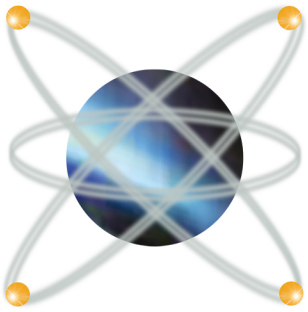


PROTEUS DESIGN SUITE

Getting Started Guide.

Labcenter Electronics
www.labcenter.com

COPYRIGHT NOTICE

WARNING

You may make a single copy of the software for backup purposes. However, you are warned that the software contains an encrypted serialization system. Any given copy of the software is therefore traceable to the master disk supplied with your licence.
PROTEUS also contains special code that will prevent more than one copy using a particular licence key on a network at any given time. Therefore, you must purchase a licence key for each copy that you want to run simultaneously.

DISCLAIMER

No warranties of any kind are made with respect to the contents of this software package, nor its fitness for any particular purpose. Neither Labcenter Electronics Ltd nor any of its employees or sub-contractors shall be liable for errors in the software, component libraries, simulator models or documentation, or for any direct, indirect or consequential damages or financial losses arising from the use of the package.
Users are particularly reminded that successful simulation of a design with the PROSPICE simulator does not prove conclusively that it will work when manufactured. It is always best to make a one off prototype before having large numbers of boards produced.
Manufacturers' SPICE models included with PROSPICE are supplied on an 'as-is' basis and neither Labcenter nor their originators make any warranty whatsoever as to their accuracy or functionality

TABLE OF CONTENTS

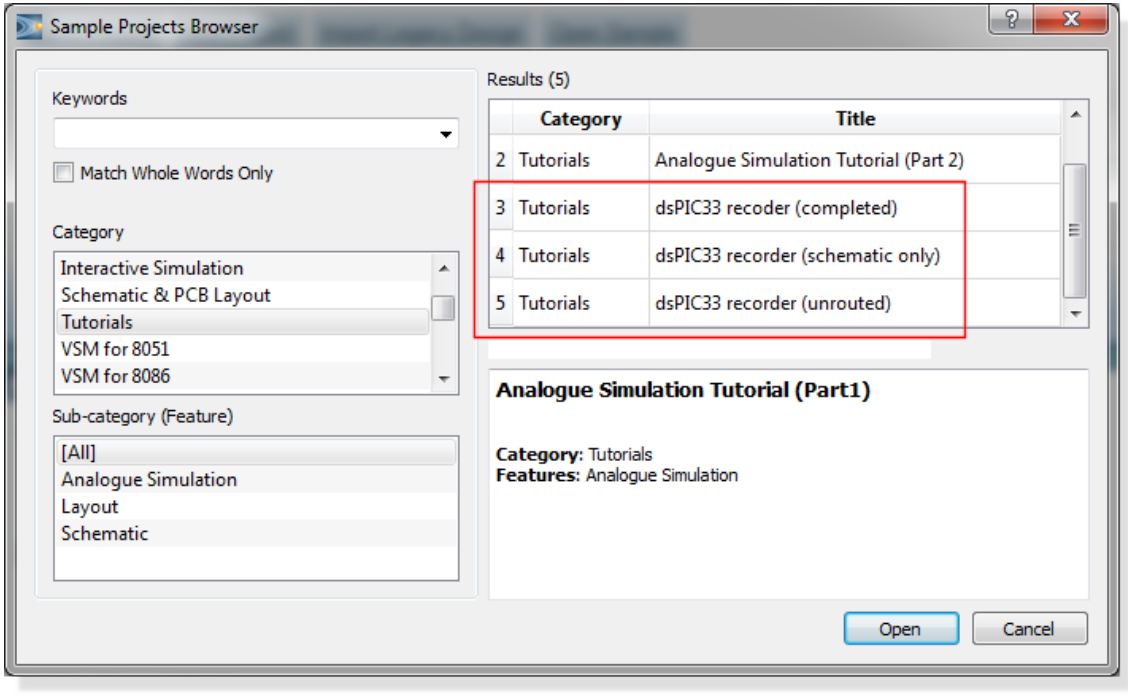
SCHEMATIC CAPTURE TUTORIAL

INTRODUCTION

The aim of this tutorial is to take you through the process of entering a circuit of modest complexity in order to familiarise you with the techniques required to drive the schematic capture module in Proteus (ISIS). The tutorial starts with the easiest topics such as placing and wiring up components, and then moves on to make use of the more sophisticated editing facilities, such as creating new library parts.

An accompanying tutorial in the PCB module (ARES) then continues the project development using the completed schematic drawn in this tutorial.

For those who want to see something quickly, DSPIC33_REC_SCHEMATIC.pdsprj contains the completed tutorial circuit but no layout while DSPIC33_REC_UNROUTED.pdsprj and DSPIC33_REC_COMPLETE.pdsprj both contain a completed schematic and a PCB. All of these projects can be loaded from the Open Sample command on the Proteus 8 home page under the tutorials category.



Note that throughout this tutorial (and the documentation as a whole) reference is made to keyboard shortcuts as a method of executing specific commands. The shortcuts specified are the default or system keyboard accelerators as provided when the software is shipped to you. Please be aware that if you have configured

your own keyboard accelerators the shortcuts mentioned may not be valid. Information on configuring your own keyboard shortcuts can be found in the General Concepts section of the Documentation.

Creating a New Project

We shall assume at this point that you have installed the Proteus 8 software package.

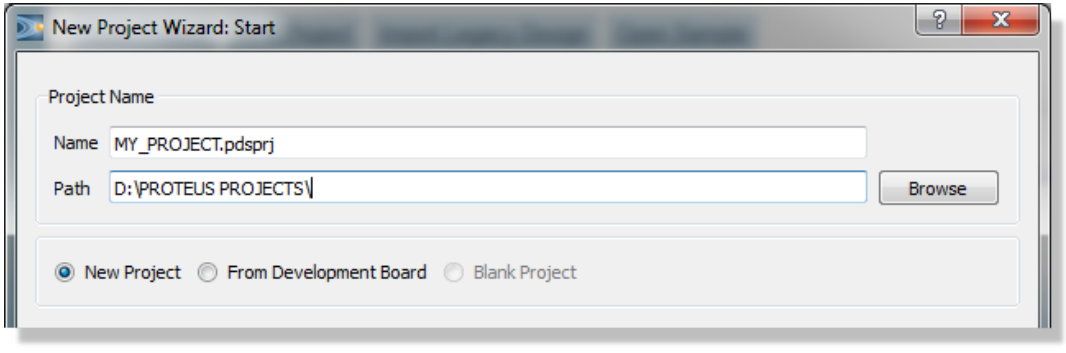
To start the software, click on the Start button and select Programs, Proteus 8 Professional and then the Proteus 8 application. The main application will then load and run and you will be presented with the Proteus home page.

If you have a Demonstration copy of the software you can start the Proteus application via the Proteus 8 Demonstration tab from the Start Menu.

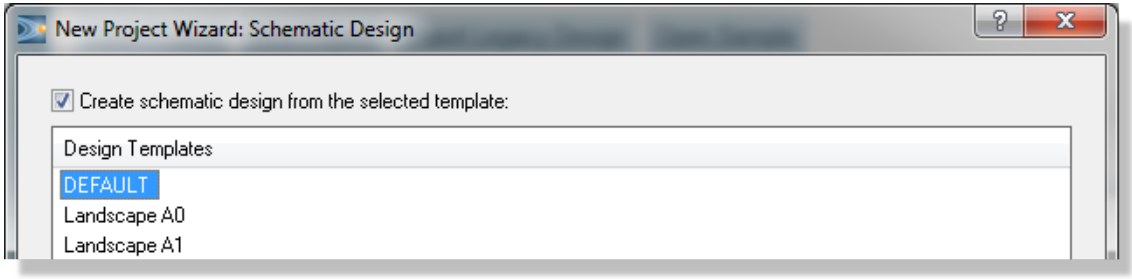
In order to create a schematic we must first create a project. Since this tutorial is partnered with the PCB tutorial we will create a project for schematic/PCB.

Start by pressing the new project button near the top of the home page in Proteus.

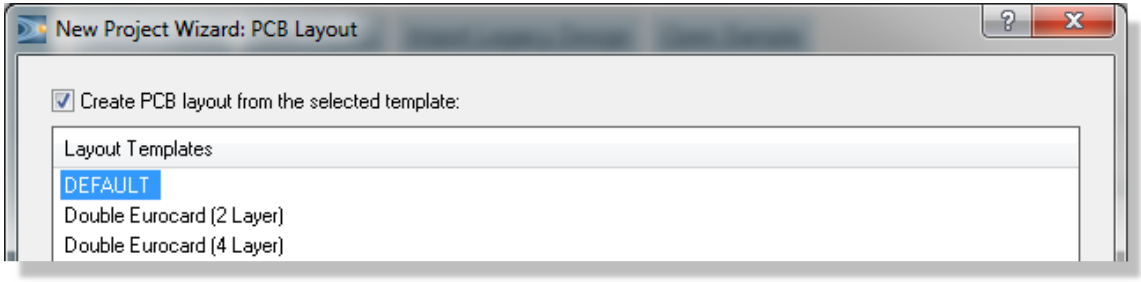
On the first page of the wizard specify a name and path for the project.



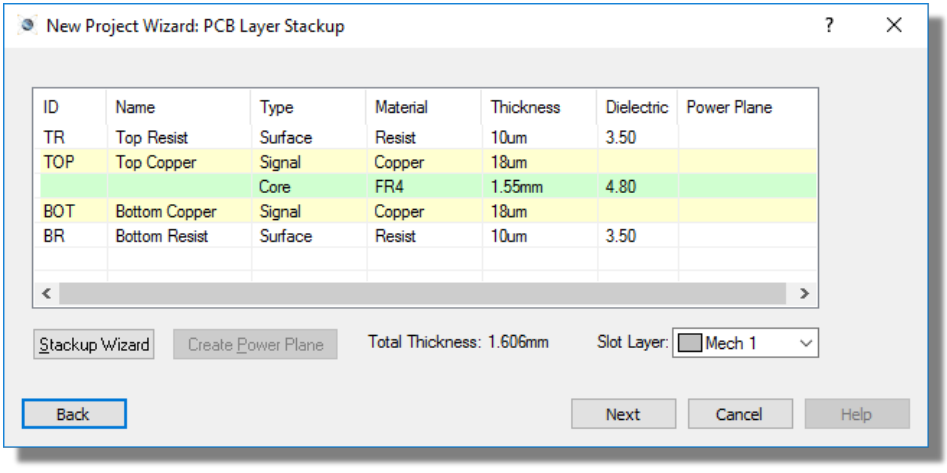
We need a schematic so check the box at the top of the next step and then choose the default template.



Similarly, we need a layout so check the box at the top of the layout page and again choose the default template.

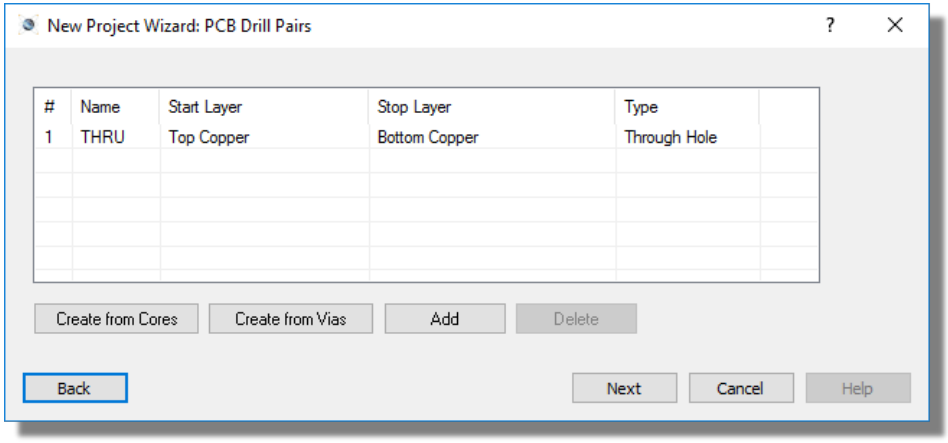


The next screen allows us to define the layer stack for our PCB. Since we will be designing a simple two layer board there is no configuration necessary here.

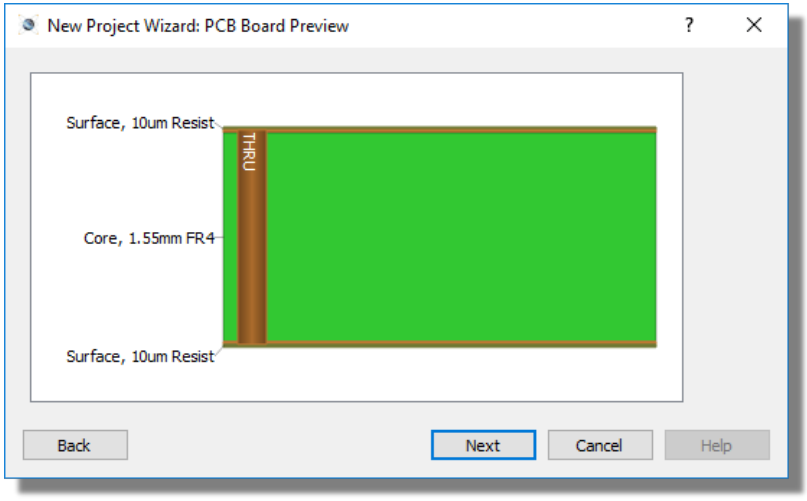


> For multi-layer PCB's the stackup wizard button would be used to define the number of copper layers, cores and pre-preg's. This is discussed in more detail in the accompanying PCB tutorial.

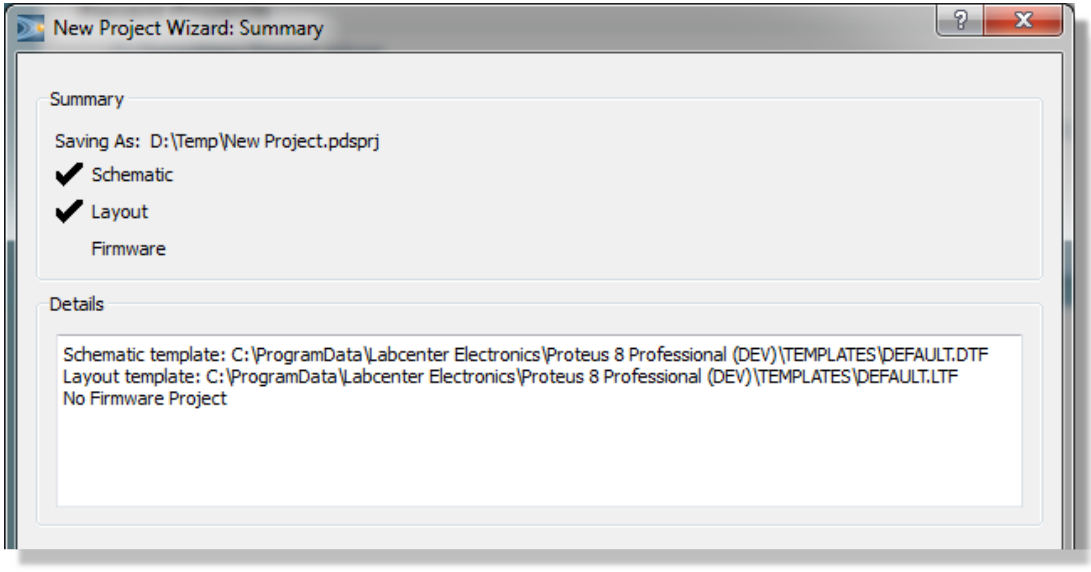
The next screen is for configuration of drill spans. Again, for our proposed 2-layer board the only possibility is thru-hole so there is no action required.



The final screen in the PCB configuration is simply a preview of a PCB cross section that displays visually what has been set up in the previous screens.



We are not simulating the design so leave the firmware page blank and continue on to the summary which should look like the following:



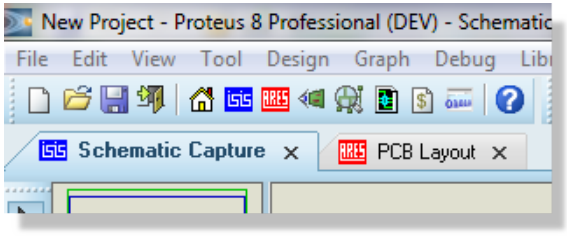
Click on the finish button to create the project.

- A schematic template can contain sheet size, colour scheme, company logo, header block and various other aesthetic presets. Further information can be found in the Templates chapter of the reference manual.

- A PCB template can contain board edge, mounting holes, design rules, layer stack and various other technology information. Refer to the Templates chapter in the PCB documentation for more information.

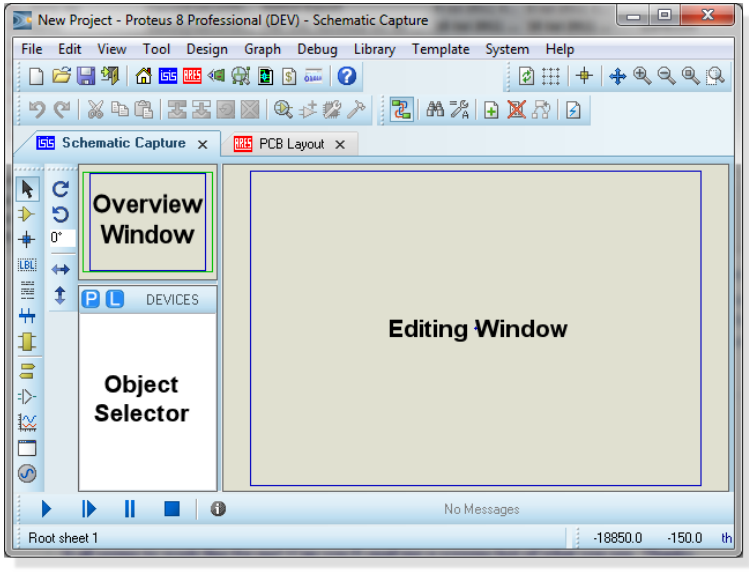
- The configuration of the Layer Stack and Drill Spans is really important for multi-layer PCB's and is discussed in some detail in the reference manual.

The project will open with two tabs, one schematic capture and the other for PCB layout. Click on the schematic tab to bring the ISIS module to the foreground.



Guided Tour

The largest area of the screen is called the Editing Window, and it acts as a window on the drawing - this is where you will place and wire-up components. The smaller area at the top left of the screen is called the Overview Window. In normal use the Overview Window displays, as its name suggests, an overview of the entire drawing - the blue box shows the edge of the current sheet and the green box the area of the sheet currently displayed in the Editing Window. However, when a new object is selected from the Object Selector the Overview Window is used to preview the selected object - this is discussed later.

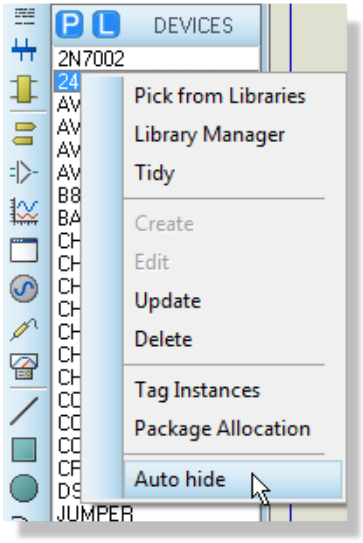


Isis Schematic Capture Window

If you don't like the default layout of the toolbars you can pick them up and dock them on any of the four sides of the application. Similarly you can move the Object Selector & Overview Window pane across to the right hand side of the application by dragging the end of it all the way across to the other side.

> Toolbars and menu options will switch according to which tab is active (at the front). Throughout this tutorial when we refer to an icon or a menu command we are assuming that the schematic tab is active.

Right clicking the mouse either in the Object Selector or in the Overview Window will provide a context menu, including the option to 'auto hide' the left hand pane. This is extremely useful if you want to maximise the editing area of the application. When enabled the Object Selector and Overview Window will be minimised to a 'flyout bar' at the left (or right) of the application by default and will appear either when the mouse is placed over the bar or when the mode of operation is changed by selecting a different icon.



Auto hide the Object Selector

Device Mode Icon

Navigation of the view displayed in the Editing Window takes two forms; adjusting the scale of the drawing (zooming) and adjusting the area of the drawing displayed (panning). These techniques are somewhat intertwined and are discussed in more detail below:

Zooming

There are several ways to zoom in and out of areas of the schematic:

- Point the mouse where you want to zoom in and out of and roll the middle mouse button (roll forwards to zoom in and backwards to zoom out).
- Point the mouse where you want to zoom in or out of and press the F6 or F7 keys respectively.
- Hold the SHIFT key down and drag out a box with the left mouse button around the area you want to zoom in to. We call this Shift Zoom
- Use the Zoom In, Zoom Out, Zoom All or Zoom Area icons on the toolbar.

Zoom Icons

- The F8 key can be used at any time to display the whole drawing.
- The Shift Zoom and the middle mouse zoom techniques can also be used over the Overview Window. That is, you can position the mouse over the Overview Window

and either roll the middle mouse button or use Shift Zoom to navigate in or out of an area of the schematic.

Panning

As with zooming, there are a number of options for panning across the editing window.

- Click on the middle mouse button to enter track pan mode. This puts ISIS is a mode where the entire sheet is picked up and will move as you move the mouse. The track pan cursor will indicate when you have entered this mode. Left click the mouse again to exit track pan mode.
- To simply 'pan' the Editing Window up, down, left or right, position the mouse pointer over the desired part of the Editing Window and press the F5 key.
- Hold the SHIFT key down and bump the mouse against the edges of the Editing Window to pan up, down, left or right. We call this Shift Pan.
- Should you want to move the Editing Window to a completely different part of the drawing, the quickest method is to simply point at the centre of the new area on the Overview Window and click left.
- Use the Pan Icon on the toolbar

> Note that when using the track pan method above you can also zoom in and out by rolling the mouse wheel. So, click the middle mouse button to pick up the sheet and move the sheet by moving the mouse and zoom the sheet by rolling the middle mouse button. Left click to 'drop' the sheet and exit track pan mode.

It is well worth spending a few moments familiarising yourself with navigation in ISIS - it is after all one of the most common operations you will perform. In particular, learning to use the middle mouse button both for track pan and for zooming will save you time during schematic design.

A grid of dots or lines can be displayed in the Editing Window as a visual aid using the Grid command on the View menu, or by pressing 'G' to toggle the grid from ‘dots’, ‘lines’ or ‘off, or by clicking the Grid Icon on the toolbar. The grid helps in lining up components and wires and is less intimidating than a blank screen. If you find it hard to see the grid dots or lines, either adjust the contrast on your monitor slightly (by default the grid is displayed in light grey) or change their colour with the Set Design Defaults on the Template menu.

Below the Overview Window is the Object Selector (sometimes known as a 'parts bin') which you use to select devices, symbols and other library objects. We'll familiarise ourselves further with using the Object Selector later.

Finally, at the bottom of the screen is the co-ordinate display, which reads out the co-ordinates of the mouse pointer when appropriate. These co-ordinates are in 1 thou units and the origin is in the centre of the drawing

Origin Co-ordinates

Note that ISIS allows you to reposition all the icon toolbars as well as move/resize the Object Selector/Overview Window. Do be aware however, that this documentation refers to all the rebars and windows in their default locations.

Visual Aids to Design

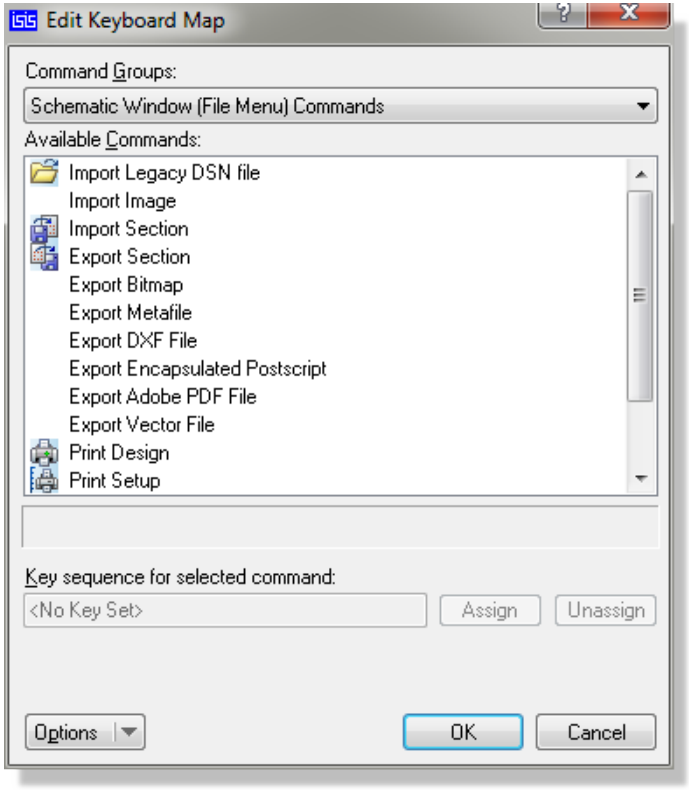
ISIS is designed to be as user friendly as possible and provides two main ways to help you see what is happening during the design process – objects are encircled with a dashed line or 'twitched' when the mouse is over them and mouse cursors will change according to function. Essentially, the object-twitching scheme tells you which object the mouse is over (the 'hot' object) and the mouse cursor tells you what will happen when you left click the mouse on that object. While extremely intuitive, a summary of cursors used, together with their actions, is provided below:

Cursor	Description
	Standard Cursor - Used in selection mode when not over a 'hot' object.
	Placement Cursor - Placement of an object will commence on a left click of the mouse.
	Hot Placement Cursor - Appears green when placement of a wire is available on left click of the mouse.
	Bus Placement Cursor - Appears blue when placement of a BUS is available on left click of the mouse.
	Selection Cursor - Object under the mouse will be selected on a left click of the mouse.
	Move Cursor - The currently selected object can be moved.
	Drag Cursor - The wire or 2D graphic can be dragged by holding the left mouse button down
	Assignment Cursor - When over an object (having set the Property Assignment Tool) You can assign the property by left clicking the mouse button.

We will see more of the different cursor types throughout the course of the tutorial.

Keyboard Shortcuts

For those customers who prefer to invoke commands and dialogues via the keyboard Proteus provides a comprehensive and flexible scheme for assigning shortcuts. The dialogue form is invoked from the System Menu – Set Keyboard Mapping command, where all commands and operation modes can be accessed from the combo box at the top. Simply select the command you want, apply a key sequence and assign it to that command



Keyboard Mapping Dialogue Form

- While there are few limits on shortcuts, a few standard windows accelerators are restricted as they are global across the entire Proteus system (e.g. CTRL+S for save).

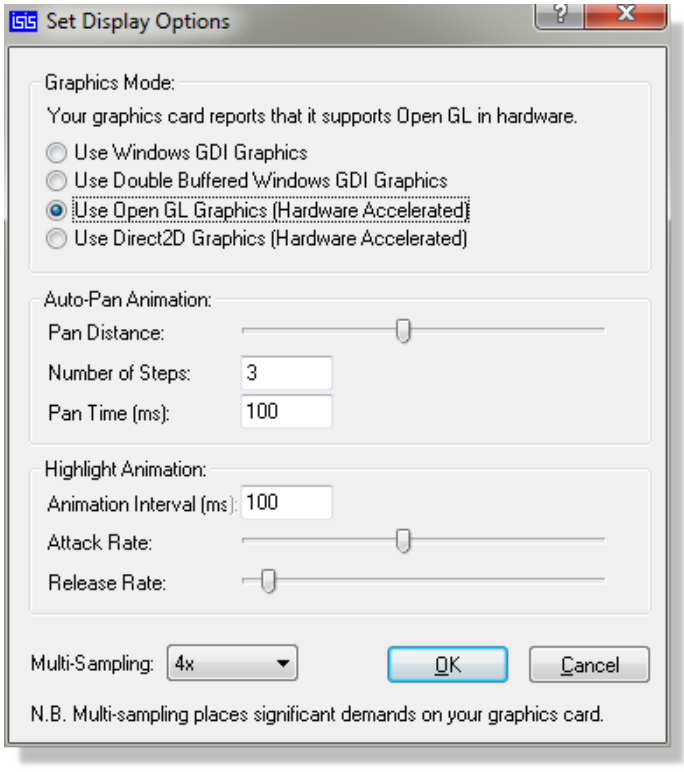
Display Options

ISIS is capable of harnessing the power of your computers graphics card to speed up operation and present a crisp and smooth display. However, as not all machines have sufficient graphics cards the software is also capable of using Windows to perform display and graphical operations. The available modes of operation are called:

- Windows GDI Mode.
- OpenGL Hardware Accelerated Mode.
- Direct 2D Hardware Accelerated Mode.

If your graphics card is not powerful enough to support hardware acceleration then ISIS will simply default to Windows GDI mode. Should your computers graphics card be capable of both Direct2D and OpenGL the system will default to Direct2D mode as this tends to be more reliably implemented by graphics card vendors.

Configuration of the screen display takes place from the Set Display Options on the System menu. Some of these options are specific to a particular hardware acceleration technology and will be disabled when a different mode is enabled.



The Set Display Options Form for OpenGL and Direct2D

The first section of the dialogue reports on whether your graphics card will support OpenGL and/or Direct2D hardware acceleration and, if so, allows you to switch between graphics mode.

The auto-pan section of the dialogue allows you to control distance, smoothness and speed of pan operations on the schematic. For example, holding down the shift key and bumping the mouse against the edge of the window will auto-pan the screen.

The highlight animation options allow you to specify the speed at which objects become active when the mouse is moved over them. The attack rate is the rate at which the object is highlighted and the release rate is the speed at which the object returns to its default state. These options are only available in hardware accelerated display modes.

Finally, the multi-sampling combo box allows you to specify the level of anti-aliasing you want when working in OpenGL mode. The higher the level of multi-sampling the better the anti-aliasing effect but the more GPU resources are consumed. If you select a level of multi-

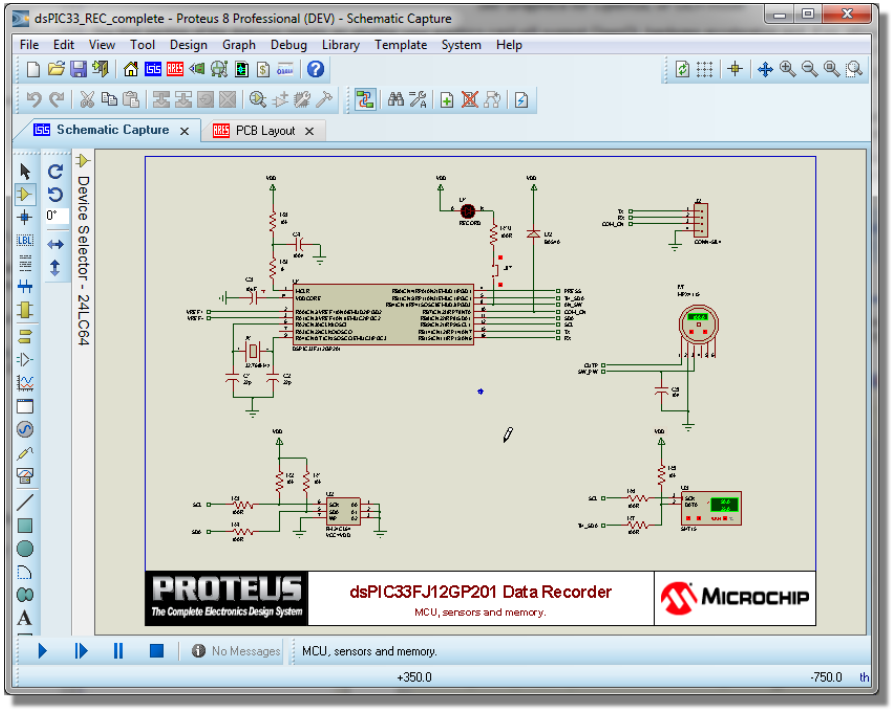
sampling which is not supported by your graphics hardware the software will reset the level to the closest one which your card can handle.

⚠ What you get with multi-sampling varies enormously between graphics cards. Some present extremely smooth text, while with others it looks blurry and on a few it causes display issues. We would suggest you being with multi-sampling off if you choose to work in OpenGL mode.

Configuration of colours and styles in ISIS takes place from the Template Menu. This allows to change everything from paper, grid and highlight colours to the thickness and colour of all the object types used in a design. Please refer to the reference manual section on Templates for more information.

Design Overview

The circuit we are going to draw is shown below. This is reasonably straightforward schematic that will nonetheless allow us to cover most of the major features of the ISIS schematic capture package. Being a real-world design, it will also allow users to follow the design through the PCB phase of development via the PCB tutorial.



The dsPIC33 Data Recorder Sample Design

General Description

The dsPIC33 Data Recorder Tutorial circuit is a solid state recorder designed to collect three of the most important environmental variables:

- Atmospheric pressure
- Ambient temperature
- Relative humidity

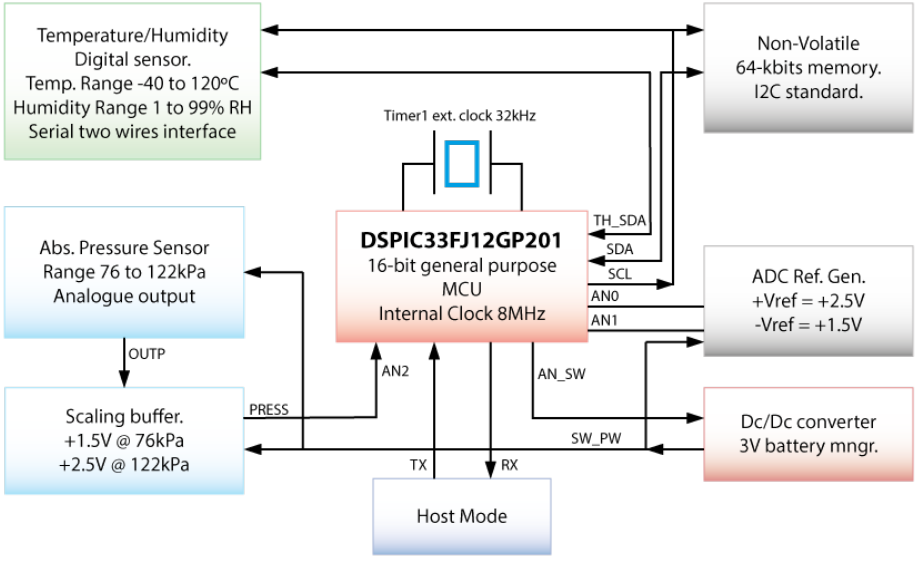
A number of low cost silicon transducers with built-in signal conditioning or digital interfaces have been used for measurement. This reduces the number of components to the minimum required for dsPIC33 interfacing.

All measurements get periodically stored into a non-volatile, low power memory in row binary format for later transfer to a host PC either by using a serial terminal or, more properly, a simple program (not supplied with the project). For the purpose of this project a serial terminal is used.

The circuit is designed to work with a battery for relatively long period in an unattended environment and, as such, a particular emphasis has been given to optimizing the power consumption and consequently the battery life. The recorder will sleep most of the time and will be woken up periodically only to accomplish the required recording operations.

Circuit Description

The following is a block diagram showing the behaviour and interaction of the circuit components.



Block diagram of the dsPIC Data Reorder Design

We'll cover the basics of the design construction as we progress through the tutorial but in order to keep the focus on the practical aspects of using the software, design decisions and theory will not be discussed in this document.

Basic Schematic Entry

We'll start the tutorial by familiarizing ourselves with the basics of schematic design; picking components from the libraries, placing them on the schematic and wiring them together.

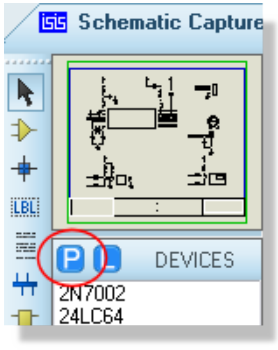
The design in question is relatively large and there is therefore a reasonable amount of drawing involved. We provide a completed schematic at the end of this section so, if you feel that you have mastered the basics at any point, there is no need to continue with drawing the remainder of the circuitry. We do however urge you to read through the full contents of the documentation as we introduce important features throughout.

The first thing we need to do is to get the parts from the libraries that we need in our schematic.

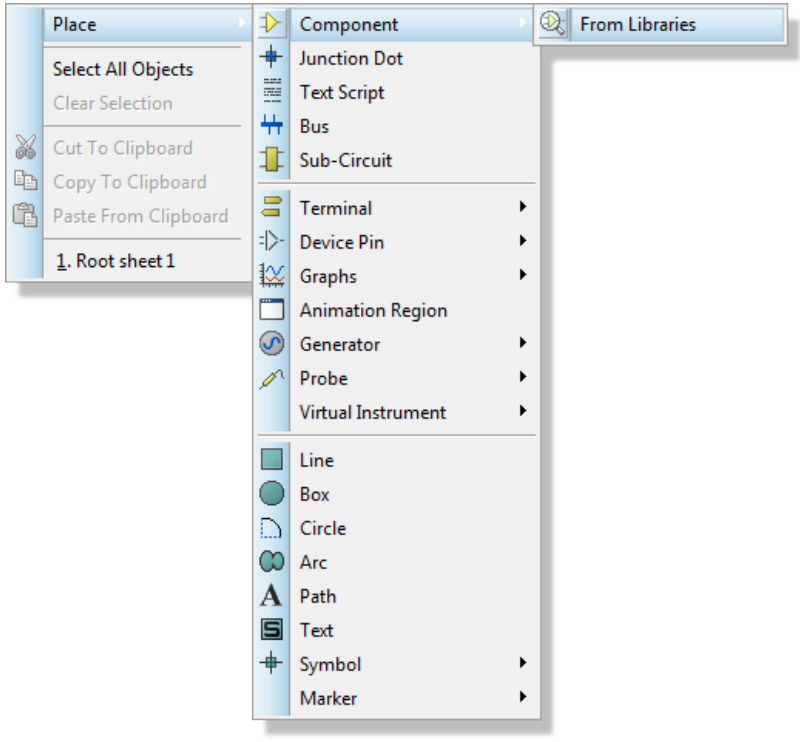
Selecting Parts from the Library

You can select parts from the library in one of two ways:

- Click on the P button at the top left of the Object Selector as shown below. You can also use the Browse Library icon on the keyboard shortcut for this command (by default this is the P key on the keyboard).



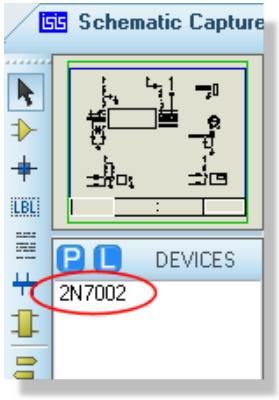
- Right click the mouse on an empty area of the schematic and select Place – Component - From Libraries from the resulting context menu as shown below:



Either of these two methods will cause the Device Library Browser dialogue form to appear. For reference, the following is a list of all the components we will need for our design:

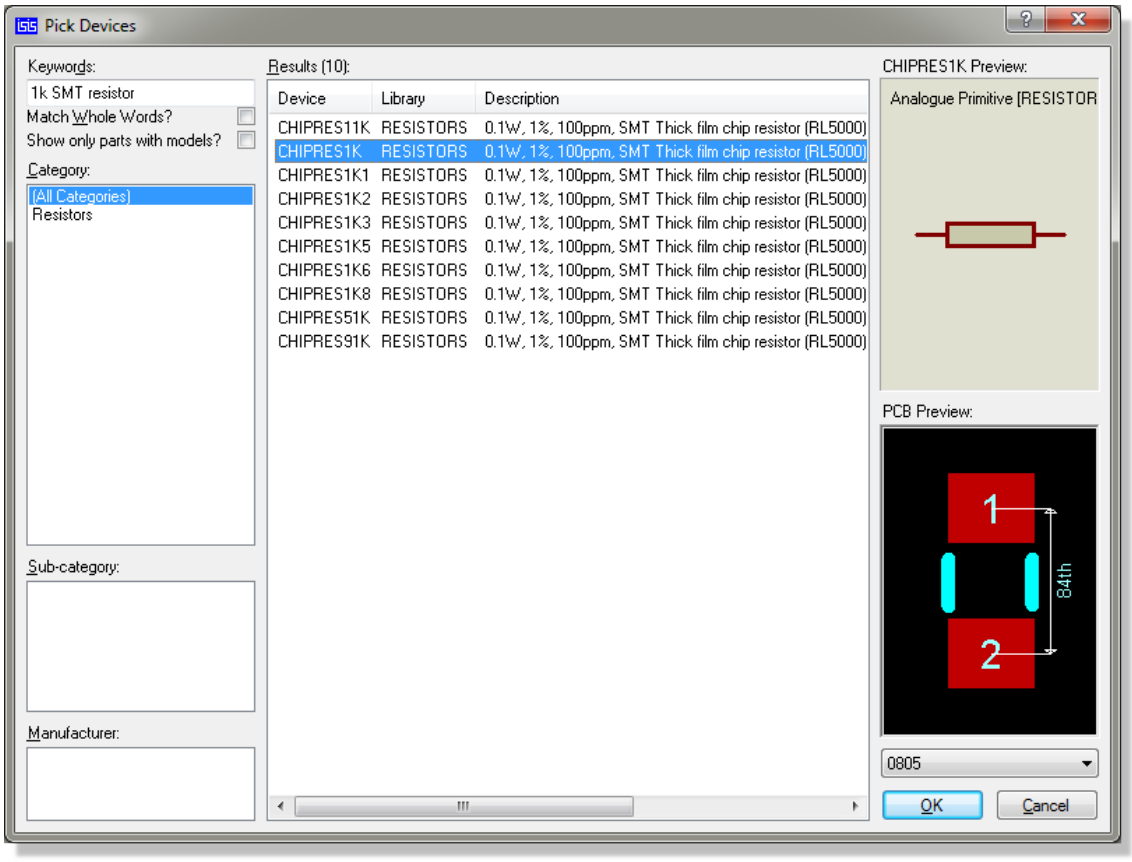
2N7002	24LC64	AVX0805NPO22P	AVX0805X7R1N	AVX08055X7R10N
AVX0805X7R100N	B82432T1103K000	BAS40	CHIPRES1K	CHIPRES10K
CHIPRES20K	CHIPRES68K	CHIPRES100K	CHIPRES100R	CHIPRES200K
CONN-H2	CONN-SIL4	CONN-SIL6	CRYSTAL	DSPIC33FJ12GP201
JUMPER	LED-RED	LM285-2V5	LM358	MAX1724
MPX4115	SHT15	TAWD106M025R0600		

There are several ways in which we can find and import components from the libraries into the schematic. In the case of parts where you would know the part name it's usually best to start the search with that. Try entering '2N7002' into the Keywords field on the Device Library Browser dialogue form. This gives a direct result and we can then simply double click on the part in the results list to import it into the schematic. When you do this you should notice that the part now populates the Object Selector as in the following screenshot.



The 2N7002 in the Object Selector

Given that we know the names of all the parts we want we could simply proceed by using this technique to bring in all the components we need. However, this may not always be the case and ISIS provides several methods for finding parts in the component libraries. One of the most intuitive is to use the library browser a little like an internet search engine, typing in descriptive keywords and then browsing the results to find a specific part. Try this now with the resistors, typing in '1k resistor' in the keywords field of the library browser dialogue to locate the CHIPRES1K component (double click on the part in the results list to import into the schematic). We could similarly search for '10k SMT resistor' to find and insert the CHIPRES10k component and so on.



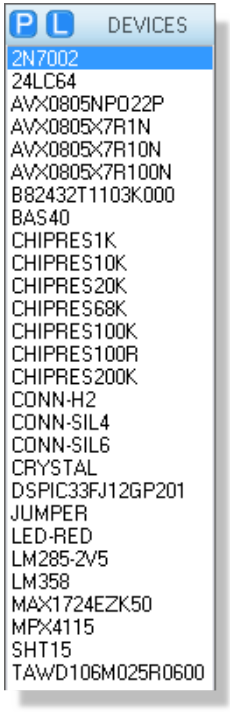
Picking a CHIPRES1K Resistor from the Pick Devices form

> You can customize the information displayed in the Library Browser's results list by right clicking the mouse on the results list. The context menu provides you with options to display Categories, Sub-Categories, Manufacturer and Library alongside each result.

It may be that we simply want to browse for a specific type of part and/or one available from a specific manufacturer. To take an example, clear out the contents of the keyword field and then select the Capacitors Category. In our design we are looking for some Nickel Barrier caps from AVX so we can further filter the results set by selecting Nickel Barrier from the sub-category field and AVX from the Manufacturer field. There are still a large number of caps available so we might type in '22p', '1N', etc. in the keywords field to isolate and select the particular components we require (AVX0805NP022P, AVX0805X7R1N and so on).

Whilst this exercise is a little contrived given that we are starting from a preset list of parts, the techniques described are flexible enough to allow you to locate any library part quickly. Work through the full parts list for the design (as provided above) now using one or more of these

methods to find and select the parts into your design. When you are finished you should have all of the required components in the Object Selector as shown in the following screenshot.



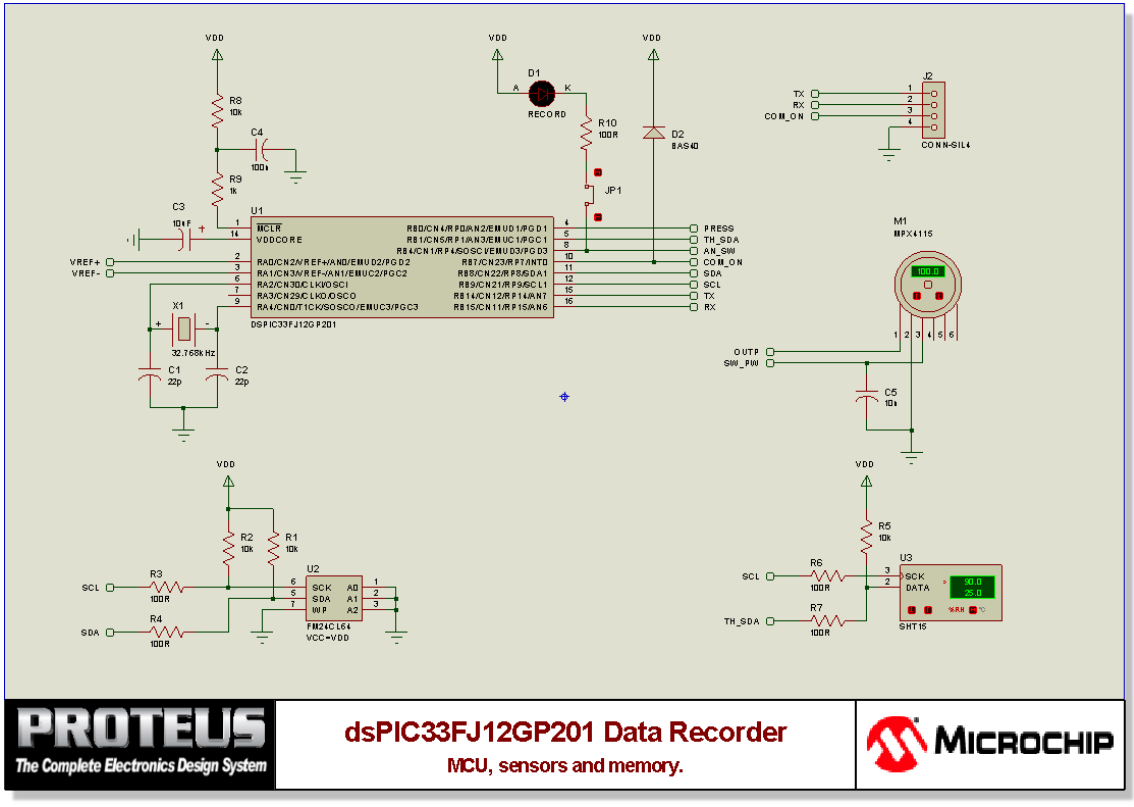
A Full list of required components

For example, if you wanted to search for a specific stockcode you would need to make sure that the stockcode column was enabled (right click on results list) and then type something like ‘digi <stockcode>’ to locate the digikey stockcode.

Similarly, if you want to search through only your own library parts, you can simple enable the library field and then type something like ‘user <partname>’; the user term will filter the results to the USERDVC library thus reducing the search range to your own libraries.

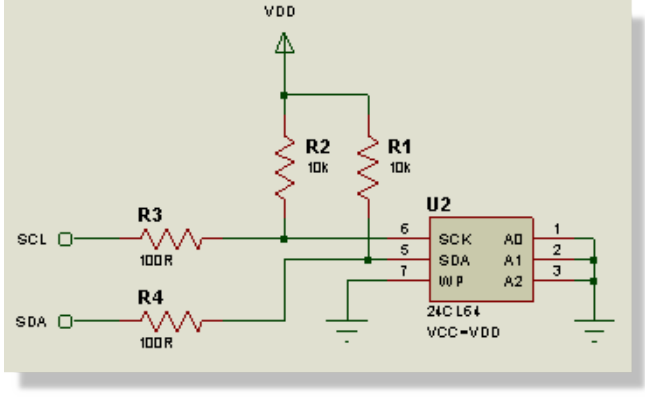
Placing Objects on the Schematic

Having selected the parts we need the next thing is to actually place them on the drawing area – the Editing Window – and wire them together. You’ll notice from the screenshot that we have also split the contents of the schematic into logical blocks of circuitry. This is partly aesthetic but also reduces wiring clutter on the schematic and will allow us to cover the use of terminals to form connections as we work through the tutorial.



Main Sheet of the dsPIC33 Data Recorder Schematic

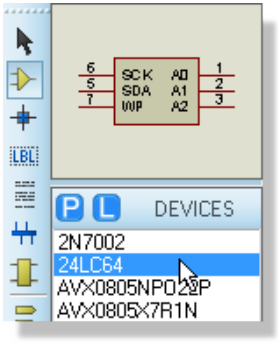
We are going to start off simply and complete the block of circuitry comprising the I2C Memory device as shown below.



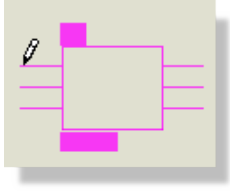
I2C circuit section

Begin by placing the I2C memory device as follows:

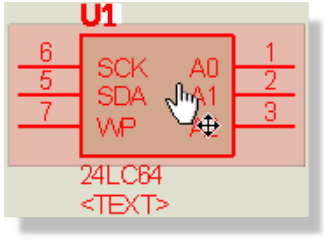
- Select the 24CL64 device from the Object Selector.



- Left click on the schematic to enter placement mode.



- Move the mouse to the desired location for the part, then left click the mouse again to 'drop' the part and commit placement.



Often we need to move parts or blocks of circuitry after placement and now is a good time to cover the different ways in which we can do that. The procedure for this should be familiar to most users; we need to select the object(s) we want to move, left depress the mouse, drag to the new location and finally release the mouse to drop.
We can select an object in several ways as detailed below:

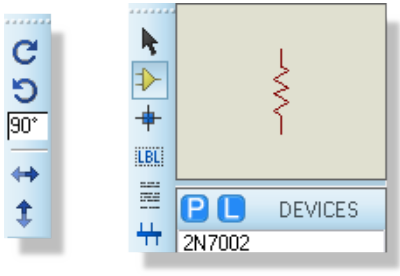
- Choose the Selection Icon and then left click on the object. This is a standard technique found in most graphical applications and will tag any object. Bear in mind when using this technique that you must change back to component mode for example, when you wish to perform other actions such as placing components etc.

- Right clicking the mouse on an object will both tag the object and present a context menu containing available actions on that object.

- Draw a tagbox around the object by depressing the left mouse button and dragging the mouse to form a box encompassing the object to be selected. This method will work for any object (or indeed sets of objects). Sizing handles are provided to allow you to resize the tagbox in the event that it does not fully enclose the object. This is the technique that should be used for moving multiple, connected objects or blocks of circuitry.

We'll get plenty of practice moving things around as we lay out the schematic; for now just use one of these techniques to move the memory device down towards the bottom left of the Editing Window in roughly the same position as in the screenshot at the top of the section.

Having placed the memory device, we now need to get the peripheral circuitry down and oriented correctly. We are going to need two 10k pull up resistors and two 100 Ohm resistors for the data and clock lines. Additionally, we are going to need to use terminals to achieve connectivity with power, ground and other sections of circuitry. Begin by selecting the CHIPRES10k device and click left once on the anti-clockwise Rotation icon (shown below); note that the preview of the resistor in the Overview Window shows it rotated through 90°.



Using the Rotation Icons with the Overview window

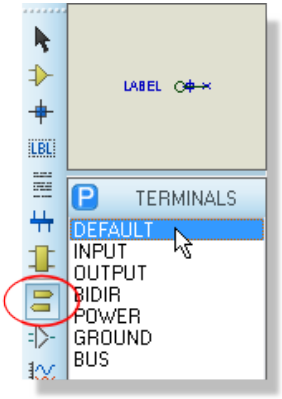
Now place the resistor above and to the left of the memory device in the same way as before. Then, simply left click again on the Editing Window to start placing the second 10K resistor just beside the first one.

Next, select CHIPRES100R, rotate appropriately (see above) and place two to the left of the memory device in line with the SCK and SDA pins.

- You can also rotate parts 'live' when in placement mode. Left click the mouse once to enter placement mode (at this point you will see the component outline following the mouse) and then use the '+' and '-' keys on the numeric keypad to rotate the component as you are placing it. Left click again to commit the placement in the normal way.

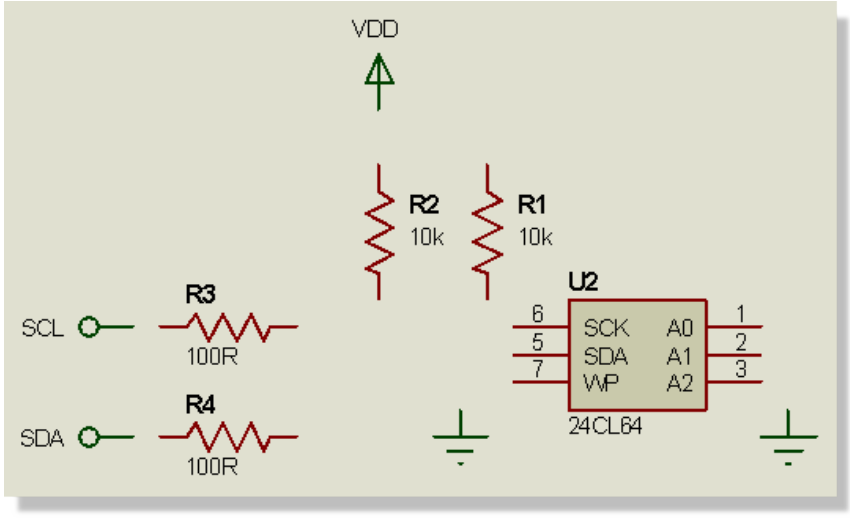
We use terminals in schematic design simply to terminate a wire and assign a connection. Often this connection is to either power or ground but it can just as easily be to another wire elsewhere on the circuit. Terminals allow us both to vastly reduce actual wiring (avoiding spaghetti schematics) and to make connections between different sheets on the schematic. To place terminals, start by selecting the terminal mode; this will switch the Object Selector and provide us with a listing of the available terminal types.

We need a power terminal, ground terminal and also two default terminals for the connections on the I2C bus. From this stage, placement and orientation are identical to any other object in ISIS and should now be quite familiar. Place the appropriate terminals in their approximate locations now, such that the area around the memory device now looks something like the following



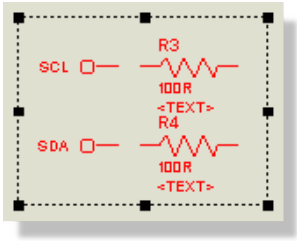
Selecting the Default Terminal

Unless you are fairly skillful, you are unlikely to have got all the components oriented and positioned entirely to your satisfaction at the first attempt, so now is a good time to practice moving things around as discussed earlier. In particular note that you can drag a tagbox around a group of objects to move the lot in one go.



Components positioned correctly

Similarly, you can clear a selection (or group of selections) either by left clicking in empty space or by right clicking in empty space and choosing the Clear Selection option from the resulting context menu



Dragging components

Remember that you can rotate while moving by using the '+' and '-' keys on the numeric keypad.

Wiring Up

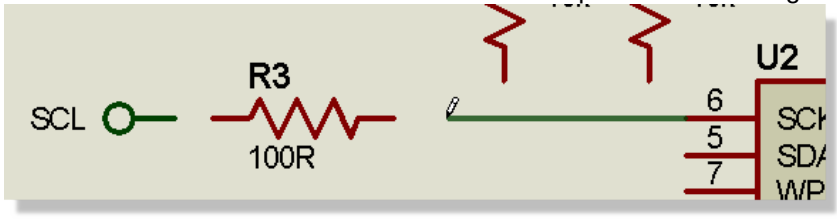
Having placed all the requisite components we now need to wire them together. There are three main techniques used to help make wiring a circuit as simple and quick as possible:

Modeless Wiring
There is no 'wiring mode' in ISIS - wires can be placed and edited at any time, without the hassle of entering a dedicated wiring mode prior to placement. This means less mouse travel, less mode switching and quicker development.

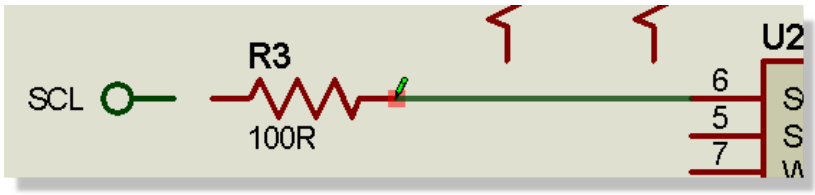
Follow-Me Wire Autorouting
After starting to place a wire, the proposed route of the wire will follow the movement of the mouse orthogonally to the termination point of the wire.

Live Cursor Display
The cursor will change as a visual indicator when wiring to show when a wire can be placed, when a wire can be terminated and when a wire is being placed.
The basic procedure for placing a wire between two pins is given below, using the connection between the SCK pin of the memory device and the 100Ohm resistor as an example:

- Move the mouse over the SCK pin on the memory device - the cursor will change to a green pen.
- Left click the mouse and then move it to the left until it is over the pin of the 100 Ohm resistor. The wire will follow the mouse and the cursor / pen is white during wiring



- Left click the mouse again to commit the connection and place the wire.

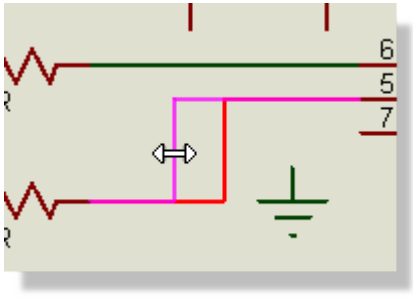


The procedure for wiring onto an existing wire is almost identical but there are a couple of items to note:

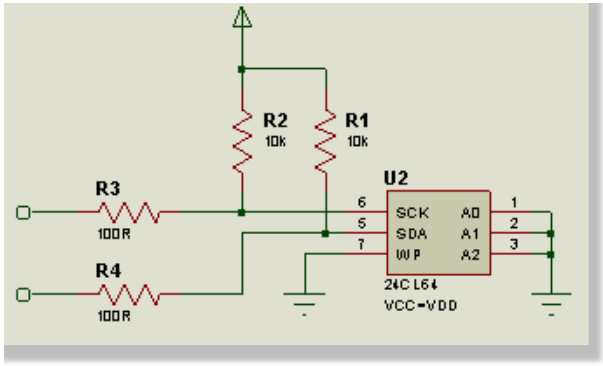
- You cannot directly start a connection from an arbitrary point on a wire; in our example you want to start the connections from the pin and terminate them on the wire.
- When you terminate the connection on another wire a junction dot will be placed automatically to complete the connection.

Dragging a wire

If, during a design, you want to make a wire-to-wire connection you must first place a junction dot on the wire and then wire from the junction dot to the other wire
If you want to adjust a wire after placement (for example, the wire from the SDA pin to the resistor) then simply right click on the segment you want to move and then either select the 'Drag Wire' context menu option or simply depress the left mouse button and drag to the new location.



Armed with the above you should now be able to connect up all of the circuitry, so that your schematic now looks something like the following



Schematic section with all parts / terminals placed

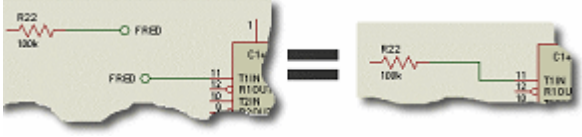
- Remember, ISIS provides visual indicators to help you. If the cursor turns green you can start/stop wire placement.

Connectivity by Touch

The following basic rules of thumb may be helpful:

- Iif you are placing the component you can attach to another component, wire or terminal directly.
- If you are dragging the component you can attach to another component pin, junction dot or terminal.
- If you are copy and pasting you can only attach it to a pin end, a bus pin, a junction dot or a terminal only.

Making Connections with Terminals
The final thing we need to do to complete this block of circuitry is label the terminals. Terminal naming is extremely important as it denotes the connection to be made. We could name the terminals in any fashion we liked but sensible names make the schematic more legible and easy to understand.



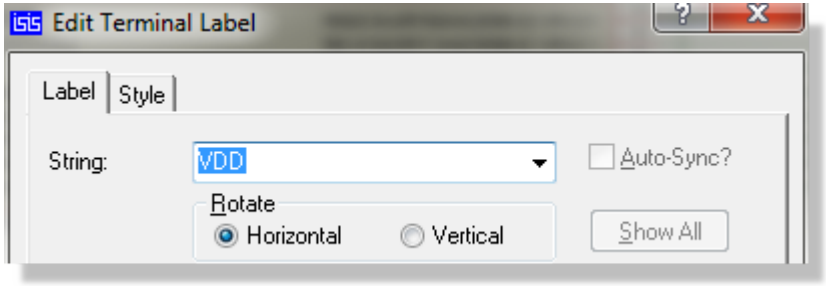
Terminal connection

- Power and Ground terminals are the exception to this rule, although there is no reason not to label them; an unlabeled power terminal is assigned to the VCC net and an unnamed ground terminal will be assigned to net GND.

Essentially what we are doing by labeling a terminal is making a connection to somewhere else on the schematic (a terminal with the same name) without placing a physical wire between the two objects.

As discussed previously ISIS is flexible enough to present you with several methods for editing parts - choose your preferred method for editing the terminal from the following:

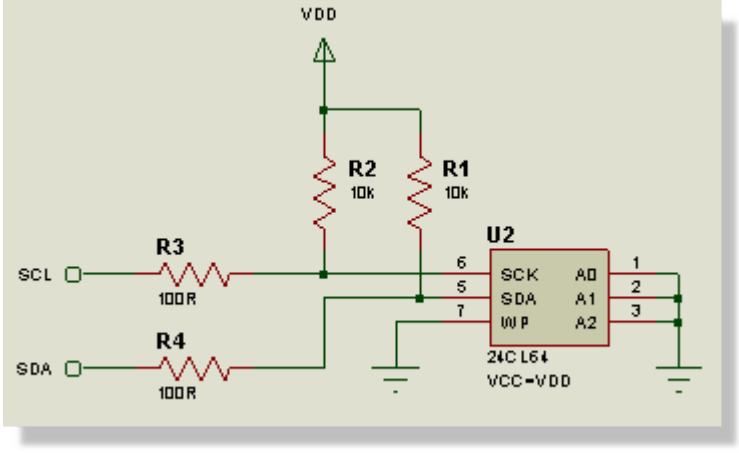
- Double left click on the terminal.
- Right click on the terminal to select it and launch the context menu and then use the Edit Properties menu option.
- Enter Selection Mode, left click to highlight the terminal, then right click to launch its context menu and use the Edit Properties menu option. Remember to exit selection mode when you are finished.



Having now launched the terminals dialogue form, type in VDD in the edit field as shown below and hit ok to exit the dialogue form.

- Note that, where appropriate you should always prefix numerical terminal values with a '+' or '-'. For example, a terminal labelled 12V is ambiguous and you should therefore ensure it is labelled +12V or –12V.

Finish the job now by editing the other terminals and labeling them appropriately such that your completed circuit block now looks like the following:



Schematic section completed

Power Connections

ISIS supports a powerful scheme for making power connections implicitly, thus vastly reducing the number of wires on the schematic. There are three main concepts involved that we will discuss in turn below.

Hidden Power Pins

Almost all relevant parts in ISIS have their power pins hidden (not visible on the schematic). The crucial point to remember is that in such cases is that by default the name of the pin denotes the net to which it will connect.

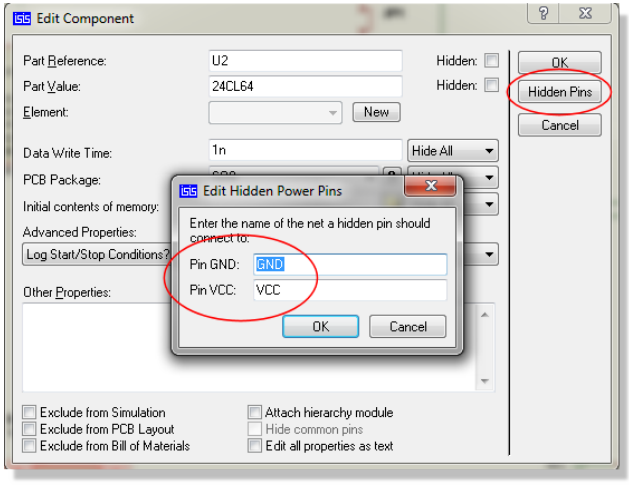
> For more information please visit Hidden Power Pins in the Online Help.

Power Rail Configuration

The easiest way to manage power nets and connectivity is through the Power Rail Configuration dialogue form, which is invoked from the Design Menu. If we do this now we can see that we have three pre-defined power supplies, namely GND, VCC/VDD and VEE. Changing the combo box at the top we should also see that the GND net is connected to the GND supply and that both the VCC and the VDD nets are connected to the VCC/VDD supply. It's worth examining how this happened more closely:

- The GND net is created by connections to an unlabeled ground terminal.
- The VDD net is created by connections to the power terminal labeled VDD.
- The I2C memory has two hidden power pins, VCC and GND, which are assigned to nets bearing their names.

Managing power nets and power supplies is a very important concept so we'll experiment a little to re-enforce the point. Exit the Power Rail Configuration dialogue form, right click on the I2C memory part and select Edit Properties from the resulting dialogue form. Next, click on the Hidden Pins button at the right hand side to view the hidden pins and their net assignments.

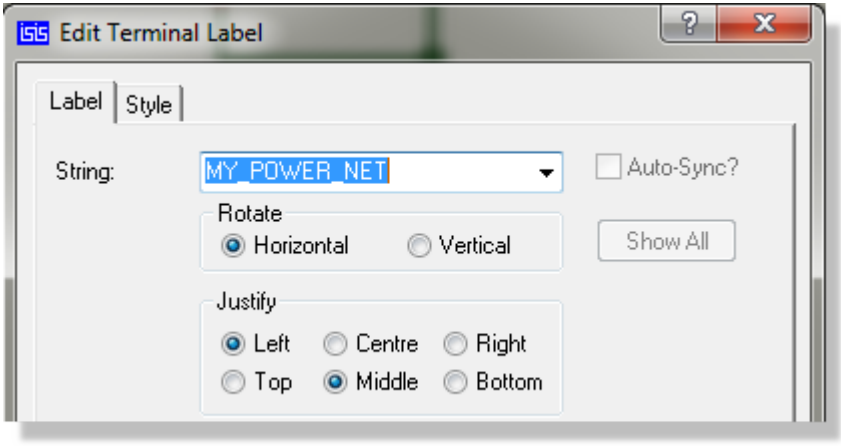


Hidden Power pins in ISIS components.

Let's change the power pin to be on net VDD, instead of the default of VCC. If we then exit the dialog and re-invoke the Power Rail Configuration dialogue and switch to the VDD/VCC power supply we should see that the VCC net is no longer present (this was the only connection on our schematic so far).

While we are here and for completeness change the Voltage of the power supply to be 3.3V, which is actually what we will be using. In PCB Design this is useful only as a reference to ourselves but actually has significance in the software for simulation purposes.
This exercise, while useful in explaining how to view/change the power nets for individual components, had no effect on design connectivity. The software is clever enough to assign both the VCC and the VDD nets to the same supply and the Power Rail Configuration dialogue allows us to configure our power connectivity at design level.

Returning to the schematic now, edit the VDD terminal (right click, edit properties) and change the terminal label to 'MY_POWER_NET'. If you go back to the Power Rail Configuration dialogue you should see that this is now an unconnected power net (a power net which is not associated with a power supply).



Changing the Terminal name

If we were doing this for real, we would now have to select the VSS/VDD power supply and then click the assign button to map our power net onto the correct supply.

> If we had simply removed the label from the power terminal (instead of renaming it), it would have been designated to the VCC net and again everything would be done for us.

A final point worth noting is that new power supplies are created automatically when you label a terminal with a voltage. For example, labeling a power terminal +12V would create a +12V supply and assign the +12V net to the supply automatically. This means again that no action is required by the user.

In general usage therefore, the software will handle the assignments of power nets to power supplies and will create new supplies automatically. However, if you need more flexibility (for example, an analogue ground and a digital ground) then using terminal labels and the Power Rail Configuration dialogue gives you the control you need. Please see the reference manual for more information if required.

Let's change our terminal label back to VDD for consistency before we move on.

- For more information please visit Power Rail Configuration in the Online Help.

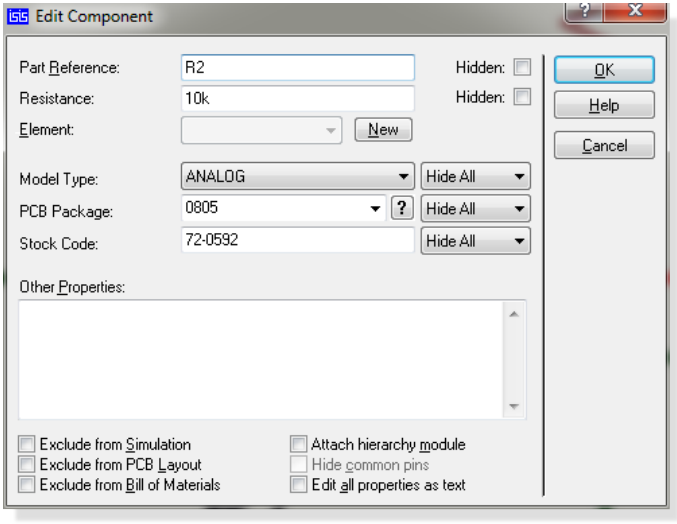
Global Power Nets
Power nets and connections are always global on a given sheet on the schematic. By default, they are also global to the design. However, if you do have a multi-sheet design and you want separate power supplies on different sheets you can isolate the supplies by un-checking the Global Power Nets box (Design Menu - Edit Design Properties dialogue form). This means that you need to explicitly wire through the sheets on the design but not within a given sheet. It is typically used only for the most complex of designs and is not relevant for the purposes of this tutorial.

