

# Welcome to PCB-GCODE-WIZARD

Copyright 2010 by PCBGCODE.COM

## HELP FILE

**Please Read Before Using!**

**version 100, December 2010**

### Terms of Use

PCB-GCODE-WIZARD is software provided "AS IS". Its author makes no warranty either stated or implied, nor any guarantee of suitability, for any particular application or purpose. The author takes no responsibility for any damage, loss, or claims resulting out of usage of PCB-GCODE-WIZARD, or from any files generated by it, whether direct or indirect, or used for actual CNC operations or not. CNC operations can be very dangerous without proper knowledge, experience, procedures, and attention to safety, and milling with small tools at high RPM is no different. *All gcode text files destined for the CNC machine should always be inspected directly beforehand with a text editor for proper syntax, intro and exit portions, etc.*

PCB-GCODE-WIZARD is not freeware or shareware. Your support will allow future improvements and features. **By using PCB-GCODE-WIZARD you agree to accept these terms and to not make or aid in distribution of unauthorized copies of this software.**

### Application Information

PCB-GCODE-WIZARD is provided for Microsoft Windows only. It has been developed on WinXP, Vista, and Window 7, but should also run OK on Win 2000 and even earlier, because it is written in Microsoft Visual Basic 6, the best programming tool ever released by them.

**You will need to have these additional VB files** (available at [pcbrcode.com](http://pcbrcode.com)) copied to the C:/Windows/System32 directory of your PC:

**richtx32.ocx** (rich textbox control)

**comctl32.ocx** (common dialog control)

Then install the VB6 runtime version by executing the file: **vbrun60sp5.exe**

The free DEMO mode distribution of PCB-Gcode-Wizard has all features enabled, but only allows up to 500 lines in the input Gcode file. This will allow you to verify it will run OK on your machine.

### How / where do I get PCB-GCODE-WIZARD?

PCB-GCODE-WIZARD and supporting info is provided via the website: [www.pcbrcode.com](http://www.pcbrcode.com)

This website is dedicated to promoting and sharing of techniques and projects suitable for PCB milling. Free project contributions are welcome.

To request a software purchase, send an email to **pcbrcodeinfo@yahoo.com**. After payment through PayPal is received, the PCB-GCODE-WIZARD file(s) will be emailed to you.

Comments, improvements, or suggestions may be emailed to: **pcbrcodeinfo@yahoo.com**

Technical support is free but limited to the author's time available.

Additional forum discussion at **pcbrcode.org** may be of interest.

Version updates will be announced at **pcbrcode.com**

### **What is PCB-GCODE-WIZARD?**

PCB-GCODE-WIZARD is a stand-alone viewer, trace outline "cleaner", rapid-movement and drill file optimizer for Gcode files generated from a printed circuit board (PCB) layout designed with Cadsoft's EAGLE software. These files allow a person to create PCBs, without using chemicals, by controlling a CNC milling machine, using very small endmills and drills (typically 8 to 20 mils diameter) at high RPM (>10k). "Gcode" is the term used for standardized text files that contain machine control commands and the XYZ coordinates for movements, often used with the .nc file extension (i.e. Numeric Control).

The Gcode files you will use should be generated by, or retain the formatting of, the popular and fantastically useful **PCB-GCODE.ULP** tool ("ULP" = User Language Program") available at **www.cadsoftusa.com** or at **www.pcbrcode.org**. Or you can write your own Eagle ULP to generate it. It is NOT used for viewing or manipulation of 3D Gcode files as might be generated with 3D solid-modelling CAD software.

The latest version of Eagle is v5.10 and for pcb-gcode.ulp it is v3.5.2.11. Together these allow you to create the CNC milling paths (Gcode) needed to isolate the boards traces and pads, and also generate drill files. Most often, it is used for 2-sided boards and generates 3 files: top outlines, bottom outlines, and a top drill file. It is run from within Eagle, so you must have Eagle to use it. Eagle v5.6 or later is recommended. PCB-GCODE-WIZARD is especially targeted for the low-cost home-brew set of folks who need the tools to create PCBs quickly and easily, especially prototypes.

