

# PROTEUS - ISIS MANUAL

F García Miranda

# ÍNDICE

Introducción.....	p.2
Acciones básicas.....	p.3
Generación de informes.....	p.6
Creación de componentes.....	p.8
Edición de componentes.....	p.12
Simulación	
Generadores.....	p.13
Gráficas.....	p.16
Instrumentos virtuales.....	p.18
Simulación de microcontroladores	
Ajustes previos.....	p.20
Compilación.....	p.21
Simulación.....	p.23
Depuración.....	p.24

# INTRODUCCIÓN

Actualmente existen en el mercado un gran número de paquetes de software que nos posibilitan la creación de esquemas de circuitos electrónicos e incluso el diseño de la PCB. Cada día es mayor el número de programas que nos ofrecen estas posibilidades y las opciones que nos ofrecen, pero cuando buscamos un programa que además nos permita simular esos mismos circuitos la cosa se complica. El campo de la simulación interactiva de circuitos electrónicos no está tan ampliamente cubierto como el de diseño de esquemas y PCBs. Podemos encontrar paquetes como OrCAD, que nos dan esa funcionalidad (simulación y diseño), pero tan sólo nos ofrecen la posibilidad de una simulación estática, por lo que pierde gran parte de la utilidad que puede venirnos a la mente al pensar en el concepto de "simulación de un circuito electrónico".

Ésta es la ventaja de Proteus, Proteus nos permite simular en tiempo real (dependiendo, claro está, de las posibilidades de nuestra máquina) todo tipo de circuitos electrónicos, pudiendo visualizar gráficas con los valores que obtengamos, comprobar valores de tensión e intensidad en cualquier parte del circuito y, además, permitirnos interactuar con todos los componentes del circuito susceptibles de ello. Además de esta posibilidad nos permite simular microprocesadores que podamos haber incluido en el circuito dándonos la posibilidad de programarlos (y compilar estos programas) e incluso depurar esos programas desde la misma aplicación. Todo ello incluyendo también un potente software de diseño de esquemas (Isis), y de PCBs (Ares), todo completamente integrado, por lo cual las posibilidades son inmensas, pudiendo hacer cosas impensables con la gran mayoría de paquetes de características similares en el mercado.

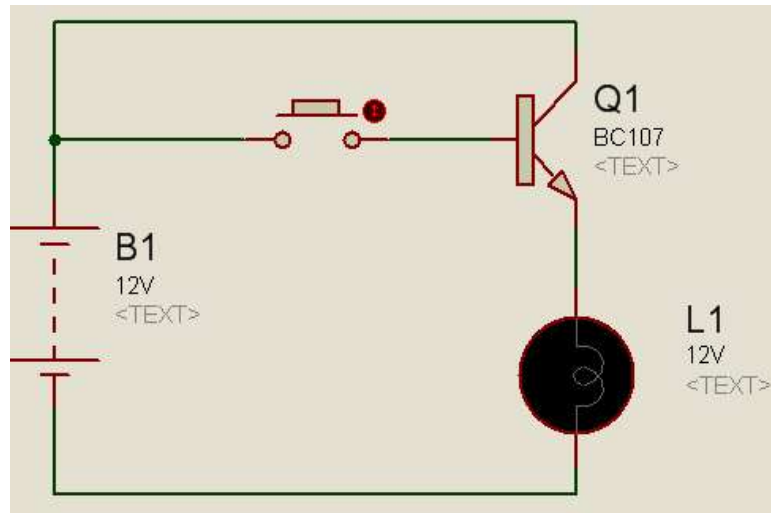
Ésta es la razón de la creación de éste manual, simplificar un poco el acceso a Proteus y proporcionar una muestra de su potencial, comenzando desde cero y acabando con el diseño y programación e incluso depurado de circuitos complejos.


El presente manual está basado en la versión 6.2 SP4 de Proteus, siendo a día de hoy la 6.3 la última versión, aunque por el momento es desaconsejable su uso debido a cierto número de errores que presenta, siendo la 6.2 la versión más estable.

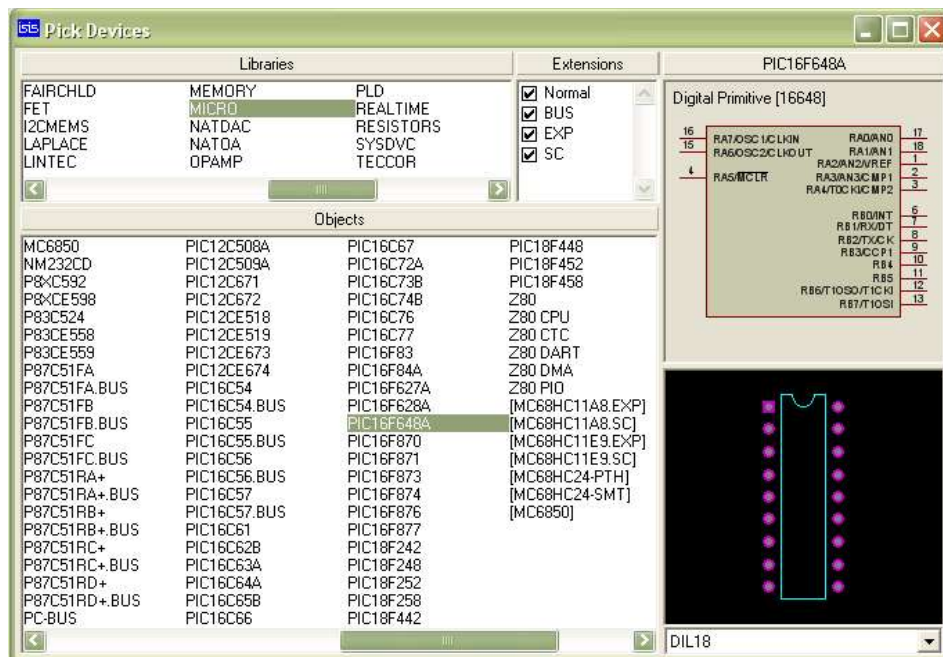
## ACCIONES BÁSICAS

Empezaremos creando un circuito sencillo para familiarizarnos con el entorno de Isis. A la izquierda del entorno de trabajo podemos ver una barra de herramientas que nos facilita el acceso a las funciones básicas necesarias en el trabajo con esta aplicación.

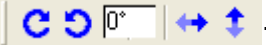
Empezaremos directamente realizando un pequeño circuito para ejemplificar esto:





El primer paso a realizar será seleccionar los componentes que necesitaremos en el proceso, para lo cual utilizaremos la herramienta componente, . Pulsando en ese botón veremos como a la derecha de la barra de herramientas se activará una lista (vacía en primera instancia) con el título de devices. Ahí es donde veremos el listado de componentes seleccionados para su inclusión en el esquema. Podemos observar como a la izquierda de este título hay dos botones **P** **L** **DEVICES**, pulsaremos sobre el botón 'P' y tendremos acceso a la ventana de selección de componentes:



En este cuadro de diálogo podemos seleccionar la librería en la que deseamos buscar y, dentro de esa librería el componente que queremos añadir a la lista. En la parte derecha de la ventana veremos una previsualización del componente en el esquema y la placa de circuito impreso respectivamente. Para añadir un componente a la lista haremos doble click sobre su nombre. Una vez hayamos añadido los componentes que necesitemos podemos cerrar la ventana y volver a la ventana de diseño. Para realizar el circuito del ejemplo necesitaremos los componentes LAMP, BATTERY y BUTTON de la librería ACTIVE y el componente BC107 de la librería BIPOLAR.

El siguiente paso será colocar los componentes en el espacio para ello destinado. Seleccionaremos el nombre del componente a colocar con el ratón y haremos click izquierdo en un espacio del esquema. Por cada click que hagamos colocaremos una nueva copia del componente. En la ventana de previsualización podremos comprobar cómo va a ser colocado el componente y rotarlo mediante los botones de la esquina superior izquierda : . Pero quizá podamos querer rotar o mover un componente después de colocarlo, esto se consigue seleccionándolo con un click derecho del ratón y arrastrándolo con el botón izquierdo (para moverlo), o manipulando los botones de rotación una vez seleccionado. Para deseleccionar un componente haremos click derecho en un espacio vacío.

Una vez colocados de forma correcta los componentes en el espacio de trabajo el siguiente paso lógico sería conectarlos entre sí mediante cables, , usando para ello la herramienta cable (wires). Para usarla solo tenemos que accionarla y hacer click izquierdo en cada uno de los pines que queramos conectar (es importante que el componente en cuestión no esté seleccionado o sólo conseguiremos editar sus propiedades), Isis se encarga automáticamente de seleccionar la mejor ruta a seguir para el trazado del cable. Aunque si no estamos satisfechos con el resultado podemos actuar sobre el cable trazado del mismo modo que lo haríamos sobre un componente. Existe una opción que nos permite eliminar de la lista de componentes aquellos componentes que todavía no hayamos utilizado en nuestro diseño. Es la opción Tidy del menu Edit. Podemos probar a incluir en la lista cualquier componente y después seleccionar la opción Tidy para ver como desaparece de la lista si no ha sido colocado antes.