Part Labels and Annotation

You should see that all the parts you have placed have both a unique reference and a value. A unique and sequential annotation is assigned to components as you place them on the schematic, although you can re-annotate manually if you need to.

You have full control over the position and visibility of part labels – you can change the values, move the position or hide information that you feel is unnecessary. The discussion below details how to manipulate part labels on a per component basis.

If you zoom in on any resistor you have placed you will see that ISIS has labelled it with a unique Reference (e.g. 'R1) and also with a Value (e.g. '10k'). You can edit both these fields and their visibility via the Edit Component dialogue form. Launch this dialogue form now by double clicking the left mouse button over the resistor (or using one of the alternative methods discussed previously).

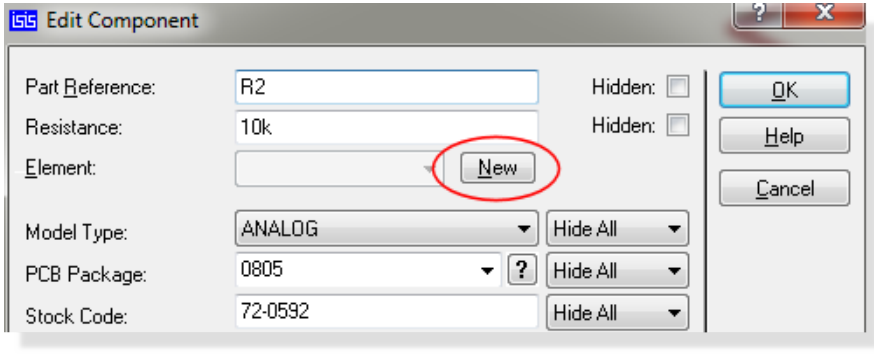


The Edit Component dialogue form

From the resulting dialogue form you can edit both the part name reference and, in this case, the resistance. Of more interest to us however, is the visibility options (highlighted in the above screenshot). It can be useful, particularly in densely populated schematics to hide some details displayed beside the parts – the tradeoff being that you would then need to edit the part as above to see its reference and value.

At this stage it is worthwhile familiarising yourself with editing components and hiding and showing references and values.

Some caution is required when changing part references in this manner. If, for example, you change ‘R1’ to be ‘R2’ then you will have two parts with the same reference on the schematic. This will cause netlist errors when working in the PCB module. If, however, you re-annotate using the new button on the edit component you are guaranteed a unique part reference.

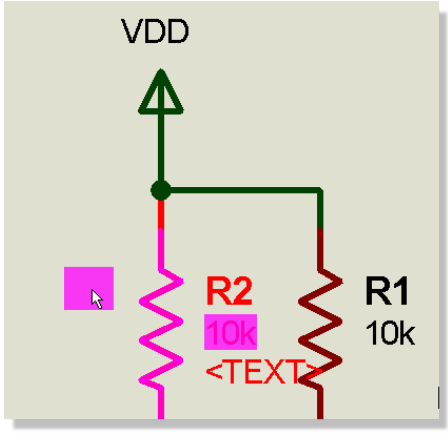


The 'New' button will automatically assign a new reference

Similarly, the Global Annotator (discussed later in the tutorial) will always avoid duplication of part references.

As well as the ability to hide part labels you can move them to a more convenient place. This is commonly used where you might want to place a wire over the current position of the label and need to free up some screen space. We will try this now with the 'R2' and the '10k' labels on the pull up resistor, moving them across to the other side of the component body.

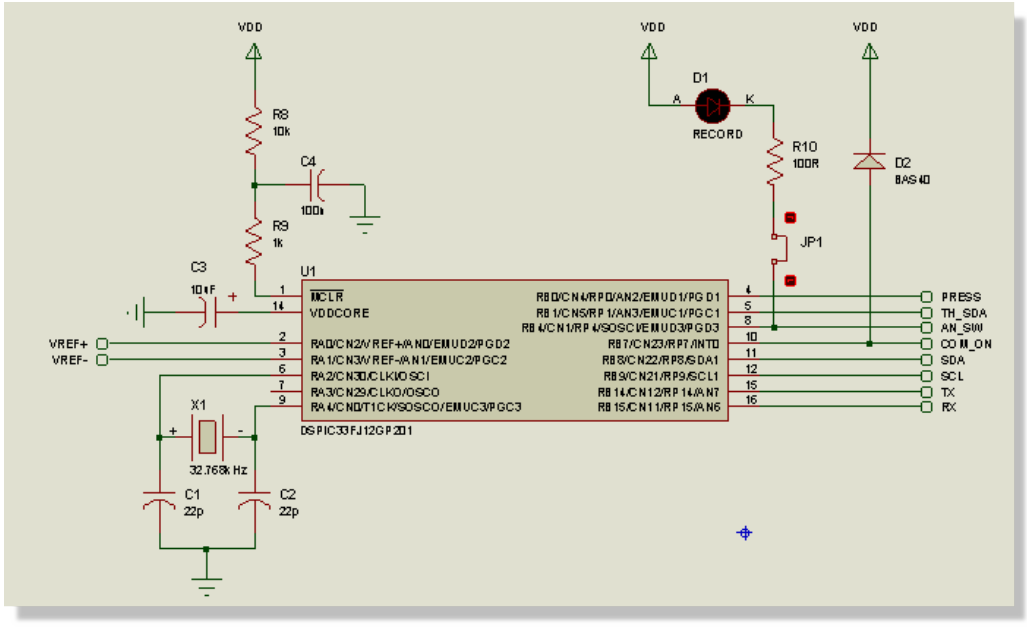
Start by selecting the resistor. Now point at the label 'R2' and with the left button depressed, drag it to its new position to the left of the component body. Then do the same with the '10k' label.



Dragging a reference

Timesaving Techniques & Block Editing

Armed with all of the basic techniques we can now look at the other blocks of circuitry on the front sheet of the design, practice what we have covered so far and introduce a couple of new timesaving techniques. Let's start by drawing the dsPIC33 block as shown here:



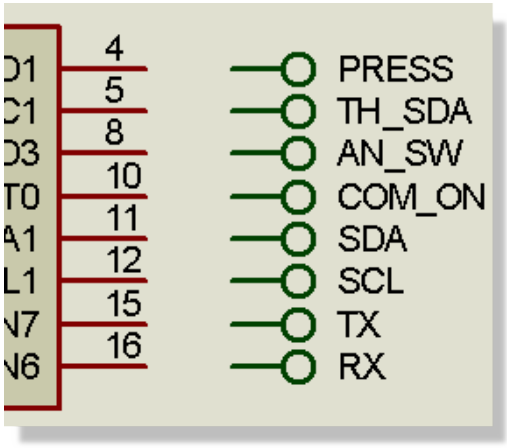
dsPIC33 Circuit section

You should be able to tell which components go where by comparing the component names in the Object Selector to the parts labels on the screenshot above. For clarity however, you can use the following table as a reference:

C1,C2	AVX0805NP022P	**X1**	CRYSTAL
C3	TAWD106M025R0600	**C4**	AVX0805X7R100N
R8	CHIPRES10K	**R9**	CHIPRES1K
R10	CHIPRES100R	**D1**	LED-RED
D2	BAS40	**JP1**	JUMPER
U1	DSPIC33FJ12GP201		

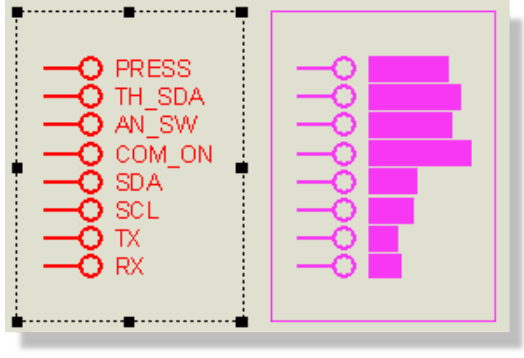
Begin by placing and wiring the oscillator and power sections (left hand side) of this circuitry using the methods outlined in the previous sections. When you are ready we'll look at the right hand side, where we can use a couple of additional features to speed up the process.

ISIS has a very useful feature to help with schematic wiring when we are laying out a series of topologically identical wires. Essentially, it allows us to auto-repeat the last wire placed to form a new connection.
Start by rotating and placing a couple of the terminals to the right of the processor.



Terminals belonging to the dsPIC33

Now drag a box around the terminals and select the block copy command; this will allow you to quickly copy out three more sets of terminals ready for connection. Right click the mouse when you are finished to exit block copy mode

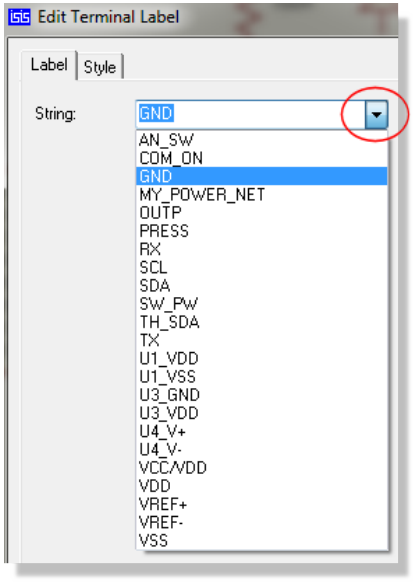


Block copying Terminals

Wire up the first connection from pin 4 (RB0) of the dsPIC to the terminal exactly as we discussed previously. Instead of repeating the process for the other seven connections simply hover the mouse over the next pin down (so that the cursor turns green) and double click. This will auto-repeat the previous connection, allowing you to rapidly wire up the other lines.

> Note that this technique works by placing an identical wire to the one placed directly beforehand. It follows that the destination must be exactly the same distance away (i.e. aligned) and that you must perform the operations consecutively

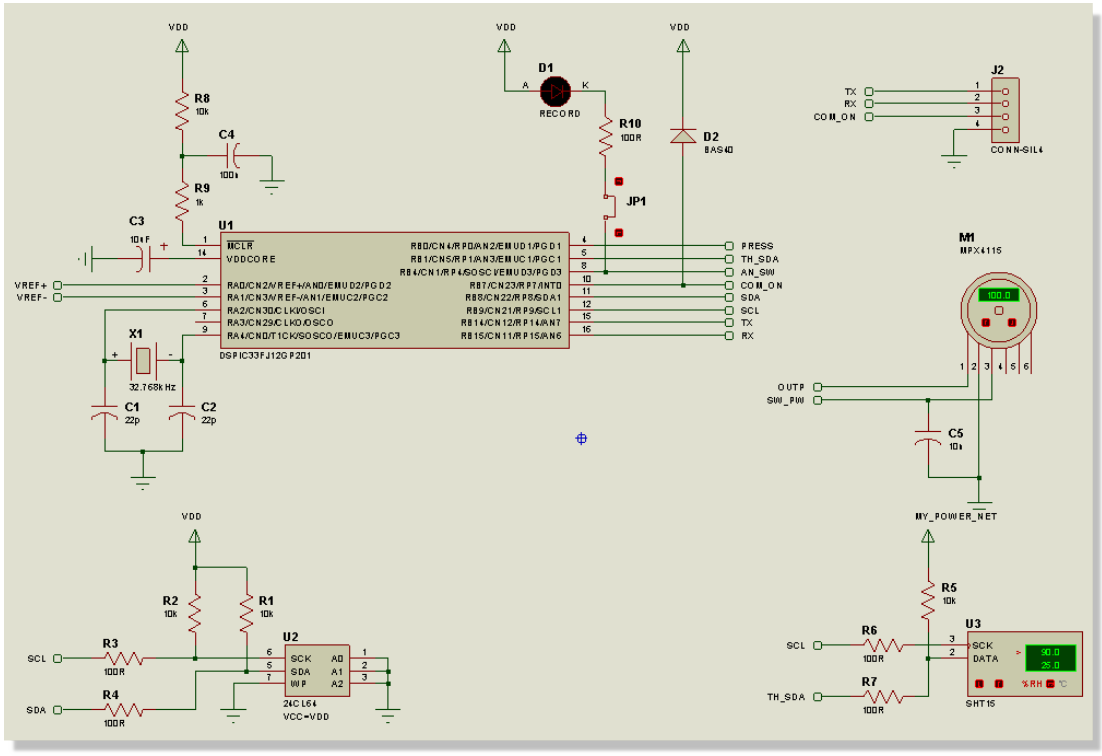
We can now complete this section of circuitry, placing and wiring up the remaining components and labeling the terminals as per the screenshot at the top of the section. Note that a list of existing terminal label names is available as a drop down list from the Edit Terminal dialogue form (refer to the section on labeling terminals if you need a reminder).



Drop down list of Terminals

It is highly recommended that you use this where applicable as it avoids the potential to make typing errors. Remember that terminal names donate connections so care should be taken during labeling.

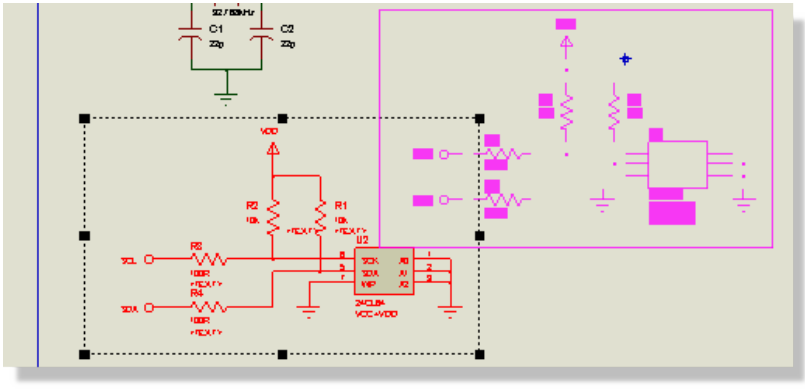
You should be getting quite proficient at basic schematic entry now and the other three small blocks of circuitry on this sheet should present no problem. When you are finished your schematic should look something like the one below:



Completed dsPIC33 circuit

If you feel comfortable with the topics we have covered to date, you can skip some of the drawing and load a pre-supplied completed schematic later in the tutorial

As a final point it is worth re-emphazing that you can re-arrange or move blocks of objects at any time by tagging a box around them and dragging them to the desired location.



Moving a block of circuitry to the desired location

Multi-Sheet Designs and Connectivity

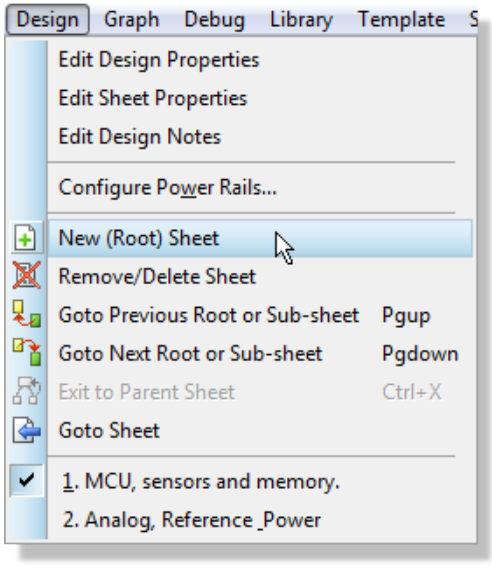
It is increasingly common in larger designs to split the schematic into multiple sheets. This serves both to reduce clutter on the schematic and also to organize the design into logical blocks. ISIS fully supports this methodology and we have arranged our tutorial design into two sheets in order to cover the relevant procedures. The work we have done so far has been to complete the first sheet (processor, sensors and memory circuitry), meaning we still need to cover the analog and power circuitry. We'll do this on a separate sheet.

While beyond the scope of this tutorial it is worth mentioning that Project Clips provide a method for re-using logical blocks of design and layout. More detail is in the reference manual and a short tutorial movie is linked on our website.

Also, note that what we are doing here is adding a second sheet at the same level as the first sheet. Proteus fully supports hierarchical designs where you use sub-circuits to contain logical blocks. This is discussed in detail in the reference manual.

Adding Sheets to a Design

To add a new sheet to the schematic we simply invoke the command from the Design menu as shown below.

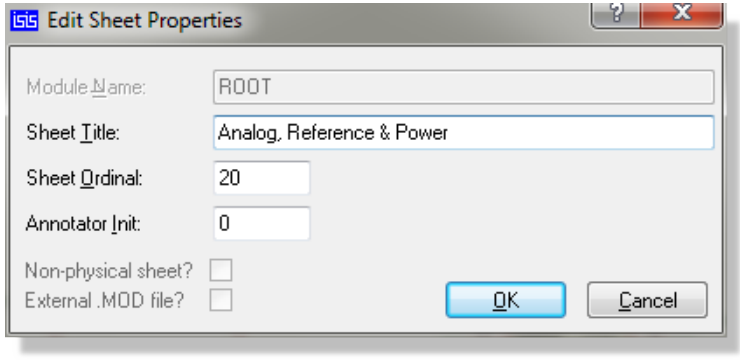


Creating a new sheet

Try this now to generate the second sheet.

Naming, Ordering and Navigating Sheets

Before we lay out the circuitry on this sheet we need to do a little housekeeping. While it's not necessary it is often helpful to name the different sheets on the schematic and also to order them in some sensible fashion. We can do both of these things via the Edit Sheet Properties command on the Design Menu.



The Edit Sheet Properties dialogue form

The Sheet Title is what will be displayed and should therefore reflect the contents of the sheet. In our case let's name it 'Analog, Reference & Power'.

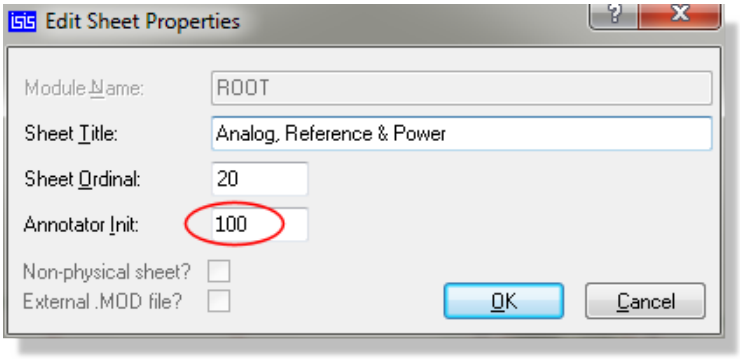
Sheets on the schematic are organized by Sheet Ordinal and by default these are 10, 20 and so on. A sheet with an Ordinal of 20 will therefore appear behind the first sheet (default Ordinal 10). If you prefer purely numeric nomenclature we can change the sheet name to be '2', to better reflect its position in the design structure.

> The default sheet names may seem inconsistent. This is because ISIS also has a Master Sheet, present in all designs, which allows the application of a company logo, revision number, author and other schematic information. This is beyond the scope of this tutorial but more information can be found in the Templates chapter of the accompanying reference manual (Help Menu – ISIS Help).

Sometimes, it is desirable to have a base annotation for a sheet; this means that the global annotator will increment from the value entered in this field. For example, if we set this to be 100 then the first resistor placed on the sheet would be annotated as R100. When left at its default value the global annotation will continue from its current position – this is fine for the purposes of the tutorial.

> Setting initial annotation for sheets is useful as it can guide the Global Annotator when batch annotating. It can - if you choose - also then be honored when annotating physically via the Automatic Annotator in the ARES Module.

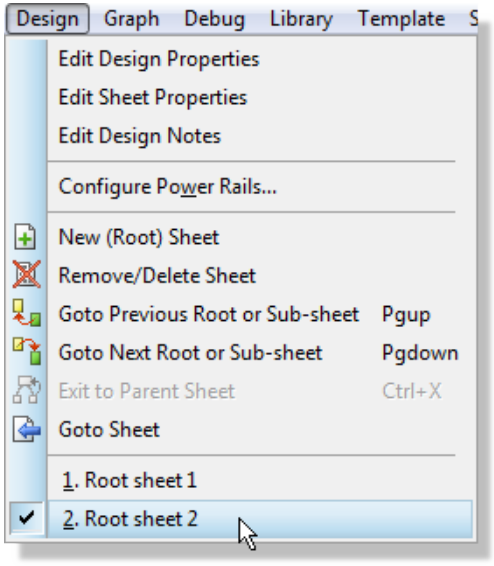
Before exiting, your dialogue form should now look something like the following:



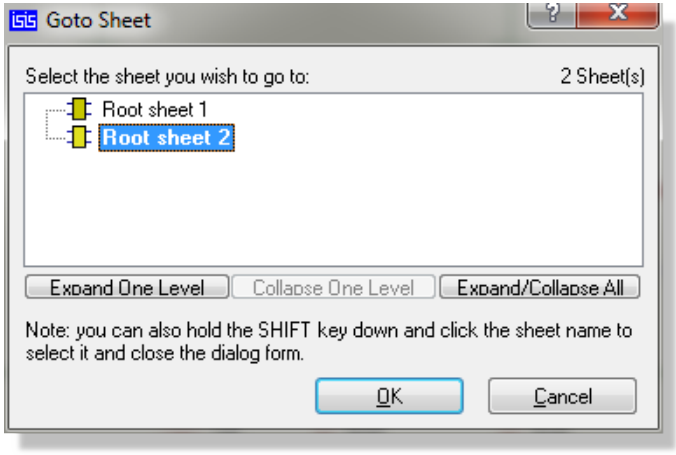
Edit Sheet properties with the Annotator set

For completeness we should also follow this process for the sheet we have just finished laying out. Before we can change the properties of a sheet we must first navigate to that sheet in the design and ISIS provides several ways to do this:

- From the Design Menu select Root Sheet 2 at the bottom of the menu.



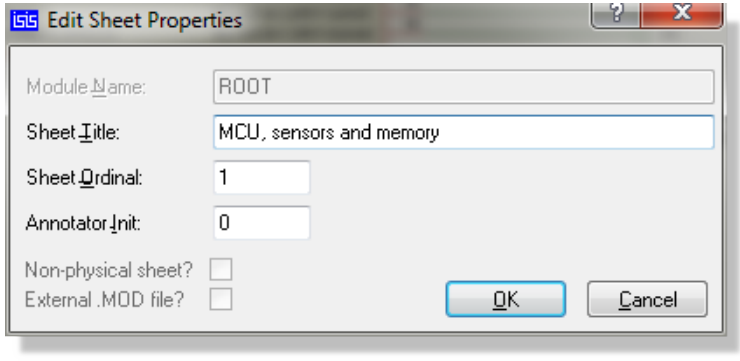
- From the Design Menu, select the Goto Sheet command, select the sheet and click OK to switch to that sheet.



- Use the PGUP and PGDOWN keyboard shortcuts to navigate between sheets. These cycle up and down the sheets respectively, although for a two sheet design they will seem to have the same function!

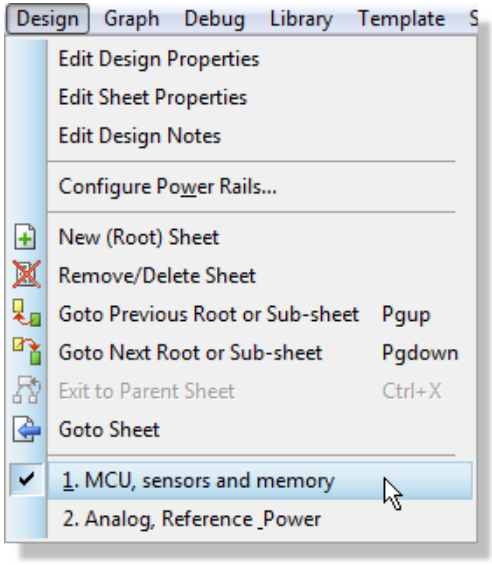
  - The Design Explorer is also a powerful way to navigate the schematic – this is covered in the Design Verification section of the tutorial and in more depth in the reference manual.

Once you find yourself back on the sheet we have laid out simply repeat the process above, with a sheet title of ‘MCU, sensors and memory’ and a sheet name of ‘1’.



Sheet 1 properties

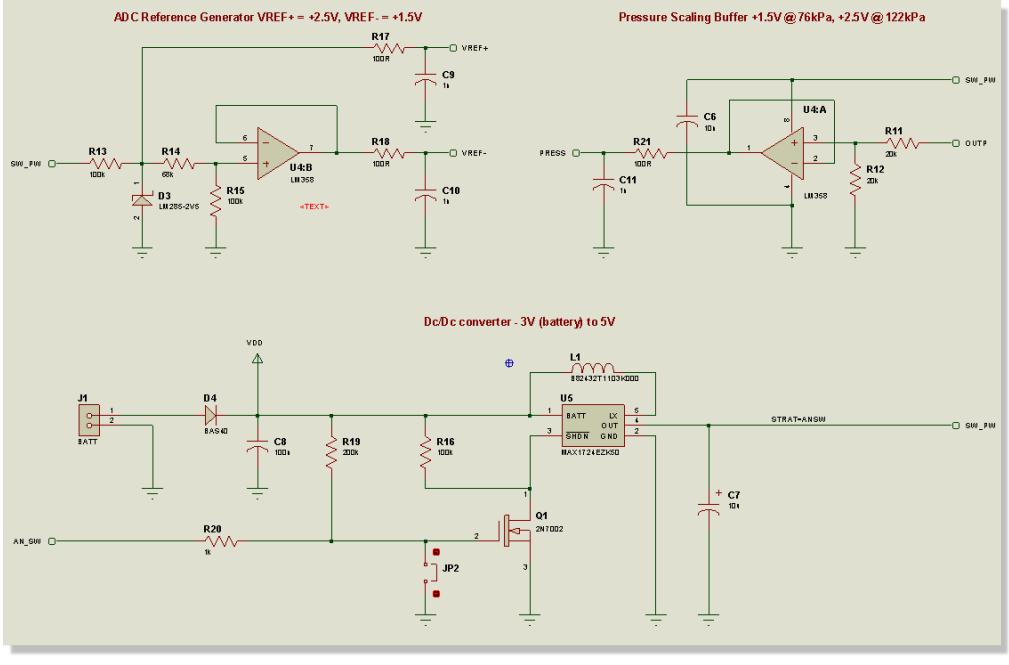
You should now see correctly organized and labelled sheets at the bottom of the Design Menu



This is what you should see in the Design menu

Connectivity between Sheets

Now that we have taken care of the naming, let's navigate back to our blank sheet and lay out the analog circuitry. The following screenshot shows the circuit we need to draw.



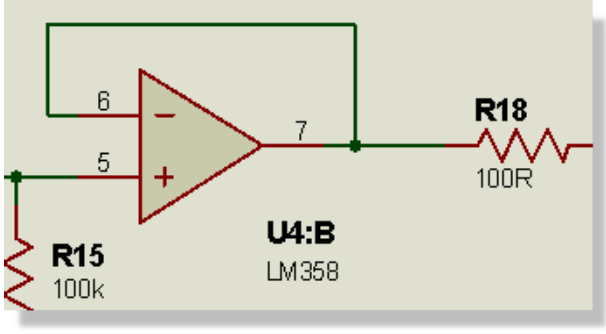
As we can see, there are three logical blocks of circuitry; a reference generator for the ADC module, an analog buffer for the pressure channel and a DC/DC converter to provide 5V from the 3V battery to the other two sections on each pressure sampling time.

If you feel that you need more practice then you can use all of the techniques we have covered so far to complete the schematic. Alternatively, if you are comfortable with the basics of schematic entry we will load a pre-supplied completed schematic in the next section. Regardless, there are a couple of points of note that are worth emphasizing.

Firstly, connectivity between sheets is achieved by having terminals with the same name on both sheets. From the screenshot above you can see the named terminals and, switching to the first sheet, the corresponding terminals with the same name. This technique allows signals to propagate not only across a sheet but also across sheets.

Secondly, it is possible when placing wires to 'guide' the wire placement as you place the wire. This is achieved by left clicking the mouse as you change direction, placing an 'anchor' to help the follow-me algorithm place the wire the way you want. Typically, this is useful on more complex wire trails but we can look at a simple example for the purposes of illustration.

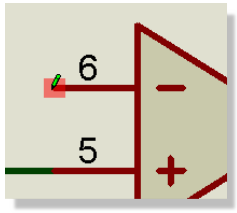
Consider the connection between the negative input of U4:B and resistor R18 as shown here:



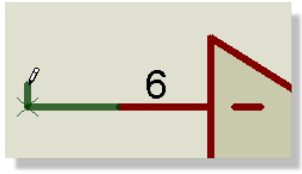
Negative input wire to R18

If we delete this wire (right click and select delete), we could replace the connection in the usual way and the follow-me algorithm would make a decent job of it. However, if we wanted to be more precise we could proceed as follows:

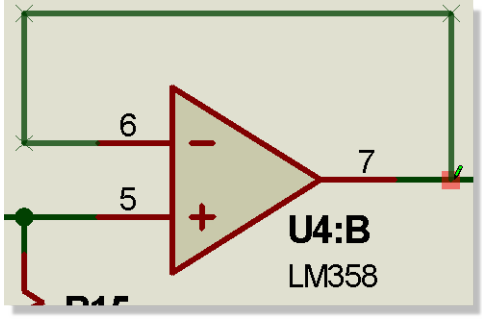
- Hover the mouse over the negative input such that it turns green and then left click to start wire placement.



- Move the mouse along to the left and then left click the mouse once to place an ‘anchor’.



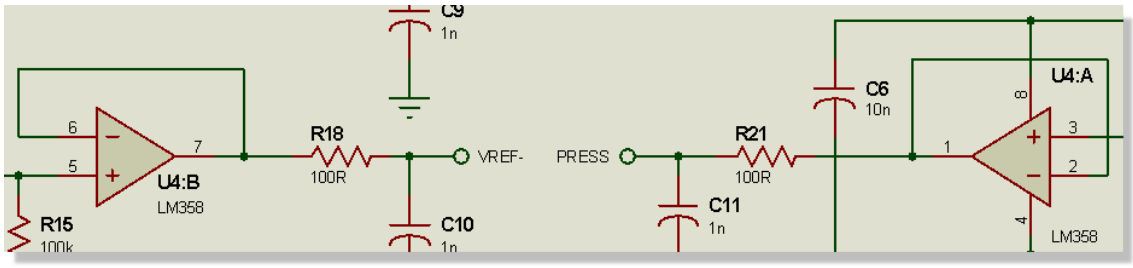
- Move the mouse up, left click to place an anchor, then right, left click to place an ‘anchor’ and finally down onto the wire between the op-amp output and resistor.



This technique constrains the follow-me algorithm and provides greater control during wire placement. If at any time during placement you change your mind and wish to remove an anchor simple right click the mouse once. Finally, if you want to completely disable the wire-auto-router and, for example, place a wire at 45 degrees, simply hold the CTRL button down on your keyboard during placement.

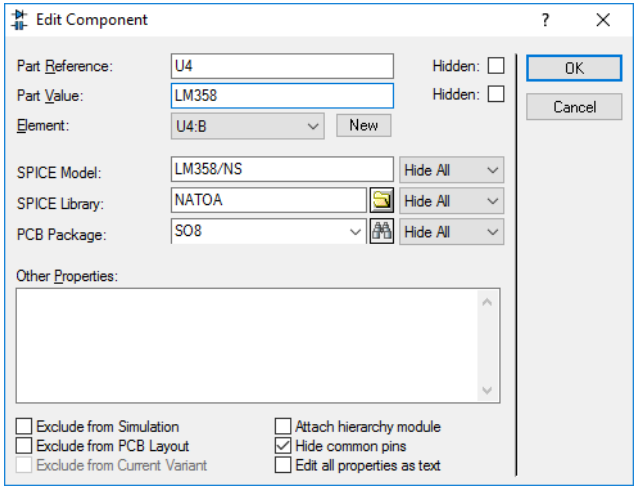
Gateswap

Now that we have a multi-element placed (LM358 dual op-amp) it's a good time to look at gate-swapping on the schematic. The reference for a multi-element part reflects both its part reference (e.g. U4) and also its gate element identifier (e.g. A) as shown below.



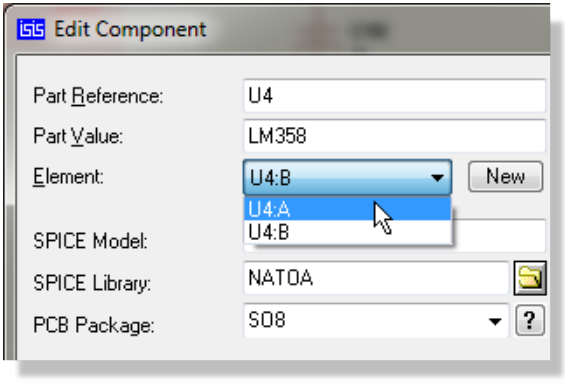
U4:A and U4:B

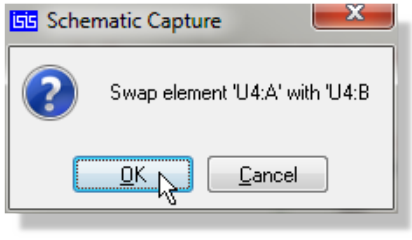
Start by editing one of the LM358 gates (right click - edit properties) to bring up the Edit component dialogue form.



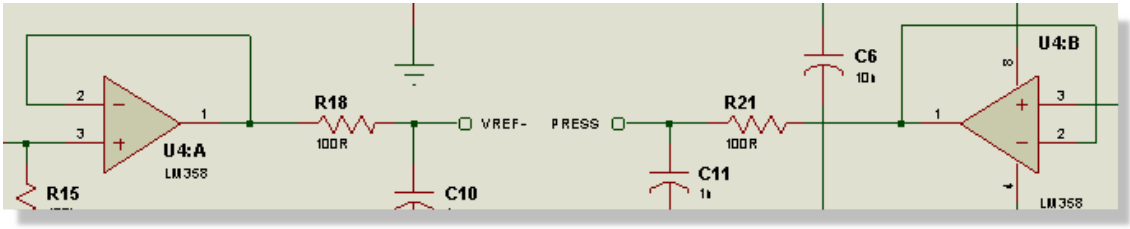
Edit Component Dialogue form for an Op-Amp

To swap the gates we are switching the element identifiers (A and B) on a single part (U4). The element field of the dialogue conveniently contains a combo box will all available gates so all we need do is change the element and hit OK to commit.





On schematic this switches the gates over according to the change specified in Edit component.



Preparing for PCB Layout

Now that we have completed the circuit we need to give some thought to PCB Layout and the information that we are going to provide from the schematic. A netlist basically consists of a set of footprint names (packages associated with schematic components) and connections (wiring on the schematic), although we can make things easier for ourselves later on by providing additional information.

If you do not have a completed schematic at this point or wish to work with the pre-supplied schematic you can load it via the Open Samples command on the Proteus home page. The file name is dsPIC33_REC_SCHEMATIC.pdsprj.

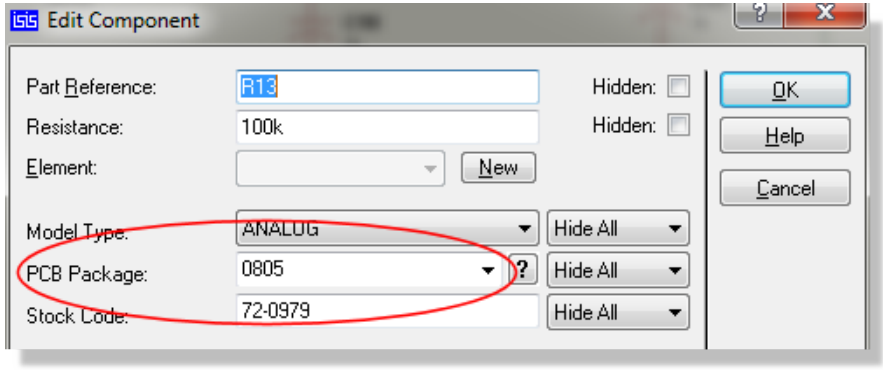
Packaging Considerations

Ideally, each component in ISIS should be associated with a footprint in the PCB editor (ARES). While you can do this at the point of placement in ARES best practice would be to sort this out in the schematic. Fortunately, the Proteus system provides a large set of pre-packaged components where this work is all done for us. As a case in point, the schematic we have just created requires no alteration or packaging work as the parts we have selected from the libraries already have footprints assigned to them.

However, it is possible that you may want to change the footprint associated with a part (e.g. PTH to SMT). We'll use the current schematic as a playground to explore how we can view and change packages to components.

To view the package associated with a component:

The footprint for a component is simply a property of the component in ISIS. We can therefore easily view the footprint by editing the component. Try this now by right clicking on one of the resistors on the schematic and selecting Edit Properties from the resulting context menu. You should see that there is a PCB Package property on the dialogue form and that the part is packaged with a standard 0805 footprint.



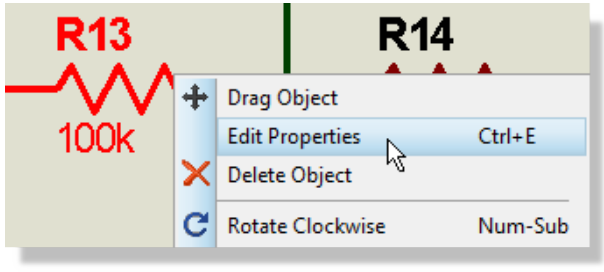
The new package appears in the Edit Component Dialogue form

- You can also see what the default footprint is for a component at the time you select it from the libraries. At the bottom of the Pick Devices dialogue form a preview of the footprint is shown for the currently selected component.

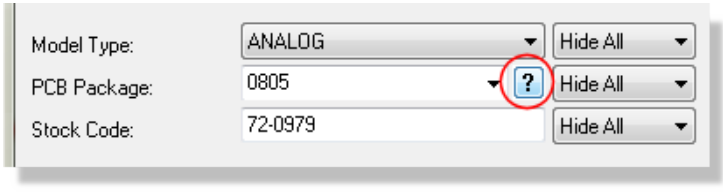
For convenience, components in the schematic often have packagings for more than one footprint. For example, if you launch the Edit Component dialogue form for the dsPIC33 component you will see that you have the option of either an SO18W or a DIL18 footprint. In this case the designer can simply change the footprint if required via the Edit Component dialogue form.

To change a package on a component

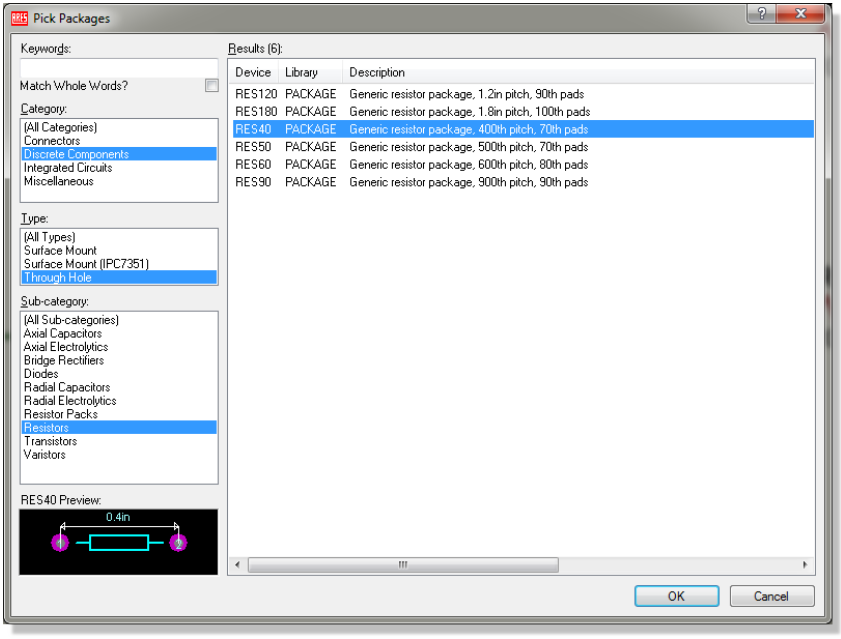
- Launch the Edit Component dialogue form by right clicking on the resistor and selecting Edit Properties from the resulting context menu.



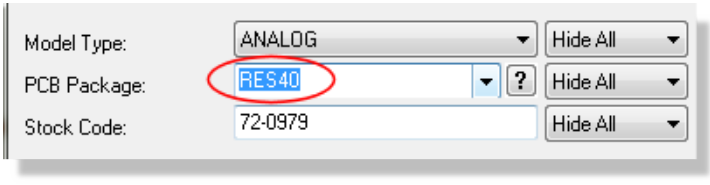
- Click on the question mark to the right of the package property to launch the footprint browser



- Clear out the text from the keywords field and then use the filters on the left hand side to narrow down the selection. We will want to select the 'Discrete Components' category with type 'Through Hole' and Sub-Category 'Resistors'. We can then select for example the 'RES40' footprint and click OK to commit.

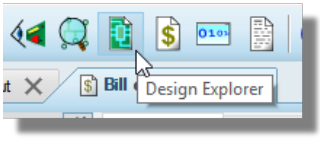


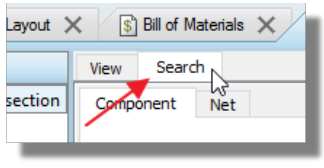
- You should now see that the RES40 is listed as the PCB Package for the component in the schematic.



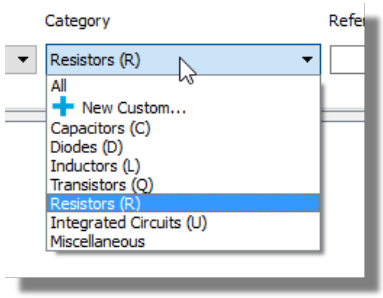
To change a package on all components of a particular type

If we want to make a wholesale change to parts already placed we can do so from the design explorer. First, open the design explorer via the icon on the main toolbar and then switch to the search tab:

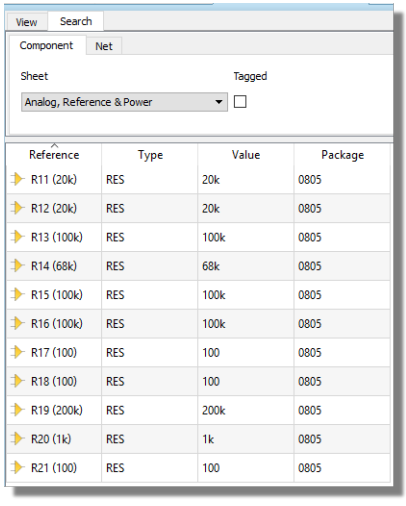




Following on from our example above, let's set category to be Resistor and hit search

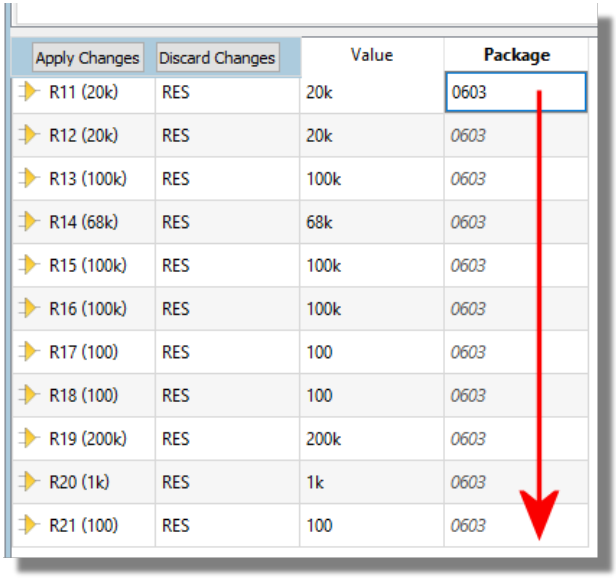


You should now see a filtered list of all the resistors on the schematic in a grid view. We could filter further by entering a resistor value.

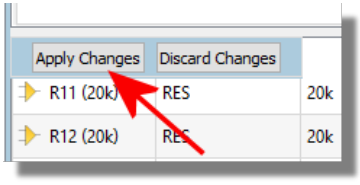


Reference	Type	Value	Package
R11 (20k)	RES	20k	0805
R12 (20k)	RES	20k	0805
R13 (100k)	RES	100k	0805
R14 (68k)	RES	68k	0805
R15 (100k)	RES	100k	0805
R16 (100k)	RES	100k	0805
R17 (100)	RES	100	0805
R18 (100)	RES	100	0805
R19 (200k)	RES	200k	0805
R20 (1k)	RES	1k	0805
R21 (100)	RES	100	0805

To change the package, simply type in the field at the top and then, much like Excel, drag the value down the other entries in the column.



Hit Apply Changes to commit the change.



⚠ Some caution is required with bulk editing like this because the software will accept your input without verification. It is however, extremely handy for changing passive footprints en-masse.

Connectivity Considerations

When we transfer our schematic to the PCB Layout software (netlist) all of the wires and connections are grouped into nets and passed through. ISIS actually does quite a lot of work for us behind the scenes at this point, automatically generating numerical names for the nets and also assigning the net to a default 'net class'.

Net names are assigned by ISIS according to the following rules:

- A group of connections which include a named terminal will be assigned a net name according to the name of the terminal.
- A group of connections which include an unnamed ground terminal will be assigned to the GND net.

- A group of connections which include an unnamed power terminal will be assigned to the VCC net.
- Any other group of connections is given an auto-generated numerical name.

What this means in practice is that all the work is done for us here; there is no real need for us to get involved in naming groups of connections. It is important however, that we understand the rules for unnamed ground and power terminals. If in any doubt, you should always name your terminals explicitly.

If a net does not have a name from a terminal you can if you wish explicitly name a net by placing a wire label with syntax (NET=NAME). This is useful if you wish to place a power plane on the net in ARES and otherwise helps to distinguish the net from other nets with auto-generated names.

A 'net class' is simply a group of nets which will have common properties in the PCB Layout. Specifically, all connections in a net class will have the same trace width and will obey a single set of rules with regard to clearances and design rules. Again, ISIS will do a considerable amount of work for us and will assign net classes as follows:

- Any net with a power or ground terminal is assigned to the POWER net class.
- Any net with a bus is assigned to the BUS net class.
- Any other net is assigned to the SIGNAL net class.

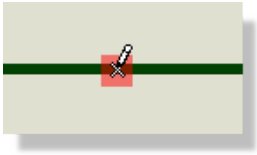
This gives us quite a bit of flexibility during PCB layout as we can define different trace widths, via styles, clearances and so on for each net class. Sometimes, however, we will need a particular set of conditions for particular connections. ISIS allows us therefore to create our own net classes and these will then be passed through to the PCB Layout software, allowing us to define routing and clearance properties specifically for those connections.

Let's take an example. On the tutorial schematic consider the output of the DC/DC converter on the Analog sheet of our design. This is the 5V switched power supply for the analog circuitry so we really want to lay this out with a track width smaller than that on the POWER net class but larger than on the SIGNAL net class. The procedure for specifying a new net class is very straightforward.

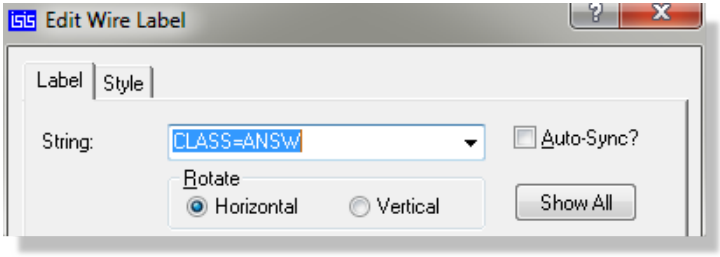
To specify a new net class

- Select the Wire Label Icon.

- Left click the mouse on a wire we wish to assign the net class to. Any wire on the net will do – in our case let's place it between the output of the MAX1724 and the SW_PW terminal. You will see a small 'x' under the cursor when the mouse is over the wire.



- Left click the mouse over the wire to launch the wire label dialogue form. Then simply type in 'CLASS=ANSW' to assign the net class.



If you have completed drawing the schematic follow this procedure now to add the net class; this will already be in place if you have loaded our pre-supplied schematic. If you work through the PCB layout tutorial, you'll see how to configure routing styles and design rules for the net class.

Design Verification

It is always a good idea to spend a little time checking the schematic before we move through to PCB layout. ISIS provides a unique and extremely powerful tool in the form of the Design Explorer that will help us catch any errors before we sign off on the schematic phase of design. In this tutorial we will introduce the tool and cover some basic functionality but we do recommend that you read the section in the reference manual (Help Menu – ISIS Help) and spend some time familiarizing yourself with its many capabilities.

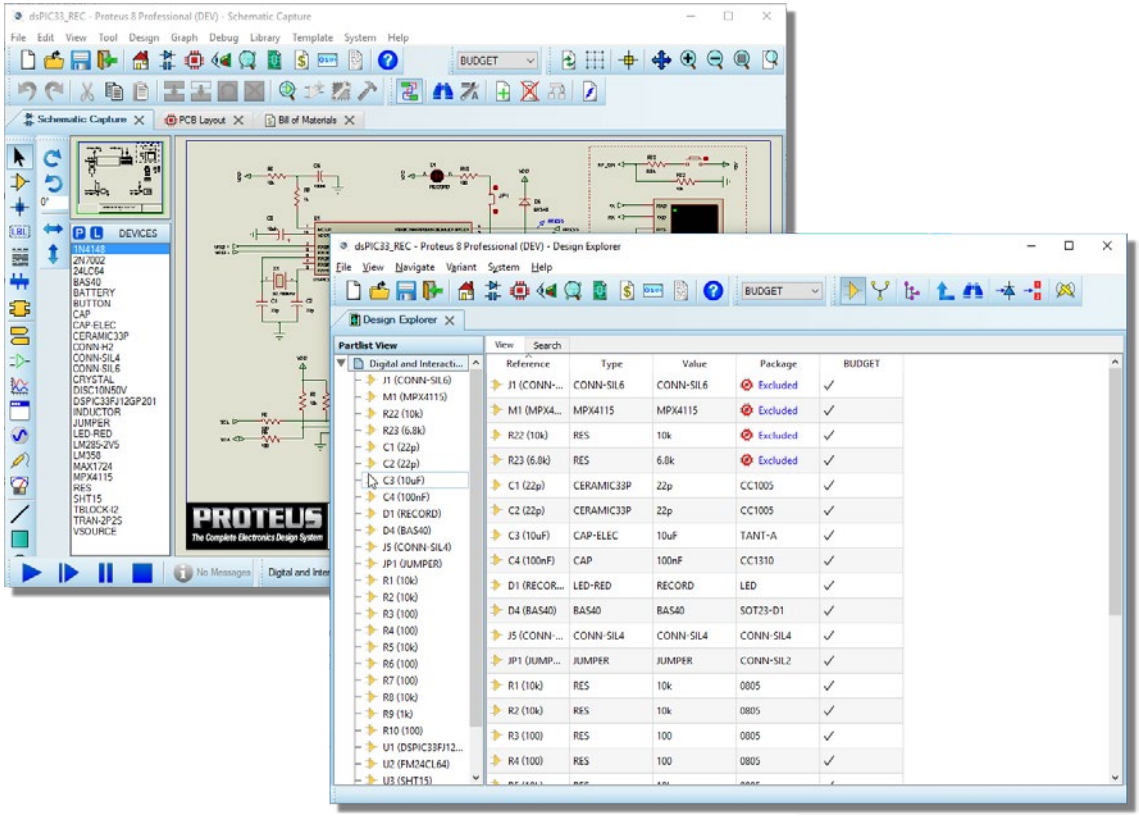
- Note that what you see in the Design Explorer throughout this section will depend on whether you have drawn the schematic yourself or loaded the pre-supplied one from disk.

The Design Explorer

The Design Explorer can be launched from top level application module toolbar and will appear as a separate tab inside the Proteus application.

The Design Explorer icon

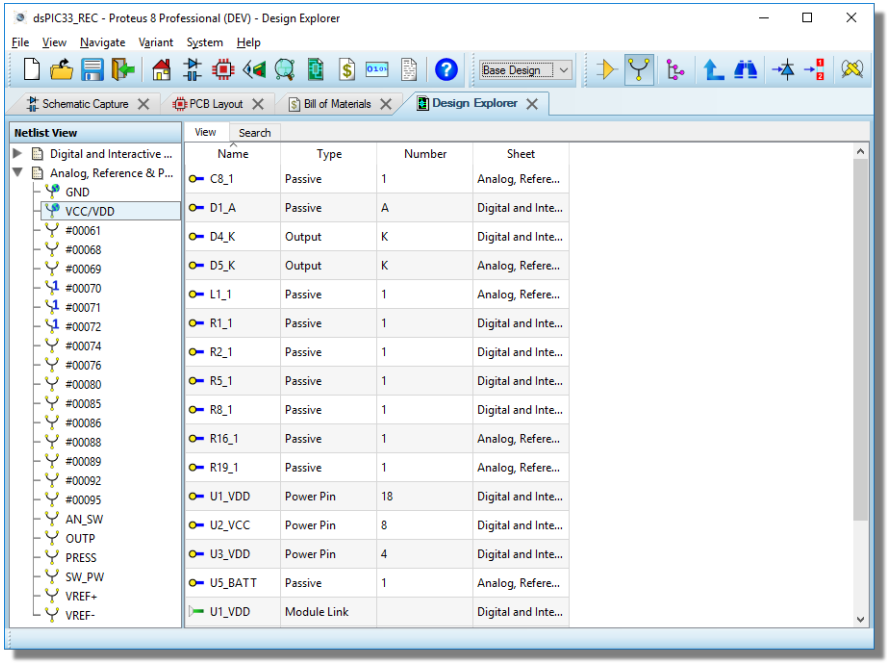
It is worth mentioning that, in terms of your Proteus workspace, you can detach tabs into separate frames. This is particularly useful if you want to view more than one module at the same time and have two monitors or a lot of space on screen. To split a tab onto its own frame simply drag and drop it onto an area of free space. This will allow you to view both the Design Explorer and the Schematic at the same time.



Split programs in Proteus

Alternatively, if you do prefer to work with tabs in a single frame you can quickly cycle between tabs using the standard Windows CTRL+TAB keyboard shortcut.

Once launched and in its default mode, you should see a dialogue form that looks a little like Windows™ Explorer. The left hand pane displays the sheets and the right hand pane displays the contents of the currently selected sheet. Clicking on another sheet in the left hand pane will force the right hand pane to show the contents of that sheet selected.

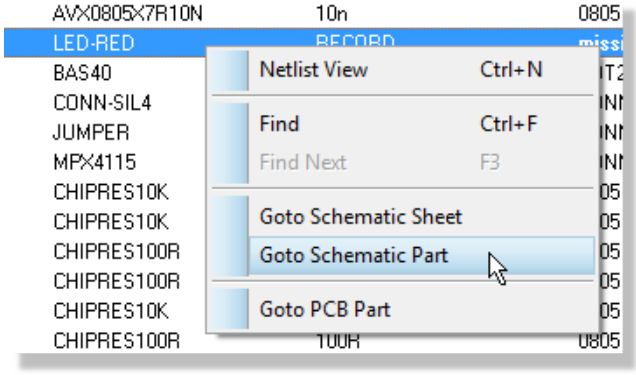


The Design Explorer

The Design Explorer has two distinct modes; partlist mode and netlist mode. The former will show a physical representation of the sheets (the components on a sheet) whilst the latter will show how the design is organized into groups of connections (nets). You can switch between the two modes via the Design Explorer Icons.

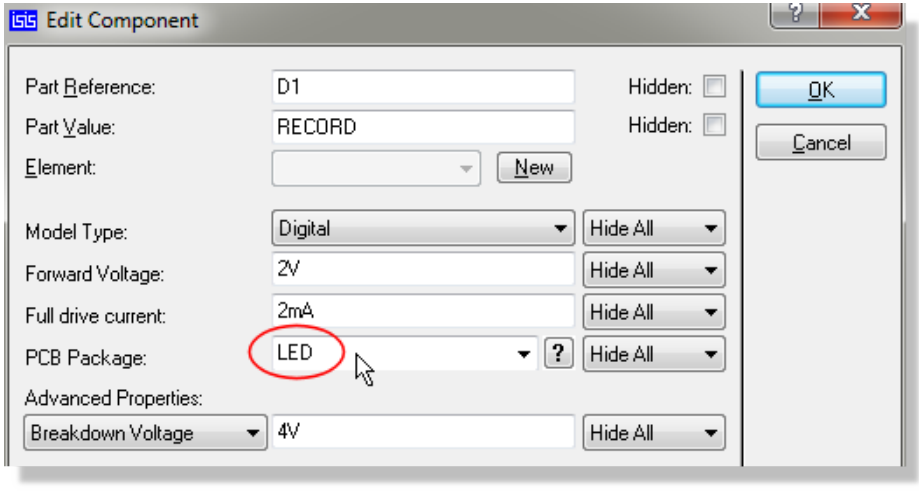
Design Explorer icon bar

Since we are in partslist mode by default let's look at what we can do from there. You may remember that earlier in this topic we discussed how to check the packaging for a single part. The Design Explorer shows us the bigger picture and allows us to make a global check of the packagings on the design at a glance. All of the footprints used are displayed beside their schematic reference at the far right hand side of the right hand pane so all we need to do is scan down the list to make sure everything is packaged, switch sheets (via the left hand pane) and repeat the process. If a part on the schematic did not have a footprint assigned it would have bright red text stating 'MISSING' in this field to highlight the potential problem.
A quick scan will immediately show that we do not have a packaging associated with the LED schematic part. Right click on the Design Explorer on the line with the LED and select Goto Schematic Part from the context menu



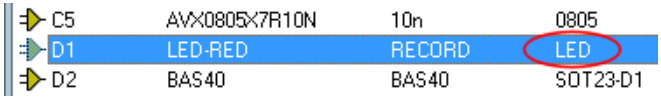
The Design Explorer's context menu

This will zoom the schematic around the part and tag it for editing. Simply right click on the highlighted part and select Edit Properties in the normal way. We could follow the procedure that we discussed earlier to find a suitable footprint but in this case take it as a given that it is named LED and simply type in LED in the PCB Package field of the dialogue form.



LED Assigned as the package

After making the change, the Design Explorer will live update to reflect the current status of the schematic.



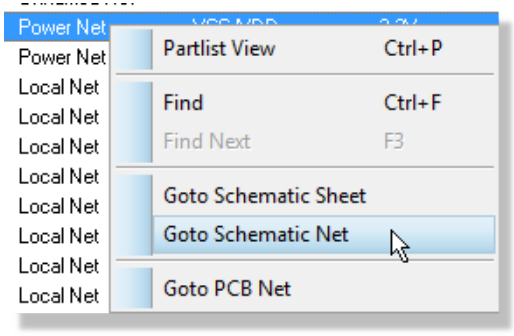
Design Explorer showing the LED packaged

Now, simply follow the same procedure as above to change the footprint to an 0805.

We can also use the Design Explorer to investigate the connectivity of the schematic – for this we need to switch into netlist mode via the icon at the top left of the dialogue form. This will give us a view as shown below.

The Netlist icon

The first point to note is the net names which, as we discussed earlier, are either named according to terminal or label or simply assigned a number if the net consists only of wire connections. We can right click on any net to view its connections on the current sheet by selecting the 'Goto Schematic Net' option from the resulting context menu. This is particularly useful when we want to check single pin nets for example, or if we want to name a net manually (see the section on connectivity considerations earlier in the tutorial). Finally, where we have a net or nets grouped with a named net class (far right column on the right hand pane) it can be useful to check our assignments.



Goto Schematic Net will highlight the selected net

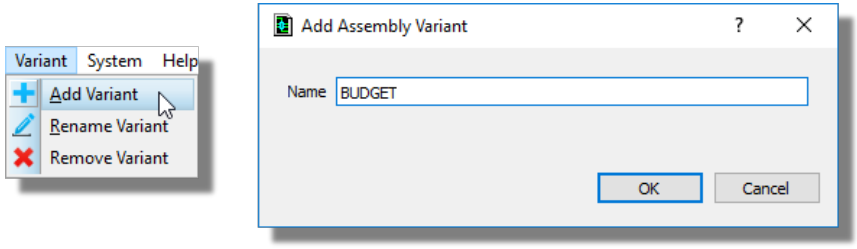
The Design Explorer is an extremely powerful tool and has many different uses throughout the lifecycle of a typical project. In particular it can be used as a link between the schematic and the PCB, facilitating cross-probing and PCB lookup. We recommend that you read and work through the chapter in the reference manual (Help Menu – ISIS Help) for more information.

Assembly Variants

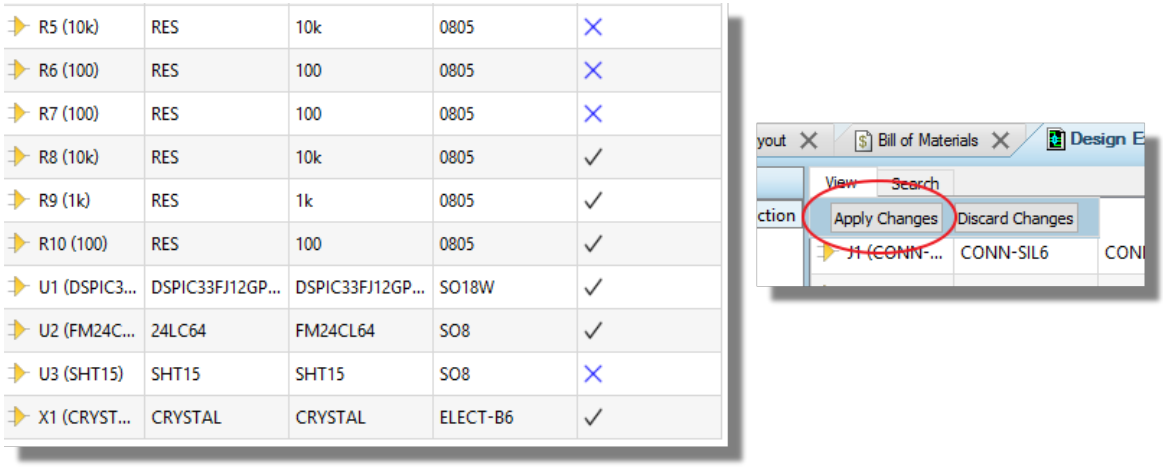
Assembly Variants provide a simple method of managing multiple product configurations in a single schematic/pcb project. This is done by specifying the fitted or not fitted status of each component on a per variant basis. They are often used for excluding parts from cheaper versions of a product or for identifying PCB versions to firmware by controlling the use of pullup resistors on an MCU port.

In Proteus, Assembly Variants are configured primarily through the Design Explorer. Let's look at a simple example and we'll remove the SHT15 temperature and pressure sensor from a new

product variant. Start by creating the new variant from inside the Design Explorer and naming it budget.

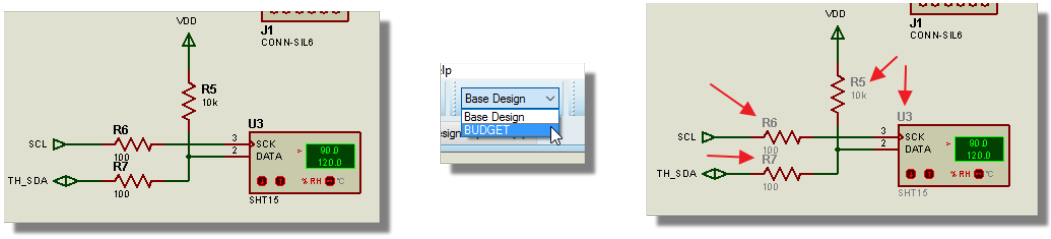


Notice that this gives us a new column in the component grid. Each entry currently has a tick which indicates that the part is fitted on the board for the variant. If we click on the entries for the budget column on U3 along with R5, R6 and R7 they will change to a cross. Clicking on the apply changes button at the top will then set these parts to be excluded from the BUDGET variant.

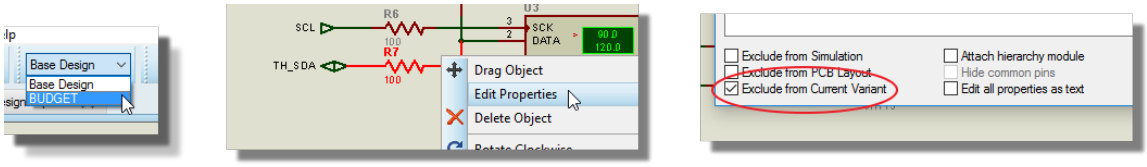


R5 (10k)	RES	10k	0805	✕
R6 (100)	RES	100	0805	✕
R7 (100)	RES	100	0805	✕
R8 (10k)	RES	10k	0805	✓
R9 (1k)	RES	1k	0805	✓
R10 (100)	RES	100	0805	✓
U1 (DSPIC3...	DSPIC33FJ12GP...	DSPIC33FJ12GP...	SO18W	✓
U2 (FM24C...	24LC64	FM24CL64	SO8	✓
U3 (SHT15)	SHT15	SHT15	SO8	✕
X1 (CRYST...	CRYSTAL	CRYSTAL	ELECT-B6	✓

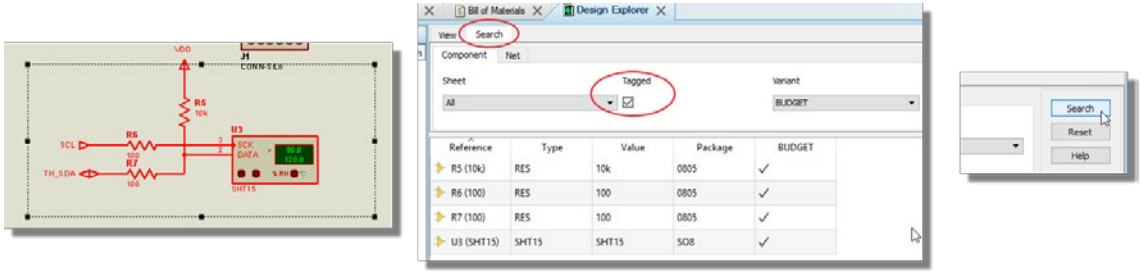
When you have more than one variant on the schematic you can switch between Active Variants via the variant combo box at the top of the screen. When we switch to budget variant you should see that the part reference for the temperature sensor circuitry will be greyed out to indicate that the part is not present in the active variant.



We could also have done this by individually editing the parts and checking the exclude from Active Variant box on the Edit component dialogue. You need to be very careful when working this way to ensure that you have first selected the variant you want from the combo-box at the top.



Finally, you can quickly manage variants for a group of parts by first dragging a selection box around the parts and then showing selected in the search tab of the design explorer.



The Proteus software manages the variant information through all of the output systems, changing the BOM Report, the Pick and Place files, the Assembly Drawings, the 3D Viewer and any export for MCAD to reflect the fitted status of the parts. Assembly variants **do not** affect the 2D PCB layout.

> You can add the @VARIANT text string to the schematic to indicate the current variant on the schematic. This is a useful reminder and particularly important for printing schematic documentation. See the header block in the schematic reference manual for more information.

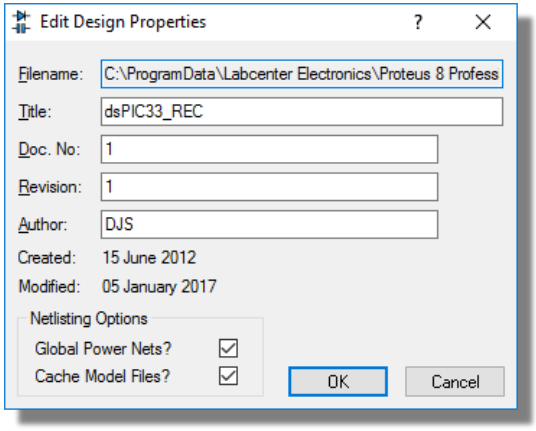
Bill of Materials

Constructing a Bill of Materials is an often necessary but frustrating task at the end of the schematic design phase. Fortunately, Proteus provides a completely flexible scheme which allows you to include as much or as little information as required.

Let's start by looking at the default Bill of Materials Output. Launch the Bill of Materials module from the application module toolbar at the top of Proteus.

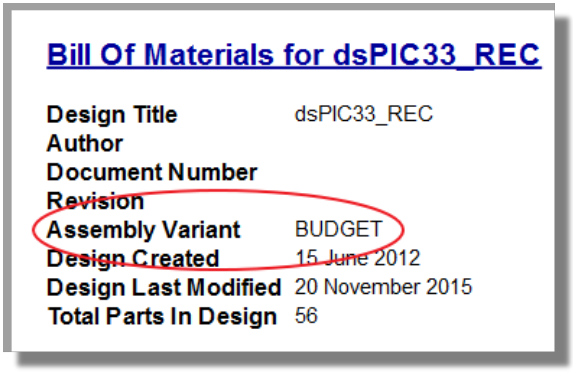
Launch the Bill of Materials from the icon

The section at the top of the report is taken from the Design Properties Command, which you can find on the Design Menu in ISIS and modify accordingly.



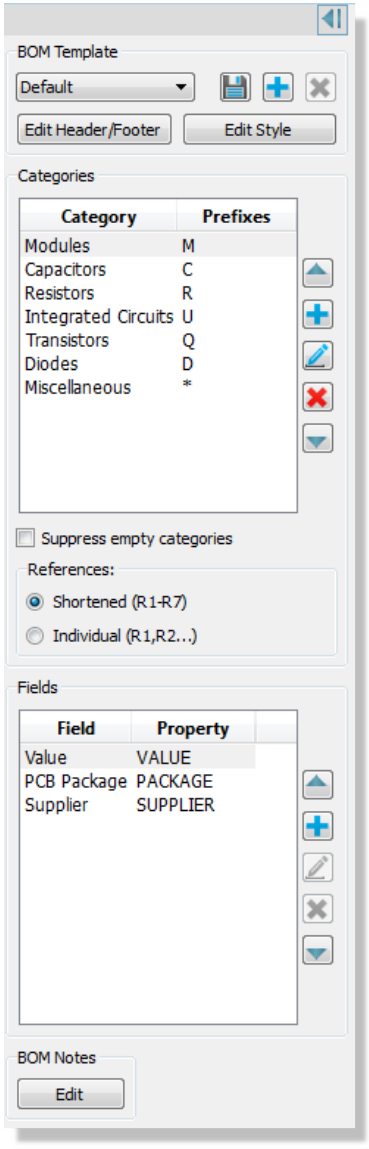
The Edit Design Properties dialogue form

If your Active Variant is not the base design then the Assembly Variant will also be included in the header block:



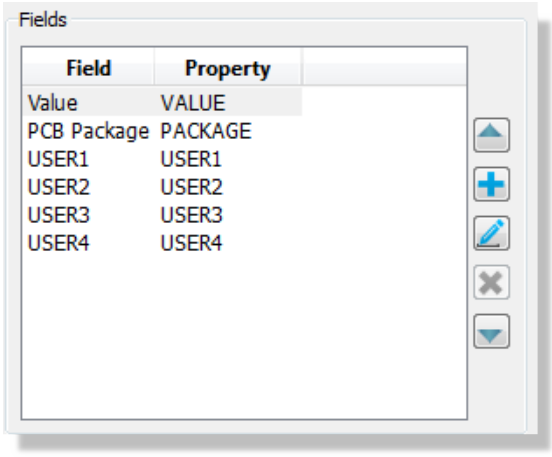
Bill of Materials categories control the grouping of parts in the report and work by comparing the reference prefix of the components. For example, all resistors have a prefix of 'R', capacitors a prefix of 'C' and IC's a prefix of 'U'. Similarly, Bill of Materials Fields control the content of the report with each field contributing a column.

