Eagle PCB & DeskCNC

How to use Eagle to produce Gerber files suitable for use with DeskCNC

(a blow by blow account)

Version 1.01 - 13th Sept. 2004

Eagle/DeskCNC Notes

Contents Page number

Introduction

These notes are an attempt to give enough details to allow you to use Eagle PCB software to produce a set of suitable Gerber files to input to a small 3 axis milling machine running Deskcnc. This is by no means the only and certainly not the best way of producing a PCB, but it works well enough for me. I would really appreciate any comments or improvements/suggestions you may have. Although I am familiar with Eagle and DeskCNC, I do not profess to be an expert in either, but I have managed to overcome many problems that folk seem to have in trying to get these two programs to work together. If these notes seem lengthy, then it is because I mention many problems that you may find along the way. I've also included high resolution screen shots of appropriate settings, etc. If you follow these notes exactly, then you should end up with a machined double sided PCB, and enough knowledge to repeat the process for your own PCBs, or be able to adapt the method to your preferred way of working.

If you are the sort of guy who expects an easy solution, then you may as well read no further, but remember before you go, 'if this stuff was easy then everyone would be doing it'. For those who stay, you will need to get your hands dirty, and get into pulling G-code apart. There is a basic tutorial about G-codes at the end of these notes, in case you need it..

Be very aware that these notes only apply to the versions of software mentioned. In particular, the functionality of DeskCNC will vary in different releases, and what worked before, may not work now, and what works in the current release may well be screwed up in the next. I do not wish to go into the reasons as to why that is, since some folk seem to find it is a quite acceptable situation, but then, some folk think Microsoft is capable of writing robust software, too.

Finally, these notes are on a 'need to know' basis. I hope I have given just enough information to explain why things do not necessarily work out as expected, but of course, you can research and try things out for yourself, and find out more background details if you need.

The whole of this document is based on original work by the author. Any trademarks/names mentioned belong to their owners.

These notes apply to the following software versions only. (They may be OK for other versions, but I'm not testing it)

Eagle Version 4.13 (for windows) (upgrade at cadsoft.com)

DeskCNC Version 2.0.0.70 (upgrade at DeskCNC.com)

M$ windows 2000 operating system

Checking PCB Layout

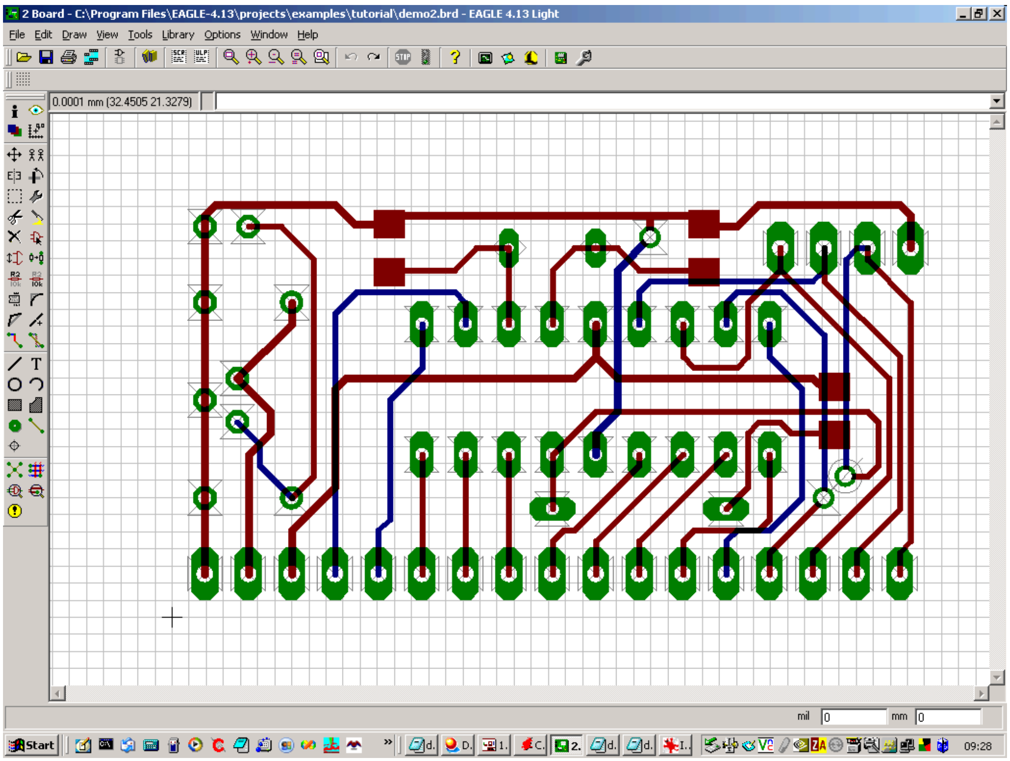
I normally use a photo mask/etch system for producing small single sided PCBs. I use an UV light source, same size transparency, and photo sensitive, pre coated copper clad board, and a high quality spray etch/wash unit. My usual PCBs are small, relatively simple, with wide traces, and most of the copper is left as a ground plane. I use the commercial 'small' version of eagle (Eagle light) which is limited to a PCB size of about 3inch by 4inch and no more than double sided tracks.

However, it is quicker, (or it was when I used the DOS version of DeskCNC) to produce one off boards by machining - there was no waiting for the etching tank to heat up, for example. A few years ago, I was writing Eagle ULP's to generate g-codes, with the help of the folk at Cadsoft in Germany. I spent a lot of effort in attempting to reduce the 'air cutting' time (with little success!)

It seems appropriate, therefore, to see if things have improved since then, since DeskCNC can now import Gerber files, and as Eagle can generate about 5 different versions of them, at least one should fit....

The example I will use is in the Eagle directory - examples - tutorial - and is named 'Demo2.brd'. This is about the sort of complexity I'm used to, and it may actually be useful to someone. However, I would normally lay this out as a single sided board, with the surface mount caps on the bottom, and use a few wire links if required on the top, but this layout is good enough for our test purposes.

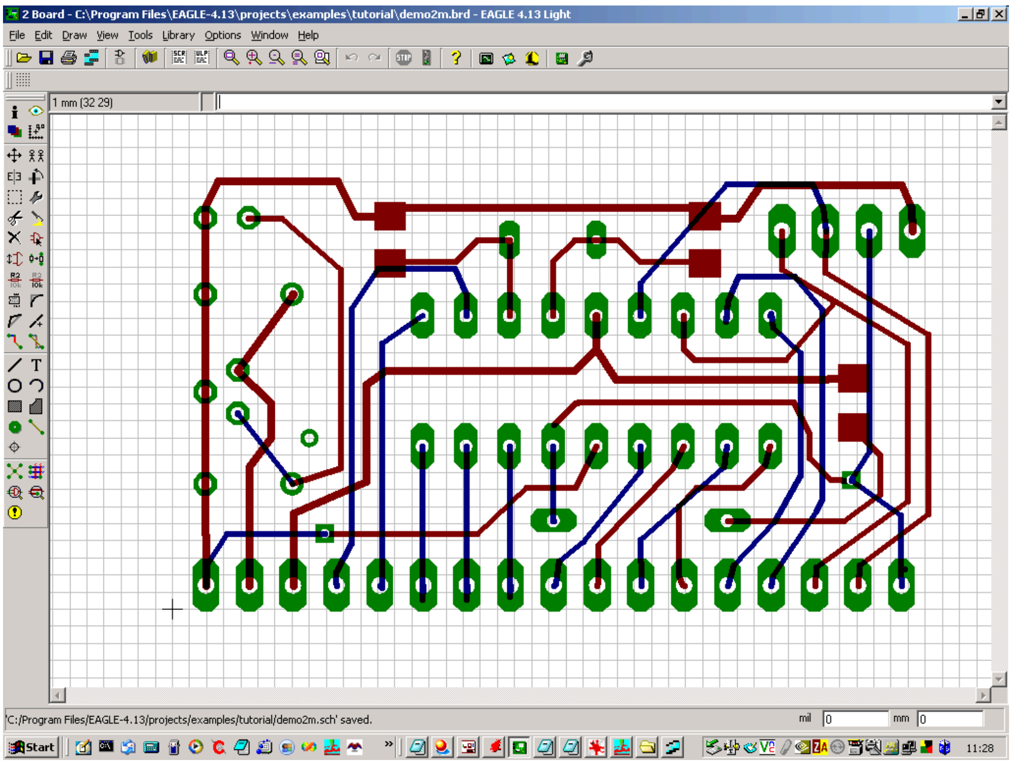
Open Eagle, and load the Demo2.brd (and the schematic). Now, what I propose, is using an 0.8 mm slot drill to drill the holes, and also mill the tracks. This saves time in changing cutters and solves the variation in width you get with using a 'V' cutter for milling, if the



PCB surface is not quite flat. I am not concerned at this stage with automatically drilling larger holes.

On a PCB of this simplicity, it should be easy to get at least 0.8mm spacing between pads and tracks. We need to be able to easily check the spacing (I know you could reroute with the 0.8 mm spacing parameter set, but for our exercise, we're doing it this way!) So, once the file is loaded, go to the view/grid settings and change the grid to mm and finest (0.0001) and set multiply to 10000, (leave alt as 0.0001).This will put a 1mm grid over our inch based layout, and we can see which tracks are likely to be closer than 0.8mm. (Please note that although the images within this document are quite high resolution, you may not see some details on a low resolution screen, and grids may well have lines missing. It may be easier if you can print this file, in colour)

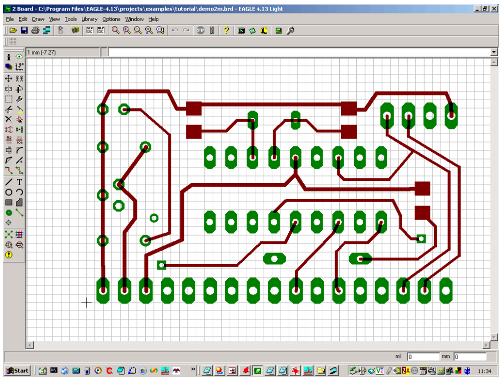
The first obvious problem, is that there will be no room for tracks between pads, and the four top layer tracks at the right hand edge are too close together. Using the move and rip up/reroute commands the PCB was re-routed as below, to give track spacing wide enough for the 8mm cutter, and this revised, slightly larger board layout was saved.