Bien, ya tenemos nuestro primer circuito diseñado, ahora vamos a probar algo de simulación básica, por el momento nos limitaremos a comprobar si el circuito se comporta como debería comportarse (encendido de la lámpara al pulsar el botón). Para ello activaremos el modo de simulación pulsando el botón de inicio de simulación de la barra inferior :  o presionando F12. Para simular la acción sobre el pulsador del circuito haremos click sobre él una vez la simulación esté en marcha y comprobaremos su efecto. Para detener la simulación pulsaremos el botón correspondiente de la barra.


Si queremos editar las propiedades de un componente primero lo seleccionaremos haciendo click derecho sobre él, para después hacer click izquierdo. Ante nosotros aparecerá el cuadro de diálogo de propiedades de componente, que variará según el componente y mediante el cual podremos configurarlo a nuestro gusto. Elementos comunes a la mayoría de componentes son su referencia (R1, R2, C1...) y valor, así como también el footprint seleccionado para pasar el esquema a ARES, la aplicación de diseño de PCB. Si el componente tiene pines de alimentación que se encuentran ocultos podemos seleccionar a que red (net) queremos conectarlos mediante el botón "Power Pins", aunque en circunstancias normales los valores por defecto son adecuados. Para este ejemplo modificaremos los parámetros de la fuente de tensión accediendo a su valor de propiedades de la manera anteriormente indicada y modificando su tensión de salida

(voltage) probando a introducir en el respectivo campo el valor de 5, por ejemplo. Si volvemos a simular el circuito podremos comprobar cómo la “iluminación” de la bombilla es de un tono más oscuro. Podemos probar distintos valores para comprobar como se nos muestran las distintas intensidades de luminosidad de la bombilla. También podemos cambiar de la misma manera la tensión a la que funcionará la bombilla comprobando, por ejemplo, que con una tensión demasiado elevada de la bombilla y demasiado pequeña por parte de la fuente de tensión no podremos encender la bombilla.

Un detalle a tener en cuenta sobre la simulación de circuitos en Proteus es que no es necesario diseñar en él el circuito completo para poder simularlo, ya que podemos obviar partes, como pueden ser las alimentaciones de los circuitos integrados, los osciladores de los microcontroladores, incluso las salidas, ya que durante la simulación se nos muestra mediante un sistema de colores el estado lógico de cada salida o extremo de un componente.

## GENERACIÓN DE INFORMES


Uno de los informes principales que podemos realizar con ISIS y practicamente cualquier paquete de diseño de esquemas es el ERC (Electrical Rules Check), para comprobar si todo está correctamente en nuestro diseño. Este informe podemos realizarlo mediante el menú *Tools/Electrical rule check...* o mediante el botón de la barra superior de herramientas . Al realizarlo tendremos en pantalla una ventana con el resultado del informe, que nos dirá si se han encontrado o no errores en el circuito y cuales son estos errores en caso de haberse producido. También nos informará de la correcta generación (o no) del netlist del circuito. En esta ventana se nos da la posibilidad de copiar su contenido al portapapeles o incluso guardarlo en un archivo para su posterior uso o consulta en cualquier momento.

Otro informe que podemos realizar con ISIS es la lista de materiales (Bill Of Materials), mediante el botón  o su correspondiente opción del menú *Tools/Bill Of Materials*. Desde este menú tenemos también la posibilidad de seleccionar el tipo de salida que queramos, como por ejemplo salida en formato ASCII (texto) o HTML. El BOM del circuito ejemplo en formato HTML sería como el de la figura:

<b>Bill Of Materials For Ej01.DSN</b>		
<b>Design Title</b>	:ej01.DSN	
<b>Author</b>	:	
<b>Revision</b>	:	
<b>Design Created</b>	:domingo, 07 de marzo de 2004	
<b>Design Last Modified</b>	:domingo, 07 de marzo de 2004	
<b>Total Parts In Design</b>	:3	
<b>1 Transistors</b>		
<u>Quantity:</u>	<u>References</u>	<u>Value</u>
1	Q1	BC107
<b>2 Miscellaneous</b>		
<u>Quantity:</u>	<u>References</u>	<u>Value</u>
1	B1	15
1	L1	250
domingo, 07 de marzo de 2004 16:50:55		

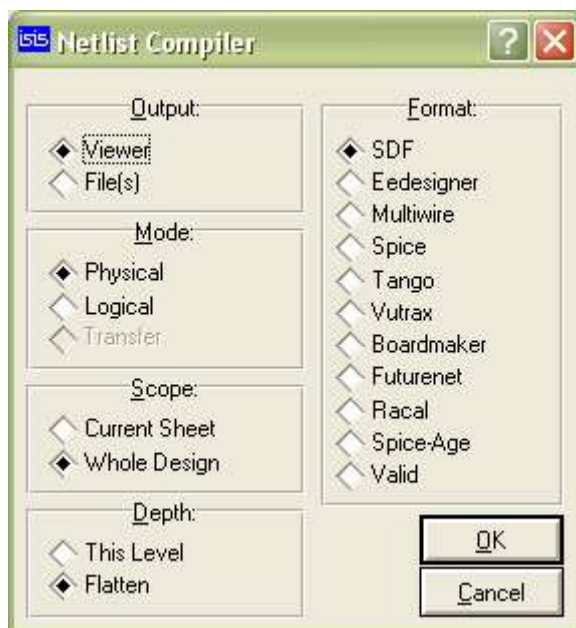
Pero probablemente el más importante de los informes que podamos generar con ISIS es el Netlist, ya que el Netlist nos permite pasar el circuito diseñado a ARES para proceder al diseño de la PCB.

Para la realización de esto tenemos varios caminos. Si lo que queremos es pasar el

circuito a ARES para diseñar la PCB directamente podemos hacerlo mediante su botón en la barra de herramientas , o su opción de menú: *Tools/Netlist to ARES*.

Dependiendo de si hemos salvado o no previamente el circuito veremos un cuadro de diálogo pidiendo que salvemos el circuito antes de proceder, tras lo cual se generará el Netlist y, si no hay ningún problema se abrirá ARES preparado para diseñar la placa del circuito.

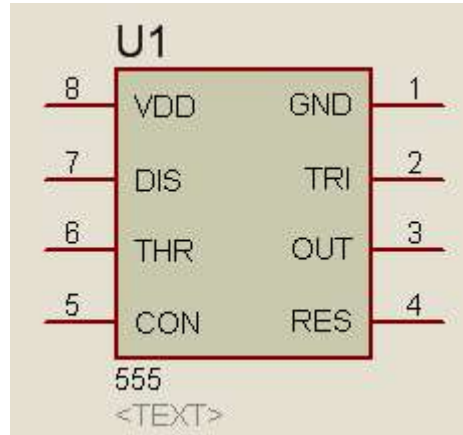
También podemos generar el Netlist aparte mediante la opción *Tools/Netlist compiler...* en la cual podemos seleccionar si queremos la salida en un visor (el mismo que nos proporcionaba por ejemplo el BOM) o en un archivo, y el formato de Netlist que queremos (lo normal sería usar SPICE).







## CREACIÓN DE COMPONENTES

En ocasiones puede interesarnos crear nuestros propios componentes para su inclusión en nuestros diseños, bien por no encontrarlo en las librerías de Proteus o por cualquier otra razón. Como ejemplo para ilustrar la creación de un componente en ISIS vamos a crear un 555, por ser un integrado sencillo y de no muchas patillas. En la figura se muestra el componente una vez creado:



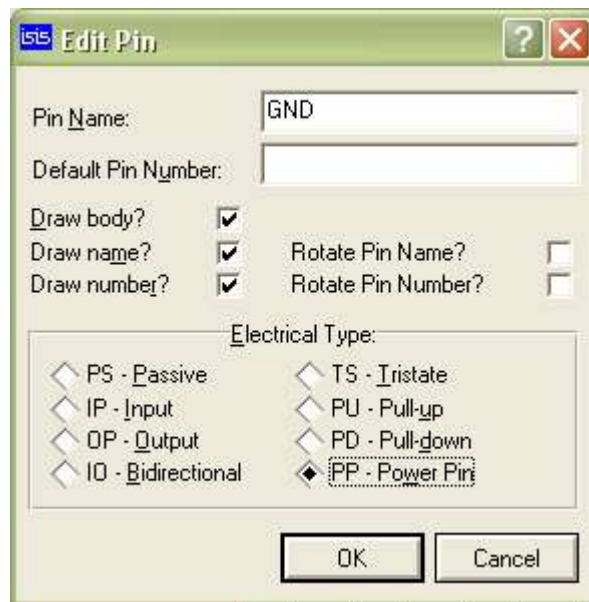
ISIS no dispone de un editor de componentes, por tanto los componentes se crean y editan en el mismo entorno de trabajo, lo que llega a ser más práctico ya que no requiere por parte del usuario acostumbrarse a varios sistemas distintos de trabajo, ya que sin salir de la misma ventana puede estar diseñando un circuito y a la vez creando un componente que pueda necesitar para llevar el diseño a término.