PCB-GCODE.ULP is easy to use. After the board layout is designed, you simply run the ULP and set up the parameters you want for your isolation: tool diameter, isolation width, and step increment. A Gcode text file is generated that you can then pull into PCB-GCODE-WIZARD for viewing, inspection, cleanup, and optimization.

### **What does the PCB-GCODE file look like?**

The Gcode file should be in the output format used by *pcb-gcode.ulp*. However, you may still need to edit the output files depending on the options selected during its setup.

Requirements for the input Gcode file are as follows (it may seem restrictive, but is not that difficult if you are starting with the *pcb-gcode.ulp* output files).

### INPUT FILE FORMAT REQUIREMENTS

- 1) Must be a pure text file format, with CR/LF to separate lines.
- 2) Line numbers may be used, or not, and are ignored.
- 3) There can be no blank lines (check at the file end also).
- 4) All moves must use either G00 or G01 commands, including drills.
- 5) Coordinate format must be no leading zeros, any resolution, with decimal point intact.
- 6) A "space" delimiter must be used to separate the Gcode on each line.
- 7) Z up and Z down must be on separate lines from XY coords.
- 8) Incremental mode (G91) and drill cycles (G8x) are not supported.
- 9) Inch (G20) or millimeters (G21) may be specified, or set manually.
- 10) All outlines must begin with a G00 rapid to the outline start position.
- 11) Then a Z negative (down), then XY points, finally a Z positive (up).
- 12) All comments must be in the beginning of the file, before the first Zup.
- 13) Commands other than G00 or G01 may only be used before the first Zup and after the last Zup.
- 14) Feedrates, M commands, and Z amounts are ignored. However, for G00 rapids Z must be positive, and negative for milling.
- 15) Maximum of 1,000,000 lines in the original gcode file.
- 16) Tools are ignored; instead the user inputs a diameter when running the program.

