

---

## CHAPTER 4

# Editing Schematics

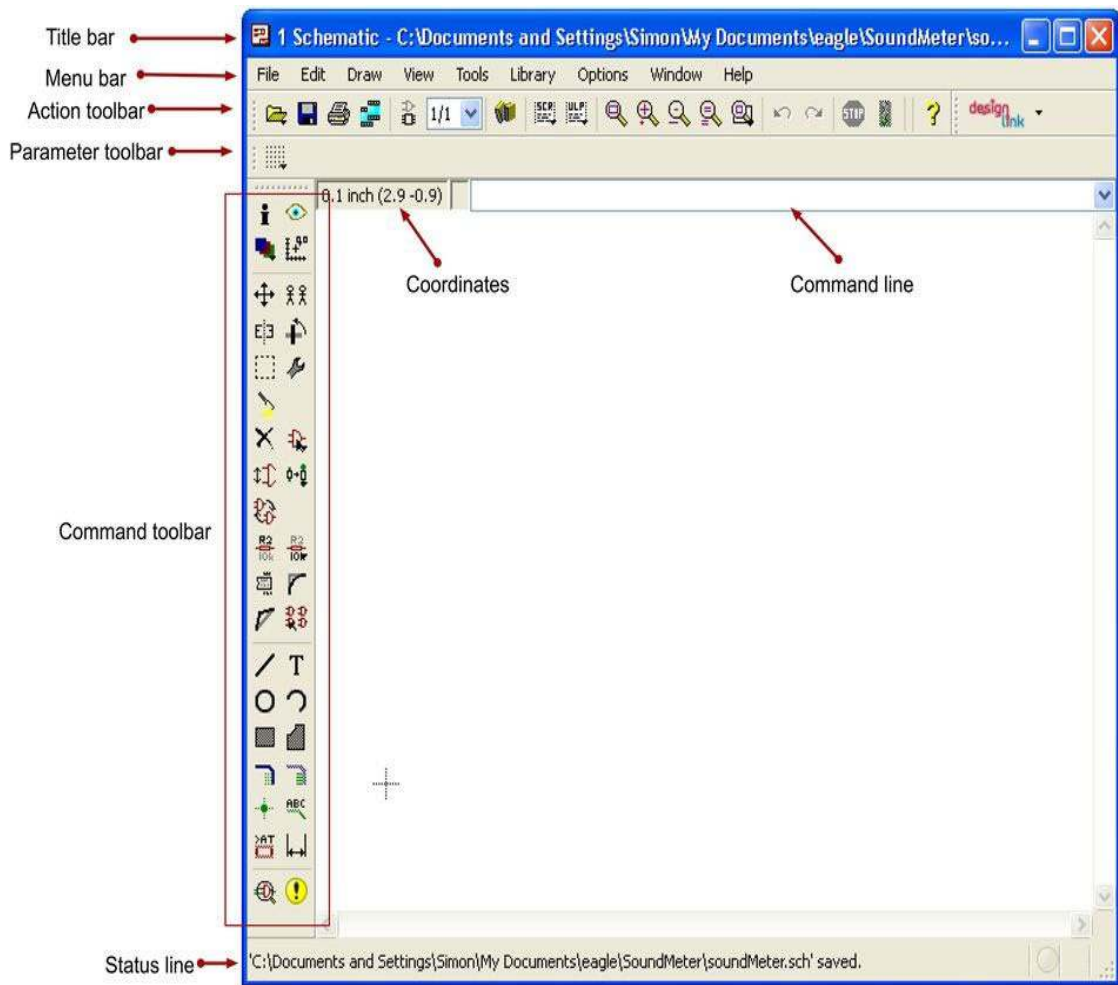
In [Chapter 2](#), we very quickly built the schematic for a simple LED flasher using a 555 timer. In this chapter, after some general explanation of the Schematic Editor, we will start developing a second project, a sound-level meter that includes a small amplifier and 10 LEDs to indicate the level of sound. This project will be continued in [Chapter 5](#) when we come to look at laying out PCBs in more detail.

Because you are going to be experimenting with various editor commands to see how things work, it is probably a good idea to create a new “play” project to which you can add a new schematic for experimenting with.

---

### The Anatomy of the Schematic Editor

[Figure 4-1](#) shows a new “Schematic Editor” window. We will now look at the parts of this window in detail and look at what everything does. We will start by identifying and naming the main areas of the window.



**FIGURE 4-1** The “Schematic Editor” window.

Starting at the top of the screen, we have the Title bar. This shows the file name of the schematic being worked on. This can be useful as confirmation that you are actually working on the document that you think you are working on.

Next, we have the Menubar. This contains options for saving the document, as well as providing an alternative means of accessing many of the functions found on other icon buttons on the editor.

The Action toolbar contains a number of commands and features that are also accessible on the menus. From left to right in [Figure 4-1](#), the icons are

- **Open.** Another schematic.
- **Save.** The current schematic.
- **Print.** Open the print dialog.
- **CAM.** Launch the CAM (computer-aided manufacturing) processor for producing design files suitable for sending to a PCB fabrication shop (see [Chapter 6](#)).
- **Board.** Switch to or open the corresponding board design for the schematic.
- **Sheet Drop Down.** We are only allowed one sheet with the light version of EAGLE, so we can ignore this.
- **Use Library.** Add a library to the list of libraries in use on this schematic.
- **Execute a Script.** See [Chapter 10](#).

- **Execute a User-Language Program.** See [Chapter 10](#).
- **Zoom to Fit.** Zooms to just fit all the components in the schematic into the editor area (useful).
- **Zoom In.** It is easier to use the scroll wheel on your mouse if you have one.
- **Zoom Out.**
- **Redraw Screen.** Occasionally, screen redraw is glitchy. This forces a redraw of the screen.
- **Zoom to Selection.**
- **Undo.** Undo the last change made to the schematic (very useful).
- **Redo.**
- **Cancel Command.** Cancel a command that is running.
- **Help.** Search for help about using EAGLE.

Under the Action toolbar, you will find the Parameter toolbar. The contents of this toolbar will change as you select different commands in the Command toolbar. It is the Command toolbar that you will use most when you are editing a schematic. We will deal with this specifically in the next section.

Underneath the Parameter toolbar is a gray area that displays the current coordinates of the cursor. To the right of this is a text entry field called the *command line*, where you can type commands equivalent to pressing buttons. We will look at this briefly in [Chapter 9](#) because generally users prefer to click on commands rather than type them.

---

## The Command Toolbar

The icons in the Command toolbar are labeled in [Figure 4-2](#). There are a lot of commands that we can use. The ones labeled in bold in the figure are the most commonly used ones. You will use these commands a lot. Some of the others deserve a mention so that you know they are there when you do come to need them.

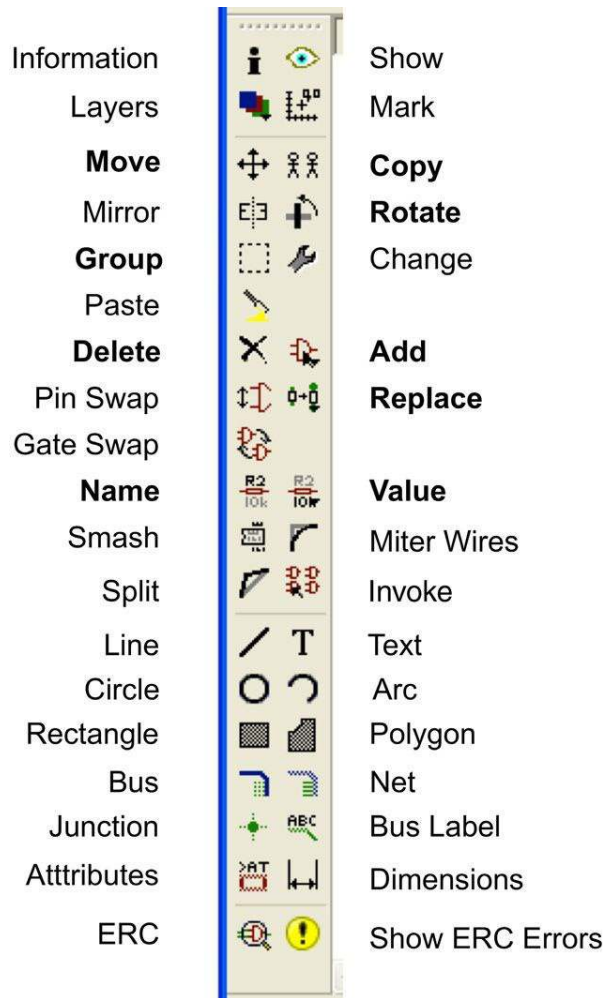


FIGURE 4-2 The Schematic Editor Command toolbar.

## Common Commands

We have already used some of the common commands in [Chapter 3](#), but we will recap on them here.

### Move

With this command selected, you can drag any components or nets around on the Schematic Editor. Sometimes it can be difficult to select the component you want. If you look carefully, you will see that each component has a little cross on it. This is the selection point, so aim for that.

To move more than one thing at a time, you have to use the Group command.

### Group

The Group command can be very baffling when you first come to use it. It allows you to select a set of parts by dragging over the set with the mouse. However, to then do anything meaningful with the selected set, you have to think ahead and select a command before you select the Group command. For example, we will look at dragging a group of items from one area of the Schematic Editor to another. This is something you are likely to do fairly often.

The sequence of actions is

1. Select the Move tool.

2. Select the Group tool.
3. Drag the cursor over the items that you want to include in the group move. They will become highlighted.
4. Right-click, and down near the bottom of the Context menu, you will see an option “Move: Group” (Figure 4-3).

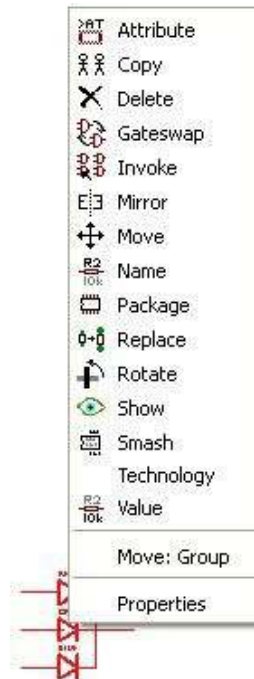


FIGURE 4-3 “Move: Group” on the Context menu.

### Delete

Delete anything you subsequently click on without further warning. You can also carry out this command as a Group command.

### Name and Value

These two commands work quite well in the sort of “verb-noun” workflow of EAGLE. Having dropped a load of resistors onto the schematic, you will probably want to go around them setting their names and values.

### Copy

Copy is a useful command that we have not met before. It should perhaps more accurately be called “Clone” or “Duplicate.”

To use it, select the “Copy” icon, and then click on the item you want to make a copy of. The copy will become attached to the cursor, which you can then move and click to deposit the copy onto the schematic.

### Rotate

To rotate a component, click on the Rotate command, and then click on the component to be rotated. This will rotate it through 90 degrees. To rotate it more, just keep clicking.

When drawing a schematic, it only really makes sense to rotate through multiples of 90 degrees. However, when designing the board layout, it is occasionally useful to be able to rotate a component to an unusual angle such as 45 degrees.

---

**TIP** *When you are moving components around, you will often find that you need to switch between moving and rotating. You can rotate a component when using the Move command by right-clicking.*

### **Add**

We have dealt pretty well with Add. Once you have selected the component that you want to add, then every click of the mouse will result in a new component being dropped onto the design area. To go back and select a different component to add, click on the `ESC` key.

One aspect of the Add command that we have not looked at is adding multigate or multipart components. Examples of this kind of component include logic chips that contain multiple gates or chips such as the NE556 that contain the equivalent to two NE555 timers in a single chip.

When you add one of these components, the first click will put the first gate or subpart onto the design area; subsequent clicks will place the other gates or subparts. Each part will be given a name, the first part of which is the component number and the second part of which is the gate or subpart within that chip.

### **Replace**

Replace is a useful command that lets you swap out one component for another that is pin-compatible. This has the advantage over deleting it and adding a new part in that it retains all the net attachments to the part.

To use it, select the command, and then browse for the replacement component. Then click on the component in the design area that you wish to replace.

### **Net**

We will cover nets in a later section.

## **Other Commands**

Other commands that you will need from time to time are detailed below. It is worth gaining some basic familiarity with them.

### **Information**

With this command selected, every time you click on an item in the editor, you will see an information window like [Figure 4-4](#) that shows you information about that item.