Para crear el componente comenzaremos dibujando su forma, para ello usaremos las herramientas de caja de gráficos 2d de Proteus:  Una vez seleccionada la herramienta crearemos un rectángulo del tamaño que consideremos adecuado para el componente haciendo click izquierdo y arrastrando hasta el tamaño correcto.

Una vez realizado el cuadro procederemos a añadir al componente los pines que tendrá, para ello empleamos la herramienta de pines :  y comprobamos que al seleccionar la herramienta en la lista de la derecha nos aparecen los diferentes tipos de pines que podemos añadir al componente, los más importantes son:

- DEFAULT -> tipo estándar de pin.
- INVERT -> Señal negada.
- POSCLK -> Entrada de reloj
- NEGCLK -> Entrada negada de reloj

Añadiremos 8 pines del tipo DEFAULT a nuestro componente, 4 a cada lado, como vemos en la figura anterior. El siguiente paso será adjudicarles los nombres a cada uno de los pines y sus características. Para ello seleccionamos un pin con click derecho y abrimos sus propiedades con un click izquierdo:

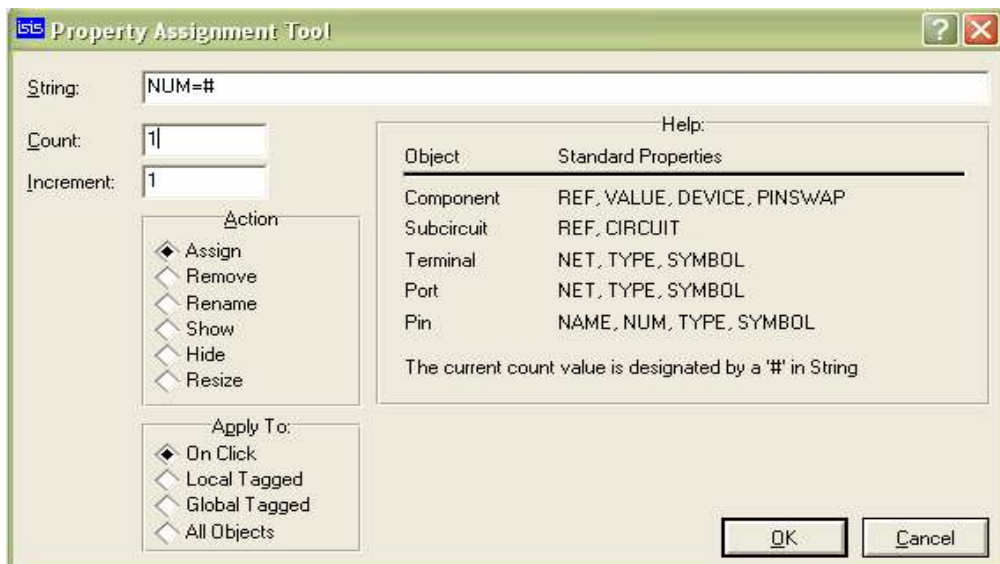


- En el campon "Pin Name" introduciremos el nombre que queremos darle al pin, que en el caso de la figura es GND (Masa).
- El campo de número de pin lo dejamos por el momento en blanco ya que posteriormente asignaremos esa numeración automáticamente.
- Las siguientes opciones nos permiten ocultar respectivamente el cuerpo del pin, su nombre y su número.
- Y por último podemos seleccionar el tipo del pin, ya sea éste pasivo, entrada, salida, bidireccional, de triple estado, salida pull-up, salida pull-down o de alimentación, como es el caso que ahora nos ocupa.

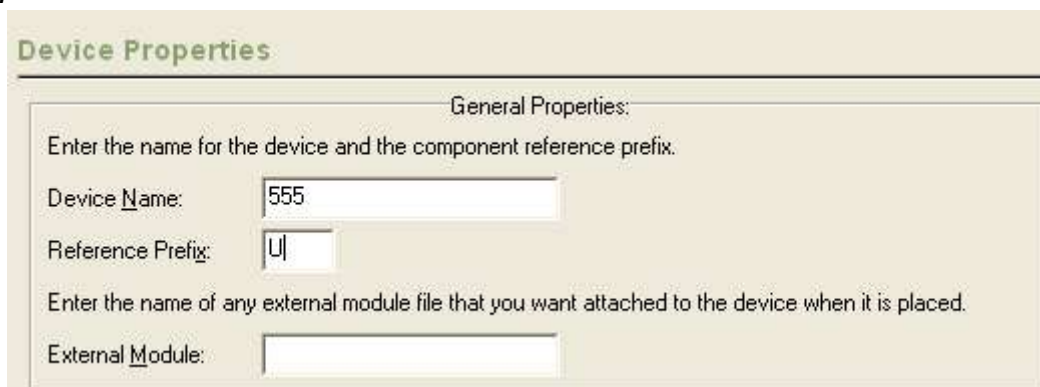
Ahora estableceremos las propiedades del resto de pines asignandoles los nombres que podemos ver en la figura del ejemplo. No debemos olvidar de asignar a los pines de alimentación su mismo tipo. Al resto de pines podemos asignarles el tipo Pasivo.

Ahora solo nos resta asignar a cada pin su número, para lo cual usaremos el *Property Assignment Tool*, que encontraremos en el menu *Tools*, o al que accederemos igualmente presionando la tecla 'A'.

En el campo String escribiremos "NUM=#", esto indica a Proteus que lo que queremos es establecer el número del pin en orden según el valor actual. El campo count se usa para indicar el valor inicial y campo increment para el incremento que habrá entre un número y otro. Para nuestro caso count deberá valer 1 e increment también. Aceptando estos valores procederemos a asignar los valores, esto se hace haciendo click en los pines del componente en el mismo orden en que queremos numerarlos, y observaremos como van apareciendo los números junto a los pines.

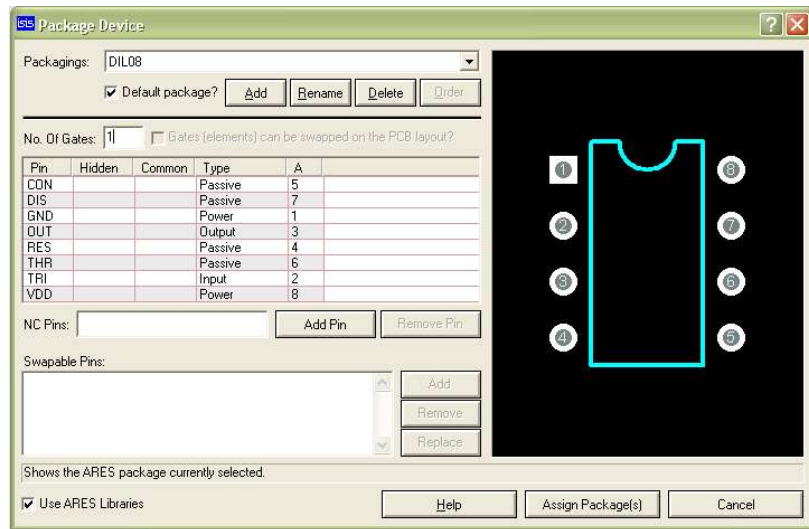


Una vez nombrados y numerados los pines ya podemos crear el componente propiamente dicho, para ello seleccionaremos todos los elementos (es decir, rectángulo y pines creados a su alrededor) y seleccionaremos del menú la opción *Library/Make device...*



Aparecerá el cuadro de diálogo de creación de componente, el cual nos pedirá en primer lugar el nombre del componente y el prefijo de referencia que utilizará, como en el ejemplo creamos un circuito integrado el prefijo será U.

El siguiente cuadro de diálogo nos informará de que no hay footprints asociados a ese componente, ante lo cual tenemos dos opciones: seguir igualmente con la creación del componente (no habrá ningún problema pero al pasar el circuito a ARES necesitaremos buscar un footprint adecuado para el componente) o seleccionar uno ahora. Vamos a seleccionar un footprint para nuestro componente pulsando en el botón *Add/Edit*.