Here is an actual Gcode file as output by PCB-GCODE-WIZARD showing the format:

```
N0010 ( * * * * PCB-Gcode-Wizard v100 * * * * )
N0020 ( Copyright 2010 by PCBGCODE.COM )
N0030 ( Date: December 20, 2010 Time: 11:53 AM ) <--- comment line
N0040 ( )
N0050 ( This Output File is: )
N0060 ( C:\PCB Projects\PCB-Gcode-Wizard\helpdemo.txt.wiz )
N0070 ( )
N0080 ( The Gcode Input File was: )
N0090 ( C:\PCB Projectss\PCB-Gcode-Wizard\helpdemo.txt )
N0100 ( )
N0110 ( PLEASE INSPECT OUTPUT FILE CONTENTS BEFORE USING !!! )
N0120 ( Note: The original Gcode file comments have been removed. )
N0130 ( -----)
N0140 ( )
N0150 G20 <---- specify Inch or MM (optional)
N0160 G90 <---- absolute coordinates only
N0170 G00 X0.0000 Y0.0000
N0180 M03
N0190 G04 P0.000000
N0200 G00 Z0.0800 <---- anything before this 1st Zup is preserved
N0210 G00 X0.0000 Y0.0000
N0220 G01 Z-.0080 F5.00 <---- 1st Zdown, milling starts here
N0230 G01 X0.0000 Y3.5000 F4.00
N0240 G01 X4.5000 Y3.5000
N0250 G01 X4.5000 Y0.0000
N0260 G01 X0.0000 Y0.0000
N0270 G00 Z0.0800 <---- Z is positive for rapids
```

```

N0280 G00 X0.5603 Y2.3053
N0290 G01 Z-0.0080 F5.00  <---- Z is negative for milling part
N0300 G01 X0.5553 Y2.3103 F4.00
N0310 G01 X0.4847 Y2.3103
N0320 G01 X0.4797 Y2.3053
N0330 G01 X0.4797 Y2.2347
N0340 G01 X0.4847 Y2.2297
N0350 G01 X0.5257 Y2.2297
N0360 G01 X0.7692 Y1.9861
N0370 G01 X0.7761 Y1.9792
N0380 G01 X0.7851 Y1.9755  <---- no comments in main body
N0390 G01 X1.8597 Y1.9755  <---- and only G00 and G01 used
N0400 G01 X1.8597 Y1.9647  <---- no blank lines anywhere
N0410 G01 X1.8647 Y1.9597
N0420 G01 X1.9353 Y1.9597
N0430 G01 X1.9403 Y1.9647
N0440 G01 X1.9403 Y2.0353
N0450 G01 X1.9353 Y2.0403
N0460 G01 X1.8647 Y2.0403
N0470 G01 X1.8597 Y2.0353
N0480 G01 X1.8597 Y2.0245
N0490 G01 X0.8001 Y2.0245
N0500 G01 X0.5603 Y2.2643
N0510 G01 X0.5603 Y2.3053
N0520 G00 Z0.0800
N0530 G00 X0.5321 Y2.8067  <-----|
N0540 G01 Z-0.0080 F5.00          |
N0550 G01 X0.5067 Y2.8321 F4.00  |
N0560 G01 X0.5067 Y2.8679          |
N0570 G01 X0.5321 Y2.8933          |
N0580 G01 X0.5679 Y2.8933          |
N0590 G01 X0.5933 Y2.8679          |
N0600 G01 X0.5933 Y2.8321          |
N0610 G01 X0.5679 Y2.8067          |
N0620 G01 X0.5321 Y2.8067          |
N0630 G00 Z0.0800          <-----|
N0640 G00 X2.4403 Y2.9047
N0650 G01 Z-0.0080 F5.00
N0660 G01 X2.4403 Y2.9753 F4.00
N0670 G01 X2.4353 Y2.9803
N0680 G01 X2.3647 Y2.9803
N0690 G01 X2.3597 Y2.9753
N0700 G01 X2.3597 Y2.9287
N0710 G01 X2.2115 Y2.7805
N0720 G01 X2.1995 Y2.7685
N0730 G01 X2.1995 Y1.8965
N0740 G01 X2.1797 Y1.8767
N0750 G01 X2.1797 Y1.8433
N0760 G01 X2.2033 Y1.8197
N0770 G01 X2.2367 Y1.8197
N0780 G01 X2.2603 Y1.8433
N0790 G01 X2.2603 Y1.8767
N0800 G01 X2.2405 Y1.8965
N0810 G01 X2.2405 Y2.7515
N0820 G01 X2.3887 Y2.8997
N0830 G01 X2.4353 Y2.8997
N0840 G01 X2.4403 Y2.9047

```

----- this is one outline  
 block

```

N0850 G00 Z0.0800
N0860 G00 X3.4403 Y1.6633
N0870 G01 Z-0.0080 F5.00
N0880 G01 X3.4403 Y1.6967 F4.00
N0890 G01 X3.4167 Y1.7203
N0900 G01 X3.3833 Y1.7203
N0910 G01 X3.3597 Y1.6967
N0920 G01 X3.3597 Y1.6633
N0930 G01 X3.3833 Y1.6397
N0940 G01 X3.4167 Y1.6397
N0950 G01 X3.4403 Y1.6633
N0960 G00 Z0.0800          <----- anything after this Zup is preserved
N0970 G00 Z0.2000
N0980 M05
N0990 M02

```

### Menu Structure of PCB-GCODE-WIZARD

Menu structure is very simple. Only command mouse-click "buttons" are used. One file at a time is operated on. Restart the program to use a different Gcode file for input.