Both fields and categories can be created, edited or deleted directly from the left hand control panel on the BOM report.



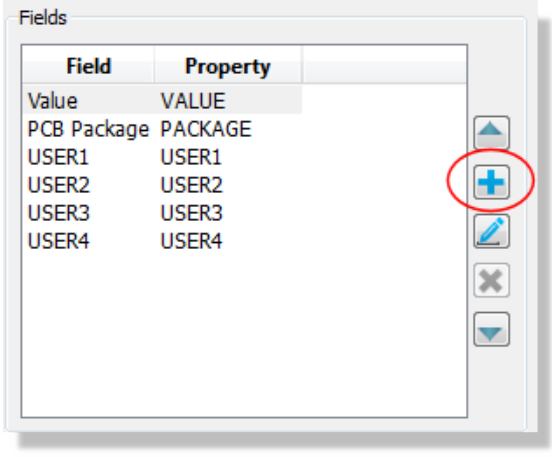
Section for editing The Bill of Materials information

Let's work through a couple of examples. First, we'll order all our parts from Digikey so let's specify a supplier in the BOM. To do this, we need a new field.



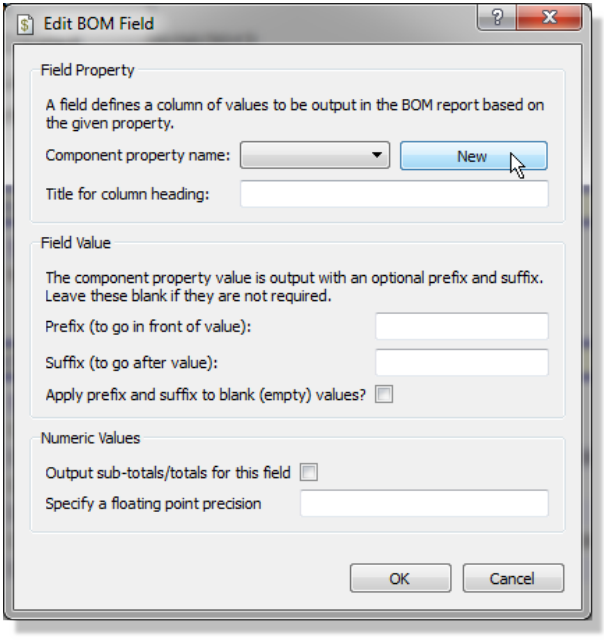
BOM Fields

Launch the BOM Field editor by clicking on the Add button underneath the Fields listing.



The 'Add' button enters the Edit BOM Fields dialogue form

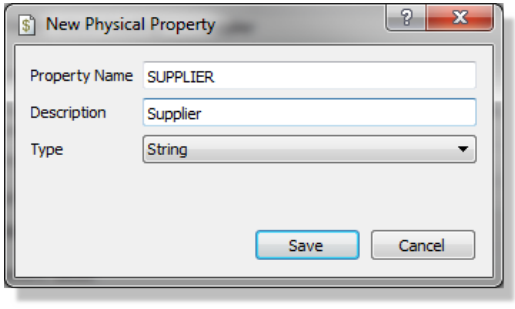
If you are adding a field that you have previously defined then you select it from the drop down combo box at the top of the dialogue form. However, since we have not defined a SUPPLIER property before click on the new button to create a property definition.



Adding a New BOM field

- There are two non-editable fields that you can apply namely .placed and .name. The placed field will show whether the part is placed on the layout (component or solder side) while the name field simply populates with the component names.

Property name is SUPPLIER, description can also be Supplier and Type is String.

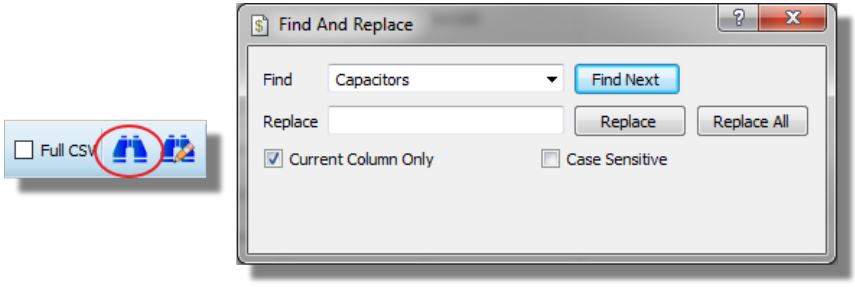


New Property dialogue filled out correctly

We don't need to prefix or suffix values and this is not a numerical property so sub-totals are meaningless. Click on the OK to exit the dialogue form and create the property.

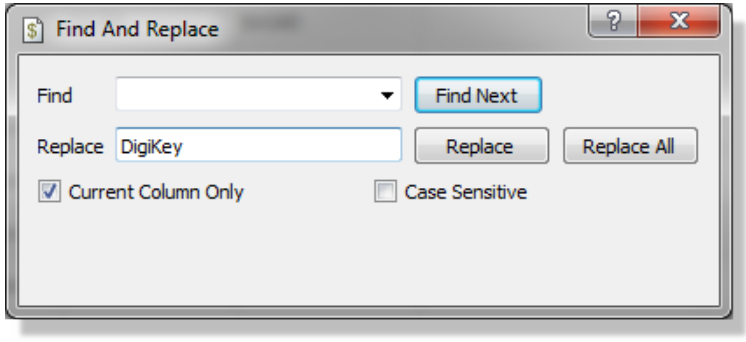
At this point the BOM has switched to the Property Editor tab which is essentially a grid view of the data in the report. We need to add information for each component in the Supplier column. We could do this individually but since all of our parts are coming from a single supplier it's far

easier to use find/replace. Left click on the supplier column and then either use the CTRL+F shortcut or click on the find/replace icon.



Find and Replace form

Leave the find edit box blank (we are entering data into blank fields), type Digikey in the Replace edit box and make sure that the current column only checkbox is selected.

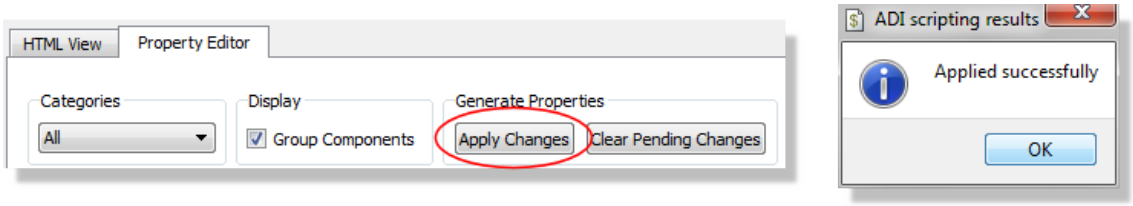


Replacing blank fields with 'DigiKey'

Hit the replace all button to enter the data. You should see that all of the fields in the Supplier column are populated and coloured in blue.

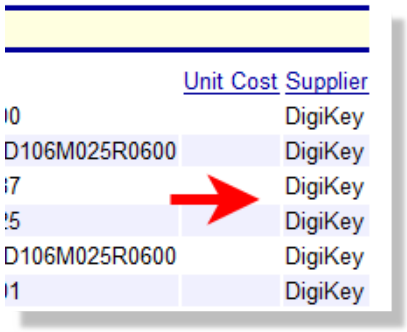
DigiKey now appears where the blank fields were

The blue indicates that the changes have not yet been committed; if you have made a mistake the clear pending changes button will revert your edits. When you have reviewed your changes use the Apply Changes button at the top to commit.



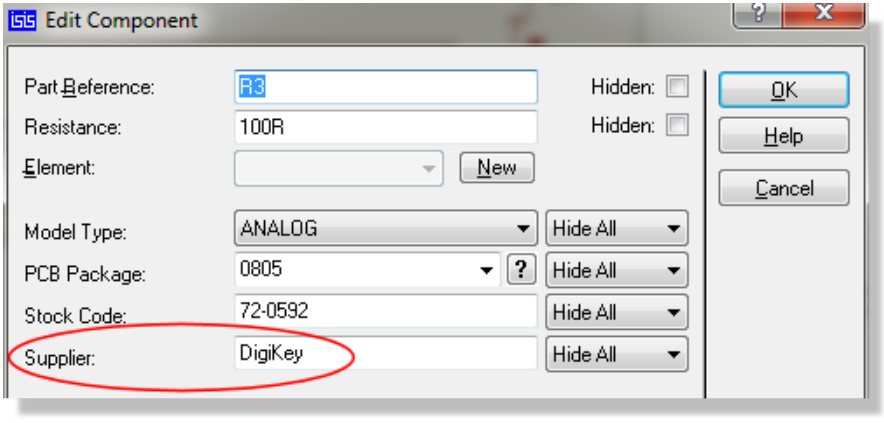
Applying BOM Changes

If you switch back to the HTML View of the report you should now see the information you entered in the report.



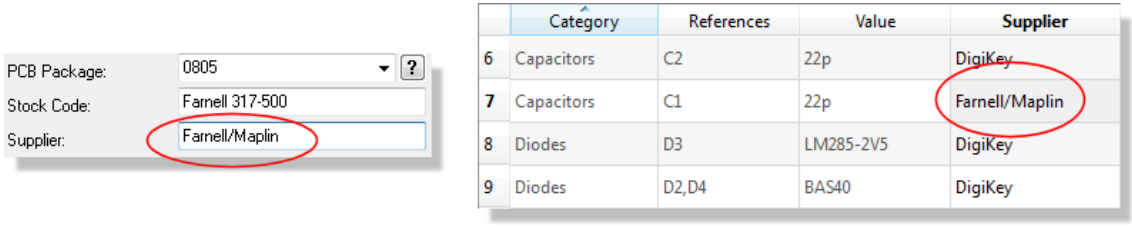
DigiKey is now assigned as the supplier

Also, if you return to the schematic and edit a component the data will be present there as well.



The Supplier field is automatically added in ISIS

Changing the data in the edit component dialogue form will live update the BOM report.

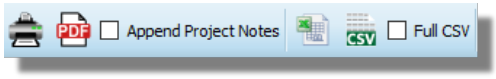


The BOM updates live with any changes in ISIS

This process can of course be repeated as required with other properties such as RoHS, MoQ, Lead Time, etc. More practical examples and detail can be found in the Bill of Materials reference manual (accessed via the Help Menu in BOM Module).

> Note that property changes are applied locally to the schematic components and not to the library parts. If you want to update your parts with newly entered information then run the 'Compile to Library' command from the Library menu. This will put all components in the design (along with their new properties!) into a library of your choice.

Having configured the information on the BOM you can generate the report from the export options from the icons at the top.

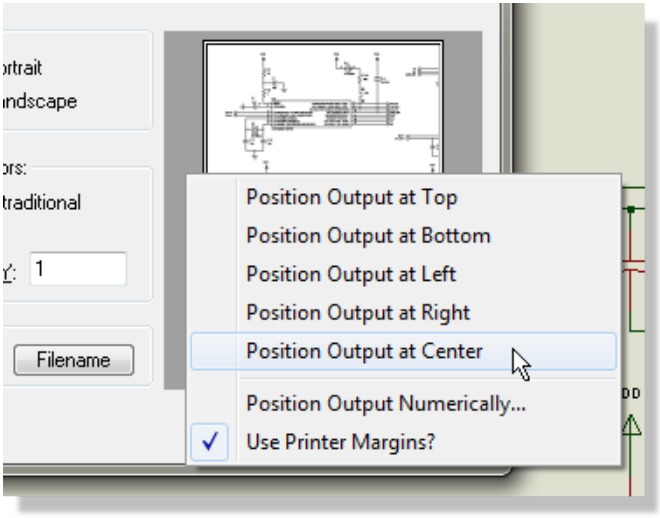


Export options for the BOM

It is beyond the scope of this tutorial but you also have full control of the colour and styling of the report. This works via the customize stylesheet button at the top of the HTML view and is discussed in detail in the BOM Reference manual.

Printing

To print the schematic, first select the correct device to print to using the Printer Setup command on the File menu. This activates the Windows common dialogue for printer device selection and configuration. The details are thus dependent on your particular version of Windows and your printer driver - consult Windows and printer driver documentation for details. When you have selected the correct printer, close the dialogue form and select the Print option on the File menu to print your design. There are a number of options available on this dialogue form, all of which have context sensitive help and are also discussed in more detail in the reference manual under Hard Copy Generation. For our purposes we will simply center the schematic output on the page and print. Do this now by right clicking on the Print Preview and selecting the Position Output at Center option as shown below.



- You can also export the schematic directly in PDF format from the Output Menu (no driver installation required).
- The current schematic is now ready for board layout so if you have purchased a Proteus PCB Design package you can move straight onto the accompanying tutorial booklet for ARES.

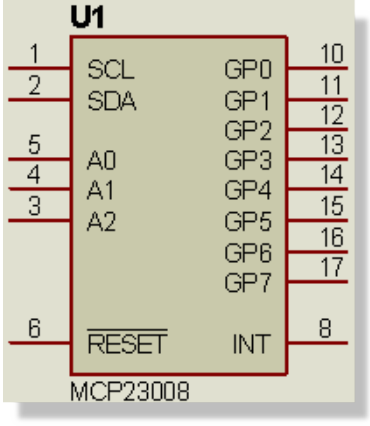
APPENDIX: Creating New Devices

If a part you need to use in your design is not present in the pre-supplied libraries you may need to create it yourself. Alternatively, you may wish to permanently adjust the properties or footprints of an existing component. This section of the documentation starts from the basics of creating a new device and covers the entire process through to storing the new part in the library.

- For most complex devices, you can often find a BSDL script from the manufacturer. This can be imported directly to make the schematic part via the command on the Library Menu.

Graphics and Pins

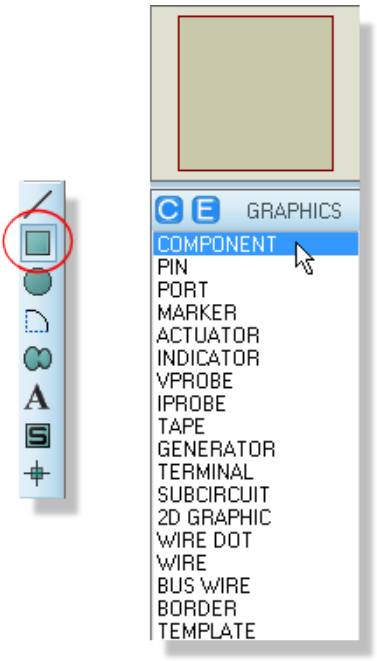
For our example part let's assume that we want to create a Microchip MCP23008 I2C expander. The first thing we need to do is to create a graphical representation of the part on the schematic. When we are finished it should look something like the following:



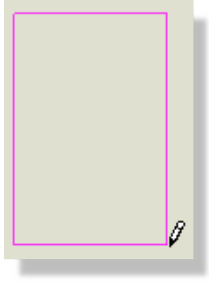
The MCP23008 Completed

Let's begin by placing the component body as follows:

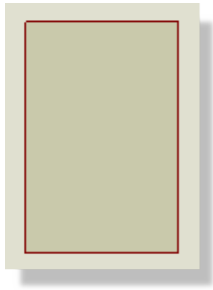
- Select the 2D Graphics Square icon and make sure the component style is selected in the Object Selector.



- Left click once on an empty area of the schematic to begin placement.



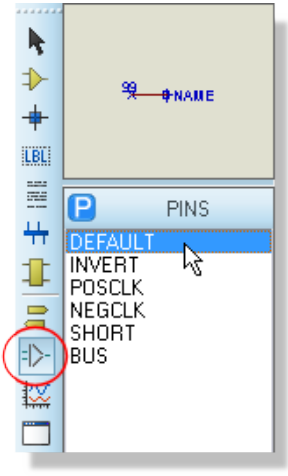
- Move the mouse until the rectangle is around the size of the part that you want. It's better to make this slightly larger than might be required as we can easily resize it later.
- Left click again to complete placement of the graphic.



- ISIS supports a powerful graphics style system of local and global styles and the ability of local styles to follow or track global styles that allows you to easily and flexibly customize the appearance of your schematic. See the section Graphics and Text Styles in the online reference manual for a complete explanation of how styles work and how they are used.

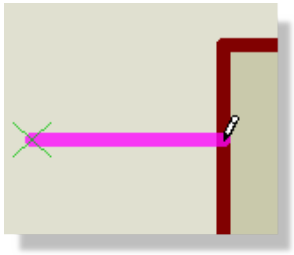
The next job is to place the component pins.

Select Component pin mode and make sure that you have the default pin type selected in the Object Selector.

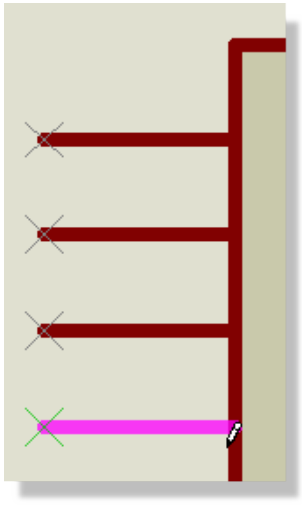


1. Left click the mouse on the schematic to begin placement. The small cross at the end of the pin indicates the side of the pin that will be wired to so it is important that this is outward facing.

2. Move the mouse so that the other end of the pin touches the edge of the component graphic and left click again to place



3. A new pin is automatically primed for placement so simply move the mouse down slightly and left click to place again.



4. Repeat the process down the left hand side of the component graphic using the completed screenshot as a reference. Right click to finish placement.

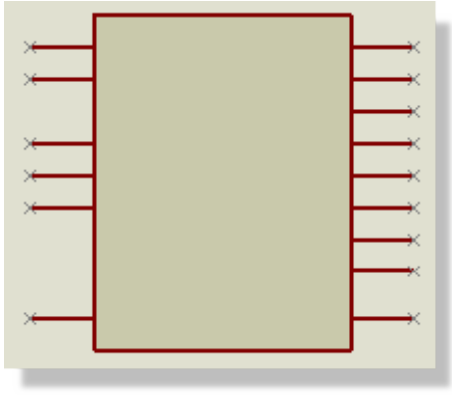
5. Click on the X-Mirror icon to point the pin in the opposite direction ready for placement down the right hand side.

6. Repeat as of stage 3 to place the pins down the right hand side of the component.

7. Right click the mouse when you are finished to exit pin placement mode.

> The Plus (+) and Minus (-) keys can also rotate the pin during placement, but you will find that when adding pin numbers, they will be on the opposite side of the pins that you placed before rotation. In this case right click on the pin(s) affected and click the Y-Mirror option to correct the pin(s).

At this stage your component graphics should look something like the following:



The body of the MCP23008

The actual positioning of the pins is of course somewhat aesthetic and you may prefer to have different groupings to those we have chosen.

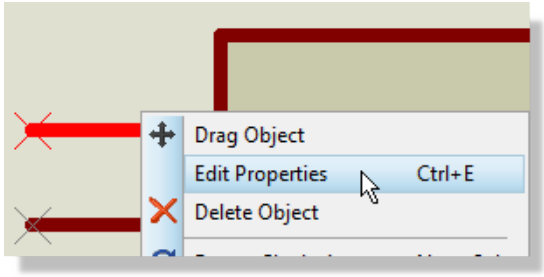
Now that we've placed the pins we may need to adjust the size of the component body. If you need to adjust the size of the component body to better fit. To do this, click on the Selection Mode icon and simply left click on the edge of the component body, then left depress the mouse on one of the nodes, drag to the new size and release the mouse to commit.

This works much like any other graphics package. For example, if you right click along the bottom you can drag up and down and if you right click on an edge and you can drag diagonally. Make sure however, that you do not leave any of the pins 'hanging' (not touching) when you do this, else you will have to subsequently move the pins back onto the component edge.

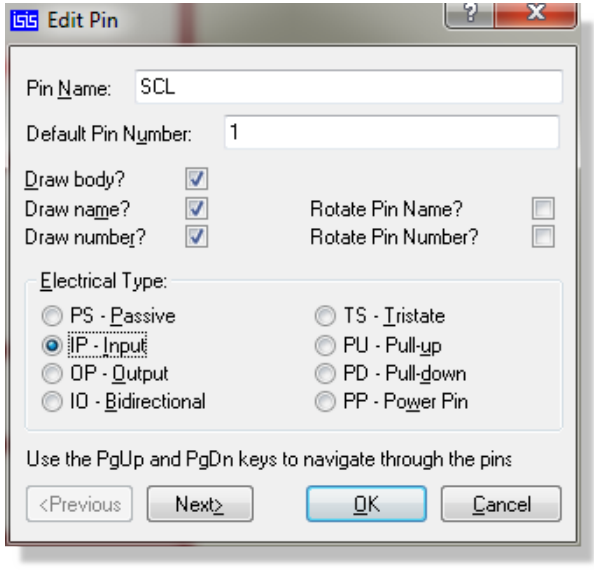
> If you want to make fine adjustments you may need to adjust the Snap level – these options are available from the View menu in ISIS (defaulted to 0.1in) and are discussed in more detail in the reference manual.

Now that we have the basic shape of the part we need to add names to the pins. This can be done as follows:

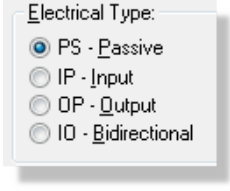
- Right click on the pin at the top left of the component and select Edit Properties from the resulting context menu.



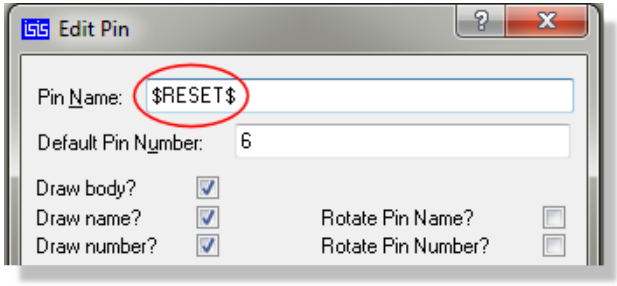
- Type in the name of the pin (SCL), the default pin number (1) and the electrical type (INPUT). You can use the other options to control visibility but for our purposes the defaults are fine.



- Hit the Next button at the bottom to switch to the next pin in the list and repeat the process – the schematic will update to show which pin you are editing. If you are unfamiliar with this part and don't therefore know the electrical type of the pin simply leave it as the default Passive type.



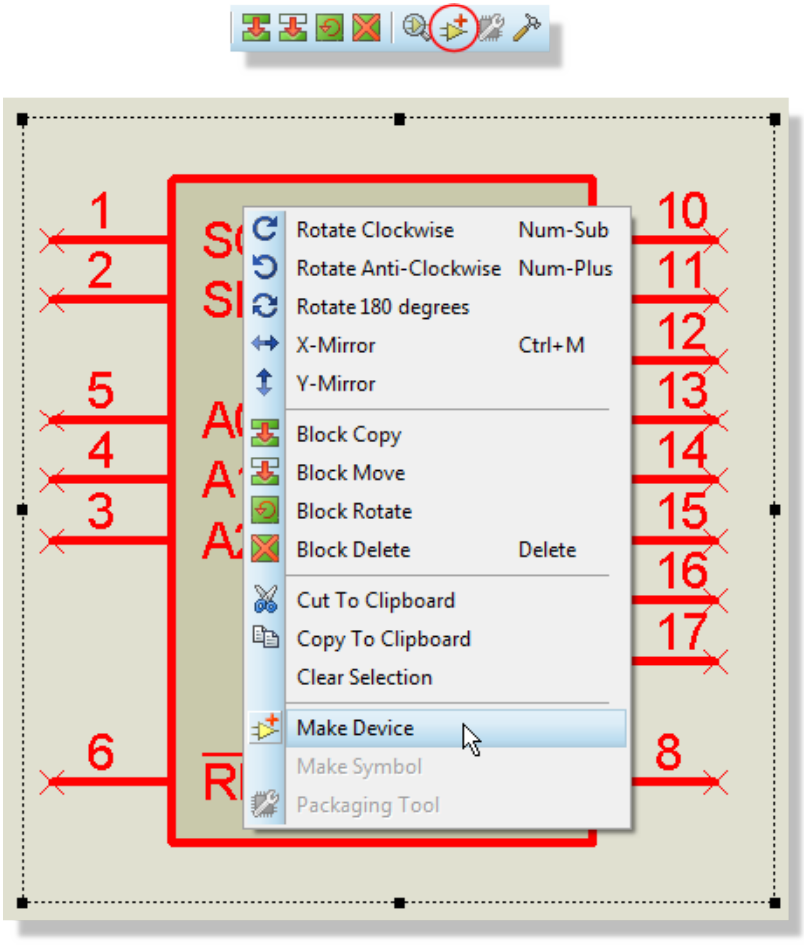
- When you get to the RESET pin you'll need to add an overbar. Do this simply by adding a dollar ('$') prefix and suffix to the name.



- Hit the OK button when you are finished.

You may have noticed that there is a significant omission from the schematic part as we have drawn it so far, namely that there are no physical power pins on the device. Since we do not need to wire power nets explicitly we do not need a physical pin for the component. We need only ensure that the power pins are handled correctly with respect to the physical PCB footprint and we'll discuss this in more detail in the next section. It is worth noting however, that while it is very common to have non-physical power pins in order to reduce clutter on the schematic, you may have a preference for everything to be explicitly wired. In this case you could simply add two extra pins and name them VDD and VSS respectively.

Our final task now is to make a device from the composite graphics and pins that we have placed. Start by drawing a tag box around the entire graphic (right depress the mouse and drag the selection box). Next, right click the mouse inside the selection box and select the Make Device option from the resulting context menu.



Make Device icon and the Context menu option

This presents us with the Make Device dialogue form, where we can customize the part. All fields throughout this wizard have context sensitive help associated with them, accessed by clicking on the question at the top right and then on the field in question. However, the only items we need to deal with here are the Device Name and the Reference Prefix. The device name should be **MCP23008** and the reference prefix should be '**U**'.

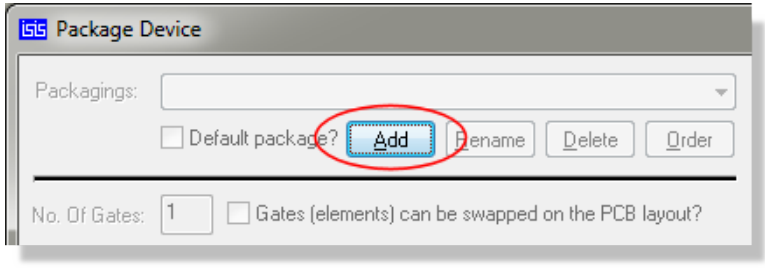
> The reference prefix serves to categorize parts for Bill of Materials Output and is also used in annotation. These are fairly standard across the industry with 'R' denoting resistors, 'C' for capacitors, 'U' for IC's and so on.

Use the Next button at the bottom of the dialogue to move onto the next screen.

Adding Footprints

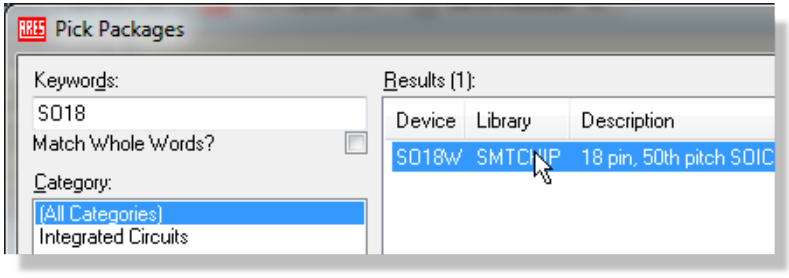
In a previous section of the documentation we saw how to quickly add a footprint to a schematic part. This is useful for small passives and simple parts but, when we are creating a device we need to follow a more complete procedure. This ensures not only that we have a correct mapping between the pins on the component and the pads on the footprint but also allows us to better handle power pins.

From the screen on the Make Device dialogue form click on the Add/Edit button to launch the Visual Packaging Tool. The first thing we need to do is to find the footprint we want to use so click on the Add button at the top of the dialogue form.



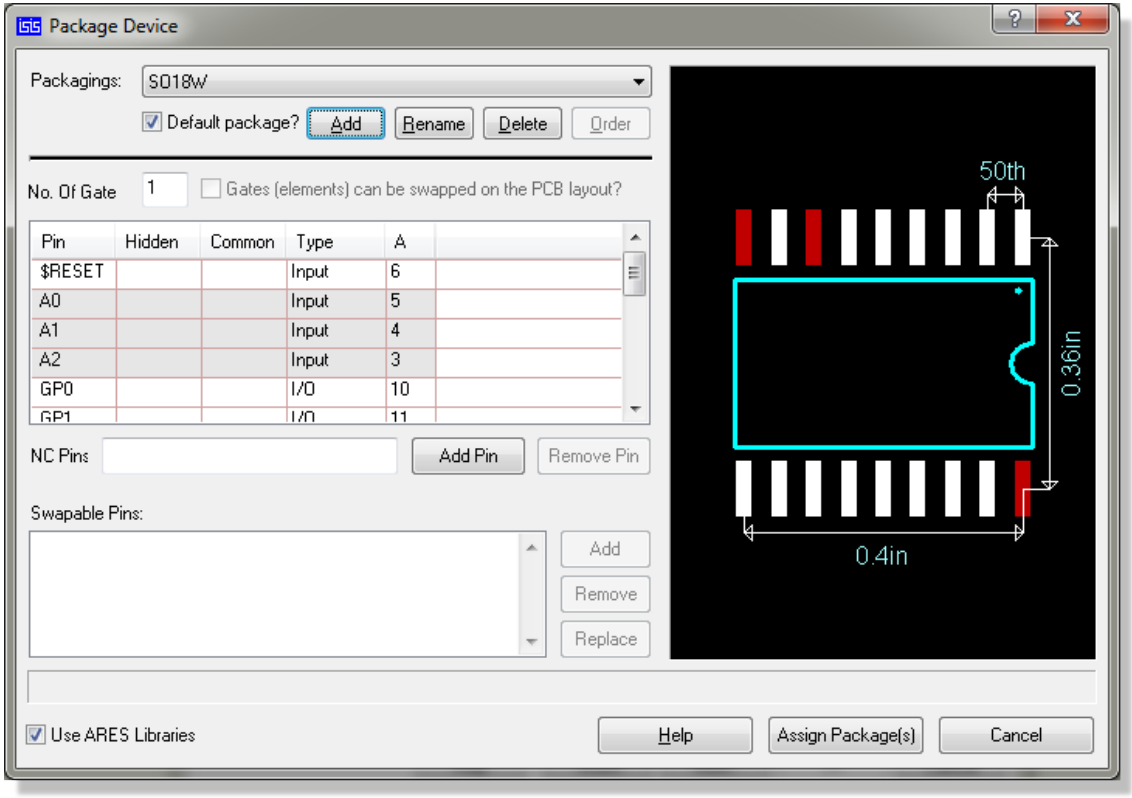
The 'Add' button in the Packaging Tool

This brings up the package browser dialogue which we covered earlier in the documentation. Assuming that we want the surface mount package, type SOP18 in the Keywords field and then double click on the part in the results list to select it.



Selecting the SO18W from the package library

The Visual Packaging Tool will now look something like the following screenshot.



There are several points of note here. Firstly, the pads in the preview which are white are deemed to be mapped onto component pins. This has happened as we have correctly specified the default pin numbers when we edited the pins earlier on (pad name maps onto pin number when present). If we had chosen to leave the pin number fields blank we would have to map all of the pins in this dialogue form.

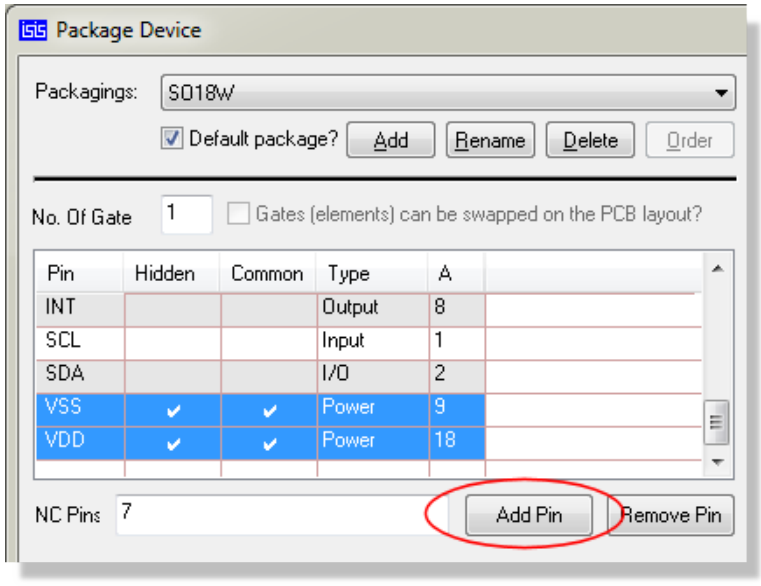
Secondly, we can see that there are three pads (7, 9 and 18) which are not mapped onto component pins. Normally we would have the datasheet beside us to get the footprint mappings but in this case pin 7 is an NC pin, pin 9 is the VSS pin and pin 18 is the VDD pin.

The NC pin is easily handled – simply type 7 in the field for NC Pins. Notice that the preview on the right hand side will update to show that this pad is now considered dealt with.

We now need to add two ‘virtual’ pins for the power pins. These are pins which don’t need to be on the schematic (their connection is made implicitly according to their name) as they will add unnecessary clutter, but do need to be mapped onto pads in the footprint.

> If you are in any doubt regarding power nets and connections please see the section earlier in the document on powering the circuit or consult the reference manual.

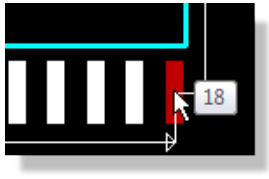
Click on the Add Pin button underneath the grid, type VSS to name the pin and then repeat the process for the VDD pin. Remember that connections to these virtual pins are made according to their name so make sure that you name them correctly.



Adding a virtual pin

Having created the pins we assign them to pads on the footprint as follows:

- Click in the column at the far right of the VSS pin.
- Click on the corresponding pad on the footprint (pad 9).
- The cursor will move down to the next pin (VDD) so all we need to do is click on pad 18 on the preview.



Clicking on pads will assign the pin

We should see that all the pads on the footprint are now highlighted, indicated that the part is fully packaged. Simply hit the 'Assign Package' button at the bottom of the Visual Packaging Tool to commit and return to the Make Device dialogue form.

- It is possible (and extremely useful) to have more than one packaging for a given component. We could, for example, follow the exact same procedure to add a

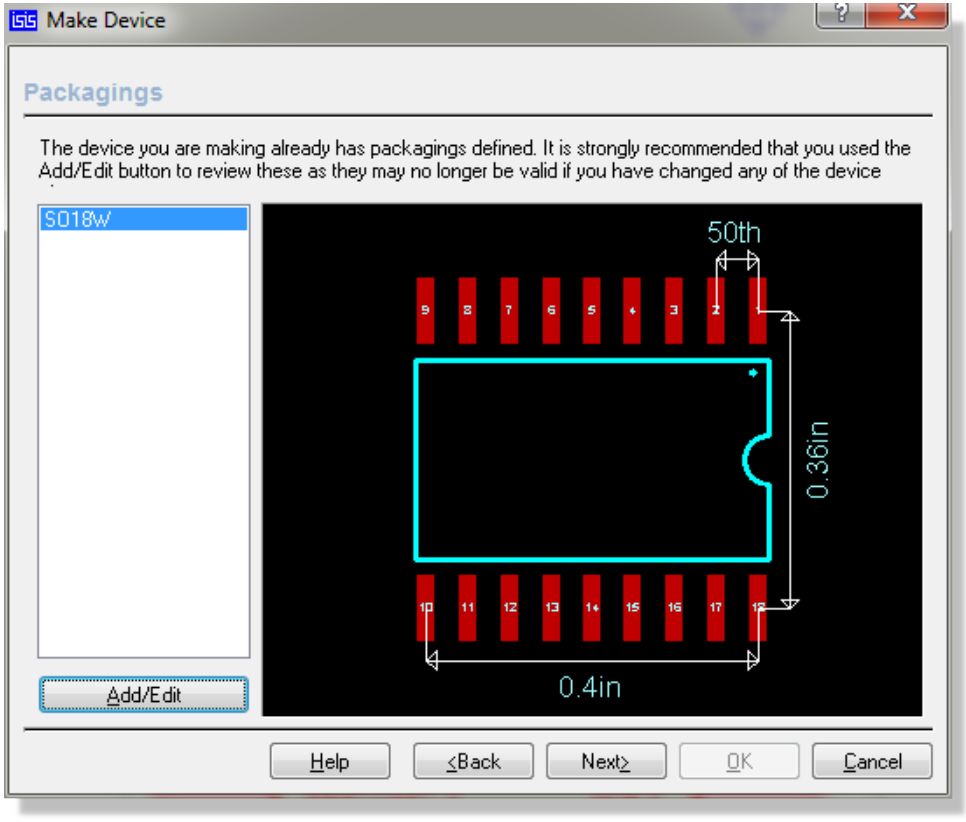
DIL18 footprint to the component. Where more than one packaging exists you can specify the one which will be used by default by checking the default box underneath the package name in the Visual Packaging Tool.

Click the next button at the bottom of the dialogue form to move onto the Property definitions section.

Adding Properties

This screen allows us to add component properties to the device. We looked at doing this on a particular instance of a device (the part on the schematic) earlier in the documentation; the difference is that properties entered here will be stored in the master library part and therefore available every time we use the part in a design.

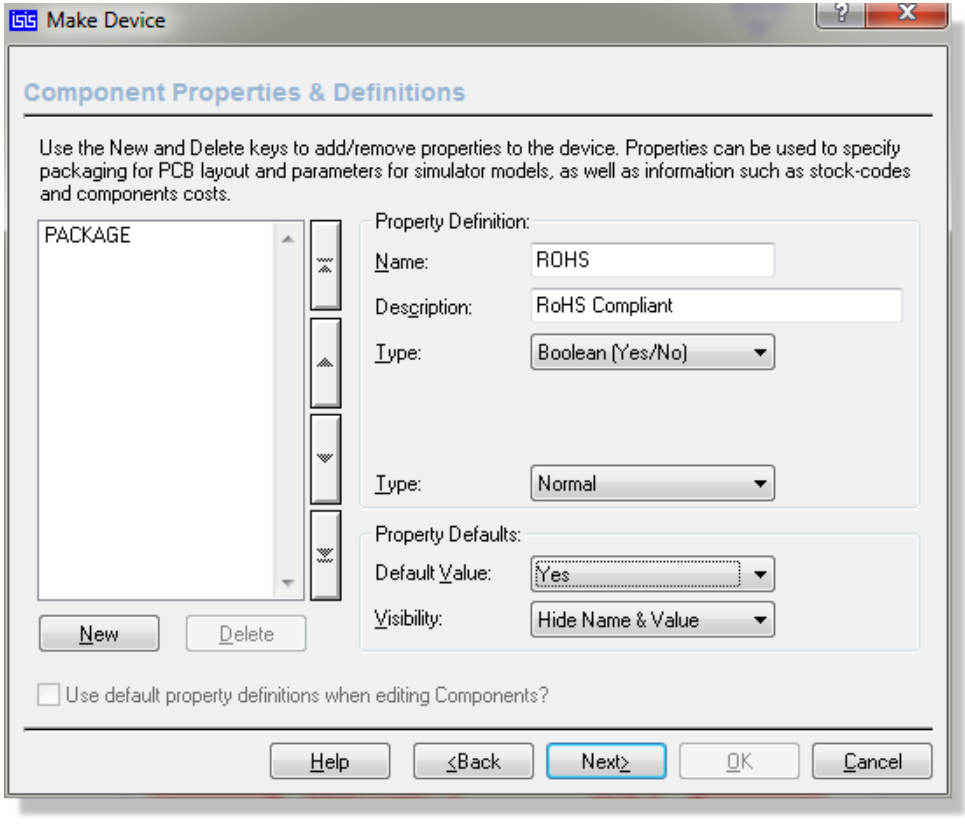
We can see that the PACKAGE property is already defined – this was what we configured visually on the previous screen



Package preview on the Make Device form

Let's assume we want to add a property for RoHS compliance.

- Start by clicking on the New button at the bottom left and then clicking on Blank Item.
- On the right hand side fill out the name as ‘ROHS’ and the description as ‘RoHS Compliant’
- The next field is the data type which in our case is a simple Boolean (Yes/No) – select this from the drop down list.
- Change the Default value to be Yes and leave the visibility defaults as they are.

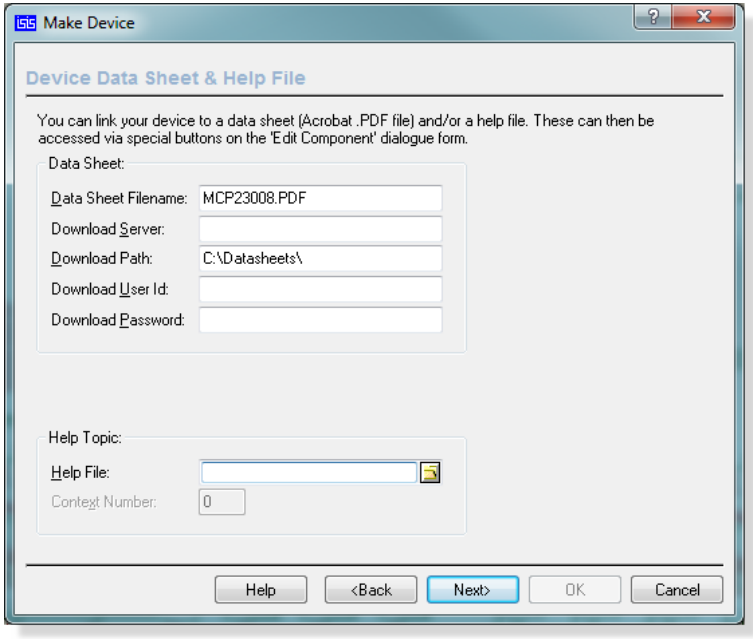


If you want to add more properties (e.g. MoQ, Price, Code, Supplier, etc.) simply repeat the process above. Alternatively, the Bill of Materials provides a convenient way to apply data to schematics as discussed earlier in the tutorial. Finally, for complete control or if you are doing this on a large scale, we recommend you read up on ASCII Data Import in the reference manual.

Click the next button at the bottom of the dialogue when you are finished to move on to the next screen.

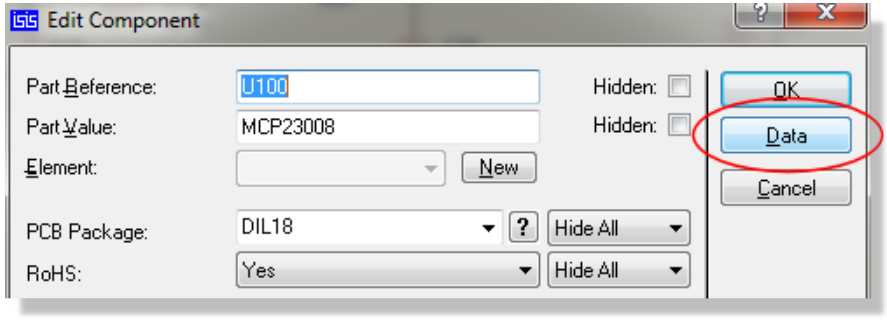
Attaching a Datasheet

This screen allows us to attach a datasheet to the part we are creating. To do this we would need to enter the filename for the datasheet and then configure the path as required. As we don't have a datasheet handy we'll ignore this step but the following screenshot shows an example of how we would set it up.



Adding a datasheet to the component

The main advantage of working through this step is to have the datasheet handy when you place the part. If present, you can simply edit the component and click on the Data button at the right hand side.

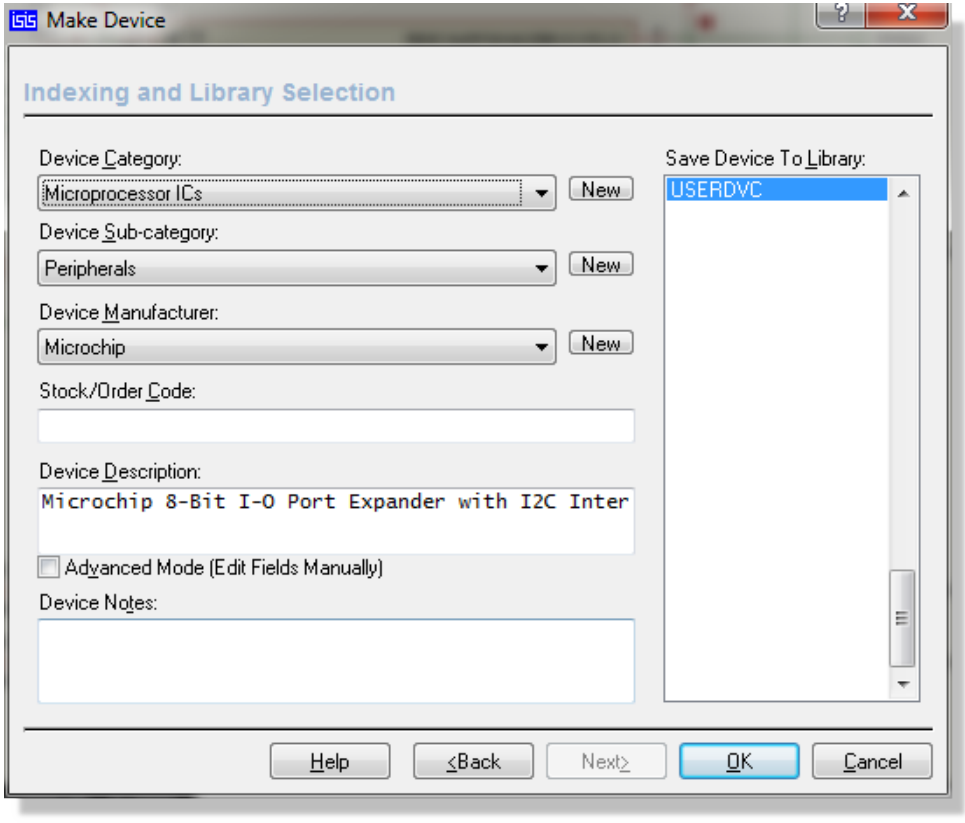


The Data button will launch the assigned Datasheet

For convenience, we have configured many of our library parts with datasheets which you can download from our web server via this button.

Indexing and Library Selection

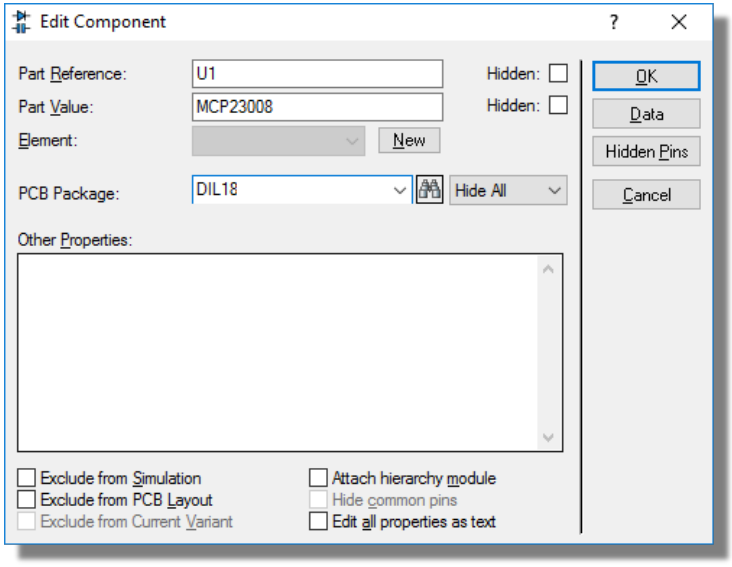
The final screen allows us to categorize our part and choose which library we will place it in. As we have seen earlier in the tutorial the part description and the categories are used as filters when finding and selecting parts from the library so it is worth giving some thought to your description. An example of what you might enter is shown in the following screenshot.



Final section where you save the component to a library

When you are finished simply click the OK button. You will be prompted to replace all instances of the device on the schematic – this allows you to update the current design if you are modifying a part which already exists on the schematic.

Finally, if you now select the part from the Object Selector and edit its properties you should see something like the following screenshot.



Edit Component for the MCP23008

It is also possible to create multi-element parts in ISIS. While beyond the scope of this tutorial these are covered in the reference manual.

PCB LAYOUT TUTORIAL

Introduction

The purpose of this tutorial is to familiarize you as quickly as possible with the main features of ARES to the point that you can use the package for real work. Users with modest computer literacy should find it possible to learn the package and produce their first board within a day or two.

The tutorial proceeds by taking you through worked examples involving all the important aspects of the package including:

- Basic techniques for placement and routing.
- 3D Board Visualisation.
- Netlist based design including both manual and automatic routing.

More advanced editing techniques such as block editing and route editing.

- Report generation.
- Hard copy generation.
- Library part creation.

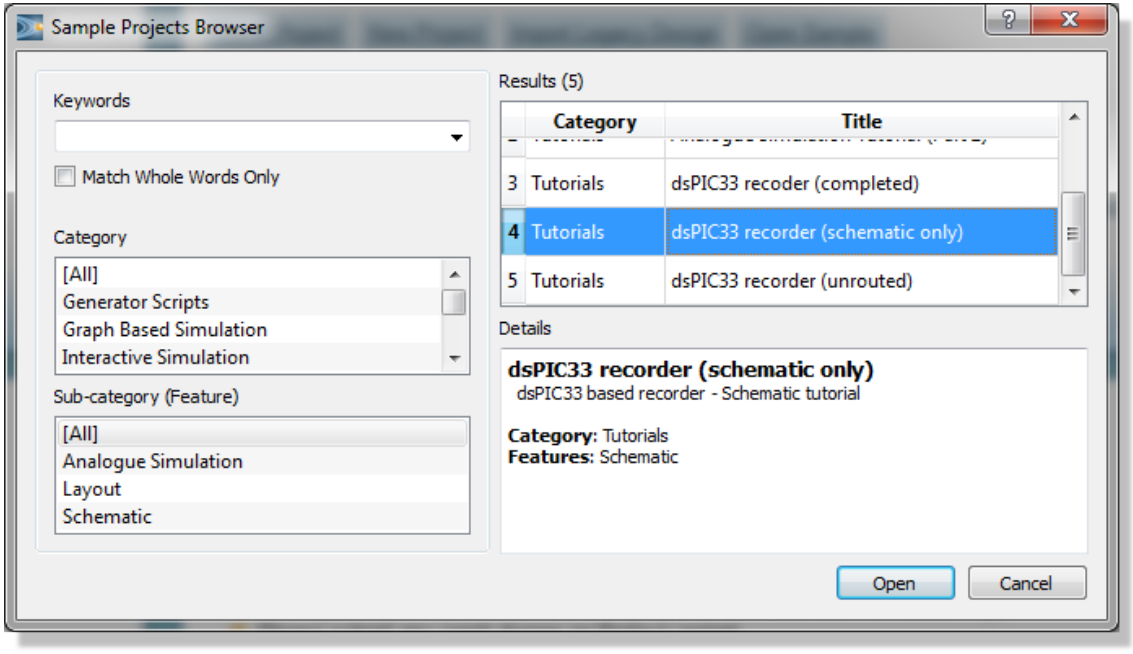
We do urge you to work right the way through the tutorial exercises as many things are pointed out that if missed will result in much wasted time in the long run. Also, having worked through the tutorial and thus got a basic grasp of the concepts behind the package you will find it much easier to absorb the material presented in the reference chapters.

- Note that throughout this tutorial (and the documentation as a whole) reference is made to keyboard shortcuts as a method of executing specific commands. The shortcuts specified are the default or system keyboard accelerators as provided when the software is shipped to you. Please be aware that if you have configured your own keyboard accelerators the shortcuts mentioned may not be valid. You can configure your own keyboard shortcuts via the System - Set Keyboard Mapping command.
- Menus toolbars and icons all switch when you change between tabs to reflect the functionality of the module you are working on. When we talk about menu commands or icons in this tutorial we assume that the PCB layout tab is selected. The menu contents will be very different if the schematic tab is selected!

Overview of the Layout Editor

We shall assume at this point that you have installed the package, and that the current directory is some convenient work area on your hard disk. This tutorial is a direct continuation of project we started in the schematic capture tutorial so we will start by loading the project file with the completed schematic.

From the home page in Proteus click on the Open Sample button, filter to the tutorials category and then select the dsPIC33 Recorder (schematic only) sample.



Opening a Tutorial sample design

This will open the tutorial project with the completed schematic from the Schematic Capture tutorial. We can start the PCB layout module (ARES) from the application module toolbar at the top of the Proteus application.

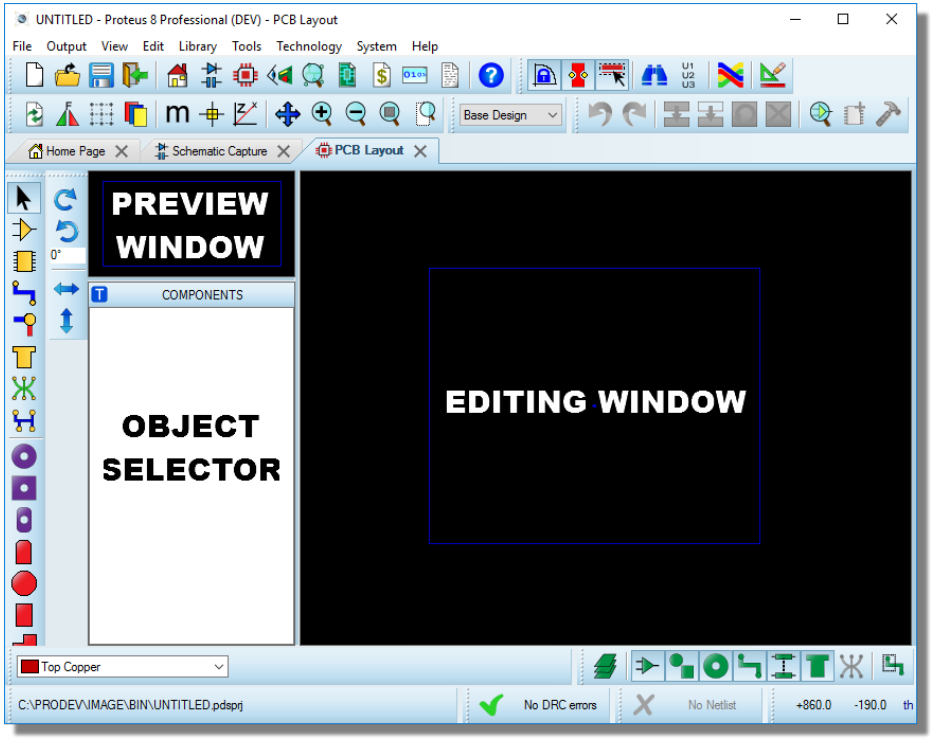
If you are working on two monitors or have free screen space you can drag and drop one of the tabs to view both modules simultaneously. If not, you can switch between the tabs with the mouse or via the standard windows CTRL+TAB shortcut key.

Much of the look and feel of the application is similar to ISIS and hopefully now familiar, although there are some important differences.

The Main Window

The largest area of the screen is called the Editing Window, and it acts as a window on the drawing - this is where you will place and track the board. The smaller area at the top left of the screen is called the Overview Window. In normal use the Overview Window displays, as its name suggests, an overview of the entire drawing - the blue box shows the edge of the current sheet and the green box the area of the sheet currently displayed in the Editing Window.

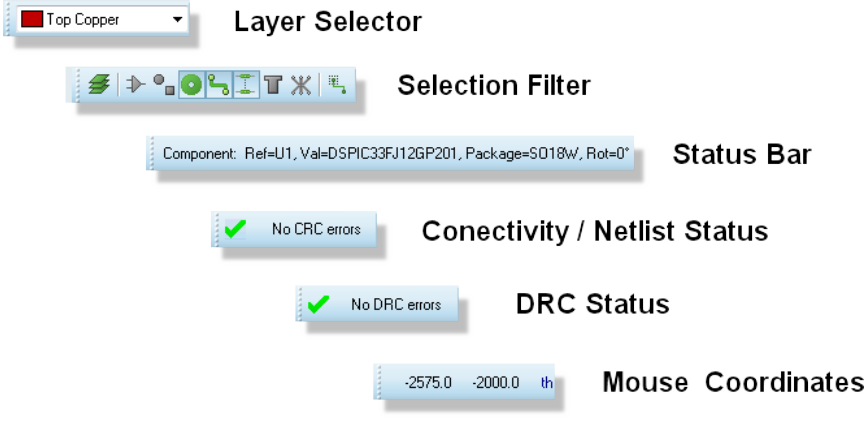
However, when a new object is selected from the Object Selector the Overview Window is used to preview the selected object – this is discussed later.



ARES PCB Layout Window

The Control Bar

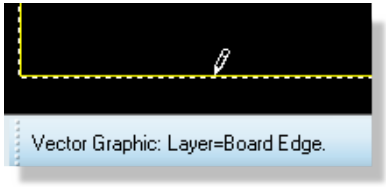
The control bar at the bottom of the application is different from what we have seen in the ISIS application and essentially splits into five sections:



At the left hand side is the Selection Filter which allows you to configure both the layers and the objects that will be selected in the current operating mode. Typically, the default rules will suffice and this serves simply as a convenient override where you may wish finer granularity in selection at a given time. The Layer Selector combo box also determines the current layer or layer set and also applies to the placement of PCB objects.

The Layer and Selection filter

In the middle is the Status Bar which provides textual 'hints' on the object currently under the mouse. This is particularly useful when you hover a mouse over a pad for example, as it will inform you which net the pad is on.



The Status bar shows what the mouse is pointing at

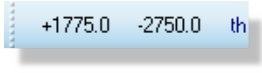
Next, we have the connectivity status which controls the syncing of the PCB to the schematic and - when everything is synchronized - doubles as a live connectivity checker.

Netlist Information and CRC information on the status bar

Next to the connectivity status is the live Design Rule Checker. This will report any physical design rule violations that occur while the board is being designed. A left click on this will launch a dialogue detailing the violations with the further option of zooming in to examine a particular error.

DRC information on the status bar

Finally, at the far right hand side is the co-ordinate display which reads out the position of the cursor when appropriate. These reflect not the exact position of the pointer but the location to which it has been snapped. Default snapping options are selectable from the View menu (or via keys CTRL-F1 and F2 -F4) and the snap values can be configured from the Grids command on the Technology menu



Mouse Co-ordinates

The co-ordinates can be in imperial or metric units as set by the Metric (default key mapping 'M') command. You can also set a false origin using the Origin command (default key mapping 'O') in which case the co-ordinates change colour from black to magenta.

The dot grid on the Editing Window can be toggled on and off using the Grid command, or via its keyboard shortcut (by default this is ‘G’). The spacing of the dots normally reflects the current snap setting, except when zoomed out. In this case, the dot spacing is set to a suitable multiple of the snap spacing.

ARES can be set to display an X cursor at the position to which it has snapped the pointer through the X-Cursor command, default key mapping is 'X'.

We’ll become familiar with all of these items and use them regularly as we work through placement and routing of the board.

Navigation

Navigation of the layout (middle mouse zoom, panning, keyboard shortcuts, etc.), is identical to the ISIS schematic capture package. Refer to the ISIS Getting Started Guide or the ISIS reference manual for more information.

Visual Aids to Design

As in ISIS the ARES package will use visual effects to help you understand what is happening during board layout. There are two principle techniques:
ARES will ‘twitch’ an object when that object is under the mouse and the selection filter enables selection of that object type. This serves to identify when an object is ‘hot’.
ARES provides dynamic cursors which change to identify what a left click will do at any given time (place an object, select an object, move an object, etc.). A list of cursor types is shown below.

Cursor	Description
	Standard Cursor
	Placement Cursor - Left click will place and object according to mode.
	Selection Cursor - left click will select the object under the mouse.
	Movement Cursor - left drag will move the object selected.

Throughout this tutorial – and indeed whenever you are working in ARES – you should make use of these visual indicators to as an aid to understanding the software.

- Proteus 8 includes a multi-level ‘undo’ and ‘redo’ system via the shortcut keys CTRL+Z and CTRL+Y respectively. If you make mistakes throughout the tutorial remember that you can unwind as many times as you need to!

Display Options

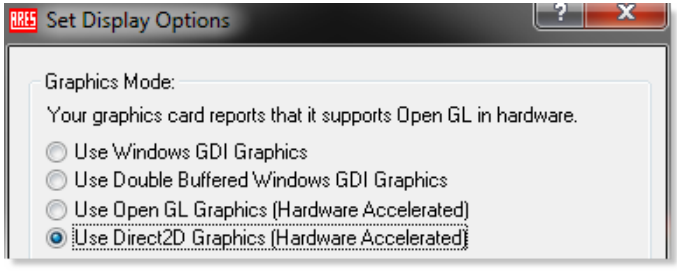
ARES is capable of harnessing the power of your computers graphics card to speed up operation and provide true layer transparency on the layout. However, as not all machines have sufficient graphics cards the software is also capable of using Windows to perform display and graphical operations. The three modes of operation are called:

- Windows GDI Mode.
- OpenGL Hardware Accelerated Mode.
- Direct2D Hardware Accelerated Mode

Proteus will test your computer when you first open the application and, if possible, will then set a default hardware accelerated mode for you. Control over hardware acceleration and display options can be found via the Set Display Options command on the System Menu. Some options are relevant only to a particular mode and will be disabled if you select a different mode.

Graphics Mode

The first section of the dialogue reports on whether your graphics card will support OpenGL or Direct2D hardware acceleration and, if so, allows you to switch from Windows GDI mode into one of these two modes.



Select the best option that your graphics card supports.

Opacity

When a hardware accelerated display mode is selected the Opacity section of the dialogue allows you to configure the transparency of various layers on the board.

Opacity options

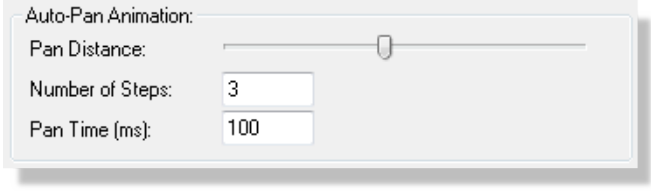
When using hardware acceleration, ARES places special significance on the current layer, giving it a higher opacity than other copper layers on the layout. This means that objects on the layer you are working are clearly visible and that objects on the other layers of the board are dimmed. You can control the relative opacity of both the current layer and the background layer(s) using the sliders on this section of the dialogue form. For example, if you wish to disable transparency altogether you could set the background layer slider to maximum.

Additionally, with hardware acceleration you can live view the solder resist and paste mask around pads and vias on the layout (turn on/off from View Menu - Layers). When enabled, you can also adjust the opacity of these layers using the appropriate slider controls

- If you are working on complex, multi-layer PCB's it can be useful to define a number of preset views for the board. This is done by creating new colour sets in the displayed layers dialogue (e.g. TOP_VIEW, BOT_VIEW) with appropriate layers visible and assigning them keyboard shortcuts.

Auto-Pan Animation

In ARES, holding down the shift key and bumping the mouse against the edge of the Editing Window allows you to pan the screen. Similarly, dragging an object against the edge of the Editing Window will pan the screen in the direction of the drag. This navigation feature is called auto-pan and is controlled via the options in this section of the dialogue form.

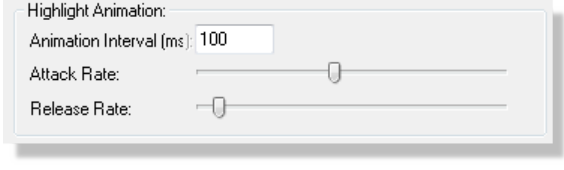


Auto-Pan options.

You can adjust the distance moved on auto-pan (pan distance), the number of steps taken to cover this distance (number of steps) and the speed of the animation (pan time). These options are available regardless of which mode of operation you have specified.

Highlight Animation

When working in hardware accelerated mode objects under the mouse with smoothly increase their intensity to inform you that they are selectable. The highlight animation options in this dialogue allows you to control the smoothness and speed of this effect.



Animation Speed options

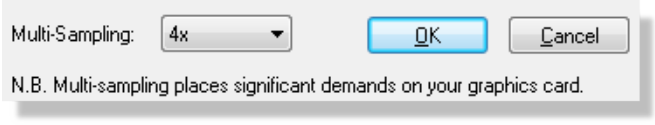
The animation interval controls the frame rate of the animation and therefore the smoothness of the fade-in/fade-out effect. Normally, the default value for this is suitable.

The attack rate specifies the speed at which an object will go from ‘fully off’ to ‘fully on’ whilst the release rate allows you to adjust the speed at which an object will return from ‘fully on’ to ‘fully off’.

These options are only available in hardware accelerated modes.

Multi-Sampling (Anti-Aliasing)

Multi-Sampling is a method used by graphics cards to reduce anti-aliasing effects when graphics are displayed at different zoom levels. It is particularly applicable to text but impacts on all graphics on the layout.



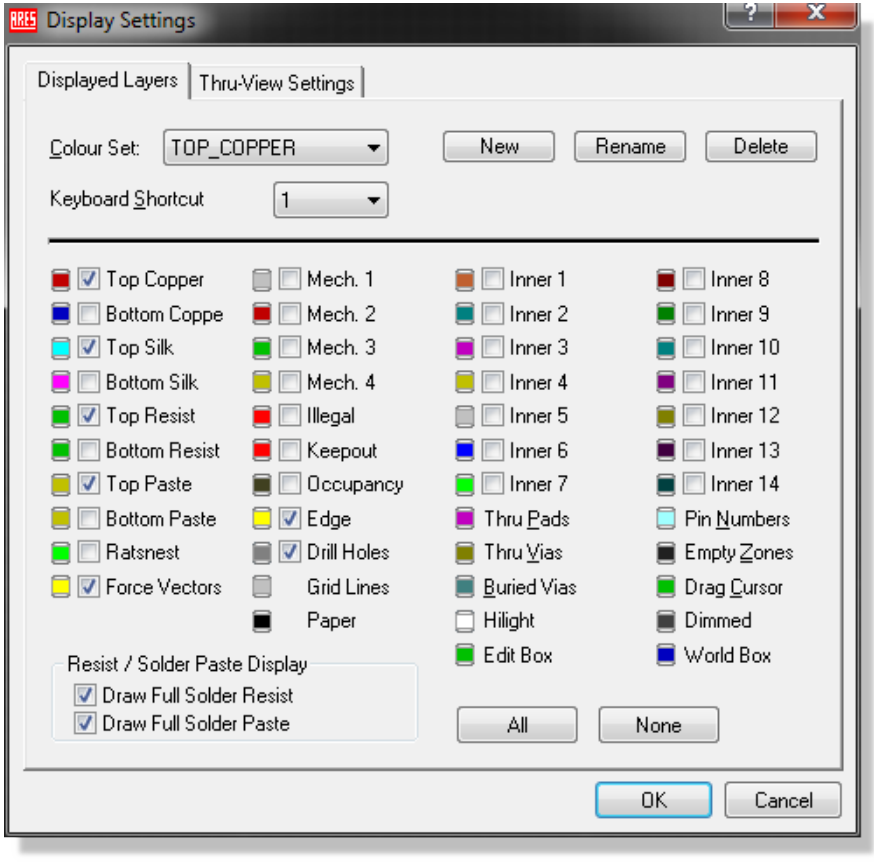
When working in OpenGL mode you can specify the level of multi-sampling you want to use. The higher the level of multi-sampling the better the resulting display but the more GPU resources are consumed. If you select a level of multi-sampling which is not supported by your graphics hardware the software will reset the level to the closest one which your card can handle. For normal operation in ARES a multi-sampling level of 4x is quite sufficient.

Displayed Layers Dialogue

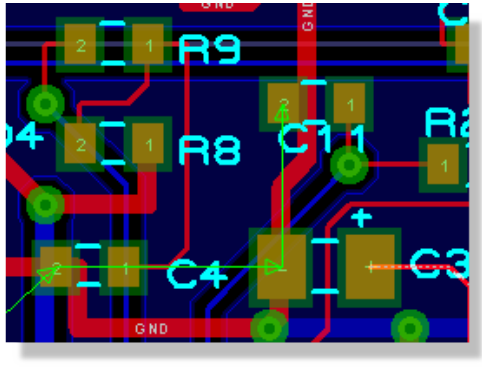
The displayed layers dialogue allows you to control the visibility and colour of the various layers on the board. You can launch this from the View menu in ARES or by clicking the mouse on the status bar at the bottom of the application.

You can change the colour of a layer by clicking on the colour beside the layer and control the visibility of the layer via the checkbox beside the layer (where appropriate). All changes are updated live on the layout.

You can switch between the pre-supplied colour profiles (print and screen), create your own colour profiles via the 'new' button at the top of the dialogue form and assign them to keyboard shortcuts. This allows you for example to have a TOP colourset showing only top copper, top silk, top resist and top paste mask. More information is available via the context sensitive help for the dialogue form (click on the question mark at the top right of a dialogue and then on a field of the dialogue form to launch).



Finally, if you are working in OpenGL or Direct2D mode, the Resist/Paste Mask Display options allow you turn on full display of these layers on the board, showing the resist and paste coverage around pads and vias. When enabled you can change the intensity of these layers by switching to the Thru-View settings tab and adjusting the appropriate slider controls.