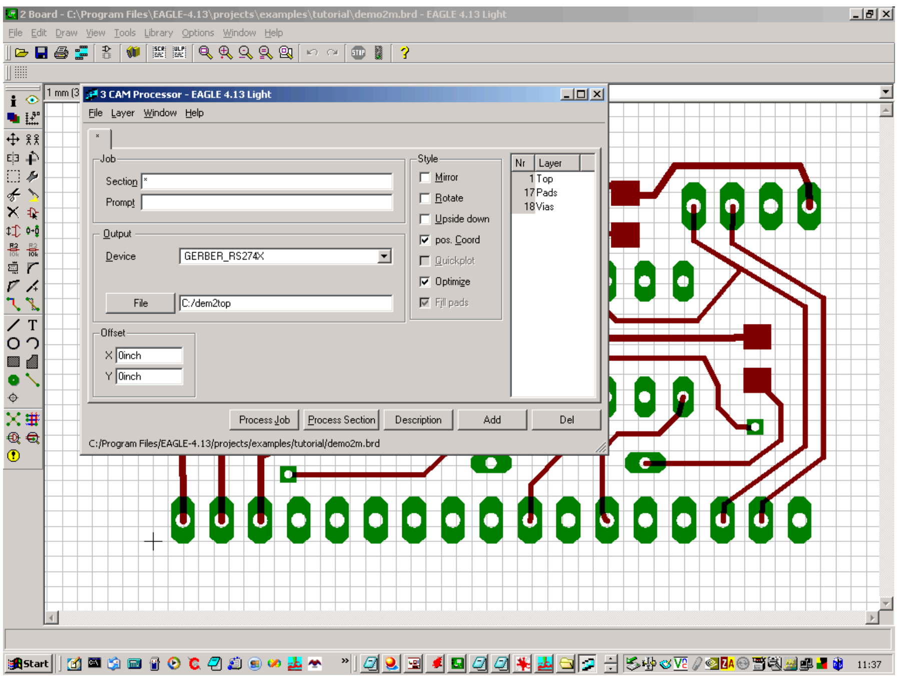
Producing Gerber Files

Eagle has five Gerber file production methods, excluding drill files. I found that none of them directly produced a file that can be imported straight into DeskCNC and give accurate results. None of them automatically give a .dat extension, that DeskCNC implies is required. Early Gerber files required an associated ‘wheel’ or aperture file. I could not readily see a way of generating this in Eagle, or anywhere else. However, by opening the files in notepad, you can see the relatively straightforward way that the data is laid out, and that the Gerber RS274X format includes the wheel information at the beginning of the file. A customer recently inquired about using this format, so I have based my tests on this. This does not mean that other Eagle Gerber formats will not produce better results, or produce them more easily. Please try them all for yourself, and let me know the results.

Anyway, we are going to first of all produce a Gerber file for the top layer of the PCB.

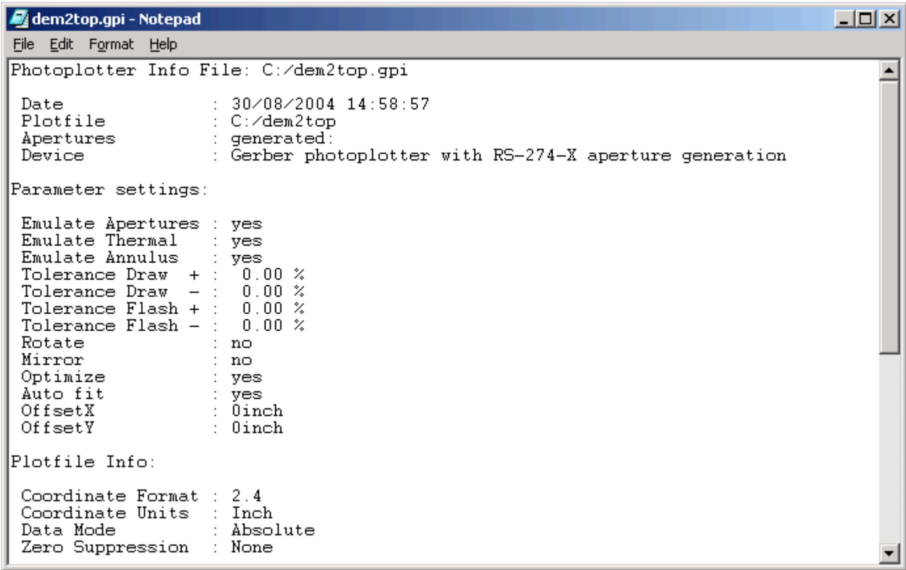


Using the CAM processor within eagle is relatively straightforward. Select the output device as GERBER_RS274X, and enter the file name to be saved, in this instance use ‘c:/demo2top’. Then select the required layers in the right hand column, so we need top, pads and vias. Check that no other layers are selected, so use menu ‘layer’ then ‘show selected’ will confirm this. Ensure ‘pos Coord’ and ‘optimize’ boxes are checked. Finally hit ‘Process Job’ to produce the Gerber file.



Editing Gerber Files

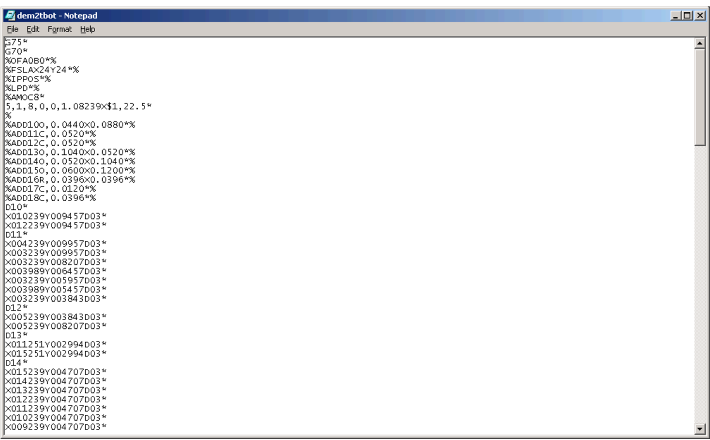
Now try and load the Gerber file into Deskcnc. Go to ‘File’, ‘Open Gerber’, open ‘Extended Gerber’, change to the ‘c: /’ directory, and you’ll see a file named ‘demo2top.gpi’. Open it, DeskCNC objects with an ‘access violation’ message (you will get plenty of these with DeskCNC!). Time to get your hands dirty. Open the same file in notepad. It is a photo plotter info file, not the one we want. We need one with a list of



co-ordinates, or something similar, the info file is basically a job description file.

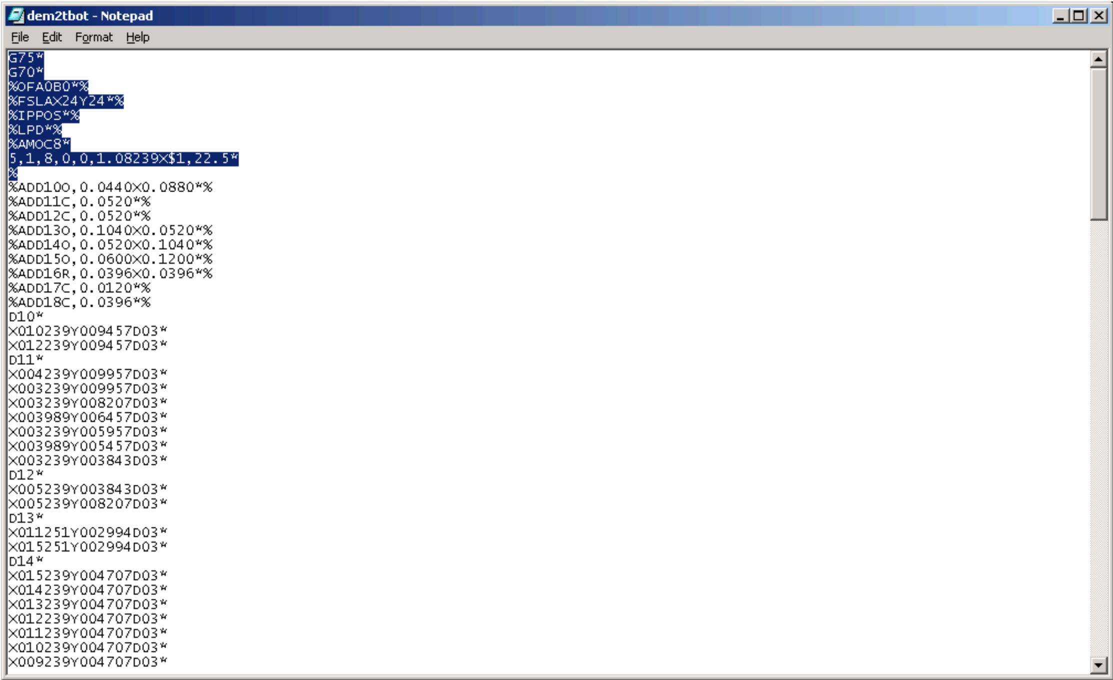
As well as being able to load all Gerber files, DeskCNC allows you to load Eagle Gerber files, (as well as others) implying a ‘dat’ extension. We haven’t got one of these either, so select ‘All files’ and look for anything starting with ‘dem2top’ (make sure you’re in the correct directory - DeskCNC seems to keep resetting things). There is a file named ‘dem2top’ with no extension. Loading that causes Deskcnc to object again.

The type of error message we are getting signifies that the DeskCNC program has encountered an error in the data that it doesn’t know how to handle. I will not comment further.



Anyway, have a look at this file in notepad, it should be similar to this, and it looks as if it may be useful, if we can find what is causing the problem with loading it into DeskCNC.

