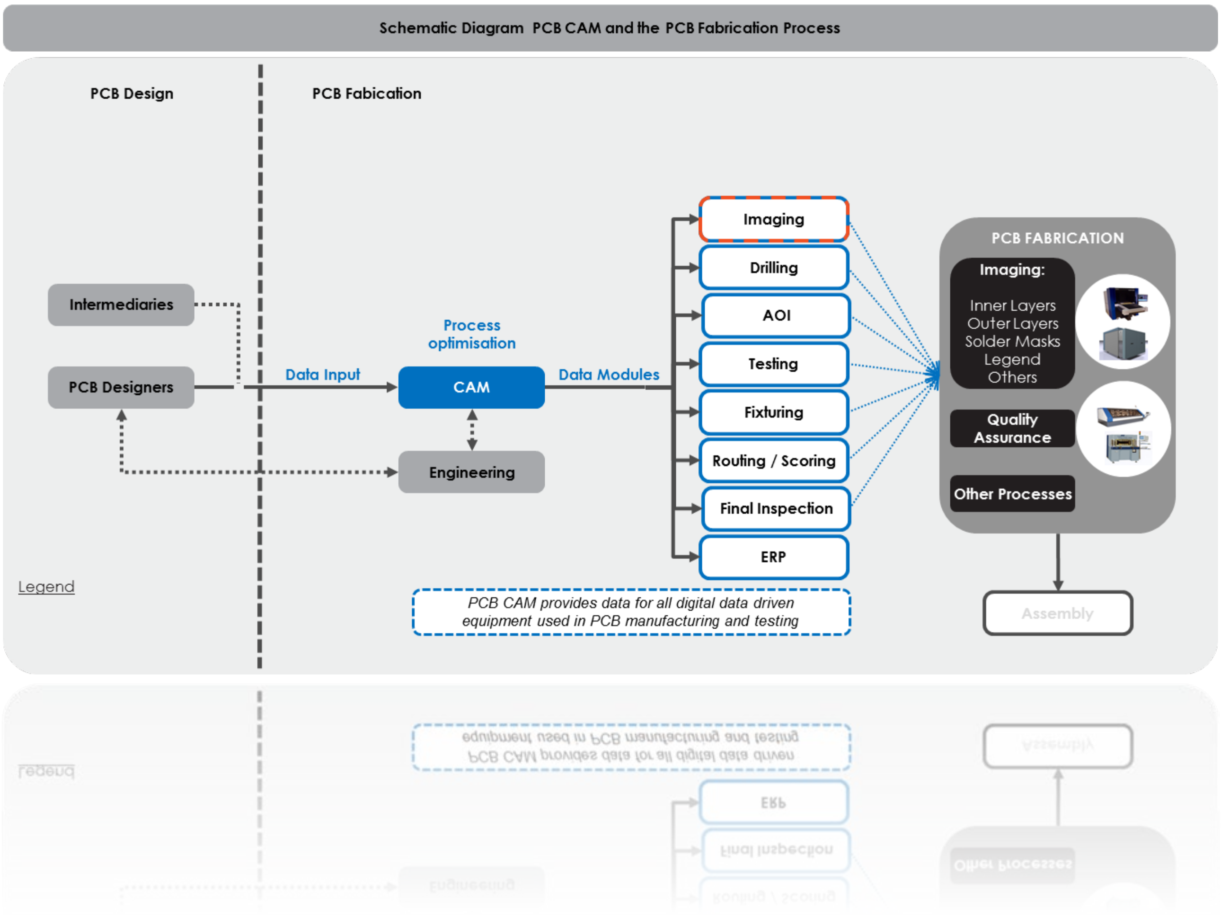
PCB Fabrication Data

Design to Fabrication Data Transfer

Karel Tavernier ©
V6
September 2015



1. Table of Contents

2. Preface

This monograph discusses the construction of PCB fabrication data sets, generated by CAD and sent to professional PCB fabricators. Where appropriate it offers recommendations. Its topic is not cover amateur production directly from CAD.

Target audience is CAD PCB layout users, developers and product managers.

It is clearly possible to fabricate PCBs from the fabrication data sets currently being used – it's being done innumerable times every day all over the globe. But is it being done in an efficient, reliable, automated and standardized manner? At this moment in time, the honest answer is no, there is plenty of room for improvement in the way in which PCB fabrication data is currently transferred from design to fabrication.