Generally, one steps through the process in the following order:

- a) Get the Gcode file and observe its statistics.
- b) Plot the trace outlines, and inspect for content and errors.
- c) Remove un-needed outlines to reduce milling time, using either Select mode or Clean mode.
- d) Optimize the G00 rapids for shortest path.
- e) Save the new data to a ".wiz" output Gcode text file.
- f) Optimize a drill file using (almost) similar steps (see below).

Here is a list of the mouse-click command buttons that form the user menu, in order from left to right on the screen.

<b>GET FILE</b>	Browse to select the input Gcode file. Then open it, read it, and show its statistics, and finally plot all the outlines. Any file syntax errors will end the program so they can be fixed.
<b>PLOT ALL</b>	Plot the outlines. Will show milled width if Tool Diameter has been set.
<b>SELECT</b>	Lets user Select an outline by mouse-click and immediately deletes it. (press Esc to cancel Select mode, U to undo the last deletion only)
<b>CLEAN</b>	Steps through all of the outlines and allows the user to Keep or Delete each.
<b>OPTIMIZE</b>	Plots all of the G00 rapid movements, then finds the SHORTEST path.
<b>ZOOM</b>	Lets the user <b>DRAG</b> a zoom area rectangle on the plot.
<b>MEASURE</b>	Lets the user <b>DRAG</b> between two points on the plot to find distance. The delta X, delta Y, and Distance are shown.
<b>TOOL DIAM</b>	Set the milling tool diameter in inches or mm. Toggle display with "T" key. The XY coordinates of the cursor, and current Tool Diameter are shown.
<b>SAVE</b>	Saves the current state of the file, defaulting to the same file name and directory as the original, but with extension ".wiz" attached.
<b>QUIT</b>	Ends the program immediately, without saving anything.

<b>INCH/MM</b>	Usually automatic, but can be set by user if not defined in the input file.
<b>KEEP</b>	Lets the user Keep the highlighted outline path during the CLEAN process.
<b>DELETE</b>	Lets the user Delete the highlighted outline path during the CLEAN process.

### Special Keys Available

When in plotting or in zoom mode, you may also use these keys:

<b>Home</b>	Plot everything using current setting of Tool Diameter.
<b>Page Up</b>	Zoom In (make larger) in steps.
<b>Page Down</b>	Zoom Out (make smaller) in steps.
<b>Arrow Keys</b>	Pan up, down, left, or right in steps.
<b>"P" key</b>	When viewing, pressing P displays the coordinates of the <b>Point</b> nearest to the current cursor position, and marks it with a red circle. Use this feature to get exact vertex coordinates, or to help search for specific points in the Gcode file.
<b>"T" key</b>	Pressing the "T" key toggles display of <b>Tool Diam</b> on or off (i.e. line width)
<b>"H" key</b>	Press the H key anytime to view the <b>Help File</b> .

When in Clean Outlines mode, you may also use these keys:

<b>Escape</b>	Keep the outline.
<b>Delete</b>	Delete the outline.

When in Select Outlines mode, you may also use this key:

<b>Escape</b>	Cancel the Select mode.
---------------	-------------------------

### The CLEAN Function

The Eagle-generated Gcode files usually contain similiar stepped milling paths, or trace OUTLINES, depending on the cutter diameter used, step size, and the amount of isolation desired. For high power or voltage circuits, the extra isolation passes may be required, but for low voltage digital circuits, all that is usually needed is enough milling to electrically isolate the traces and pads (perhaps 10 mils). If you are lucky enough to have a 60,000 RPM air spindle and fast servo driven CNC machine, and don't mind using up a lot of expensive carbide milling bits, then you may not need to CLEAN the file. But for slower low-cost homebuilt machines it may cut an hour or more off board milling time.