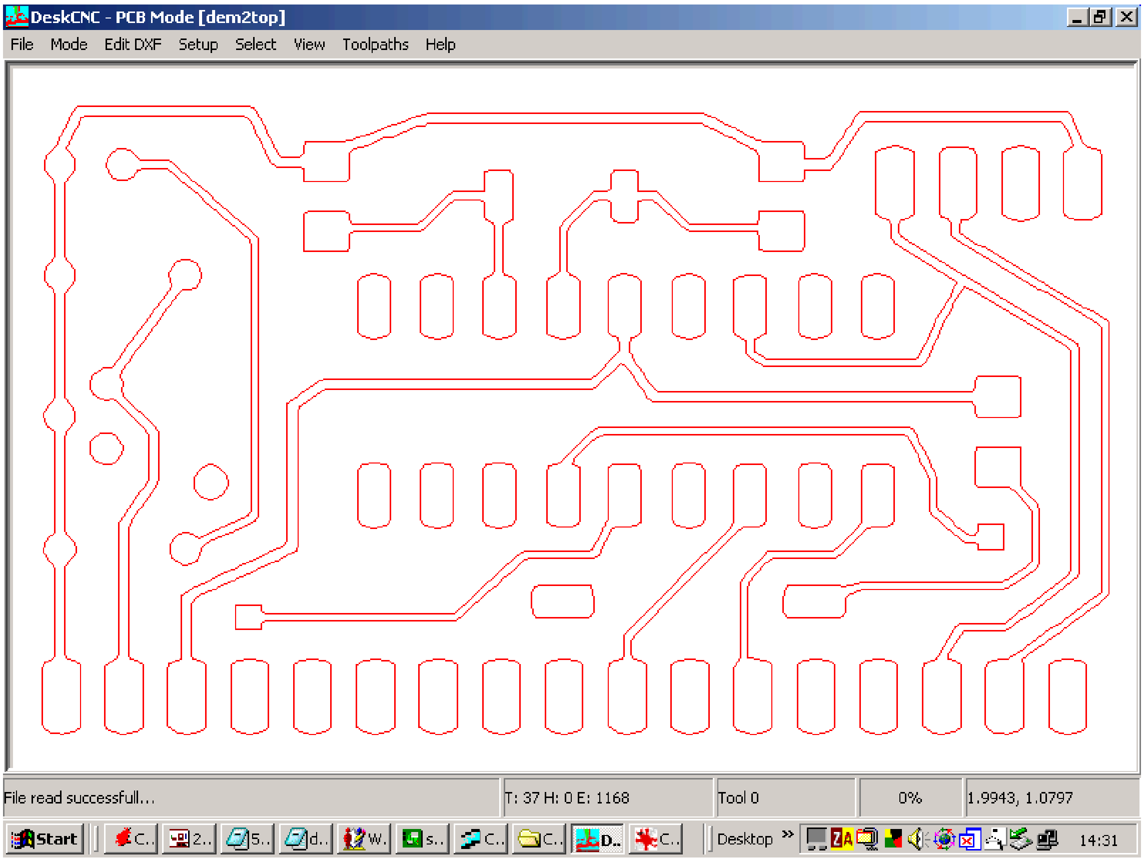
To save your time in messing around with various other Gerber files, notepad and DeskCNC, I have found that if the first few lines, down to and including the '%' are deleted, the resulting file will load into DeskCNC. It may not be necessary to remove all these lines, there could be a coded value missing, or whatever, but what I suggest, I have tried, and it works OK for me.



Loading Gerber file into Deskcnc

In the last section, I said that the edited file loaded OK. Well, what has happened to the pads on the left hand side? These are all on components, the vias added when I re-routed some of the tracks have come out fine. Is it something to do with the Eagle component library, maybe the components need 'smashing'? If you go to the Eagle readme file, they mention that some plotters do not know how to handle octagonal pads. These component pads are octagonal. Go to the 'Eagle.def' file (in the program- eagle- bin directory, find the RS247X section and comment out/uncomment the lines as Eagle recommends. Run the CAM process again, remove the first few lines (Notepad), load the edited Gerber file into Deskcnc, and hopefully you will have cause for a minor celebration.

We are now in the position that we can generate a Gerber file and load it into Deskcnc, and it looks OK.



Moving Around

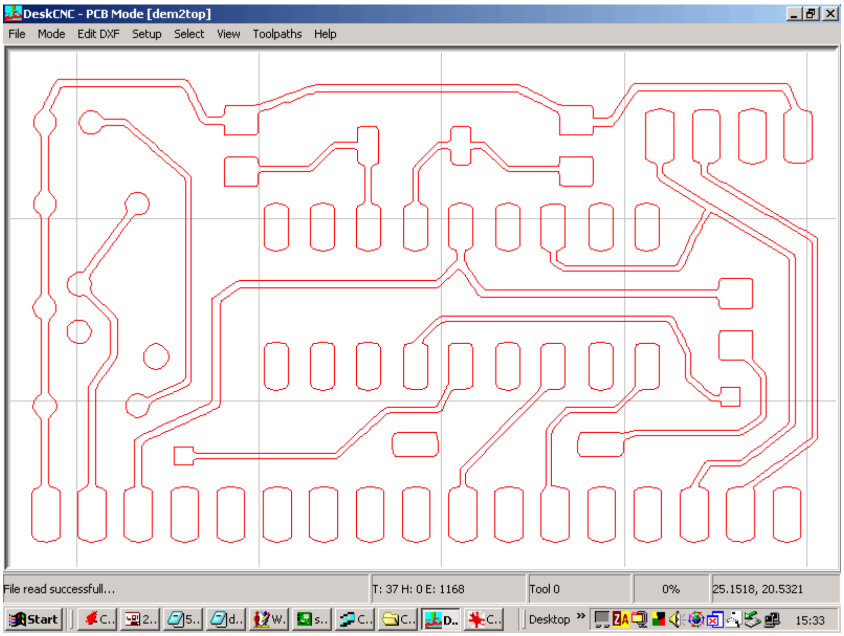
Of course, there is more to do. My milling machine is calibrated in mm, as are the PCB cutters, but the PCB layout is calibrated in inches. To change the image in DeskCNC to mm, simply scale it by 25.4. The procedure is 'Open DXF' (drop down menu) (what dxf, you might think? Well it seems the Gerber file has been converted, sort of, to a dxf format), then select 'Scale, Move, and Rotate', to open a window entitled 'Move, Scale, and Rotate' (why the difference?), then enter 25.4 in the scale entry field (note that the values in the left hand part of the drop down box can be more or less anything - this is flawed as it can show values left over from previous images, but it was fine, as far as I remember, in some previous releases), then press the 'Scale' button and then 'Cancel' (another idiosyncrasy - it should be 'close' as in 'close the window', 'cancel' usually means 'cancel the operation', also don't you love the non standard way of data entry in DeskCNC - you have to delete the existing value before you enter the new, since a new decimal point will not be entered). Then you need to press 'view', then 'zoom all' to see it.

This would be more or less all we would need to do, if it were a single sided board we were dealing with, but as it is a double sided PCB, we need to get some common reference point, that will be visible in both top and bottom layers. There are a number of ways this can be achieved, some more suited to multi runs of pre-cut PCB's, others suited to one offs.

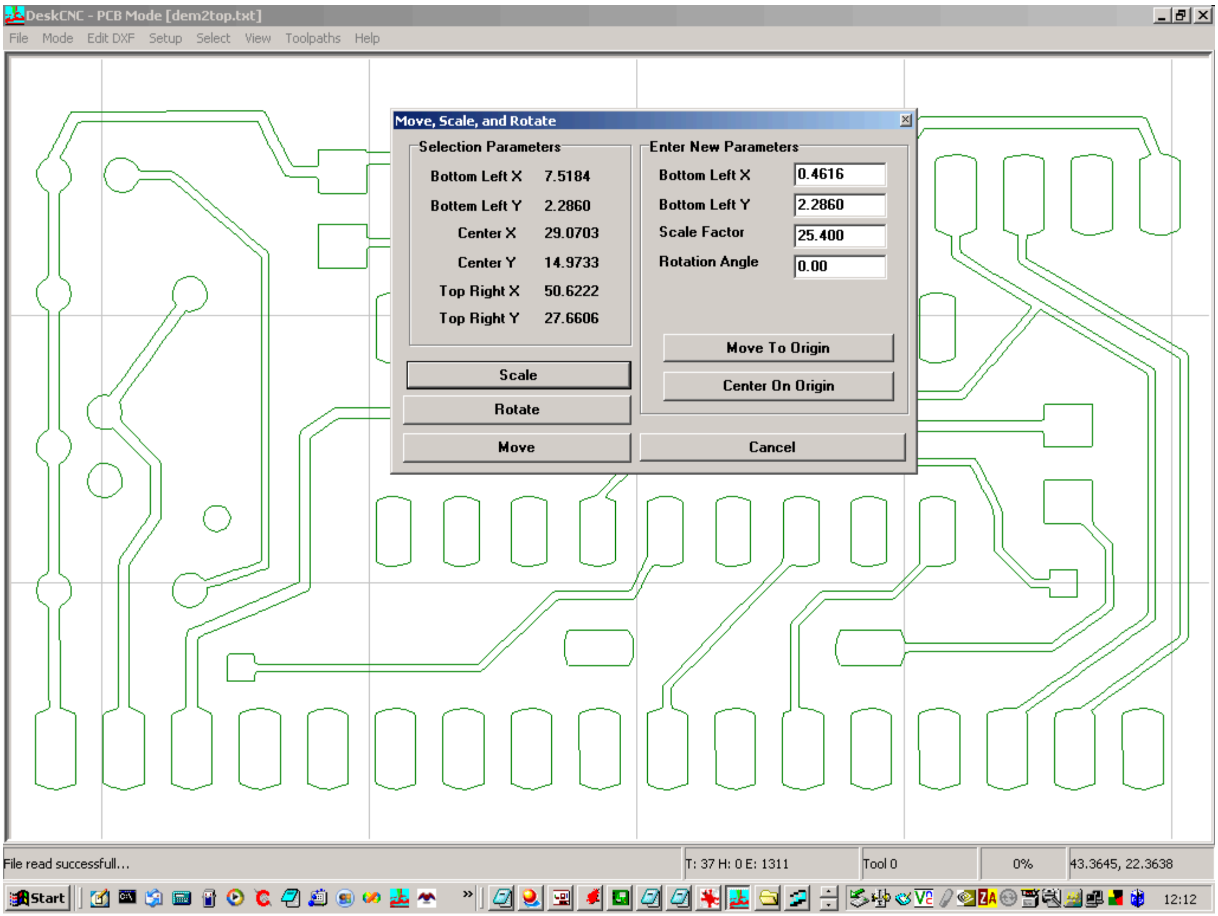
The method I suggest is as follows.