Component Placement

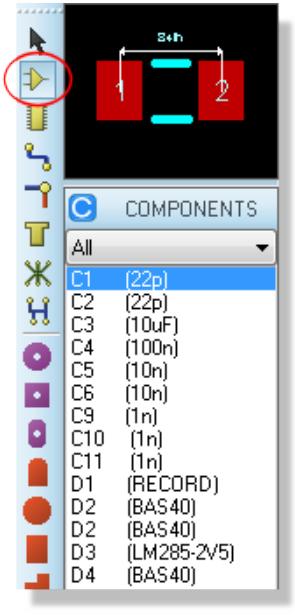
Proteus 8 works with a live netlist so the ARES module already has much of the information we need to start the layout process. In particular, we have specified which footprints are associated with each schematic symbol and ARES can therefore pre-select these for us ready for placement. This brings us to an important distinction in the software; the difference between a component and a package.

Components and Packages

- A component is an instance of a footprint that has been netlisted through from the schematic.
- A package is a physical footprint that exists in the ARES Libraries.

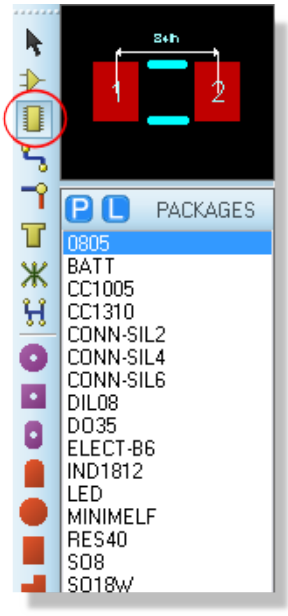
Selecting component mode will access footprints which have been specified as relating to parts in ISIS and carry connectivity information whereas selecting package mode will access 'unbound' physical instances of a footprint. When working with a layout which is driven from a schematic we therefore exclusively use component mode.

The Component Mode icon is second from the top directly underneath the Selection icon. Clicking on this will display a list of items in the Object Selector which correspond directly to the parts in ISIS that we used to create the schematic.



Component Mode

The Package Mode icon is directly underneath the Component mode icon and clicking on this will show us the physical footprints corresponding to the components in the layout.



Package Mode

When placing, routing and laying out a PCB based on a schematic (such as with this project) we will be working with components.

The Board Edge

Before we can place the components on the board we need to define what shape and size the board will be. For our project we need only a simple rectangular board edge but we do want to control the dimensions of the board (115mm wide and 40mm high).

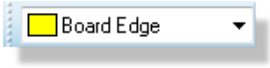
The first thing to point out here is that ARES will operate in either imperial or metric units and you can switch between modes either by toggling the Metric command on the View menu or via the 'M' keyboard shortcut. You may need to switch units for the placement of the board edge and also elsewhere in the documentation.

> You can set your default or preferred units from Technology Menu - Grid Configuration command.

To start placing the board edge, select the 2D Rectangle icon from the left hand side of the application.

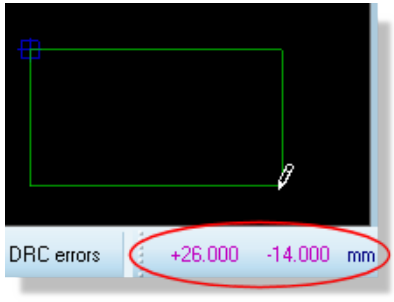
2D Box Icon

Next, change the Layer Selector to the Board Edge Layer.



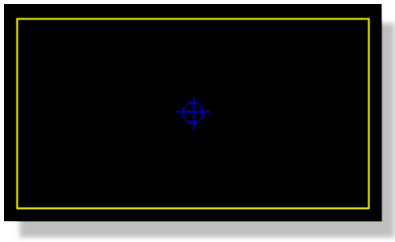
Board Edge layer selected

Move the mouse to the approximate starting point (e.g. top left) for the board edge. Now, hold the mouse still and press the 'O' key on the keyboard to set the origin of the co-ordinate system to the point under the mouse. This will be reflected in the co-ordinate display at the bottom right of the application. Left click to start placement and drag in the normal way. The co-ordinate display at the bottom will show you the dimensions of the board edge as you drag.



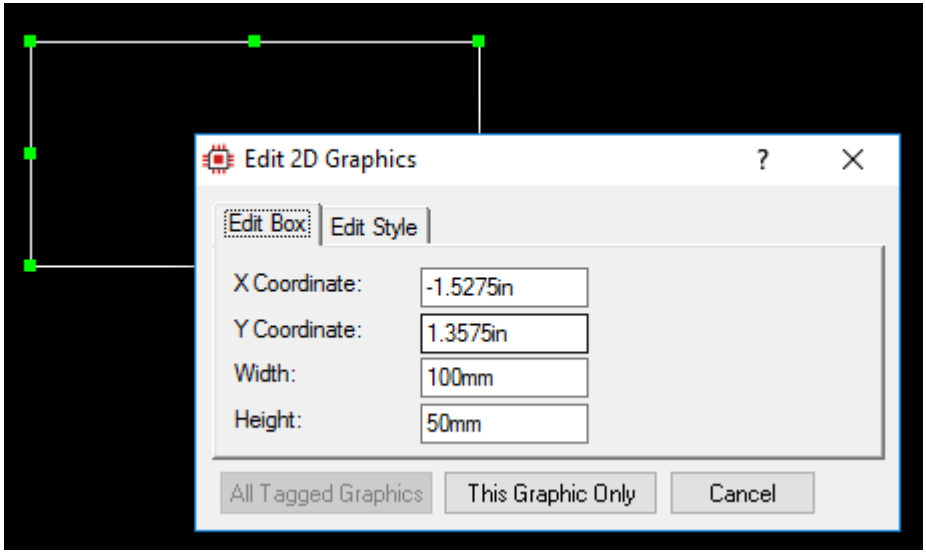
Drawing a board edge

Press the 'M' key on the keyboard to toggle between metric and imperial units as required. Once you have the desired size of board edge click left again to commit placement. Don't worry about where in the Editing Window you have placed the board edge – we will move it to the centre of the world area shortly.

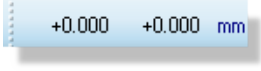


Placed board edge which is always Yellow

If you would like the board edge to be a specific size, edit the board edge graphic and set the dimensions to your requirements:



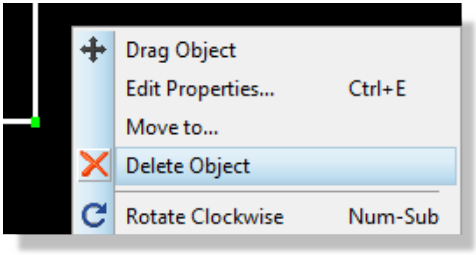
Finally, restore the co-ordinate system to its global origin by pressing the ‘O’ button on the keyboard again. The colour of the co-ordinates will change from Magenta to black to indicate that the global origin is now in use.



Co-ordinates at 0,0

- Placing a board edge should always be the first task as the software needs to know the boundaries of the board in order for example, to know the limits of where the autorouter can function or the size of a power plane.

If you need to delete the board edge or resize it you can do so via the context menu options and drag handles that will appear when you right click over the board edge graphic.

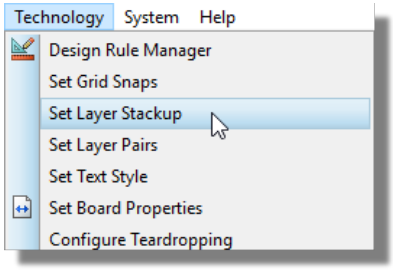


Select Delete from the context menu

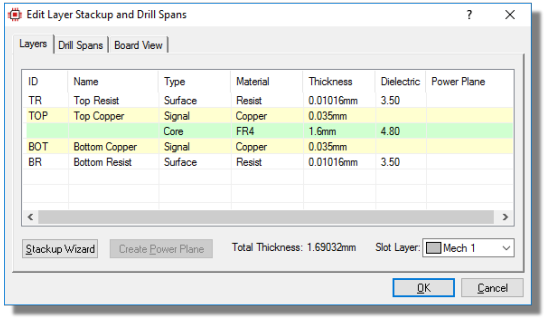
Complex board edges can be directly imported onto the board edge layer using the Import DXF function on the File Menu.

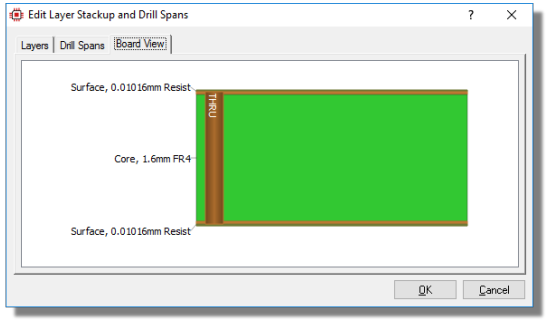
Layer Stackup and Drill Spans

Before we begin layout, we need to define our layer stackup and specify which drill ranges are valid for vias. Start by launching the layer stackup dialogue from the Technology menu.

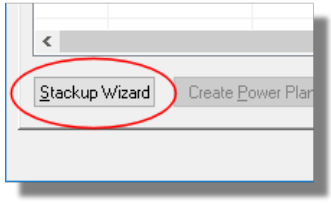


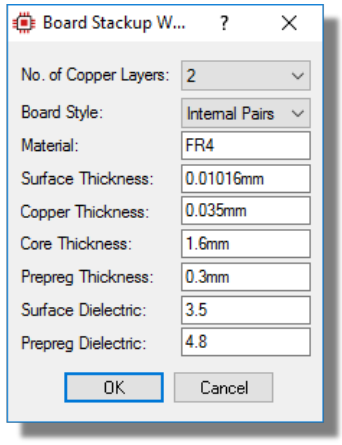
We will be making a simple two-layer board with a single core between top copper and bottom copper. This will be shown visually on the editable grid and as a cross section in the board view tab.



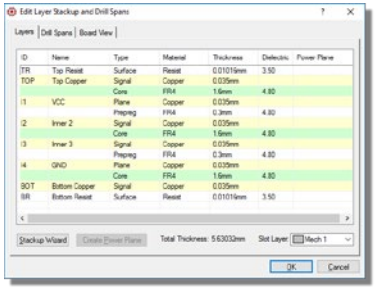


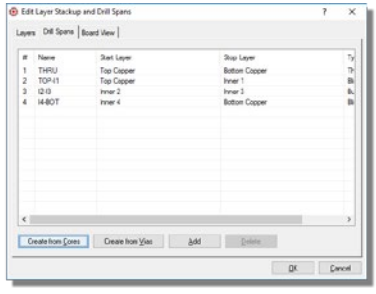
Since the defaults are fine for us, there is no action required here. If we did want to edit the stackup then we would do so via the stackup wizard button which would allow us to change the number of copper layers and the construction method for the board.

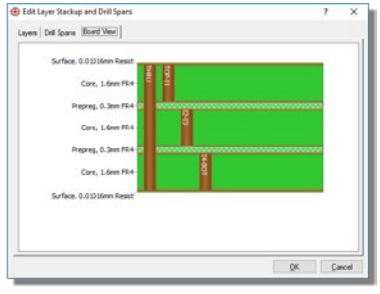




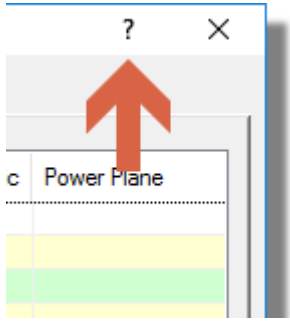
Similarly, on the drill spans tab there are no additions required, since our target PCB is 2-layer only. However, to provide some context to the topic, the following are some screenshots from a 6-layer stackup.







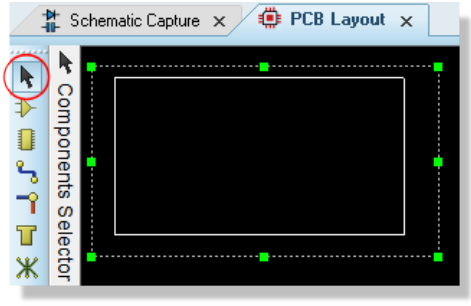
The most important thing to remember is that for multi-layer boards, configuration of this dialogue is really important. We'd encourage you to read through the dialogue level help and if you are unsure about anything, the topic is discussed in detail in the reference manual.



Work Area, Co-ordinates and Snap

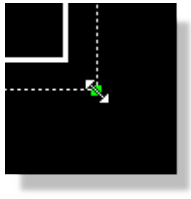
It is likely that your board edge covers a small portion of the Editing Window, which is less than ideal as all of the work will be taking place inside this area. We can of course zoom into the area (point the mouse and roll the middle mouse button or use the F6 key) but the default zoom level is designed to show the entire Work area. The Work area is the area inside the dark blue box at the edges of the Editing Window. Let's tidy this up a bit before we move on.

Start by switching into Selection Mode and then left click and drag a selection box around the board edge.



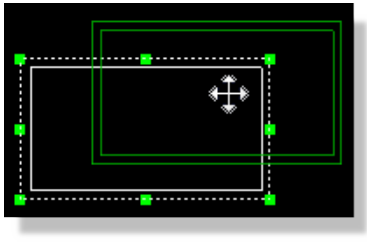
Selection mode and the board edge selected

If you get it wrong and don't completely cover the board edge, use the green handles to resize the selection box until it completely encompasses the board edge.



Resizing a box by the drag handles

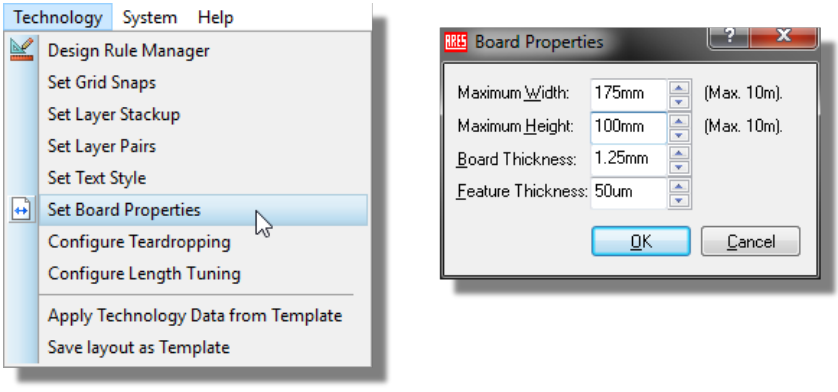
Now place the mouse inside the selection box, depress the left mouse button and drag into the centre of the work area, releasing the mouse button to commit the placement.



Moving the board edge

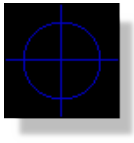
This is a very important technique worth mastering as it allows you to easily block select and then perform actions on (move, delete, etc.) groups of objects.

Having centered our board edge in the Work area we can shrink the work area to a suitable size. From the Technology Menu, select the Board Properties command and then set to something like 175mm by 100mm, such that the resulting work area forms a reasonably boundary to the board.



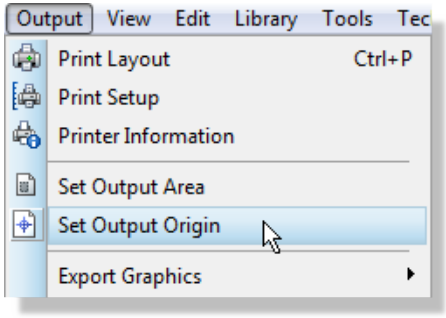
Setting the board properties

We saw briefly when placing the board edge that we can specify a temporary origin and use that to place items of a given size (or to place items a specific distance away from our origin point for example). We can however, also specify the location of the Output Origin. This is the default origin used whenever we do not explicitly set a temporary origin and is shown on the layout as a small blue marker.

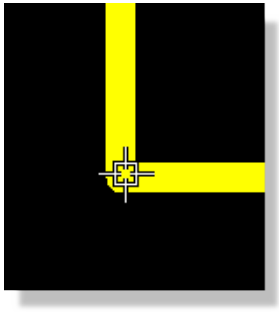


It can be useful to move this to a corner of the board and particularly so if we have mechanical constraints that we need to consider during placement (e.g. mounting hole locations). For our design, we'll move the origin to the bottom left corner of the board edge.

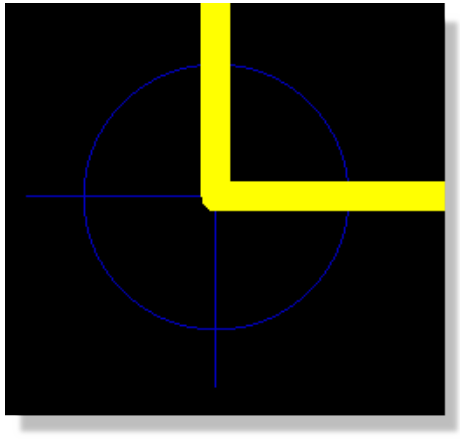
1) From the Output menu choose the Set Output Origin command.



2) Move the mouse towards the bottom left of the board edge. Roll the middle mouse button as required to zoom in for more accurate placement.



3) Left click the mouse to commit placement.



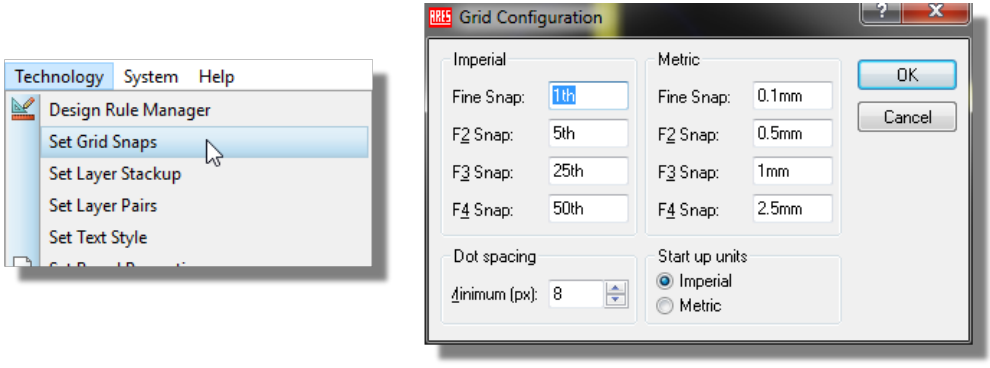
The co-ordinate display will now provide values relative to the position at the bottom left of the board by default, and relative to any specified temporary origin when set.

- Remember that you can also toggle metric or imperial units via the command on the View menu or the ‘M’ keyboard shortcut.

A useful trick once we have specified our origin is to use the Goto-XY command on the View Menu to move the mouse to the exact location we want to place position sensitive parts. This command will automatically switch between X-Y or Radius-Theta depending on whether you are in polar or cartesian co-ordinates.

ARES has a default grid and will snap objects onto the grid, making it easy to form connections and control board layout. There are four hotkeyed snap settings for both metric and imperial units, changeable from the View Menu or by keyboard shortcuts. It follows that changing the snap setting to a lower unit will allow finer granularity, whereas raising the snap setting will make it easier to select objects at higher zoom levels. It is not advisable to change the snap settings regularly through a design but rather to choose the highest appropriate setting for the board you are working on.

You can change the four default snap settings via the Grids command on the Technology Menu.

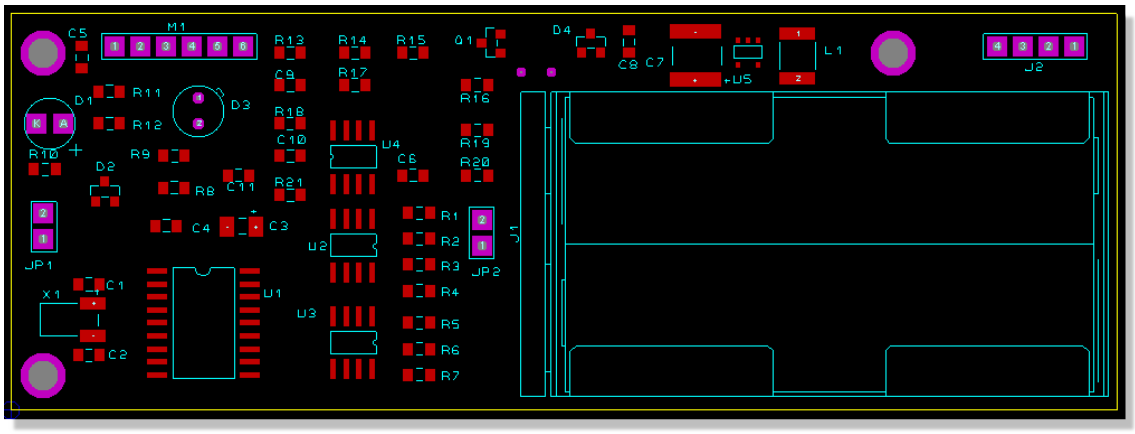


Grid settings can be set according to the users requirement

The grid display itself can be toggled between off, dots and lines via the ‘G’ key on your keyboard, whilst the colour of the grid can be configured from the Displayed Layers Dialogue (View Menu).

Placing Components and the Ratsnest

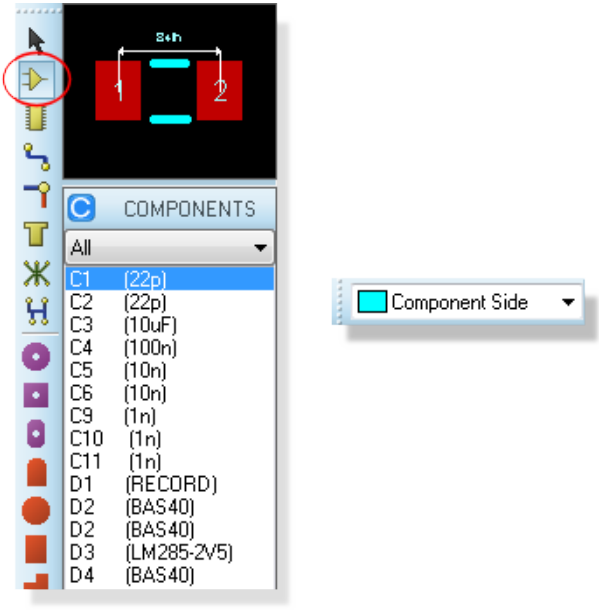
Now that we’ve handled all the basics we can finally start to get our components on the board. The following screenshot shows a fully placed board and we can use this to get our approximate positioning.



dsPIC33 board un-routed

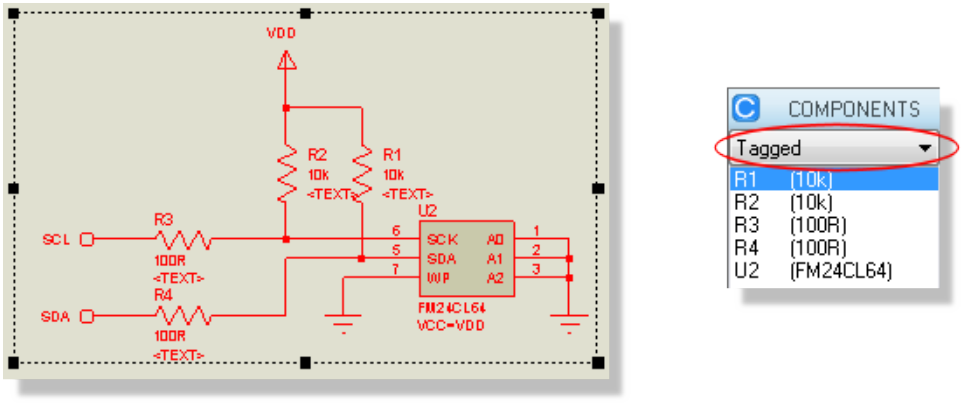
Placing a component in ARES is very similar to that in ISIS.

Start by selecting Component Mode from the left hand side icon set and then ensure that the layer selector is set to Component Side – we won't be placing any solder side components in this project.



Component mode and the Layer Selector

The object selector by default will contain a list of all parts to be placed. For more complex boards however it is worth noting that you can restrict the display to only items selected on the schematic or only those on the currently selected schematic sheet.

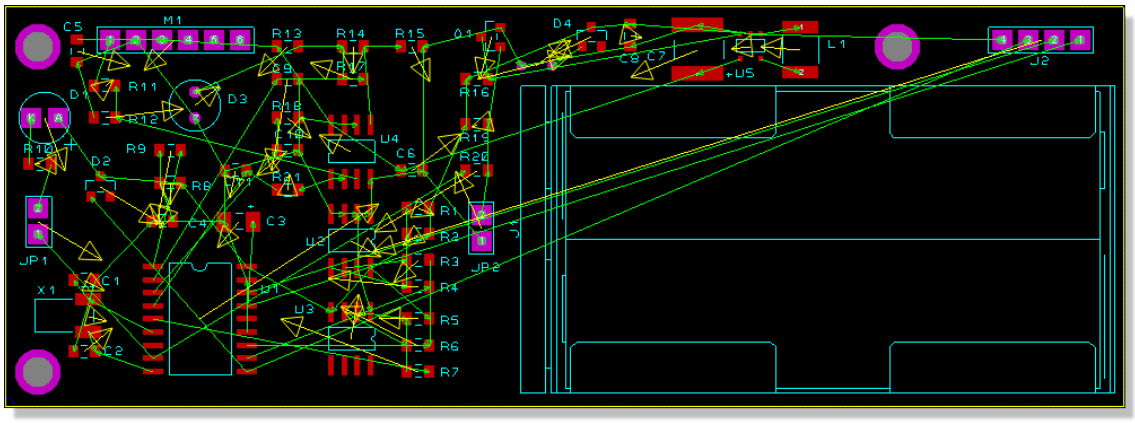


Components selected in ISIS and shown in ARES Component 'tagged' mode

This allows you to select parts of your design in the schematic and then work on placement and routing of them in ARES without the clutter of other components in the object selector. Our layout however is relatively simple so we will work globally (all of the components available in the object selector).

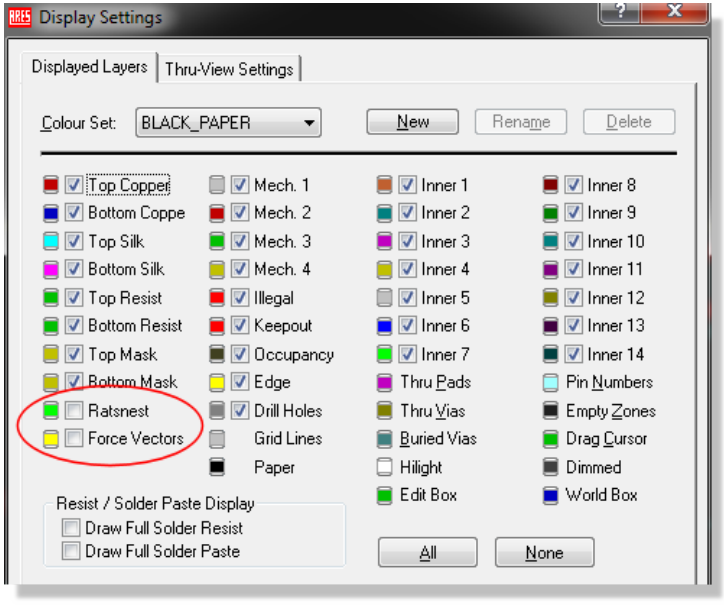
Let's get the AA battery holder down first on the right hand side of the board. Select J1 from the Object Selector, position the part using the previous screenshot as a guide and then left click again to commit placement. Note that once we have placed the part it is removed from the Object Selector and we can continue by placing the J2 connector above and towards the right of the battery.

You should notice both during placement and afterwards that there are green 'elastic' lines between the two components and also a yellow arrow from each component. The green lines are ratsnest lines and indicate connections that have to be made between the two parts, whilst the yellow arrows (named 'force vectors') indicate a preferred position for the part to minimize ratsnest distance. The force vectors are provided as guidance only and are based solely on logic to reduce ratsnest lines. Since we will be using the earlier screenshot to dictate positioning we can turn them off.



dsPIC33 board as you see it having placed all the components

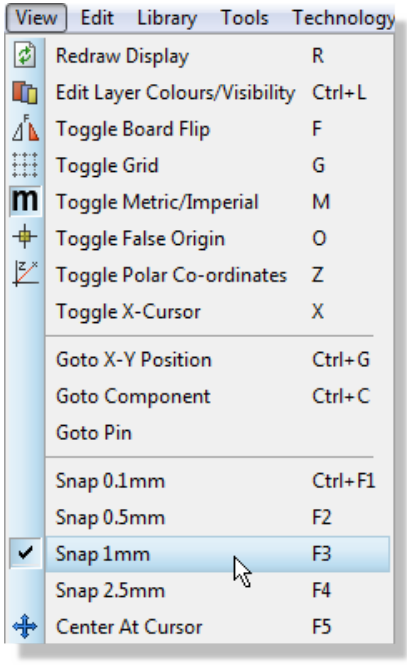
From the View Menu, select the Edit Layer Colours/Visibility command. The resulting dialogue form shows all the layers in ARES with colour and visibility configuration options. All we need do here is deselect the checkbox for the 'Vectors and Ratsnest' layer and exit the dialogue.



The Ratsnest and Force Vector layers turned off

- You can also launch this dialogue form by clicking the mouse on the status reporting bar at the bottom of the application.
- It is important to remember that this dialogue form controls visibility only; to control whether objects on a layer were selectable/editable we would use the Selection Filter which is discussed in more detail later in the documentation

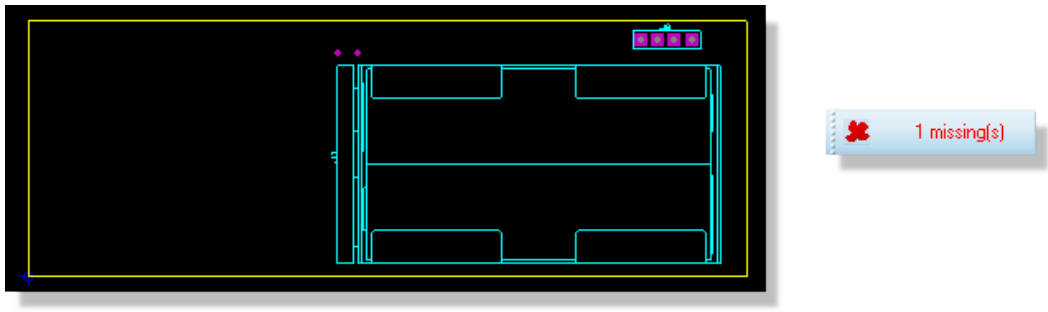
You may find that you need more control over the positioning in order to move connector J2 into position. Remember that objects are linked to a snap grid so all we need do is reduce the snap grid via the options on the View Menu, for example to 1mm snap.



If you are working in imperial units you can either change the snap to 25th or use the ‘M’ shortcut to switch to metric units

Now that we have the granularity we need, simply right click on the part to select it and then drag it into the desired position. We can move the connector closer to the board edge if we move the part label underneath the part. Again, the process is identical to the one we covered in the ISIS tutorial – remember this time to right click on the label itself and not the component body.

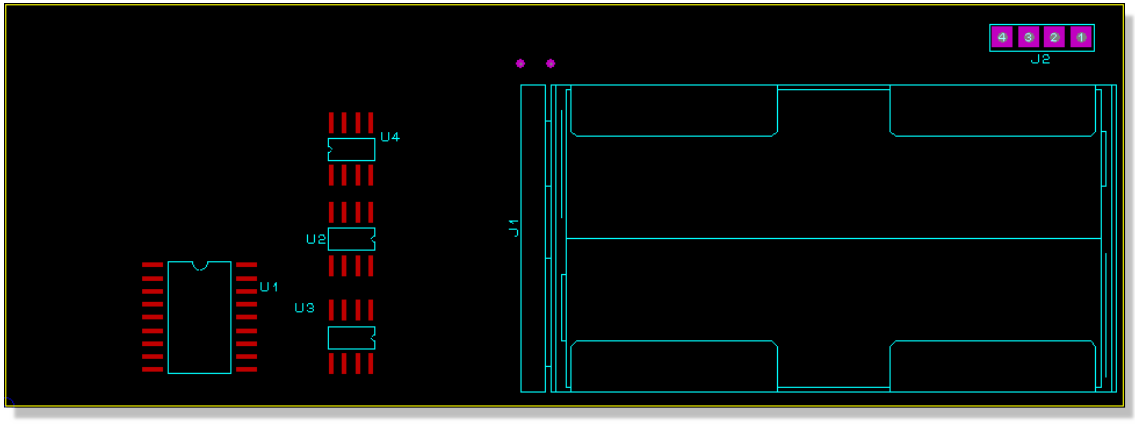
As you place components and ratsnest lines appear the connectivity status also will update to show the number of missing connections.



J1 and J2 placed with the CRC showing 1 connection missing

A missing connection is essentially a wire connection on the schematic that has not been routed on the layout. It follows that when the board is completed the connectivity status should report no missings. We'll look at this is more detail when we discuss routing later in the tutorial.

If we now consider the other principle components we can go ahead and place U1 (dsPIC33), U2 (I2C memory), U3 (temp/humidity sensor) and U4 (dual op-amp) in exactly the same way, such that we end up with something like the following screenshot.



dsPIC33 board so far...

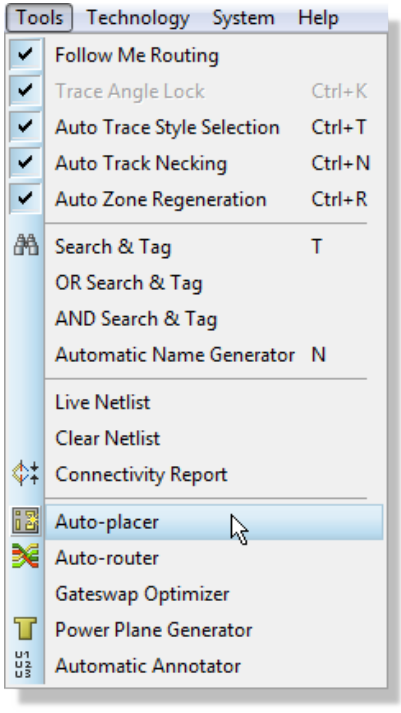
Note that the S08 footprints used for the U2, U3 and U4 components have been rotated appropriately to reduce ratsnest length. This is best done at placement time as the ratsnest display will update live to give visual guidance.

Start by placing U1 in the normal way and then start placement of U2 by left clicking once on the Editing Window. Now use the '+' and '-' keys on your numeric keypad to rotate the component as you move it into position, left clicking the mouse again to commit placement as before. Repeat the process for U3 and U4, moving or rotating parts after placement (right click and then context menu options) until your board approximates the previous screenshot.

If you are placing components in a high density area where routing space is at a premium you can hide the part reference via the edit component dialogue form (right click on the part and select edit properties from the resulting dialogue form).

Generally speaking you have two options for placement of the board. You can either manually place the components in their positions or, if you have the Advanced Features Set (Proteus PCB Design Level 2, Level 2+ or Level 3), you can use the Auto-placement feature to get all the components on the board and then move them into the desired locations. In either case, you may find it useful to temporarily disable the ratsnest lines during placement; remember that this is controlled from the Layers dialogue form as discussed earlier.

The auto-placer can be invoked from the Tools Menu in ARES and for our purposes all the default options will suffice.



Selecting the Auto-Placer from the Tools menu

Whichever route you follow the task is to lay out all the components on the layout, using the previous layout as a reference. Try to leave some space at the bottom of the board so that we can run traces down and along from connector J2.

The following points are useful to bear in mind as you continue placement:

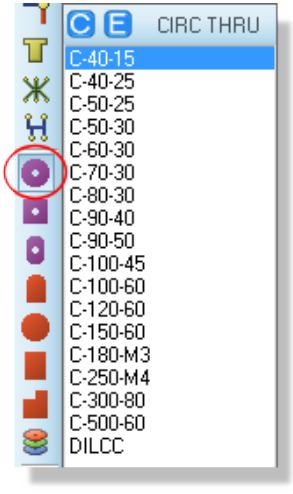
- The middle mouse will zoom you in and out as you place components (as will keyboard shortcuts F6 and F7)
- Right clicking on a component once placed will present a context menu option, allowing you to move, rotate or delete the part.
- Changing the snap settings to a finer grid will allow more accurate positioning at the expense of more precision being required to select the part.
- If you place a part in an illegal position (e.g. over another part) you will get one or more design rule violations. For now, simply move the part to a legal position – we will look more closely at design rules in the next chapter.

DRC errors

Once you feel comfortable with placing and moving parts, feel free to move on to the next section. We will load a board with all the parts placed before we discuss design rules and routing.

Mounting Holes and Pad Styles

Before we move on to look at connectivity considerations we should complete the physical layout by placing mounting holes for the board. In our case we want to use pad with 3mm hole and a diameter of around 0.18in and to position them conveniently for mini-locking PCB supports. The first thing we need to do is switch into circular through hole pad mode and scan for a suitable pre-supplied pad in the Object Selector.

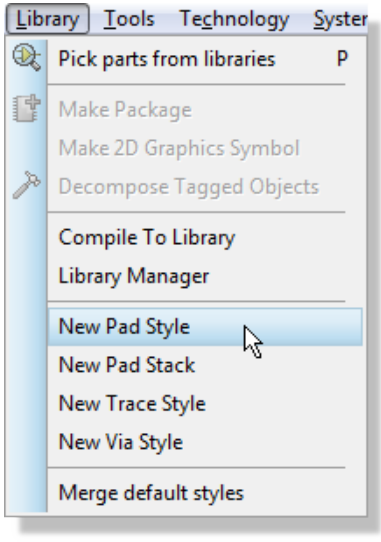


Select the correct pad style for mounting holes

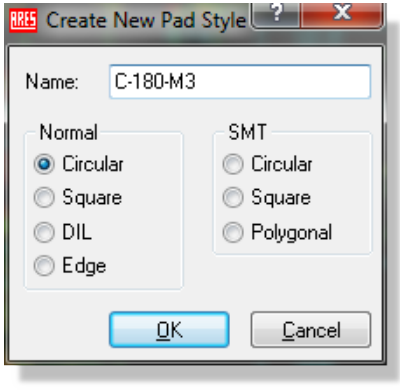
The nomenclature of pads in ARES is designed for easy reading and tends to follow the following format:

`<PAD TYPE> - <DIAMETER/SIZE> - <HOLE>`

Units are in imperial unless prefixed with an 'M' so for example a C-40-15 is a circular pad with a 40th diameter and a 15th hole and a C-200-M3 is a circular pad with a 0.2in diameter and a 3mm hole. Our spec requires a pad with a 0.18in diameter and a 3mm hole, which we can see does not exist in the pre-supplied set. We therefore need to create the pad as follows:
1) From the Library Menu select the New Pad Style command.

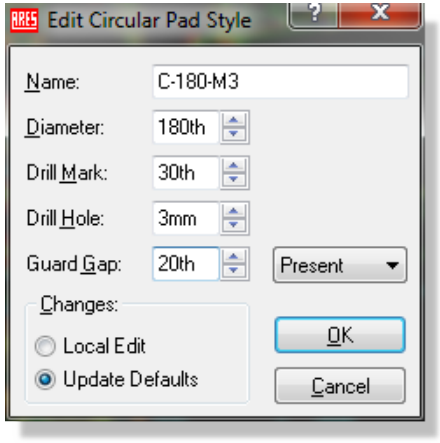


2) Enter a name for the pad; we recommend you follow the standard naming convention (i.e. C-180-M3).

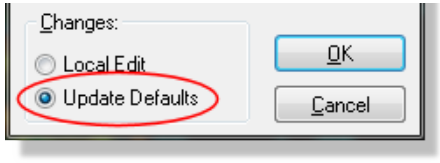


3) Specify the pad type – in our case we want Circular through hole pad.

4) The diameter of the pad is 180th or 0.18in.The drill mark is the size of the mark output in a drill plot, 30th would be fine. The drill hole needs to be 3mm and the guard gap should be enlarged to 20th. The guard gap (or solder mask expansion) is the amount by which the pad diameter is expanded on the resist plot.



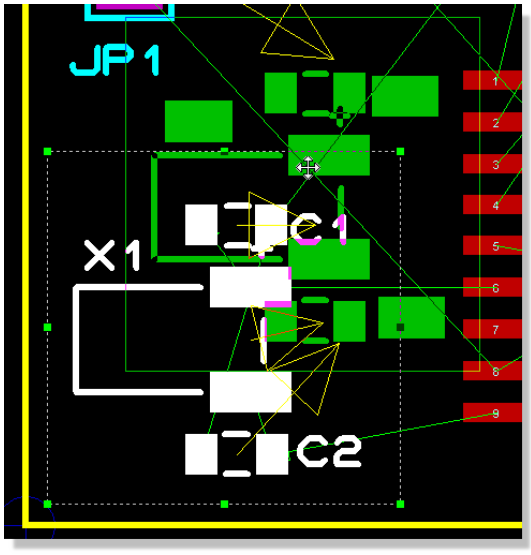
5) At the bottom of the dialogue form we have the choice whether to make this pad style permanent for future designs (Update Defaults option) or local only to this design (Local Edit option). Unless there is a particular reason for a local edit only we recommend that you leave this on the default setting.



6) When you exit the dialogue form you should see that the new pad style is available for placement from the Object Selector.

When entering specific values you may find it easier to simply type in the values that you want rather than using the up and down arrows beside the controls.

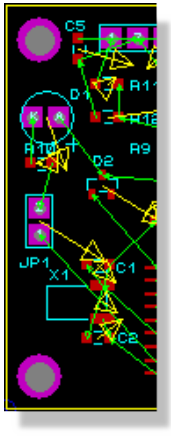
We are going to want to place two mounting holes at the top and bottom left of the board and then a third at a specific location (for mini-locking PCB supports). You may therefore have to move some circuitry out of the way to make room; for example, the crystal block at the bottom or the pressure transducer at the top. As we've seen earlier we can easily do this by entering selection mode, drawing a selection box around the circuitry and then dragging to a new position. This is shown below with the crystal circuitry.



Moving a selection of circuitry

Remember that you can resize the selection box to include or exclude items using the drag handles if you don't get the right size the first time.

Once you've cleared some space, go to the circular PTH pad icon again, select theC-180-M3 Pad style and place two at the top and the bottom left of the board.



Mounting holes placed

We want to position the third mounting hole above the battery and in a reasonably accurate position. Specifically, we want to place the hole 35mm up from the bottom of the board and 87.5mm along from the left. We've already set our global origin to the bottom left of the board so this is set to be the reference point for the co-ordinate system. All we need do therefore is move

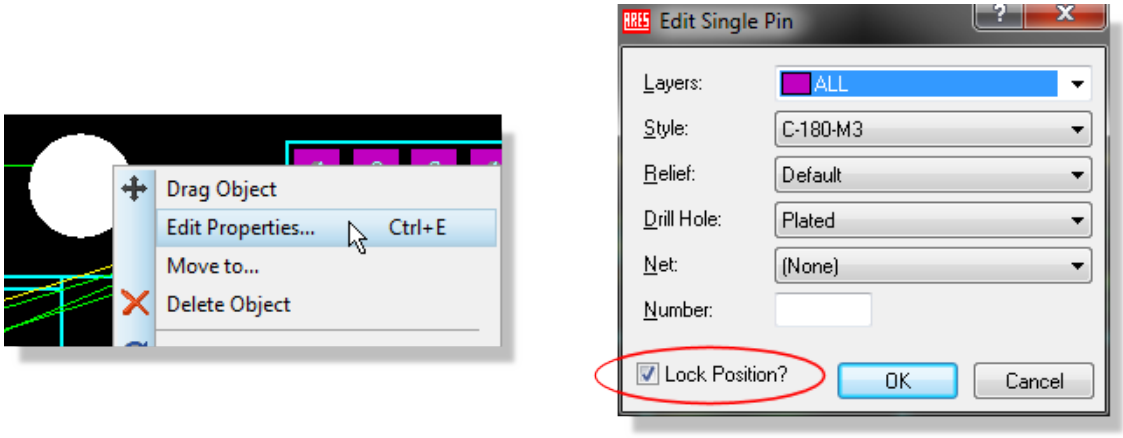
the mouse up and right from the bottom of the board edge until the co-ordinate display reads correctly (you may need to switch into Metric units using the 'M' shortcut key toggle).

If required, we can move the DC/DC converter circuitry as before, then switch back to pad mode and place the final mounting hole at the correct co-ordinates.

- Alternatively, having set the output origin and knowing the co-ordinates you can right click on the mounting hole and select the move-to command from the context menu to programmatically position the part.

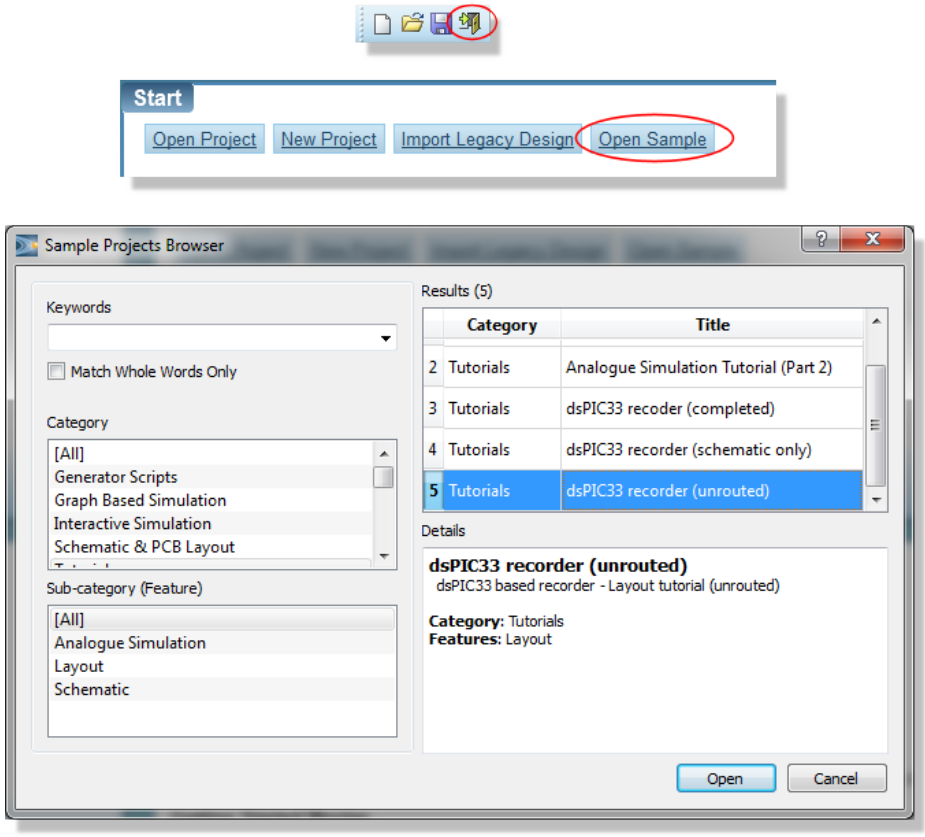
With position sensitive components it is often useful to ensure that they cannot be nudged or moved inadvertently after placement. You can lock any object in position by moving the mouse over the object until it is encircled by a dashed line, right clicking on the object and selecting Edit Properties from the resulting context menu. The Lock Position checkbox can then be selected to prevent movement or deletion of the part.

- You can also set a specific minimum zone clearance for the pad to reserve some extra space for the bolt.



Locking pads and components

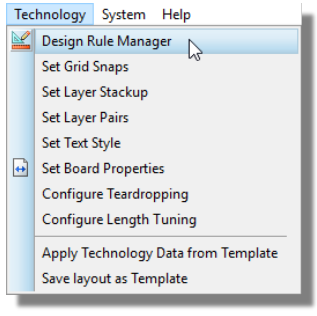
We have now completed the physical layout of the board. If you decided not to follow the full layout process you can load a version of the board in its current form by opening the home page, clicking on the Open Sample button. The project you want this time is the unrouted dsPIC33 Recorder project from the Tutorials category.



Opening a sample project

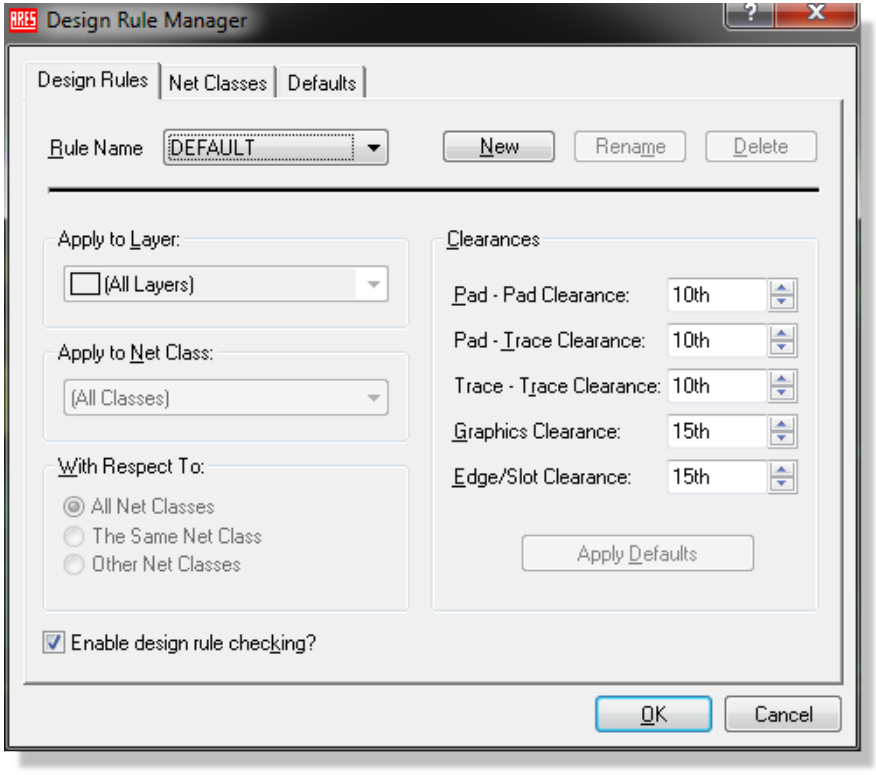
Design Rules and Net Classes

Now that we have a placed board we need to configure the software to obey any design constraints or electrical considerations relevant to the layout. We can do this largely from a single dialogue form called the Design Rule Manager. Start by launching this dialogue form now from the Technology Menu in ARES.



Design Rules

The first tab of this dialogue form allows us to configure constraints and minimum clearances for the layout. We have a DEFAULT rule loaded which must apply to all layers and all net classes, providing a set of clearances between objects equivalent to manufacturing guidelines.

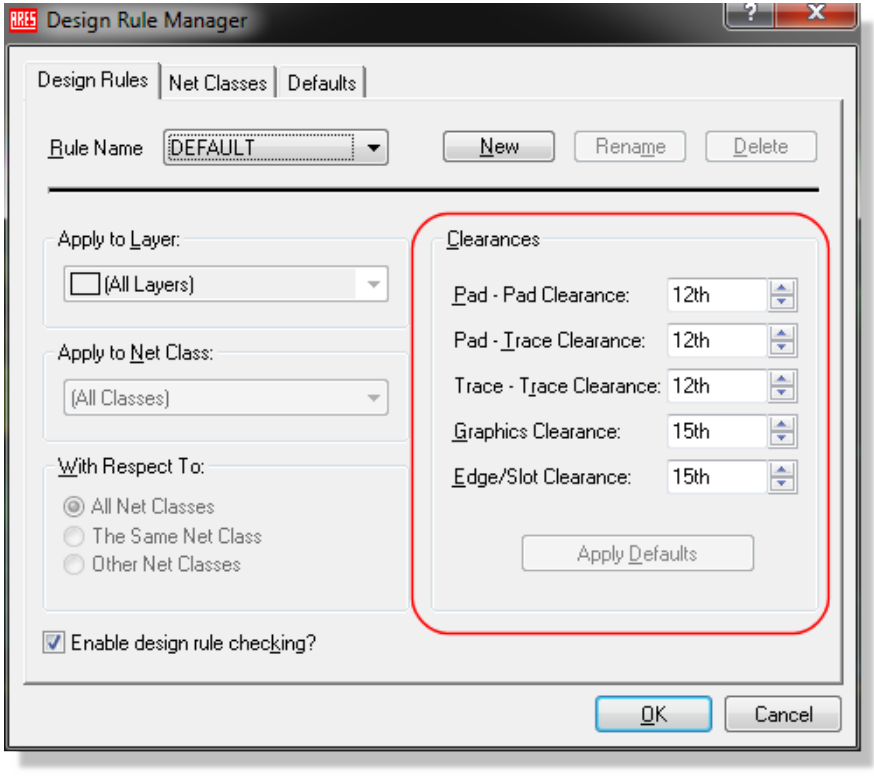


The Design Rule Manager

This rule and these clearances are created automatically for each layout in order to provide a minimum set of constraints for a board. You can change the values of the default clearances applied to new layouts via the Design Rule Manager on the Technology menu.

The first thing we need to decide is whether a single rule is suitable for all layers and all traces on the layout. It is possible to create as many new rules as required and we can limit their effect to a given layer and/or a given set of connections (net class). A great deal of information on this is available in the reference manual but for our purposes we can manage fine with configuration of the existing rule.

Given that this equipment is intended to work out-doors we'll need to increase the clearances between pads and traces to improve the insulation against condensing moisture. A 20% increase should be sufficient so we need to change the Pad-Pad, Pad-Trace and Trace-Trace clearances from 10th to 12th.

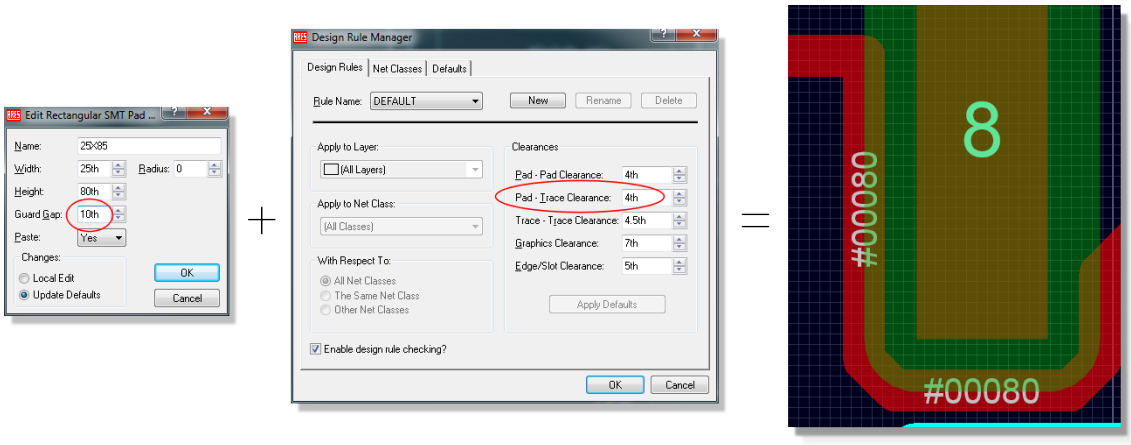


Default clearance settings in the DRC manager

The graphics and edge clearances are fine at the default values.

> Note that the edge/slot clearance field defines the clearance for power planes against the board edge.

In addition to these clearances there is a board global rule for Resist-Trace clearance. This rule requires no user interface and tests for exposed slivers of copper that can occur if the guard gap (solder mask expansion) on a pad style is larger than the pad-trace design rule.



If, during route placement, you find these DRC errors, the solution is to either move the trace or edit the pad style and change the guard gap (solder mask expansion) for that pad style.

Since there are no new rules required for this board we can move on to the net classes tab of the dialogue.

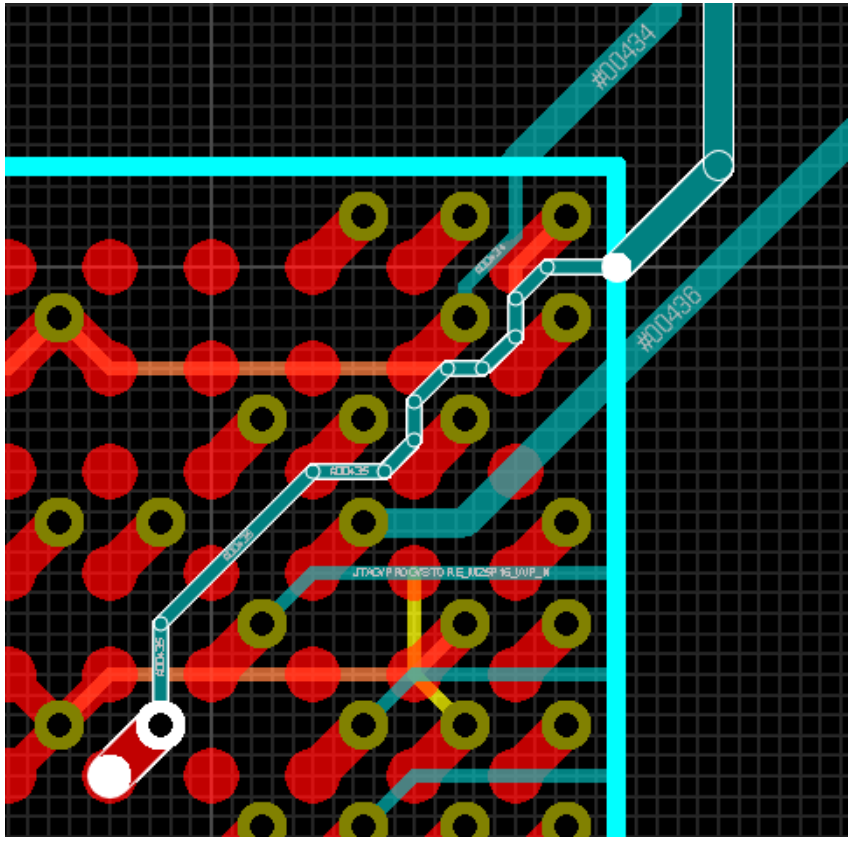
Net Classes

This is the place where we configure trace and via styles and control which layers we route on when auto-routing the board. The selector at the top allows us to switch between different net classes and configure each separately.

Let's start with the POWER Net class, which should be the default selection. As we discussed in the ISIS Getting Started Guide any nets that include a power or ground terminal are automatically assigned to the POWER net class, unless manually overridden.

We'll set the track / trace size to 25th, not so much for current considerations but to reduce track impedance (we'll also place a single low impedance ground plane later to help with this). In ARES nomenclature this corresponds to track style T25.

The neck style – if configured – specifies the track style for necking (thinning) to help the track pass through obstacles. Together with our design rule aware routing (which knows which objects are obstacles) this is very powerful. When set, you can hold the SHIFT key during route placement to toggle between necked and normal track styles which is invaluable in tricky or dense routing.



Advanced necking: The SHIFT key is held down while routing the escape from the BGA and then released to resume routing at normal track width.

Our board is not likely to be complex enough that this will be an issue but we'll set the neck style to 15th so that we can practice.

Given that we don't really have current constraints the choice of vias is a tradeoff between the higher manufacturing costs of smaller vias and the decreased routing quality of large vias. For the current this circuit is dealing with and for a standard 1.6mm FR-4 support plated with 35um copper, vias with the standard 0.4mm hole are a good compromise. For the power tracks we need a suitable annular size and 40th is a reasonable choice. We can therefore set the via style to be V40.

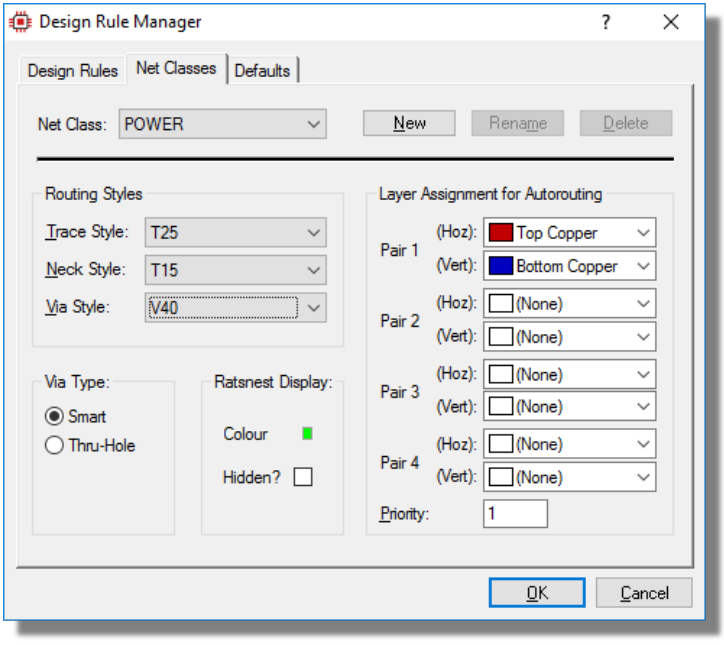
> If you are not sure of the characteristics of a particular style, you can view them by selecting the appropriate mode icon (via, track, pad, etc.), highlighting the style in question in the Object Selector, and then clicking on the 'E' button at the top of the Object Selector.

The options at the bottom of the dialogue form allow us to change the via type (clearly not relevant on a two layer board) and also to change the colour or visibility of the ratsnest lines.

The latter can be useful if we are manually routing and wish to quickly distinguish between power and signal connections.

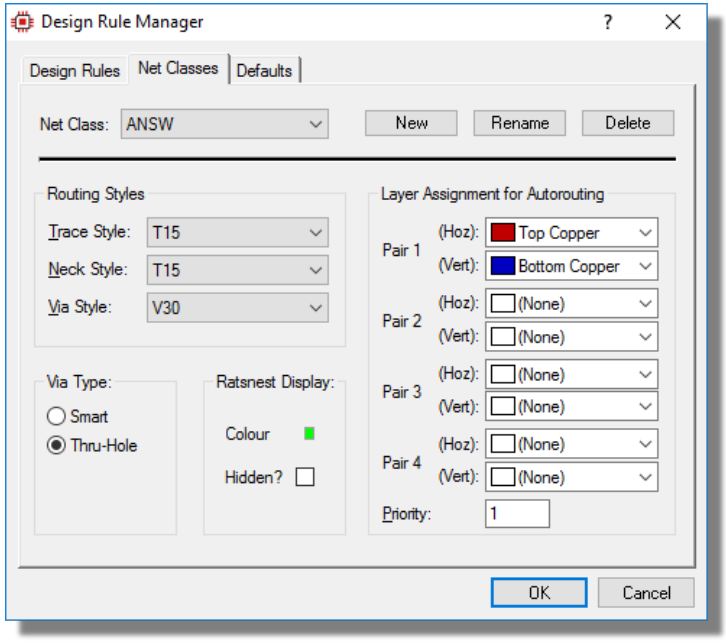
The layer assignment pairs on the right hand side tell the autorouter which layers to route on for multi-layer boards. Again, for our two layer board, there is nothing to configure here.

We should now have finished setting up the POWER net class – your dialogue form should now be configured as per the following screenshot.



Specifying the Power net settings

Let's move on to the next net class in the selector; the ANSW class. You may remember that we specifically named this net class in the ISIS Getting Started Guide in order that the 5V switched power supply for the analog circuitry (the output of the DC/DC converter) could be handled separately in ARES. What we want for these connections is a track size larger than the standard SIGNAL net class but smaller than the POWER net class, so let's change the Trace Style to T15 (15th track). For consistency, the neck style can also be T15 and we can use a smaller via (with the same 0.4mm hole) by selecting the V30 Via Style.



Specifying the ANSW net settings

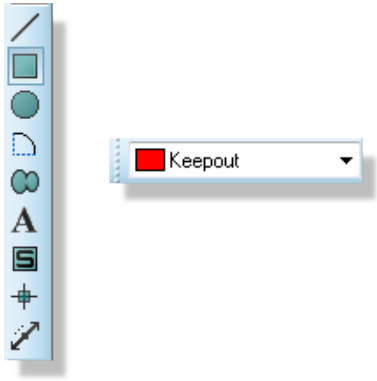
- If you don't have a V30 via style you can create one easily by going to the Library Menu, selecting New Via Style and filling out appropriately.
- Creating your own net classes is extremely simple and provides you with the flexibility you need both with regard to trace and via configuration but also with design rule constraints

The final net class we have is the standard SIGNAL net class that encompasses all non-power and unspecified connections. We've deviated from the widely used "8/10 Rules" (8th trace, 10th clearance) in order to compensate for outdoor usage but the standard 8th track width (style T8) is fine. We can also set the neck style to 8th and keep our standard V30 via style for these connections.

Having finished configuration, exit the dialogue form to return to the layout.

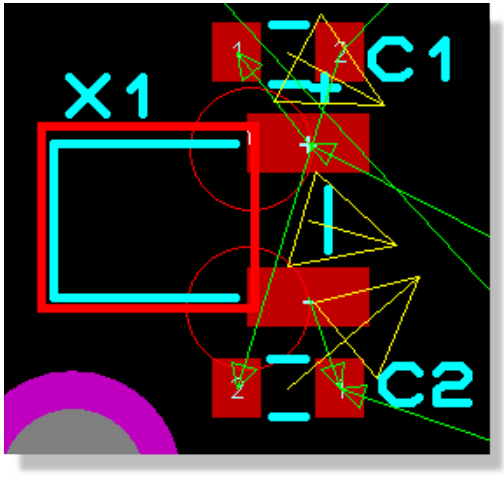
Keepout Areas

We can also introduce tracking constraints by limiting areas in which tracks can be placed. A good example of this is the crystal towards the bottom left of the layout where we do not want tracks under this area. To form a keepout area, start by selecting the 2D Box graphic icon and changing the layer selector to be keepout.



Selecting the Keepout layer

Next, place a small box around the silkscreen of the crystal in exactly the same way as we did for the board edge (left click to start placement, drag out the area, left click again to commit placement).

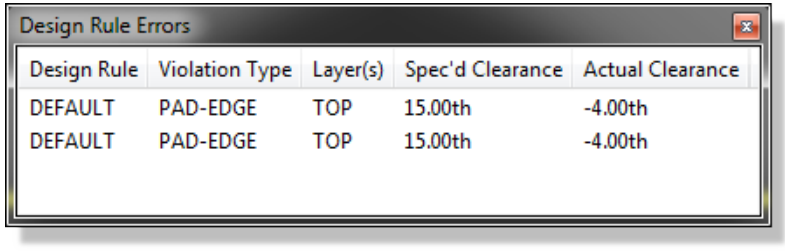


Keepout placed produces 2 DRC errors

Unless you are very skillful you will be presented with a box indicating that design rules have been detected.

First DRC warning

Checking the box on the dialogue form will prevent it from appearing in the future. If you left click the mouse on the DRC section of the status bar a small window will appear providing information on the errors. You should see that they are of type PAD-EDGE as the pads of the crystal are closer to the keepout graphic than the 15th specified in the Design Rules.



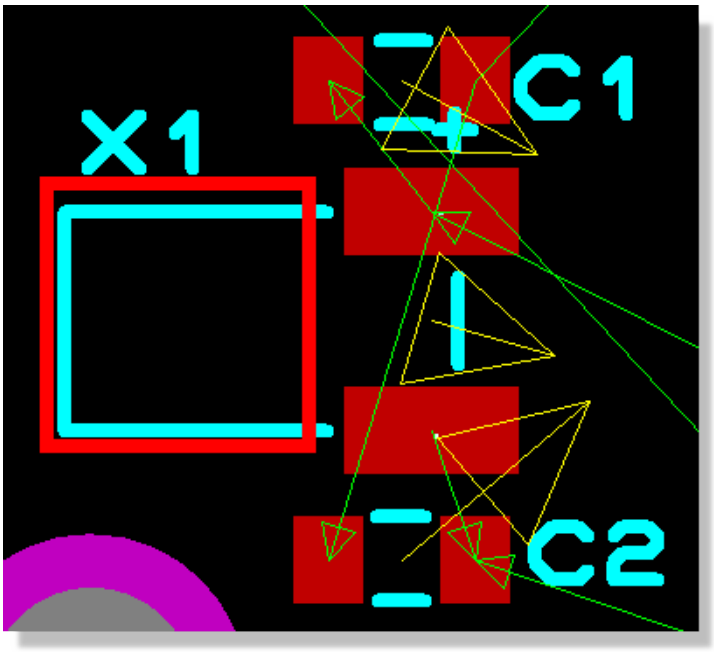
Displaying DRC errors

We have two options here:

1) Ignore the DRC errors. The keepout graphic will not impact on connectivity.

2) Move the graphic to a legal distance from the pads. The easiest way to do this is to first change the snap level down (View Menu), right click on the graphic and select the Drag Object context menu option, and then change the snap level back when you are finished.

When you are finished your keepout area should look something like the following screenshot.

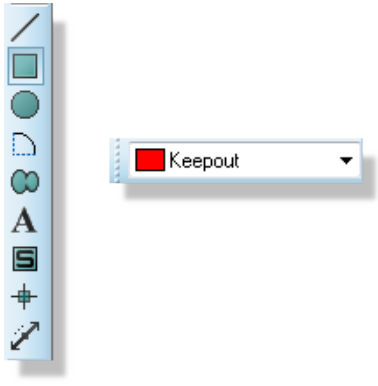


Correctly placed Keepout

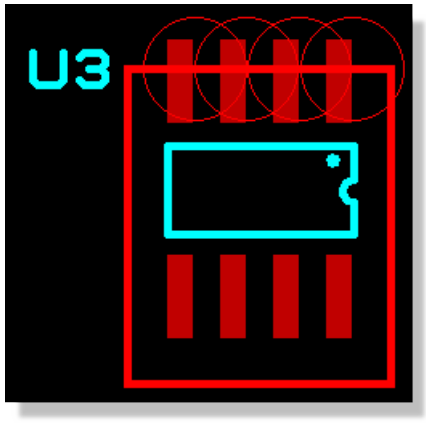
The default for the keepout is that it applies to all layers. However, after placing the graphic on the keepout layer, you can right click change it to apply only to a specific copper layer from the resulting context menu. This can be very useful on more complex or densely packed board.

We have a similar problem around the temperature/humidity sensor (U3). We are going to want a slot machined on the board to cut the thermal path and reduce the measurement errors of the sensor (we want to measure environment temperature, not the one conducted by the PCB). We therefore need to ensure that no tracks are placed in the area we are going to slot through:

1) Select the 2D Graphics rectangle icon and then change the layer to be KEEPOUT.

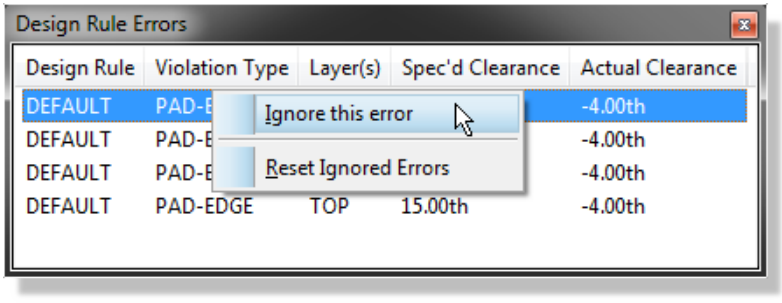


2) Place a rectangle around the bottom half of the IC as shown below.



