH = en el caso de los pines SMTJ y SMTZ, define la longitud de la base del pin.

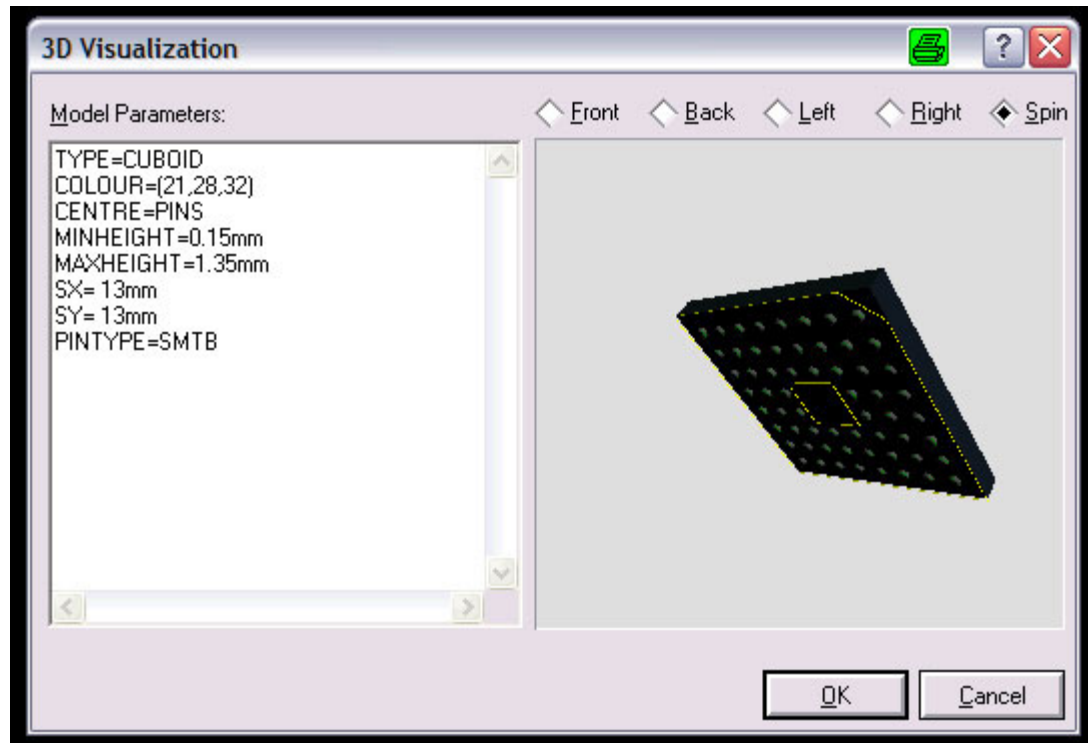
PINTHICKNESS = en el caso de los pines SMTJ y SMTZ, define el grosor del pin.

PINWIDTH = en el caso de los pines SMTJ y SMTZ, define el ancho del pin.

La mejor forma de conocer y aprender el uso de estos parámetros es mirar como están definidas las vistas tridimensionales de los

encapsulados incluidos en las librerías estándar de Proteus e ir haciendo pruebas.

Terminamos este apartado viendo otro ejemplo de definición de un encapsulado tipo BGA.



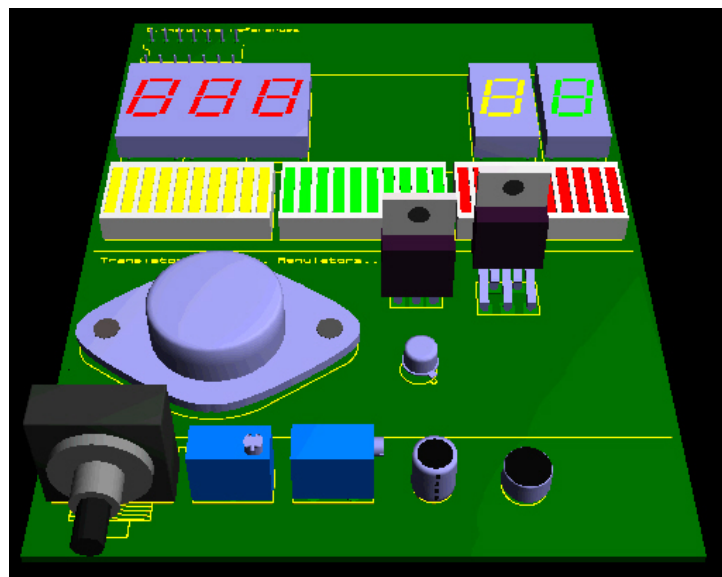
4.-Visualizaciones tridimensionales artísticas.