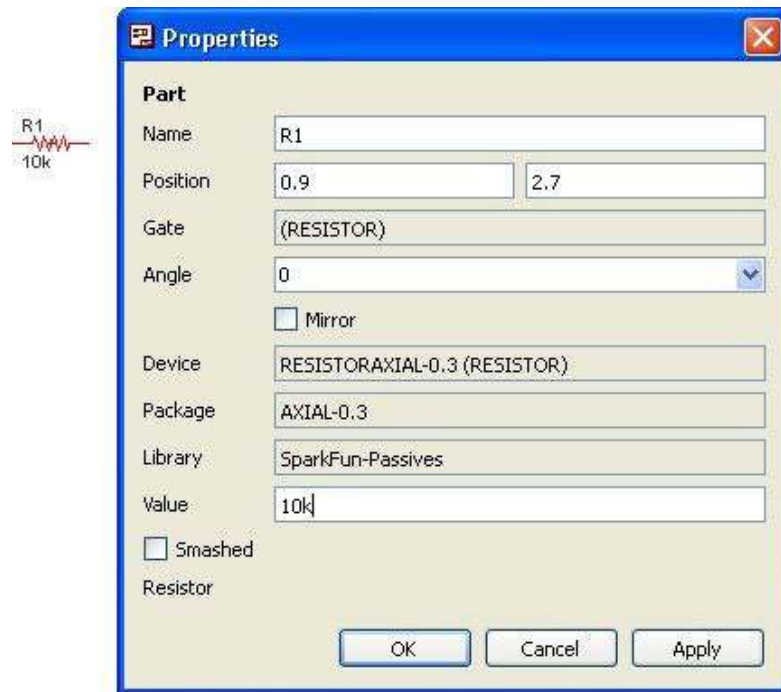


FIGURE 4-4 Information about a resistor.

This actually provides an alternative way of changing information about the item, such as its name and value.

### Layers

You can control what information is visible on the schematic. This is not really very useful on a schematic but becomes essential when we look at board layout in [Chapter 5](#).

### Mirror

Mirroring a component flips it through its vertical axis. [Figure 4-5](#) shows a transistor and a mirrored transistor next to each other. It just allows the circuit symbol to be displayed the other way around.

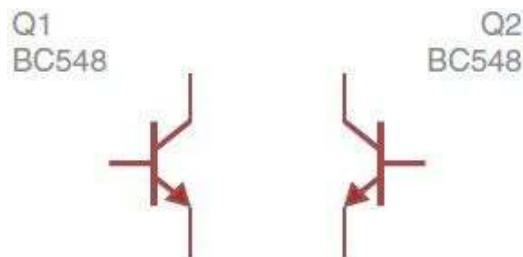


FIGURE 4-5 A mirrored transistor.

### Paste

Paste is the same concept as the Paste in Cut and Paste, but this being EAGLE, it works in a different way. To make a copy to be pasted, you select items either individually or using the Group command. The Paste command will then attach a duplicate of everything selected to the cursor so that you can click to place it in the Schematic Editor.

### **Pin Swap**

Some components such as logic gates have pins that are interchangeable. The Pin Swap command allows you to swap the pins. There is no reason why you would want to do this while designing a schematic, but when routing the PCB, you might find that things would become a lot easier if you could just swap over which pin is used.

### **Gate Swap**

As with Pin Swap, this feature only becomes useful when you start designing the PCB layout and suddenly discover that the layout would be much easier if only you had used a different gate or subpart of the chip. By switching back to the Schematic Editor, you can fix this.

### **Smash**

This rather dramatic sounding command is nowhere near as exciting as you might expect. It simply allows you to separate the Name and Value labels that accompany a circuit symbol from the symbol itself so that you can move them around independently. This helps to keep your schematics clear and easy to read.

### **Split**

The Split command allows you to add a waypoint to a net line. Select the command, and then click on the length of the line somewhere. You can then drag a point out, and the line will follow. This is useful for changing the path of a net line without having to delete it and redraw it. Note that you can also use the Move command on net lines.

### **Line, Circle, Rectangle, Text, Arc, and Polygon**

All these commands allow you to add decorations to a schematic. These take no part in the electrical side of the schematic. They will not have any influence over the board layout; they just allow you to add further information to the schematic diagram.

### **Bus and Bus Label**

Schematic diagrams can become messy, especially in digital electronic designs when you have a lot of wires running from one chip to another. To keep things neat, the Bus command allows you to group the net lines together. See the separate section on buses later in this chapter for more information.

### **Junction**

EAGLE does a pretty good job of automatically marking junctions between one net and another with a little blob. The Junction command allows you manual control over this process. You may never use this feature, though.

### **Attributes**

You can add your own custom attributes to a component and then decide if you want them to be displayed. [Figure 4-6](#) shows a POWER attribute being added to resistor R5. We have also specified that just the value (1 W) should be visible.



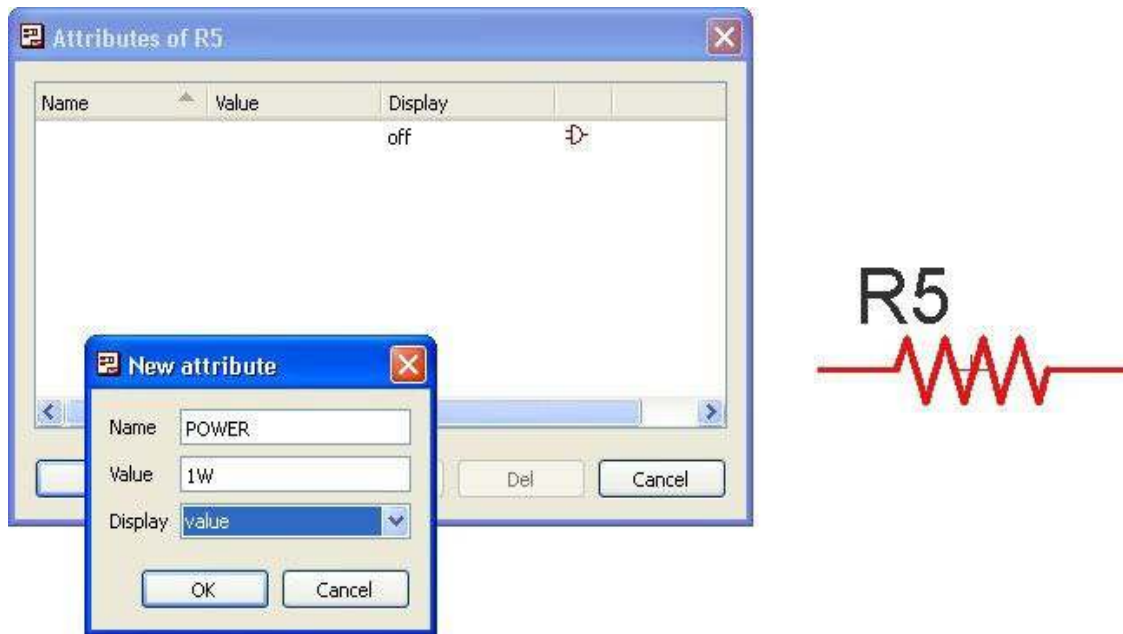


FIGURE 4-6 Adding an attribute.

If the attribute is displayed in the wrong place, for example, overlapping the component graphic, then you can move it by using the Smash command to separate the circuit symbol from its labels.

### ERC and Show ERC Errors

These two commands launch the electrical rule checker (ERC) and open the window showing the results of the check, respectively. We touched on this in [Chapter 2](#) and will meet it again later in this chapter.

### Show

The Show command has a similar purpose to the Information command, except that rather than opening a little window giving you details of the component selected, it displays the information in the status area at the bottom of the screen ([Figure 4-7](#)). Because it is not opening a new window (that then needs closing), this can be a faster way to see what's on a schematic.

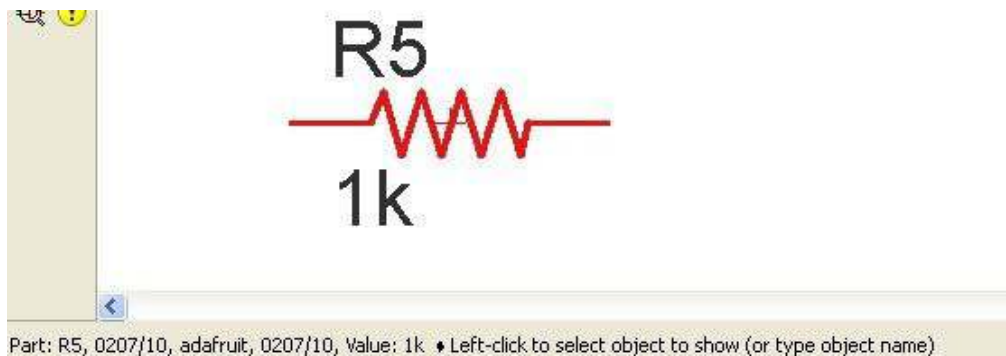


FIGURE 4-7 The Show command.

### Mark

This is not something you are ever likely to use on a schematic. The Mark command allows you to set

a local origin anywhere on the schematic so that you can see the coordinate values relative to that point. Clicking on the “GO” button (looks like a traffic signal) cancels the Mark command.

### **Change**

The Change command allows you to change almost anything about an object on the schematic diagram. You can, of course, change these things using the Information command. The Change command is not that useful on a schematic but comes into its own when you are designing a board layout because it lets you easily change such things as track widths, silk-screen fonts, and so on.

### **Miter Wires**

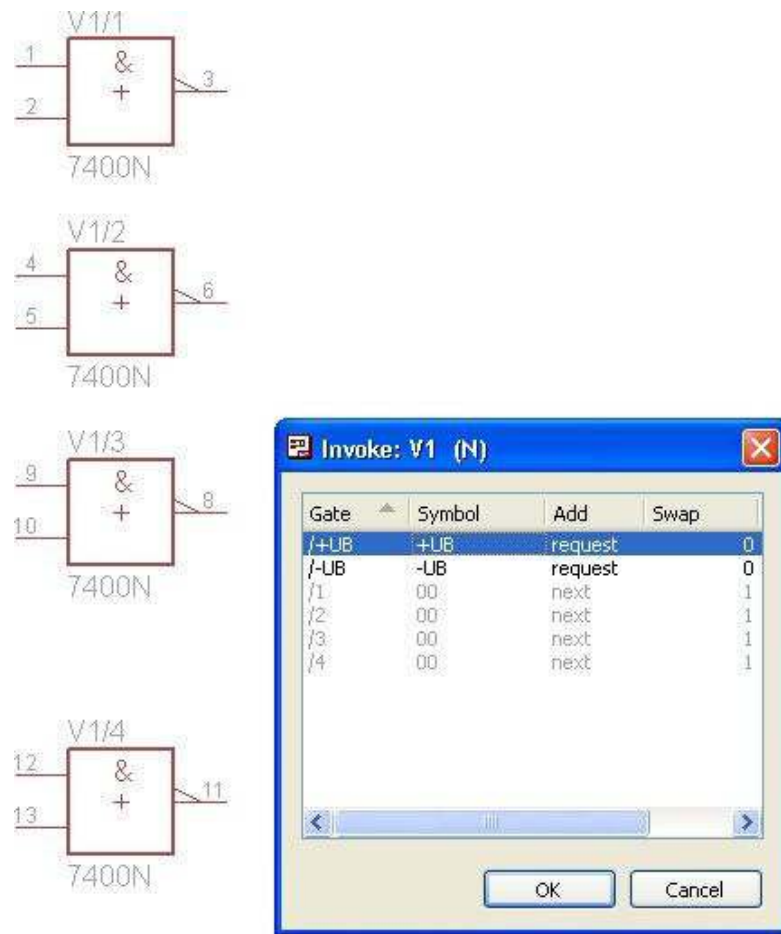
You may never use this feature when editing a schematic. It allows you to put a curve in your net lines. To use it, select the command, and then click on a corner of a net line. You can then set the radius in the dropdown in the Parameter toolbar.

### **Invoke**

This is possibly the worst-named command in EAGLE. You might think that it runs a script. No, not at all. This command would perhaps be better named Fetch. It is used in multipart components such as logic gates. As you saw in the section on the Add command, when adding such components, they appear a gate at a time each time you click on the design area.

The Invoke command gives you more control over this process, allowing you to select the order in which the components are added. This serves little practical purpose.

However, the useful thing you can do with Invoke is to add the power connectors for an integrated circuit (IC). After clicking out four gates from, say, a 7400 IC, if you select the Invoke command and click on one of the gates, a menu like the one shown in [Figure 4-8](#) will appear, allowing you to select either the positive supply for the chip or the negative supply. To add both, action the command twice.



**FIGURE 4-8** Using Invoke to add power-supply pins.

These pins then can be explicitly connected to the appropriate power nets. Note that ICs such as this will have specific net names associated with the power pins, so you do not have to connect them up explicitly as long as you pay attention to the ERC results to make sure that the power nets are all defined correctly.

### Dimension

This command adds dimension labels. There is no reason to do this in a schematic diagram.

### Nets

Aside from the components themselves, nets are the most important part of any design. When you are drawing a schematic, you will use nets to connect one component lead to another. This may be done directly or by connecting a net from one lead of a component to a net that already exists on the diagram.