U3 with a Keepout

This time we cannot move the keepout to remove the DRC errors as the positioning is sensitive. Instead we can manually override the errors by launching the DRC dialogue, right clicking and selecting the Ignore option from the resulting context menu.



Ignoring DRC errors

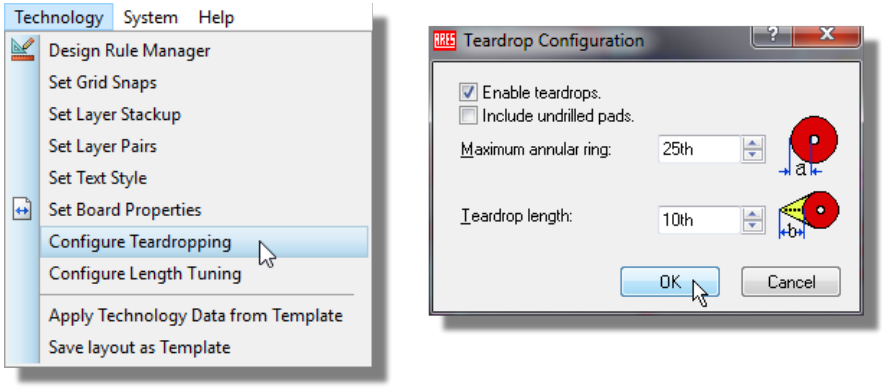
Routing the Board

Having configured the board constraints we can now move on to actually making connections and routing the board.

Teardrops

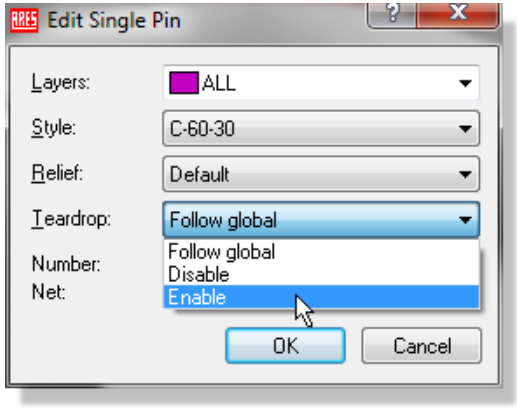
> Teardrop functionality is included in the Advanced Feature Set and requires a license for PCB Design Level 2 or higher.

Placing teardrops at the connection point between track and pad is often useful in preventing drill breakout during board manufacture. We can configure teardrops from the Set Teardropping command on the Technology Menu.



The dialogue form lets us set the maximum annular ring size on a pad below which a teardrop connection will be made and then also lets us set the length of the teardrop. You can, if you want, also apply teardrops to surface mount pads although we are not aware of any manufacturing reason for doing so. On surface mount pads the annular ring parameter is considered to be center pad to nearest pad edge.

A per pad override for setting/clearing a teardrop connection can be accessed from the edit single pin dialogue. This provides an additional level of control over where teardrop connections are made.



Note that the creation of a teardrop connection is dependent on the design rules. If making a teardrop connection would violate one of the design rules then it will simply not be created.

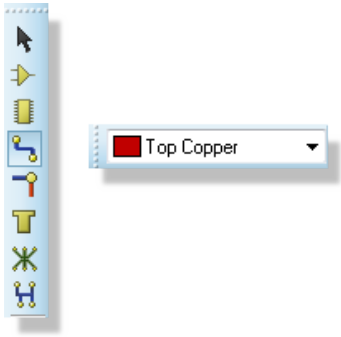
Once configured teardrops connections will be made dynamically as we complete routes in the sections below.

Placing a Route Manually

Before we start, we will want the ratsnest lines turned back on. If you have them disabled (can't see any green lines between pads), turn them on from the View Menu - Displayed Layers dialogue.

Let's begin by manually placing some tracks on the board. Typically, we would manually lay out connections where a specific path is desired for the track or where we need greater control over track position. On our layout we want to make sure that our connections from connector J2 follow a sensible path around the board so we can start here and route them manually.

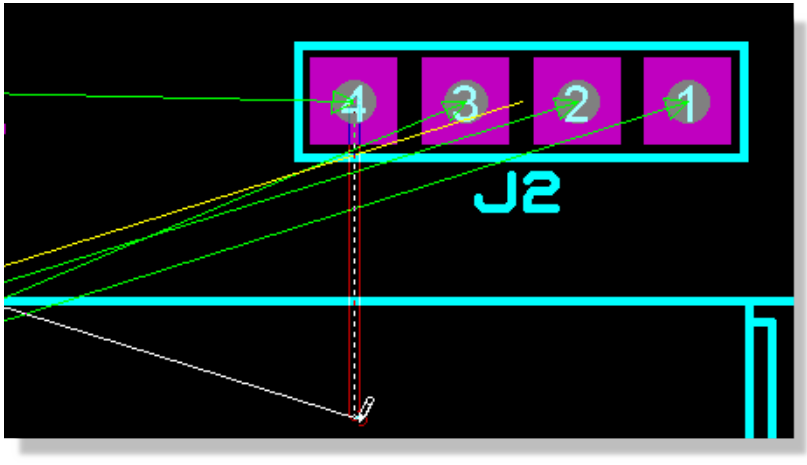
Start by selecting track mode at the left hand side of the Object Selector and changing the layer selector to be on Top Copper.



Trace / Track selection and the Layer selector

If we look at pad 4 of the J2 connector we can see that the closest ratsnest line is directing us across to the GND pad on the step converter. This is not ideal as we would have to navigate the mounting hole and then track into the small SMT pad. Instead, we will route this down and across the bottom of the board to a more convenient ground connection.

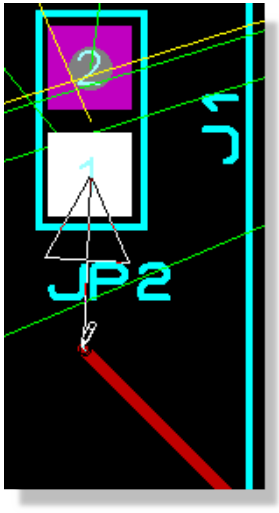
ARES features a sophisticated 'follow me' routing algorithm for manual routing in which the route being placed will follow the path of the mouse as best it can while still obeying all of the design rules for the board that we set up previously. You start track placement by left clicking the mouse on pad 4 and then move the mouse downwards.



Routing J2 Pin 4

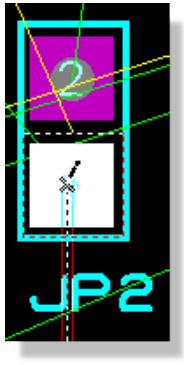
You should see that the closest legal destination for the track is now highlighted in white. This will update as we move the mouse and since we are not going to route to this destination we can ignore it for now. When the mouse approaches the bottom of the board, you could just change direction to the left and the track will corner to follow the mouse. However, since we want a tighter corner it is better to left click the mouse to place an anchor and then change direction to the left.

You will see that as we move further to the left the ratsnest guide will change to show us that we can terminate the track on pin 1 of jumper JP2.



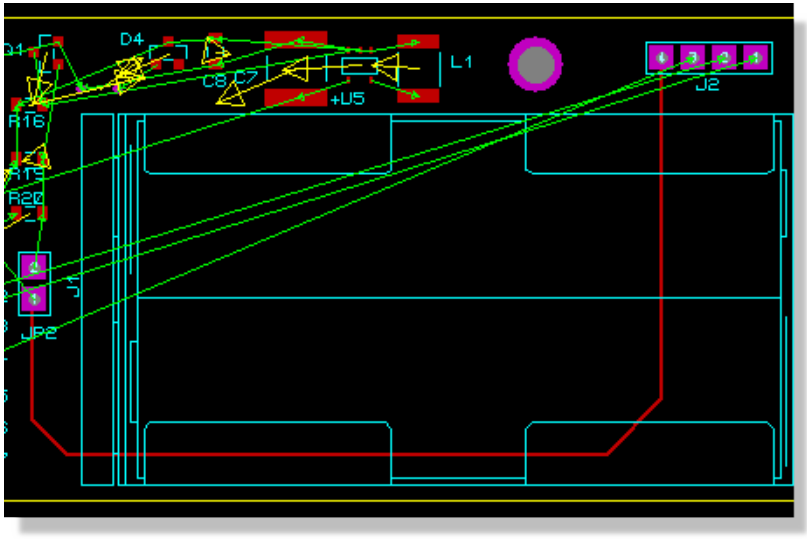
Connecting to JP2 Pin 1

Once we are underneath the jumper we can change direction and move upwards towards the destination pin



Connecting the trace to the pin

Finally, we can click left over the pin to finish placing the route. This will both commit the route and remove the ratsnest line corresponding to this connection.



Finished route from J2 pin 4 to JP2 pin 1

Note that we did not need to select the width of track to route with as we configured this in the previous chapter. ARES recognizes that we are routing a track on the GND net, applied the rules for the POWER net class and selected the specified 25th track style for us.

Manual routing is probably the most common action you will perform with the software and it is vital that you understand how it works. The basic rules of operation are:

- Left click on pad, track or zone border to begin routing from that object.
- Left click at any point during routing to commit the route up to the mouse position (we call this anchoring).
- Right click to terminate the route at its last committed / anchored point.
- ESCAPE key to abandon route entry completely.
- SPACEBAR to float a via on the end of the route and left click to then place the floating via.
- Double left click to drop a via at the current mouse position.
- Move the mouse backwards over existing tracking to rub-out.
- SHIFT key toggles the neck style on/off (SHIFT key down = neck style on, SHIFT key up = neck style off).

We **strongly** recommend experimenting with manual routing for a while on this board until you are comfortable with how it works. The following is a summary of how to perform common actions; try these while routing any of the remaining connections.

Panning and Zooming

While placing a track you can use the middle mouse wheel (or F6 & F7 keys) to zoom in and out during routing. Panning will happen seamlessly when the mouse is at the edge of the Editing window during routing.

Placing Anchors

The follow me routing algorithm will move the track being placed according to the way you move the mouse. If you want a track to follow a particular path then you need to help by left clicking whenever you change direction. This places an anchor or commits the route up to the mouse pointer such that the follow me router will not change it. You will see this happening as the outline track becomes solid.

Getting Stuck / Re-routing while routing

Since the manual routing system obeys the design rules for the board you don't need to worry about clearances while placing routes. You can however route to a place where you are blocked (the routing icon will change to a no-go sign at this point). Often you just place a via and continue on but sometimes it is better to rewind and try another path across the current layer.

Moving the mouse backwards over the route being placed will rub out that portion of track so to rewind and change direction just move the mouse back to the last good point - anything you had placed from that point onwards will be removed.

- On densely packed boards in particular, speed of mouse movement is an issue. Remember that you are guiding the placement so moving slowly through tight spaces will work far better than ripping the mouse from source to destination. Use the SHIFT key to neck down and back up - this will get you through tight spaces.

Placing Vias

If you double click during placement you will place a via at the point the mouse is at and can then continue routing on the associated layer. If you press the spacebar you will float a via on the end of the mouse and can then position the via manually before placing with a left click. Using the spacebar has the advantage of snapping to legal objects (e.g. via under SMT).

In either case the placement of the via is also design rule aware and ARES will not let you drop a via in an illegal spot. If you use the floating via method (with the spacebar) ARES will attempt move the via to the nearest legal spot. This can be extremely useful on busy boards or when you want to butt a via tightly against another object.

Layers used for vias are defined in the Layers Pairs dialogue on the Technology Menu. Examples of routing with vias and discussion on layer pairs follows further in the tutorial.

Object Hugging

Since the design rules are all live during placement it is quite easy to hug a track to another track or to wrap a track around an object. If the mouse is over an object where the route being placed cannot travel then the route will follow the mouse path as closely as it legally can, essentially hugging the barrier object.

Abandoning a Route

- If you want to abandon the route at the last committed (solid) segment then right click the mouse.

- If you want to abandon the route at the mouse pointer then left click to commit up to mouse pointer and then right click to terminate the track.

- If you want to abandon the route completely then hit the ESCAPE key.

Making Connections

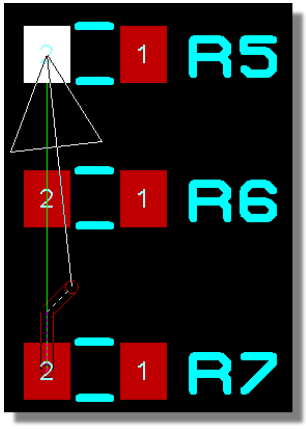
If you are connecting directly to a pad then a left click of the mouse will complete the connection and terminate route placement.

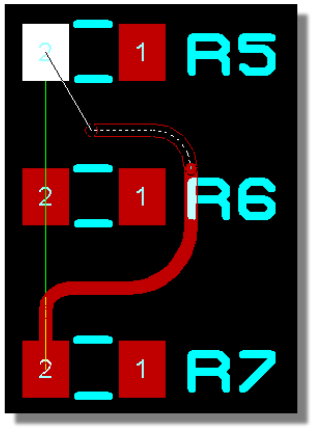
If you are connecting to a track then left click will commit the placement and right click will terminate the route forming the connection. The same procedure applies if you are connecting to a zone except that you must connect to the zone border.

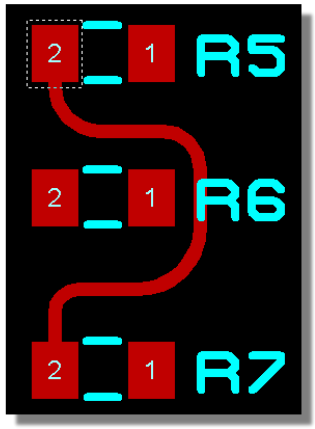
Placing a Curved Route

Placing a route with curved corners can be helpful with signal integrity and net tuning or simply for aesthetic reasons. In Proteus, you can switch to curved route placement by holding down the CTRL key during route placement. Releasing the CTRL key will immediately switch back to linear routing giving you the control to curve only specific parts of the route. The process of placing the route is exactly the same as before, except we hold CTRL down on the keyboard for curves. So:

1) Left Click the mouse to start the route.
2) Hold CTRL down to curve the corners and guide the route towards the destination pad with the mouse.
3) Left click the mouse on the destination pad to terminate the route.



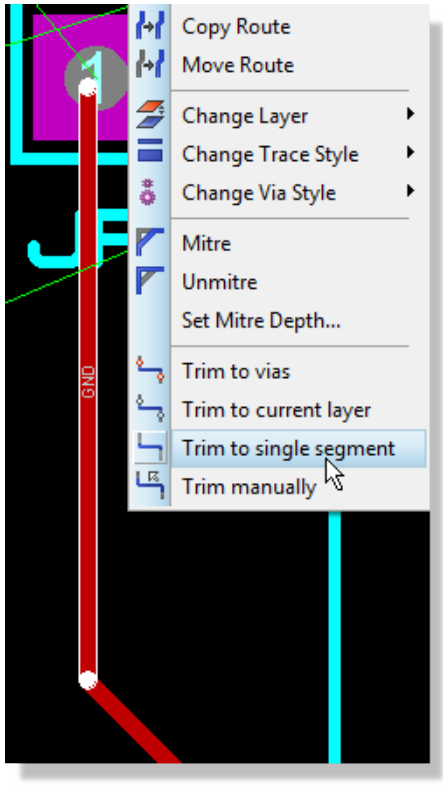




Deleting a Route

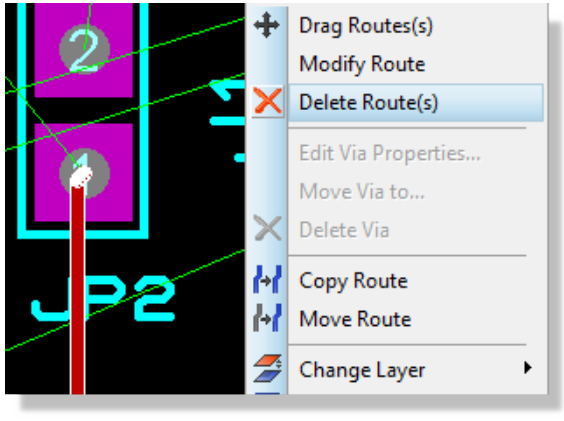
Once the route has been completed we can delete either the full route or a specific part of the route if we are not happy with the placement. Let's assume that we were not happy with our placement of the last piece of track (upwards to pad 1 of JP2).

Start by right clicking on the track over this segment of track. This will highlight the entire track and the Delete Route option near the top would then remove the complete trace from the board. However, we have far more control if we use the options at the bottom of the context menu. In this particular case, select the Trim to Single Segment option; this will change the selected area of track to the segment we have clicked on.



Trimming a trace to a single segment

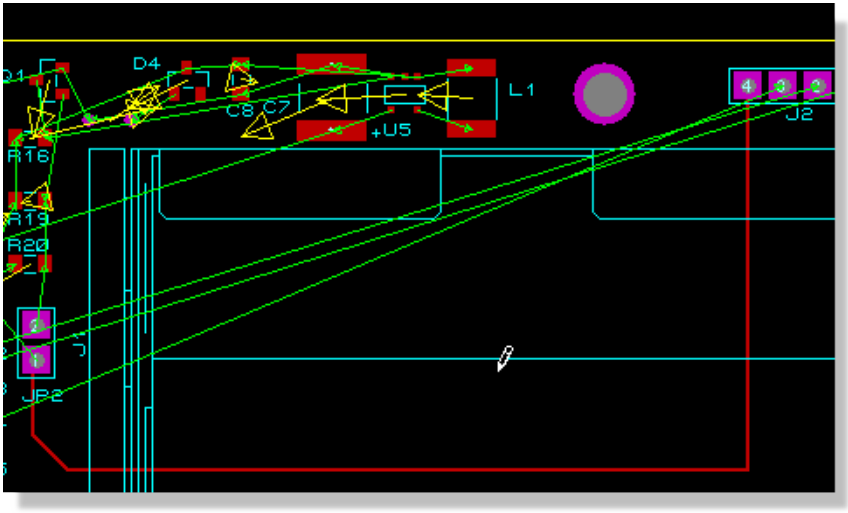
Next, right click on the highlighted segment and select the Delete Route command; this will delete only this segment of the route.



Deleting a trace via the context menu

Finally, zoom in, change the snap settings if need be and replace the route from the existing track to the terminating pad.

If things have gone more badly wrong you can simply delete the entire route and start again; what you are aiming for is something like the following screenshot:



Editing a Route

It is often the case that routes need to be nudged or moved into position after placement and we definitely don't want to be deleting and replacing routes or parts of routes all the time. To take an example, let's move our track lower down the board a little, towards the bottom board edge.

Start by right clicking on the horizontal section (segment) of the track; this is the section we want to move. Next, select Drag Route from the resulting context menu, move the mouse down to 'pull' the track into the desired location and then left click again to commit placement.

> Moving tracks works as we've seen on a track segment. Using the Trim Manually option on the context menu for a track allows you to define your own segment and you therefore have complete flexibility with altering track topology.

You may want to perform operations on multiple tracks at one time. The obvious example of this is changing the width of the track(s) and/or the via style used when routing them. You can do this by first switching to selection mode, then using CTRL+left click to select and finally by right clicking on any of the selected tracks to access the context menu.

Layer Pairs and Manual Routing

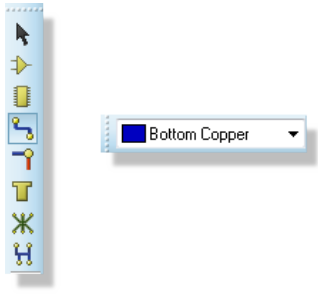
The track we have placed is of course only on top copper, whereas we often want to via up and down through the board when routing. ARES handles this with a concept called Layer Pairs. This means that every layer on the board has an associated layer so that via destinations are known during placement. For a two layer board this is obvious with top copper being associated

with bottom copper and vice versa but with multi-layer boards configuring the layer pairs (Technology Menu – Layer Pairs) can be an important step.

> Don't be surprised to see much of layer pairs dialogue disabled. The available layer pairs are restricted to the signal copper layers set up in your layer stack.

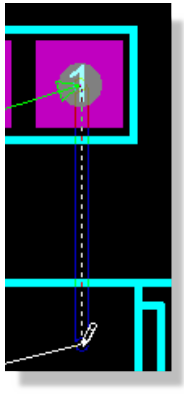
In our case, the default assignments are correct and no action is required. Let's place a couple more routes manually to see how this works. We'll look at connecting pins 1 and 2 of the J2 connector which are the transmission lines from the USART on the dsPIC processor.

We'll start with pin 1 and route from the pin on the bottom copper layer. Make sure that you have track mode selected and then hit the space bar on the keyboard, noticing that this will toggle the layer between the two layers associated as a layer pair. If you have ended up on top copper simply hit the space bar again to set the layer to bottom copper.



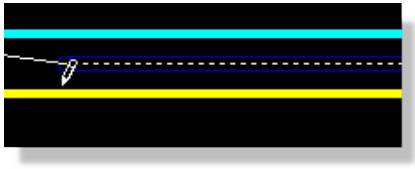
Selecting the Bottom Copper layer

The processor is some distance from the connector and the easiest path would seem to be down and along the bottom of the board. Begin placing the track by left clicking on pin 1 on the connector and then moving the mouse downwards



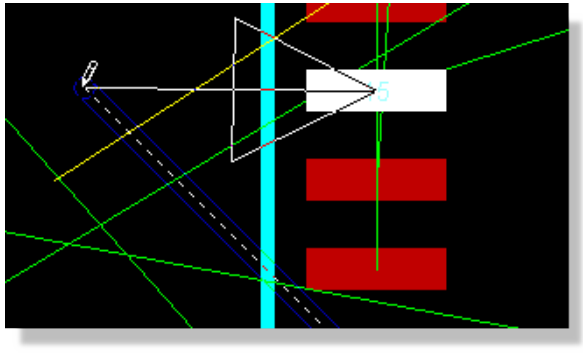
Routing on Bottom Copper

When we get near the bottom of the board, left click the mouse to place anchors and guide the mouse to the left. To maximise space on the board you can route across the bottom of the board with the mouse over the board edge graphic - this will hug the track to the absolute edge of the board leaving only the Edge Clearance between the two.



Tracks will hug the board edge

Once you get near the U1 IC bring the track up just past the right hand pads of the processor footprint.



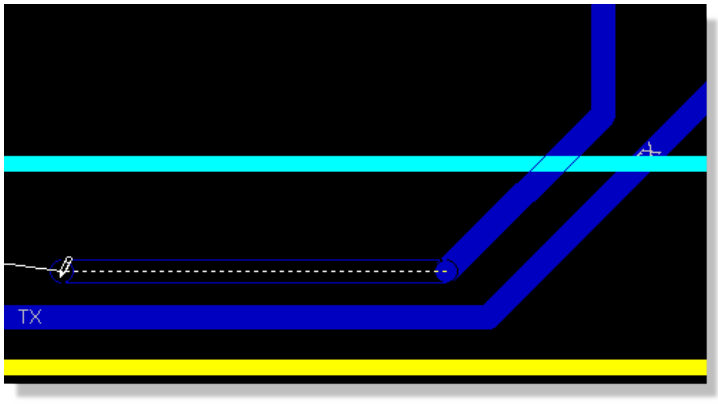
Routing to the left of the right hand pads of the Processor

We now need to place a via before we can connect to the SMT pad on top copper. As described above you can either double click the mouse at the point you wish to place the via or you can press the spacebar to float a via and then guide the placement point with the mouse before left clicking to commit. The latter method has the advantage of allowing you to butt the via as closely as possible against the pad thereby minimising the stub track length on top copper.

Placing a via by pressing the space bar

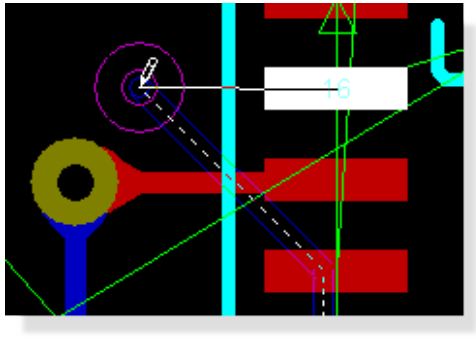
Quite a lot has been covered in making these connections and, unless you are fairly skillful, there has probably been some finger trouble along the way. We'll continue with the other connections from the J2 connector to help familiarize ourselves with the techniques we've introduced.

Pin 2 of the connector should follow an almost identical path to pin 1 and again we can start placement on bottom copper (check the layer selector before you start placement; the space bar will toggle the layers). We can easily hug the track we have just placed by moving the mouse parallel to the path we want to travel but over the original track as shown below.



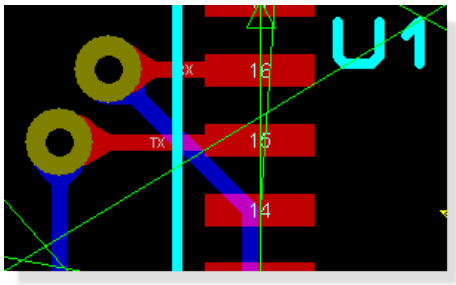
Tracks will also hug other tracks

In order to make the connection, we can then route vertically underneath the pads and make a short 45 degree track before double clicking to place the via and complete the connection.



Routing a 45 degree angle

Of course there are many ways we can route the board and personal preference also plays a part. Feel free to experiment with the other connection on the connector, working around the battery holder, using the ratsnest to find a legal destination and then completing the route. The following screenshot is an example of what you might have when you are finished.

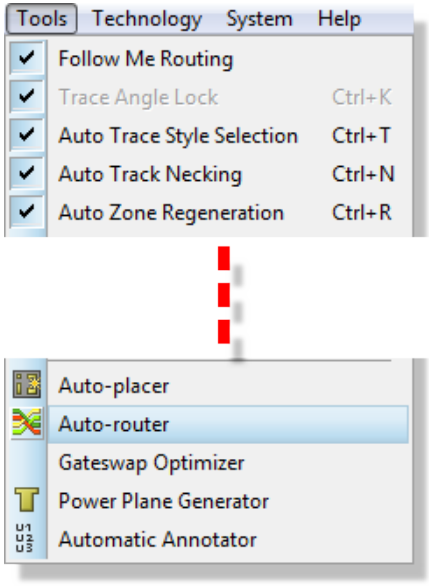


Both tracks fully routed

Basic Auto-Routing

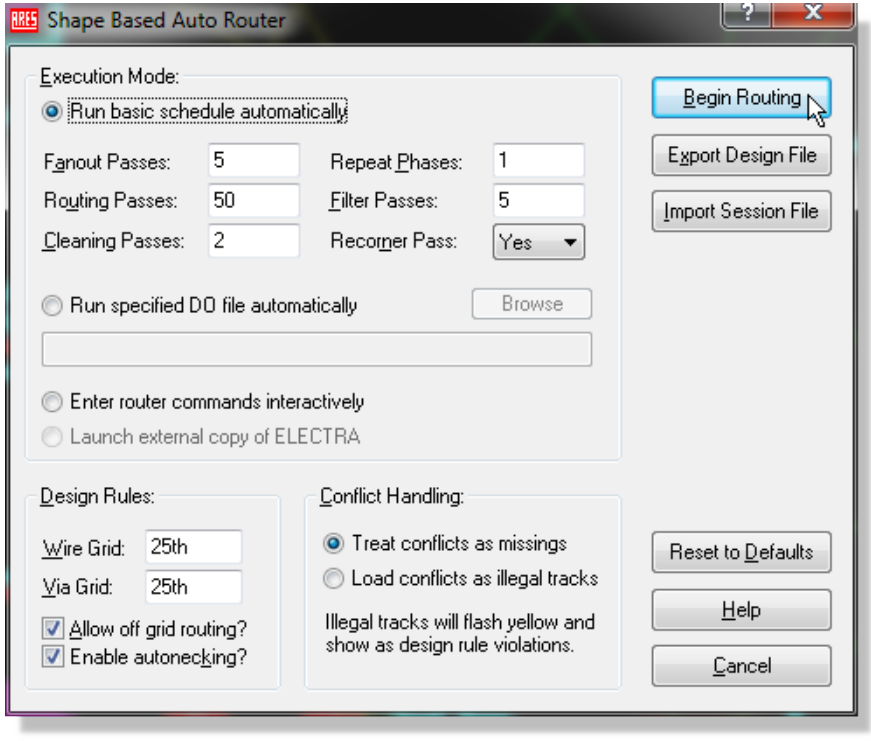
From this point on we'll use the auto-routing engine to complete placement of the routes. As with the manual tracking the auto-router will obey all the design rules that we configured earlier.

Start by invoking the auto-router from the Tools Menu in ARES or from the icon at the top of the application.



Select the Autorouter from the Tools menu

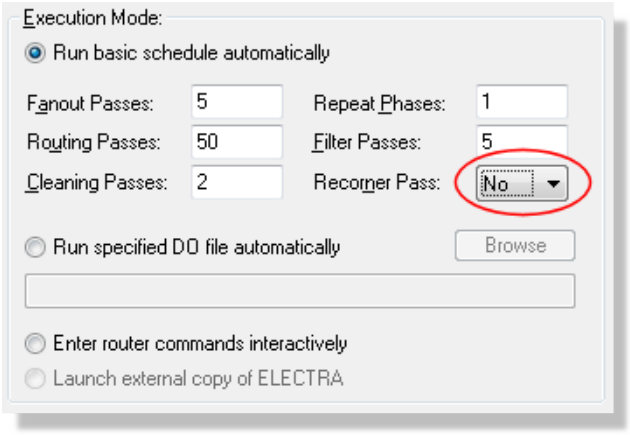
The resulting dialogue form is reasonably complex but all fields have context sensitive help associated with them. You can also find additional information about the various type of routing pass in the reference manual. For our purposes (and indeed for most boards of small/medium complexity) the defaults are more than adequate. We'll start by running in 'fully automated mode' which should be selected by default so simply hit the begin routing button to complete the remaining connections.



The Autorouter dialogue form

Everything will happen quite quickly from this point onwards but you should see routing progress on the status bar as the engine works towards completion. Once the board is complete there are two immediate points of note:

- The auto-router has preserved those tracks that we placed manually and has not ripped and replaced them whilst working on the rest of the board.
- Once the autorouter has completed a final pass is made to corner the tracks. If you prefer this not to happen you can remove the Recorner Pass option on the dialogue form before invoking the router

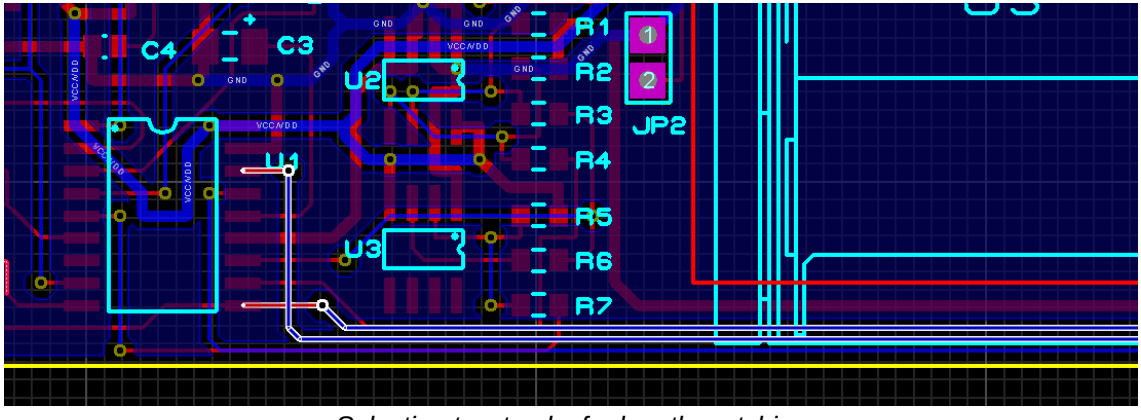


Set the Recorner Pass to be NO

Length Matching Routes

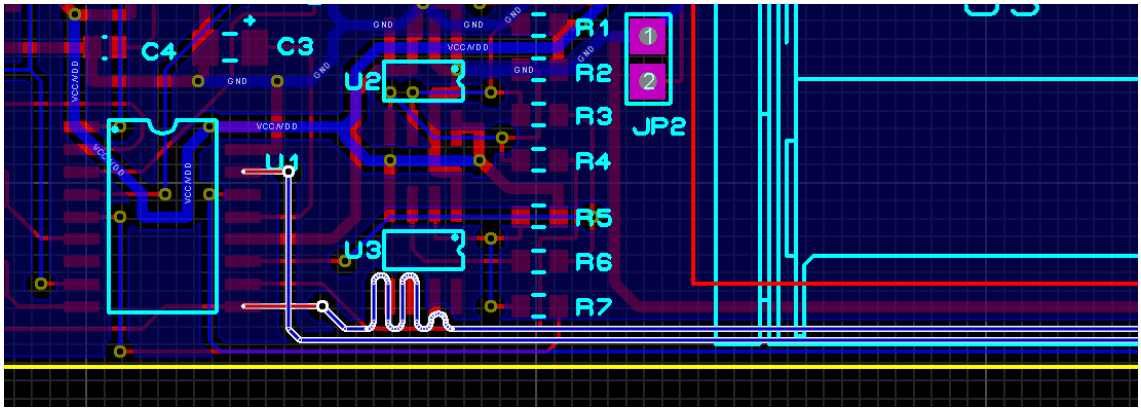
> Automatic length matching is part of the advanced feature set and requires PCB Design Level 2 or higher.

For high speed signal lines it is common to need to match the lengths of the tracks to ensure that all relevant signals arrive inside the sample and hold time of the receiving chip. While not required for the tutorial design we now have a fully routed board so it is a good opportunity to explain how the feature would be used. The first step is to select the tracks that you wish to length match by holding the CTRL button down and left clicking on each track. We've chosen to do this with two large routes into U1 as shown here:



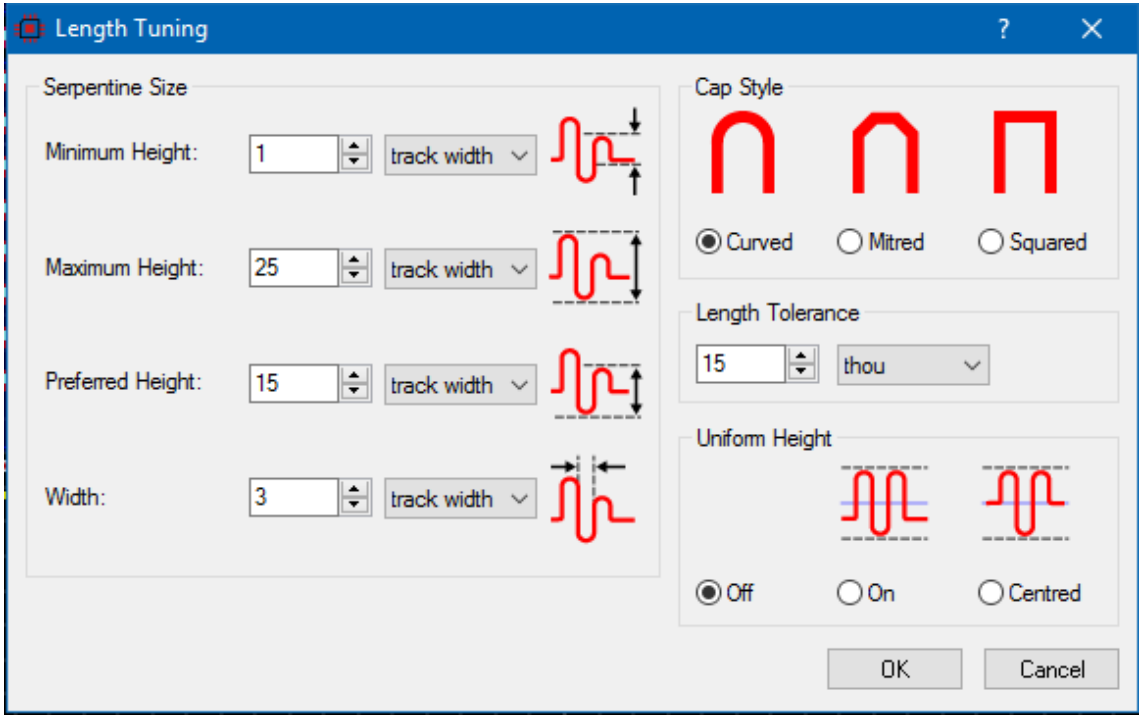
Selecting two tracks for length matching

Next, right click on one of the tracks and select the match route lengths command from the context menu. If successful you will see track segments added in a serpentine pattern to the shorter route in order to match its length with the longer route.



Matched lengths

Full configuration of the added length can be found on the Configure Length Tuning command on the Technology Menu or from the context menu.



Configuring Length Matching Options.

Other points to note about length matching.

- You can match multiple tracks at the same time (e.g a bus).
- The length of a track is shown on the status bar when it is selected.
- If a track is part of a group that has been length matched the target length is also shown in the status bar when you select the track.
- The pre-production check will test all matched tracks against their target length and warn you if their length has changed.
- If you choose routes with vias the length (depth) of the via is calculated from the thickness values you enter in the layer stackup.

The Selection Filter

Now that we have a completed board it's worth spending a little time looking at the techniques for selecting various object types on different layers.

ARES uses the Selection filter at the bottom left of the application window to determine which objects are available for selection at any given time.

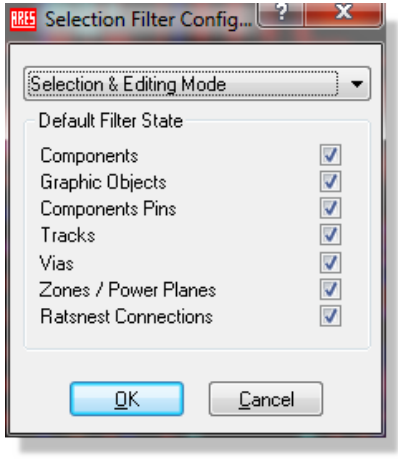
The left most button will determine whether the layer selector is active:

When this is toggled off the selection applies to all layers on the board.

When toggled on the selection applies only to the layer specified in the layer selector

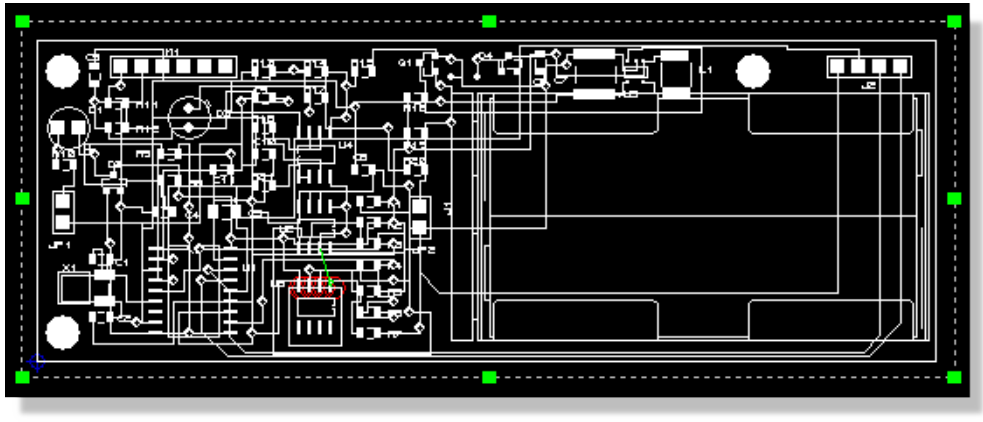
The other buttons represent different object types (tracks, components, graphics, etc.) and determine whether those object types are selectable. Hover the mouse over an icon for a tooltip description.

If you switch between the different modes of operation (for example from selection mode to track mode to component mode) you will see that the selectable object types will change according to the mode you are in. Whilst these are good defaults for normal operation you can change at any time simply by toggling on or off the layer switch or object type that you do or do not want to be selectable. If you find yourself changing selectable items regularly you can also change the defaults via the System Menu – Set Selection Filter command.



The Layer Selection options

Let's take a practical example and delete all the traces on the top of the board, except the one we have manually placed. Start by entering Selection Mode and left dragging a selection board around the entire board.

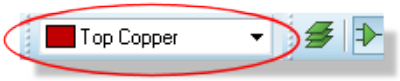


Select the entire board

Next, deselect those objects that we do not want to delete, namely everything apart from tracks and vias. The tagged items will update automatically to provide visual confirmation of what is selected.

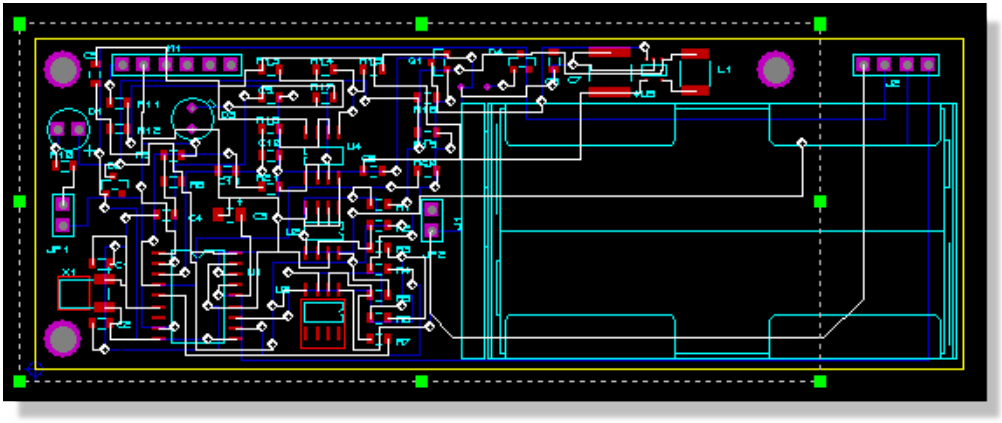
Only the tracks will be selected

Now change the layer to be Top Copper on the layer selector and toggle the layer button such that the selection filter applies only to the current layer



Top copper selected

Use the green drag handle at the right of the tagbox to move the selection filter in until it does not encompass the J2 connector.



Notice that Only the traces on Top Copper are Selected

Finally, use the icon at the far right of the selection filter to deselect items that are only partially inside the tagbox; in our case this will deselect the track we manually placed from J2, pin 4.

Tag box toggle

We can now simply hit the delete button on the keyboard or right click inside the tagbox and select Block Delete icon to remove all the top copper tracks.

> Bear in mind that the selection filter controls what objects are available for selection at any given time. If you ever find that you can't select an item the first thing to check is whether that object type is enabled for the mode that you are in. Alternatively, simply switch to Selection Mode where all items are selectable.

Advanced Auto-Routing

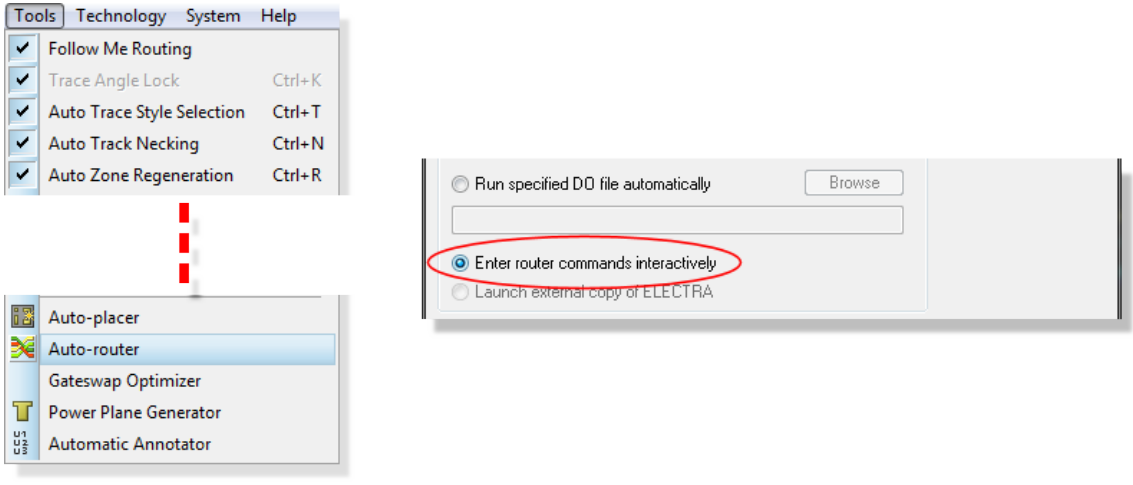
Given that we've ripped a lot of partials and removed vias as well we've actually made a bit of a mess here. Fortunately, we can fix the situation by simply re-invoking the auto-router and re-placing all our tracks. The router has its own clean up phase so it will tidy up all the redundant partial tracks for us as it completes the new routing.

If you have a standard features version of Proteus (PCB Starter Kit, Level 1 or Level 1+) simply run the autorouter as we discussed in the previous section to re-route the board.

For those using the Advanced Feature Set, we'll use this section to show in brief some of the additional features available. These fall into two main categories:

- The ability to route only a specified area or set of connections.
- The ability to control the routing script; i.e. to determine which routing commands are executed and in which order.

Start by invoking the autorouter dialogue form (Tools Menu), switching to Interactive mode and then selecting the Begin Routing button.



Entering the router interactively

A command window will open at the bottom of the Editing Window, which will allow us to direct the routing progress interactively. ARES provides a rich command set to control routing, including such things as bend radius for mitering tracks and fanout length and direction from SMT pads. This is fully documented in the reference manual so we will concentrate on some basic practical examples in this tutorial.

There are a couple of important points worth emphasizing at this stage:

Any commands entered will action on a set of tagged connections or on the whole board if nothing is highlighted.

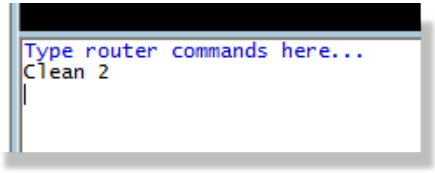
Changing the circuit or switching modes will automatically exit the routing interface.

Essentially, what this means is that we can dictate what connections are routed by controlling which items are highlighted.

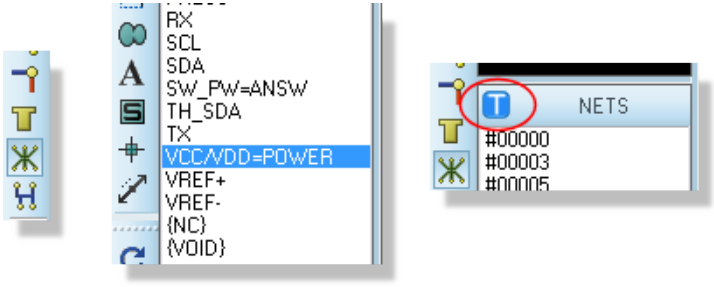
First of all let's clean up the loose tracking on bottom copper resulting from our previous rip of the top copper tracks. The basic syntax of most commands is:

```
<command> <number of passes>
```

so we can start by typing clean 2 to clean up the superfluous tracking.

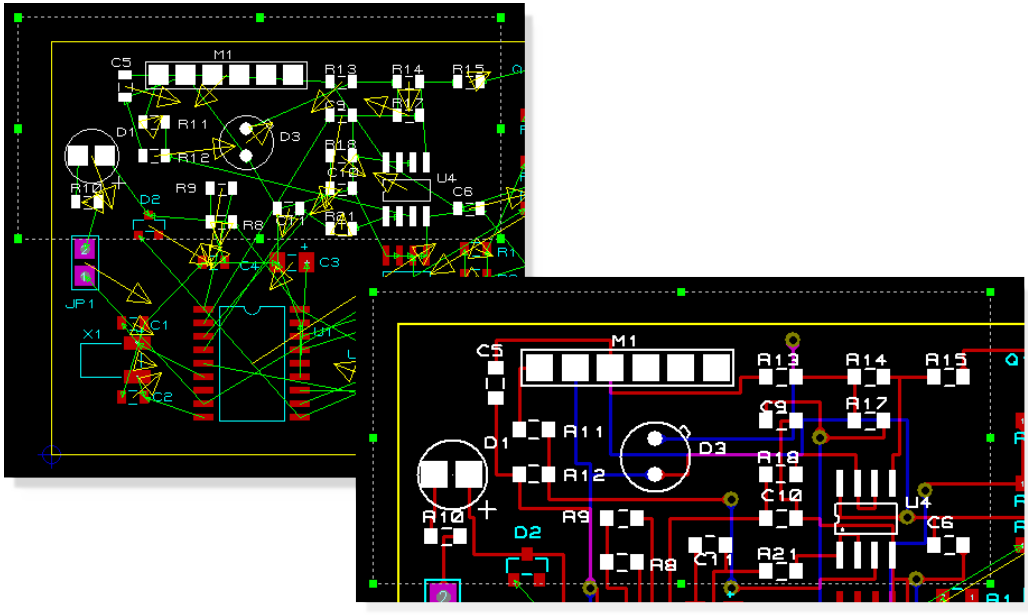


Next, let's assume that we want to route all the VCC connections. Select the VCC/VDD=POWER net in the Object Selector and click on the small 'T' button above the Object Selector to highlight all the connections on that net. Finally type in 'route 5' to route these connections.



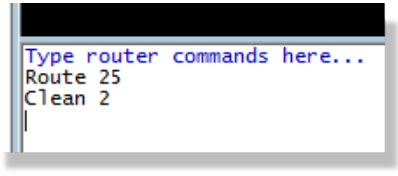
- Clicking anywhere in the Editing Window will deselect the currently tagged items. When the focus is in the Editing Window you can also use the middle mouse (or keyboard shortcuts) to zoom in and out in the normal way.

Similarly, we can highlight connections on an area of the board and route them independent of the rest of the board. For example, right depress the mouse and drag a selection box around the left half of the board and then type 'route 10' to complete open connections in that area.

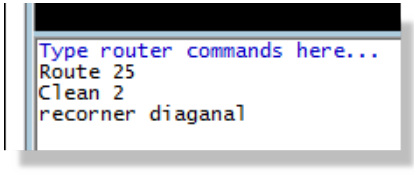


Autorouting an area of the board

You may notice that this has left a couple of ratsnest lines or has not completed all of the connections. We could try to resolve this by issuing clean, filter and more route commands, but for our purposes it's easier to simply revert back to a 'whole-board' scenario and then completing routing. You can do this by left clicking on an empty area of the Editing Window to clear the selection. Typing route 25, followed by clean 2 should see the board completed



Finally, we can reduce trace length by typing recorner diagonal as a finishing touch.



Do note that there is much more flexibility in the command set than that shown in this tutorial, both in the number of commands and in the parameters that control the command actions.

- You can exit auto-routing at any time via the ESCAPE button on your keyboard.

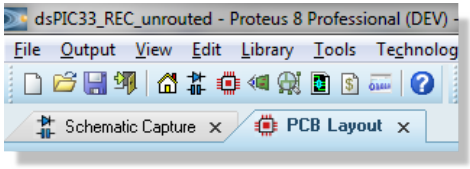
Netlists and Design Changes

We have now placed and routed our layout based on the schematic we designed in the ISIS tutorial. In practice however, it is very unlikely that we would not be making design changes during the PCB process so we'll devote a little time to examining a typical workflow.

Clearly, we will be editing both the schematic and the layout modules so both need to be open. If the schematic tab is closed you can open it from the application module toolbar at the top of the Proteus application.

Isis icon on the Application toolbar

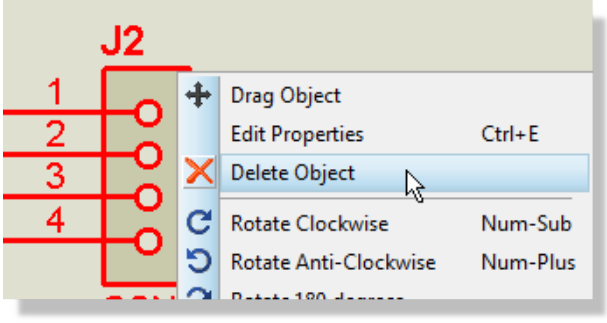
Since we will be working with both modules you may want to split them into two frames if you have two monitors. To do this simply drag one of the tabs onto the other monitor. For this documentation, we will assume that ISIS and ARES are open as tabs inside a single Proteus frame.



Multiple tabs in one frame

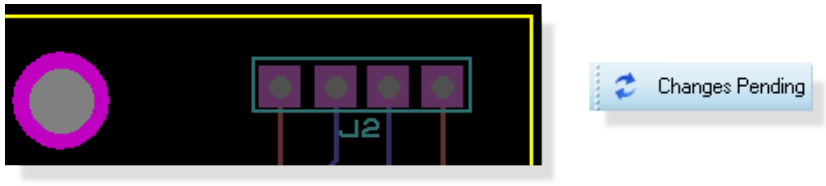
Live Netlisting

First, let's delete the J2 connector on the schematic. Switch to the ISIS tab, right click and select delete object from the resulting context menu.



Deleting a component in ISIS

Now switch back to the PCB tab. Note that the J2 part has been dimmed to show that it is not present on the schematic and that the connectivity status shows that we have pending changes (we are out of sync).



Netlist is out of sync

Since the action taken in ISIS removes from the board it requires manual confirmation. We can now either hit undo (CTRL+Z) if we don't like the change or we can click on 'Changes Pending' in the connectivity status to accept the change. When we do this, the 'shadow' part will disappear.

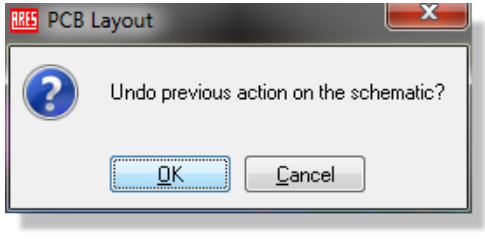
Clicking on Changes Pending will remove J2 and correct the netlist

Note that tracks that were connected will also be deleted. If you want instead to delete only the footprint (leaving the connected tracks in place as stubs) then you can do so by directly deleting the package in ARES.

The undo system is still available to revert the changes. Clicking on the UNDO icon (or CTRL+Z) once will bring back the shadow objects and repeating the process a second time will bring back the schematic part and the real PCB footprint.

Undo and Redo icons

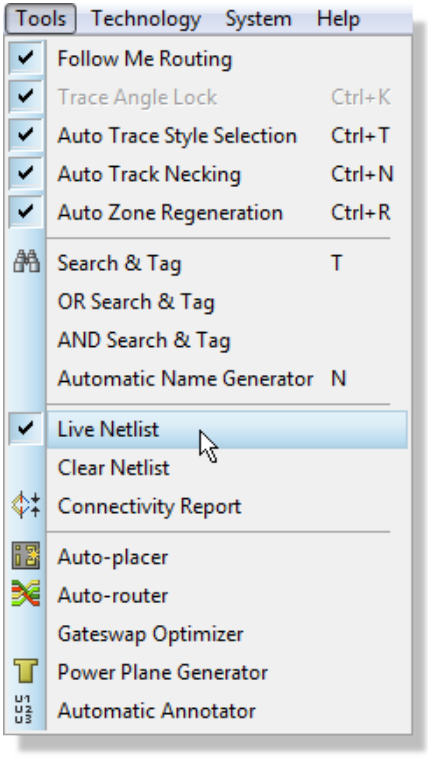
If you click on undo while in the ARES tab but the undo action applies to the schematic you will be prompted to confirm the operation. In our case, we do want to bring back the connector so accept the prompt to return the PCB to the original state.



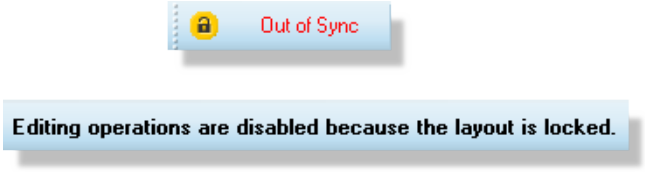
This is a simple example of a change which affects objects already placed on the layout. The important thing to remember is that while the netlist is live, you must manually confirm any change which affects the connectivity of objects you have already placed.

Batch Netlisting

If you prefer, live netlisting can be turned off and a more traditional batch mode netlist used. This is controlled from the 'live netlist' toggle on the Tools menu (in the PCB module).



Having turned live netlisting off, let's repeat the same procedure to see the difference. Switch to the schematic tab, delete the connector and then switch back to the PCB tab. This time the connector on the PCB remains present and the connectivity status reports that the layout is locked and out of sync.



Status bar messages when 'out of sync'

This means that we have not loaded the changed netlist and cannot work on the layout until we bring the netlist across. Proteus works with a single database of parts that include schematic components and pcb footprints so - while you can work on the schematic - you cannot work on the PCB until you sync it with the schematic. Click on the connectivity status to unlock the PCB

and bring the current netlist across (this is exactly the same as the 'Netlist to ARES' command in previous versions).

This brings us to the same state as we arrived at automatically with live netlisting on. We have a shadow part in ARES and the connectivity status reports changes pending.

The main difference between live netlisting and batch netlisting is only that you must confirm each netlist load manually in the latter mode. We recommend that live netlisting is used for normal working, although for particularly large designs batch mode may provide a useful tool.

> Some functionality will require a live netlist and so if you are working in batch mode you may find menu commands disabled until such time as you bring the netlist back in and sync.

Annotation

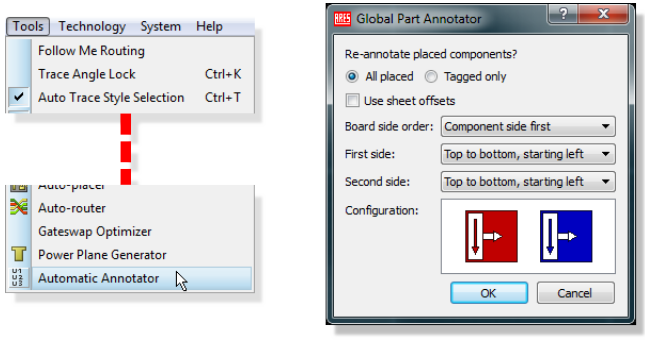
Annotation changes from schematic to layout or from layout to schematic are also a common editing operation when both modules are in use.

Schematic to PCB

Annotation changes made in the schematic will automatically be reflected in the PCB when live netlisting is on and will require unlocking when live netlisting is off. The Global Annotator on the Tools Menu on the schematic can automate this task.

PCB to Schematic

You can re-annotate the board either manually (editing the part reference labels) or globally from the automatic annotator command on the ARES tools menu.



Launching the Automatic Annotator

When live netlisting is on, this will automatically reflect to the schematic. With live netlisting off, the schematic will update and the layout will then lock as the netlist has changed and needs to be manually passed through to the layout.

General Netlisting Rules

There are several important rules of thumb to understand about netlisting in Proteus.

When live netlisting is on, a change that adds connectivity will happen automatically (e.g. adding a wire on the schematic will add a ratsnest on the layout). If you make a change that removes connectivity the layout will show the change by dimming the deleted objects and you will need to manual confirm the operation.

When live netlisting is off (batch netlisting) any and all changes to the netlist will require you to unlock and apply the netlist before you can continue working on the layout.

Clicking on changes pending to remove footprints that have been deleted on the schematic will also remove associated tracks. If you want to remove footprint only and leave the tracks simply delete the part in ARES (rather than ISIS).

Dimmed objects in ARES are visual placeholders only - they do not exist on the board. Best practice is always to click on changes pending on the connectivity status and to arrange the layout to conform to the schematic.

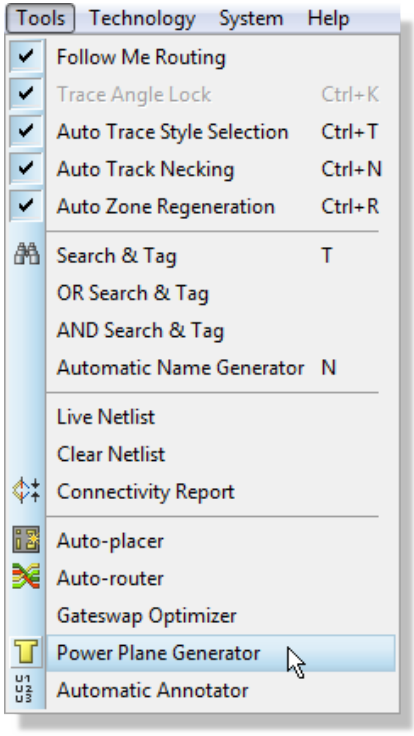
Imported projects (from older versions) will always start with live netlisting off as there is no way to tell whether they are in sync. By contrast, new projects will always start with live netlisting on. In either case you can switch according to preference from the live netlisting toggle on the Tools menu in the PCB layout module.

Power Planes and Slots

If you switch back to the layout tab we will continue the layout phase of the project. To minimize track impedance we are going to place a power plane covering the entire board.

Placing Power Planes

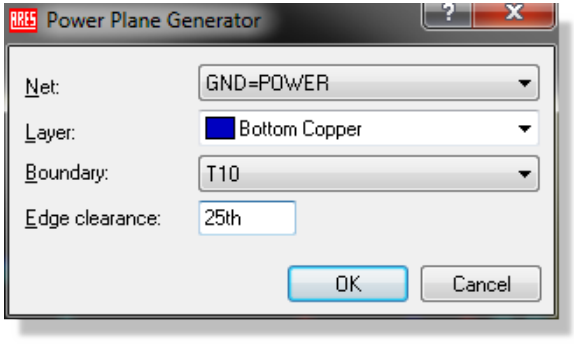
This is actually the easiest type of power plane to place and is available in all levels of the professional software. Start by invoking the power plane generator command from the Tools Menu



Opening the Power Plane Generator from the Tools menu

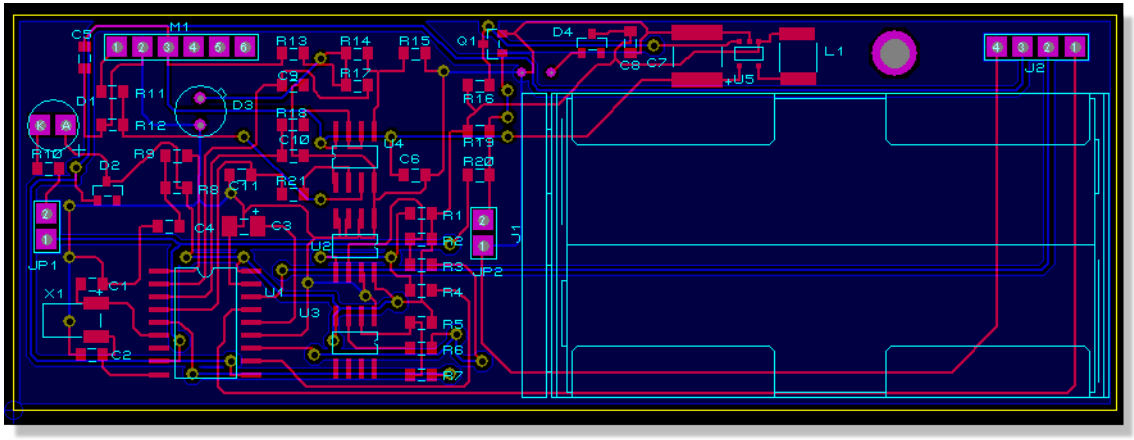
From the resulting dialogue form select the GND=POWER net, keeping the layer as bottom copper and setting the boundary style to be T10. This is the trace style in which the inner and outer boundaries of the zone are drawn and also determines the thinnest section of copper by which the power plane can make a connection. Setting this larger will prevent the copper flowing through small gaps (e.g. between pins) but making it smaller means that connectivity may be made only by thin sections of copper.

We can leave the clearance between the power plane and the board edge at the default value.



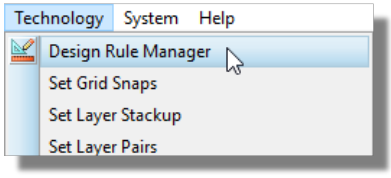
Power Plane Generator Dialogue form

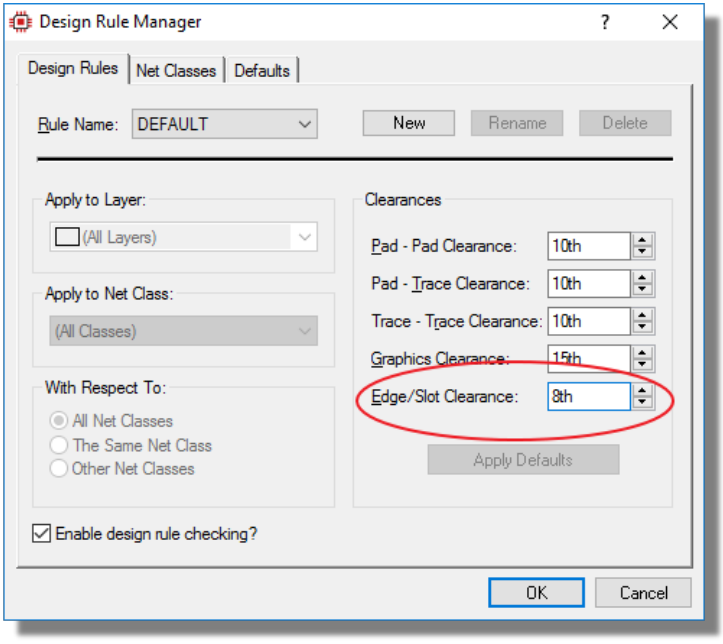
After you exit the dialogue form you should see that the power plane is generated across the entire board.



Power Plane has flooded the entire layer

The clearance from the zone to the board edge is defined by the Edge/Slot design rule. We can change this via the Design Rule Manager on the Technology and the zone will re-generate to the new clearance when we have finished.





- Since you can set a new design rule per layer it is possible to have different clearance rules for power planes on different layers

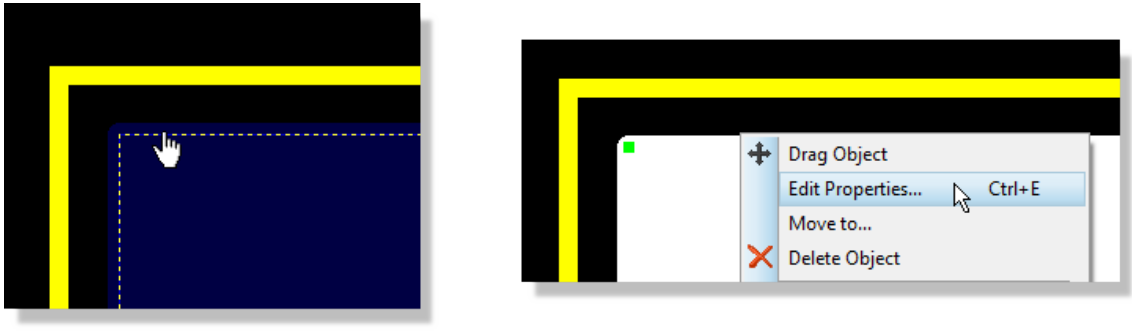
Nesting and Islands

There are several additional configuration options available to us now that we have placed the zone. Editing a zone is slightly different from most other objects in ARES in that you must right click on the border of the zone (to avoid continuous unwanted selections).

Firstly, check the selection filter and make sure that the zone object type is selectable (or switch to selection mode).

Zone icon in the Selection Filter

Next, zoom in and move the mouse over the edge of the zone, right clicking the mouse when the zone is shown as active under the mouse. Select the Edit Properties from the resulting context menu.



Right click on the edge of a zone to edit its options

The main options of interest are towards the bottom of the dialogue form and are explained below:

Relieve Pins

When checked pins on the same net as the zone will have thermal relief applied to them; the thickness of the thermal relief is determined by the Relief field of the dialogue form.

- The software will prevent you from using a relief track style that is larger than the boundary style thickness; this is to protect against the reliefs 'sticking out' of the boundary.

- The topology of the relief stems on a particular pad can be changed to a diagonal 'X' by editing the pad itself after placement. This is sometimes useful to maximise contact with a zone.

This option should almost always be left in its default checked state.

Exclude Tracking

If this option is checked, the zone will treat tracks on its own net as obstacles. Otherwise the zone flows over such tracking, effectively ignoring it. Tracks on other nets, or loose pieces of track on no net are always treated as obstacles.

This option is normally left unchecked.

Suppress Islands

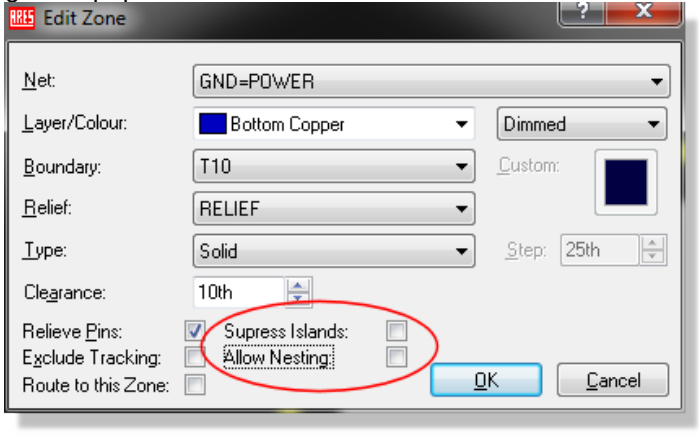
An island is defined in ARES as a block of copper or zone area in which no valid connections can be made. When checked, ARES will remove all such blocks from the board, leaving only areas of copper with connections.

Allow Nesting

It is quite common, particularly with larger boards, that the flow of a zone will encounter an obstacle through which it cannot pass. When checked, the zone will jump over the obstacle and continue flowing across the board.

This is a very useful option for example on busy boards or where you want to connect a pad where the topology of the board makes it impossible for a single zone flow to get through.

Given our relatively simple board, the default options are ideal for our purposes. Selecting the nesting option will have no effect for example as there are no inner boundaries on the board. However, if you want to experiment you should find that if you uncheck ‘suppress islands’ option, the resulting zone populates far more of the board that in our initial configuration.

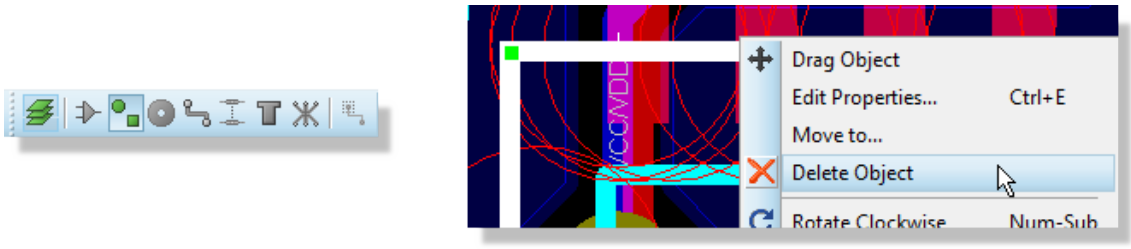


We have covered only the basics of power plane functionality here relevant to our current design. For more information, including split planes, zone keepouts, stitching and bridge tracking please see the power planes section of the reference manual.