En este diálogo podemos seleccionar el footprint adecuado para nuestro componente e incluso comprobar que es el adecuado y cambiar la numeración de los pines por si no encajara con nuestro footprint o viceversa. Para añadir un footprint al componente pulsaremos el botón *Add* y, en el cuadro de diálogo resultante seleccionaremos de la librería *Package* el footprint *DIL08* con un doble click sobre su nombre.

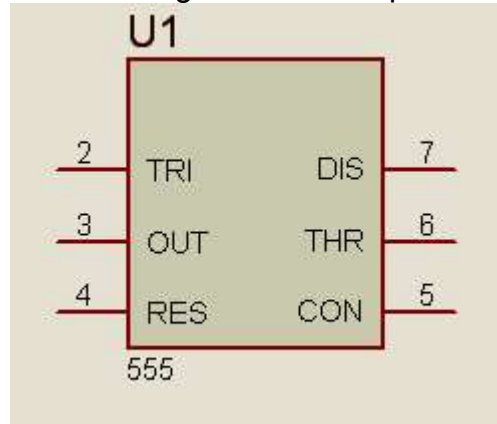
El siguiente diálogo nos pregunta por algunas más de las propiedades de nuestro componente, pero debido a que se refieren una vez más a opciones que ya hemos seleccionado al escoger nuestro footprint podremos dejar las opciones por defecto, pasando al siguiente diálogo.

En éste cuadro se nos ofrece la posibilidad de indicarle una dirección remota del datasheet del componente para poder visualizarlo en cualquier momento desde ISIS. Nos pide para ello el nombre del datasheet, la dirección del servidor ftp donde estaría alojado, y el nombre de usuario y contraseña de ese servidor.

Por último se nos pide la librería donde guardar el componente, solo tenemos que seleccionar la que queramos y aceptar, y ya tenemos nuestro componente creado y listo para ser usado en cualquier diseño que hagamos.

## EDICIÓN DE COMPONENTES

Supongamos que no queremos crear un componente sino tan sólo modificar uno ya existente, por ejemplo, supongamos que ahora decidimos que es más práctico a efectos de nuestro circuito que nuestro 555 tenga ocultos sus pines de alimentación:




Para ello colocaremos una copia del componente y después la seleccionaremos mediante el botón derecho del ratón, seleccionando después la opción del menú *Library/Decompose*, con lo cual conseguiremos que el componente se divida en las partes de que se compone, esto es, de un rectángulo y varios pines asociados. Modificaremos las propiedades de los dos pines de alimentación deseleccionando la opción "Draw body", para indicar que ese pin pasa a ser invisible. Una vez realizados los cambios pertinentes en el componente volvemos a seleccionar todas las partes del mismo y seleccionar la opción *Library/Make Device*, tal y como hicimos anteriormente para crear el componente, con la diferencia de que ahora los valores por defecto en los cuadros de diálogo son los valores del componente ya creado, con lo cual si lo que queremos es sobrescribir el componente antiguo podemos aceptar sin más todos los cuadros de diálogo y decir si a la pregunta final de Proteus sobre si deberá sobrescribir el componente. Una vez realizados todos estos pasos ya tendremos el componente con nuevas características en nuestra librería.


# SIMULACIÓN

## GENERADORES


Probablemente la virtud fundamental de Proteus sean sus potentes herramientas de simulación, que nos permitirán incluso simular uno o más microcontroladores en un mismo circuito.

Trataremos con varios ejemplos de comprender el funcionamiento del sistema de simulación de Proteus, empezando con simulaciones sencillas de cálculo de tensiones e intensidades con sus respectivas formas de onda para pasar más adelante a la simulación de microcontroladores e incluso el depurado de su código.


Una herramienta muy útil y que probablemente necesitaremos con frecuencia en el transcurso de nuestras simulaciones son los generadores. Se activan con el siguiente botón de la barra de herramientas lateral : . Al seleccionar esta herramienta comprobaremos que nos aparece la lista de posibles generadores de que disponemos en Proteus. A continuación hay una breve explicación de los más importantes (las capturas proceden de las propiedades de cada uno de ellos):

- DC  -> Produce una señal de corriente continua. Sólo nos permite configurar la tensión de salida que por defecto al colocar el generador en el espacio de trabajo es de 0v.

Voltage (Volts):	5.0
------------------	-----


- SINE  -> Es una onda senoidal de corriente alterna. Podemos parametrizar la tensión de salida y su frecuencia. La tensión de salida podemos expresarla mediante su amplitud, su tensión de pico a pico y su valor eficaz. La frecuencia podemos indicársela mediante el valor de la frecuencia propiamente dicha o de su periodo.

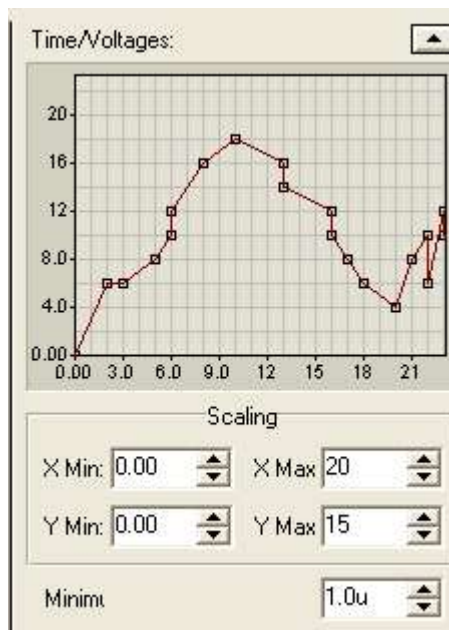
Amplitude (Volts):	
◆ Amplitude:	5.0
◇ Peak:	
◇ RMS:	
Timing:	
◆ Frequency (Hz):	50
◇ Period (Secs):	
◇ Cycles/Graph:	


- Pulse  -> Es un pulso digital como bien podría ser una señal de reloj. Podemos configurar su tensión a nivel bajo, su tensión a nivel alto, el instante en que comenzará la onda, sus tiempos de subida y bajada (que deben existir por razones de simulación aunque los proporcionados por Proteus no deberían tocarse). También podemos especificar la duración que queremos que tenga el pulso a nivel alto, expresándolo como un porcentaje del periodo total de la onda o por tiempo. Y por último podemos configurar la frecuencia de la onda tanto por el valor de su frecuencia

como por el de su periodo.

Initial (Low) Voltage:	0.00
Pulsed (High) Voltage:	5.0
Start (Secs):	0.00
Rise Time (Secs):	1.0u
Fall Time (Secs):	1.0u
Pulse Width:	
◇ Pulse Width (Secs):	
◆ Pulse Width (%):	50
Frequency/Period:	
◆ Frequency (Hz):	50
◇ Period (Secs):	
◇ Cycles/Graph:	

- Pwlin  -> Es el generador más configurable de todos ya que nos permite incluso dibujar la forma de onda resultante que queremos obtener. Sólo tenemos que hacer click izquierdo sobre los puntos de la gráfica que queramos establecer. Y en los campos inferiores podemos indicar los mínimos y máximos de la escala que queremos usar.



- - Audio  -> Este generador nos permite simular una entrada de audio (a través de un archivo .WAV) de la cual podemos indicar su amplitud y canal que queremos extraer (izquierdo o derecho).

WAV Audio File:

Amplitude (Volts):

◆ Amplitude:

◇ Peak:

Offset (Volts):