Let's create a few nets, see how they get named, and see what happens when we connect one net to another. Start by adding four resistors and a net between R1 and R2 and also between R3 and R4 ([Figure 4-9](#)).

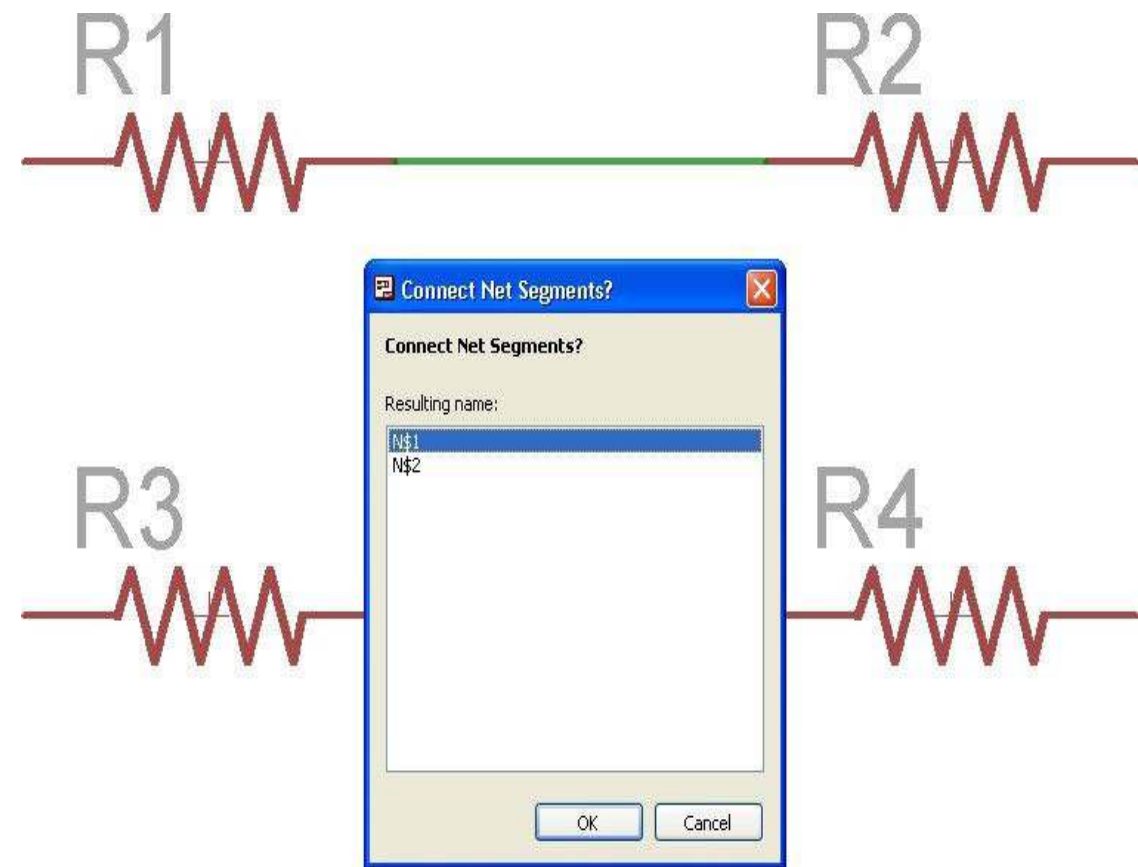


**FIGURE 4-9** Adding nets.

When adding nets, you can click at any point to add a waypoint (bend in the line), although for this exercise the resistors are lined up so that you don't need any corners. When you get to the end point for a net such as the left-hand lead of R2, the net should stop drawing. It is a sure thing that you missed the connection point on the component if the net unexpectedly carries on drawing after you think you have connected it.

If you click the Show command and then select the net between R1 and R2, it will probably tell you that the net is called N\$1, and the other net will be called N\$2.

Now draw another net between the two nets, starting from the middle of the net between R1 and R2. When the net reaches the net between R3 and R4, a little window will pop up ([Figure 4-10](#)).



**FIGURE 4-10** Connecting two nets.

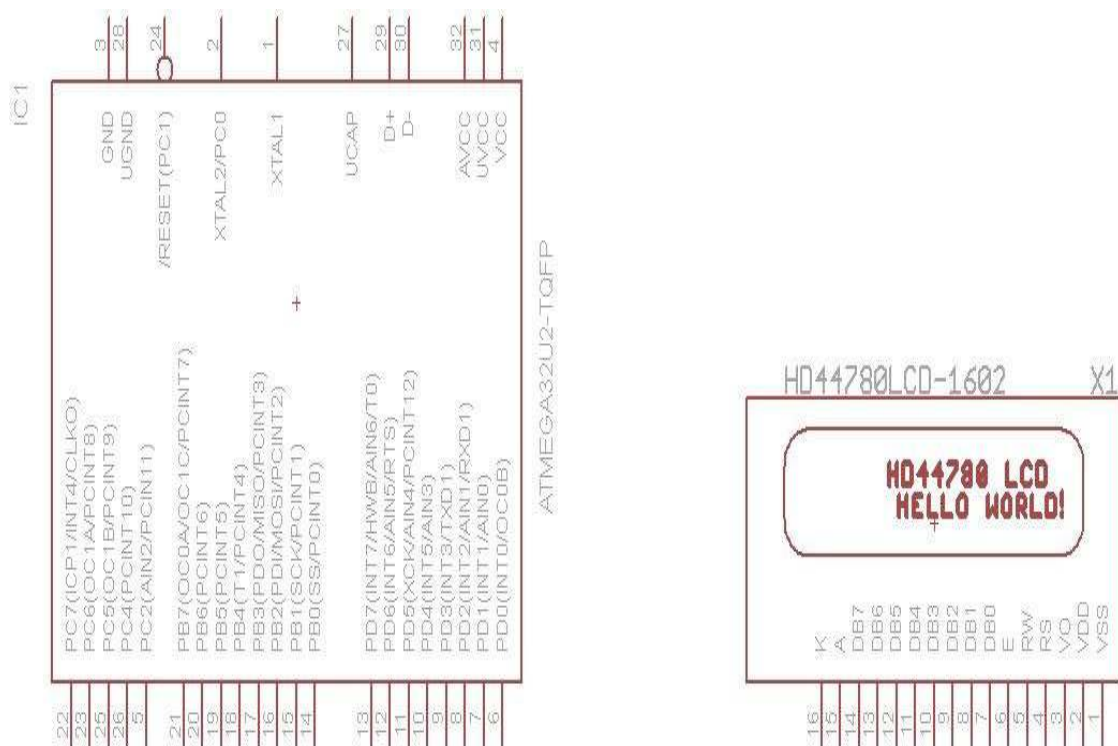
This is offering us a choice of names for the new net that will be the merger of the two nets. It does not matter what name you chose.

---

## Buses

A typical example of where a bus is useful would be where a microcontroller is connected to an LCD module using the HD44780 chip. This requires four data lines and three control lines. In the following example, we will use a bus to connect the display's data and control pins to the microcontroller.

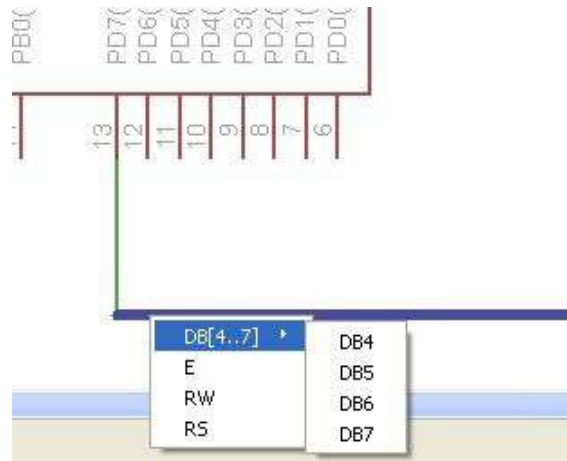
Add the two components (search the libraries for "atmega32u2" and "hd44780"), rotate them, and add a bus, all as shown in [Figure 4-11](#).



**FIGURE 4-11** Two components and a bus.

Now here's the trick: to allow connections to the bus, a special convention is used in naming the bus. Change the name of the bus to "LCD\_BUS1:DB[4..7], E,RW,RS." You can change the name of the bus using the Name or Information commands. This name tells the bus object that any net being connected to it can be connected to one of eight possible slots: DB4, DB5, DB6, DB7, E, RW, or RS. The square brackets and ".." specify a range of values. We could also have set the name to "LCD\_BUS1:DB4,DB5,DB6,DB7,E,RW,RS," and it would work just as well.

Now, when you drag out a net from, say, PD7 on the microcontroller to the bus, it will present a list of possible connection slots (Figure 4-12).



**FIGURE 4-12** Connecting a net to a bus.

At the display end of the bus, you connect nets from the display to the bus using the corresponding slot names.

### Worked Example

Now that you are familiar with the tools available to us with the Schematic Editor, we can start building up a second example schematic. The example project is a sound meter. This will use an amplifier chip, a bar-code chip, and a set of LEDs. In [Chapter 5](#), we will design first a through-hole and then a surface-mount version of the PCB layout for the example.

[Figure 4-13](#) shows the finished schematic that we are aiming for. You will notice that it looks considerably more professional than the schematic from [Chapter 2](#).

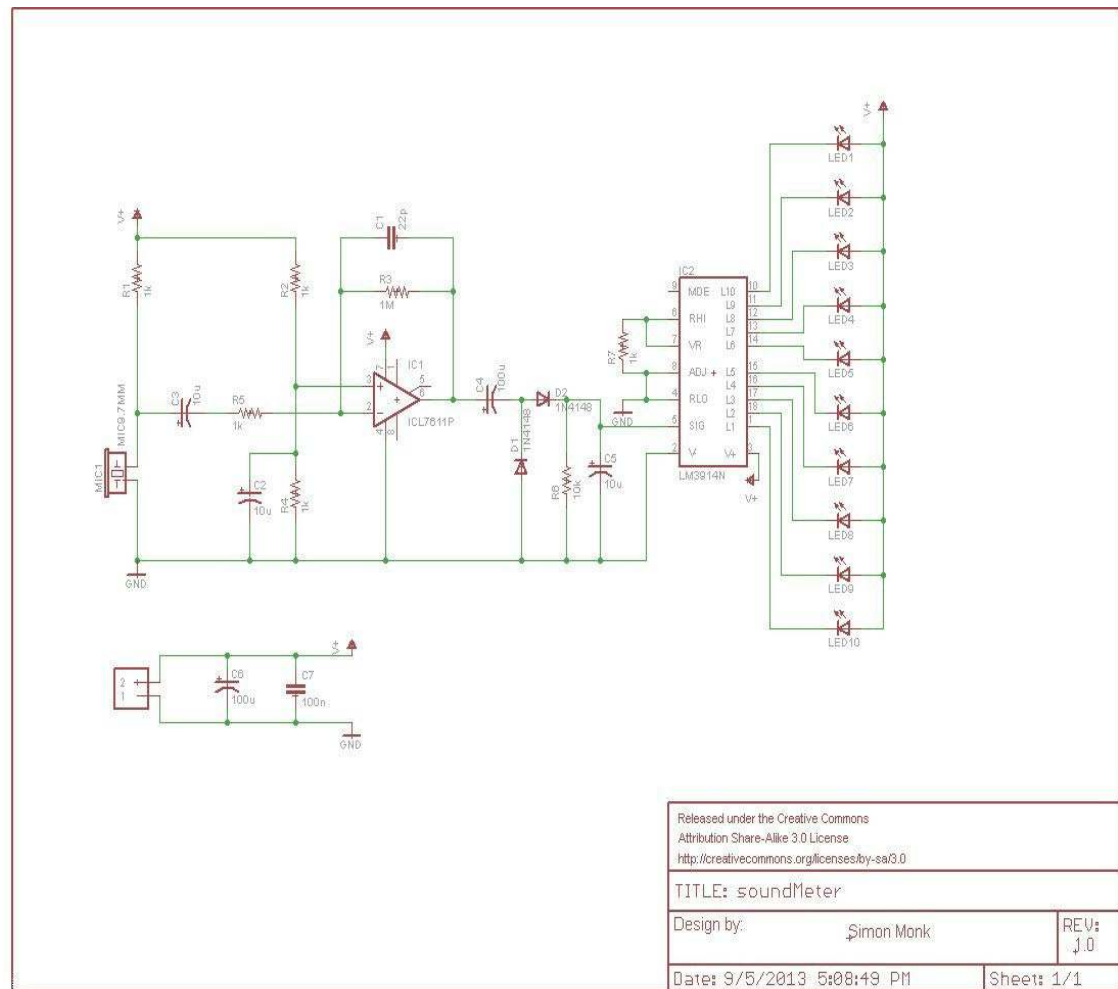


FIGURE 4-13 Final schematic for the bar-code example.