Place a pad in a known position on the PCB, preferably towards one edge or corner. Make sure the pad is big enough to give plenty (i.e. a mm or so) clearance round its central 1mm hole. This will be the origin of your machine co-ordinates. You then manually drill, or as the first of the drill operations, machine drill that hole. It is usual to drill the PCB first, it avoids the small pads tearing away, if the drill is a bit blunt. You can then use double sided adhesive (carpet) tape to fix the PCB to a backing board on the mill table, or use some other clamping method. You can use a locating pin through that hole (provided you arrange not to machine drill that location, and you make sure that the clearance or tool paths will avoid the pin when machining). If you do not reset the machine origin, simply turn the PCB over and position the same hole over the pin. You need a straight line reference parallel to the x or y axis for this to work, and the PCB must have one straight edge to line up with it. If you do not use the pin, then you can reset the machine origin, in one axis only, when you turn over the PCB.

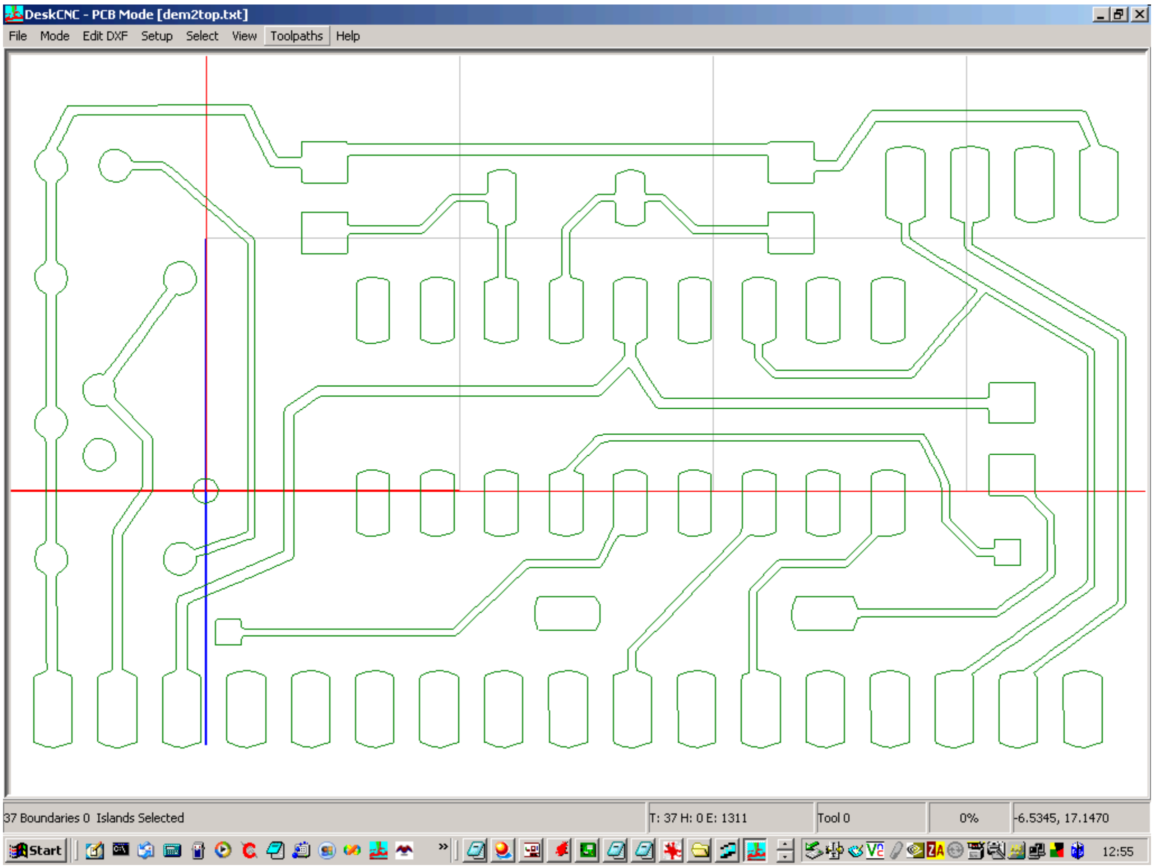
So, having proved the Gerber stuff worked, I went back into Eagle and placed the origin pad, at a location x8, y10mm (from bottom left hand corner of the board), saved, edited, loaded, and scaled, giving the resulting DeskCNC screen shot below.



We now need to position the origin pad, at the origin of the DeskCNC workspace, so we open the ‘scale, move and rotate window’, and enter the required offsets, -8 and -10. No, it didn’t work for me either!

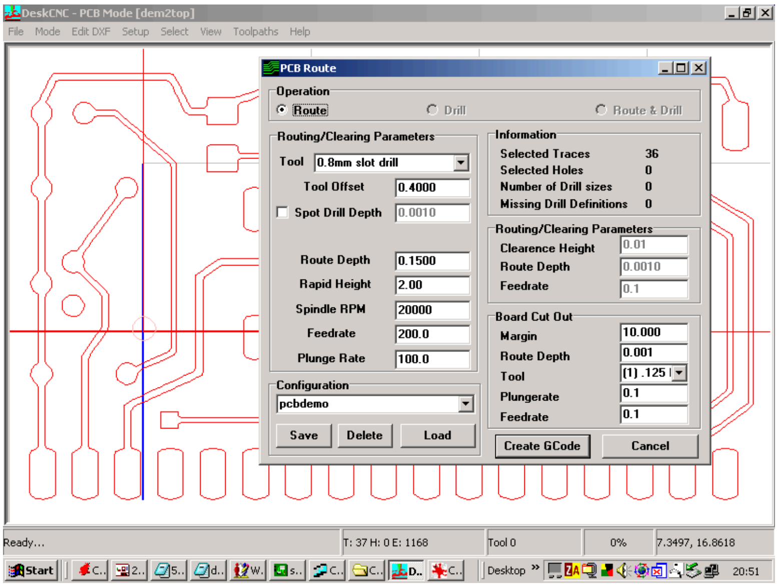


But persevere, and you can position the origin pad over the origin of the workspace. Like much other stuff in DeskCNC, it all works in an unusual manner, and it seems to vary from release to release. Anyway, you can get there if you try, as the next screen shot shows (NB, do not pay too mach attention to the colours, these screen shots were taken from two different computers, with different option values.)

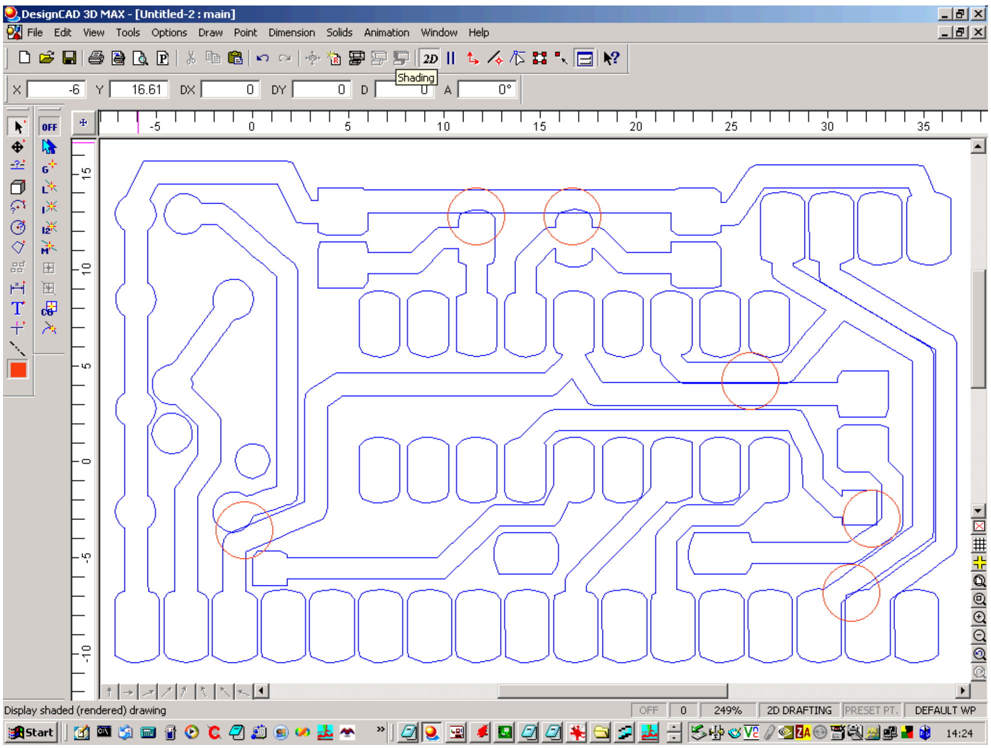


Saving as DNC

The next stage is to prepare the g-code (.dnc) file. Usually, if using a dxf file, you have to pre select the regions you work with. As this was initially loaded as a Gerber file, everything is selected. However, we do not need to mill out the origin locating pad, so deselect it, then open up the ‘Toolpaths’, ‘Route’ menu item.. As we are using a 0.8 mm slot drill, the offset will be 0.4mm. Adjusting the offset value can give thicker or thinner PCB tracks and pads, of course.