This is not about the formats used, which for over 90% of the world's PCB fabrication, is Gerber: there are very rarely problems with Gerber files themselves – they allow images to be transferred without a hitch. In fact the Gerber format is part of the solution, given that it is the most reliable option in this field. The problems actually lie in how the format is used and, more often, in how it is not used.

The monograph first explains what happens to the designer's data once it reaches the fabricator. This is important because for designers to construct truly valid PCB data sets, they must have a clear understanding of how their data is used. This, more than anything else, clarifies how it should be prepared.

This monograph does not deal with how to design PCBs for easy fabrication, which is completely outside the remit of the mere developer of the Gerber format and a matter for the PCB fabricators themselves.

Comments or questions can be sent to gerber@ucamco.com

3. Synopsis

All pads must be flashes. All flashes must be pads., 29

Always include an IPC-D-350A netlist file in the PCB fabrication data set., 28

Always use positive copper layers., 38

Always use the standard file extension “.gbr” for Gerber files., 40

CAD output is CAM input. It is *not* machine input., 9

Check the error messages of your Gerber viewer., 42

Do not use application formats such as Word or Excel format., 33

Do *not* use Standard Gerber., 43

Drawings are no substitute for digital data., 31

Express the layer structure in a standardized way by using Gerber X2. If the standard is not available use clear file names., 21

Include a separate file with a contour defining the true profile., 16

Make the fabrication data set as simple as possible, but no simpler., 34

Output all layers in the same coordinates, 12

Output all layers in the same precision., 39

Output *all* your drawings in Gerber., 32

Output your drill and rout files in Gerber., 19

Put all your general data in human readable text files, informal or structured as in CSV, XML and YAML., 33

Put only data in data files., 26

Put plated and non-plated holes in separate files., 24

Use the maximum precision for coordinates., 39

When creating fabrication data from CAD, do not ask yourself ”What can I do to create better production tools?“ but ask yourself:“What can I do to create better CAM input?”, 9

All pads must be flashes. All flashes must be pads., 29

Always include an IPC-D-350A netlist file in the PCB fabrication data set., 28

Always use positive copper layers., 38

Always use the standard file extension “.gbr” for Gerber files., 40

CAD output is CAM input. It is *not* machine input., 9

Check the error messages of your Gerber viewer., 42

Do not use application formats such as Word or Excel format., 33

Do *not* use Standard Gerber., 43

Drawings are no substitute for digital data., 31

Express the layer structure in a standardized way by using Gerber X2. If the standard is not available use clear file names., 21

Include a separate file with a contour defining the true profile., 16

Make the fabrication data set as simple as possible, but no simpler., 34

Output all layers in the same coordinates, 12

Output all layers in the same precision., 39

Output *all* your drawings in Gerber., 32

Output your drill and rout files in Gerber., 19

4. What does a PCB fabricator do with CAD fabrication data?

PCBs are typically fabricated in about 22 steps, many of which are digitally controlled and require dedicated data modules called production tools.

Some designers believe that their PCB fabrication data will drive the fabricator's production machines directly; that the Gerber files will be used directly on the PCB fabricator's photoplotter; that Excellon drill files will go straight onto the fabricator's drilling machines; and that IPC-D-356A netlist will go right into electrical test machines.

Not so. Fabricators never use the Gerber or Excellon files directly on their equipment. Never.

There are many reasons for this, the simplest of which is panelization. Even though the designer's data describes a single PCB or an array, the job is never manufactured as such. It is always put on a production panel, which will typically have multiple jobs on it, as well as a border for plating, test coupons, etc. This is illustrated below.