## Starting the Schematic

If your design is a one-off, is just for you, and is never likely to be released for others to use, then you can take the approach that we did in [Chapter 2](#). However, you never know where a design might lead, so it can be a good idea to design neatly.

When you look at other people’s schematic designs, you will often see the schematic itself framed, with an information panel that provides useful information such as the name of the document, the author, version, and other information such as the licensing of the design. The frame used in [Figure 4-13](#) is the letter-sized frame from the Sparkfun-Aesthetics library and indicates that the design is released under a creative commons license.

The frames can be added as if they were any other part. To add a frame, search the library for “frame.” This will bring back frames of different sizes. The size “Letter” will be fine for small designs; for bigger designs, “A3” is a better option. If a design grows unexpectedly, then you can always replace the frame with a bigger one.

Place the frame with the bottom-left corner at the origin. Some of the details of the frame, such as the document name and date last saved, will be automatically shown on the sheet. Note that for them to be updated, you may have to select “Redraw” from the View menu. To change the text in other fields, such as the “Design By” and “Rev” fields, there are two possible ways. The neater way only works if the lettering in the frame has field markers, for example, “>AUTHOR.” If it does, then from the Edit menu, select the option “Global Attributes..,” and add a new attribute and value, where the



attribute name is the same as the word in the frame but without the > character on the front. Then redraw the screen to see the attribute value.

Unfortunately, most of the frames in the library, including the Sparkfun library, do not have such variables defined, so to add the text, use the Text command from the toolbar, enter the text, but before you drop it over the field, change the layer in the Action toolbar to be layer “94 - Symbols.” You can also change the text size here or at any time using the Information command, clicking on the text.

## Adding the Components

The final set of files for this design, and all the designs in this book, can be downloaded from the author’s website ([www.simonmonk.org](http://www.simonmonk.org)). However, I recommend that you follow the instructions to draw the design for yourself from scratch.

### Add the Components

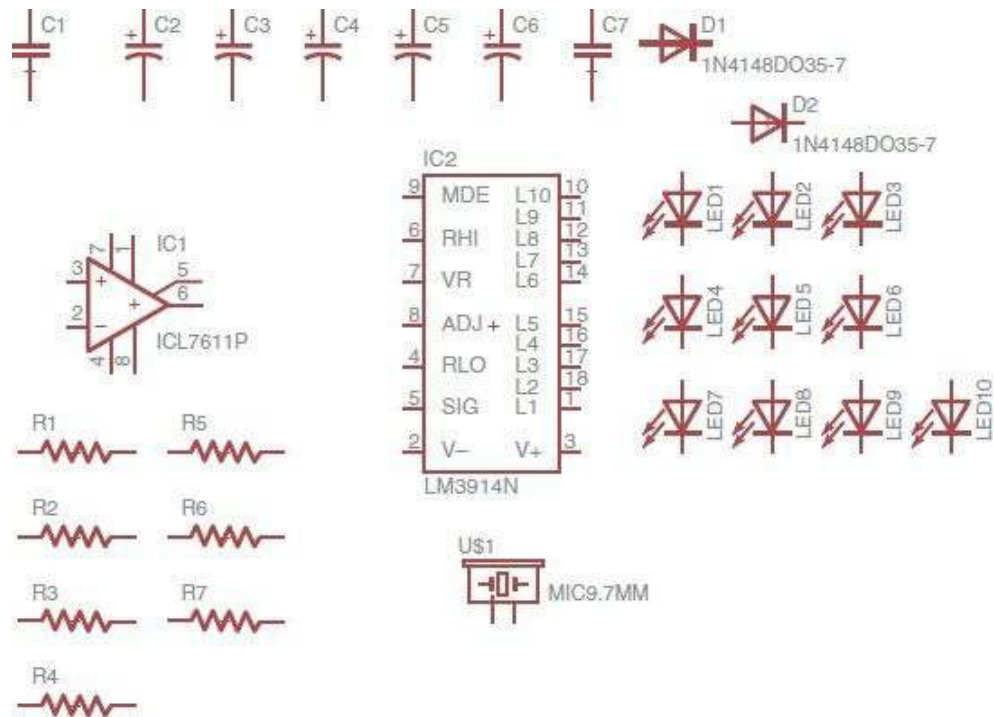
Everyone has different ways of working, but I find it easiest to add all the components that I am going to need, lining them up neatly before moving them to where they are likely to be needed on the schematic.

Add the items in [Table 4-1](#) to the schematic. Where possible, the components are from the Sparkfun library. Searching on the values in the “Device” column works best when searching for the components in the libraries.

**TABLE 4-1** Components Used in the Example Project

Part	Value	Device	Package	Description
C1	22 pF	CAPPTH	CAP-PTH-SMALL	Capacitor
C2, C3, C5	10 $\mu$ F	CAP_POLPTH2	CPOL-RADIAL-10UF-25V	Capacitor polarized
C4, C6	100 $\mu$ F	CAP_POLPTH2	CPOL-RADIAL-10UF-25V	Capacitor polarized
C7	100 nF	CAPPTH	CAP-PTH-SMALL	Capacitor
D1, D2	1N4148	1N4148D035-7	D035-7	Diode
IC1	ICL7611P	ICL7611P	DIL08	Op amp
IC2	LM3914N	LM3914N	DIL18	Dot/bar display driver
LED1-10	—	LED3MM	LED3MM	LEDs
R1, R2, R4, R5, R7	1 k $\Omega$	RESISTORPTH-1/4W	AXIAL-0.4	Resistor
R3	1 M $\Omega$	RESISTORPTH-1/4W	AXIAL-0.4	Resistor
R6	10 k $\Omega$	RESISTORPTH-1/4W	AXIAL-0.4	Resistor
U\$1	MIC9.7MM	MIC9.7MM	MIC-9.7MM	Omnidirectional electret microphone
J1	1X2-3.5MM	1X2-3.5MM (1X2)		Two-way screw terminal

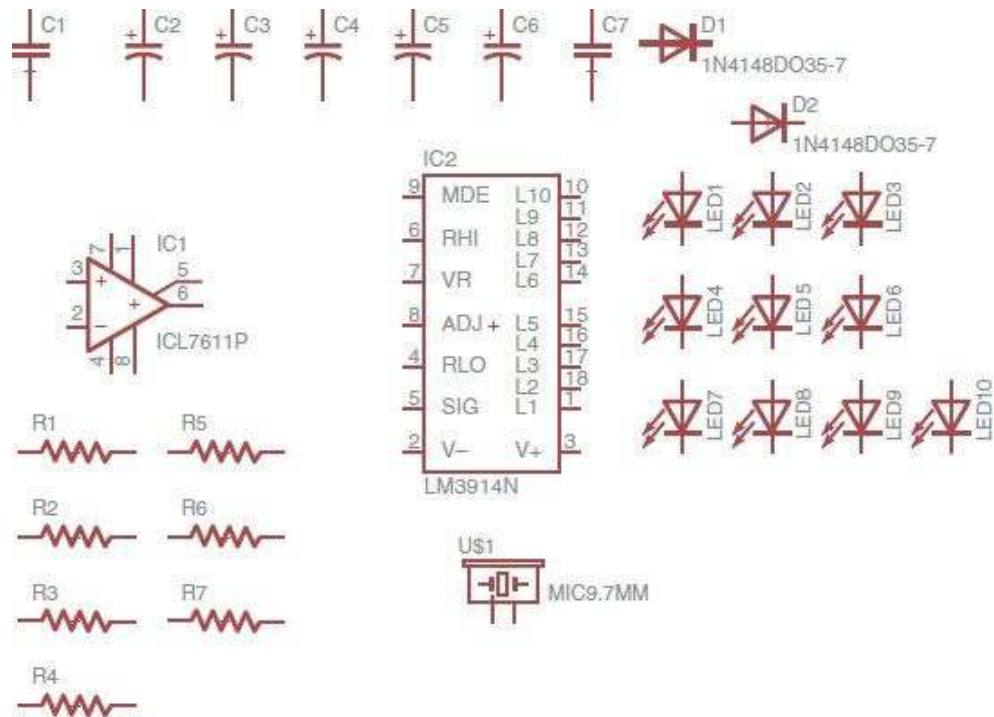
After all these components have been added to the schematic, it will look something like [Figure 4-14](#). Note that this figure does not show the screw terminal J1.



**FIGURE 4-14** Schematic with all components added.

Having added the basic component shapes, you can then go through and set their values according to [Table 4-1](#). D1 and D2 have rather long and overly specific values, so change them to be just 1N4148. You will have to override the warning that they have no user-definable values. This is also a chance to rename the microphone to something more obvious such as MIC1.

With all the components named and the terminal block J1 added, your collection of components should be looking like [Figure 4-15](#).



**FIGURE 4-15** Setting component values and renaming.

We can now start positioning the components in the approximate location where they will be needed. It is best to place the big multipin devices first, fairly centrally, and then place the other components around them. Logically, the amplifier stage using IC1 happens before the LED bar-code chip does its thing, so put the op amp to the left.

Position and rotate all the components where necessary until they look like [Figure 4-16](#). Remember that you can rotate when you are in Move mode just by right-clicking the mouse.

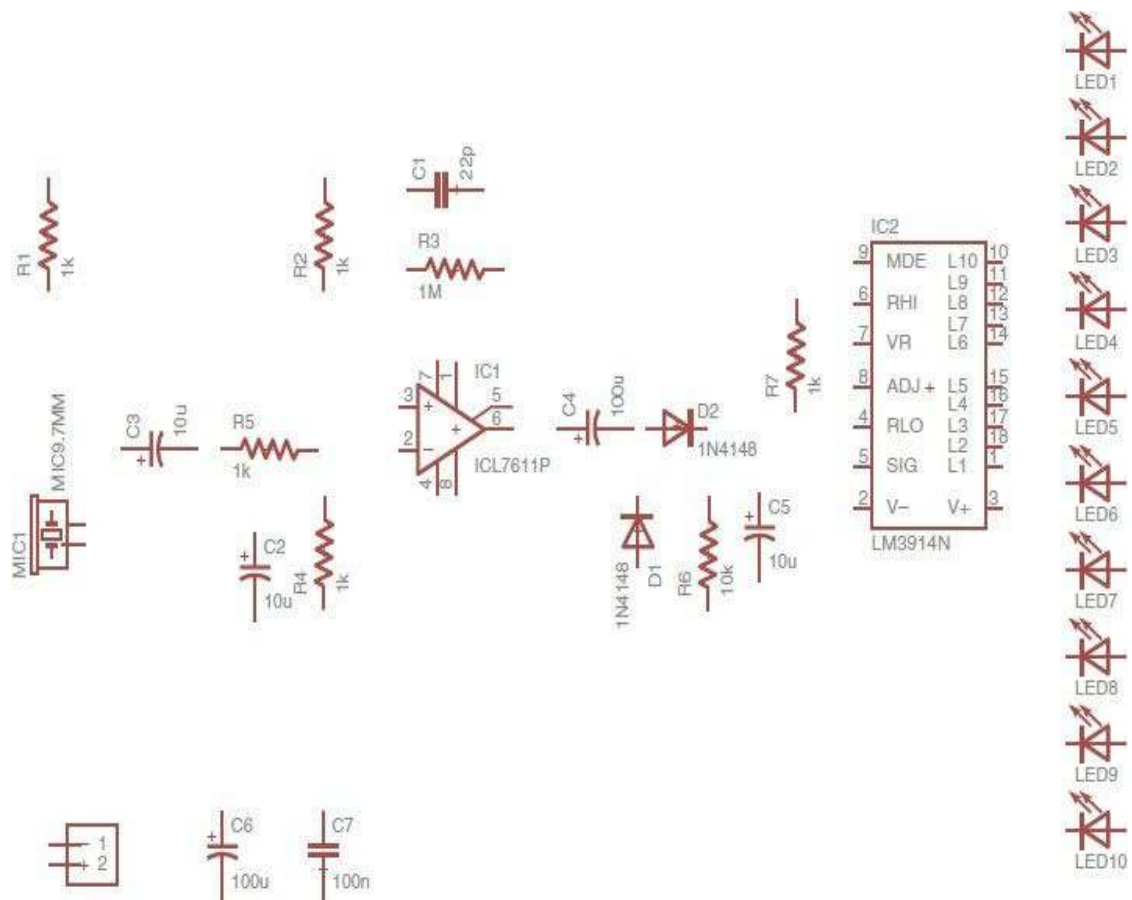


FIGURE 4-16 The components in the correct positions.

## Adding the Supplies

In our first design, back in [Chapter 2](#), we glossed over the whole idea of supplies. Power came from two pins of a connector, and we did not do anything special to identify them as power rails. We didn't have to. It worked as it was, and that was our rough and ready chapter.