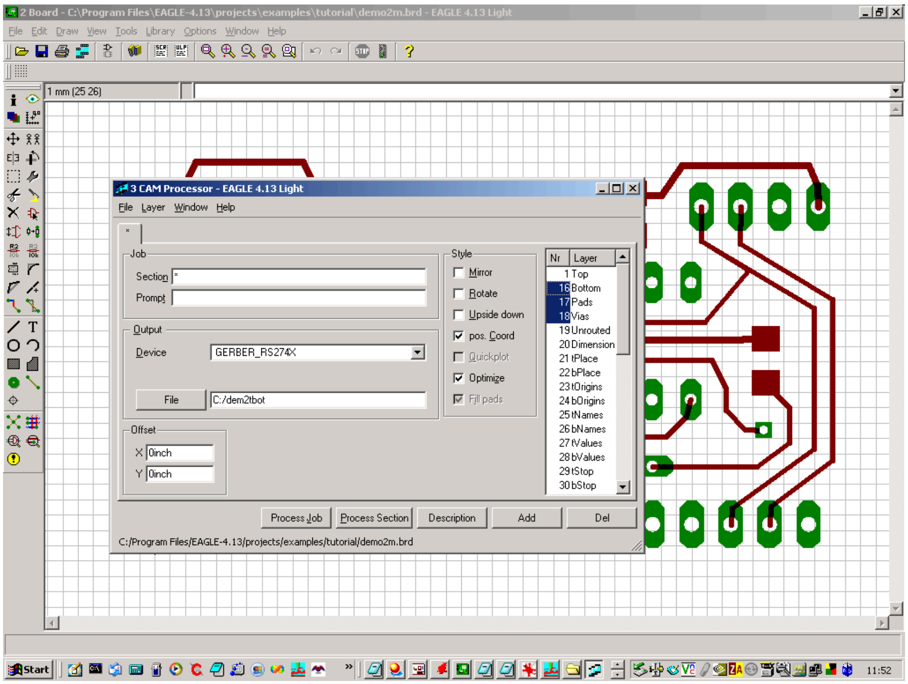


Have a good look at the resulting cutter path drawing. You may well find that some cutter paths cross over each other. That means the adjacent track/pad will be machined away. I have highlighted some of these errors in the view below. (dxf imported into cad program so I could highlight these track work errors with red circles.)

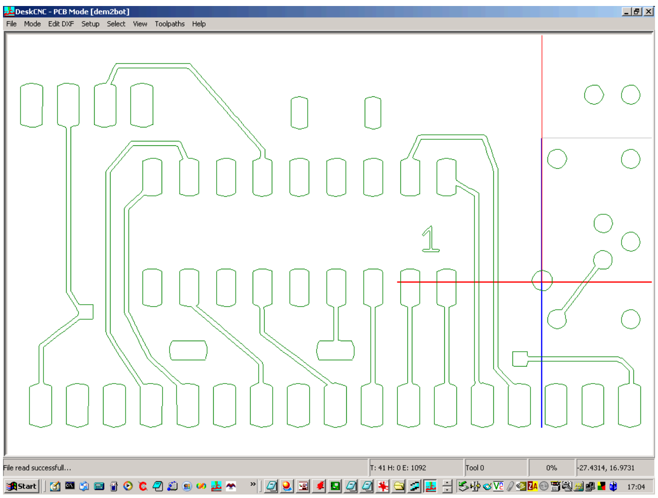


So, either ignore these errors, and suffer the consequences, or go back to Eagle and slightly adjust the layout, and regenerate the Gerber file, and so on. By now, you may think this is all rather tedious, but if you keep the Eagle program, notepad, and DeskCNC open, it does not take too long to make these changes, and later on I may mention a few more rapid methods. Also Eagle auto router can space track work and pads etc. to your specifications, so it should be easier starting a PCB layout from scratch.

Having saved the top layer Gerber file, we perform a similar work pattern for the bottom



layer. Remember to change the Gerber file name being saved in Eagle! You can give a unique file extension name if you wish, Eagle does not seem to alter that for the data files.



Now, we need to generate a milling .dnc file for the top layer, and also one for the bottom layer. We have previously seen how to do that in DeskCNC, so repeat the method for the top layer only, and save the file.

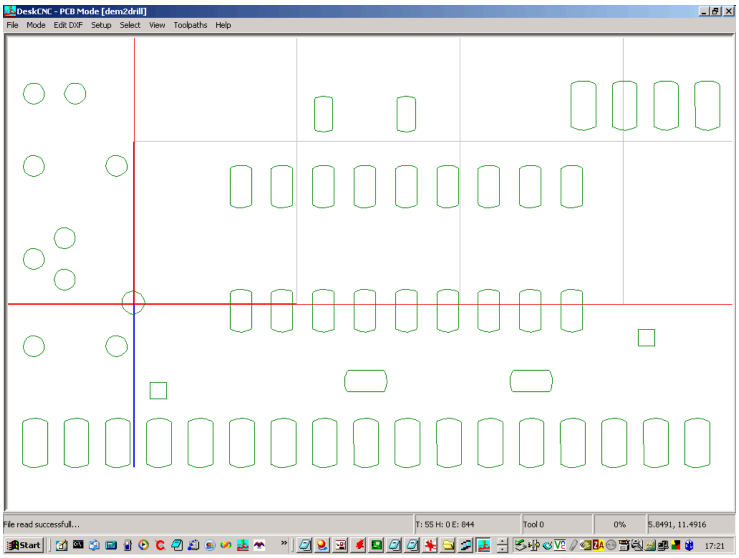
The bottom layer has to be mirrored, since the Eagle layout is viewed from the top of the PCB, but the bottom layer is milled from the bottom side, not through the PCB!

I used the DeskCNC mirror function in the 'Edit DXF' menu. You need to select 'Mirror Board X'. It makes little difference if you do the mirroring before or after the scaling. You then need to position the origin pad over the DeskCNC origin again, most of the board will lie to the -ve X side of the origin. Then generate and save the .dnc file as for the top layer.

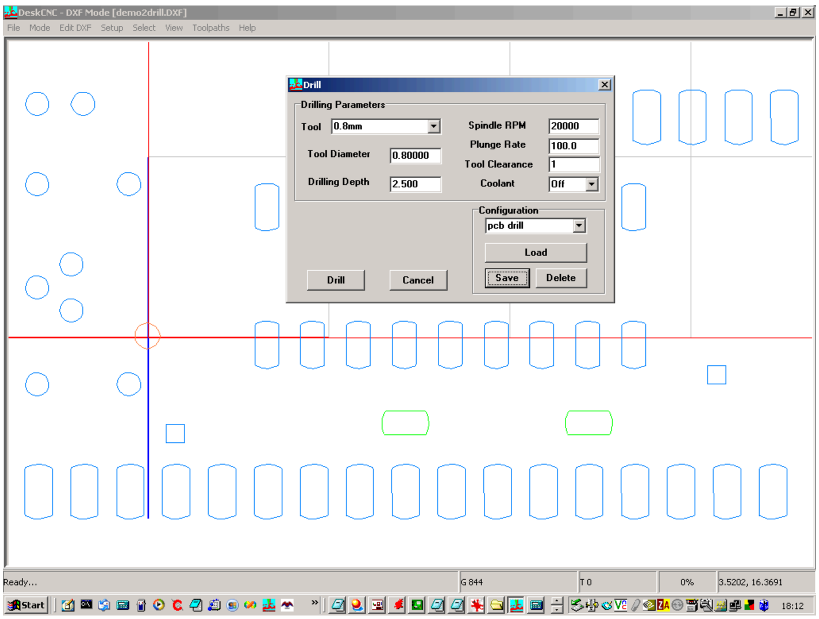
Drill File Generation

We need to now consider the hole drilling and board cut-out if required. In fact, the board cut out is often more easily done by hand writing G code, or most likely by simply guillotining the material after milling the tracks.

In previous versions of DeskCNC, the excellon drilling method worked fine, you can try it for yourself in this version. Also there is a Gerber drill routine in Eagle PCB. However, I decided to use the very useful Deskcnc drill facility, which will drill at the centre of any closed circle. In fact it seems to drill at the centre of any closed shape, such as the oval pads. (In some circuit components, the drill hole is towards one end of the oval pad - This Deskcnc method will drill them central to the pad). So in the Eagle Cam processor, select the pads and vias only, and save as a Gerber file. Remove the first few lines, as for the other Gerber files, and load into DeskCNC, scale and move so that the origin pad is over the origin in DeskCNC. This view is shown below.

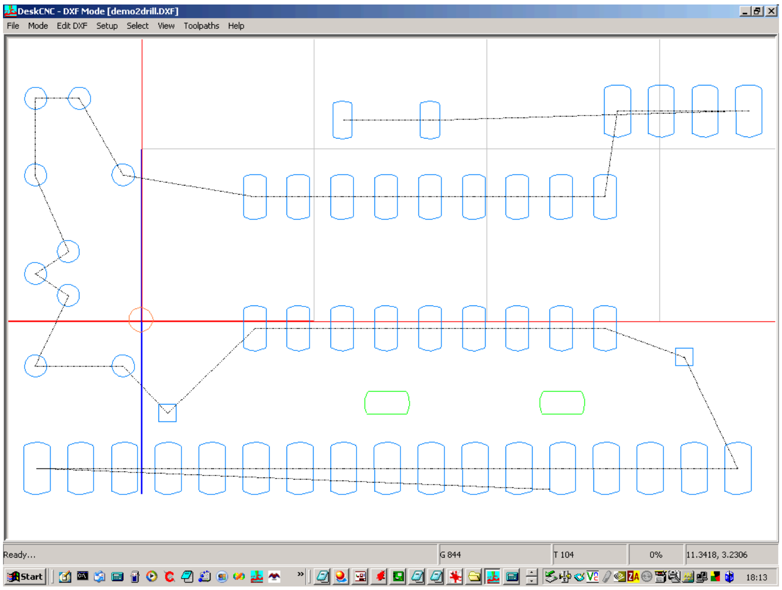


.The drill function menu is not accessible if you have loaded a Gerber file, so you need to save this file as a .dxf file and reload it. Go to 'File' menu, then 'Save Geometry DXF', then 'File', 'Open DXF', and reload it as a dxf file. The 'Drill' function can now be found under the 'Toolpaths' menu. Do a 'Select', 'Select All' (deselect the origin pad, if you will be using a locating pin to align the PCB) then choose 'Toolpaths', 'Drill'.



Complete the boxes, set the drill depth to be about half a millimetre or so more than the PCB thickness, and hit the 'Drill' button. Then select 'File', 'Save Tool paths CNC, and save the .DNC file.