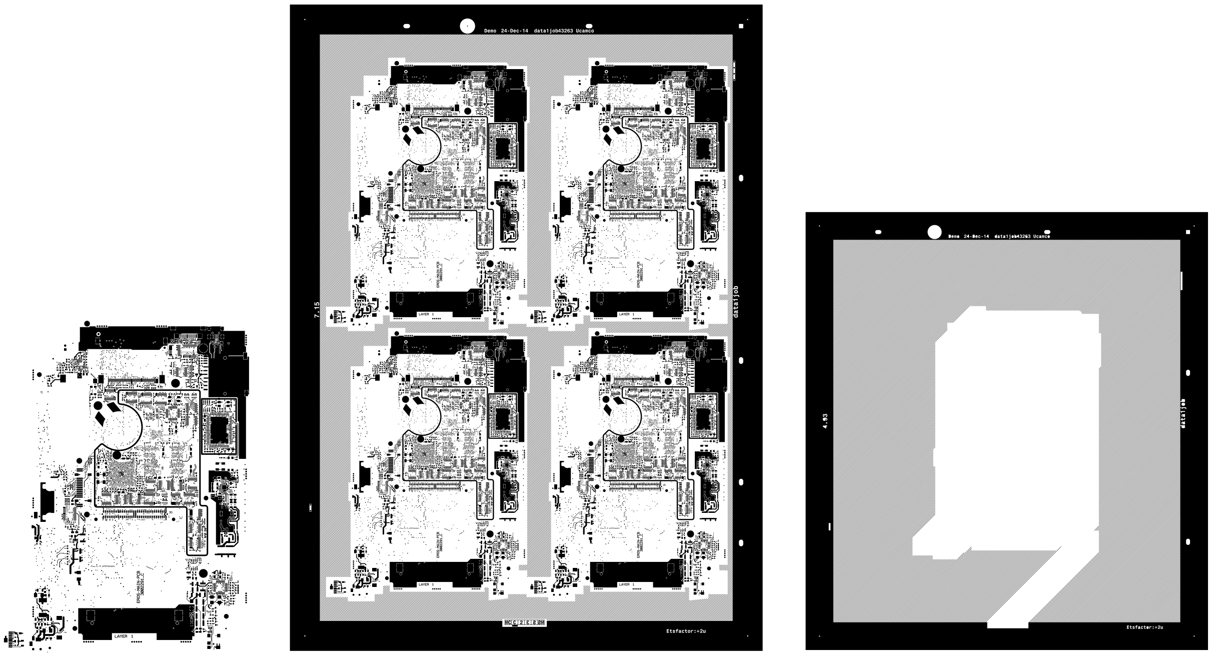


Fig.1 Single PCB, Panel with 4 PCBs, Panel with one prototype PCB

It follows that the fabricator can do nothing with production tools for a single job: he needs films and drill files for the production of his panel. Another reason is that the fabrication processes inevitably introduce deviations, such as layer distortion during lamination and line width reduction during etching. These deviations must be compensated for prior to manufacture. A third reason is that the production tools driving the fabricator's equipment must fit the fabricator's specific requirements and must often be in a proprietary format associated with the machine.

For all these reasons, the production tools that will drive the fabricator's equipment are generated by the fabricator's CAM department. PCBs cannot be professionally fabricated without this step: no CAM, no fabrication – it's as simple as that.

The PCB CAM system typically performs the following steps (see Fig. 4):

- Input digital data (Gerber, Excellon, etc)
- Reconstruct a physical model of the PCB. This may require the manual addition of information that has been 'lost in translation’
- Check for errors and if necessary communicate with the designer
- Send product information to ERP to support quoting
- Send product information to the engineering department who will decide how and on which equipment the PCB will be made, produce the travelers etc.
- Optionally tweak the design for DfM and if necessary communicate with the designer.
  At this stage CAM has a model of the single PCB as it will be delivered
- Create the customer panel (aka array or shipment panel)
- Create the production panel (aka working panel)
- Compensate for deviations in the manufacturing processes (e.g. scaling to compensate for distortions during lamination)
  At this stage CAM has a model of the production panel as it will be manufactured
- Send fabrication data to ERP
- Output dedicated digital production tools to drive the NC fabrication equipment (photo plotting, direct imaging, legend printing, drilling, routing, scoring, AOI, electrical test files, AVI). The content of these production tools is very specific to the fabricator's and customer's setups, as the information needed and format used are often proprietary and specific to the equipment brand and model, while the equipment must “know” customer-specific details such as the location of the fiducials for registration on the direct imager and “don't care” zones for the AOI system.