However, generally, except from the very simplest designs, there are good reasons to identify which nets are supply nets and both name them appropriately and associate them with special “supply” parts. This can help to keep diagrams neat because you can repeat GND and other supply symbols so that you do not need to connect every point on the diagram that is GND to every other GND with a line—something that very rapidly gets unmanageable as a design becomes more complex. You can just use the GND symbol as a stand-in.

Therefore, we are going to add two new supply parts, one for GND and one for V+. Select the Add command, and then search for “supply” ([Figure 4-17](#)).

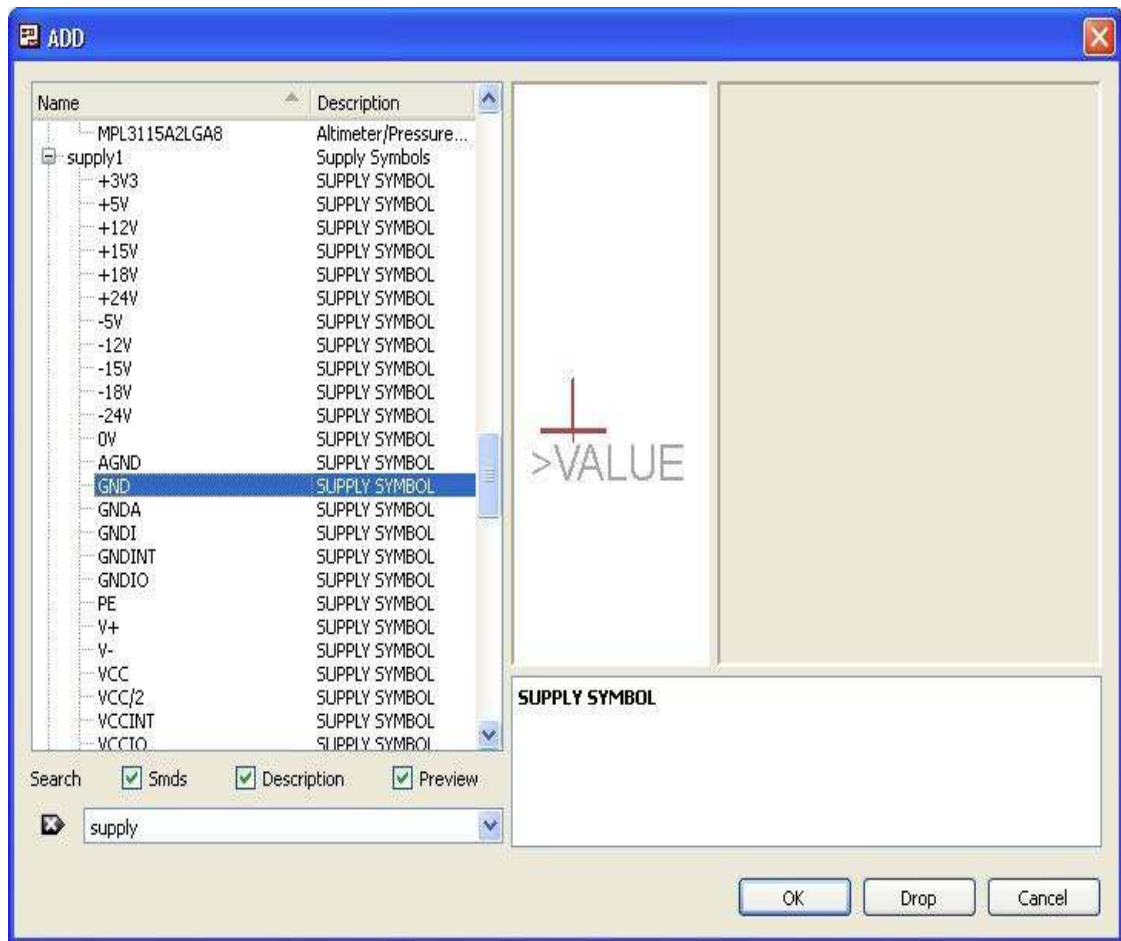
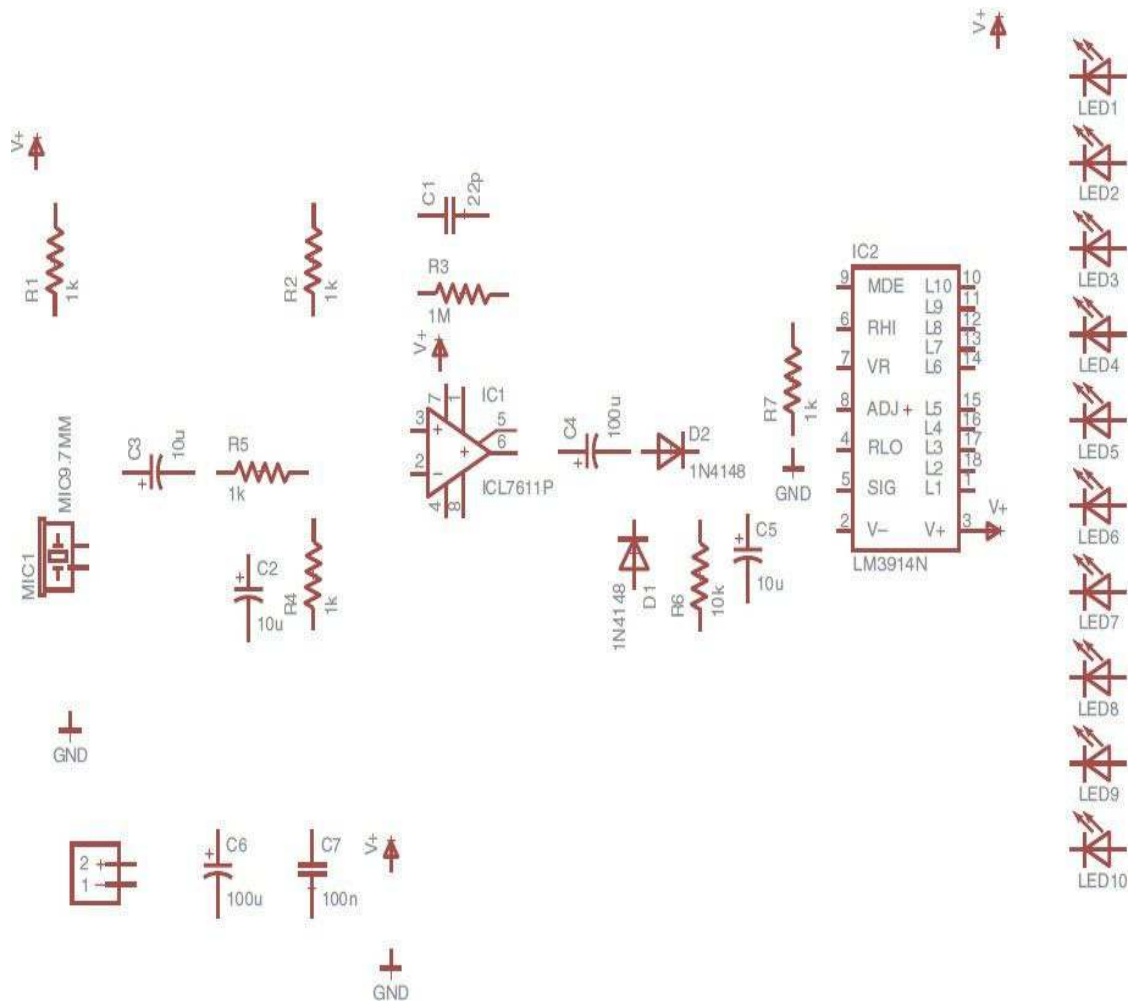


FIGURE 4-17 Supply parts in the libraries.

As you can see, there are many parts to choose from. We will just use GND and V+ in the supply1 library. Add one of each near C6 and C7 at the bottom of the diagram and also +V above R1, above pin 7 of IC1, and right next to pin 3 of IC2 and near LED1. Add GND symbols below the Mic and near pin 4 of IC2. When these are in place, your schematic should look like Figure 4-18. Note that in this figure we have also rotated J1 because it was the wrong way around for making connections in Figure 4-16.

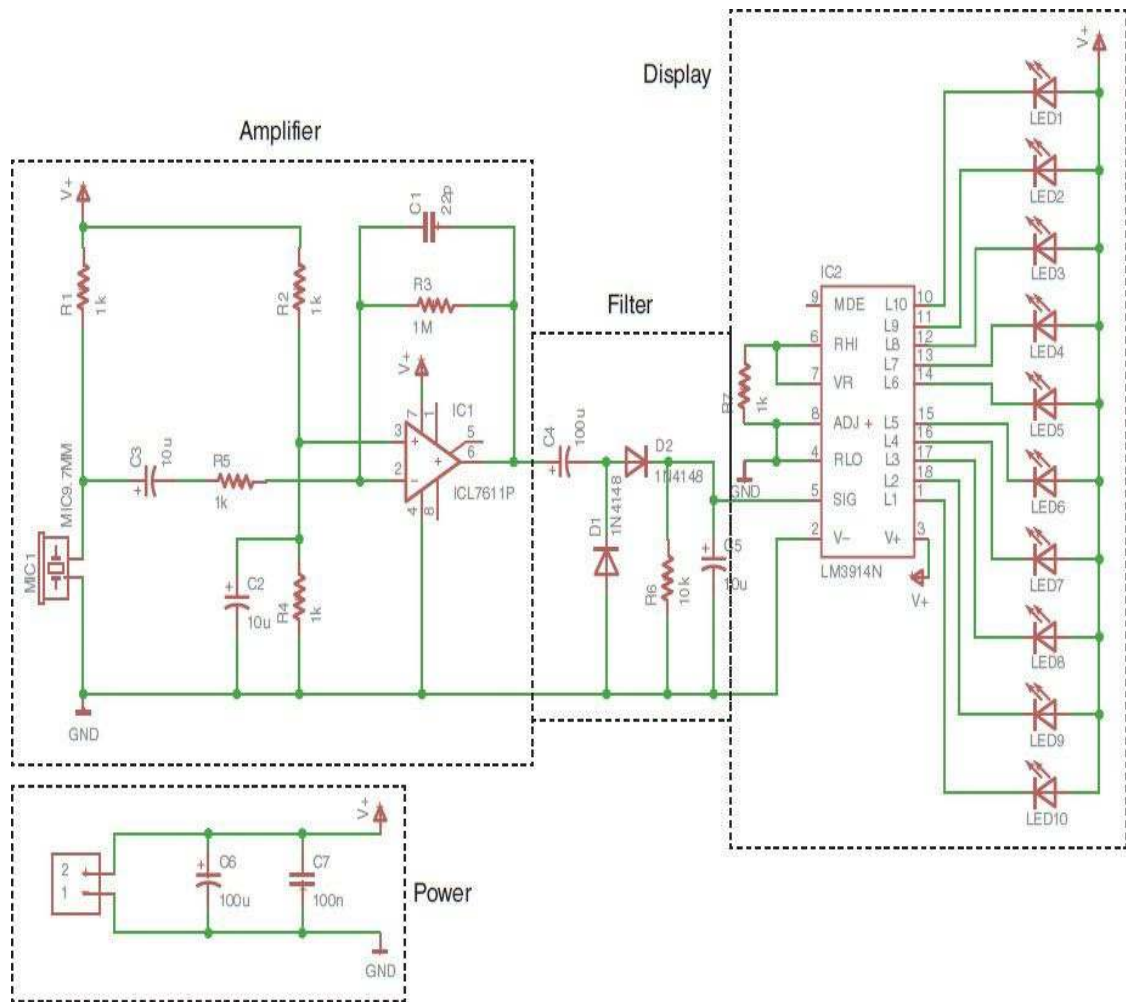


**FIGURE 4-18** Supply parts added to the schematic.

You may find that putting all the components in place in one go just gets too complicated, in which case it can be easier to start connecting up some of the nets. However, once you start connecting things together, moving things gets more tricky because you need to worry about where all the lines are going.