Slots

To complete the layout of the board we need to return to our temperature/humidity sensor (U3). In order to make accurate measurements, we want a thermal cutout around this part, ensuring that the temperature we are measuring is actually the environmental temperature and not conducted heat from the PCB.

You may remember that we placed a keepout area around this part to prevent the autorouter from placing tracks through the area we want to cut out. Start by zooming in around U3 and removing the keepout. You'll need to either switch to Selection Mode or to change the selection filter such that you can select 2D Graphics

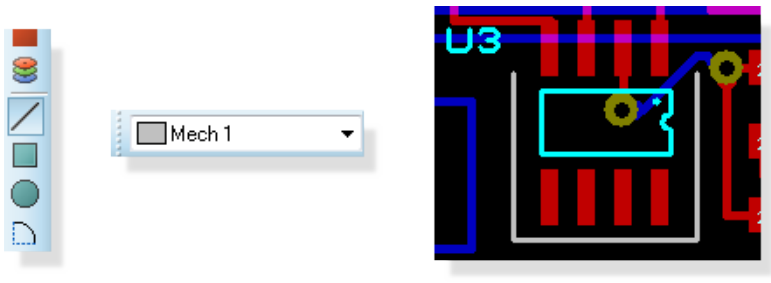


Deleting a Keepout area

While we are here, we might as well move across and repeat the process by deleting with the other keepout around the crystal.

Returning to U3, we now need to outline the cutout region. In ARES this is a two stage process; we need to place graphics appropriately on a mechanical layer and then designate the mechanical layer as the slotting layer when we generate output for manufacture.

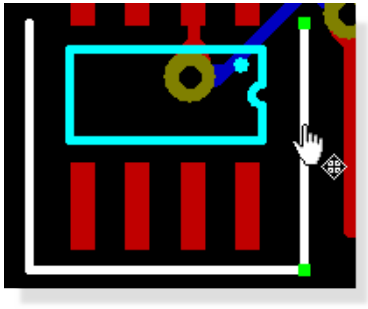
Select the 2D Graphics Line icon, change the layer selector to be MECH1 and then place three lines to form a 'U' shaped cutout.



Placing a slot on MECH 1

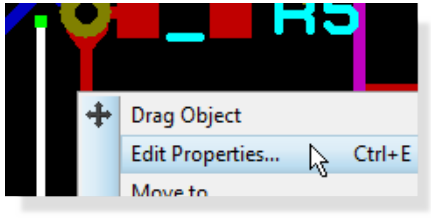
Finally, we can thicken these three lines to something more appropriate as follows:

Enter selection mode, hold the CTRL button down on the keypad and left click once on each line. This will select all three lines

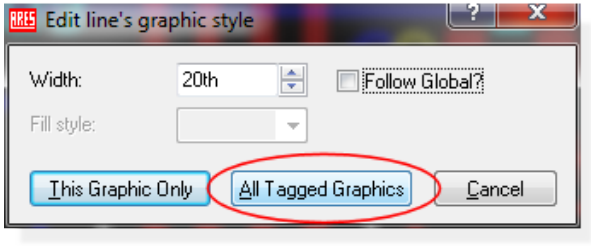


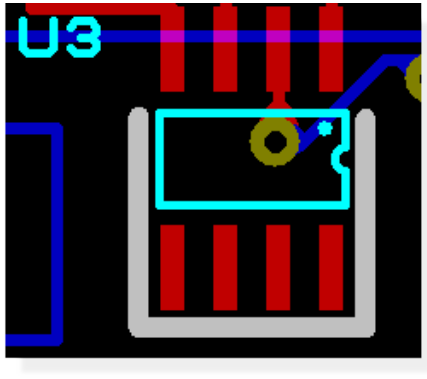
Selected items are white

Right click on any line and select Edit Properties from the resulting context menu.



Uncheck the Follow Global checkbox, change the width to be 20th and then apply to all tagged graphics.





Changing multiple objects at once

- We will cover how to specify MECH1 as the slotting later in the documentation in the section on manufacturing outputs.

3D Visualisation

Now that the board is now routed and ready for production we first want to examine it in 3D in order that we can properly preview how it will look in real life and possibly make final design alterations prior to prototyping. Start by invoking the 3D Visualisation Engine from the module toolbar at the top of the application

3D Viewer icon

This will open and load the 3D view of the board in a separate tab. As with any tab, you can drag onto a separate frame if you want to look at both the 2D layout and the 3D view simultaneously.

Basic Navigation

The first thing we can do is view the board from different preset angles. Five preset views are supplied: top view, front view, back view, left view and right view and these are accessible via any of the following methods:

- Menu options on the View menu in the 3D Viewer
- From the navigation toolbar at the bottom of the 3D Viewer
- From keyboard shortcuts F8 through F12 whilst in the 3D Viewer.

Now that we can look at the board from a number of angles the next thing is to be able to look at it at a specific zoom level. Again, there are numerous ways to zoom in and out of the board:

- Roll the middle mouse wheel in and out (recommended)
- Menu options on the View menu
- From the icons on the Navigation Menu
- From keyboard shortcuts F6 (zoom in) and F7 (zoom out)

Experiment now with custom zoom levels and different preset views – whilst it is pretty subjective we envisage most users changing views via the navigation toolbar or keyboard shortcuts and using the middle mouse wheel to zoom in and out.

Bareboard View and Height Clearances

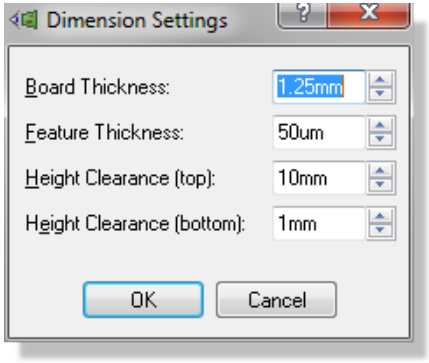
For inspection of resist and hole depth it is often useful to view the board without components. Selecting the bareboard view will remove all the physical components from the board.

Show Components icon

By contrast, if you need to fit the PCB into a chassis and wish to check height clearances you can enable the height boundary box via the icon at the bottom of the display.

Casing icon

Specification of the height takes place from the Dimensions command on the Settings menu of the viewer.



Board properties dialogue form

Custom Views

The next logical step is to be able to customise the view. This works conceptually by ‘attaching’ the mouse to the camera such that as you move the mouse the camera moves to the area of the board that you are interested in. You can invoke the Navigation mode either from the View menu, the crosshairs icon on the Navigation toolbar or simply by clicking the left mouse button.

3D Mouse cursor

You will know as soon as you are in Navigation mode as a crosshair cursor will appear over the mouse and your view of the board will change as you move the mouse. Using this together with the middle mouse wheel zoom will allow you to both ‘fly pass’ the board and to easily zoom in to closely examine a particular area of the board. Exiting navigation mode is as simple as right clicking the mouse.

For example, if we start in Front View (use the F9 keyboard shortcut) and we want to examine the resistors on the right we might proceed as follows:

1) Left click the mouse to enter navigation mode.
2) Move the mouse over the resistors.
3) Roll the middle mouse button to zoom in as required.
4) Right click the mouse to exit navigation mode.

The final necessary piece to completely customize the view is to allow users to spin or 'orbit' the board. This is done in navigation mode by holding down the left mouse button and moving the mouse. Essentially this will spin the board as you move the mouse – when you release the mouse button the camera will follow the mouse around the current view of the board as normal. Try this now, experimenting with different views of components on the board.

Remember that, if you are struggling to get the view you want you can use the keyboard shortcuts or navigation toolbar to return to one of the preset views. You should find however, that with only a little practice you become quite proficient at navigation.

To summarise:

- Left click enters navigation mode.
- Camera follows mouse around the board in navigation mode.
- Using the middle mouse wheel (or shortcut keys) allows you to zoom as you move the camera.
- Left depressing the mouse in navigation mode allows you to spin/orbit the entire board.
- Right click of the mouse exits navigation mode.

3D Models (STEP/IGES)

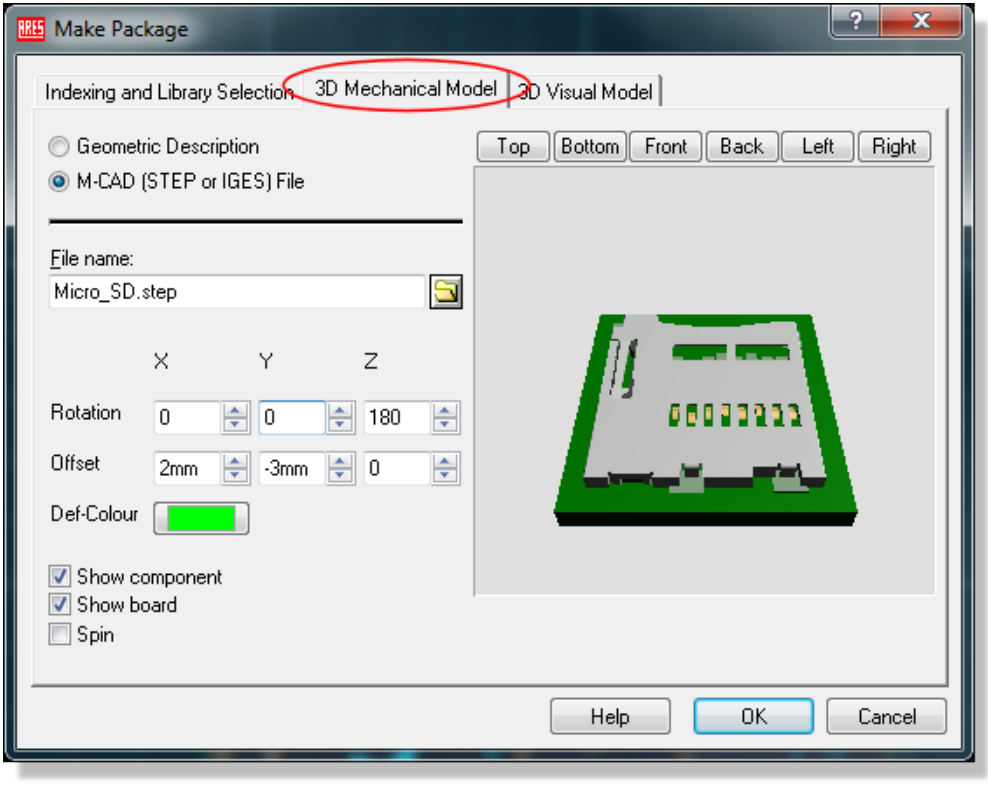
Each footprint on the layout can also have an associated 3D model. Normally, these are either STEP Files (for large or complex parts) or scripted geometric models (for simple passives).

In the case of STEP or IGES files, the procedure is:

1) Download the file from the internet or otherwise source the 3D model. There are many sites available, with www.3dcontentcentral.com and www.tracepartsonline.net being amongst the most popular.

2) Save the file into the MCAD directory of your Proteus installation. By default, this is:
`C:\ProgramData\Labcenter Electronics\Proteus 8 Professional\MCAD\`

3) Right click on the footprint in ARES and select Make Package, and then click on the 3D Mechanical Models tab.



4) Type the name of the STEP file and then the rotation and offset required to align the STEP file with the pads on the board.

For geometric models, the difference is basically that you describe the part in terms of size and shape via a simple scripting language. This is covered in the reference manual.

Live Update

Like any other module in Proteus, the 3D Viewer will update live as changes are made to the ARES PCB layout module. In practice, the usefulness of this will depend very much on the power of your machine (number of cores, memory, etc.) and the complexity of the board.

On a board of modest complexity and a modern computer the redraw time is not significant and - particularly on split frames - is very useful for movement and positioning.

- More information on 3D Visualization including creation your own 3D models, exporting 3D STEP Files for MCAD import, customizations and applying 3D data to legacy designs can be found in the 3D Viewer section in the online reference manual.

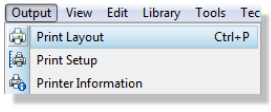
Board Output Options

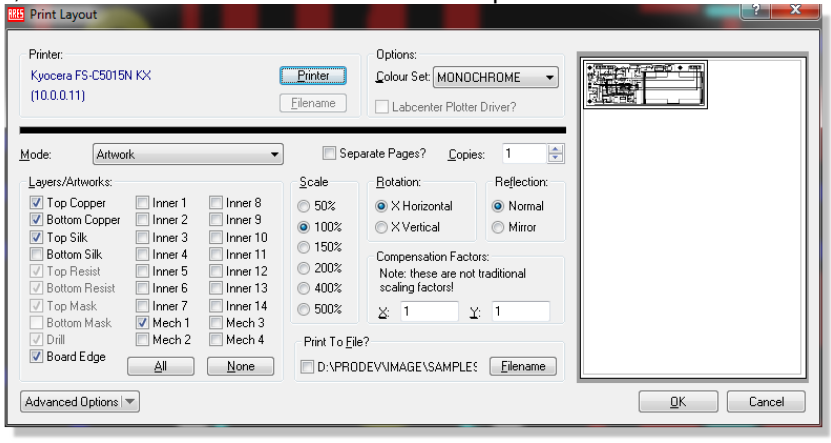
Last, but by no means least, we come to the crucial business of reproducing the pretty on screen graphics on paper or film. Under Windows, most hard copy devices are supported through the normal Windows printer drivers. Additionally, we supply our own drivers for pen-plotters, Gerber photoplotters and Excellon NC drill machines.

Printing

We will deal here firstly with printing to an ordinary Windows printer device – it is unlikely that you will have a photoplotter to hand! The first step is to select the correct device to print to using the Printer Setup command on the Output menu. This activates the Windows common dialogue for printer device selection and configuration. The details are thus dependent on your particular version of Windows and your printer driver - consult Windows and printer driver documentation for details.

Then, with a layout loaded, invoke the Print command from the Output menu.





The dialogue forms offer a number of controls, all of which have context sensitive help associated with them (context sensitive help on dialogue forms is accessed with a ‘point and shoot’ mechanism via the ‘?’ key at the top right of the dialogue form). The default settings should do for getting something and you commence output generation by clicking on OK. Output can be aborted by pressing ESC, although there may be a short delay before everything stops whilst ARES and your printer/plotter empty their buffers.

With plotters in particular, you will probably need to experiment with pens, paper, and the various settings on the Set Devices dialogue form in order to get optimum results. Full details may be found under the chapter Hard Copy Generation in the reference manual.

> ARES will remember your printer settings from the Printer Setup dialogue and maintain them independently of your printer settings for other applications. This means that you can configure a default set of printer options solely for use with the ARES application.

Output for Manufacture

ARES provides two main output options for board manufacture:

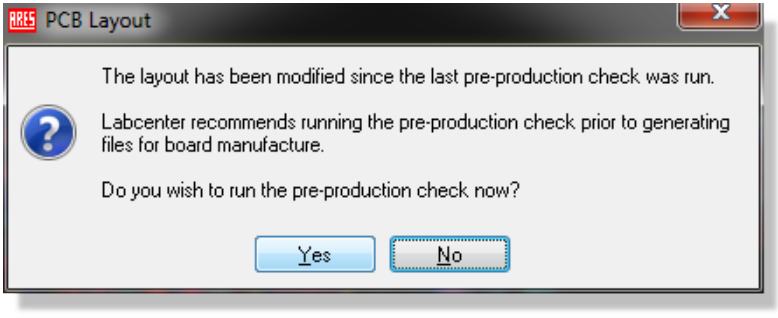
- Traditional Gerber/Excellon (available in all professional versions).
- ODB++ Manufacturing Output (available in Advanced Feature Set only).

From a user interface perspective both options are very similar but from a manufacturing perspective, the ODB++ option provides far more information than the older Gerber formats. Examples include:

- Inclusion of the connectivity information (the netlist) with the output fileset.
- Explicit support for plated/unplated specification on pads.
- Explicit support for fiducials.

All of this means that when viewing the output fileset from ODB++ the verification process is simpler and more complete. However, traditional Gerber/Excellon output is still prevalent in manufacture and would be sufficient for most purposes.

Regardless of your option the basic procedure is the same. When you invoke either of the output options you will most likely be prompted to run a pre-production check. This runs an automatic checklist to test for some common design errors and will report either a pass or a fail.

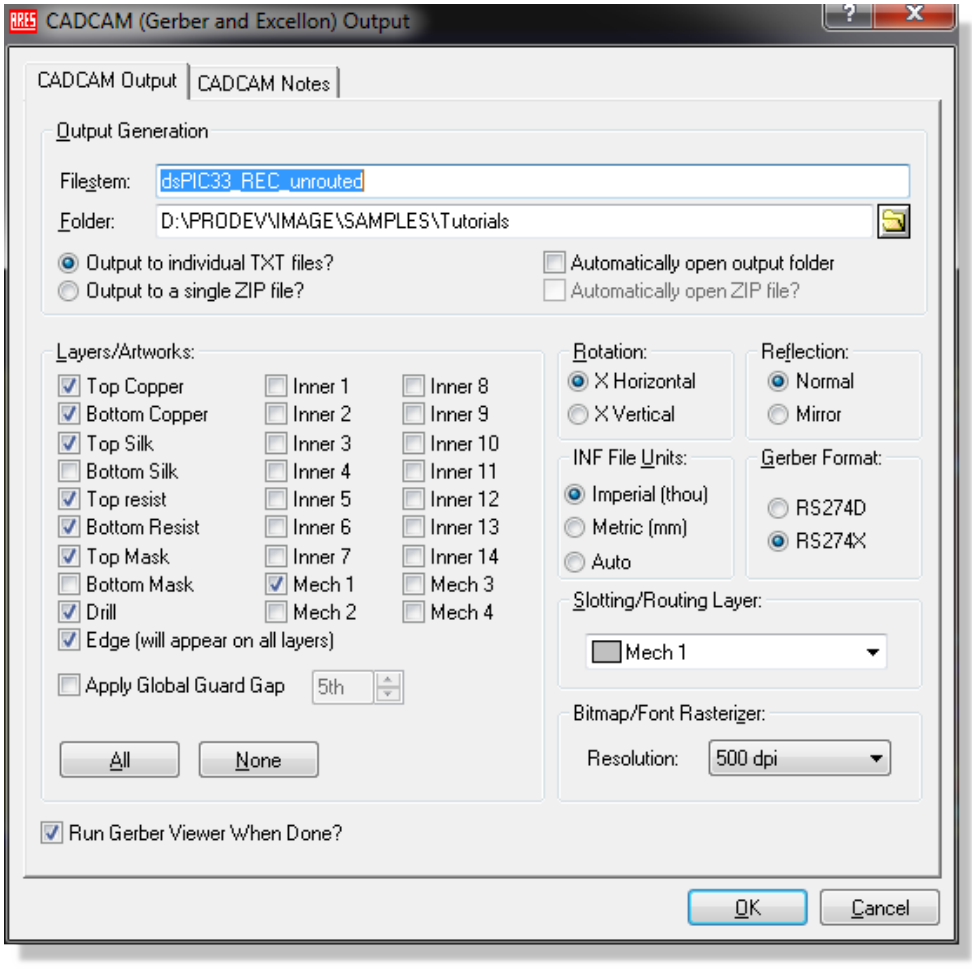


The Pre-production check (PPC) prompt

If your pre-production check reports errors we strongly recommend that you resolve them before proceeding to the manufacturing output dialogue. Please also note that pre-production check is an aid to the designer in quality assurance but not a guarantee; manual inspection of the layout is recommended and a prototype should always be made and tested before mass production.

More information on the Pre-production Check can be found in the CADCAM Output section of the online reference manual (Help Menu).

Assuming a pass you will then be provided with the Manufacturing Output dialogue form.



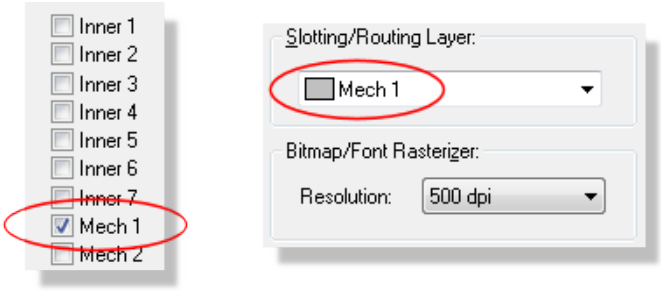
The Gerber / Excellon Output dialogue form

The options in the top section of the dialogue form are self-explanatory but there are some points of note regarding the rest of the configuration options.

The software attempts to populate the layer set with those layers used on the layout but this should always be verified. One of the most common problems with manufacture occurs when the manufacturer is not supplied with the full board information.

The Apply Global Guard Gap option will set the expansion of the resist plot around pads and vias to the distance specified when this option is checked. This will take effect on all pads and vias except those which have been manually altered on the layout. Some board manufacturers prefer to create the resist plot in house and this option could then be used to remove any resist expansion pre-manufacture. Unless otherwise directed we recommend that you leave this option unchecked.

The Slotting /Routing Layer option specifies explicitly which layer of the board is to be used for defining the routing strokes for cutouts and slots. In our case, we have used MECH1 and we must therefore set this via the selector.



Select the required slotting layer

The Bitmap/Font Rasterizer option controls the thickness of trace used to render bitmaps and, more importantly, power planes. The higher the resolution the better tonality of the resulting bitmaps but the larger the files. It is also possible that some manufacturers have a minimum width requirement and in such cases it may be necessary to reduce the DPI settings to conform with this. Generally speaking however, the default settings are fine.

The option at the bottom left allows you to automatically load your output into either the Labcenter Gerber Viewer (CADCAM Output) or the Valor ODB++ Viewer (ODB++ Output). This is useful if you want to verify the output fileset before passing to your manufacturing house.

> If you intend to panelize the board you should use the Gerber X2 output in the CADCAM Output command, then **close the project**, open the gerber module and import your CADCAM files, selecting Panelization Mode from the resulting dialogue form. **Panelization is only available with Gerber X2 output** - you cannot create panels from RS274X Gerber.

Finally, the Notes tab allows us to insert any relevant information or special considerations for the manufacturer. This is very important when we have specified a slotting layer as there is no standard way to pass this information through. A simple note that the information on MECH1 is for slotting will suffice.

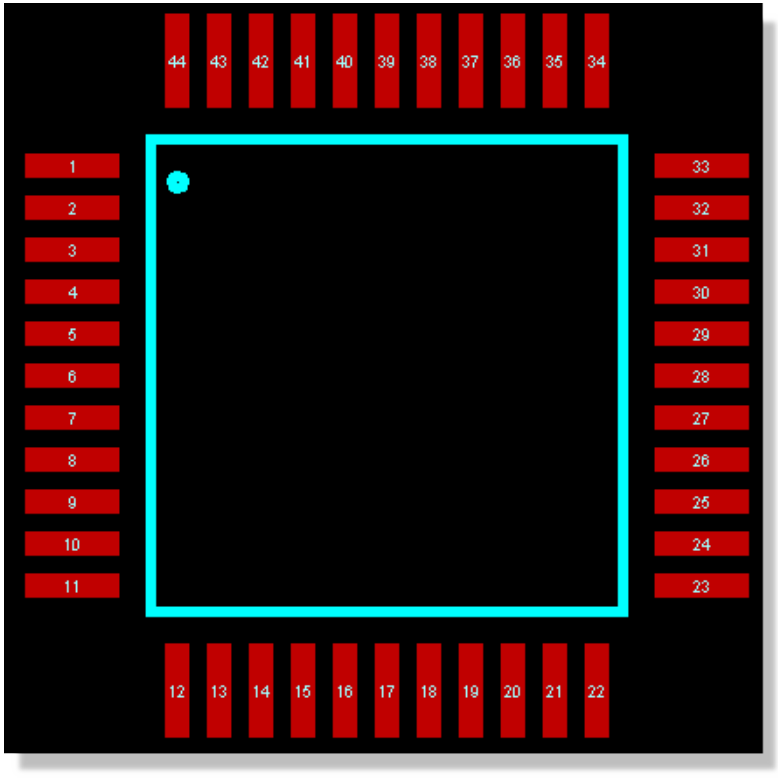
Having configured the necessary option we can generate the files and send for manufacture.

APPENDIX: Creating New Packages

ARES comes pre-supplied with a large quantity of footprints, and we have seen previously how to select and place these parts on to a layout. However, it may be necessary at times to create your own custom footprints or symbols – also a simple task with ARES – and this process is detailed below.

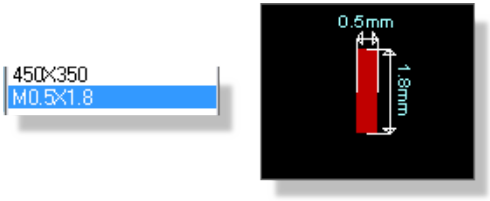
Drawing the Footprint

In our example, we will create a SQFP44 footprint with 0.8mm pitch and 12mm width. You may also wish to refer to the more advanced video tutorials linked below:



Finished SQFP44

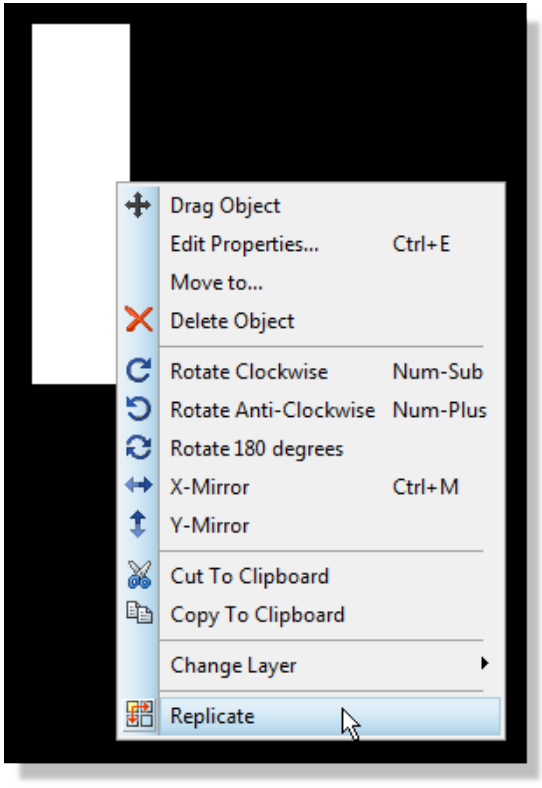
Start by selecting rectangular SMT mode. We want a pad 0.5mm by 1.8mm that should already exist as M0.5x1.8.



The Pad used in the M0.5x1.8

If the pad style does not exist you can easily create it as discussed earlier in the tutorial (Click on the 'C' button above the Object Selector).

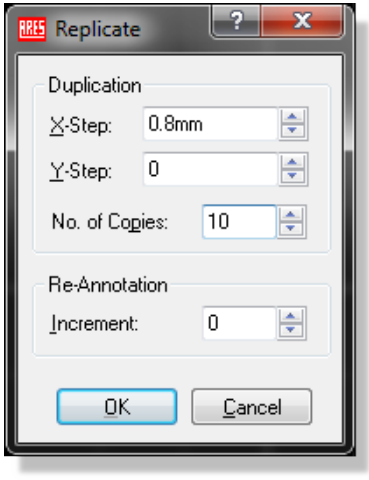
Make sure that the Layer Selector is on Top Copper and place one of the pads in the usual way. Right click to cancel placement immediately after placing a single pad and then left click on the pad to highlight it. With the pad highlighted, go to the Edit Menu and select the replicate command. Alternatively you can right click on the pad and select replicate from the resulting context menu.



Right click and select the replicate command

The replicate command will action on tagged objects so you should ensure that you have only the placed pad highlighted before invoking this command.

There are 11 pads on each side of the footprint so we need to replicate 10 times with an X-Step of 0.8mm as per the following screenshot.



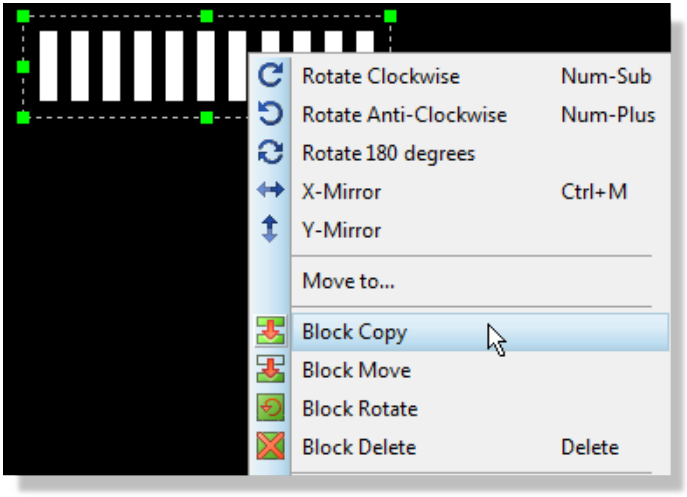
Replicate Dialogue

⚠ Note that you need to include the units of either millimetres (mm) or Thou (th) when entering values in to the X/Y-step fields.

This will give us a single row of pads of the correct pitch (zoom in, change the snap settings and measure if you would like to confirm). The next step is to duplicate this row of pads to form the bottom of the footprint.

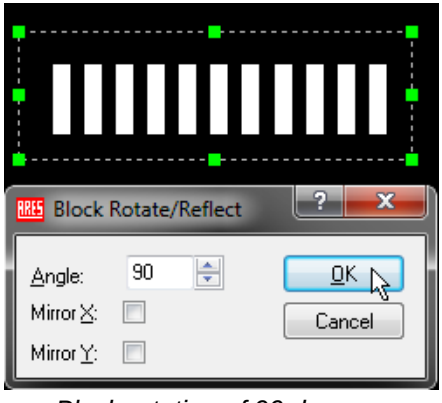
Start by right dragging a tagbox around the entire row of pads and then go to the Edit menu and select replicate. This time we want only one copy, 12mm down (or up) from the current row. Use negative co-ordinates if you want the duplicated row beneath the current one or positive co-ordinates if you want it above.

We now need to repeat the process with the other two rows. Drag a tagbox around a full row, right click inside the tagbox and select Block Copy from the resulting context menu.



Block copy will copy everything inside the tag box

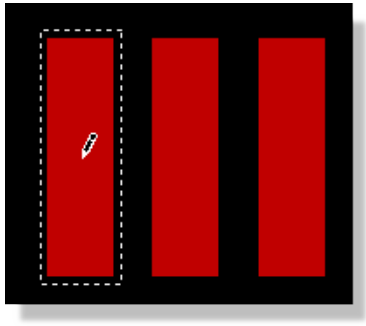
Move the mouse and drag the copy to a free area, left clicking to place and then right clicking to exit copy mode. Now, drag a tagbox around the row, right click and select the Block Rotate option, specifying a 90 degree rotation to align the pads correctly.



Block rotation of 90 degrees

In order to position this row of pads correctly it's best to first set place a marker on the location we want to move the row of pads to. For this footprint the centre of the top pad on the left hand side is 2mm below and 2mm to the left of the centre of the left hand most pad on the top row. This gives us enough information to accurately position the row.

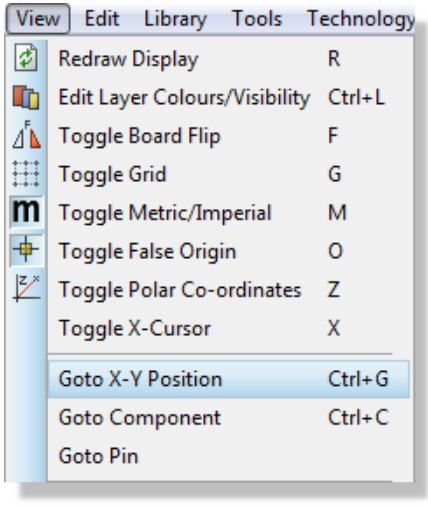
Select marker mode in preparation and then move the mouse over the left hand most pad on the top row until it is encircled. You'll want to be on a fairly precise snap setting for this (e.g. 0.5mm or F2 shortcut key).



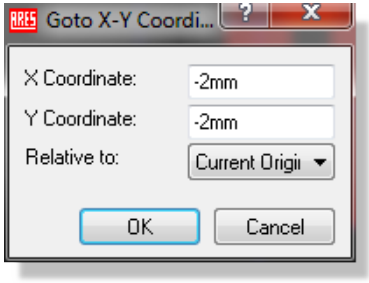
Make sure the pad is encircled

If you find your snap settings in thou you need to switch to Metric mode, either by hitting the 'M' key on the keyboard or via the Metric option on the View Menu.

Now, hit the 'O' key on the keyboard to set a false origin at this location and then invoke the Goto-XY command from the View Menu.

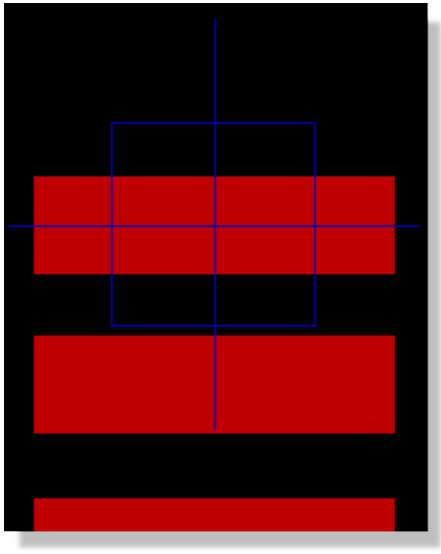


Type in –2mm in both the X and the Y co-ordinate fields (i.e. down and left) and make sure that the offset is relative to the current origin. Select OK and click the mouse twice to place the marker at this location.



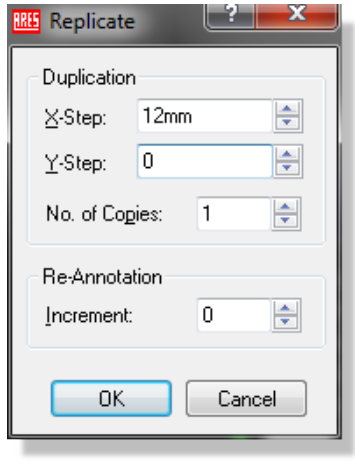
OK will take the mouse to the specified location

The marker we have placed will serve for now as the location we want to centre the top most pad of our third row. Simply draw a tagbox around the row, left depress the mouse inside the selection box and drag into position. You will almost certainly need to use the zoom commands (F6, F7 or the middle mouse wheel) and possibly also adjust the snap setting (CTRL+F1) to get accurate positioning.

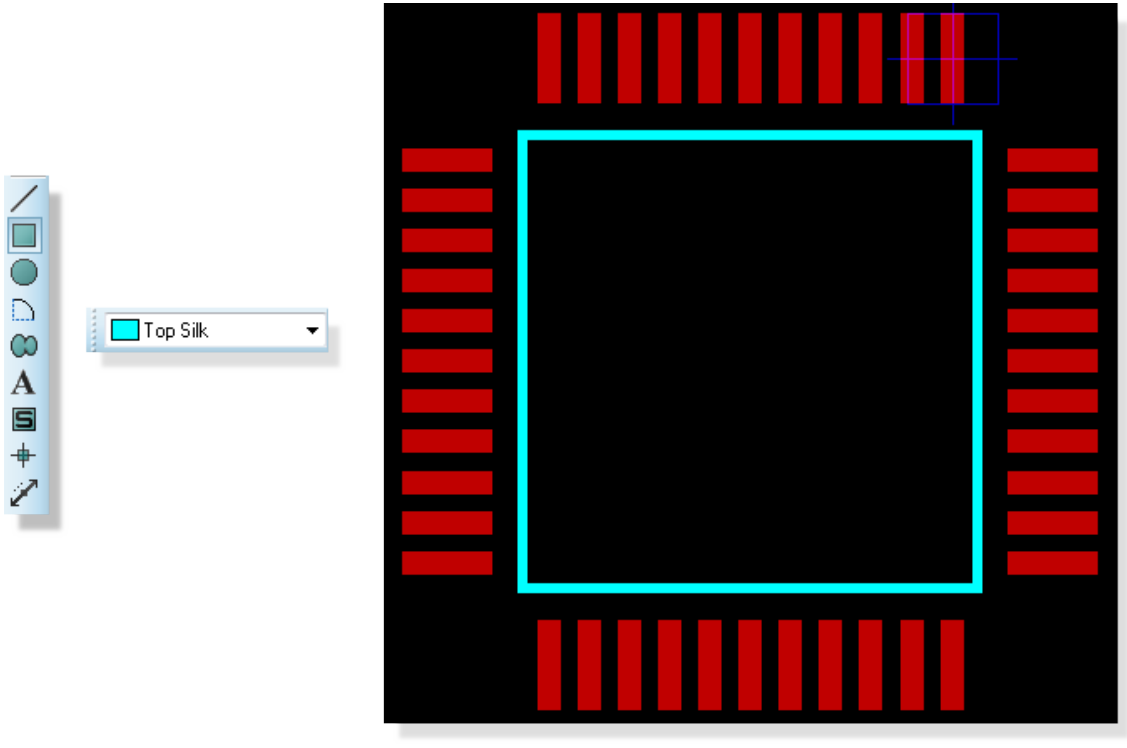


Moving the centre of a pad on to the marker

Having placed the row of pads right click on the marker and delete it – we'll come back to marker placement later on. The final stage of pad placement is to replicate the newly placed row of pads onto the right hand side. As before, draw a selection box around the row of pads and use the replicate command with an X offset of 12mm to duplicate the row.



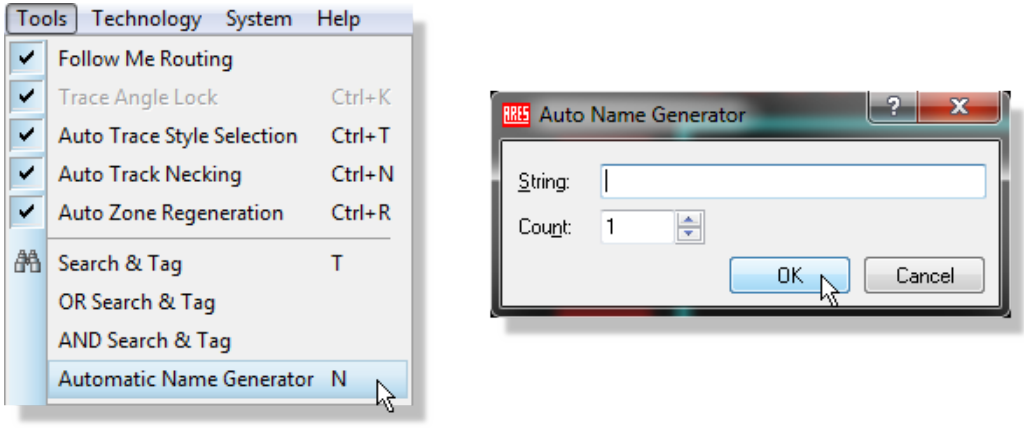
Adding the silkscreen graphics is now straightforward. Select the 2D Graphics Line icon, make sure the Layer Selector is on top silk and place four lines along the inside edges of the pads to form a box. You'll find this much easier to do if you change the snap settings upwards (e.g. F2).



Placing the Top Silk area

The next job we have is to number the pads. Start by invoking the Auto Name Generator from the Tools Menu. We don't need to enter anything in the string field here; simply leave the

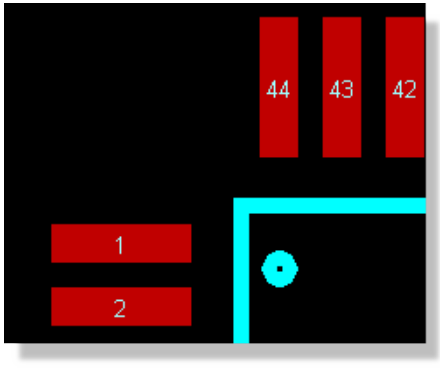
defaults and click on the pads consecutively to number them, Number 1 is the top pad on the left hand row.



Auto Name Generator

Remember to hit the escape key on your keyboard when you have finished numbering pin 44 to exit assignment mode.

It's common to place a small dot beside pin one and you can do this via the 2D Graphics circle icon. Make sure that the Layer Selector is on Top Silk and turn the snap setting down to minimum (CTRL+F1) to achieve finer control over the size.

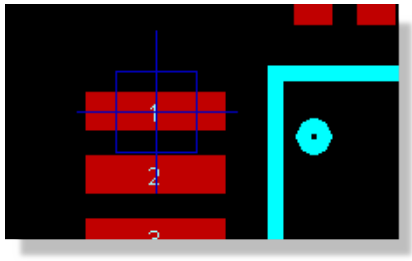


Place a small dot where Pin 1 is

The final step is to specify where we want the origin of the device to be (for placement purposes) and where we want the silkscreen graphics for the package reference to appear. Both of these are done with markers.

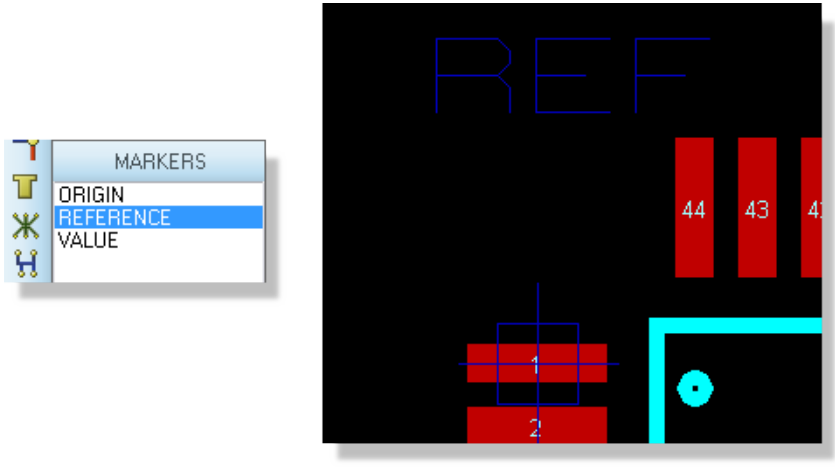
Select marker mode and make sure that ORIGIN marker is highlighted in the Object Selector. You'll recall that we used this as a reference point earlier but its real job is to specify the origin of the component for placement and rotation. This is really a decision for the user but typically the origin is either pin 1 or the centre of the component. For simplicity here, we will place the

origin on pin 1; left click to begin placement, move the mouse over the centre of pin 1 and left click again to commit the origin marker.



Origin marker placed

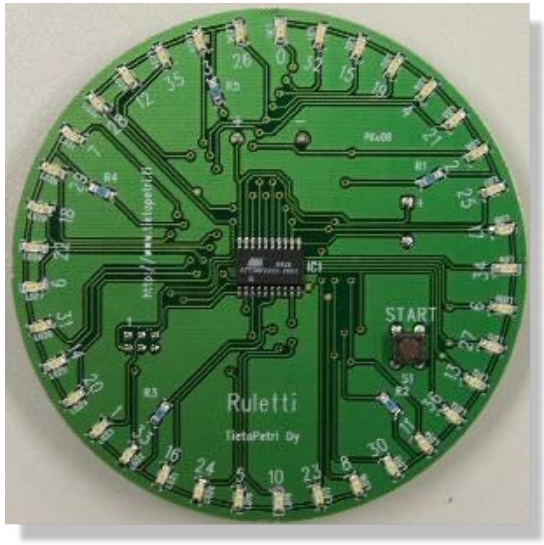
Now, change the marker type to be REFERENCE in the Object Selector. The Reference marker dictates where the component reference (e.g. U1, R12, C3) is placed by the software relative to the component. Again, this is very subjective but we'll set it above and slightly to the left of the part.



Reference marker placed

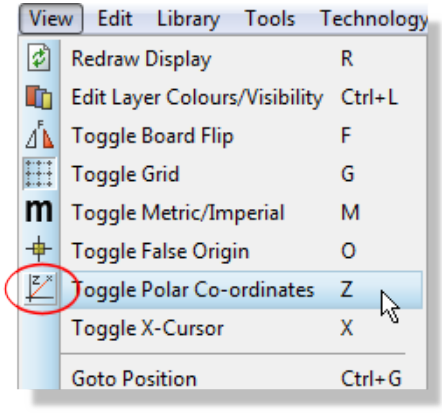
Having completed the layout of the footprint, we can now move on to the (far simpler) process of packaging the part into the libraries.

Circular Pad Placement

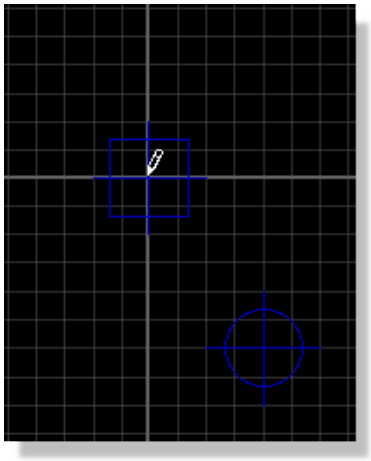


The replicate command is dynamic based on whether you are in cartesian co-ordinates or polar co-ordinates (View Menu Toggle). If you needed to place pads in a circular shape you would:

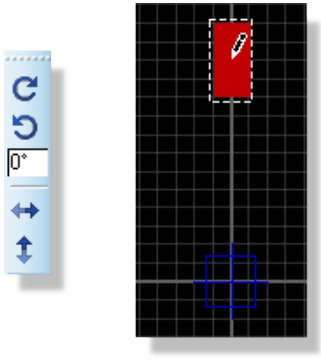
- Select Polar mode (View Menu).



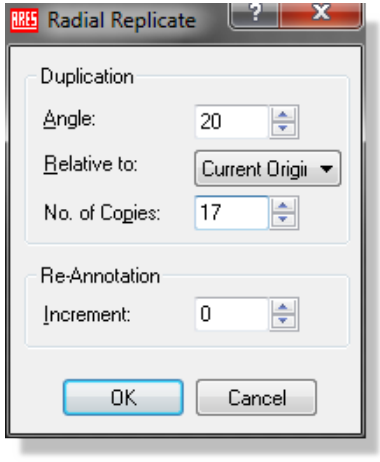
- Place a False Origin ('O' shortcut key).



- Orient your first pad and place at radius distance from (false) origin.

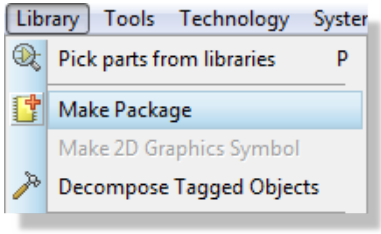


- Right click on the pad, select replicate command and specify angle (e.g. 20) and number (e.g. 17) to complete the circle.



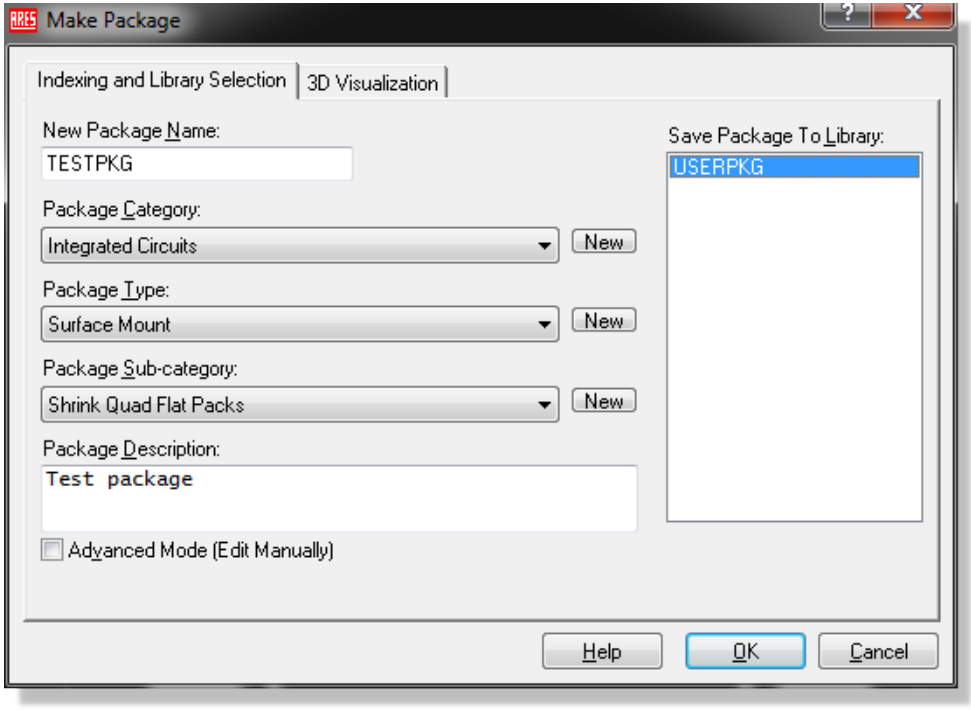
Packaging the Footprint

Drag a selection box around the entire footprint and then select the Make Package command from the Library Menu.



Launch 'Make Package' from the Library menu

The first screen is fairly self-explanatory and similar to that we've seen in the schematic capture. Note that the package description is searchable when we are browsing for footprints so a little effort to make this as descriptive as possible is worthwhile. You will also want to create the part in the USERPKG library which is the default library supplied for user footprints. For our purposes, we'll call this package TESTPKG and give it some basic entries.



The Make Package dialogue form

- You can create your own libraries via the Library Manager – please see the reference manual for details. You should not make your own parts into the other

pre-supplied libraries as Labcenter may overwrite these libraries during upgrade installs.

When this is filled out, switch tabs to the 3D Visualisation tab (do not hit the OK button at this stage – we still have some work to do). Essentially what we need to do here is provide as much information as possible in order to get a sensible 3D image of the part which can then be used when we use the 3D Viewer to examine a board. This job is greatly aided by a 3D Preview on the dialogue form that will update live as we adjust parameters. Discussion of parameters and values is beyond the scope of this tutorial and is discussed in some depth in the online reference manual (Help Menu in ARES – Help Index). For our purposes, simply fill out the property fields as shown in the next section.

3D Visualisation

This is where you would either provide a STEP file or use the Labcenter script to provide a geometric model for the part. In either case, you will get a 3D representation of the part in the 3D Viewer and can then export to Solidworks or other MCAD toolchains for case and mounting design.

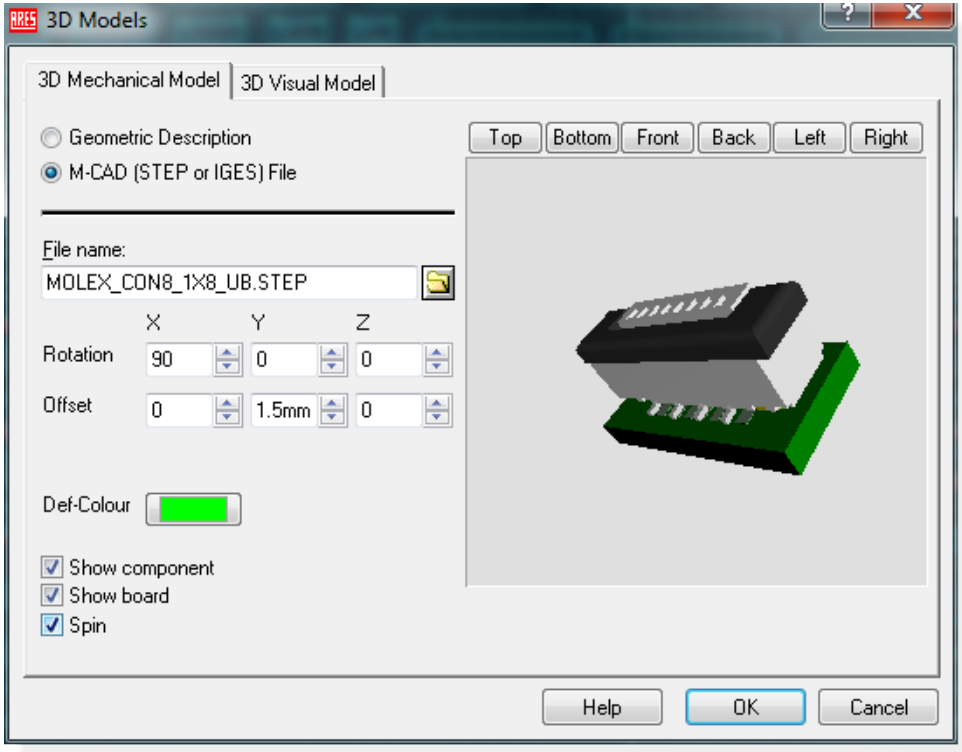
If you are providing a STEP file you must:

1) Download the STEP file from the internet and place in the MCAD directory of your Proteus installation.

2) Import via this dialogue.

3) Rotate and offset such that the part is aligned correctly on the 2D courtyard. Note that you can zoom in and out on the preview with the middle mouse wheel (or buttons) and the flip command will mirror the board at its current position.

Fortunately, this job is greatly aided by the number of STEP files available on the internet (e.g. www.3dcontentcentral.com) and a 3D preview on the dialogue form to help with alignment.



Setting the 3D parameters

Discussion of geometric model parameters, values and importing models is beyond the scope of this tutorial and is discussed in some depth in the online reference manual (Help Menu in ARES – Help Index). Geometric models are ideally suited to large pin count parts with small extrusion (e.g. BGA, QFP)

> STEP files, while common, are often poorly made resulting in large file sizes. Bear in mind the bigger the STEP file the slower the 3D Viewer will render. Consider using geometric models where no sensibly sized STEP file is available.

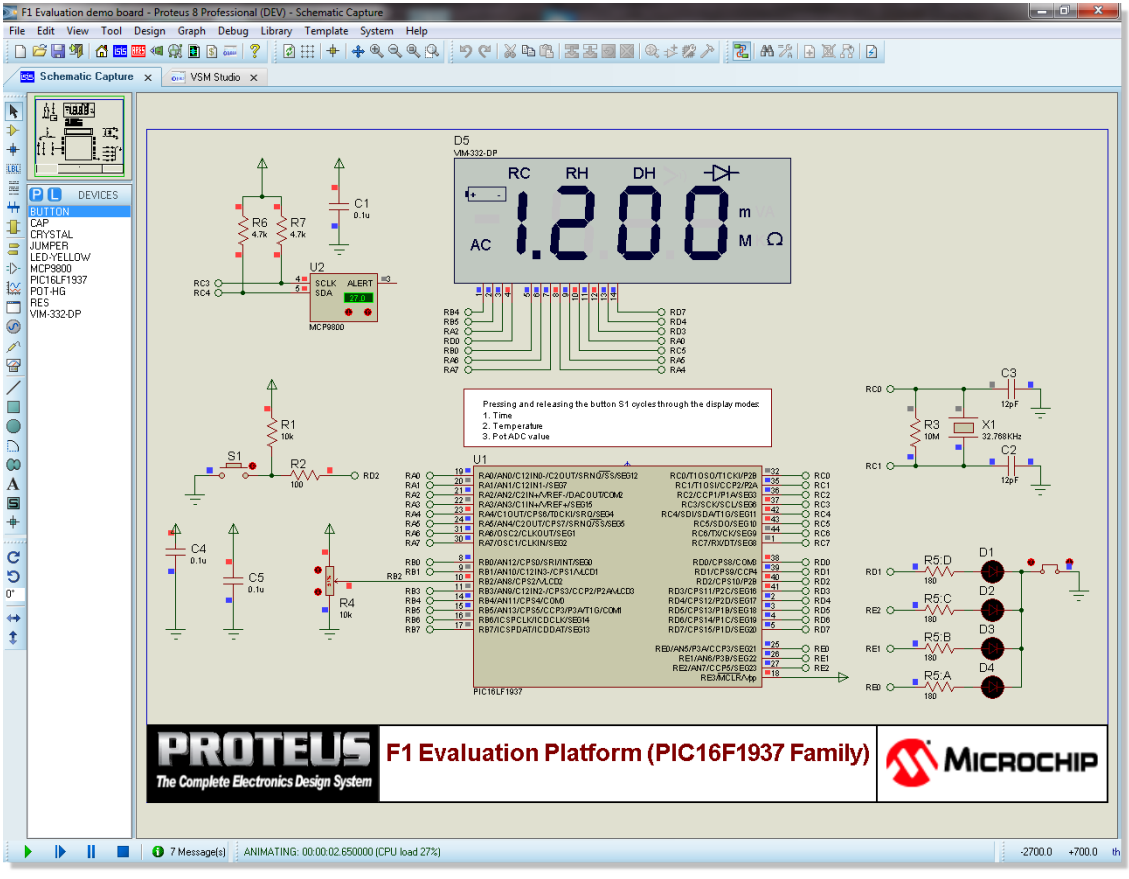
If you now select the Package Icon, you will see that TESTPKG has appeared in the Object Selector, and can be placed like any of the packages you have used so far. Also, if you place the part on the layout and invoke the 3D Visualisation engine from the Output Menu (PCB Design Level 2 or higher required) you will see the 3D rendered image of the part.

VSM TUTORIAL (Interactive)

Introduction – Interactive Simulation

The purpose of this tutorial is to show you how to conduct an interactive simulation with a microcontroller using Proteus VSM and the VSM Studio IDE. The emphasis will be on practical usage of the simulator and IDE, with more detailed coverage of each topic being available in the reference manuals. This tutorial does not cover schematic entry; if you are not familiar with drawing in ISIS then you should take the time to work through the tutorial content in the ISIS reference manual.

We will use a pre-drawn schematic of the Microchip F1 Evaluation Board as shown below.



Starting from the basics of driving the simulation from the VSM Studio IDE we will then look at some of the various debugging and measurement tools available inside the Proteus VSM software.

Requirements

To work through this tutorial you will need:

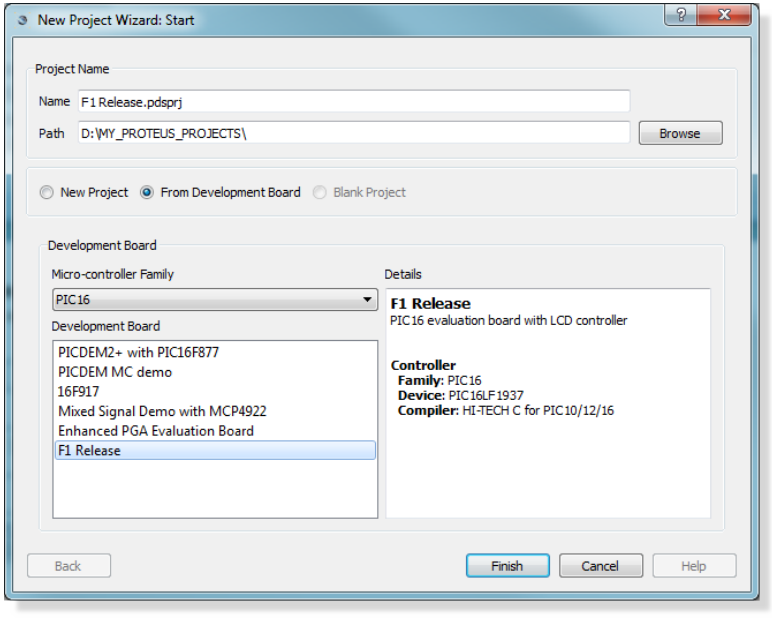
- An installed copy of the Proteus Software at Version 8.0 or later. A demonstration copy of the software can be downloaded free of charge from the Labcenter website if you do not have a professional copy.
- An installed copy of the Hi-Tech PIC16 compiler at Version 9.8 or later. Download and installation of this compiler can be managed from within the Proteus software VSM Studio IDE); we will cover the procedure for this later in the Project Setup section of the tutorial below.

We would still recommend you read through the tutorial, even if you do not have the tools installed. Most of the material - and all of the debugging techniques – are generic and will prove useful in your own designs.

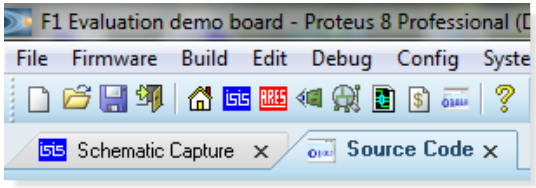
Project Setup

The first thing we need to do is setup our Proteus 8 project. Since we are using a pre-drawn schematic in the form of a virtual development board this procedure is much simplified.

From the home page in Proteus launch the new project wizard and select the development board option. Next, change the microcontroller family to be PIC16 and finally the F1 Evaluation board (F1 Release) from the listbox at the bottom. Click finish to import the project.



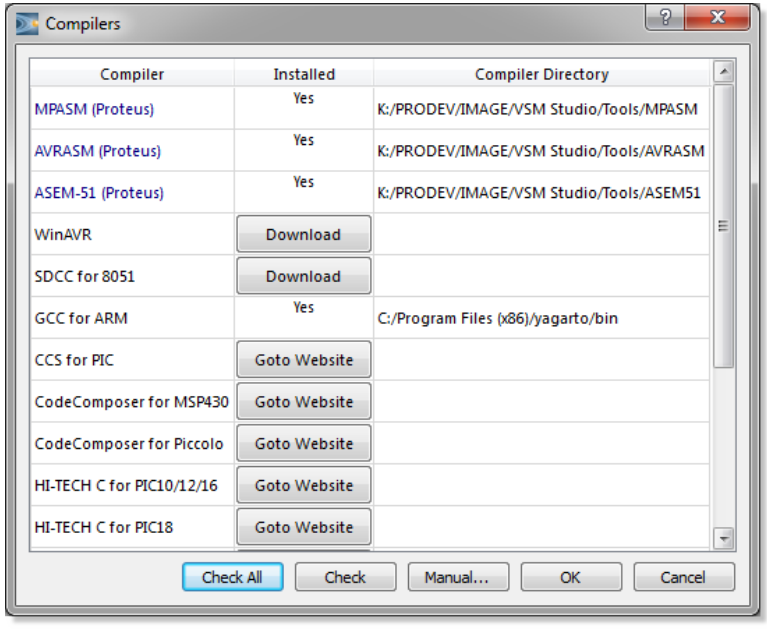
After project import you should see two tabs in the application, the schematic capture module with the F1 Evaluation Board design and the VSM Studio IDE containing the standard Microchip source code for driving the hardware.



The first thing we need to do is compile the source code and produce some firmware that we can test in simulation.

Compiler Configuration

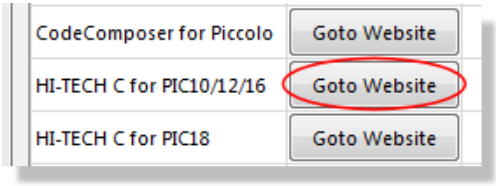
The Microchip source code was written using the Hi-Tech PIC16 compiler so we need to have this tool installed in order to compile firmware. Switch to the VSM Studio tab and launch the compilers configuration dialogue form from the Config Menu.



The dialogue form lists all of the supported compilers along with an indication of whether or not they are installed and configured or not. Pressing the 'Check All' button at the bottom of the dialogue will scan your computer for compilers and - if found - will configure them to work directly from within Proteus.
Open source compilers can be downloaded and installed directly from Labcenter's servers. Links are provided to the download page on the vendors website for proprietary compilers. The

Hi-Tech compiler we need requires download from Microchip's website so if you don't have this compiler installed you will have to:

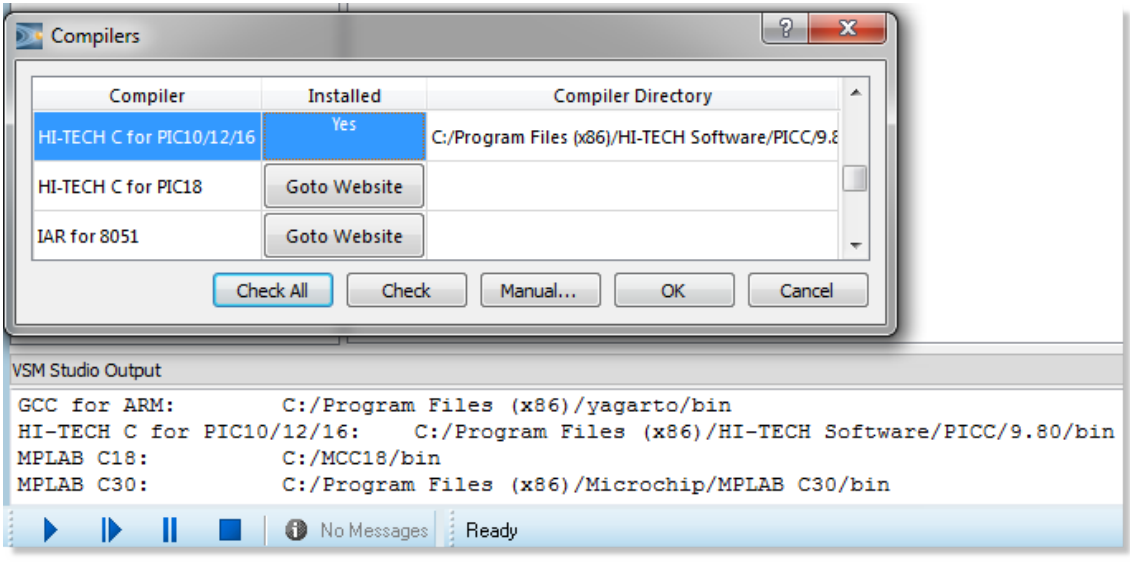
1) Click on the goto website button beside the compiler.



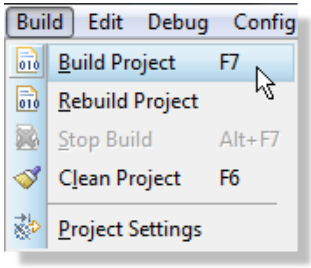
2) Download and install the compiler.

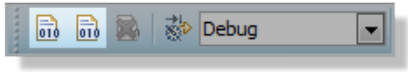
3) Click on the Check All button at the bottom of the dialogue form to configure the compiler to work with Proteus.

You should notice after running this command that the Hi-Tech compiler row on the dialogue form is marked as installed. The output window at the bottom of the IDE will also report on the results of the compiler checking.

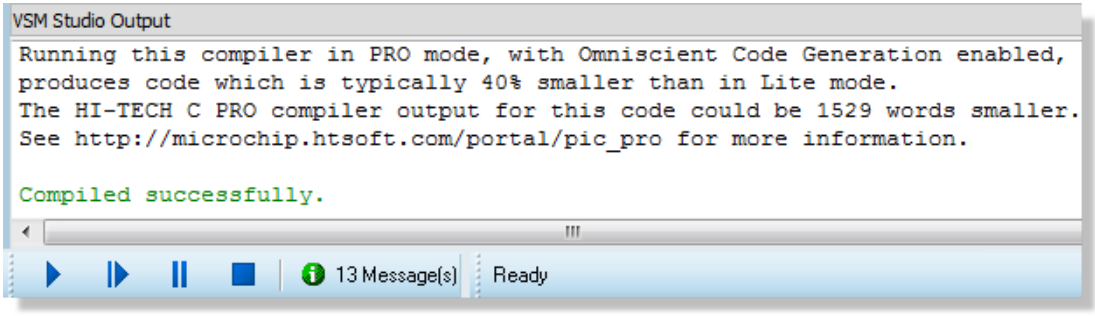


At this stage we have compiler, source code and schematic all in place. Use the Build Project command on the Build Menu in VSM Studio (or the build icon) to compile your firmware.





Output from the compiler is written to the bottom pane of the IDE and you should get a compiled successfully message on completion.



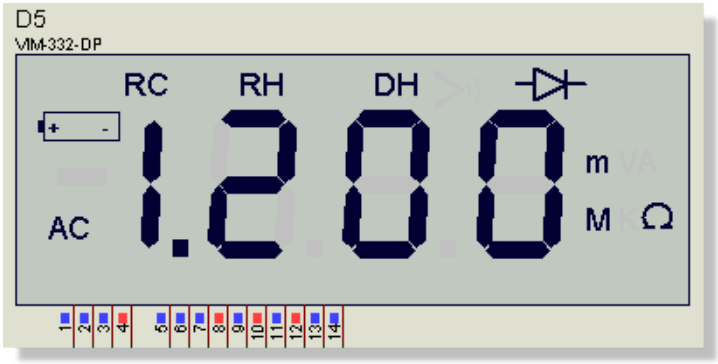
The next step is to run the simulation and test our compiled code on the virtual hardware.

> VSM Studio automatically configures common compiler options to correctly build your programs for simulation in Proteus. If you need to tweak settings (e.g. link to external library) you can do so via Project Settings on the Build Menu.

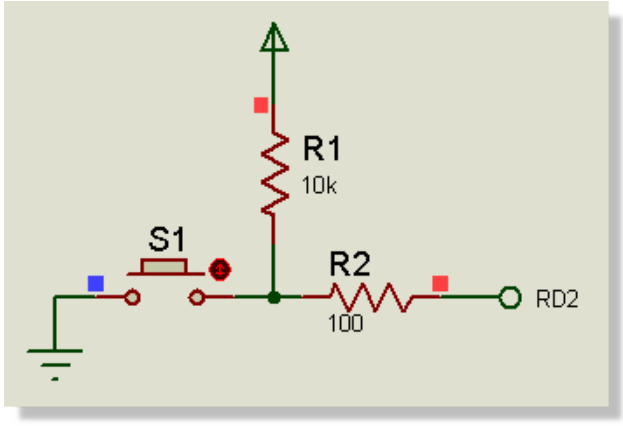
Running a Simulation

Running a simulation is as simple as pressing the play button on the animation control panel at the bottom left of Proteus.

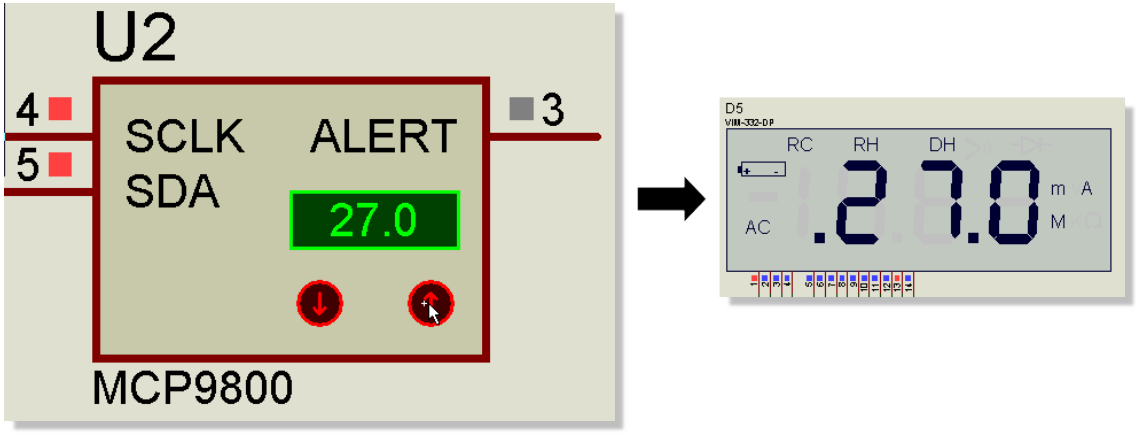
You should see that the schematic tab comes to the foreground and that the LCD panel displays the default time.