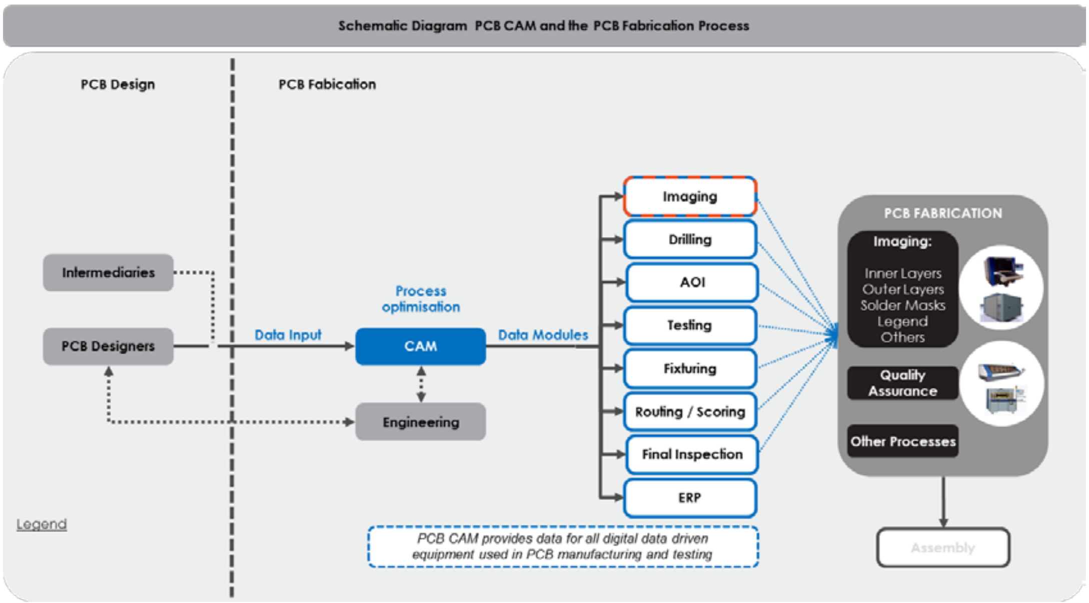


Fig. 2 Schematic of the PCB CAM and fabrication process

To generate the production tools the CAM operator needs a perfect physical model of the bare PCB. As we have said, he client's incoming fabrication data is used as digital data to reconstruct that physical model.

It may at first sight seem strange that machine files such Gerber, Excellon and IPC-D-56A are used for this, but it actually makes perfect sense. The Gerber format has evolved way beyond its origins as a photoplotter driver to become the perfect vehicle with which to transfer digital image and drill information from CAD to CAM. And it is precisely because of those origins that Gerber files are so perfectly suited to their current role of accurately representing where copper and other materials are. Similarly, Excellon drill files correctly specify where the drill holes are. This makes these indeed formats capable of describing a PCB.

So the incoming Gerber, Excellon and other data is always read into the fabricator's CAM system which analyses, reworks and transforms the image and drill information into production tools. This is a very different proposition from using the designer's datasets directly as production tools in three aspects:

- The data files are not standalone items, but part of an interconnected dataset that, together, describes a PCB.
- PCB CAM needs to ‘know’ more about a PCB than just the image. CAM needs to ‘know’, for example, which pads are edge connectors, because these need to be gold plated; CAM needs to ‘know’ which drill holes are vias, because the solder mask is treated differently around via pads than around component pads; and so on.
- CAM will change (edit) the image files before production, for example the solder mask openings are typically adapted to fit tolerances. The files therefore need to be editable efficiently. Without this restriction a bitmap (TIFF) or contour (PDF, DXF) would be a sensible data format. The problem is that these formats are not editable as a PCB – they are general image formats. A properly constructed Gerber file ís editable.

These three points affect the way in which the designer's PCB fabrication data must be structured. The designer need not worry whether his or her files will be able to drive NC production machines - it is the job of the fabricator’s CAM to manipulate the design data so that it will run on real production equipment. The designer's focus should be on specifying the end product accurately, completely and unequivocally.

CAD output is CAM input. It is *not* machine input.

When creating fabrication data from CAD, do not ask yourself ”What can I do to create better production tools?“ but ask yourself:“What can I do to create better CAM input?”

5. Coordinates: alignment, registration, mirroring

Never mirror or flip layers! All layers must be viewed from the top of the PCB. (This means that the text must be readable on the top layer and mirrored on the bottom layer). Alas, in a mistaken attempt to be helpful, some designers flip even numbered layers because they must indeed be mirrored on the photoplotter. This could be helpful in a world where the designer's files are used directly in fabrication, but this is never the case. The files are actually input for the fabricator's CAM system to reconstruct a model of the PCB, with all layers correctly aligned, exactly as in the CAD system. The fabricator's CAM software will optimize and panelise the PCB. On output of the final, panelized fabrication tools, it will mirror, rotate, shift and scale as required by production. If a designer that mirrors layers one can only hope that the CAM engineer notices this and 'unmirrors' them, or scrap is produced. Spotting mirroring is not easy on symmetrical jobs, even less when there is no text.