### Adding the Nets

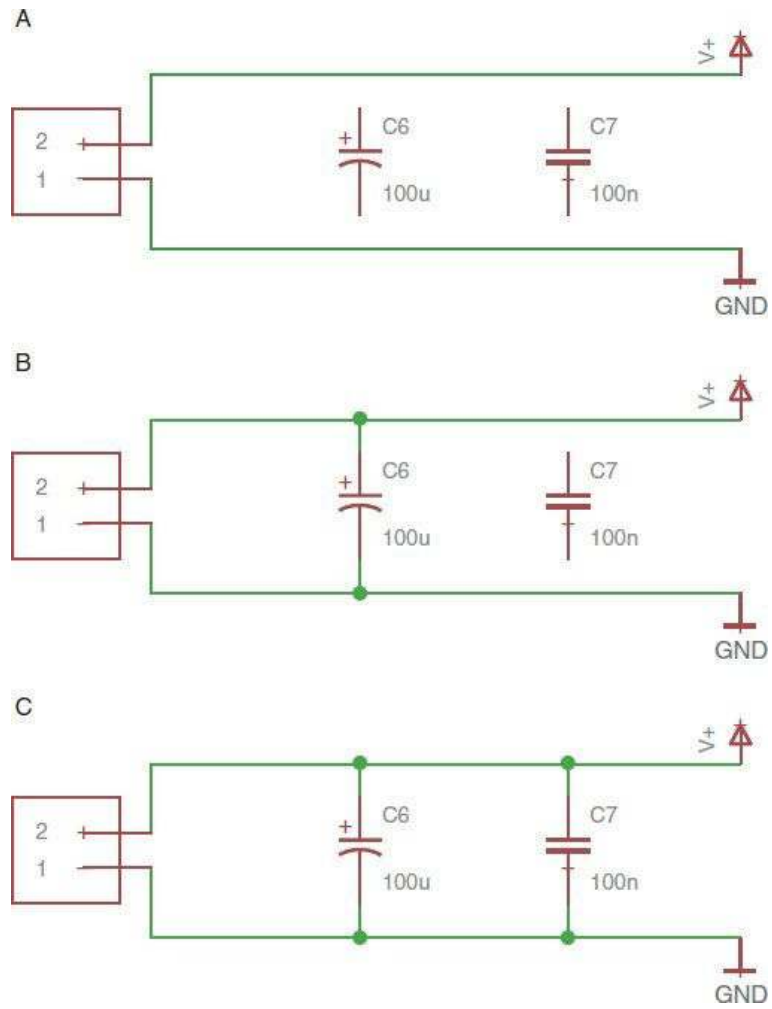
It's time to start adding the nets, and no doubt, we will need to move things around a little, but hopefully not too much. We will draw in the nets for this example in four stages, corresponding to four areas of the design (Figure 4-19).



**FIGURE 4-19** Logical areas of the design.

The first trivial area is the area at the bottom of the screen comprising the power connector and the two decoupling capacitors. Then we have the amplifier section, centered around IC1, the low pass filter between C4 and C5, and then the bar-code area around IC2.

Let's start by connecting nets between the parts in the power area at the bottom. [Figure 4-20](#) shows the sequence of connections being made.



**FIGURE 4-20** Adding nets to the supply area.

By starting with connections between the screw terminals J1 and the supply symbols, it makes it easy to then reliably connect up C6 and C7. The temptation to just run the supply line level across the top of the components ([Figure 4-21](#)) should be resisted. This will not actually connect them to the supply nets. You can tell that they are not connected because there are no little dots where the capacitor leads meet the GND and V+ nets.



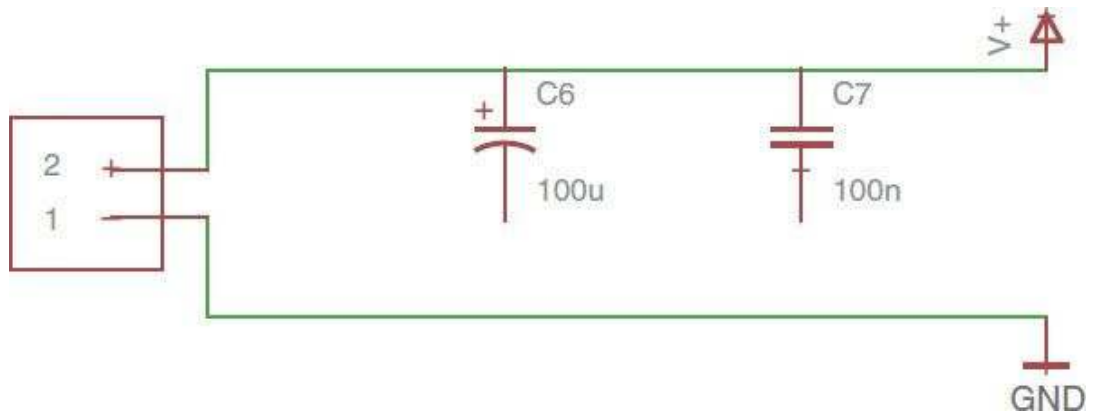
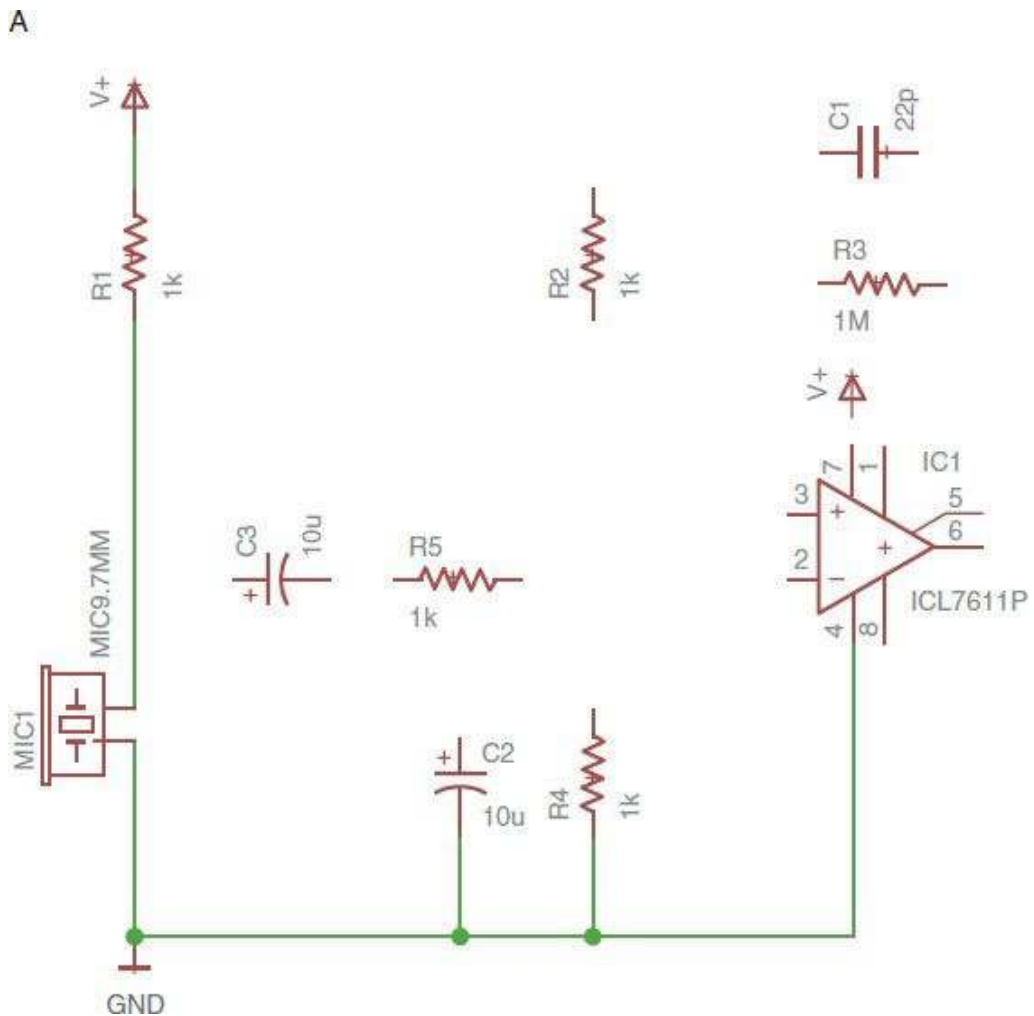
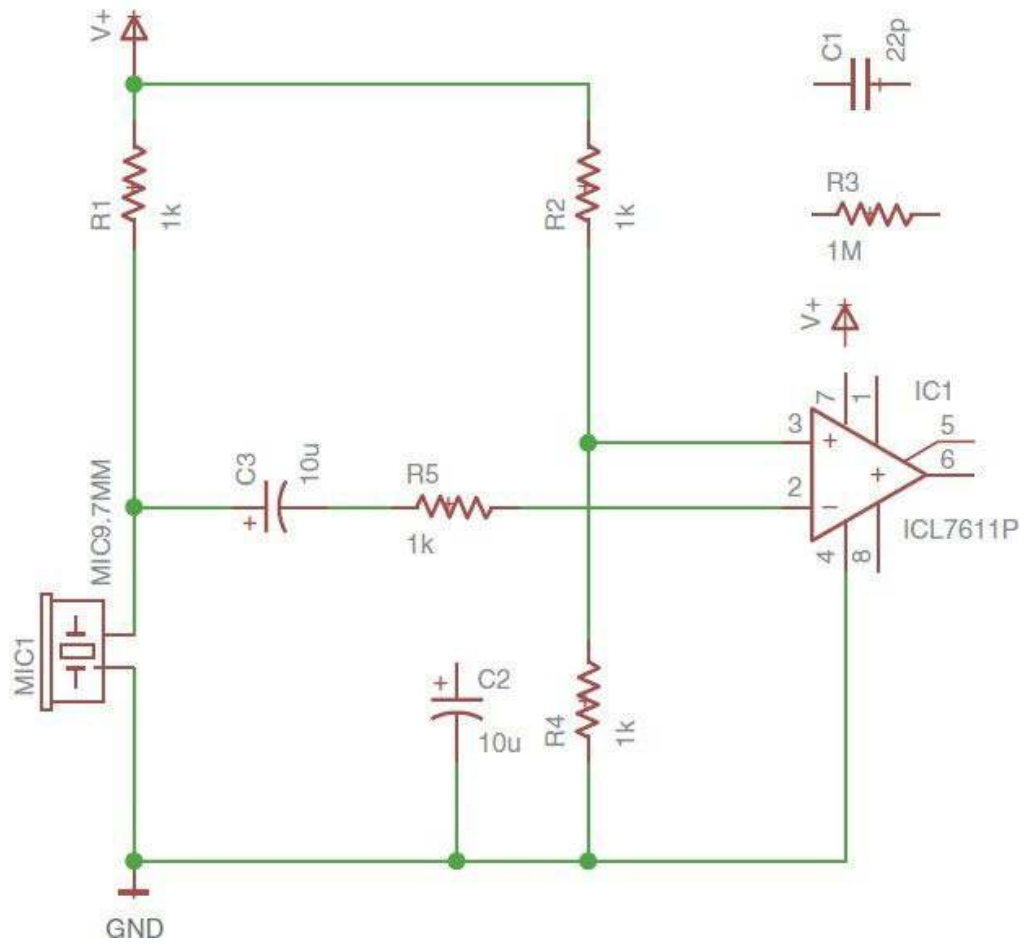


FIGURE 4-21 How not to connect nets.

Now turn your attention to the amplifier stage, around IC1. This is probably the busiest section of the design. Working from left to right, [Figure 4-22](#) shows how I connected these up, starting with the supply connections. Some of the parts will need moving a little so that everything lines up nicely.