The minimum milling time would of course be achieved with just one milling pass around each trace. Now, if you use GROUND PLANE FILL which is very useful for making connections (or any other plane fill type, i.e. power), it turns out that EAGLE generates an outline around those objects also, so at the minimum you get TWO sets of outlines. But they are redundant, because you already have an outline to cut around every trace, so they can be removed. This will cut milling time almost in half, as well as tool wear (keeping in mind that even carbide bits only have hundreds of inches of cutting life, since copper and fiberglass is very abrasive).

CLEANING a gcode file is the process of stepping through all of the milling paths, or outlines, and deciding to either *Keep* or *Delete* them. The process used by PCB-GCODE-WIZARD is entirely graphical and user driven. This was done, quite simply put, because the computational algorithms required to determine which outlines are not needed is very intense. However, it is very easy to see graphically, plus it allows the user to modify the milling as desired.

The tradeoff, of course, is the time it takes to look at every outline and decide. Each outline is shown, one at a time in red (with the rest of the outlines in blue so you can see what is nearby), and zoomed in to view that area of the board. Then you just click KEEP (or press Esc key) or DELETE (or press Delete key). You should CLEAN the gcode, if desired, before OPTIMIZING its G00 rapids. Note that the plot scale is constantly changed depending on the outline size. Very small outlines in particular may look much larger than actual size. After running a few files, it is easy to decide which outlines are not needed.

To shorten the cleaning effort, set up the isolation parameters within the *pcb-gcode.ulp* menu to achieve the minimum number of milling passes (i.e. steps) that is needed. To generate only one pass, set the maximum isolation value to less than the tool size, so only one outline is generated (again, if you have copper fill areas, you will appear to have two sets of outlines, one around the trace, and one around the fill).

Note that the Clean function skips over any outlines that have no XY movement, for example spot drills which only have a G00 followed by Zup and Zdown. So all of these Gcode segments will always show up in the output file, unless they are deleted using the Select mode.

### **The SELECT Function**

SELECT mode works just like the CLEAN process (see above) except it allows you to individually select one outline at a time by mouse-clicking on it. For larger files with just a few outlines to remove, it is more convenient than having to step through all of the outlines. In Select mode all outlines are drawn with zero-width lines.

You'll need to zoom in *before* you select, for accurate outline selection, and *you must click near a point in the outline for best results*. To reverse a single deletion, you can press the "U" key (for Undo) immediately after the deletion, but it must be done before another click. When you are done selecting, press the Escape key to cancel Select mode. Note that panning and zooming are disabled in select mode, so that you can see the outline come back if you Undo it. Press the Esc key to exit Select mode and re-enable pan & zoom and "T" key functionality.

You can also delete spot drill locations this way, which is handy since they are skipped over using the Clean cycle. Select and Clean modes always draw outlines with zero-width lines for better visibility. Any outlines that are deleted will not be included in optimizing later, or saved to the output file.

### **The OPTIMIZE Function**



Eagle generates the outlines in more-or-less random order, resulting in totally random G00 rapid movements as the CNC goes from one to the next. The OPTIMIZE command looks at the possible paths through the outlines and determines the shortest one. This is often MUCH less travel than it would otherwise be (typically less than 20%). You should always CLEAN the file before OPTIMIZING it.

The OPTIMIZE feature first draws a full plot showing the original G00 rapid movements, which will look very un-organized. After doing the optimization calculations, it reports the original and optimized total G00 path length, then replots showing the new path, which will appear to be a squiggly line. The 1st move from the origin is drawn in green to help see the start of the path.

The OPTIMIZE function may also be used for outline files with "spot drill" sequences included (i.e. using the same milling bit as was used for milling trace outlines), as allowed by pcb-gcode.ulp. These are simply Zup/Zdown movements with no XY travel. These spot drill locations will show up in plots as red circles (if Tool Diam = 0) or filled dots (if Tool Diam >0). Spot drill locations ARE included in the optimizing function. However, you MUST set up the OPTIONS in pcbgcode.ulp to specify "Use Simple Drill Code" so that each drill or spot uses normal G00/G01 commands instead of G82 (not supported yet).