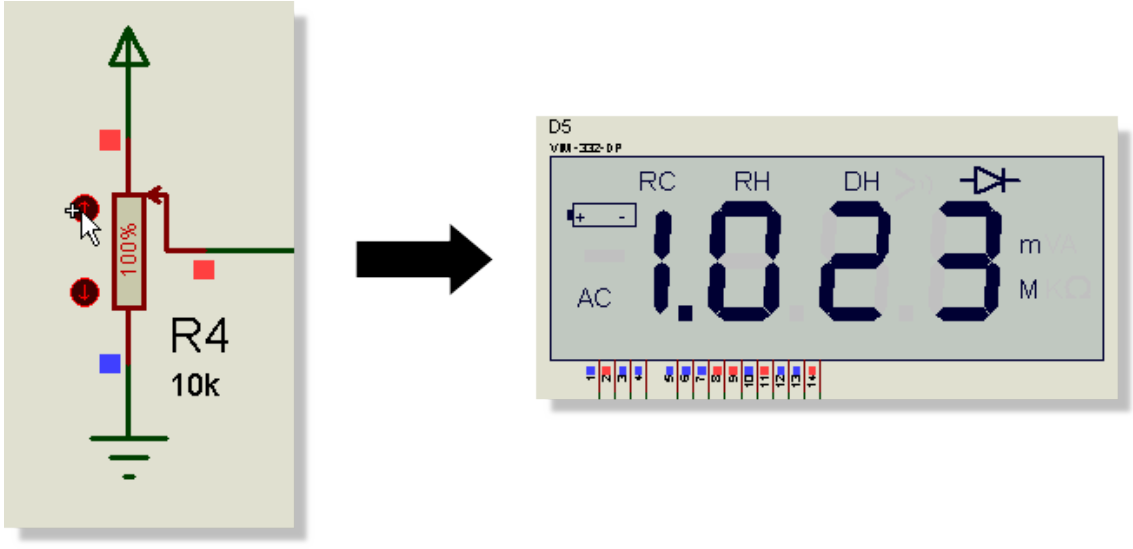
The Microchip demonstration code for the F1 Evaluation Board has three modes, one for time, one for temperature and a third which gives an ADC reading from the POT. You can interact with the simulation and cycle through the modes of operation by pressing the control button on the schematic.



When in temperature mode you can adjust the temperature on the MCP9800 temperature sensor by clicking on the increment or decrement buttons. The current temperature is transmitted on the I2C bus to the PIC processor and then formatted and displayed on the LCD.



When in voltage mode you can adjust the value of the POT. The voltage reading is converted by the processor and then displayed on the LCD panel.



- The PIC microcontroller has a 10-bit LCD so our displayed value range is 0 to 1023 over the 0 to 3.3V range.

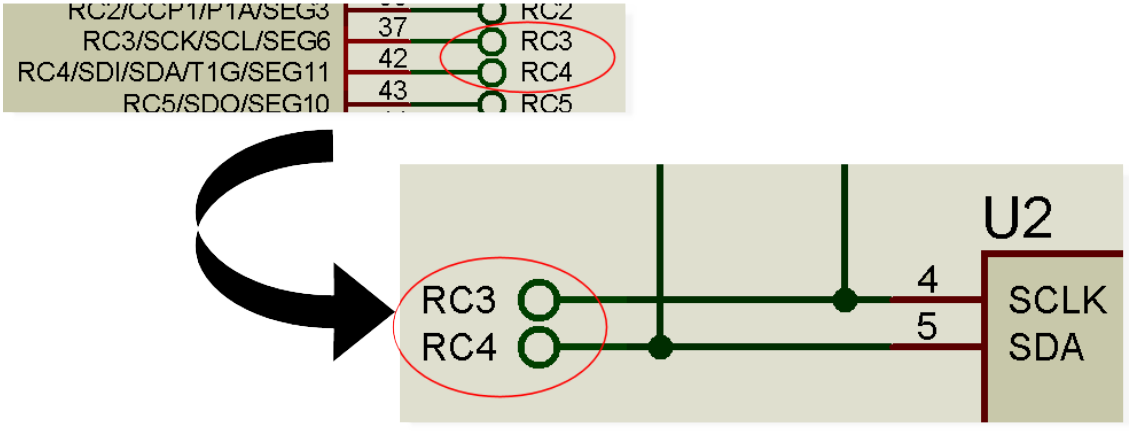
When you are finished press the stop button on the animation control panel to stop the simulation.

Important Notes

Quite a lot is happening when we run a simulation in Proteus that might not be immediately apparent.

Firstly, the micro-controller on the schematic is executing the compiled firmware file in exactly the same way as a physical PIC would execute its programmed firmware. Working with the VSM Studio IDE simply automates the 'programming' step by sending the output file to the schematic after a successful compile.

You will have noticed that almost all of the wires on the schematic end in a terminal with a name. For those unfamiliar with terminal connections it is worth explaining how this works. On the schematic, any two terminals with the same name are considered connected (like a 'virtual' wire). For example, the RC3 and RC4 terminals at the side of the temperature sensor correspond to the RC3 and RC4 terminals connected to the I2C pins on the PIC processor.



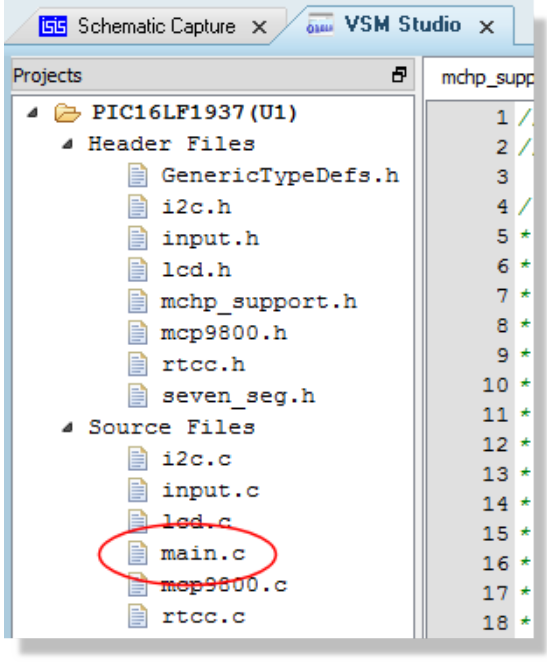
We tend to lay out schematics like this to avoid spaghetti wiring and to split the schematic into easily identifiable logical blocks. The important thing to understand is that all signals travel on the visible wires. This means that you can attach a voltage probe or measurement instrument anywhere that you can see a wire (indeed, we will do both later in the tutorial). All that ISIS does is 'jump' the signal between the two terminal ends as though there was a single wire connection. You can read up more on this type of connectivity in the ISIS reference manual.

Finally, the status bar at the bottom of the application reports the simulation time elapsed. Depending on the power of your computer, the clock speed of the processor and complexity of the schematic, a particular simulation may not run in real time but it will always run synchronously and the execution time is reported on the status bar.

As an example, on a particularly slow computer this simulation in clock mode may not advance as the clock on the wall advances but it will always advance as the simulation time advances.

Writing Firmware

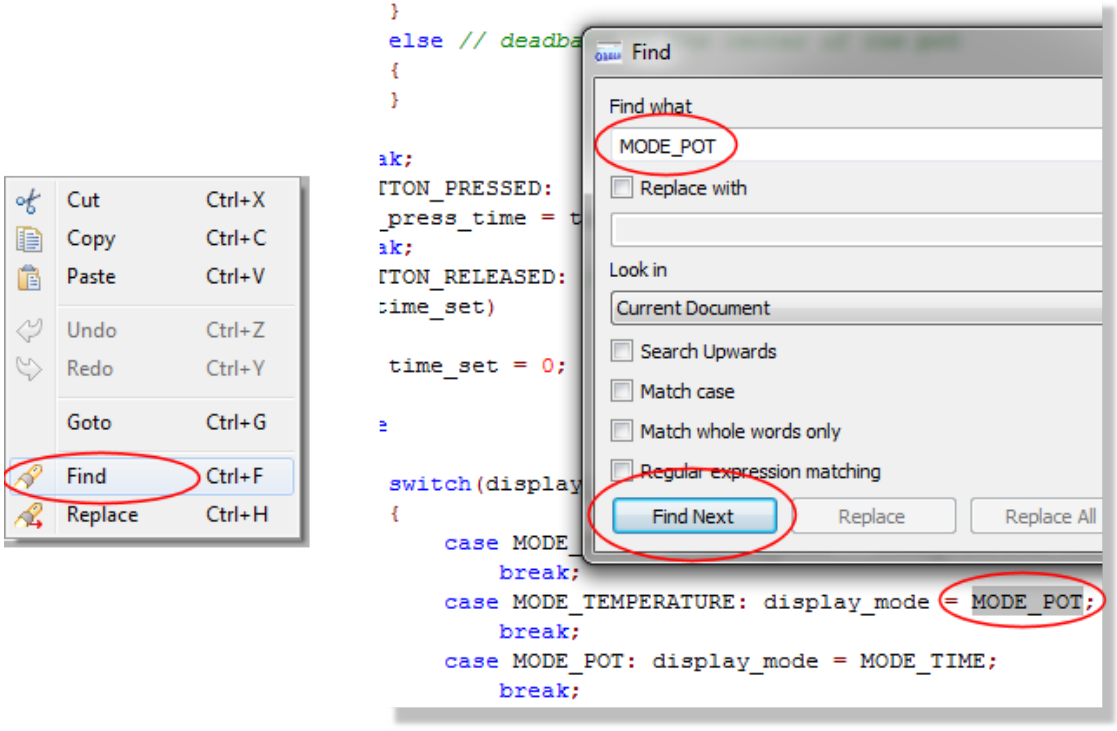
Now that we have completed the workflow through simulation let's add some functionality to the program. We know that pressing the button switches between time display, temperature and POT display so we can easily add another mode of operation here. Start by switching to the VSM Studio IDE tab and opening up the main.c file (double click on the item in the project tree).



You will see near the top of the file that there are some function prototypes and an enumeration of the modes of operation. We need to add a new function prototype for our test mode and also add a corresponding value to the enum. You can add the following:

```
40 void display_time(void);
41 void display_temp(int t);
42 void display_int(int t);
43 void display_rpm(int r);
44 void display_pot(int p);
45 void display_test(void);
46
47 volatile char blink;
48
49 typedef enum { MODE_TIME, MODE_TEMPERATURE, MODE_POT, MODE_TEST} mode_t;
```

Next, we need to find the button switching logic in order to add our new mode of operation. The easiest way to do this is to search for one of the other modes. Right click on the editing window in VSM Studio, select Find from the resulting context menu and then enter MODE_POT as the search term.



Almost the first hit is the one we are looking for, namely a switch statement on button release that cycles through the setting of the display modes. All we need do here is add our new mode.

```
switch(display_mode)
{
    case MODE_TIME: display_mode = MODE_TEMPERATURE;
        break;
    case MODE_TEMPERATURE: display_mode = MODE_POT;
        break;
    case MODE_POT: display_mode = MODE_TEST;
        break;
    case MODE_TEST: display_mode = MODE_TIME;
        break;
    default: display_mode = MODE_TIME;
        break;
}
```

Note that you need to change the display mode on the previous case statement as well.

Just below this in the code there is another switch statement which actions the current display mode. Again, we need to our case here and call our function.

```
switch(display_mode)
{
    case MODE_TIME:
        display_time();
        break;
    case MODE_TEMPERATURE:
        display_temp(mcp9800_get_temp());
        break;
    case MODE_POT:
        display_pot(input_pot());
        break;
    case MODE_TEST:
        display_test();
        break;
    default: display_mode = MODE_TIME; break;
}
```

The last step is to write a function that does something when we enter our mode of operation. We'll add this at the bottom of the file.

```
336 void display_test(void)
337 {// Do Something...
338 }
```

You can experiment and add anything you like here but we'll do something simple, writing a value to the display and adding a little binary counter on the LED's.

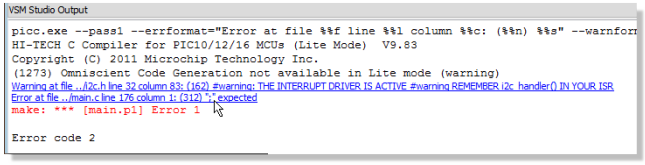
```
343  void display_test (void)
344  { static int i,d;
345
346     display_int(1111);
347     if (++d > 400)
348     { d = 0;
349       PORTE=i;
350       i = (i+1) % 8;
351     }
352   }
```

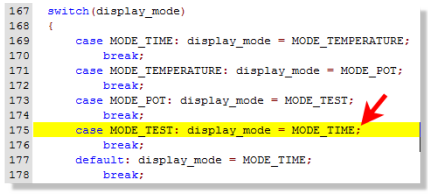
Since we are using PORTE to write the LED's here we also need to configure the port pins. We'll set up ANSELE in the peripheral config section at the top of the main function.

```
78      ANSA1 = 1;
79      ANSA3 = 1;
80      ANSB3 = 1;
81      ANSB1 = 1;
82      ANSELE= 0;
```

The final step is to build the project and launch ISIS as discussed before. If you have made a mistake (as we have here) you will get a compiler error in the output window. Clicking on this

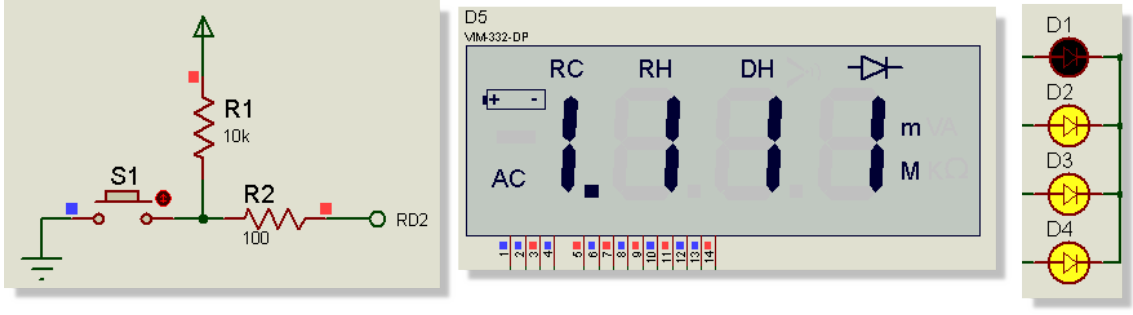
error will navigate to the problem. In our case we have missed a semi-colon from the end of the line.





Some errors - such as linker errors - will obviously not be navigable. In these cases you may need to change the options/includes via the project settings command.

When we run the simulation this time we can use the button to cycle through the modes of operation until our own is reached (time -> temperature -> test). We should see our displayed number on the display at this point and a byte cycle count on the LED's



Again, press the stop button on the animation control panel to stop the simulation.

This example is a little contrived and not terribly exciting but it does show how you can quickly write and simulate code on virtual hardware.

Active Popups

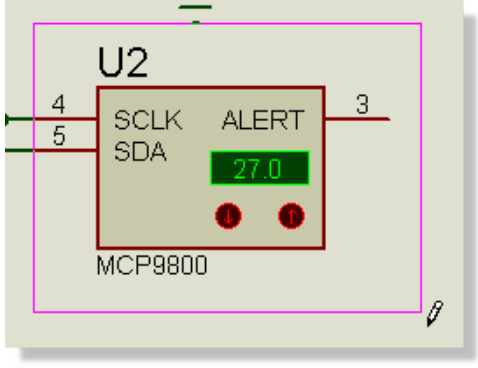
So far, we have been writing our code in the VSM Studio tab and simulating on the schematic capture tab. There is nothing wrong with this and it can work very well on two monitors if you drag one of the tabs onto another screen.

When debugging however, you are typically more interested in stepping the code and looking only at small sections of the board for verification. The active popup system in Proteus is designed to do exactly that - bringing defined sections of the schematic into VSM Studio during simulation.

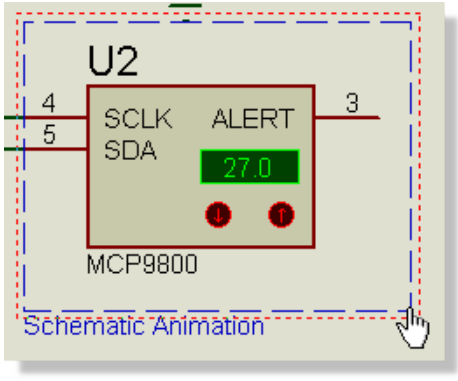
In our case, the temperature sensor is a good example of something we might want to see and interact with during debugging. To add it as an active popup:

1) Switch to the ISIS tab and selecting the Active Popup icon.

2) Start at the top left and drag a box around it.

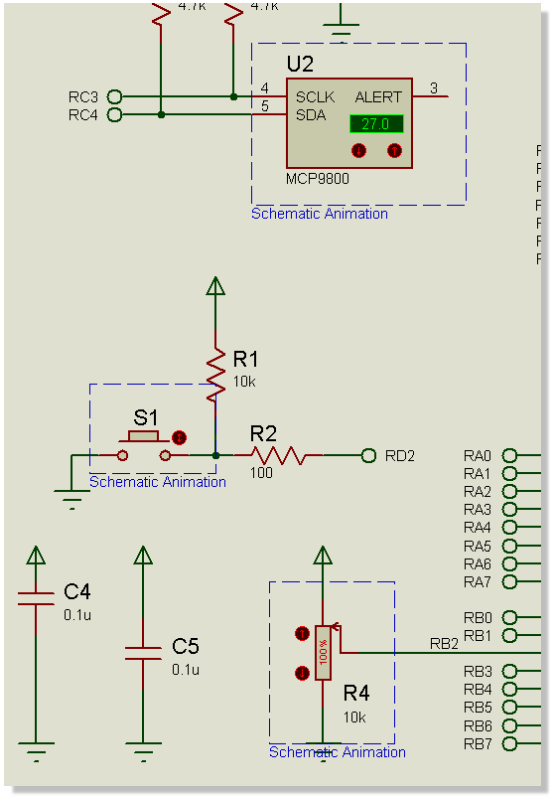


3) You should see a blue dashed line around the area when you are finished.

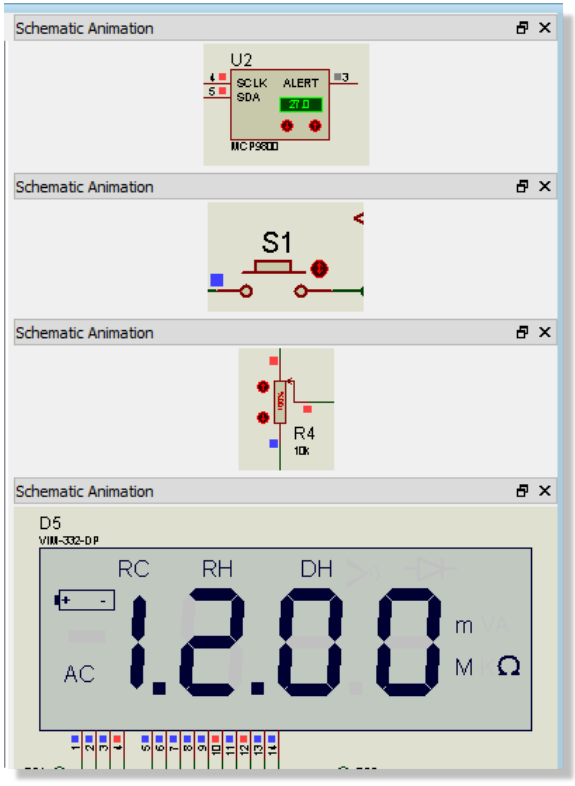


If you make a mistake you can right click on the dashed line and delete. Similarly, if your positioning is wrong, you can right click on the dashed line and drag.

Other likely candidates for active popups are the mode selection button, the POT and the display itself so we can repeat the process above, dragging additional boxes around these items. Your schematic should look something like the following when you are finished.

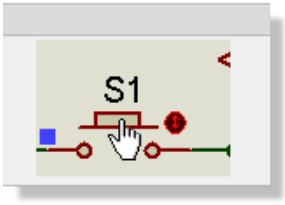


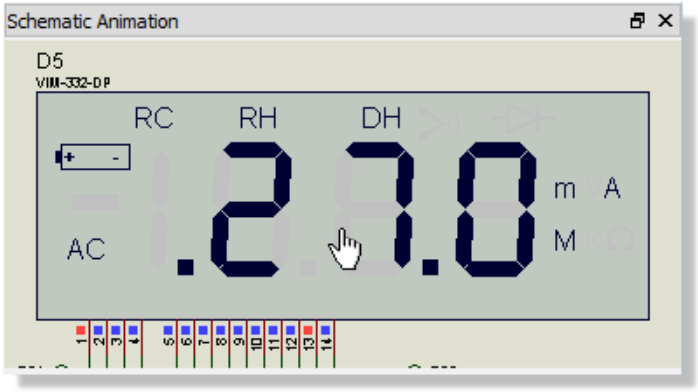
Since we have specified active popups the assumption is that we will be working (debugging) inside VSM Studio. If you press the play button to run the simulation now you should find that you switch to the VSM Studio tab and that the active popups appear docked on the right hand side of the IDE.



The source and variables areas of the tab will display a simulation running message at this point as we are freely executing code. We'll cover debugging in the next section of the tutorial.

Active popups are clever not only because they bring areas of the schematic into the debugging environment but because you can interact with them. For example, if you click on the S1 button to switch modes you should see the display change to the temperature on the I2C temp sensor and if you then adjust the temperature on the sensor the display should reflect the changes.





You can switch back to the schematic tab and you will see exactly the same status as in the active popups. When you stop the simulation (from the animation control panel) the active popups will disappear and VSM Studio will switch from its 'debugging layout' to its 'design layout' where you can once again edit and compile your source code.

> You can only create or adjust active popups on the schematic when the simulation is stopped. You can however resize active popups inside VSM Studio by dragging while the simulation is running.

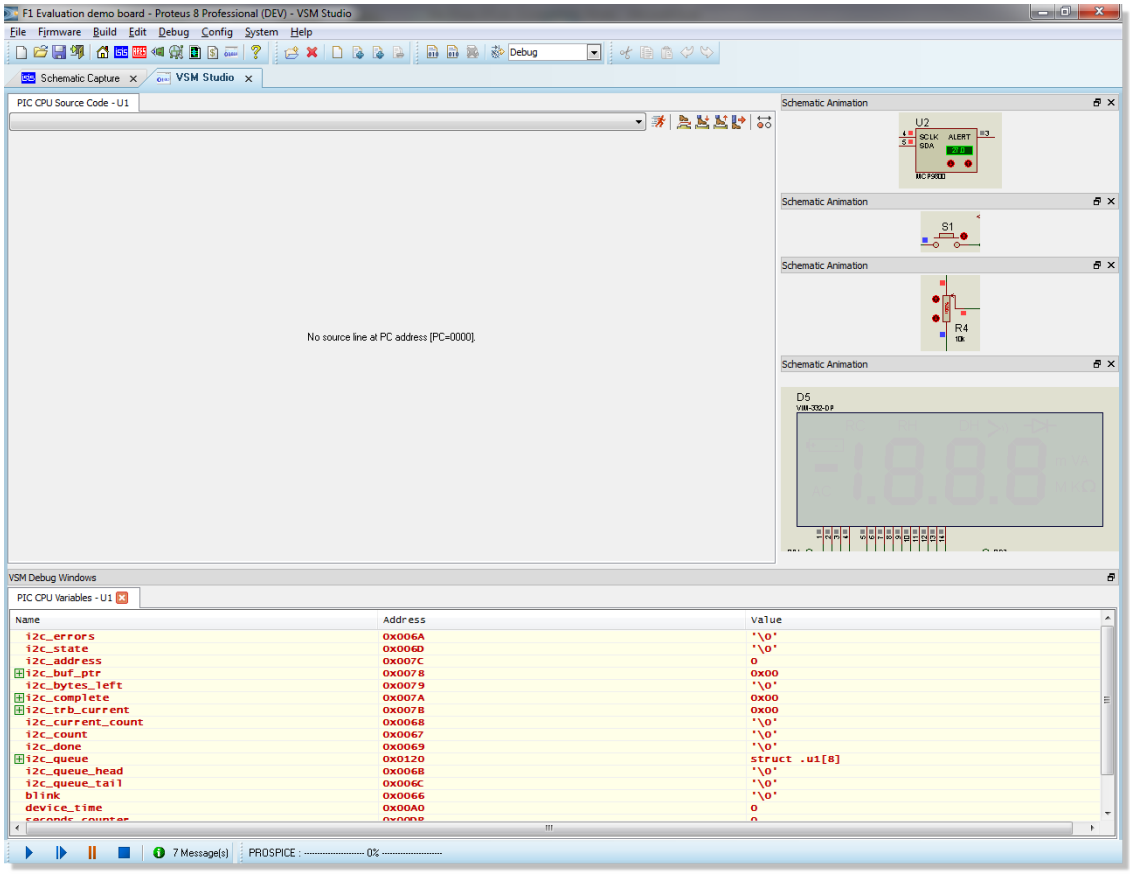
Now that we have configured our active popups we can start to look at how to debug problems when things don't quite work the first time.

Basic Debugging

Much of the real power of Proteus VSM comes from its debugging capabilities. We have already seen how we can write code and test during free running simulation and we'll now look at how to step through our code in simulation time.

Start the simulation with the pause button on the animation control panel

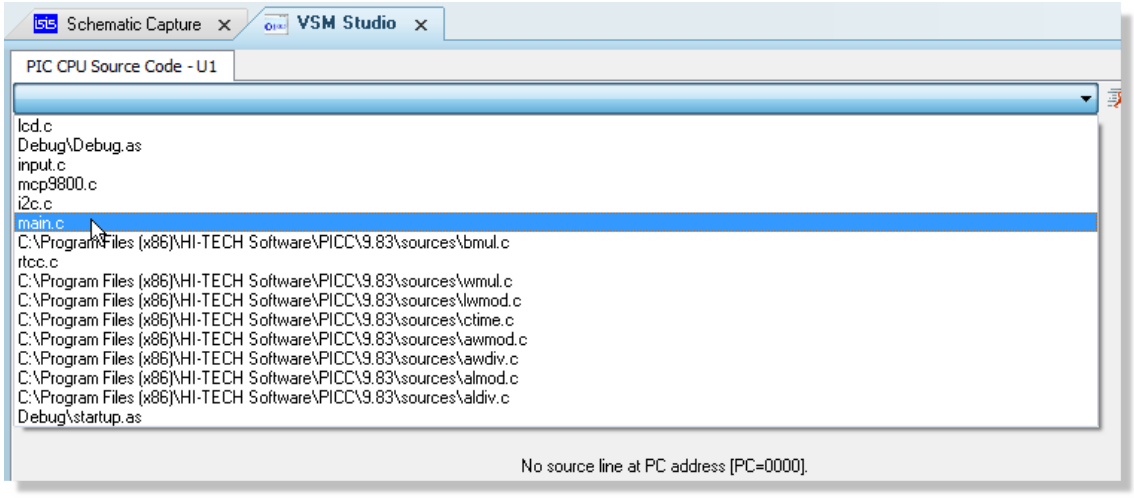
The program will switch to the VSM Studio tab, active popups will display on the right hand side, the main panel will report a 'no source line message' and a list of program variables will appear at the bottom of your screen.



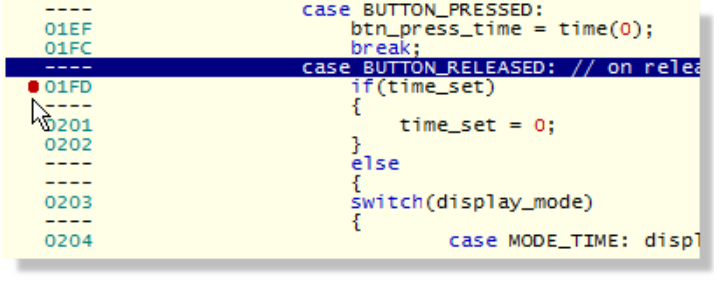
At this stage the simulation has 'booted' and a stable operating point has been found but no instructions have executed and no real time has elapsed.

No source code is shown as there is no source line at the current program counter value

You can select any of the program source files from the combo box at the top of the window but we will start with the main.c file which contains the principle program control loop.



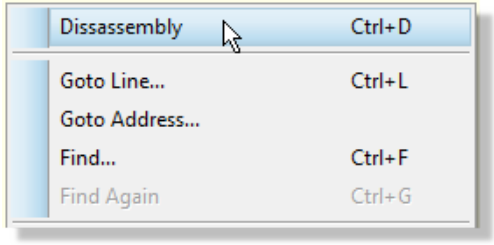
To take a simple example, let's set a breakpoint in the main event loop when a button is released. The easiest way to set/toggle a breakpoint is to double click on the line of code that you wish to break on. The breakpoint indicator should appear at the left of the source window beside the line of code in question. If you set a breakpoint by mistake then you can remove it or toggle it off by double clicking again on the line (or from the right hand context menu).



If we run the simulation now (play button) it will switch happily execute in free running mode until we release the mode switch button (S1) at which point the code for button release will be hit and the simulation will pause. You can do this easily by clicking on the button in the active popup window.

Having hit our breakpoint we can single step our code (and the system) via the usual commands which can be found at the top right of the source window or on the debug menu in VSM Studio. Alternatively you can use the F10 and F11 shortcut keys for 'step over' and 'step into' respectively.

If you really need to examine behaviour in detail you can even step the code at assembly level. Right click on the source menu and select 'disassembly' from the resulting context menu and then use the same step commands as before.



Right click and select disassembly again to return to high level source code stepping

> You may notice during step debugging that the active popup for the display does not show a full legible output. This is correct and a consequence of the display being multiplexed.

If you want to run through to the next button release simply press the play button on the animation control panel and then click the button on the active popup to trigger the breakpoint again.

Right click and select clear all breakpoints when you are finished and then press the stop button on the animation control panel to end the debugging session.

Important Notes

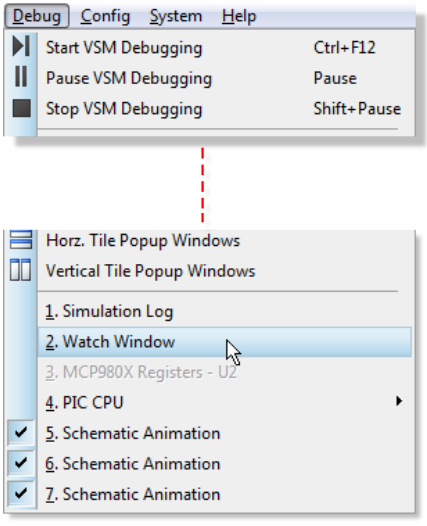
When you hit a breakpoint or single step debug in Proteus it is important to understand that you have the entire system under time control. This means for example that capacitors will not discharge or motors lose momentum when the system is paused. When you step over a command the instructions are executed, the effect of those instructions are propagated through the system and then the system will pause again. You can see this controlled advance of time via the output on the status bar during debugging.

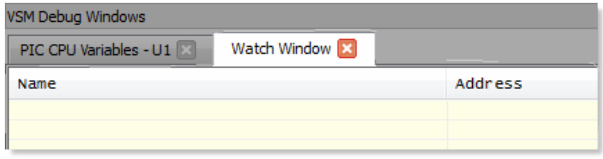
Many additional debugging windows are available which are not discussed in this tutorial. All of these can be launched from the debug window in VSM Studio and will appear at the bottom of the IDE. Data will be displayed on the windows whenever the simulation is paused. The watch window - which is a special case - is discussed separately in the next section of the tutorial.

The Watch Window

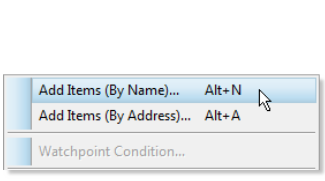
The watch window is the one debugging window that provides live data during free running simulation and it also gives us a different way to target breakpoints. To start with, let's use the watch window to monitor the ADC conversions from the POT.

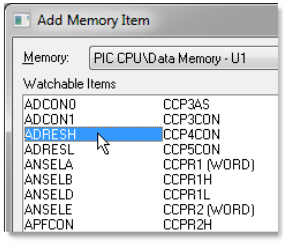
Start the simulation using the play button at the bottom of Proteus and then launch the watch window from the debug menu in VSM Studio. You should see it appear as a docked tab at the bottom of the IDE.

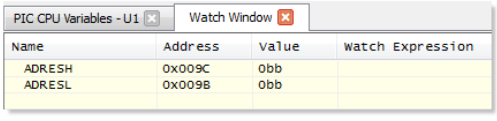




Next, right click on the watch window and select 'Add items by name' from the context menu. We want to add the ADC registers ADRESH and ADRESL which you can do by double clicking on an entry. Exit the dialogue form when you are done and you should see both items appear in the watch window display.

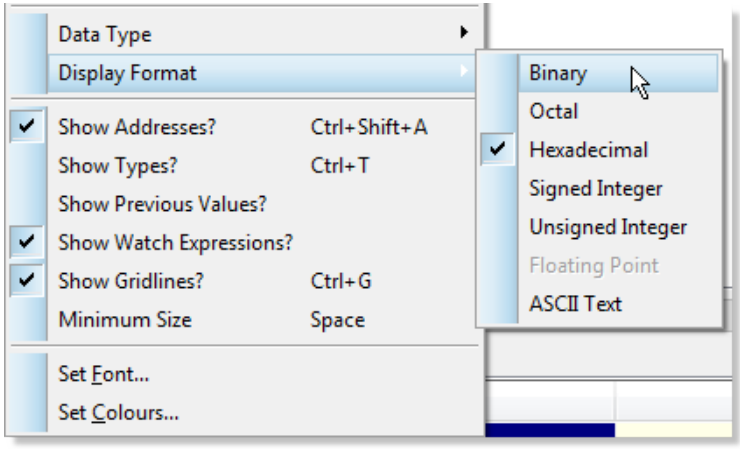




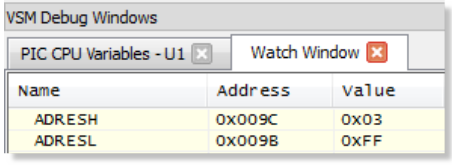


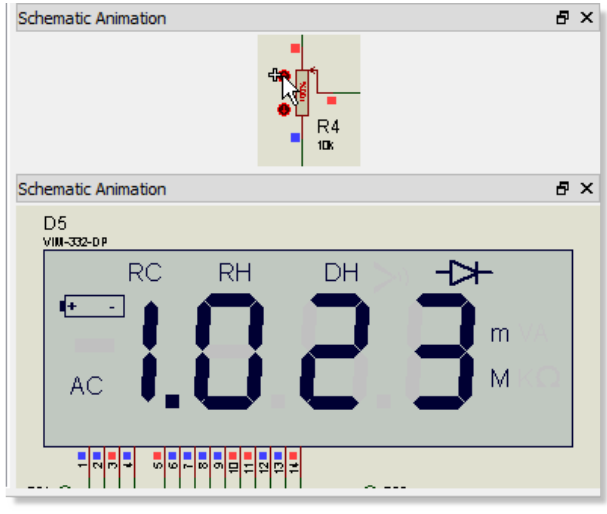
We now need to use the active popups (or switch to the schematic tab) to switch into POT reading mode. Click on the control button three times to cycle the modes and then use the actuator buttons at the left of the device to vary the value on the POT.

You can change the display format of the items to binary if you find it easier to understand the results. You would do this by right clicking on each item in the watch window and changing the display format to binary.

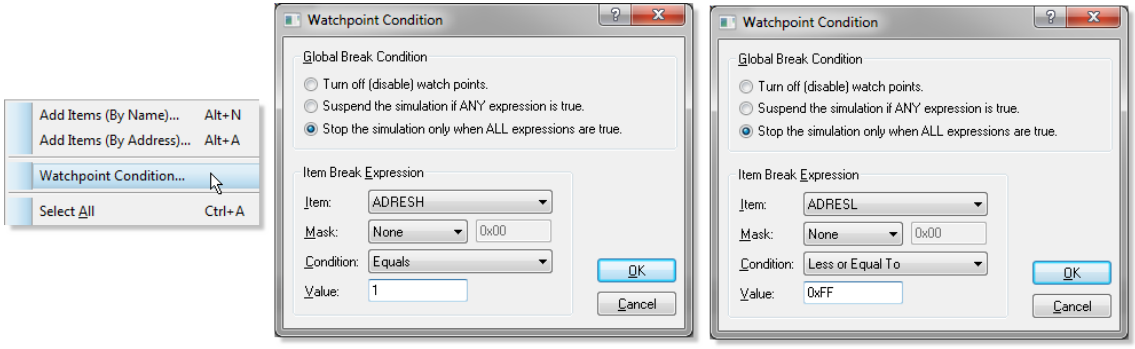


Since this is 10-bit ADC we would expect the maximum value to be 1023 or 0x03FF and we should see this value when the POT is fully ramped up.

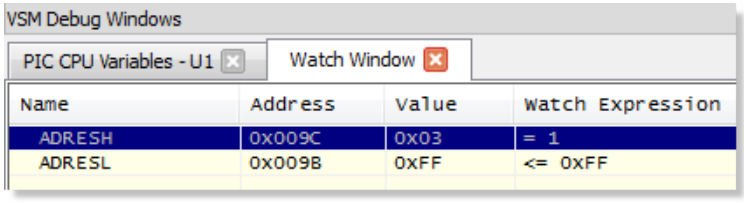




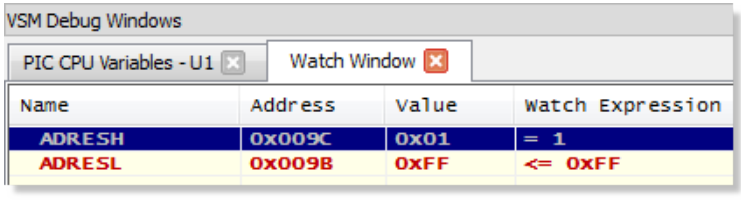
Now let's assume that we needed to catch the midpoint on the way back down. We know that the midpoint value result should be around 0x1FF so we can set conditions on the watch items to trigger a breakpoint. We do this by right clicking on a watch item and selecting watchpoint condition from the context menu. In our case we would want to stop when both ADRESH is equal to 0x01 and when ADRESL is less than or equal to 0xFF. This will work as the first time we hit this value on the way down from the maximum will be at the midpoint voltage.



Note that you need to separately configure ADRESH and ADRESL. The watch window display should look like the following when you are finished.



Having set up the watch conditions all we need do now is adjust the POT back down until the breakpoint kicks in when the conditions are met. As before you can do this either via the POT active popup inside VSM Studio or by switching to the schematic tab. When triggered you should see that the value on the watch items is 0x1FF and - if need be - you could start single stepping your code at exactly the point the conditions were triggered.



- Watch conditions are particularly useful for things like debugging timer code when you want to trap an overflow condition.

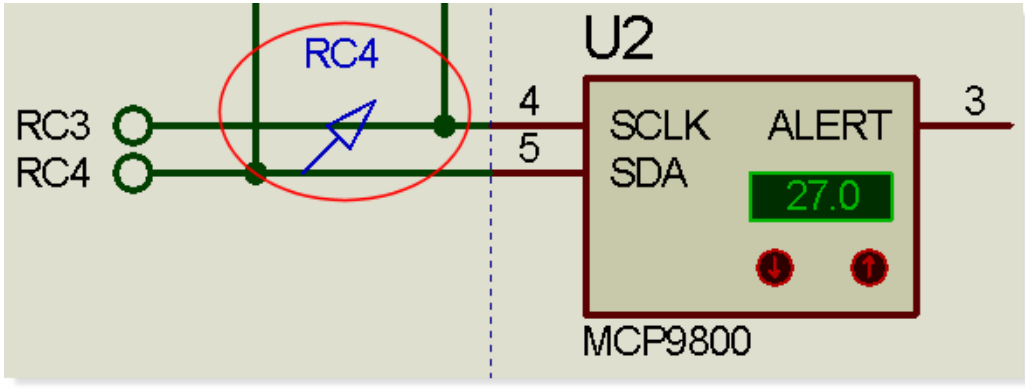
- You can disable watch point expressions by right clicking on a watch item, selecting watchpoint condition and then selecting the 'turn off watchpoints' option from the resulting dialogue.

Before we move on to discuss hardware breakpoints either disable the watchpoint conditions (as described above) or delete the watch items completely and then press the stop button on the animation control panel to end the simulation.

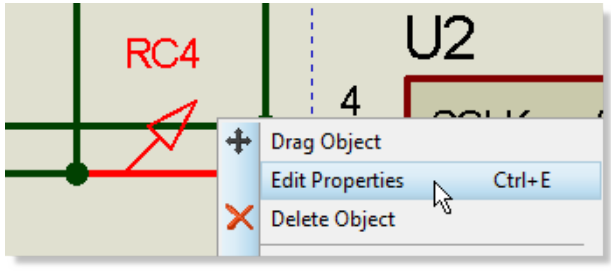
Hardware Breakpoints

So far, we have looked at how to break into the simulation based on software conditions (watchpoints and breakpoints). It is also possible to use hardware breakpoint objects in order to check the code when a hardware condition occurs. As an example, if we wanted to catch the beginning of activity on the I2C bus we could set up a hardware breakpoint on the I2C line as follows.

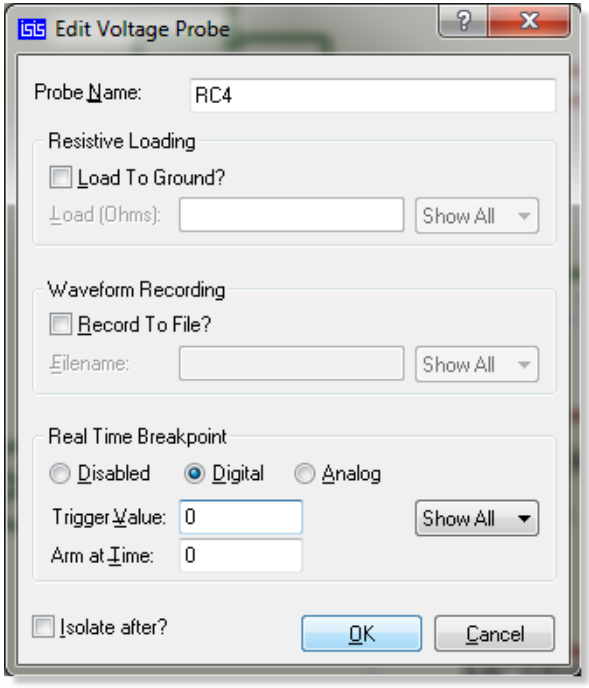
Switch to the schematic tab and then select the voltage probe icon from the mode selector on the left hand side. Next, click on the editing window to begin placement, move the mouse over the SDA line on the bus and left click again to drop the probe on the wire.



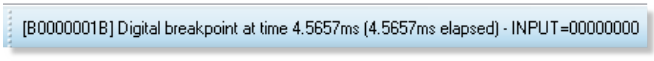
Now right click on the probe and select edit properties from the resulting context menu.

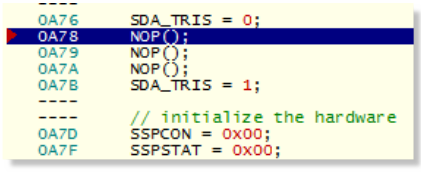


Select digital breakpoint and we want to trigger on active low so enter 0 as the trigger value.

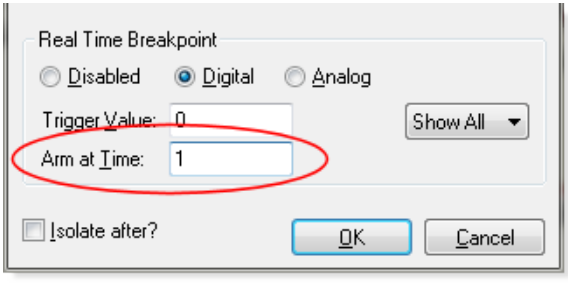


Hit OK and then press play to run the simulation. You should find that the simulation will pause almost immediately and the source window in VSM Studio will show that we have just executed a command to set SDA low.

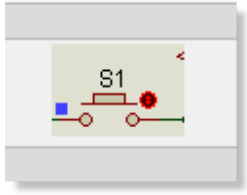




This breakpoint is actually taking place inside the initialization routine. If we wanted to skip this and activate only when we enter I2C mode we need to set and arm time on the breakpoint. Do this by stopping the simulation, switch to the schematic module and edit as before. Set the arm time to 1 second.



This time when we run the simulation the breakpoint will not trigger until we press the mode selection button to enter temperature mode (where temperature is transmitted on the I2C bus).



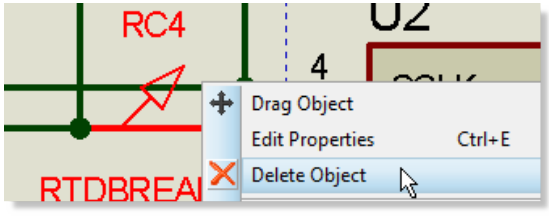

```
----     }
002B    i2c_handler();
002E }
----
---- void display_temp(int t)
---- {
----     static int counter = 0;
----     static int integrator=0;
----     int average;
----     static int v = 0;
```

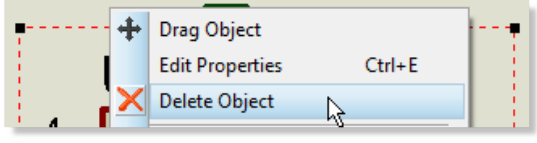
You should find that you end up inside one of the I2C routines in the source code. You can follow the code path from here using the stepping and debugging described in previous sections. Stop the simulation again when you are finished.

Interactive Measurements

It is quite common to want to examine signals and waveforms on the design and Proteus provides two main ways of doing this. As an example, let's assume we need to examine the I2C traffic from the MCP9800 temperature sensor; we can do this either interactively or by more traditional graph based methods.

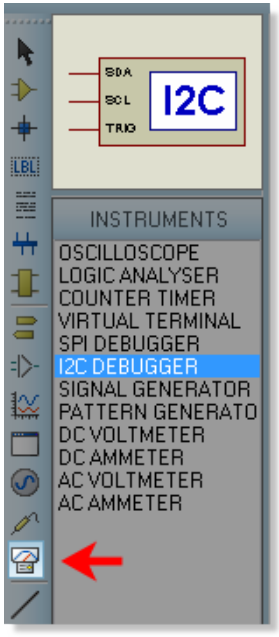
The first thing we need to do is tidy up the schematic and remove the debugging items we have used previously. You can remove the voltage probe by right clicking on the probe on the schematic and selecting delete from the context menu. Similarly, we can right click on the border of the active popup objects and delete them.



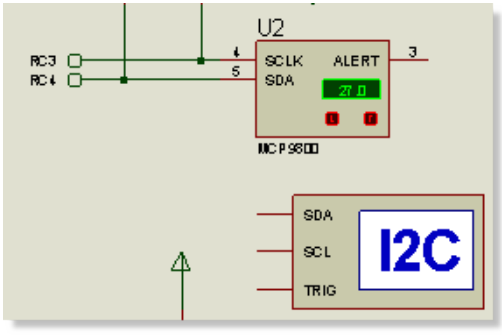


> When an active popup is defined on the schematic Proteus will switch to VSM Studio when the simulation starts as it is assumed you are primarily debugging source code. Since we are now analyzing signals we want to remove the active popups so that the simulation starts with the schematic tab open.

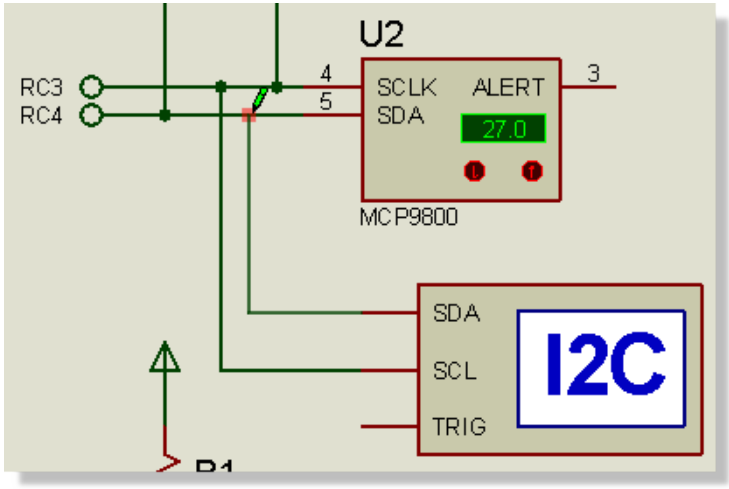
The next thing to do is to get the I2C debugger placed and connected. Select the instrument icon which produces a list of available interactive measurement tools in the parts bin. In our case, the obvious candidate is the I2C protocol analyser.



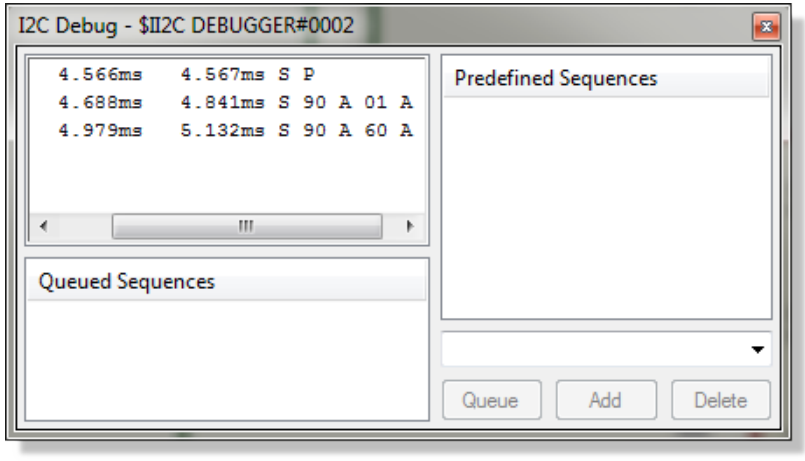
Click on the I2C analyser in the parts bin to select it and then place it on the schematic in the usual way. For ease of wiring, you'll want to drop it underneath the temperature sensor.



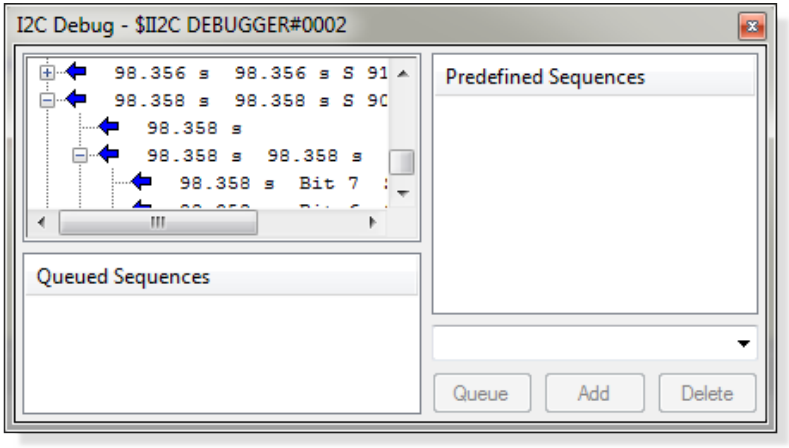
Next, we need to wire up the pins. Note that the cursor turns green on both a starting and stopping point for a wire. The wiring procedure is therefore to position the mouse over the starting point (the pin on the I2C debugger) until the cursor turns green, left click to start placement, move mouse to destination point (cursor turns green) and finally left click to terminate placement. Use this technique to connect both the SCL and the SDA lines to the I2C bus.



Once we have wired up the analyser we can start the simulation. You should see straight away that there is an additional window for the I2C debugger and what seems to be some initialization/identification traffic. To view normal transmission we need to press the button on the schematic to switch into temperature mode display.

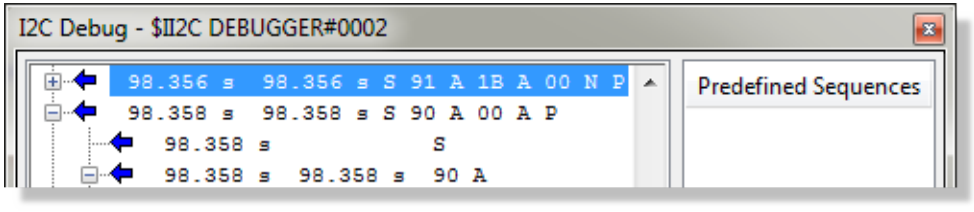


Since the code is constantly polling you should then pause the simulation from the animation control panel so that we can take some time to analyse the traffic.

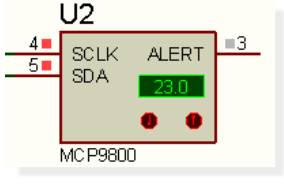


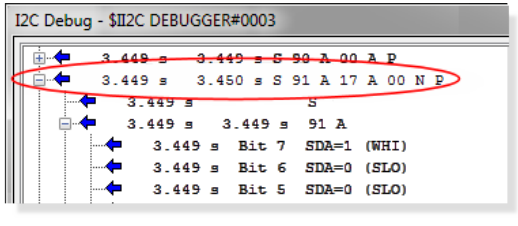
If you are working within a single frame the VSM Studio tab will come to the foreground when the simulation is paused as it is assumed you are interested in the source code pending execution. The analyser will however remain on top. If you want to see both the schematic and the source at the same time you can either use Active popups or drag the VSM Studio tab onto some free space to separate it.

The syntax used in the protocol analyser is standard and should be familiar. You can expand any sequence via the '+' box on the left hand side. The peripheral (MCP9800) has a standard address of 0x90 so we can see from the read requests that the sequence is start (S), followed by read request 0x91 (Read request, bit 0 set), followed by 0x1B (data) and so on. As you would expect the data received (0x1B) is 27 decimal corresponding to the temperature currently displayed on the peripheral.



You can experiment with this by running the simulation, changing the temperature on the MCP9800 schematic part and then pausing again to check the display output on the I2C debugger.





Do note that with constantly polling firmware such as this there is considerable overhead in terms of performance as we are constantly writing textual data to the display. However, in most cases you are using the instrument for testing or debugging, at which point the simulation speed is secondary to problem solving. As with all debugging windows you can close the debugger when not in use and then re-open it from the Debug Menu in either VSM Studio or ISIS.

Further Reading:

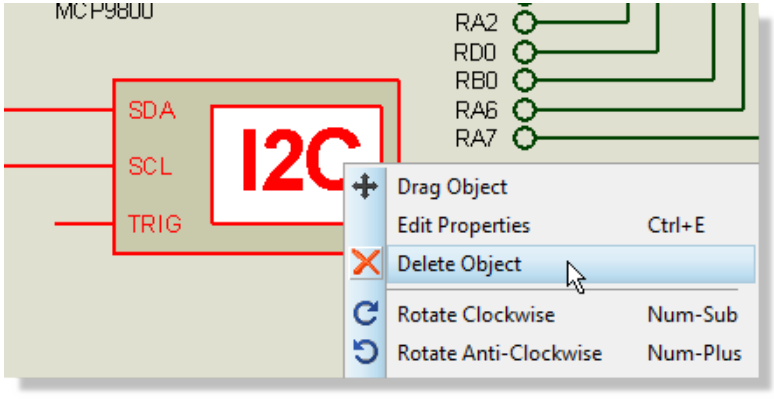
- More information on the protocol analyser - along with other instrumentation – is available in the Proteus VSM reference manual. In particular, note that you can use the analyser as an I2C master (or slave) device as well as simply as a monitor.
- More information on picking, placing and wiring on the schematic can be found in the ISIS tutorial documentation, launchable from the home page.

Graph Based Measurements

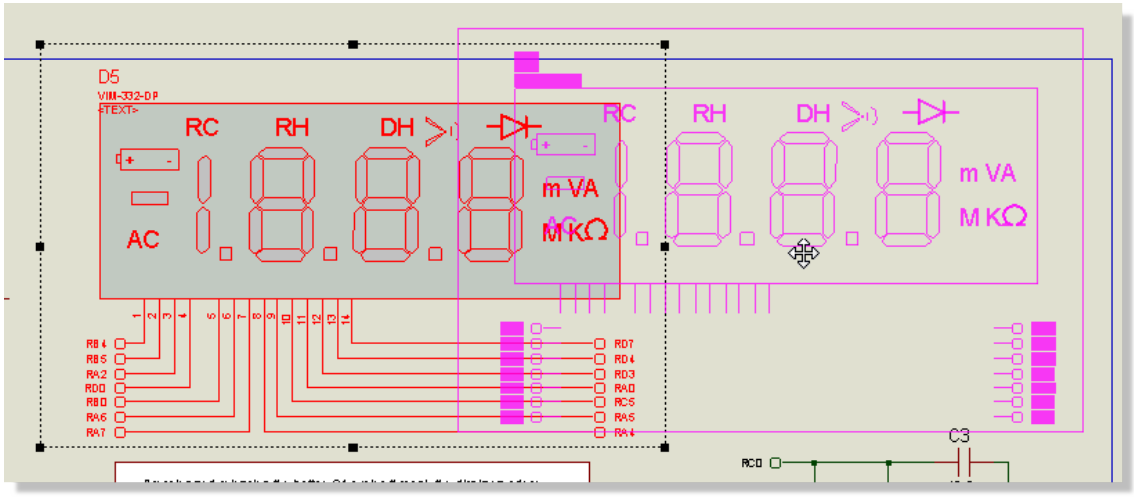
We can look at the same traffic in a different way by using graph based simulation. There are however some important differences which affect how we set up the simulation, namely:

- You cannot interact with the circuit during a graph based simulation.
- A graph based simulation runs for a specified period of time.
- The results are not visible until this period of time has finished and the simulation has stopped.

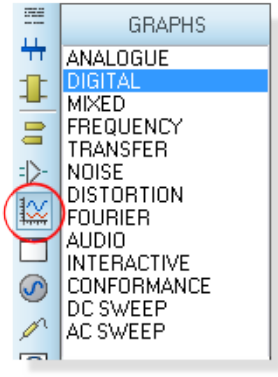
Given the above, let's look at how we might analyse the I2C traffic with a digital graph. First of all, delete the interactive I2C debugger by right clicking on it and selecting delete from the context menu.

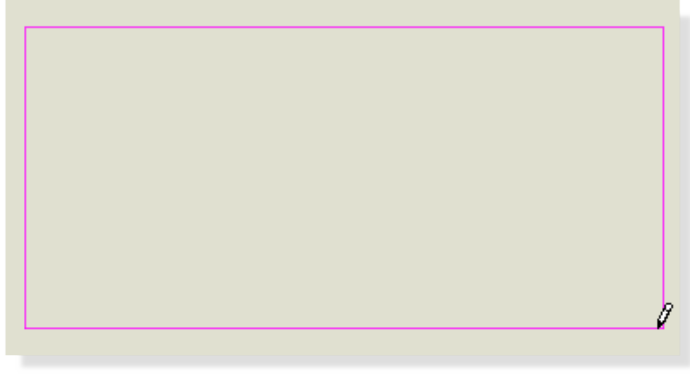


Next, make some space for the graph by dragging a tagbox around the display and wires with the right mouse button so that they all highlight. Left depress the mouse and drag the lot to the right of the schematic.



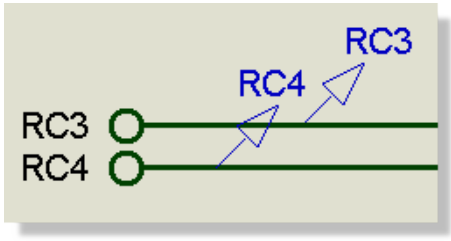
Now select the graph icon and then the digital graph from the parts bin. Left click the mouse at the top left of where you want the graph to be and then move the mouse down and to the right. Left click again to commit placement.



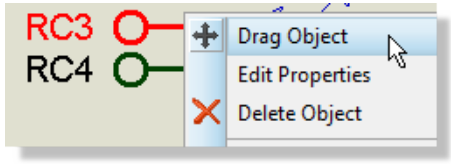


Note that the graph on the schematic doesn't need to be particularly large. We will maximise it later when we want to analyse the simulation results.

Having added the graph we now need to tell it which wires we want to trace. We do this by attaching probes to the wires and then specifying the probes as traces on the graph. Select the voltage probe icon and then place a probe on each of the two I2C lines. This is done exactly as we discussed earlier in the section on hardware breakpoints.

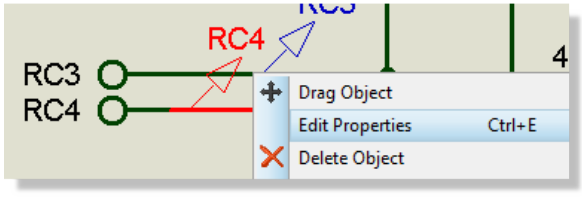


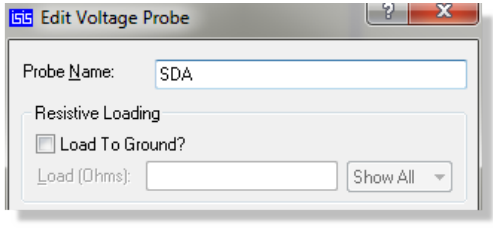
If you need to make some room for the probes, right click on the terminal at the end of the wire, right click and select drag object from the context menu and move to the left to make room for the probe.



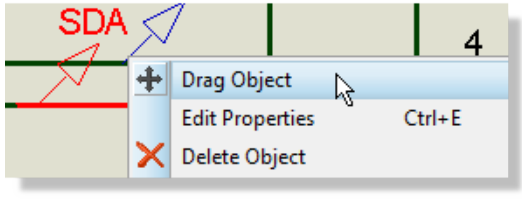
You can use the middle mouse button or the F6/F7 keys to zoom in and out around the mouse position if you need to get in closer. The F8 key will return to the default view of the schematic.

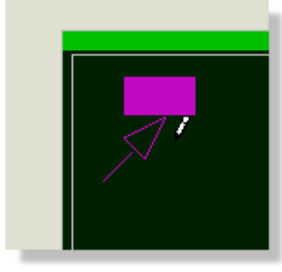
By default, the voltage probes will pick up the name of the terminal. It's useful to rename them to be more meaningful in the context of our simulation. You can do this by right clicking on the probe, editing its properties and then changing the name to be SCL or SDA respectively.



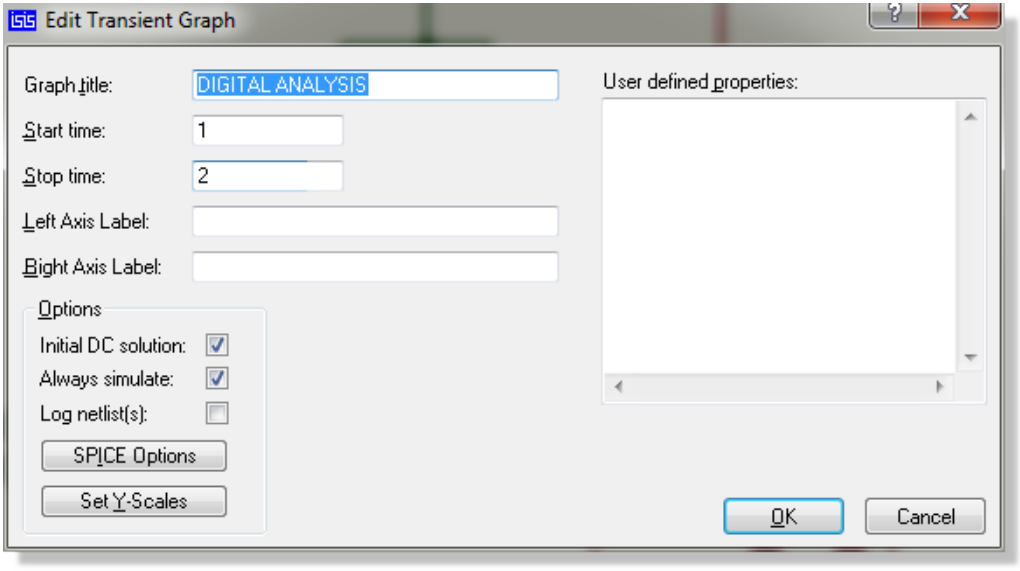


Now that we have the probes on the wires and the graph on the schematic we need to add the probes to the graph. There are several ways to do this but probably the easiest is simply to drag the probe onto the graph. To do this, right click on the probe, select drag object, move the mouse over the graph and then left click again to drop. Do this for both probes.



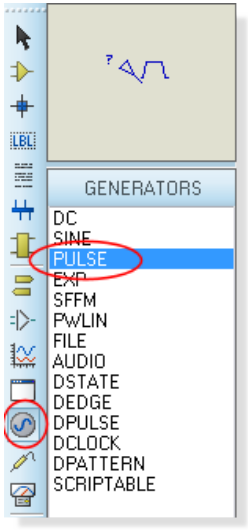


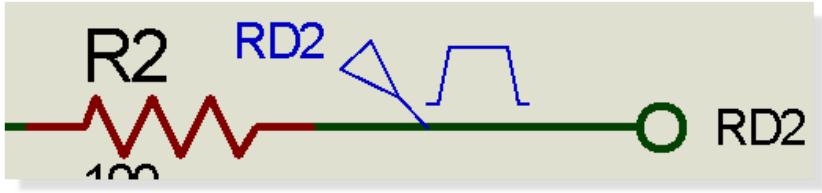
By default, a graph based simulation will run from time zero to one second of simulation time. In our case, it would make much more sense to run from one second to two seconds, thus skipping the initialization traffic. We can do this by editing the graph (right click and edit properties) and changing the start and stop time.



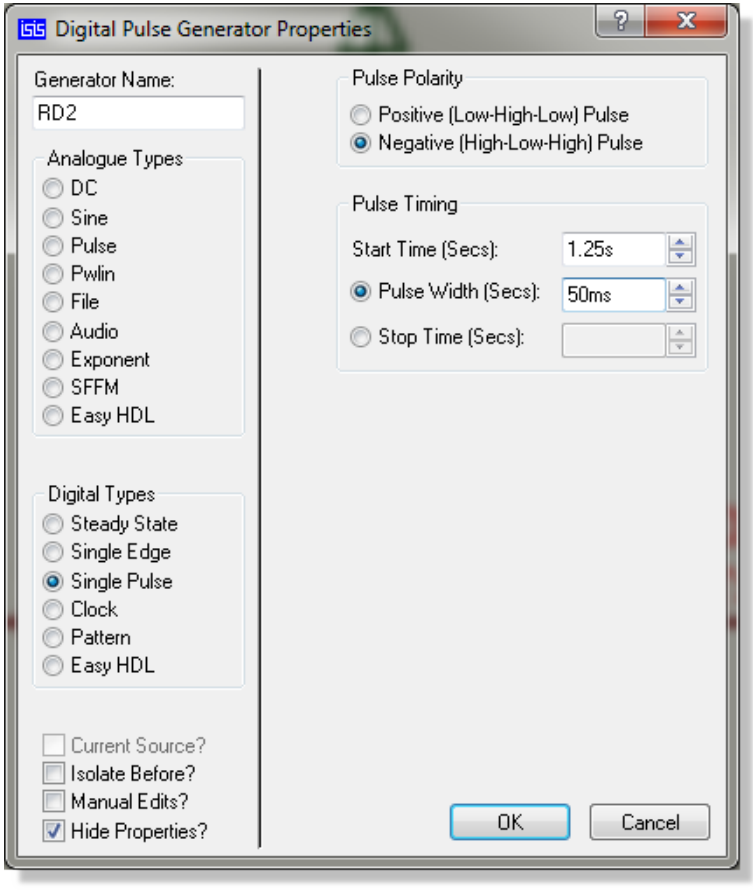
> Left clicking on an empty area of the schematic will de-select any currently selected objects. Normally, this is done when you exit an edit properties dialogue form as the object will still be highlighted.

We are now ready to simulate except for one significant problem. As mentioned previously, you cannot interact with a circuit during a graph based simulation but we need to switch mode in order to poll the temperature sensor and trigger the I2C traffic. Since we can't press the button we need to inject an equivalent signal to the PIC so that the firmware switches into temperature mode. We do this with generators. Select the generator icon and then the pulse generator from the parts bin. Place the generator to the right of the button beside the RD2 terminal - you may need to move the terminal first to make some space on the wire for the generator.

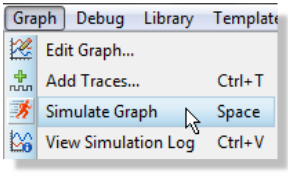


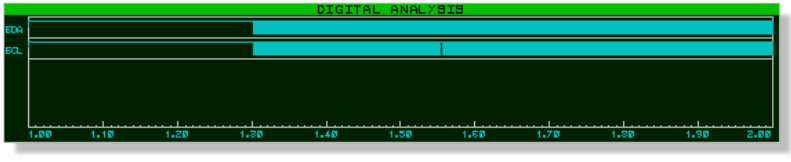


Now edit the generator in the usual way (right click - edit properties). Select single pulse from the digital generator types and change the pulse polarity to be negative so that the pulse signal represents a button push. Since we are simulating from 1s to 2s set the start time to 1.25s and the pulse width to something like 50ms.

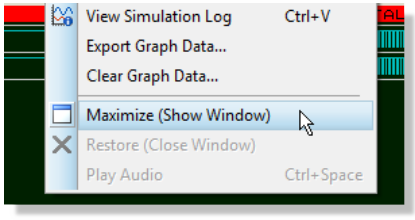


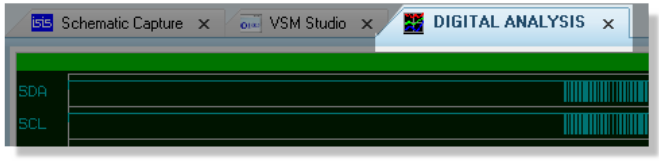
We can now run the simulation. Unlike an interactive simulation the animation control panel is now used (we are simulating for a fixed time which does not necessarily start at time zero). Instead, use the 'Simulate Graph' command on the graph menu. You should then see results which look like those below.



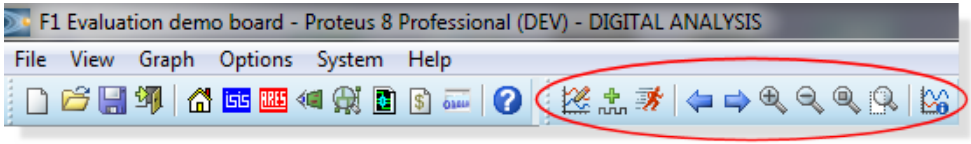


Aside from the fact that the traffic didn't start until around 1.25s (as specified by the pulse) we can't see much information here. Right click on the graph and select maximise from the context menu to launch the program in a new tab.