B



C

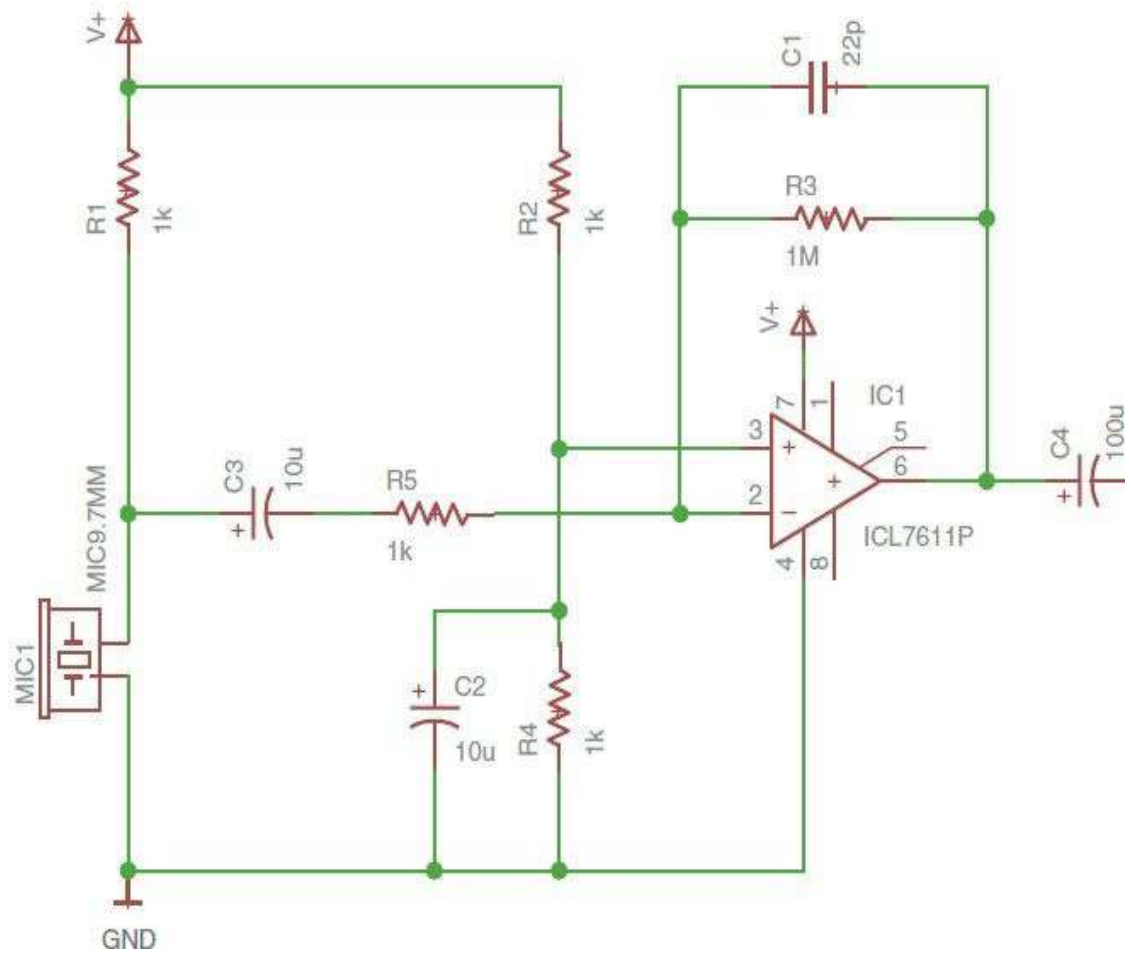
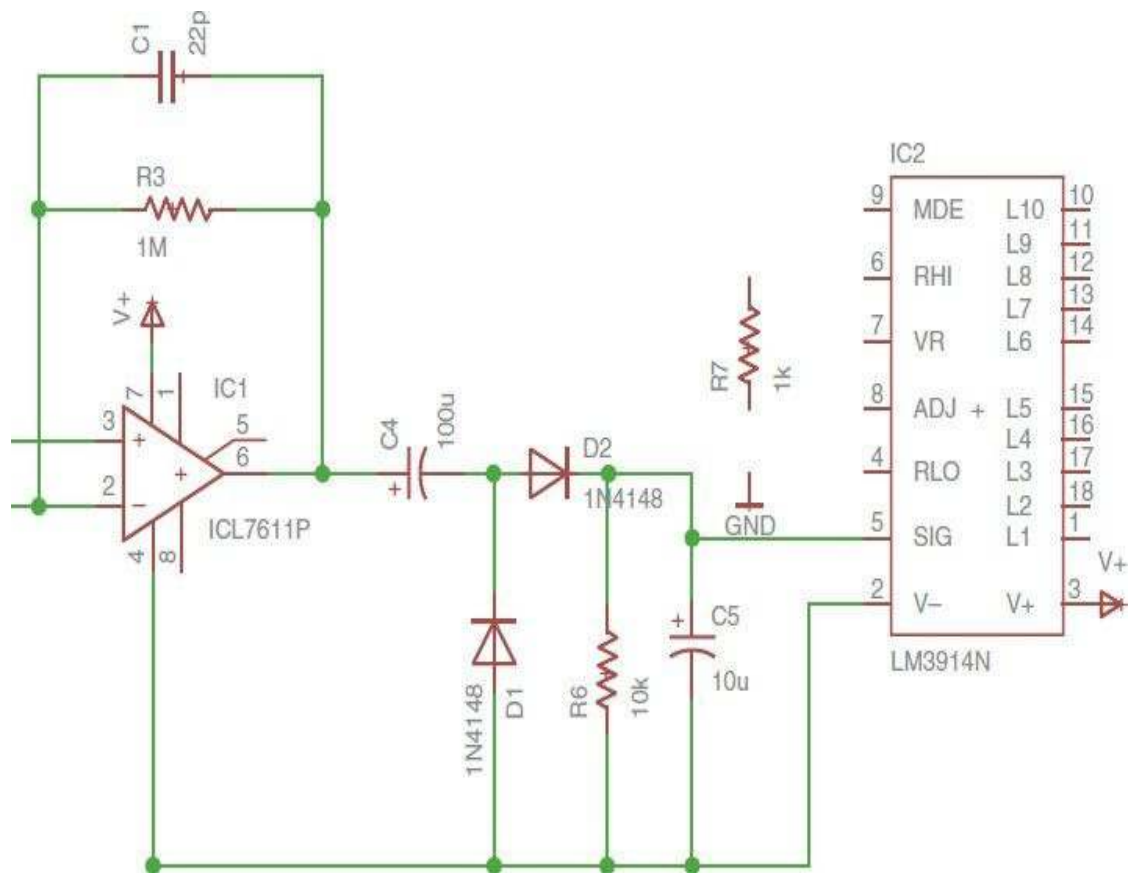


FIGURE 4-22 Connecting up the amplifier section.

Next, continuing to the right of the schematic, we wire up the filter section so that it looks like [Figure 4-23](#).



**FIGURE 4-23** Connecting up the filter section.

Note that we have continued the GND net from the amplifier actions. I also had to move the GND supply near pin 4 of IC2 because it was in the way.

The final stage is to connect up the bar-code part of the schematic around IC2. Getting all the nets between the output pins of IC2 and the LEDs can be a bit tricky. You could, if you prefer, use a bus for this (see the earlier section in this chapter). The name of the bus needs to be something like “LEDS:CATHODE[1..10]” to allow the 10 connections to the LED cathodes to be made from IC2. The bus and direct-connection versions of the schematic are shown in [Figures 4-24](#) and [4-25](#). On balance, I think the direct-connection approach is clearer.

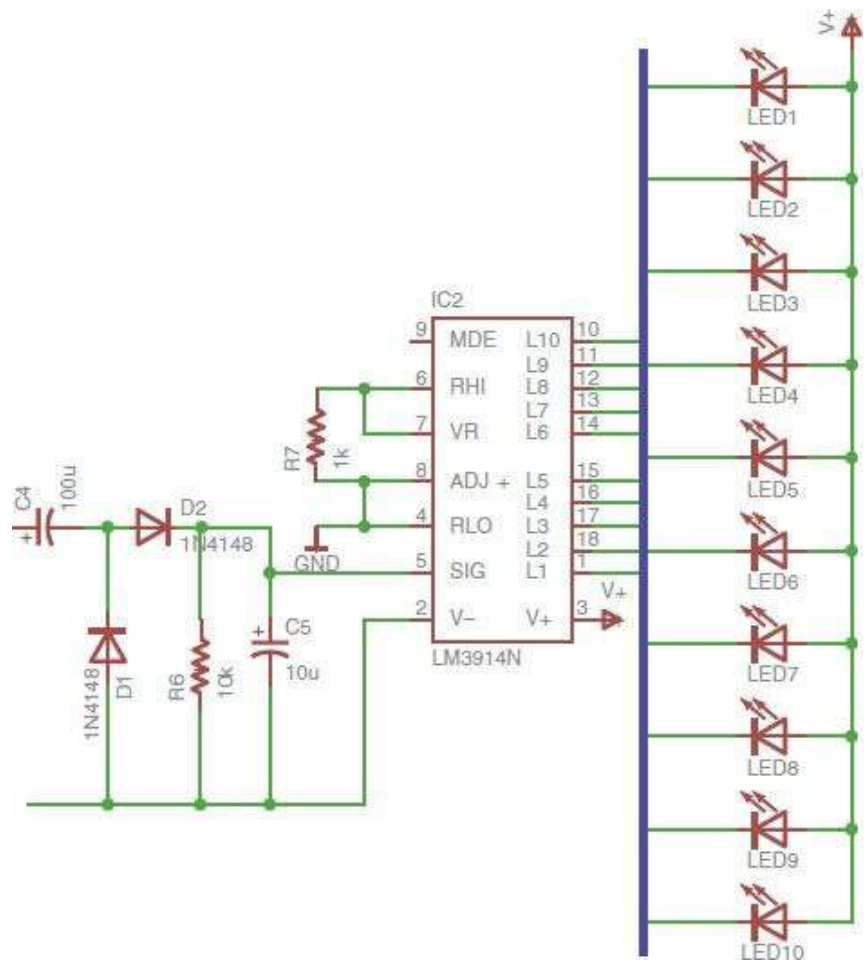


FIGURE 4-24 Bar-code section of the design—using a bus.

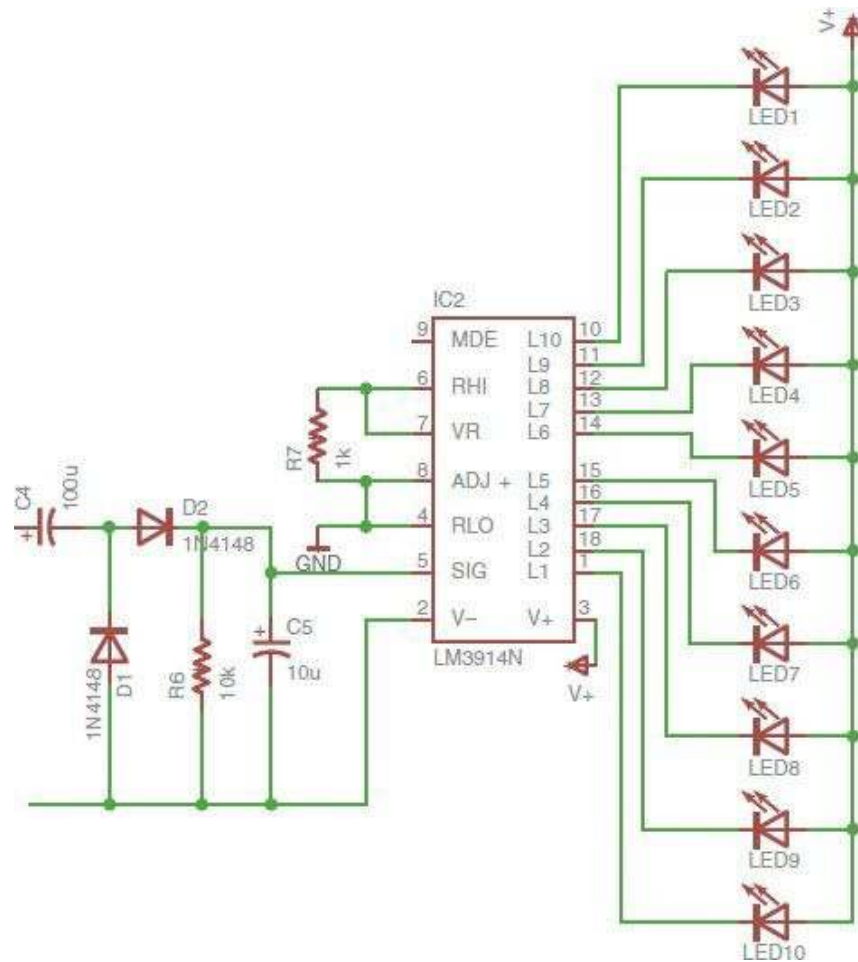
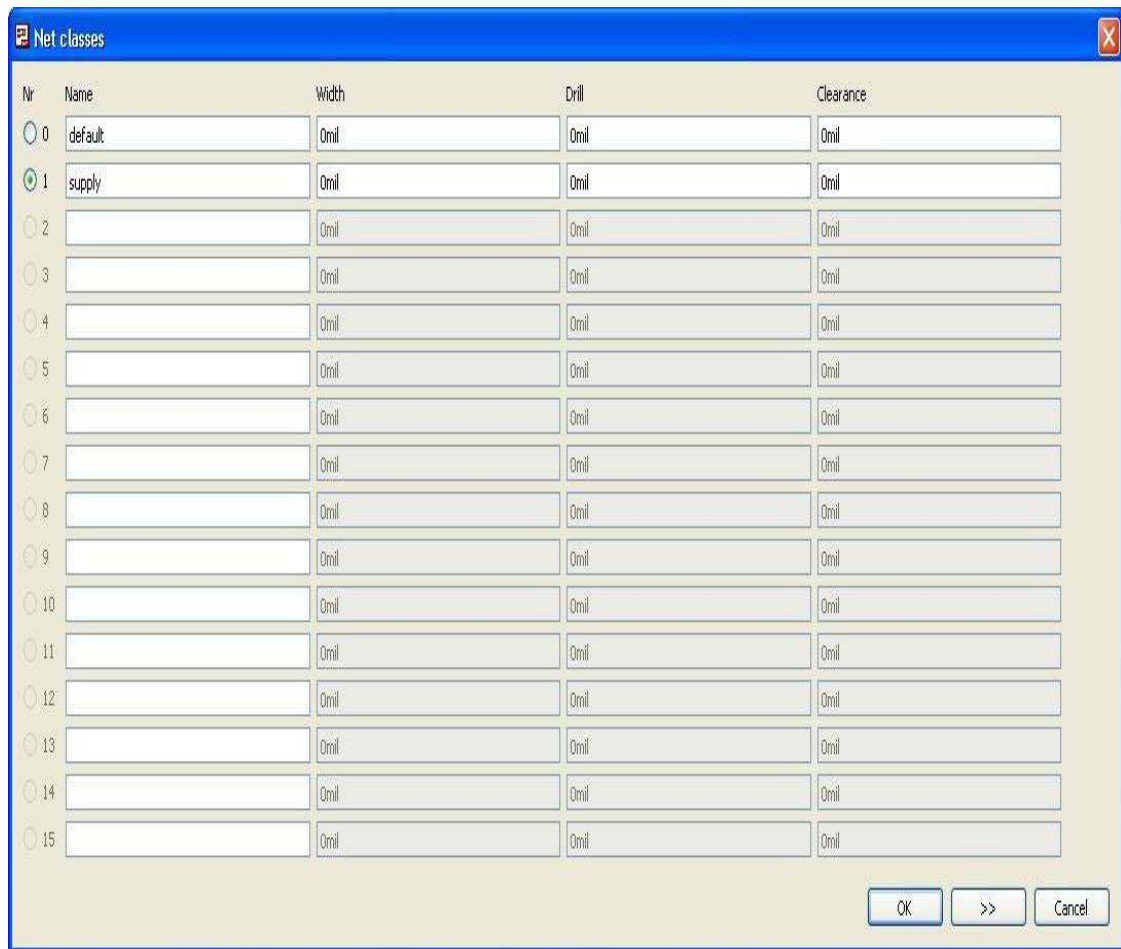


FIGURE 4-25 Bar-code section of the design—direct connections.