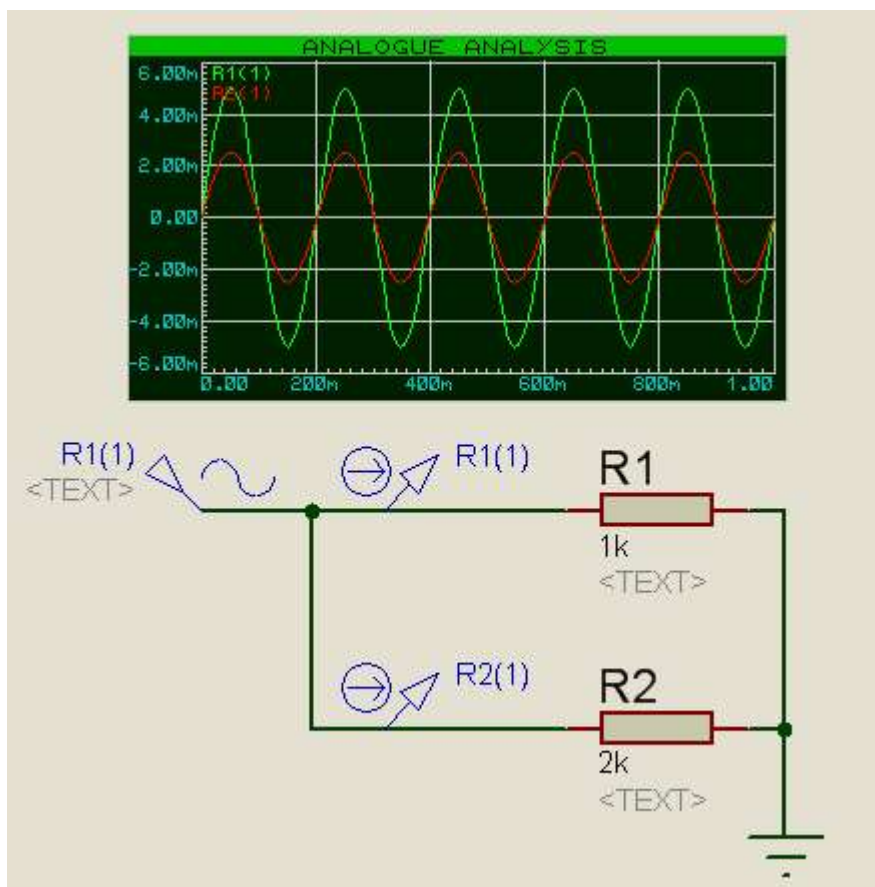
Channel:

Un detalle importante a tener en cuenta en el trabajo de generadores es que a la hora de unirlos a cualquier elemento de un circuito mediante la herramienta de cable debemos pinchar antes en el otro elemento y después en el generador, ya que Proteus necesita saber antes de conectar un generador donde va a ser conectado.






## GRÁFICAS


Ahora realizaremos un sencillo circuito para ilustrar el uso de generadores, sondas y gráficas.




Así a simple vista aparecen unos cuantos elementos nuevos, por tanto iremos por partes en el desarrollo de esta simulación:

- El primer paso es montar el circuito. Las resistencias podemos encontrarlas en la librería RESISTOR bajo el nombre de MINRES1k y MINRES2k respectivamente. El generador lo obtendremos de la herramienta generadores como hemos visto anteriormente (como propiedades le pondremos un amplitud de 5v y una frecuencia de 5Hz) y la masa sale de la herramienta conectores , bajo el nombre de GROUND.

- El siguiente elemento "extraño" son los símbolos que encontramos antes de las resistencias. Son las sondas de intensidad. Proteus dispone de sondas de tensión  y sondas de intensidad . Cada sonda tiene su botón correspondiente en el panel

de herramientas lateral :  y para colocarlas solo tenemos que pulsar el botón

correspondiente y pinchar sobre el cable donde queramos ubicarlas, e instantáneamente tomarán el nombre del componente al que se hayan conectado. Una vez conectadas las dos sondas de intensidad de la figura podemos proceder a simular el circuito mediante el botón  y comprobaremos como las sondas van adquiriendo unos valores (recordemos que la onda es senoidal por tanto no tendrá unos valores fijos).

- El siguiente paso es añadir la gráfica, para ello seleccionamos la herramienta gráficas




, y señalamos el tipo de gráfica ANALOGUE (que es el tipo de gráfica más relevante de cuantas aparecen en la lista y el que nos servirá para analizar todas nuestras señales analógicas), y la ubicamos en el espacio de trabajo haciendo click en él y arrastrando el ratón hasta un tamaño que consideremos apropiado. Con esto ya tenemos gráfica, aunque ahora esté vacía. Lo primero que haremos será editar sus propiedades (seleccionar con click derecho, propiedades con click izquierdo) y veremos que podemos cambiar su título, su instante de inicio y su instante final. Cambiaremos el título de la gráfica a aquel que queramos y le pondremos como instante final 0.20 ms para ver tan sólo un ciclo completo de la onda senoidal.

Graph title:	<input type="text" value="Gráfica"/>
Start time:	<input type="text" value="0.00"/>
Stop time:	<input type="text" value="1.00"/>
Left Axis Label:	<input type="text"/>
Right Axis Label:	<input type="text"/>

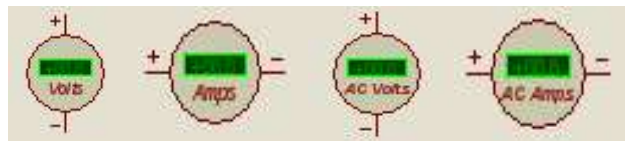
- Una vez establecidas las propiedades de la gráfica sólo nos queda arrastrar ambas sondas hacia el espacio de la gráfica y veremos cómo en su esquina superior izquierda aparecen los nombres R1 y R2, lo cual significa que ya han sido incluidos en la gráfica. Para generar los valores de la gráfica pulsamos la tecla *ESPACIO* y al instante aparecerán los valores en la gráfica. Haciendo click en el título de la gráfica ésta se abrirá en una ventana nueva.
- Una vez en la gráfica podemos actuar sobre los trazos como si de componentes se tratara. Podemos seleccionarlos, deseleccionarlos e incluso editar sus propiedades o eliminarlos.

Sería interesante ahora probar ese mismo circuito con distintos generadores y distintas configuraciones para estos para comprobar como cada vez que pulsamos *ESPACIO* se actualiza la gráfica para ajustarse a los nuevos valores. También podemos alterar los valores de las resistencias seleccionándolas mediante un click derecho y haciendo click izquierdo en el valor de la resistencia.

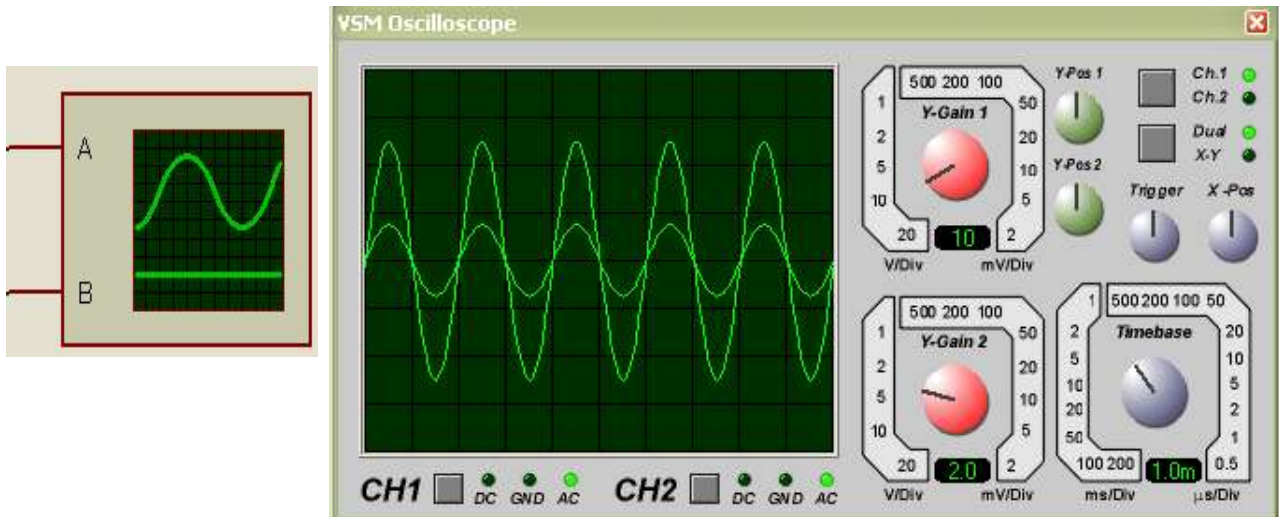
# INSTRUMENTOS VIRTUALES

Proteus pone a nuestra disposición una serie de elementos de medida y señal virtuales enormemente útiles a la hora de simular nuestros circuitos. Estos se encuentran en la herramienta *instrumentos virtuales* de la barra de herramientas . Una vez la seleccionamos podemos ver la lista de instrumentos virtuales que hay a nuestra disposición. A continuación se incluye una breve explicación de los más relevantes:

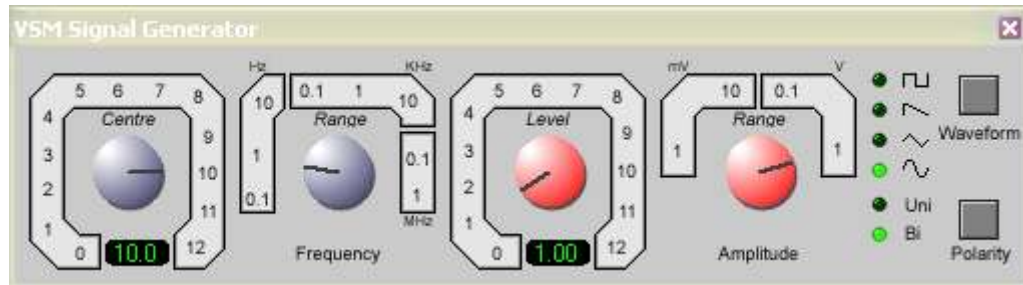
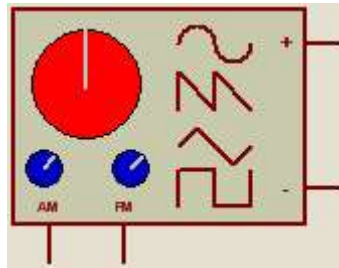
- Voltímetros y amperímetros -> Se incluyen en primer lugar por ser los más simples. Existen en Proteus 2 voltímetros y 2 amperímetros, de corriente continua y alterna respectivamente. Si editamos sus propiedades podemos especificar la unidad de salidas, esto es, si queremos la salida en voltios (o amperios según el instrumento) o alguno de sus submúltiplos.