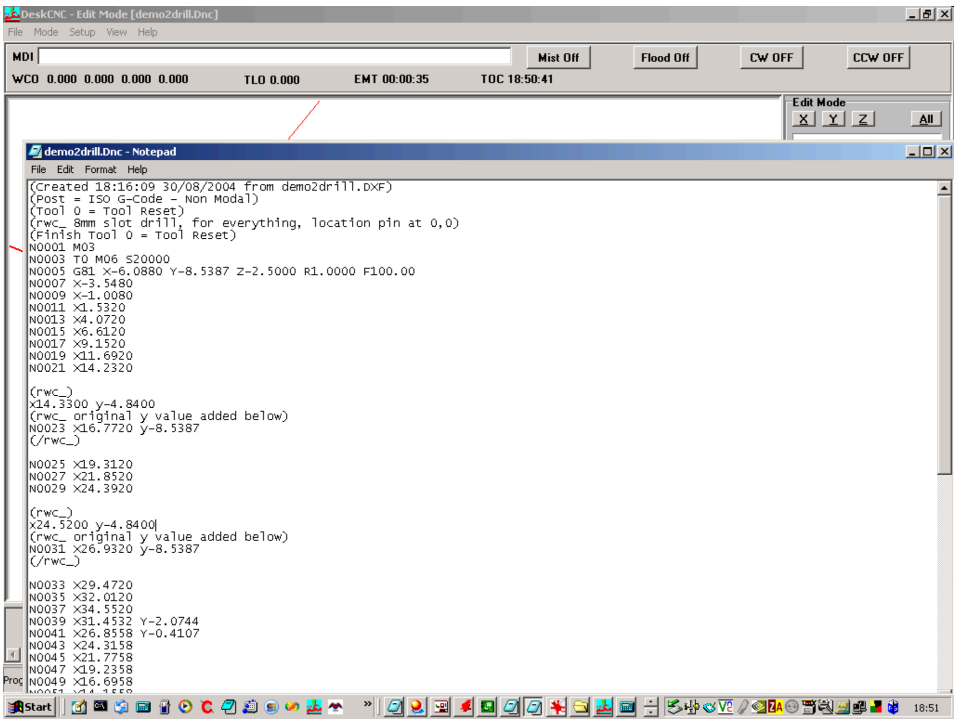
But, there is a problem. Did you notice the two diode pads were not drilled? Also it does not seem possible to select them (other than to delete them - don't do that!). It seems there is an error in the way that Eagle represents these, they do not act as complete closed entities, certainly not the same as other pads. A simple way out would be to return to Eagle and drop a couple of pads on these, prior to selecting the drill CAM process within Eagle. Maybe a more pragmatic solution for a one off PCB would be to simply drill them by hand, but now is a good time to do a spot of G-code editing.



Position the cursor over the centre of the left hand diode pad, and make a note of the xy co-ordinate values. You could zoom in if you want more precision, but I make it to be X14.33, Y-4.88. Get the co-ordinates of the right hand pad, X24.42, Y-4.88.

Open up the saved drill.dnc file with notepad, and you can see how the G-codes are constructed for drilling (the Appendix to these notes explains the basics of G-codes).

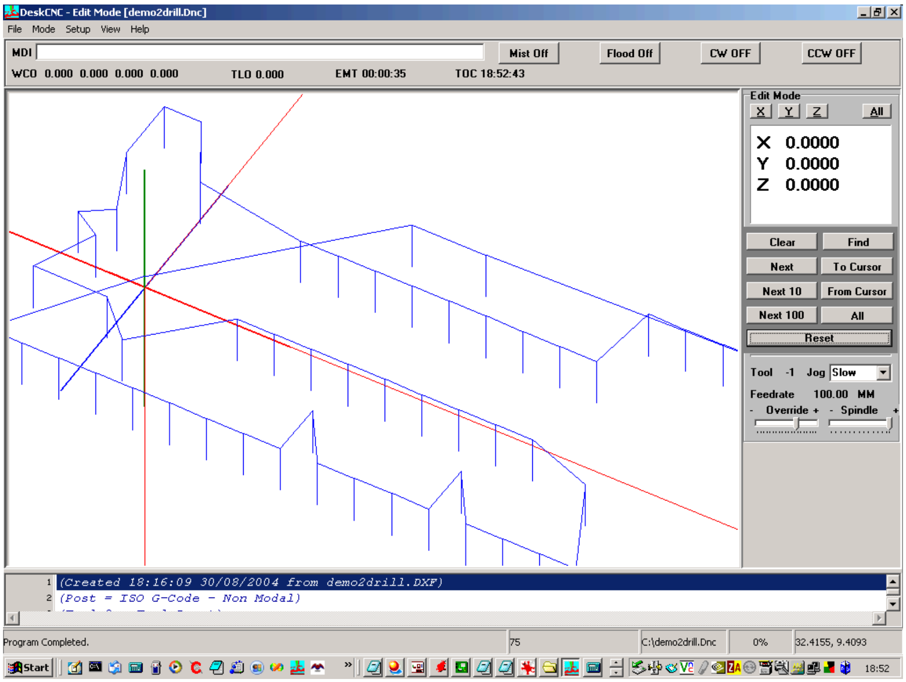
What we want to do, is add in our two diode pad drill locations, near to existing points, so that the machine is not wasting time in cutting air. I have shown how I modified the file the altered code lies between (rwc_) and (/rwc_) comments. I also took the opportunity to add in a line describing the drill diameter. Instead of using Notepad, you could use the editor built into Deskcnc, but I find the 'right click/save as' etc. in Deskcnc tends to be



forgotten. As you may guess, I use notepad a lot to edit G-codes, and write html, etc., but you can use whatever plain text ASCII editor you wish, but never use a word processor, since you will forget to save the file as ASCII.

Anyway, since the DNC file is modal, the earlier values carry on to the following lines, so we can not just put in the xy co-ordinates of our points, we have to reset the y value afterwards. Looking down through the code, we can see that the drill will start at the bottom right corner, and go along the long row of pads in the plus X direction, so it should be apparent why I've added in the code at the particular places.

Load your modified .dnc file into Deskcnc. The mode will probably switch to 'Edit Mode'. Select 'View', then 'View All', and 'Dynamic 3D' (you may have to switch dynamic 3D a few times to get a 3d view. The 3D view will readily show the location of the drilling points. If you step through, you can see that the holes will be drilled in a reasonable sequence.



One other Deskcnc gotcha. I have a number of PCs with DeskCNC installed, not all are at the same version number. When installing a new version, sometimes I do not correctly install the milling machine setting values. If I display this .DNC file in a machine set to four axis, then I get no visible changes in the y axis screen plot direction, the holes look as if they would be drilled in a row parallel with the x axis.

However, we now have three .DNC files, one for drilling, one for milling the bottom of the PCB, and one for milling the top layer of the PCB.

Fixing PCB Material

A relatively easy method of fixing items for light machining is to use double sided adhesive tape. There are many versions of these, you need to experiment with what you can buy locally, or otherwise acquire. As well of different adhesive strengths, the substrate will have considerable effect on its use. Some tapes increase their bond strength over time, and you may find it difficult to remove the machined item, and impossible to remove the glue residue without damaging the item. Obviously, a thick paper or cloth substrate, with a thick coating of tacky adhesive will not allow ultra precise machining. A thinner tape may not conform to the surfaces, or otherwise stick so well. Often, it is not possible to reposition items with tape, you have to line it up first time. Another aspect, is that many tapes will not stick to surfaces where cutting oil has been, and obviously do not stick to dust, swarf , waxy or powdery surfaces, such as mdf. Also, they may not perform well if cutting oil is used. (but don't use cutting oils on PCB material - or mdf!)

One of the best ways of holding PCB or other sheet material is to use a vacuum table. You can readily make a table, from two sheets of 8mm aluminium or hard plastic, say. You will have to acquire a vacuum pump, however. You can not easily use a sacrificial backing plate with a vacuum table, so you will be unable to drill holes straight through the PCB, and milling the second side will be tricky, since the vacuum will not be formed due to the grooves in the first side.

If you are using precisely cut, pre-cut PCB material, then you can build a frame on the backing sheet, using plastic strips, or even off-cut strips of PCB material, fixed around the edges of one of the PCBs as a pattern. This will stop the board moving sideways on the table when being machined. If the PCBs are small and flat, then a couple of small clips at the edges can stop it being lifted up.

Large sheets of PCB are usually not flat enough for successfully holding with clips, you need to use double sided tape, or a vacuum table, unless you have the type of milling head/cutter that incorporates a depth stop.

In all cases, you need to ensure the PCB is not resting on swarf or dust. Any clamping forces need to applied at least 5mm from the edge. I have seen a type of flat vice with about 4inch shallow jaws, with a v groove for clamping the edge of material. These may be OK for holding engraving plastic, but they will cause the PCB material to bow.

For double sided PCBs, you need to ensure you can identify the origin from both sides of the PCB. If the pre-cut PCBs are precisely cut, then you can use a corner. If you are cutting from a roughly guillotined sheet, say, then drill a small hole, and refer the origin on each side to that . You also need to ensure that when the board is turned over, the axes stay the same, i.e. The board is not rotated. This can be achieved by having a straight edge, or other line parallel with one of the machine x or y axis, and a corresponding straight edge on the PCB blank.

Machining

Now is the time to test out our efforts. You can repeat my method, or invent your own

I clamped a sacrificial sheet of perspex to the machine table for use as a backing plate, since I do not want to drill into the machine table. I normally have an inch thick sheet of hard PVC permanently fixed to the table, but I've been 'messing around' a lot recently, and the perspex was nearby. It is not really thick or stiff enough, and it bowed slightly, which will cause errors in the milling depth, but I am not concerned with that for this test.

I then milled a slot across the backing plate, in the X direction, by lowering the cutter, and jogging at my medium speed setting. This milling machine was home/scratch built, to my own design. I am well aware of its limitations. I have recently changed the cutting head from a 500w Ferm router to an Axminster AMT, a sort of Dremmel type tool, but with better tool holding for small cutters then the router provided. Also the AMT allows far better visibility of the work piece.

The machining sequence used was drill, mill top layer, turn PCB over, mill bottom layer.

I cleaned the backing plate and PCB with a solvent free cleaner (Ambersil FE10) then applied a strip of double sided tape to the back of the PCB area being machined. I then carefully stuck it down, such that the straight edge of the PCB lined up with the slot in the backing plate, and in a position such that turning it over and lining up the origins would be possible, without moving clamps etc. I had previously marked with a pencil, the approximate position of the origin.