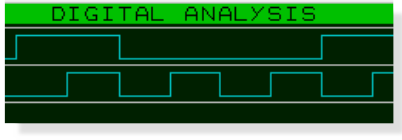




When the graph tab is in the foreground the menu's and icons change to show things we can do with the graph. In particular, notice the new graph toolbar at the top of the application



The zoom commands can be found here and probably the most useful of these is the 'zoom to area' icon. Click on this icon and then drag around a small area to localise the display. If you repeat this you will soon be able to zoom in on a single I2C sequence.

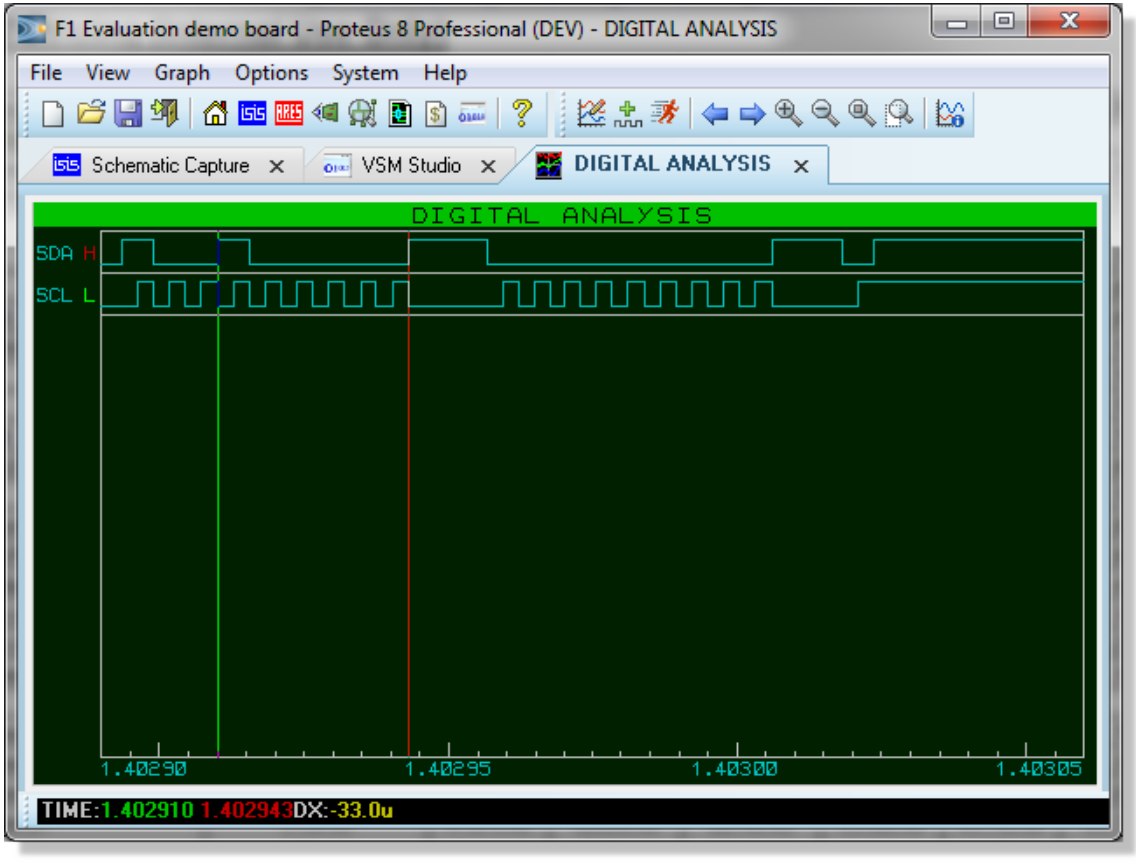


- Alternatively, you can hold the SHIFT button down and drag a box with the mouse to zoom into the area inside the box.

Those familiar with the I2C protocol can then verify the raw data on the graph against the sequencing we saw earlier on the protocol analyser.

Finally, you can take measurements with the graph as follows:

- Left clicking the mouse on the graph window will place a (green) cursor on the graph.
- Left dragging the mouse will allows you to position the cursor
- Holding the CTRL key down and left clicking will place a (red) cursor.
- Left dragging the mouse with the CTRL key down will allow you to position the (red) cursor.
- Data for each cursor and delta between them is shown on the display at the bottom of the graph window.



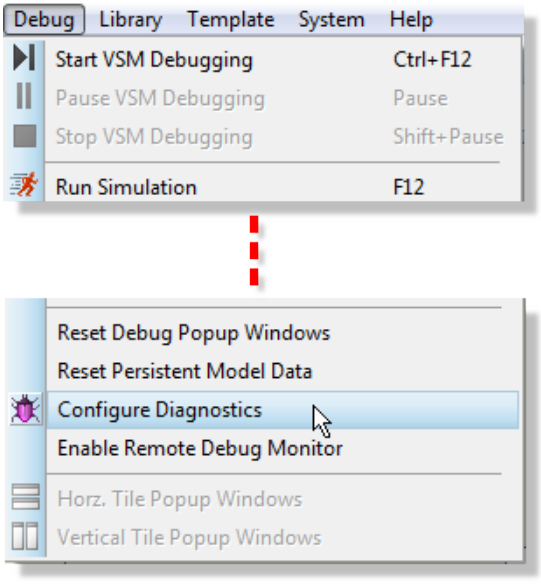
Further Reading

- All of the general placement and editing techniques discussed briefly here are covered in far more detail in the ISIS tutorial documentation.
- More information on Graphing is provided later in this documentation and a full discussion of generators, probes and graph based simulation can be found in the Proteus VSM reference manual.

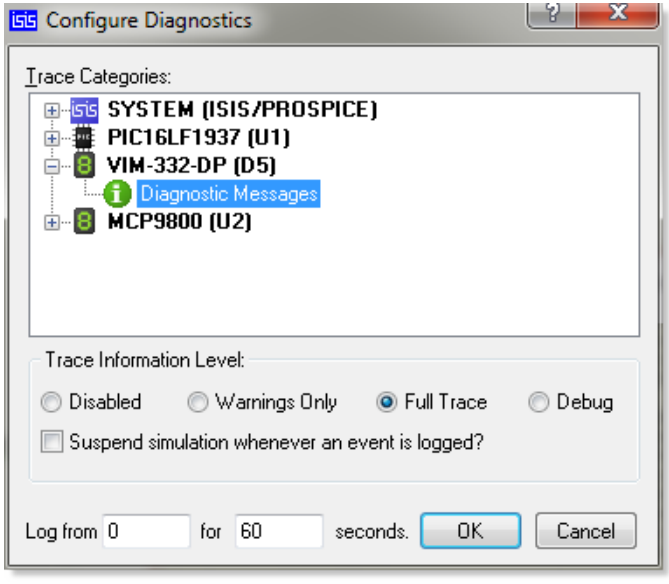
Diagnostic Messaging

Another - often overlooked - tool provided in Proteus is diagnostic messaging. This is an interface that allows you to ask the system for a plain text report of activity in a particular peripheral or peripherals. To look at this we will turn on diagnostic messaging for the LCD panel.

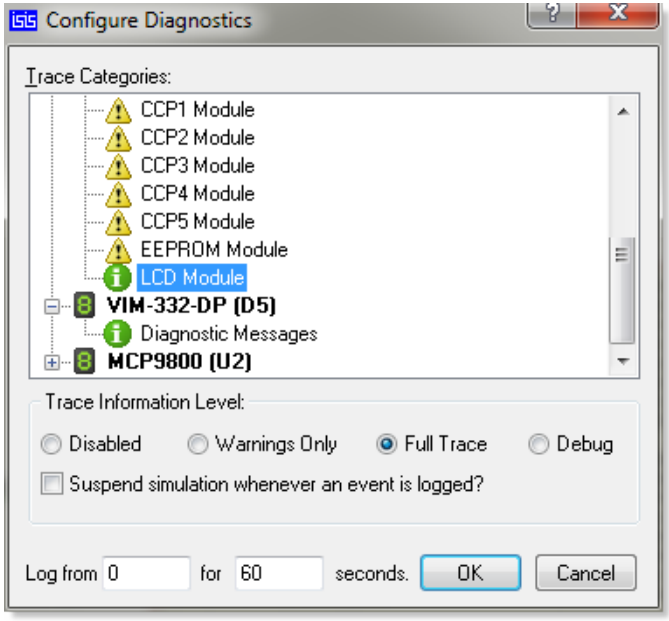
Switch to the schematic tab and then launch the configure diagnostics command from the Debug menu.



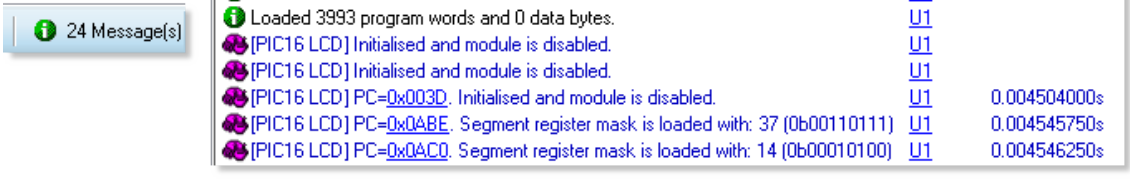
Expand the entry for the display panel, click on the entry and then change the trace information level to full trace.



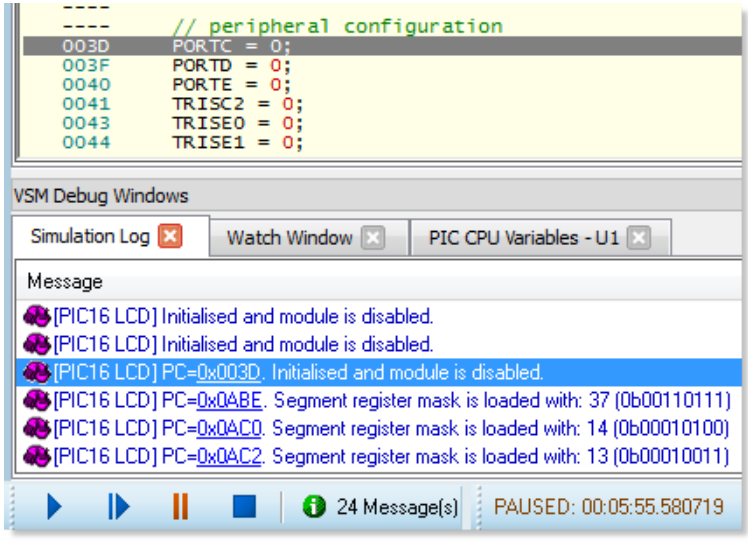
Now expand the PIC16 and scroll down to the LCD module entry, again changing the trace information level to 'full trace'.



Start an interactive simulation from the animation control panel and then click on the simulation advisor to view the diagnostic messages. This will switch Proteus to the VSM Studio tab and the simulation advisor will appear as a docked window at the bottom of the frame.



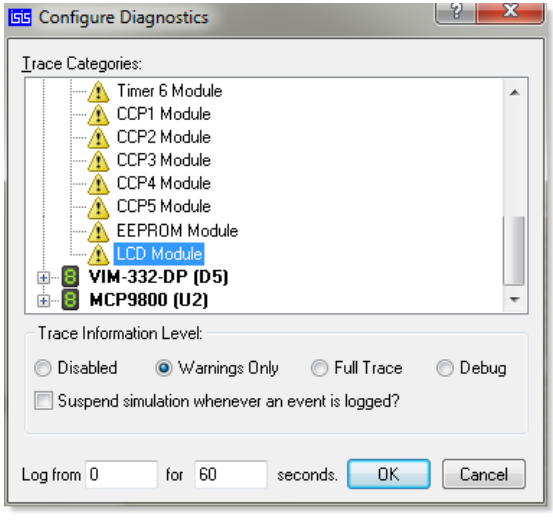
If you pause the simulation note that the LCD messages contain PC values which you can click on to locate the instruction in your source code that caused the condition.



Navigation to source code is only possible when the simulation is paused.

Finally, stop the simulation and turn off diagnostics in the same as they were enabled, except change the trace information levels back to warnings only.

Diagnostic messages are particularly useful in a problem situation where you need to gather information before deciding where to hunt for the bug. It does generate a major performance hit try it with the PIC MSSP peripheral and MCP9800 in temperature mode for example) and should therefore only be enabled when required. You can also choose to log over only a limited time period if you know when the activity band of interest is.



VSM TUTORIAL (Graphs)

The purpose of this tutorial is to show you, by use of a simple amplifier circuit, how to perform a graph based simulation using Proteus VSM. It takes you, in a step-by-step fashion, through:

- Placing graphs, probes and generators.
- Performing the actual simulation.
- Using graphs to display results and take measurements.
- A survey of the some of the analysis types that are available.

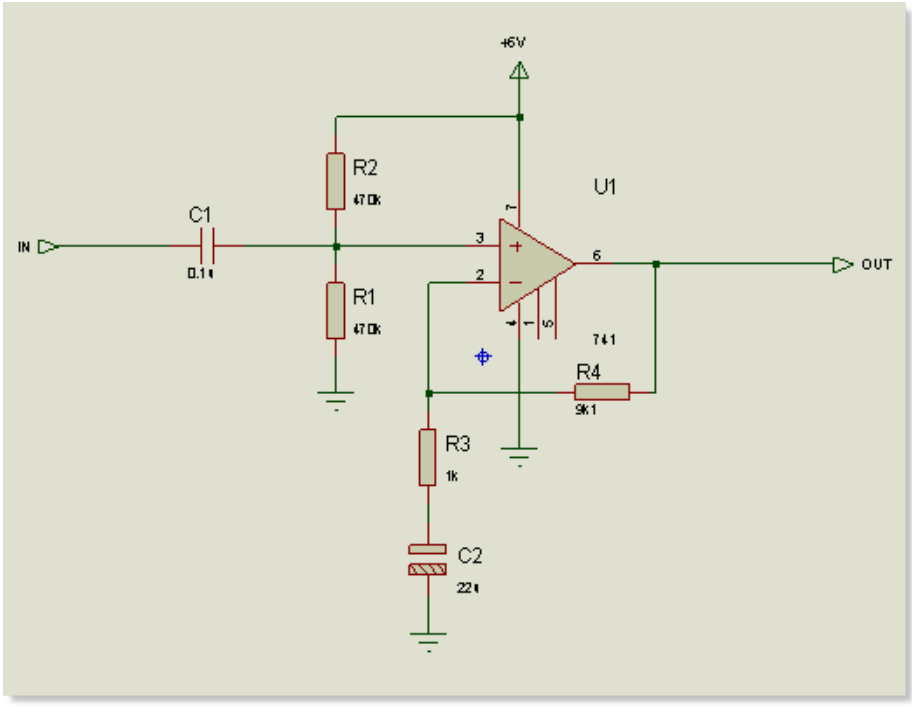
The tutorial does not cover the use of ISIS in its general sense - that is, procedures for placing components, wiring them up, tagging objects, etc. This is covered to some extent in the Interactive Simulation Tutorial and in much greater detail within the ISIS manual itself. If you have not already made yourself familiar with the use of ISIS then you must do so before attempting this tutorial.

We do strongly urge you to work right the way through this tutorial before you attempt to do your own graph based simulations: gaining a grasp of the concepts will make it much easier to absorb the material in the reference chapters and will save much time and frustration in the long term.

Getting Started

The circuit we are going to simulate is an audio amplifier based on a 741 op-amp, as shown below. It shows the 741 in an unusual configuration, running from a single 5 volt supply. The feedback resistors, R3 and R4, set the gain of the stage to be about 10. The input bias components, R1, R2 and C1, set a false ground reference at the non-inverting input which is decoupled from the signal input.

As is normally the case, we shall perform a transient analysis on the circuit. This form of analysis is the most useful, giving a large amount of information about the circuit. Having completed the description of simulation with transient analysis, the other forms of analysis will be contrasted.

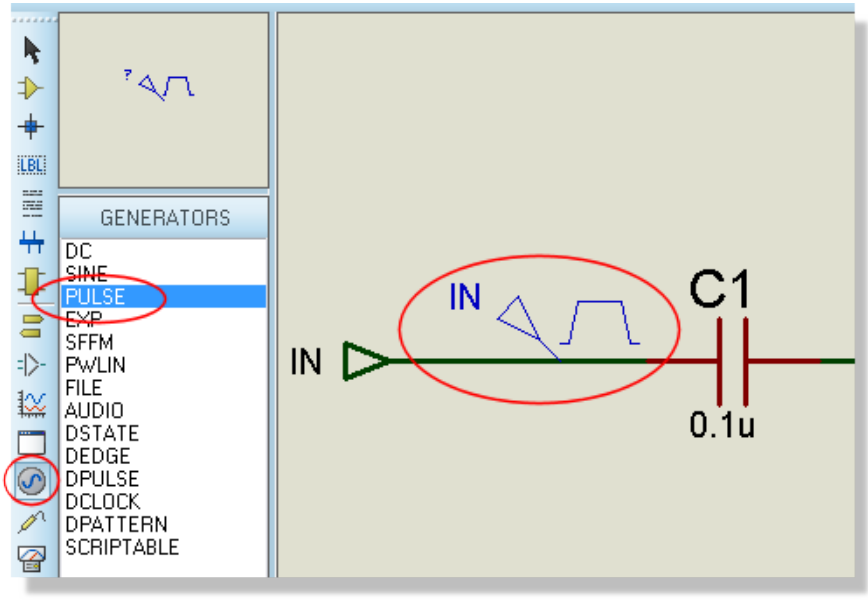


If you want, you can draw the circuit yourself, or you can load a ready made design file by File Menu ->Open Sample -> Tutorials -> Analogue Simulation Tutorial (Part 1) within your Proteus installation. Whatever you choose, at this point ensure you have ISIS running and the circuit drawn

Generators

To test the circuit, we need to provide it with a suitable input. We shall use a voltage source with a square wave output for our test signal. A generator object will be used to generate the required signal.

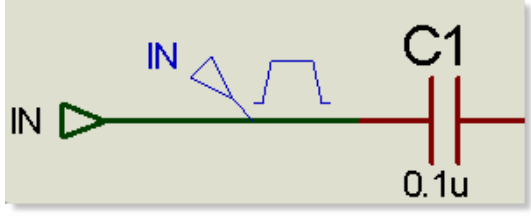
Click on the Generator icon: the Object Selector displays a list of the available generator types. For our simulation, we want a Pulse generator. Select the Pulse type, move the mouse over to the edit window, to the right of the IN terminal, and click left on the wire to place the generator.



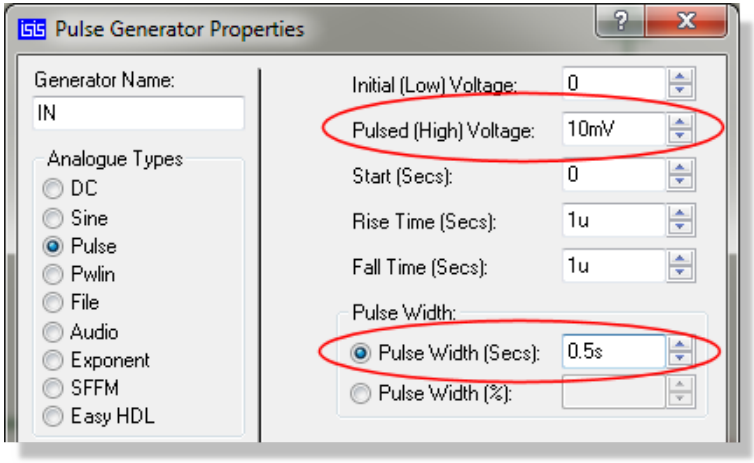
Generator objects are like most other objects in ISIS; the same procedures for previewing and orienting the generator before placement and editing, moving, re-orienting or deleting the object after placement apply (see Generators and Probes in the reference manual).

As well as being dropped onto an existing wire, as we just did, generators may be placed on the sheet, and wired up in the normal manner. If you drag a generator off a wire, then ISIS assumes you want to detach it, and does not drag the wire along with it, as it would do for components.

Notice how the generator is automatically assigned a reference - the terminal name IN. Whenever a generator is wired up to an object (or placed directly on an existing wire) it is assigned the name of the net to which it is connected. If the net does not have a name, then the name of the nearest component pin is used by default.



Finally, we must edit the generator to define the pulse shape that we want. To edit the generator, right click on it and select Edit Properties from the resulting context menu. Select the High Voltage field and set the value to 10mV. Also set the pulse width to 0.5s.



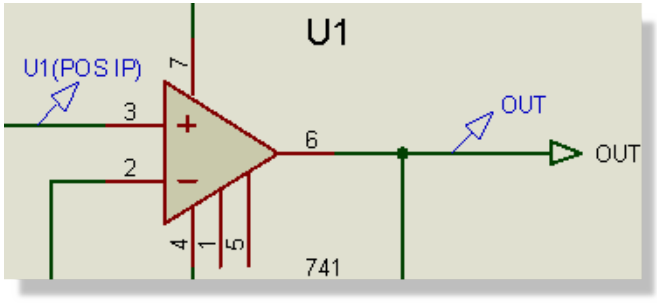
Select the OK button to accept the changes.

> For this circuit only one generator is needed, but there is no limit on the number which may be placed.

Probes

Having defined the input to our circuit using a generator, we must now place probes at the points we wish to monitor. We are obviously interested in the output, and the input after it has been biased is also a useful point to probe. If needs be, more probes can always be added at key points and the simulation repeated.

To place a probe, click left on the Voltage Probe icon (ensure you have not selected a current probe by accident - we shall come to these later). Probes can be placed onto wires, or placed and then wired, in the same manner as generators. Move the mouse over to the edit window, to the left of U1 pin 3, and click left to place the probe on the wire joining pin 3 to R1 and R2. Be sure to place the probe on the wire, as it cannot be placed on the pin itself. Notice the name it acquires is the name of the nearest device to which it is connected, with the pin name in brackets. Now place the second probe by clicking left just to the left of the OUT terminal, on the wire between the junction dot and the terminal pin.



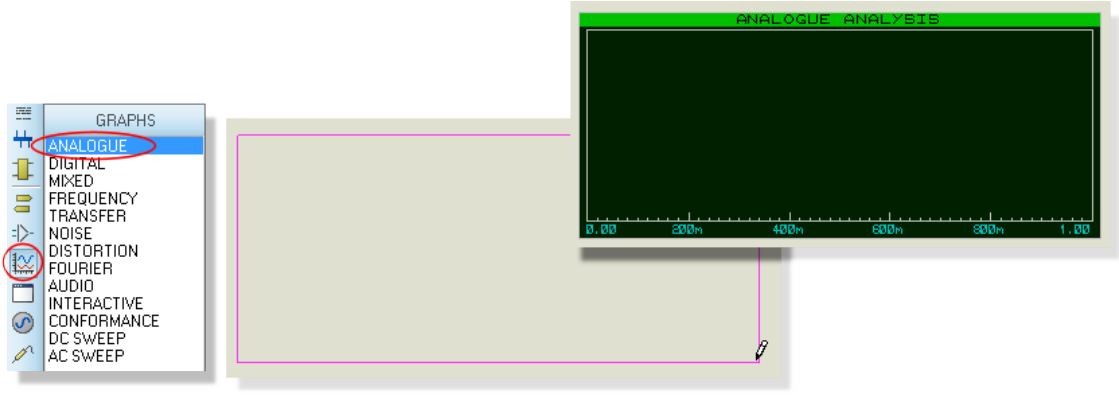
Probe objects are like generators and most other objects in ISIS; the same procedures for previewing and orienting the probe before placement, and editing, moving, re-orienting or deleting the probe after placement apply (refer to the section on Probes in the reference manual for more information). Probes may be edited in order to change their reference labels. The names assigned by default are fine in our case, but a useful tip when tagging probes is to aim for the tip of the probe, not the body or reference label.

Now that we have set up the circuit ready for simulation, we need to place a graph to display the results on.

Graphs

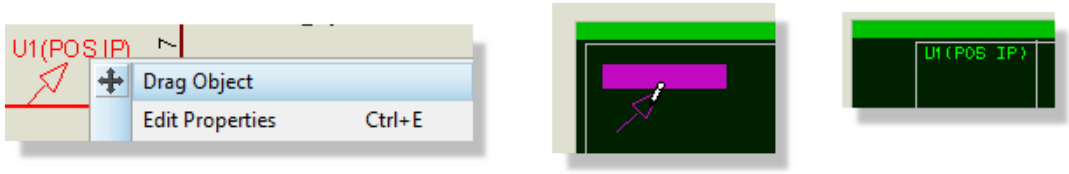
Graphs play an important part in simulation: they not only act as a display medium for results but actually define what simulations are carried out. By placing one or more graphs and indicating what sort of data you expect to see on the graph (digital, voltage, impedance, etc.) ISIS knows what type or types of simulations to perform and which parts of a circuit need to be included in the simulation. For a transient analysis we need an Analogue type graph. It is termed analogue rather than transient in order to distinguish it from the Digital graph type, which is used to display results from a digital analysis, which is really a specialised form of transient analysis. Both can be displayed against the same time axis using a Mixed graph.

To place a graph, first select the Graph icon: the Object Selector displays a list of the available graph types. Select the Analogue type, move the mouse over to the edit window, click once to start placement, drag out a rectangle of the appropriate size and click a second time to place the graph.



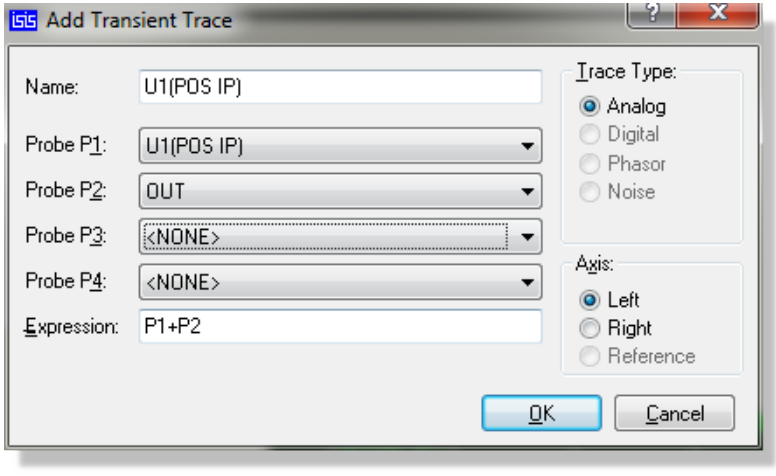
Graphs behave like most objects in ISIS, though they do have a few subtleties. We will cover the features pertinent to the tutorial as they occur, but the reference chapter on graphs is well worth a read. You can tag a graph in the usual way with the left mouse button, and then (using the left mouse button) drag one of the handles, or the graph as a whole, about to resize and/or reposition the graph.

We now need to add our generator and probes on to the graph. Each generator has a probe associated with it, so there is no need to place probes directly on generators to see the input wave forms. There are three ways of adding probes and generators to graphs:
The first method is to tag a probe/generator and drag it over the graph and release it - exactly as if we were repositioning the object, repeat for each probe/generator. ISIS detects that you are trying to place the probe/generator over the graph, restores the probe/generator to its original position, and adds a trace to the graph with the same reference as that of the probe/generator. Traces may be associated with the left or right axes in an analogue graph, and probes/generators will add to the axis nearest the side they were dropped. Regardless of where you drop the probe/generator, the new trace is always added at the bottom of any existing traces.

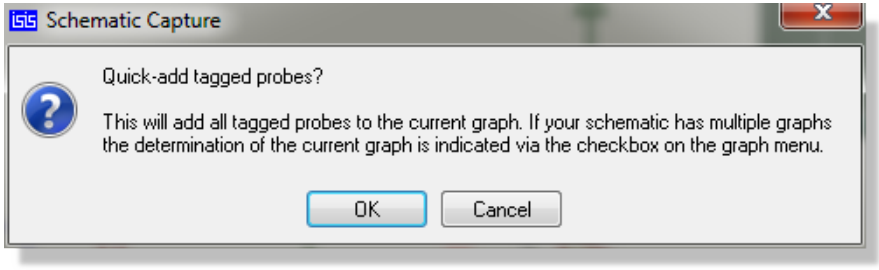


The second and third method of adding probes/generators to a graph both use the Add Trace command on the graph menu; this command always adds probes to the current graph (when there is more than one graph, the current graph is the one currently selected on the Graph menu).

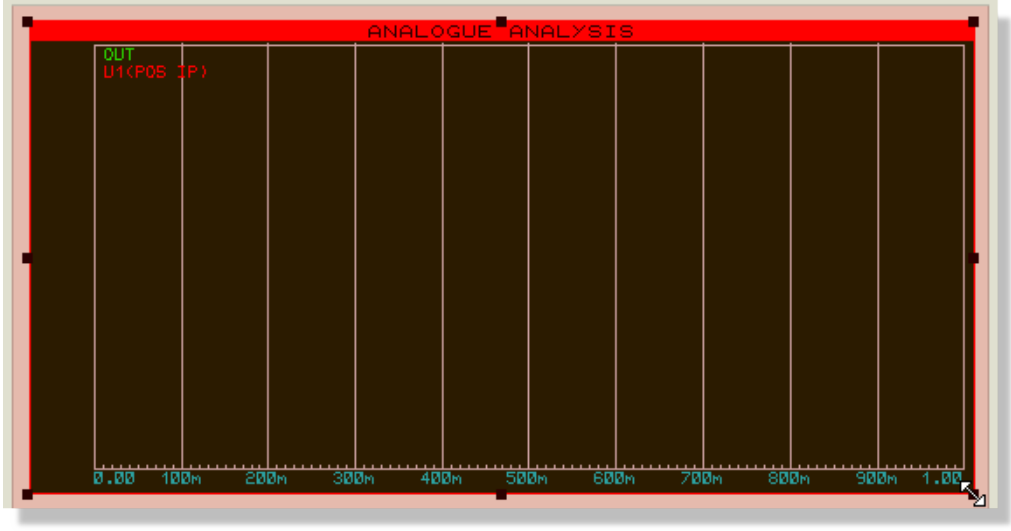
If the Add Trace command is invoked without any tagged probes or generators, then the Add Transient Trace dialogue form is displayed, and a probe can be selected from a list of all probes in the design (including probes on other schematic sheets).



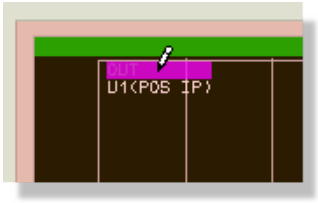
If there are tagged probes/generators, invoking the Add Trace command causes you to be prompted to Quick Add the tagged probes to the current graph; selecting the 'Cancel' option invokes the Add Transient Trace dialogue form as previously described. Selecting the 'OK' option adds all tagged probes/generators to the current graph in alphabetical order.



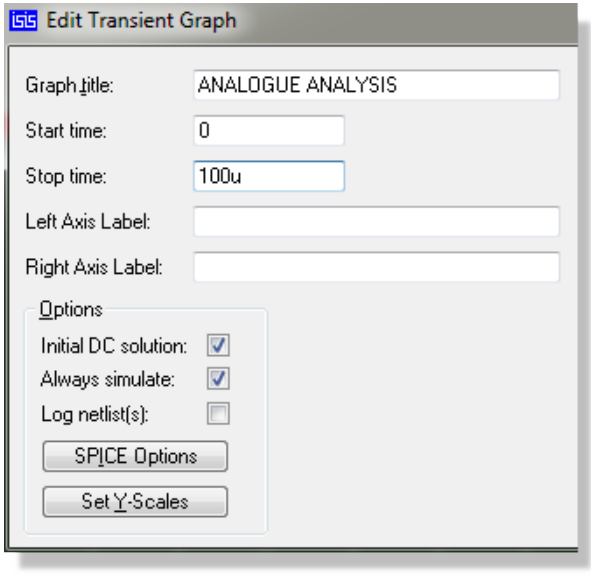
We will Quick Add our probes and the generator to the graph. Either tag the probes and generators individually, or, more quickly, drag a tag box around the entire circuit - the Quick Add feature will ignore all tagged objects other than probes and generators. Select the Add Trace option from the Graph menu and answer Yes to the prompt. The traces will appear on the graph (because there is only one graph, and it was the last used, it is deemed to be the current graph). At the moment, the traces consist of a name (on the left of the axis), and an empty data area (the main body of the graph). If the traces do not appear on the graph, then it is probably too small for ISIS to draw them in. Resize the graph, by tagging it and dragging a corner, to make it sufficiently big.



As it happens, our traces (having been placed in alphabetical order) have appeared in a reasonable order. We can however, shuffle the traces about. To do this, ensure the graph is not tagged, and click left over the name of a trace you want to move or edit. The trace is highlighted to show that it is tagged. You can now use the left mouse button to drag the trace up or down or to edit the trace (by clicking left without moving the mouse) and the right button to delete the trace (i.e. remove it from the graph). To untag all traces, click the left mouse button anywhere over the graph, but not over a trace label (this would tag or delete the trace).



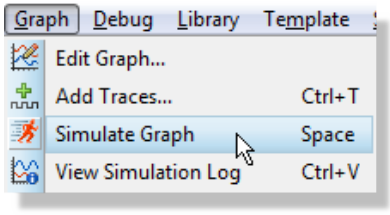
There is one final piece of setting-up to be done before we start the simulation, and this is to set the simulation run time. ISIS will simulate the circuit according to the end time on the x-scale of the graph, and for a new graph, this defaults to one second. For our purposes, we want the input square wave to be of fairly high audio frequency, say about 10kHz. This needs a total period of 100ms. Tag the graph and click left on it to bring up its Edit Transient Graph dialogue form. This form has fields that allow you to title the graph, specify its simulation start and stop times (these correspond to the left and right most values of the x axis), label the left and right axes (these are not displayed on Digital graphs) and also specifies general properties for the simulation run. All we need to change is the stop time from 1.00 down to 100u (you can literally type in 100u - ISIS will covert this to 100E-6) and select OK.

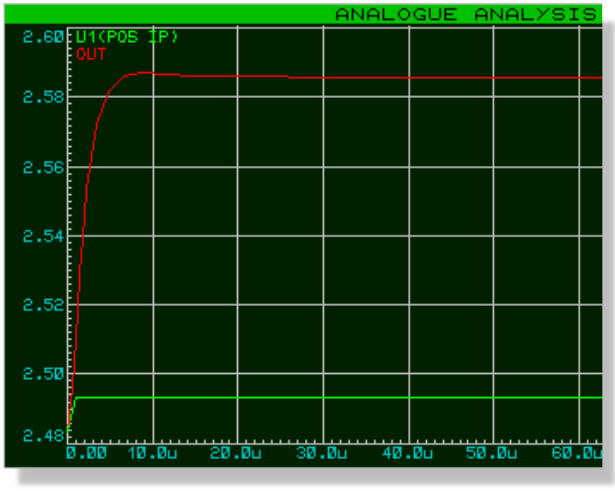


The design is now ready for simulation. At this point, it is probably worthwhile loading our version of the design (Open Sample -> Tutorials -> Analogue Simulation Tutorial (Part 2)) to avoid any problems during the actual simulation and subsequent sections. Alternatively, you may wish to continue with the circuit you have entered yourself, and only load the default file if problems arise.

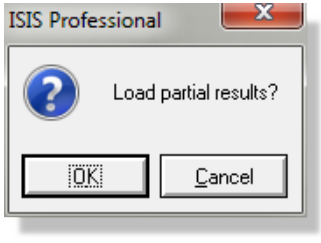
Simulation

To simulate the circuit, all you need do is invoke the Simulate command on the Graph menu (or use its keyboard short-cut: the space bar). The Simulate command causes the circuit to be simulated and the current graph (the one marked on the Graph menu) to be updated with the simulation results.





Do this now. The status bar indicates how far the simulation process has reached. When the simulation is complete, the graph is redrawn with the new data. For the current release of ISIS and simulator kernels, the start time of a graph is ignored - simulation always starts at time zero and runs until the stop time is reached or until the simulator reaches a quiescent state. You can abort a simulation mid-way through by pressing the ESC key.

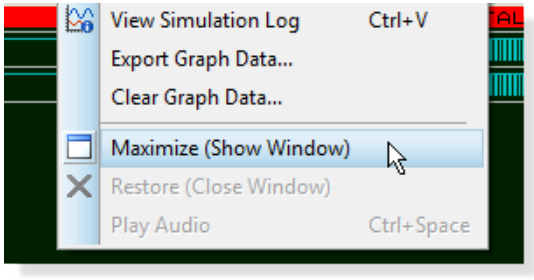


A simulation log is maintained for the last simulation performed. You can view this log by clicking on the simulation advisor at the bottom left of the schematic frame. The simulation log of an analogue simulation rarely makes for exciting reading, unless warnings or errors were reported, in which case it is where you will find details of exactly what went wrong. In some cases, however, the simulation log provides useful information that is not easily available from the graph traces.

So the first simulation is complete. Looking at the traces on the graph, it's hard to see any detail. To check that the circuit is working as expected, we need to take some measurements.

Taking Measurements

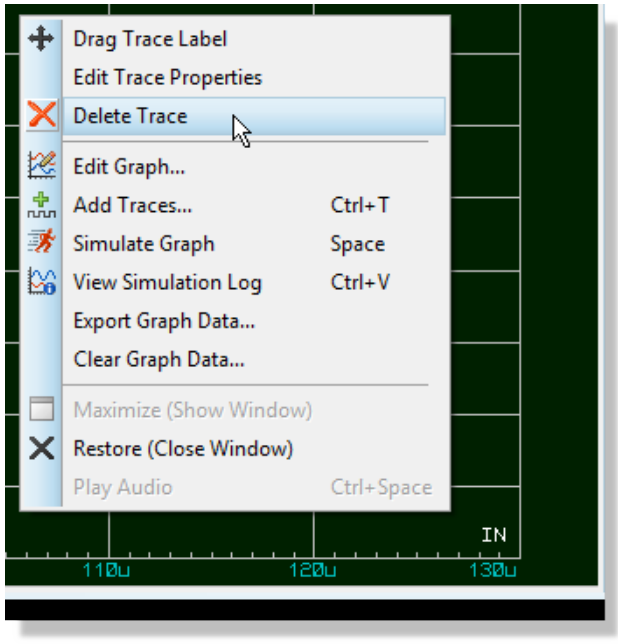
A graph sitting on the schematic, alongside a circuit, is referred to as being minimised. To take timing measurements we must first maximise the graph. To do this, right click on the graph and select the maximise option from the resulting context menu.



This will open the graph in a new tab and change the menus and icons to include functionality applicable to the graph.

On the left side of the screen is an area in which the trace labels are displayed and to right of this are the traces themselves. When cursors are placed the data and delta values are displayed at the bottom of the graph. As this is a new graph, and we have not yet taken any measurements, there are no cursors visible on the graph, and the status bar simply displays a title message.

The traces are colour coded, to match their respective labels. The OUT and U1(POS IP) traces are clustered at the top of the display, whilst the IN trace lies along the bottom. To see the traces in more detail, we need to separate the IN trace from the other two. This can be achieved by using the left mouse button to drag the trace label to the right-hand side of the screen. This causes the right y-axis to appear, which is scaled separately from the left. The IN trace now seems much larger, but this is because ISIS has chosen a finer scaling for the right axis than the left. To clarify the graph, it is perhaps best to remove the IN trace altogether, as the U1(POS IP) is just as useful. Click right on the IN label and select Delete Trace. The graph now reverts to a single, left hand side, y-axis.



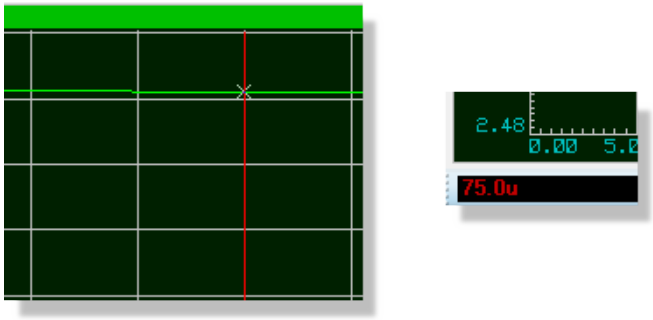
We shall measure two quantities:

- The voltage gain of the circuit.
- The approximate fall time of the output.

These measurements are taken using the Cursors.

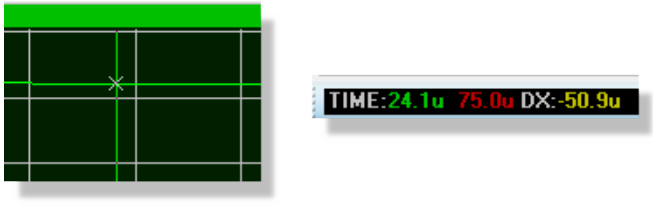
Each graph has two cursors, referred to as the Reference and Primary cursors. The reference cursor is displayed in red, and the primary in green. A cursor is always 'locked' to a trace, the trace a cursor is locked to being indicated by a small 'X' which 'rides' the waveform. A small mark on both the x- and y-axes will follow the position of the 'X' as it moves in order to facilitate accurate reading of the axes. If moved using the keyboard, a cursor will move to the next small division in the x-axis.
Let us start by placing the Reference cursor. The same keys/actions are used to access both the Reference and Primary cursors. Which is actually affected is selected by use of the CTRL key on the keyboard; the Reference cursor, as it is the least used of the two, is always accessed with the CTRL key (on the keyboard) pressed down. To place a cursor, all you need to do is point at the trace data (not the trace label - this is used for another purpose) you want to lock the cursor to, and click left. If the CTRL key is down, you will place (or move) the Reference cursor; if the CTRL key is not down, then you will place (or move) the Primary cursor. Whilst the mouse button (and the CTRL key for the Reference cursor) is held down, you can drag the cursor about. So, hold down (and keep down) the CTRL key, move the mouse pointer to the right hand side of the graph, above both traces, and press the left mouse button. The red Reference cursor appears. Drag the cursor (still with the CTRL key down) to about 70u or 80u

on the x-axis. The title on the status bar is removed, and will now display the cursor time (in red, at the left) and the cursor voltage along with the name of the trace in question (at the right). It is the U1(POS IP) trace that we want.

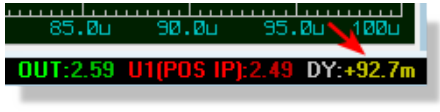


You can move a cursor in the X direction using the left and right cursor keys, and you can lock a cursor to the previous or next trace using the up and down cursor keys. The LEFT and RIGHT keys move the cursor to the left or right limits of the x-axis respectively. With the control key still down, try pressing the left and right arrow keys on the keyboard to move the Reference cursor along small divisions on the time axis.

Now place the Primary cursor on the OUT trace between 20u and 30u. The procedure is exactly the same as for the Reference cursor, above, except that you do not need to hold the CTRL key down. The time and the voltage (in green) for the primary cursor are now added to the status bar.



Also displayed are the differences in both time and voltage between the positions of the two cursors. The voltage difference should be a fraction under 100mV. The input pulse was 10mV high, so the amplifier has a voltage gain of 10. Note that the value is positive because the Primary cursor is above the Reference cursor - in delta read-outs the value is Primary minus Reference.



We can also measure the fall time using the relative time value by positioning the cursors either side of the falling edge of the output pulse. This may be done either by dragging with the mouse, or by using the cursor keys (don't forget the CTRL key for the Reference cursor). The Primary cursor should be just to the right of the curve, as it straightens out, and the Reference

cursor should be at the corner at the start of the falling edge. You should find that the falling edge is a little under 10m.

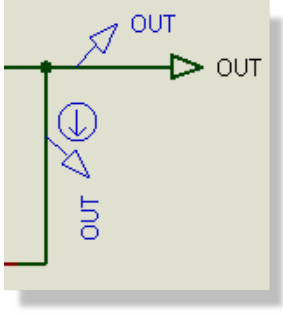
Using Current Probes

Now that we have finished with our measurements, we can return to the circuit – just close the maximised graph for the cross at the right of the tab.

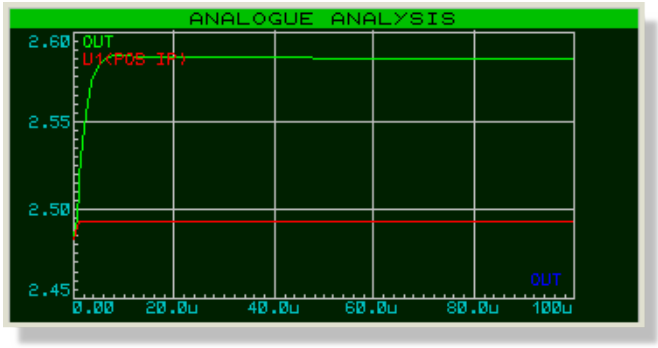
We shall now use a current probe to examine the current flow around the feedback path, by measuring the current into R4.

Current probes are used in a similar manner to voltage probes, but with one important difference. A current probe needs to have a direction associated with it. Current probes work by effectively breaking a wire, and inserting themselves in the gap, so they need to know which way round to go. This is done simply by the way they are placed. In the default orientation (leaning to the right) a current probe measures current flow in a horizontal wire from left to right. To measure current flow in a vertical wire, the probe needs to be rotated through 90° or 270°. Placing a probe at the wrong angle is an error, that will be reported when the simulation is executed. If in doubt, look at the arrow in the symbol. This points in the direction of current flow.

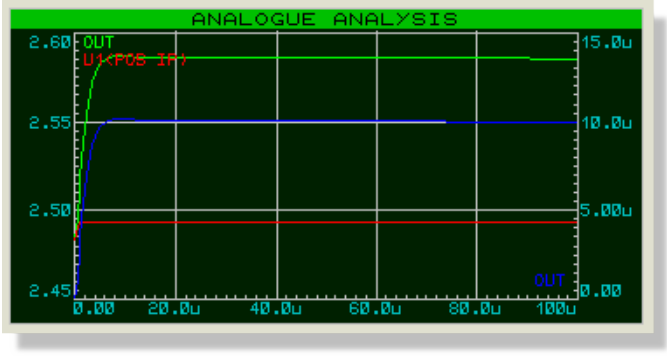
Select a current probe by clicking on the Current Probe icon. Click left on the clockwise Rotation icon such that the arrow points downwards. Then place the probe on the vertical wire between the right hand side of R4 and U1 pin 6.



Add the probe to the right hand side of the graph by tagging and dragging onto the right hand side of the minimised graph. The right side is a good choice for current probes because they are normally on a scale several orders of magnitude different than the voltage probes, so a separate axis is needed to display them in detail.



At the moment no trace is drawn for the current probe. Press the space bar to re-simulate the graph, and the trace appears.

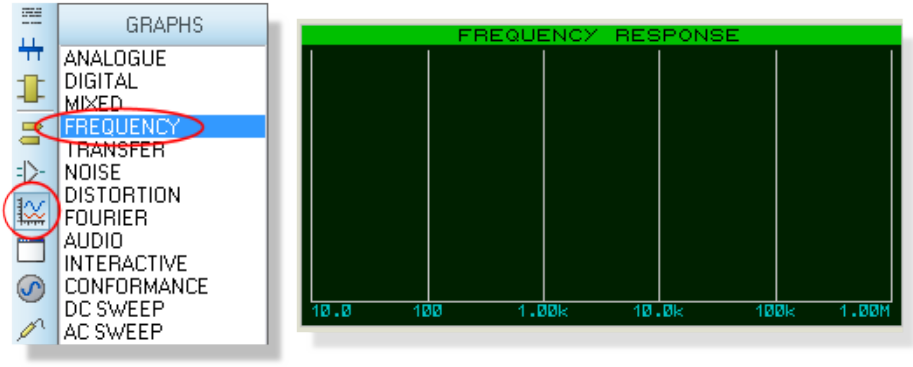


Even from the minimised graph, we can see that the current in the feedback loop follows closely the wave form at the output, as you would expect for an op-amp. The current changes between 10mA and 0 at the high and low parts of the trace respectively. If you wish, the graph may be maximised to examine the trace more closely.

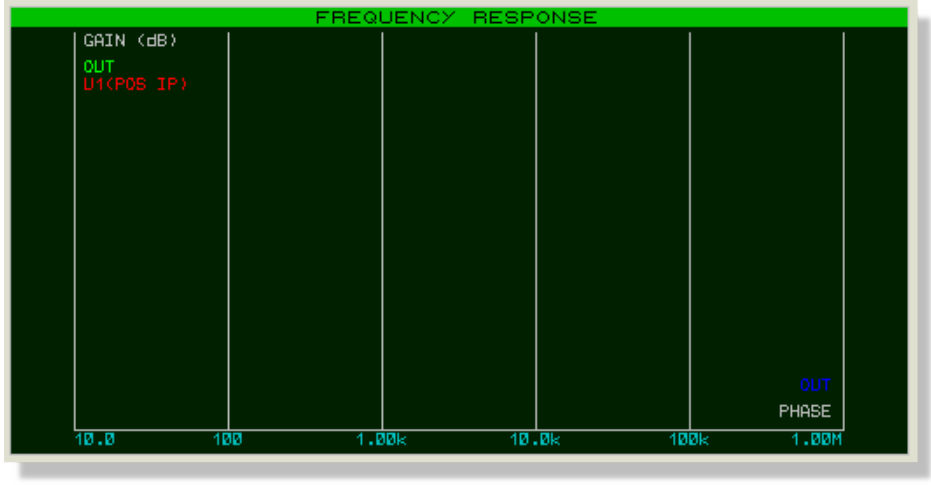
Frequency Analysis

As well as transient analysis, there are several other analysis types available in analogue circuit simulation. They are all used in much the same way, with graphs, probes and generators, but they are all different variations on this theme. The next type of analysis that we shall consider is Frequency analysis. In frequency analysis, the x-axis becomes frequency (on a logarithmic scale), and both magnitude and phase of probed points may be displayed on the y-axes.

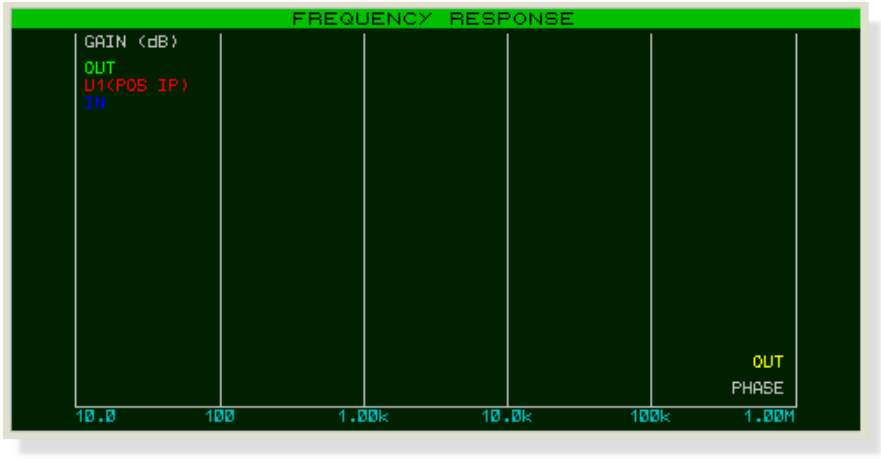
To perform a frequency analysis a Frequency graph is required. Click left on the Graph icon, to re-display the list of graph types in the object selector, and click on the Frequency graph type. Then place a graph on the schematic as before, dragging a box with the left mouse button. There is no need to remove the existing transient graph, but you may wish to do so in order to create some more space (right click and delete object from the resulting context menu).

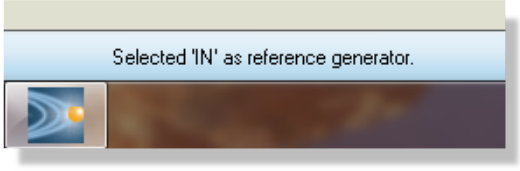


Now to add the probes. We shall add both the voltage probes, OUT and U1(POS IP). In a frequency graph, the two y-axes (left and right) have special meanings. The left y-axis is used to display the magnitude of the probed signal, and the right y-axis the phase. In order to see both, we must add the probes to both sides of the graph. Tag and drag the OUT probe onto the left of the graph, then again drag the robe onto the right. Each trace has a separate colour as normal, but they both have the same name. Now tag and drag the U1(POS IP) probe onto the left side of the graph only.

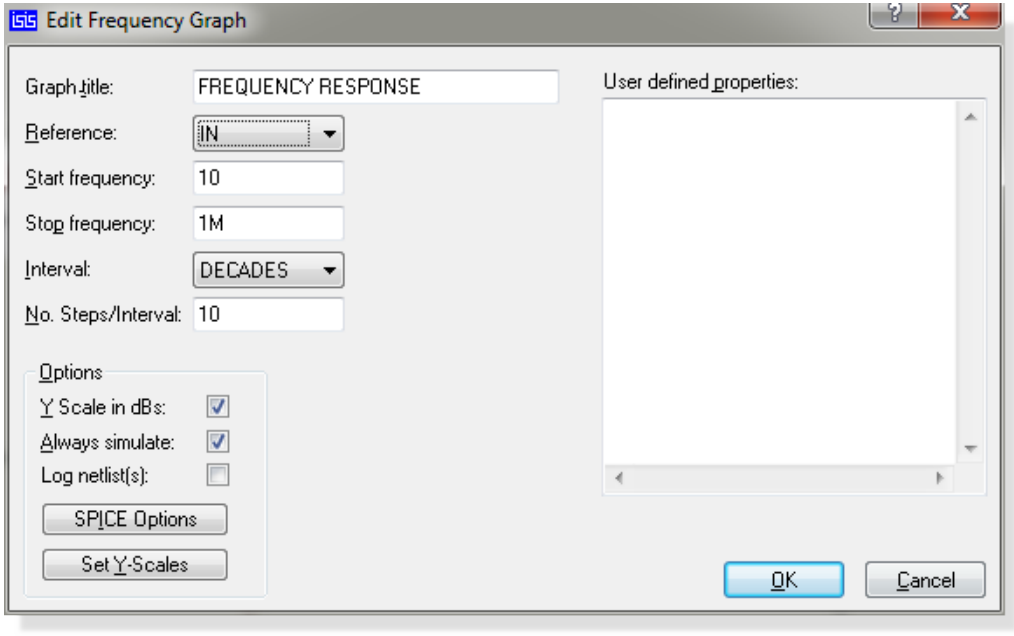


Magnitude and phase values must both be specified with respect to some reference quantity. In ISIS this is done by specifying a Reference Generator. A reference generator always has an output of 0dB (1 volt) at 0°. Any existing generator may be specified as the reference generator. All of the other generators in the circuit are ignored in a frequency analysis. To specify the IN generator as the reference in our circuit, simply tag and drag it onto the graph, as if you were adding it as a probe. ISIS assumes that, because it is a generator, you are adding it as the reference generator, and prints a message on status line confirming this. Make sure you have done this, or the simulation will not work correctly.



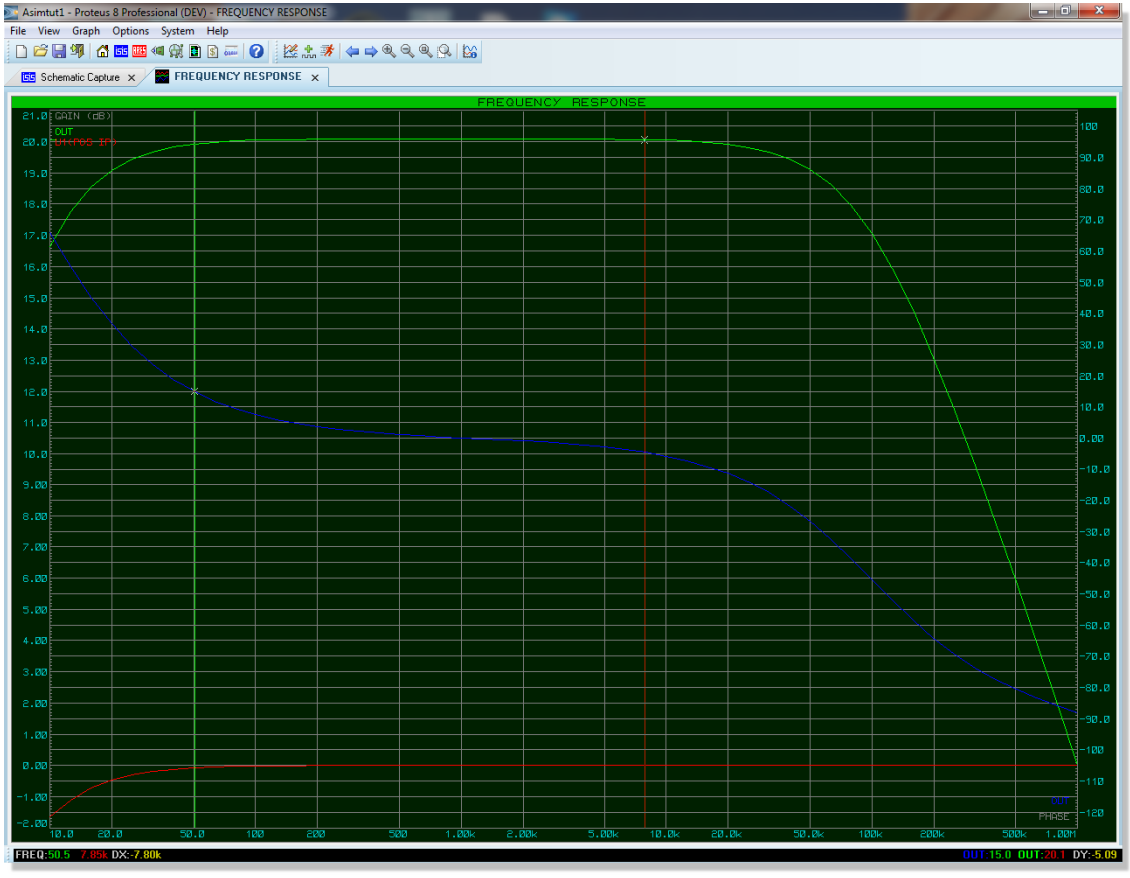


There is no need to edit the graph properties, as the frequency range chosen by default is fine for our purposes. However, if you do so (right click on the graph and select Edit properties), you will see that the Edit Frequency Graph dialogue form is slightly different from the transient case. There is no need to label the axes, as their purpose is fixed, and there is a check box which enables the display of magnitude plots in dB or normal units. This option is best left set to dB, as the absolute values displayed otherwise will not be the actual values present in the circuit.



Now start the simulation by right clicking on the graph and selecting the simulate option from the context menu. When it has finished, maximise the graph by opening it in a new tab as before (right click context menu).

Considering first the OUT magnitude trace, we can see the pass-band gain is just over 20dB (as expected), and the useable frequency range is about 50Hz to 20kHz. The cursors work in exactly the same manner as before - you may like to use the cursors to verify the above statement.



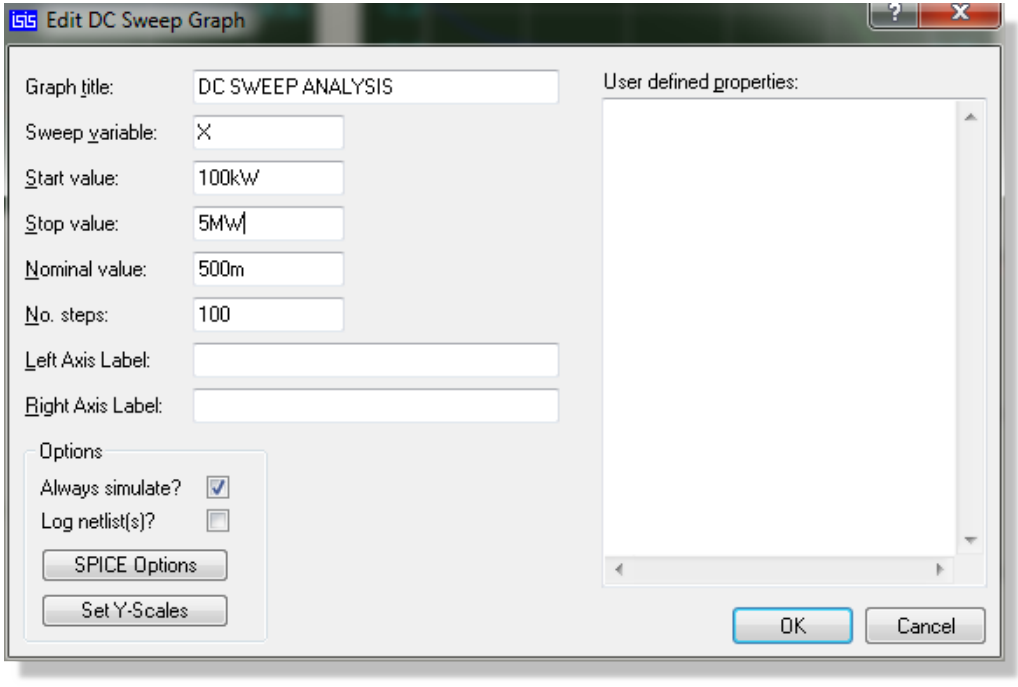
The OUT phase trace shows the expected phase distortion at the extremes of the response, dropping to -90° just off the right of the graph, at the unity gain frequency. The high-pass filter effect of the input bias circuitry can be clearly seen if the U1(POS IP) magnitude trace is examined. Notice that the x-axis scale is logarithmic, and to read values from the axis it is best to use the cursors.

Swept Variable Analysis

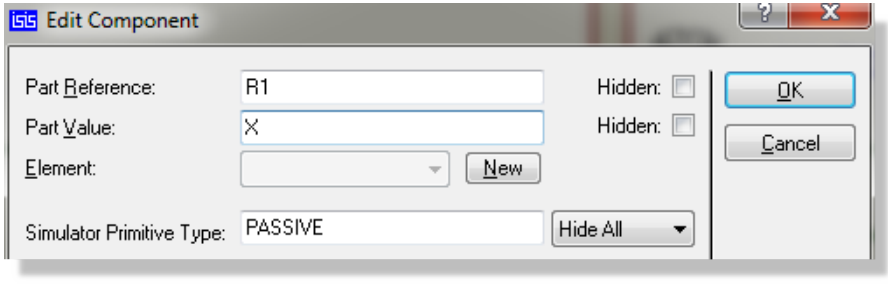
It is possible with ISIS to see how the circuit is affected by changing some of the circuit parameters. There are two analysis types that enable you to do this – the DC Sweep and the AC Sweep. A DC Sweep graph displays a series of operating point values against the swept variable, while an AC Sweep graph displays a series of single point frequency analysis values, in magnitude and phase form like the Frequency graph.

As these forms of analysis are similar, we shall consider just one - a DC Sweep. The input bias resistors, R1 and R2, are affected by the small current that flows into U1. To see how altering the value of both of these resistors affects the bias point, a DC Sweep is used.

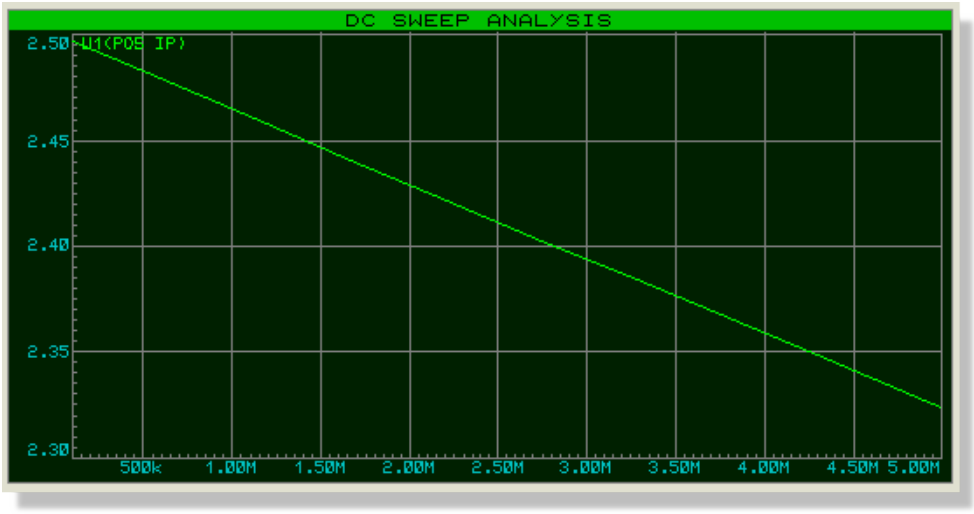
To begin with place a DC Sweep graph on an unused space on the schematic. Then tag the U1(POS IP) probe and drag it onto the left of the graph. We need to set the sweep value, and this is done by editing the graph (right click - edit properties). The Edit DC Sweep Graph dialogue form includes fields to set the swept variable name, its start and ending values, and the number of steps taken in the sweep. We want to sweep the resistor values across a range of say 100kW to 5MW, so set the Start field to 100k and the Stop field to 5M. Click on OK to accept the changes.



Of course, the resistors R1 and R2 need to be altered to make them swept, rather than the fixed values they already are. To do this, double left click on R1 to edit it, and alter the Value field from 470k to X. Note that the swept variable in the graph dialogue form was left at X as well. Click on OK, and repeat the editing on R2 to set its value to X.



Now you can simulate the graph by right clicking on it and selecting the simulate option. Then, by maximising the graph, you can see that the bias level reduces as the resistance of the bias chain increases. By 5MW it is significantly altered.



Of course, altering these resistors will also have an effect on the frequency response. We could have done an AC Sweep analysis at say 50Hz in order to see the effect on low frequencies.

Noise Analysis

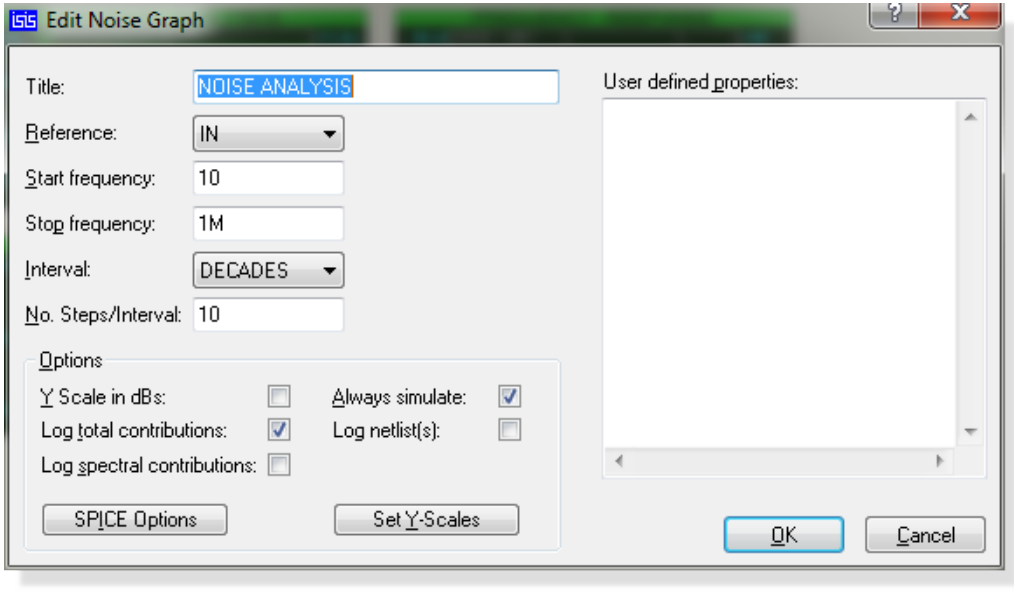
The final form of analysis available is Noise analysis. In this form of analysis the simulator will consider the amount of thermal noise that each component will generate. All these noise contributions are then summed (having been squared) at each probed point in the circuit. The results are plotted against the noise bandwidth.

There are some important peculiarities to noise analysis:

- The simulation time is directly proportional to the number of voltage probes (and generators) in the circuit, since each one will be considered.
- Current probes have no meaning in noise analysis, and are ignored.
- A great deal of information is presented in the simulation log file.
- PROSPICE computes both input and output noise. To do the former, an input reference must be defined - this is done by dragging a generator onto the graph, as with a frequency reference. The input noise plot then shows the equivalent noise at the input for each output point probed.

To perform a noise analysis on our circuit, we must first restore R1 and R2 back to 470kW. Do this now. Then select a Noise graph type, and place a new graph on an unused area of the

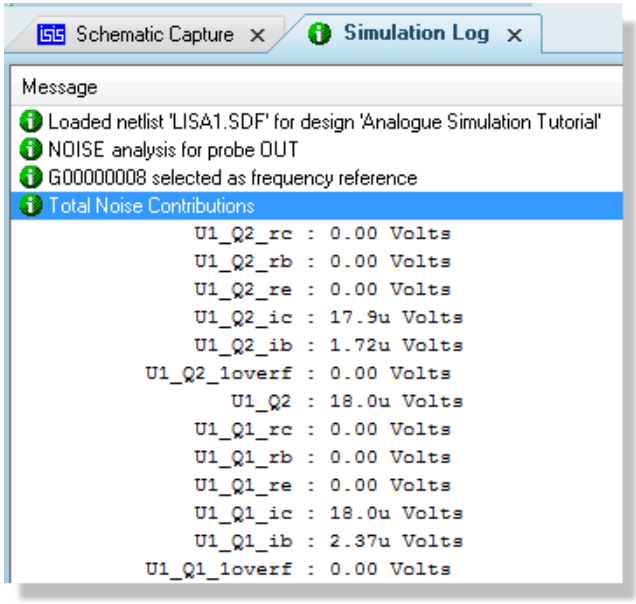
schematic. It is really only output noise we are interested in, so tag the OUT voltage probe and drag it onto the graph. As before, the default values for the simulation are fine for our needs, but you need to set the input reference to the input generator IN. The Edit Noise Graph dialogue form has the check box for displaying the results in dBs. If you use this option, then be aware that 0dB is considered to be 1 volt r.m.s. Click on Cancel to close the dialogue form.



Simulate the graph as before. When the graph is maximised, you can see that the values that result from this form of analysis are typically extremely small (pV in our case) as you might expect from a noise analysis of this type. But how do you track down sources of noise in your circuit? The answer lies in the simulation log. View the simulation log now, by clicking on the simulation advisor at the bottom of the schematic frame.

Use the down arrow icon to move down past the operating point printout, and you should see a line of text that starts

'Total Noise Contributions at ...'



This lists the individual noise contributions (across the entire frequency range) of each circuit element that produces noise. Most of the elements are in fact inside the op-amp, and are prefixed with U1_. If you select the Log Spectral Contributions option on the Edit Noise Graph dialogue form, then you will get even more log data, showing the contribution of each component at each spot frequency.