- Osciloscopio -> Proteus dispone de un osciloscopio de 2 canales que nos permitirá medir señales de cualquier tipo. Para su utilización tan sólo tenemos que conectar una de sus entradas a la parte del circuito que queramos medir e iniciar la simulación. La ventana del osciloscopio aparecerá automáticamente en pantalla, momento en el cual sólo tenemos que actuar sobre los mandos como haríamos con un osciloscopio real hasta obtener la forma de onda deseada. **Nota:** el osciloscopio debe aparecer automáticamente al iniciar la simulación, pero para ello debe ser el último componente introducido en el circuito, esto probablemente sea debido a un fallo en la versión 6.2 SP4 (sobre la que se hizo este manual) y será corregido en versiones posteriores.



- Generador de señal -> es un generador de baja frecuencia, para su utilización lo añadiremos al circuito (seguir para ello las mismas recomendaciones que el osciloscopio) y conectaremos el pin + a la parte del circuito en la cual queramos inyectar la señal. Después iniciaremos una simulación. La ventana del generador de señal se nos abrirá automáticamente, permitiéndonos en ella cambiar el tipo de forma de onda que queremos inyectar al circuito, su frecuencia y su amplitud.



- Terminal virtual -> Se usa para simular un puerto RS232 mediante el cual podemos producir una entrada o salida a un micro que haya en nuestro diseño. La interacción se produce mediante el teclado. En las propiedades de este terminal virtual podemos cambiar la velocidad de la conexión en baudios y otras especificaciones de la conexión.

# SIMULACIÓN DE MICROCONTROLADORES

## AJUSTES PREVIOS

Sin duda el principal atractivo de Proteus radica, a pesar de todas sus ventajas en muchos ámbitos con programas de características similares, en la posibilidad de simular microprocesadores y, particularmente, microcontroladores, como es por ejemplo el caso de los PIC de MicroChip, y de los cuales nos ocuparemos ahora.

Ya hemos discutido anteriormente que el punto fuerte de Proteus es su versatilidad, y ahora es cuando verdaderamente vamos a comprobar has qué punto es cierto esto. No sólo nos permitirá Proteus incluir microcontroladores en nuestros diseños y simularlos, si no que además nos permitirá depurar sus programas e incluso crearlos y compilarlos sin salir del entorno de Proteus. Para familiarizarnos con todo el proceso crearemos desde cero un programa en ensamblador para el pic16f84, lo compilaremos, lo incluiremos en un circuito, lo simularemos y lo depuraremos. Pero vayamos por partes, lo primero que debemos hacer es hacer unos ajustes en Proteus para que podamos compilar nuestros programas:

- Entraremos en el menú *source/Code generation tools...* , seleccionaremos de la lista el compilador MPASMWIN y escribiremos en command line : `/p16f84 /q "%1"` .

Code Generation Tool

Tool: MPASMWIN Browse

Path: C:\Archivos de programa\Labcenter Electronics\Proteus 6 Professional\TOOLS\MPASM\MPAS

Make Rules

Source Extn: ASM Obj. Extn: HEX Always Build: ☐

Command Line: /p16F84 /q "%1"

Use %1 for source file, %2 for object file, %3 for list file

Debug Data Extraction

List File Extn: LST Browse

Path: C:\Archivos de programa\Labcenter Electronics\Proteus 6 Professional\Tools\MPASM\mpas

- **Nota:** La configuración comentada, como puede comprobarse es para programar un pic16f84, en caso de querer programarse otro modelo de PIC se debe sustituir el 16f84 por el modelo en cuestión para evitar fallos de compilación.
- Además de esto cambiaremos el editor de código de Proteus, ya que el existente no nos da la posibilidad de crear un archivo desde cero. Para ello seleccionaremos el menú *source/setup external text editor...* y cambiaremos el ejecutable que aparece por cualquier otro que usemos, en el ejemplo se usa el bloc de notas de windows pero puede ser cualquier otro.

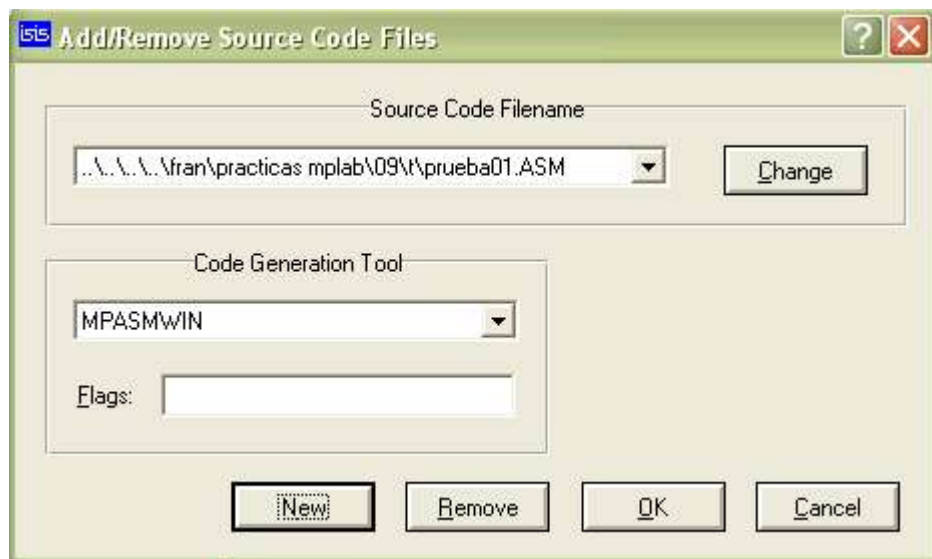
Executable

C:\WINDOWS\notepad.exe Browse

## COMPILACIÓN

Una vez realizado este paso ya podemos compilar nuestros programas con Proteus, vamos a comprobarlo. Entramos en el menú *source/add-remove source files...* y aparecerá un cuadro de diálogo parecido al de la figura. Los pasos que hemos de seguir para añadir un archivo de código son los siguientes:

- Seleccionar el compilador de la lista. En nuestro caso el compilador es el MPASMWIN, por tanto lo seleccionaremos de la lista.
- Pulsar el botón **New** para añadir un archivo de código a la lista, cada vez que lo pulsemos añadiremos un archivo de código al proyecto (usaremos el botón remove para borrar archivos de código del proyecto)
- En un cuadro de diálogo se nos pedirá el nombre del archivo .ASM que usaremos. Podemos coger un archivo .ASM ya creado o crear nosotros uno, como es el caso. Por tanto entraremos en la carpeta donde queramos guardar el archivo y le pondremos por nombre, por ejemplo, prueba01. El programa nos informará de que dicho archivo no existe y nos preguntará si deseamos crearlo (obviamente responderemos que sí).