## Drill Files

Unfortunately, in this version, a standalone drill file cannot be read directly, but may be partially *optimized* with some manual file editing. A drill file typically uses several sized drills, or Tools. To optimize, the user must manually split up the drilling into *one Gcode file per drill size*, and then merge them back together afterwards. All you need to have in each file is to copy the drill movements (three G00/G01 Gcode lines per hole), then use PCB-GCODE-WIZARD to optimize and save the output, then paste it back into the original file where that Tool portion was initially. The majority of drilling is usually done with only one or two drill sizes that have enough holes to really benefit from optimization, so it really may not be that much work.

The temporary drill file will only contain lines like shown below (3 lines per hole). You MUST set the OPTIONS in pcbgcode.ulp for "Use Simple Drill Code" to get this format.

```
G00 Z0.0100      <----- start with initial Zup
G00 X0.1280  Y0.8320  <-----|
G01 Z-0.0850 F10      |----- 3 lines per hole
G00 Z0.0100      <-----|
G00 X2.4780  Y1.2600
G01 Z-0.0850 F10
G00 Z0.0100      <----- end with Zup
```

If you want to drill all holes with just one size drill (say in case you don't have an automatic Tool changer), another way to optimize is then to setup pcb-gcode.ulp to output the "spot drills" and set the drill depth to go *through* the board. It will generate all of the drill movements in one section of the file (without any tool size and tool change commands), that you can cut and paste into a separate file, and then optimize. After the board is done milling/drilling, you can go back



and manually drill out any other sized holes, assuming you only have a few to do. You might want to use a different pad shape for different hole sizes to make identification easier.

### **Misc Notes and Tips**

**1) THE BEST RESULTS START WITH THE RIGHT INPUT FILE FORMAT! So take the time to look at it in a file editor (Notepad etc). Standard output from pcbgcode.ulp should work without modification.**

The main reason for the format requirements is that the cleaning and optimization process might modify the body of the file in ways that would affect the meaning of comments and commands (other than G00/G01). So .... keep it simple.

- 2) When plotting, the X and Y axis are drawn with a red line, and the (0,0) origin marked with a small red circle. Screen orientation is always positive X up, positive Y to the right.
- 3) You must press the Zoom or Measure button EACH TIME before attempting to do a rectangle zoom, or measure. But the pan/zoom/home/page up/page down keys can be used often.
- 4) You can use the Tool Diam button setting to see how the finished board would look with different diameter milling tools, perhaps in case you don't have the exactly right size.
- 5) When doing a CLEAN, the entire file is stepped through. If you click on QUIT before finishing, you will have to start over from the beginning. You cannot go backwards through the outlines (and *undo* is not yet supported) so be sure when you DELETE. If you aren't sure, click KEEP, and view the file again first. (remember also, Esc key = Keep, Delete key = Delete)
- 6) The original input gcode file is left intact, with no changes. You can use any file extension for either input or output file, as long as its contents use plain text format with CR/LF.
- 7) When SAVING the output file you are given the opportunity to keep or discard the original comment lines, and whether or not to add the "N" line numbers, re-numbered in order, in increments of 10. Any original N numbers are always removed. Having line numbers is very helpful in locating file format errors. *If you have a file that doesn't have line numbers, simply load it, then Save it, and click Yes when asked if you want to add line numbers.*
- 8) The output file will contain all lines in the input file up to "the first Zup just before the first Zdown", and from "the first Zup after the last Zdown" to the end. Except for comments if you choose not to keep them when saving.
- 9) PCB-GCODE-WIZARD is not meant to read 3D Gcode files, or all sorts of other Gcode machine formats. Only the file format as stated above is valid.

### **Version History**

v100	Dec 2010	Initial release. View, Clean, and Optimize rapids and single Drills.
------	----------	--