Para terminar este tutorial vamos a presentar la opción que nos ofrece Proteus para obtener un aspecto más artístico de nuestra vista tridimensional.

Proteus nos permite asociar al encapsulado una imagen realizada con herramientas de diseño 3D almacenadas según el formato estándar de dibujo *.3ds.

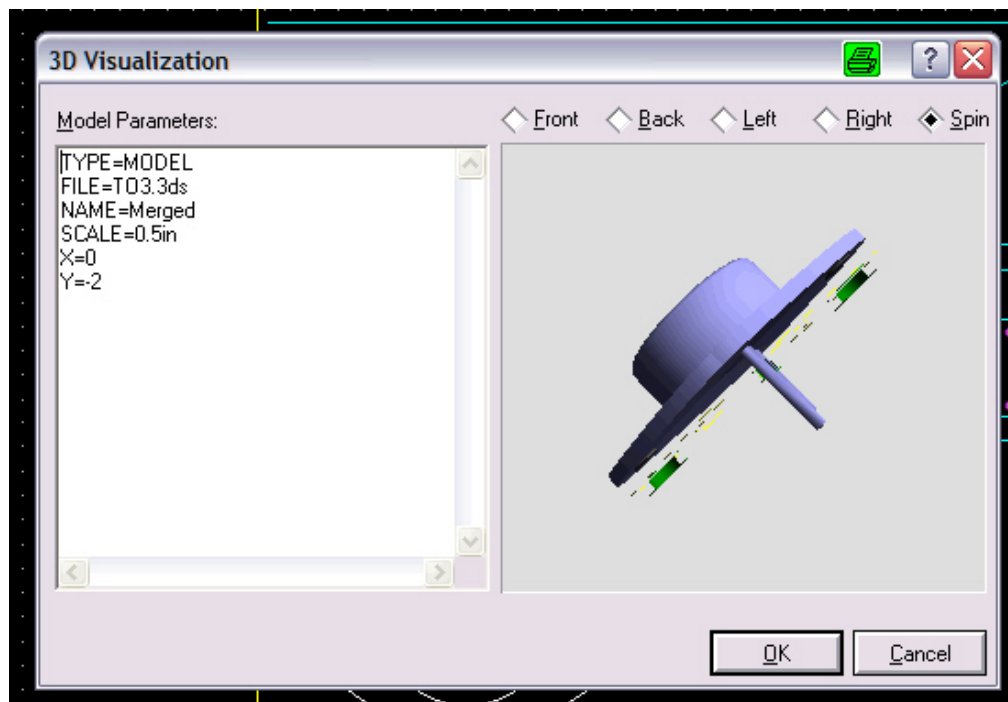
El siguiente ejemplo es una representación de un circuito que utiliza como vistas tridimensionales ficheros de formato *.3ds. Hemos tomado el ejemplo sacado de la siguiente página web:

http://perso.orange.fr/sonelec-musique/logiciels_proteus_lib.html



Rémy Mallard no solo ha hecho un magnífico trabajo y posibilita la utilización de estos modelos, sino que también ha escrito un interesante tutorial en inglés y francés, explicando como los ha diseñado.

Para asociar una determinada imagen tridimensional a un encapsulado debemos escribir los siguientes parámetros.



Como vemos definimos como tipo de modelo MODEL, que es la forma de indicarle que usaremos un fichero externo. El parámetro FILE nos indica el fichero donde se guarda el modelo y el parámetro NAME el nombre del modelo dentro del archivo.

El parámetro SCALE nos permite adaptar la imagen al tamaño de nuestro layout.

Es muy interesante consultar los modelos incluidos en Proteus y ver como los utiliza para aprender más sobre el uso de los modelos tridimensionales externos.