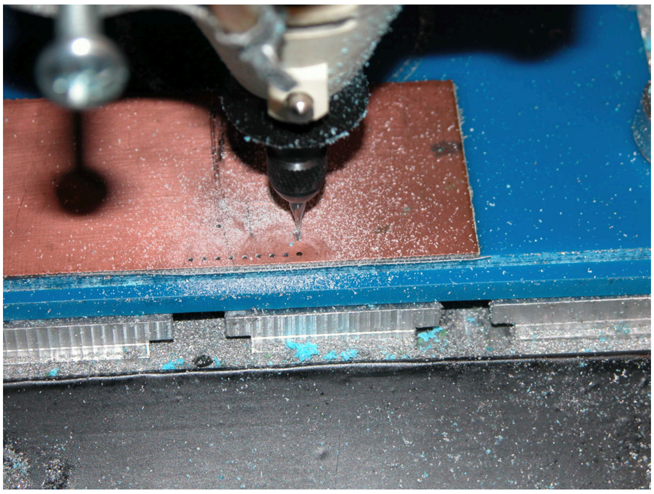
There are a number of ways of setting z0 to the surface. I simply jogged down the z axis until the tip of the 0,8mm cutter touched the PCB. I did this more or less in the centre of the PCB work area. I have a work light, which can be positioned near to the work surface, to cast a long shadow of the cutter across the PCB. A small z movement makes a large shadow change, I simply jogged down until the shadow end touched the cutter, which happens at the instant the cutter touches the PCB surface. Feeler gauges, cigarette papers, electronic devices can be used. On a bigger mill, I can jog down until the stepper stalls. For PCB work, it is probably best to turn on the spindle and jog down until it just starts cutting. The really critical distance, is the value to the bottom of the top layer of the copper, i.e. the depth of cut.

I had set the milling depth of cut to 0.2mm in the routing software generation window. This was a guess. However, given an accurate machine, I should have jogged down, with the cutter running, jogged around a bit, and set the z value such that I had the depth of cut I needed. I should have then raised the spindle by 0.2mm, and set the z0 origin to that level. Or, I could have used a micrometer, or whatever, to find the thickness of the copper. Anyway, the objective for me was to show that I, and hence you, could produce a PCB with the drills lined up in the pads on both sides, I was not actually concerned too much in producing a usable PCB, but I expect you are, so set the Z origin properly.

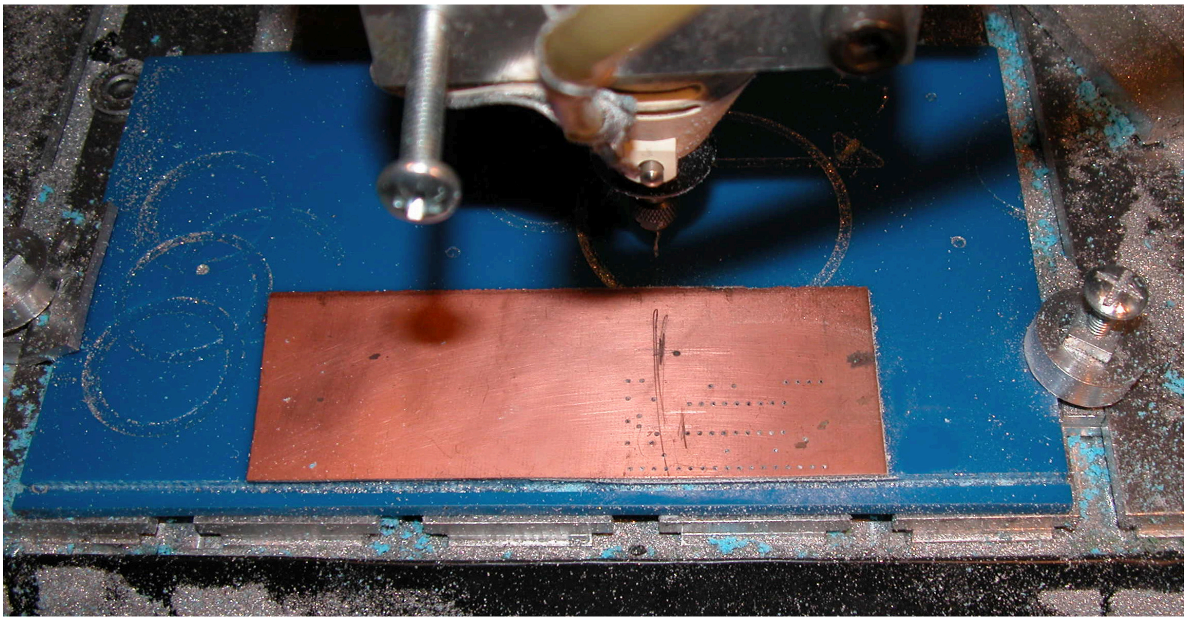
I then moved the router over the origin pad position, and set the X and Y origins.

I loaded in the drilling.DNC file, pressed 'Reset(Estp)' a few times, and then 'Go'.

This image shows the orientation slot milled in the blue perspex. The double sided tape protruded at the straight edge side of the PCB, which would normally make precise lining up difficult. However, it all looked good to me. The bottom row of holes were being drilled in a straight line, and everything sounded fine, but maybe the z feed rate was too fast. Anyway, the drilling pass went without mishap.



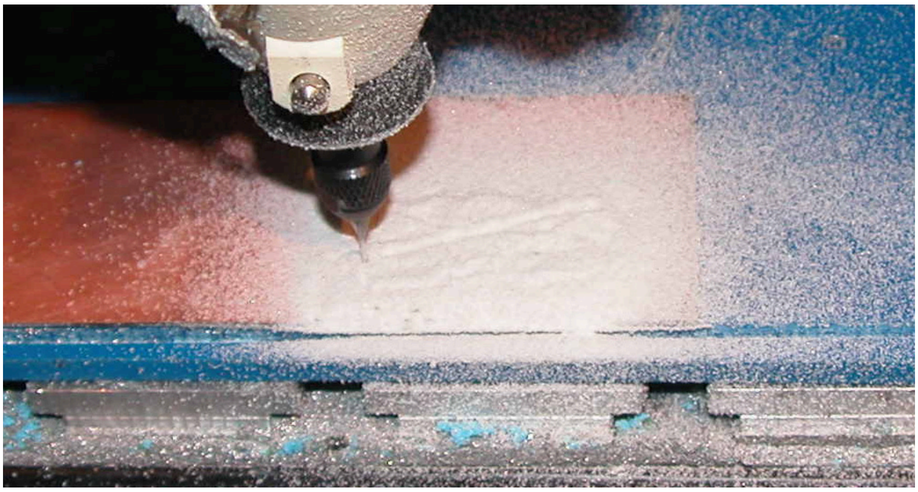
The picture below shows the situation after all the holes were drilled. You can see the crudely clamped piece of perspex, and the fact it has been used before. You can also see that it would be tricky to fit a locating pin into this material, so I didn't!



Now, although I jogged the cutter away after drilling, so I could take the previous photo, I made sure I did not reset any of the origins. I jogged the cutter up well clear of the board before moving horizontally.

The top layer .DNC file was loaded next, pressed 'Reset(Estp)' a few times, and then 'Go'

The picture below is the result of milling the top layer of the PCB. Of course, a blower should be used to keep the swarf clear of the cutter, but This was interesting, in that an image of the PCB layout was replicated in dust.!



To mill the bottom side of the PCB, required its removal form the perspex backing plate, turning over, and sticking back down again.

The cutter was jogged up, and then jogged back, out of the way of the work area.. The swarf was brushed off, well away from the work area, and the PCB was prised from the perspex sheet.. The old double sided tap removed, and a new piece applied to the top surface, ensuring there was no burrs or swarf remaining. The PCB was turned over, and its straight edge lined up with the groove in the perspex sheet, before finally pressing down to fix it in position.

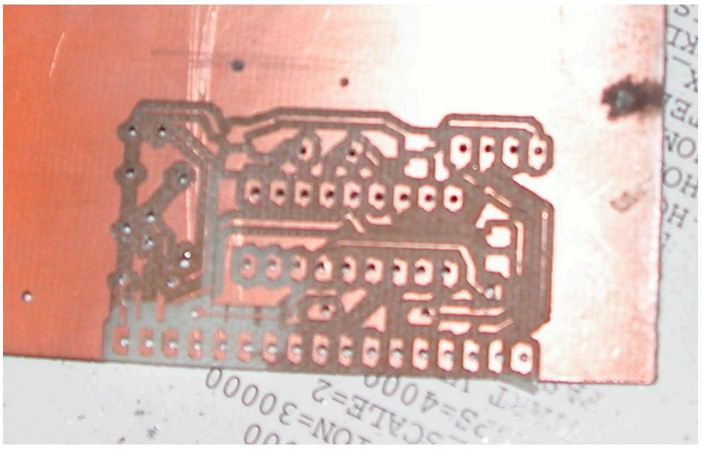
The cutter was jogged forward, and then located over the origin hole. It was then only necessary to reset the X origin. The Z origin was also checked, but was found to be good enough.

The bottom layer .DNC file was loaded next, I pressed ‘Reset(Estp)’ a few times, and then ‘Go’ to mill the bottom board layout.

Jog the tool well away before prising off the PCB, and that’s it. Simply shear the PCB to size.

Results

The photo below is of the board produced as described above. Look at it. carefully, and



we can work out what is wrong. In fact, although the PCB is unusable, there is very little that will need correcting, in order to produce good results. We have certainly solved the Gerber program conversion problems, and are only left with tool height setting and machine adjustment errors.

Perhaps we should start with the good points.

The holes are correctly drilled, and measuring their pitch, the scale conversion from the eagle inch based diagram to the metric machine worked fine, as it did for the track work milling..

Some of the track work, in particular the long straight tracks, is almost good enough.

The holes are nominally in the centre of the pads.

But lets also look at the faults, with the view of being able to eliminate them.

1) the milling is too deep. More care needs to be taken in setting the z origin, and in calculating the best cutting depth. A 'V' profile cutter may give a better shaped wall to the milled slots, but there can be width problems (alluded to earlier).