- Ya podemos salir del cuadro de diálogo. Si entramos en el menú *source* comprobaremos como en su parte inferior hay una opción nueva con el nombre del archivo que acabamos de añadir al proyecto. La seleccionaremos y se nos abrirá el editor de texto que hayamos seleccionado. Ahora introduciremos el siguiente código en él:

```
LIST P=16F84
STATUS EQU 3
PORTA EQU 5
TRISA EQU 5
PORTB EQU 6
TRISB EQU 6
RP0 EQU 5
ORG 0XC
CONTA RES 1 ; REGISTRO GENERAL PARA EL RETARDO

ORG 0X00
INICIO BSF STATUS,RP0 ; ENTRA EN EL BANCO 1
        MOVLW 0X1F ; MUEVE 1Fh A W
        MOVWF TRISA ; CONFIGURA EL PUERTO A COMO ENTRADA
        MOVLW 0X00 ; MUEVE 00 A W
        MOVWF TRISB ; CONFIGURA EL PUERTO B COMO SALIDA
        BCF STATUS,RP0 ; SALE DEL BANCO 1
        CLRF PORTB ; LIMPIA EL PUERTO B
```

```

OTRA BTFSS PORTA,4 ; PREGUNTA SI EL PIN 4 DEL PUERTO A ES CERO
      GOTO OTRA ; SALTA A LA ETIQUETA OTRA
      MOVLW 0x18 ; MUEVE 18h A W
      MOVWF PORTB ; CARGA W EN EL PUERTO B
      CALL RETARDO ; LLAMA A LA ETIQUETA RETARDO
      BTFSS PORTA,4 ; PREGUNTA SI EL PIN 4 DEL PUERTO A ES CERO
      GOTO OTRA1 ; SALTA A LA ETIQUETA OTRA1
      MOVLW 0x24 ; MUEVE 24h A W
      MOVWF PORTB ; CARGA W EN EL PUERTO B
      CALL RETARDO ; LLAMA A LA ETIQUETA RETARDO
      BTFSS PORTA,4 ; PREGUNTA SI EL PIN 4 DEL PUERTO A ES CERO
      GOTO OTRA1 ; SALTA A LA ETIQUETA OTRA1
      MOVLW 0x42 ; MUEVE 42h A W
      MOVWF PORTB ; CARGA W EN EL PUERTO B
      CALL RETARDO ; LLAMA A LA ETIQUETA RETARDO
      BTFSS PORTA,4 ; PREGUNTA SI EL PIN 4 DEL PUERTO A ES CERO
      GOTO OTRA1 ; LLAMA A
      MOVLW 0x81 ; MUEVE 81h A W
      MOVWF PORTB ; CARGA W EN EL PUERTO B
      CALL RETARDO ; LLAMA A LA ETIQUETA RETARDO
OTRA1 CLRF PORTB ; LIMPIA EL PUERTO B
      GOTO OTRA ; SALTA A LA ETIQUETA OTRA
RETARDO MOVLW 0x32 ; MUEVE 32h A W
      MOVWF CONTA ; CARGA W A CONTA3
RET NOP ; NO OPERA
      NOP
      NOP
      NOP
      NOP
      DECFSZ CONTA,1 ; DECREMENTA CONTA1 EN 1 Y SALTA SI ES CERO
      GOTO RET ; SALTA A LA ETIQUETA RET1
      RETURN ; RETORNA
END

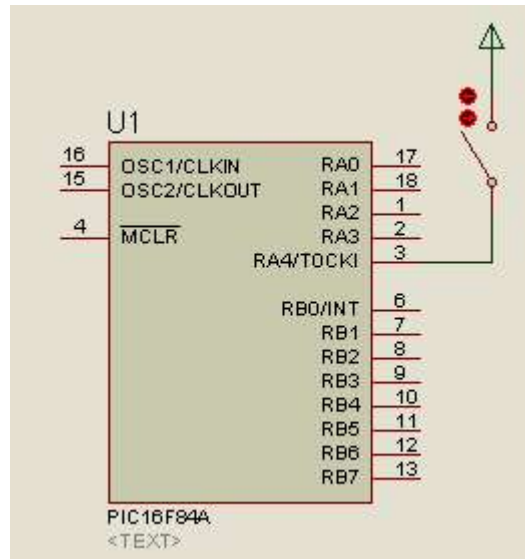
```

Todo aquello que siga a un ; son comentarios y no es necesario copiarlos para el correcto funcionamiento del programa, aunque sí son recomendables para su posterior legibilidad. Una vez escrito el programa y guardado procederemos a compilarlo. Para ello usaremos la opción de menú *source/build all*, la cual abrirá el compilador y procederá a crear el archivo .HEX de nuestro programa. Si no ha habido errores (el compilador nos informará en caso de haberlos) ya tenemos nuestro archivo hexadecimal con el cual podemos simular ya o incluso programar un PIC real mediante un circuito programador y un software destinado a tal efecto como ICPROG. Si ha surgido algún error de compilación debemos verificar el código introducido.



# SIMULACIÓN

Ya tenemos nuestro archivo hexadecimal, ahora vamos a proceder a simularlo y depurarlo. Para ello crearemos el siguiente circuito :



- El PIC16f84A lo encontraremos en la librería MICRO, mientras que el interruptor se encuentra en ACTIVE bajo el nombre de SWITCH. La alimentación se encuentra en la herramienta conectores como POWER. Ahora solo resta asociar al microcontrolador el hexadecimal que hemos creado. Para ello abriremos las propiedades del componente, y en el campo *Program File* introduciremos la ruta del .HEX creado (también puede ser cualquier otro .HEX, para lo cual no se necesita ninguno de los pasos anteriores).
- El circuito, como puede comprobarse es extremadamente sencillo, y esa es otra virtud de Proteus, permitirnos realizar la simulación necesitando para ello un número mínimo de componentes.
- Probemos a iniciar la simulación pulsando F12 o su botón correspondiente en la barra de herramientas inferior y activemos el interruptor para comprobar como reacciona el microcontrolador. Observaremos cómo se activan y desactivan las salidas sin necesidad de añadirle otros componentes como leds o sondas lógicas. Azul se corresponde a un estado bajo de la salida y Rojo a un estado alto. Gris se corresponde a un estado indeterminado.



Resumamos los pasos para crear y compilar nuestro propio programa:

- *source/add-remove source files...* -> Indicamos MPASMWIN como el compilador a usar y le damos el nombre del archivo o archivos que queremos compilar.
- Seleccionaremos en el menú *source* el nombre del programa que deseemos crear o editar y realizaremos sobre él las acciones oportunas o presionar *ALT-S* y el número del archivo deseado.
- *Source/build all* para compilar y generar el archivo o archivos .HEX
- Cargar el programa en el microcontrolador

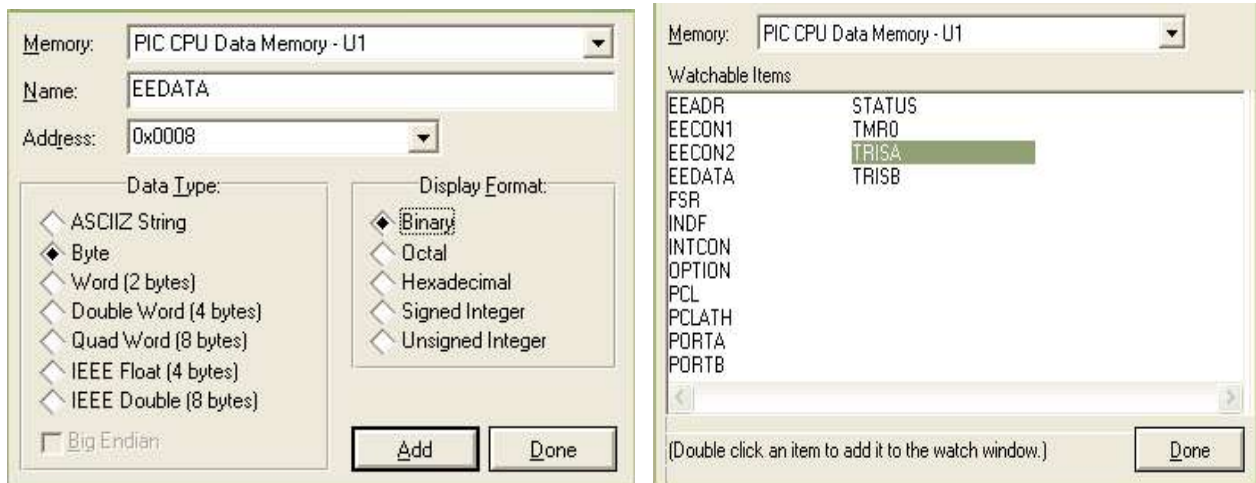
Si no queremos compilar el programa sino tan sólo simular un fichero .HEX que hemos compilado previamente con cualquier otra aplicación tan sólo este último paso es necesario.