### Assigning Net Classes

This step is not essential, but it is a good idea. It allows you to specify what kind of net a net is. We only really have two types of nets in this design—power and signal—but you could imagine a design that controlled a 110-V relay and had some nets that were high voltage and high current. While this has no bearing on the schematic, when we come to lay out the tracks on the board, it would be useful to know that those tracks should be wide and well separated from other tracks. Similarly, it is common to use slightly thicker tracks for supply nets.

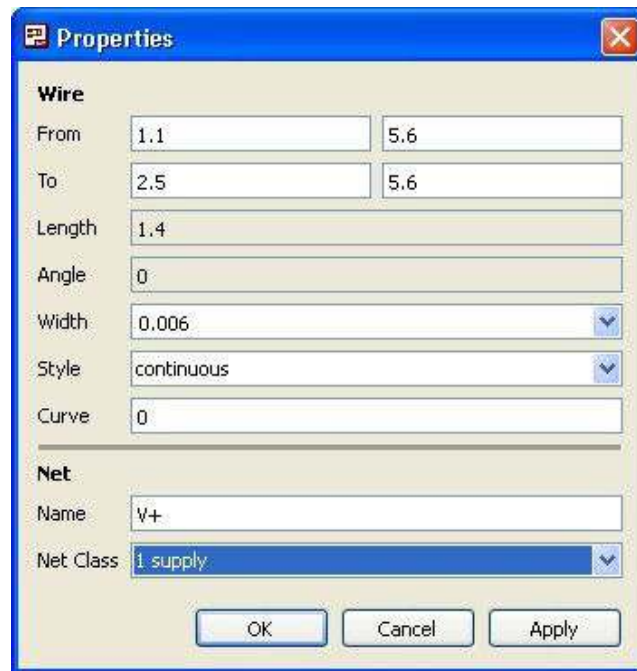
As the design currently stands, all the nets will be of a class called *default*. If you use the Information command and select a few nets, you can see the dropdown near the bottom of the window that allows you to select a class for the net. Currently, there is only one entry there (“Default”). To add another net class for supply, select the option “Net Classes..” from the Edit menu, and type a name in the second row (Figure 4-26).



**FIGURE 4-26** Defining a new net class.

We will return to the other parameters of a net class in [Chapter 5](#) when we come to lay out this board. For now, just add the name.

Now change the class of the GND and V+ nets to “Supply” using the Information command ([Figure 4-27](#)).



**FIGURE 4-27** Using the Information command to change the net class.

This completes the connecting up of the nets. The next step is to validate the schematic using the ERC.

## Running the ERC

Before we start on the layout, we need to run the ERC. This will tell us about any problems in the schematic. It is important to understand that the ERC does not simulate our circuit; if the design is wrong, it will not tell us that it is wrong. It basically just checks that there are no dangling connections that have been missed or any nets that run too close to each other.

It is surprisingly easy to create a schematic in EAGLE that looks like everything is connected just fine, but it turns out that there are some connections that do not quite meet.

Launch the ERC, either from the Tools menu or from the ERC command, bottom left of the Command bar. The result should look like [Figure 4-28](#). The ERC is very helpful because if you select an item on the list, it will highlight the relevant area on the schematic.



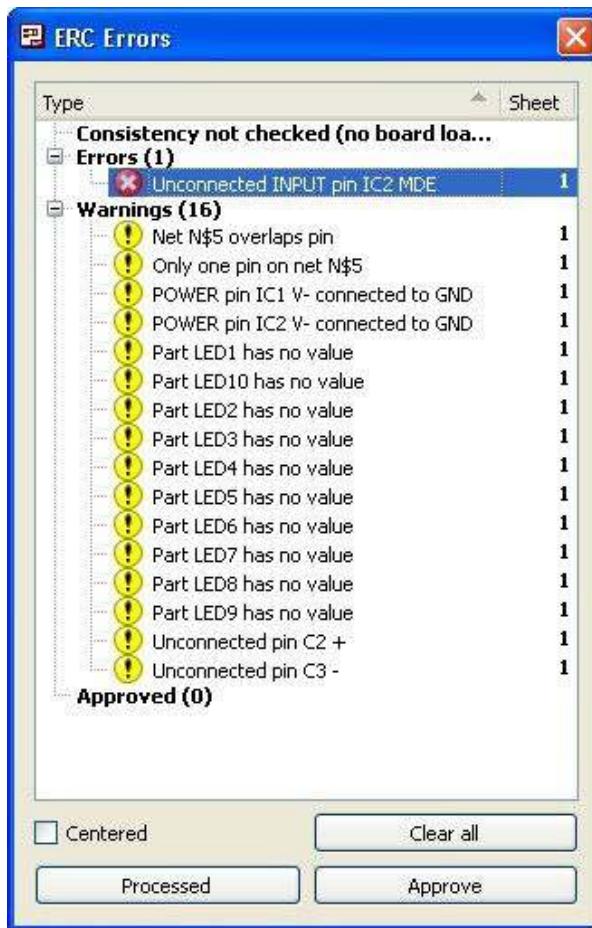


FIGURE 4-28 ERC results.

As you can see from [Figure 4-28](#), the ERC is only actually reporting one error. The error is that IC2 has an unconnected pin (pin 9). You may have noticed that pins 1 and 8 of IC1 are not connected to anything either, yet the ERC is not complaining about them. This is so because the part for IC1 from the library defines those pins as being allowed to be unconnected, whereas the definition for IC2 says that pin 9 must be connected.

Looking at the datasheet for the LM3914, it states that if you want the bar-code driver to operate in “dot” mode, then pin 9 should be left unconnected. Therefore, actually, the definition for the IC2 part is incorrect. In [Chapter 11](#) we will look at how you can copy parts from one library to your own library and modify them. For now, though, we are going to allow this error to stand because we know that it is not a real error but just a bug in the part definition.

In the warnings section, if you have been careful connecting up your nets, you will not have a warning like the first one: Net N\$5 overlaps pin.

To see what this warning refers to, click on the warning in the list, and a line will spring forth pointing to the part of the schematic where the problem is located ([Figure 4-29](#)).

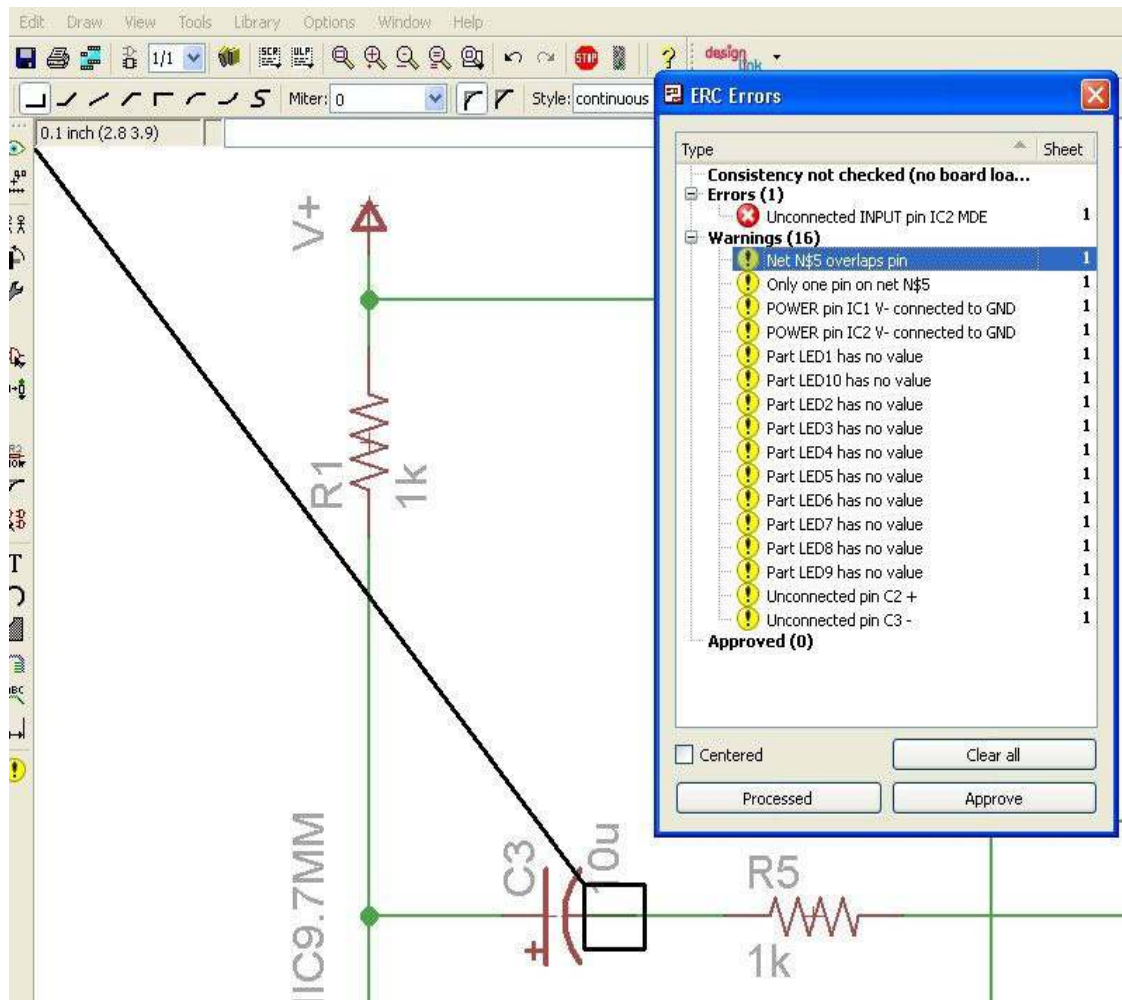


FIGURE 4-29 Net overlapping pin warning.

What this error means is that when I was drawing the net between C3 and R5, I started too far to the left on C3 and missed the lead. This means that C3 is not actually connected to R5, which is why there is a second warning that says *Only one pin on net N\$5*.

See what I mean about a schematic looking correct but not actually being connected up right? This is why it is very important to understand every one of the ERC results and either fix them if there is genuinely something wrong (as in this case) or ignore them if it is safe to do so (as in the case of the unconnected input on IC2). Once you get to laying out the PCB, it may not be obvious that there is a connection missing, and you may end up fabricating boards that are useless.

The best way to fix this is to delete the net and draw it again. Thus, close the ERC results window, select the Delete tool, click on the net in between C3 and R5, select the Net command, and then carefully draw the net in again from the right-hand lead of C3 to the left-hand lead of R5. Run the ERC again to make sure that it is fixed.

The next two warnings are just about the naming of power nets. Both IC1 and IC2 expect the negative supply pins to be called V-, and we are connecting both to GND. This can safely be ignored.

The next 10 warnings (one for each LED) just tell us that we have not given the LEDs values. They don't need values, although you could use the "Value" field to specify the color, say, but we will chose to ignore these warnings.

The final warning in the list shows that we have just plain forgotten to connect the top end of C2 to anything. This is easily remedied.

---

## Summary

EAGLE is a very sophisticated tool. Although we have built on the simplified schematic design process that we first started in [Chapter 2](#), there are still features of EAGLE that we have not used. We have, however, learned how to use all the features that you will need to use for a fairly straightforward schematic diagram.

In [Chapter 5](#), we will give an introduction to how to use the Board Editor in general and then work through the bar-code example laying out both fully through-hole and surface-mount versions of the PCB design.