2) the holes are not quite central to the pads. Look at the IC socket drillings. The bottom holes are placed to the left of the pads, the top row is placed to the right. Invariably incorrect backlash settings cause this sort of error. The cutter is being pushed from the right to the left in drilling the bottom row, and from left to right in the top row. Without checking, easy enough by single stepping through the dnc file in DeskCNC, also it may be that the pad outline milling is in the opposite order.

3) Not shown here, but I compensated for the 0.2mm depth of cut when I milled the bottom side. However, after the first few tracks, it was apparent that the cutter was not even scratching the surface of the PCB, never mind cutting through the copper layer. The z axis was gradually creeping up. Again this is a backlash error problem.

I have mentioned how to adjust backlash in an appendix.

In the meantime, remember **'if this was easy, everyone would be doing it!'**

Summary and Improvements

In my opinion, this method works quite well. The errors shown are all concerned with the machine tool, or the setting up of the work piece, and not the software/data conversion process. If you set your machine accurately, you will get accurate results.

However, now I've shown how to do it, can it be streamlined, into a more 'turn key' type of solution?

Well, I've always found that I end up manually routing the PCB tracks in Eagle. I can achieve 'better' results with small single sided boards manually, than by using the autorouter. For new boards, it will be easy enough to allow a wider track spacing than in the demo2.brd used in this example. It could be fun for larger/more complex layouts, however.

If you open the Eagle CAM processor, and then select 'File', 'Open', 'Job' you will get a list of predefined CAM jobs. Open the gerb274x.cam file. You should be able to edit the various sections to process the board and produce the Gerber files you require. Save it with a new name, of course. (In fact, when I edit any file in order to make a new one, I usually save it immediately, before I make any changes, but with the new name - this prevents accidentally overwriting the original file.). This will simplify the work flow, since you do not need to manually run three separate CAM process jobs.

If you understand the technique, then a Perl script, or other method could be used to automatically modify the Gerber files (removing the first few lines, which I did manually using notepad.) You could perhaps do these two steps in one with the aid of an Eagle ULP. You may find one of the other Gerber file types works in Deskcnc without this editing. You could also combine the drill and top surface file into one, saving a load file function in Deskcnc.

For a one off PCB, it is quicker to not use the locating pin, just line up the origin, as I did for this example. However, for a short batch run, using the pin will save time. If you will be more than a casual user, then two locating pins may be a better solution, a single small clamp can hold then hold the PCB down in position, so the double sided tape need not be used..

Use a substantial sacrificial backing plate, one that will not warp or bow.

Email me with your results, and thoughts, please

In the meantime, remember **'if this was easy, everyone would be doing it!'**

The G-Code Interpreter

DeskCNC supports the following G and M Codes

G0 rapid positioning

G1 linear interpolation

G2 circular (clockwise XY Plane Only)

G3 circular (counterclockwise XY Plane Only)

G4 Dwell

G10 coordinate system origin setting

G17 xy plane selection

G20 inch system selection

G21 millimeter system selection

G40 cancel cutter diameter compensation

G41 start cutter diameter compensation left

G42 start cutter diameter compensation right

G43 tool length offset (plus)

G49 cancel tool length offset

G53 motion in machine coordinate system

G54 use preset work coordinate system 1

G55 use preset work coordinate system 2

G56 use preset work coordinate system 3

G57 use preset work coordinate system 4

G80 cancel motion mode (including any canned cycle)

G81 drilling canned cycle

G83 chip-breaking drilling canned cycle

G85 boring, no dwell, feed out canned cycle

G86 boring, spindle stop, rapid out canned cycle

G87 back boring canned cycle

G88 boring, spindle stop, manual out canned cycle

G90 absolute distance mode

G91 incremental distance mode

G92 offset coordinate systems

G92.2 cancel offset coordinate systems

G93 inverse time feed mode

G94 feed per minute mode

G98 initial level return in canned cycles

G99 R-point level return in canned cycles

M0 program stop

M1 optional program stop

M2 program end

M3 Spindle on Clockwise

M4 Spindle on Counter Clockwise

M5 Spindle Off

M6 tool change

M7 Mist Coolant On

M8 Flood Coolant On

M9 Coolant Off

It is not my intention to explain in detail the above codes, but hopefully I can give you enough information to get you over the initial stages of manually editing the g-code (.dnc) files that Deskcnc uses in its machining process. There is plenty of detailed information available on the Internet or elsewhere, and the best way for you to learn is by experimenting in altering your existing code or producing new.

A common way of producing g-codes within Deskcnc is by inputting a dxf or other drawing file. The resulting codes may need editing, in particular if you have appended a number of files, so it is essential that you have at least a basic understanding of g-codes.

Each line of code is to be considered as a block. This block contains at least one letter followed by a numeric field which refers to the value to be assigned to that letter or function. Comments can be added and are enclosed in brackets. You can relatively easily write correctly formatted g-codes, or edit existing g- codes with a standard ASCII text editor, such as windows notepad.

Spaces or tabs are ignored, but their use increases readability. Blank lines are also ignored. The code is case insensitive, upper or lower case characters are translated the same. If you append a number of files, you will find that the machining stops when it hits the first M2, so you will probably want to delete these from the files, (leave the last one, however) and possible the preceding and following G0 x0 y0 z0 blocks or whatever.

Line numbers begin with an N. Within DeskCNC this line number is not referred to or otherwise used, but it helps you to trace errors or edit the code, or track the machining process (Note that the line highlighted in the small edit window at the bottom of the machine screen is not the line being machined at the moment.)

Many codes are modal, that means they are active until they are changed, even outside of their block. For example the line of code g01 Z-5.00 will set the z axis to -5.00. Subsequent moves such as g01 x5 y25.44 will move the cutter from where it was to x5, y25.44 at a depth of 5.00. (It is usual to set the top surface of the workpiece as Z0, thus any cutting into the work will be given by negative Z values.)

The feed rate is also modal, and like other commands it applies to its current block and subsequent ones. Thus G01 x5.0 f150 will move the cutter at a speed of 150 to a new position at x5.0 subsequent g01 moves will be at the same speed, until another ‘f’ value is entered. Feed rate does not apply to G00 moves, of course.

For circular moves, G2 and G3, as well as the X,Y end of arc coordinates there will either be an I and J value, or an R value. The example below demonstrates this use. I, J is the distance from the start of the arc to the centre, or R is the radius.

```
N10 G00 z2.00
N20 G00 x20 y20
N30 G01 z-5 f200
N40 G02 x30 y10 I0 J-10 f300
N50 G00 z2.00
```

This will move the cutter above the work surface, then go to x20, y20 at rapid speed, then feed down at a rate of 200, cut a clockwise arc of radius 10 (centred on x20,y10) to a position x30,y10 at a feed rate of 300 then raise the cutter to clear the work piece again. The line N40 could also be written as N40 G02 x30 y10 r10 f300.

M6 will normally stop the machine to allow a tool change. You can write your own macro code to move the tool to the appropriate position, (see the software help documentation). The results of M commands for coolant are brought out to terminals on the controller board, and you can use relays or whatever to switch coolant on or off, or use these commands to control other functions.

The modal drilling canned cycle is entered in the form

```
G81 x10 y10 z-15  r-10 f75
X20 y20
X30 y30
G80
G00 z2
```

This will drill three holes, at x10,y10 x20,y20 x30,y30 to a depth of 15. The r value is the start depth of drilling, so the above code would apply to a surface which had already been milled out to a depth of more than 10 (assuming surface was at z0.) The feed rate (75) will only apply to the drill macro, and the machine will revert to the previously feed rate setting after the holes have been drilled

G83 is the chip breaking version, it will withdraw the drill to clear swarf, and is useful for deep holes. The code looks like

```
G83 x10 y10 z-25 r1 q4 f75
```

This is similar to code G81, except it will withdraw the cutter every 4 units of drilling, i.e. It will go to x10,y10,z1 then go to z-3, withdraw, go to z-6, withdraw, go to z-10, withdraw, and so on. This is often referred to pecking, for obvious birdlike reasons.

Remember to cancel the canned cycles with code G80.

It is suggested that you experiment with the codes that you are likely to use, so that you gain a full understanding of how your machine behaves. It is simple to write fragments of code, or alter existing code with the editor, in the machine mode, and immediately test the effect by running the code. The results can be seen on screen, provided the controller board is powered up, you need waste no material in testing.

Backlash Settings

In a few words - **Backlash settings do not work**

A few more words - Backlash settings have never worked in DeskCNC, and it is unlikely that they will ever work properly.

I first noticed the problem in January 2004, and have been trying to persuade Carl, the software author, to sort it out. However, he prefers to mess around with twain input, laser scanners and other novelties, rather than get the fundamentals functioning as advertised.. I guess its about the usual standard for American software.

Backlash compensation worked fine in the DOS version, so the requirements are known. Basically there is a conflict of interest. For continuous contouring, required when cutting foam with hot wires, and maybe for wood machining, you need to keep the cutter travelling at nominally the same speed, you do not want to stop at every change of circle quadrant, so the backlash compensation steps are ‘trickled in’ over the next movement.

For traditional milling work, in metal, accuracy is important. You need to stop the axis and feed in the backlash compensation steps, and then start milling in the new direction. If you do not do it this way you will get circles with flats, or steps or other errors, for example. Look back through the DeskCNC forum for folk getting this type of error. Look at the answers given. Surely its not difficult to say ‘ sorry, backlash compensation does not work’.

I do not want to go on to more general software design issues, but my views are that it is time for a complete rewrite of Deskcnc, with a firm understanding of what it is trying to achieve, and some formal quality controls adopted. However, due to market pressures, and other priorities, this is unlikely to happen.

The situation is, that if you enter a value for backlash steps, it may actually increase your errors in some circumstances, and the amount will vary depending on what version of Deskcnc software you are using.

The best solution I can offer is to set the backlash values to zero, and reduce the backlash at source by using anti-backlash leadscrew nuts, etc. Despite what some folk would have you believe, there must be backlash for anything to move. You need to just try and get it small enough so that it doesn’t show. All large, expensive, professional machine tools allow for, and use, software backlash compensation.

If you are retrofitting Deskcnc to one of the low cost Chinese imported type milling machines, you may find that the backlash is about 0.5mm, and due to the lack of adjustments, it will be difficult or expensive to eliminate it by mechanical means.

But always remember **‘if this was easy, everyone would be doing it!’**