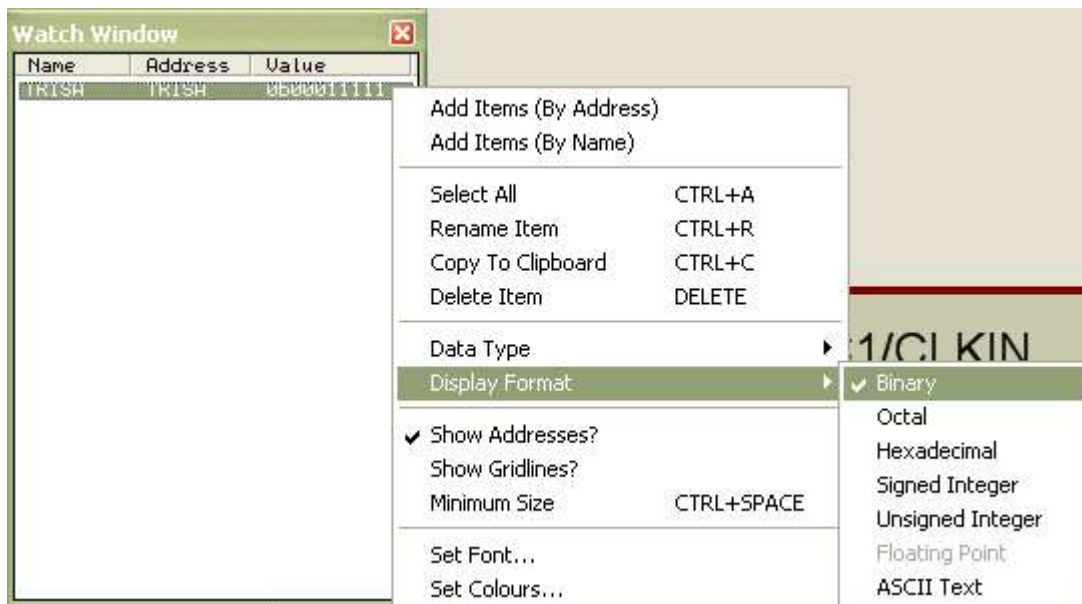
# DEPURACIÓN

Pero para sacarle el máximo partido a Proteus no debemos conformarnos con compilar y simular nuestros programas. También deberemos ser capaces de depurarlos, que es lo que vamos a hacer ahora. Para ellos emplearemos el menú *debug* y sus múltiples opciones. Para acceder a estas opciones debe estar en marcha la simulación y pausada mediante el botón , ya que mientras la simulación se está produciendo no se activarán las ventanas de depurado. Una vez iniciemos la simulación y la pausemos exploraremos el menú debug pinchando en el o pulsando *ALT+D* y el número de la ventana que queramos activar. Mientras simulamos podemos ir paso a paso en la simulación con el botón . A continuación se explicarán brevemente las ventanas de depuración más importantes:

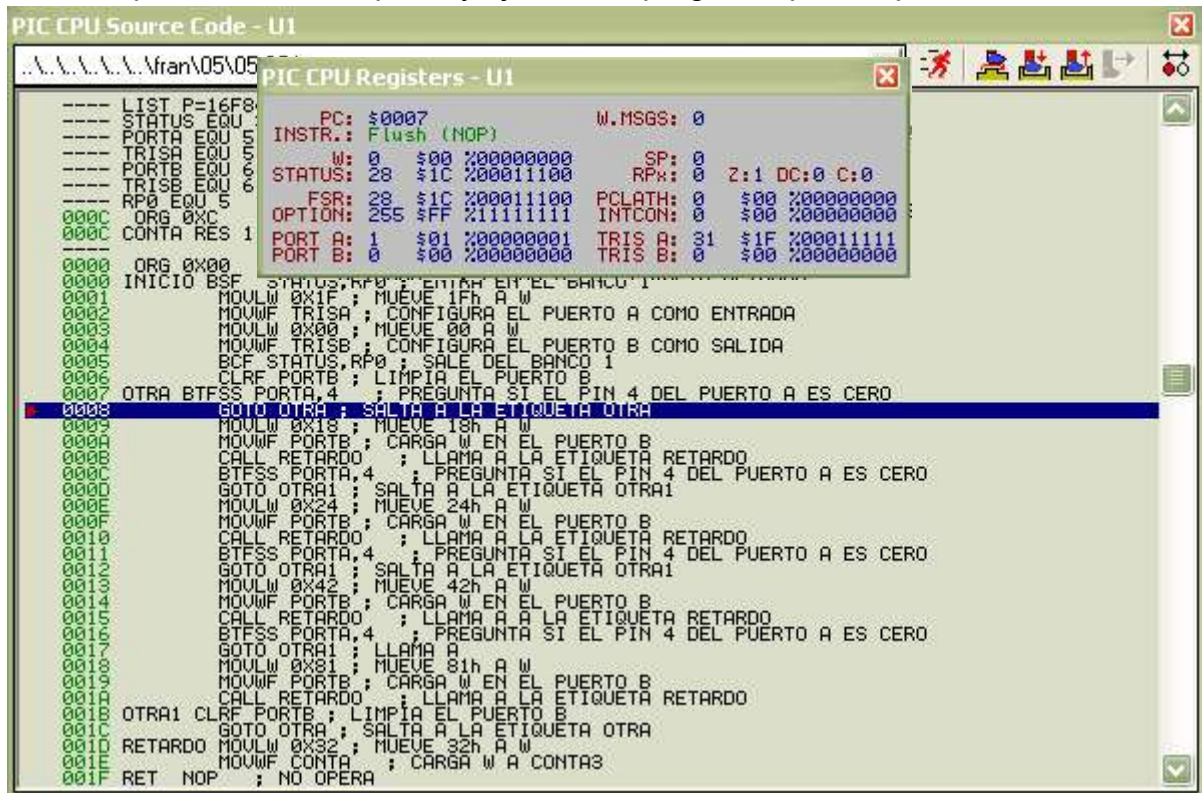
- 2-Watch window -> Es el inspector de variables. Con él podemos vigilar el valor de cualquier registro o zona de memoria. Para ello haremos click derecho sobre el inspector de variables, seleccionando *Add Items*, pudiendo hacerlo por dirección o por nombre. Una vez seleccionada la opción podremos, como en la figura de la izquierda, seleccionar la dirección de memoria que queremos vigilar o, como en la figura de la derecha, seleccionar el nombre del registro en cuestión.



Si queremos cambiar el formato en el que se mostrarán estos datos usaremos el menú contextual sobre el registro y seleccionaremos *Display format*, y ahí elegiremos la opción deseada.



- 3-CPU source code -> en esta ventana podemos depurar el código ensamblador en sí, controlando en cada momento qué instrucción se está ejecutando, permitiéndonos introducir puntos de interrupción y ejecutar el programa paso a paso.



- > Simular – Reanuda la simulación.
- > Step over – Ejecutar hasta fin de la función o subrutina.
- > Step into – Ejecutar hasta entrar en la función o subrutina.
- > Step out – Ejecutar hasta salir de la función o subrutina.
- > Ejecutar hasta la posición del cursor.
- > Activar o desactivar punto de interrupción en la posición del cursor.

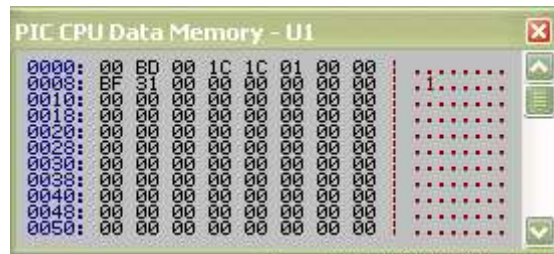
Cuando durante la ejecución se llega a un punto de interrupción ésta se detiene

instantáneamente.

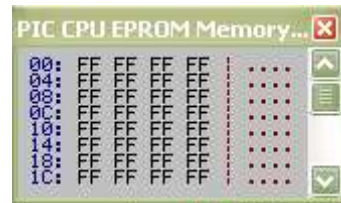
- 4-PIC CPU Registers->Ventana con información de la instrucción que se está ejecutando actualmente y valores de los principales registros de memoria del PIC, tales como el registro de trabajo, el registro de estado, la configuración y valor actual de los puertos, etc...



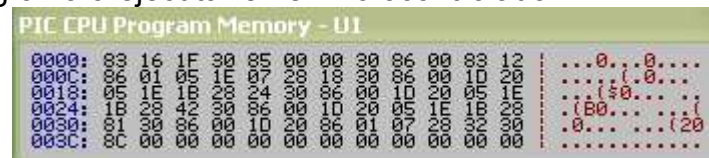
- 5-PIC CPU DATA Memory -> Esta ventana, al igual que las siguientes ya se usa para un depurado a más bajo nivel, en ella podemos ver la memoria de datos del microcontrolador, en la cual se almacenarán los contenidos de las variables que se usen. Con el inspector de variables podemos hacer un seguimiento más exhaustivo de cualquier posición concreta de memoria, por lo cual esta ventana se reserva para usos más generales.



- 6-PIC CPU EPROM Memory->En esta ventana se nos muestra el contenido de la memoria EPROM del PIC. Es muy útil cuando tenemos programas que deben leer o escribir de ésta memoria para verificar que estas operaciones se realizan correctamente.



- 7-PIC CPU PROGRAM Memory-> Ésta es la memoria de programa del PIC. Aquí se almacena el programa a ejecutar en el microcontrolador.



- 8-PIC CPU STACK->Aquí podemos ver la pila de ejecución del microcontrolador.