This is how a job with flipped layers and misaligned drill file looks in CAM.

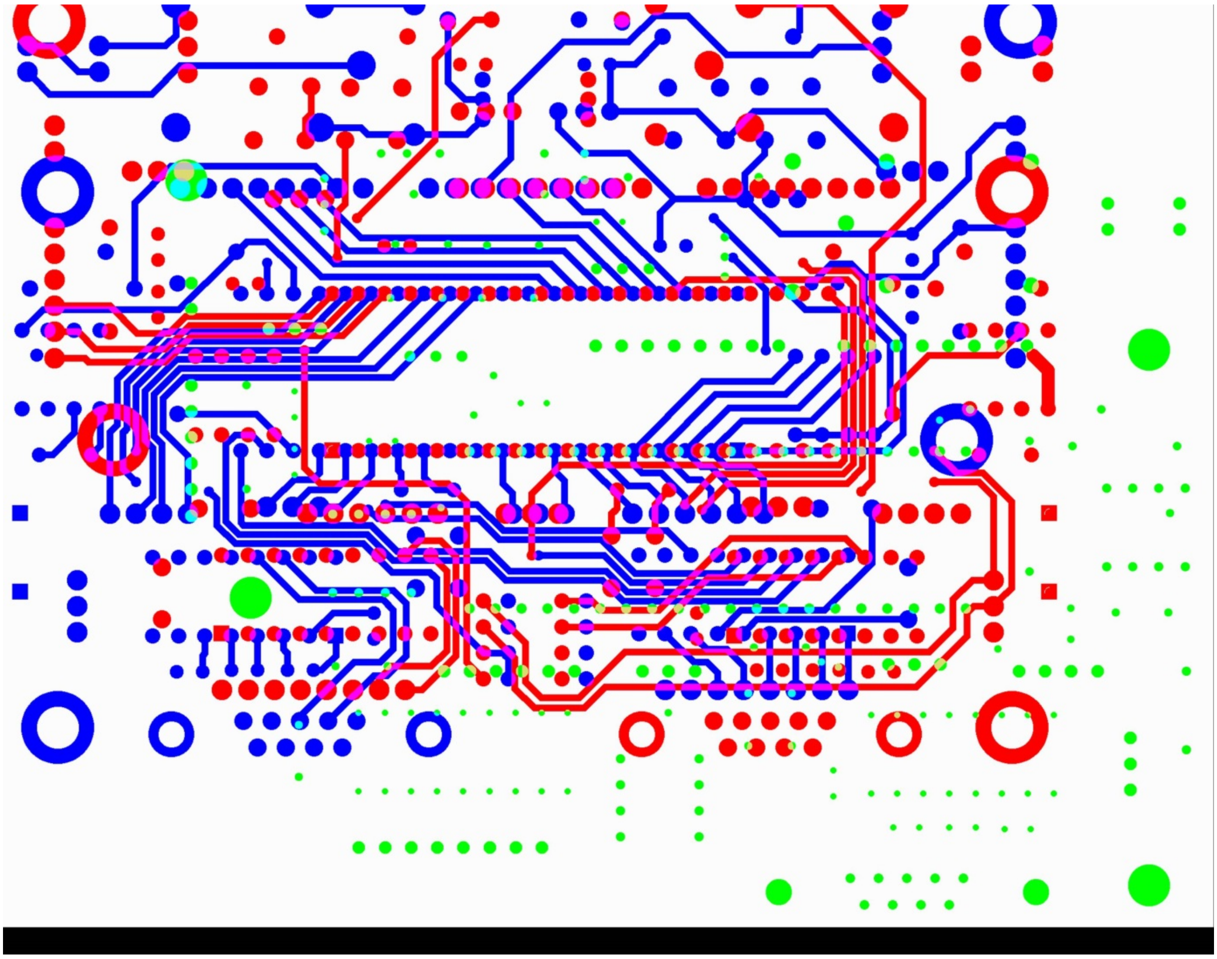


Fig. 3 Job with flipped layers and misaligned drill files

And this is how it looks with properly aligned layers looks in CAM. Which do you prefer?

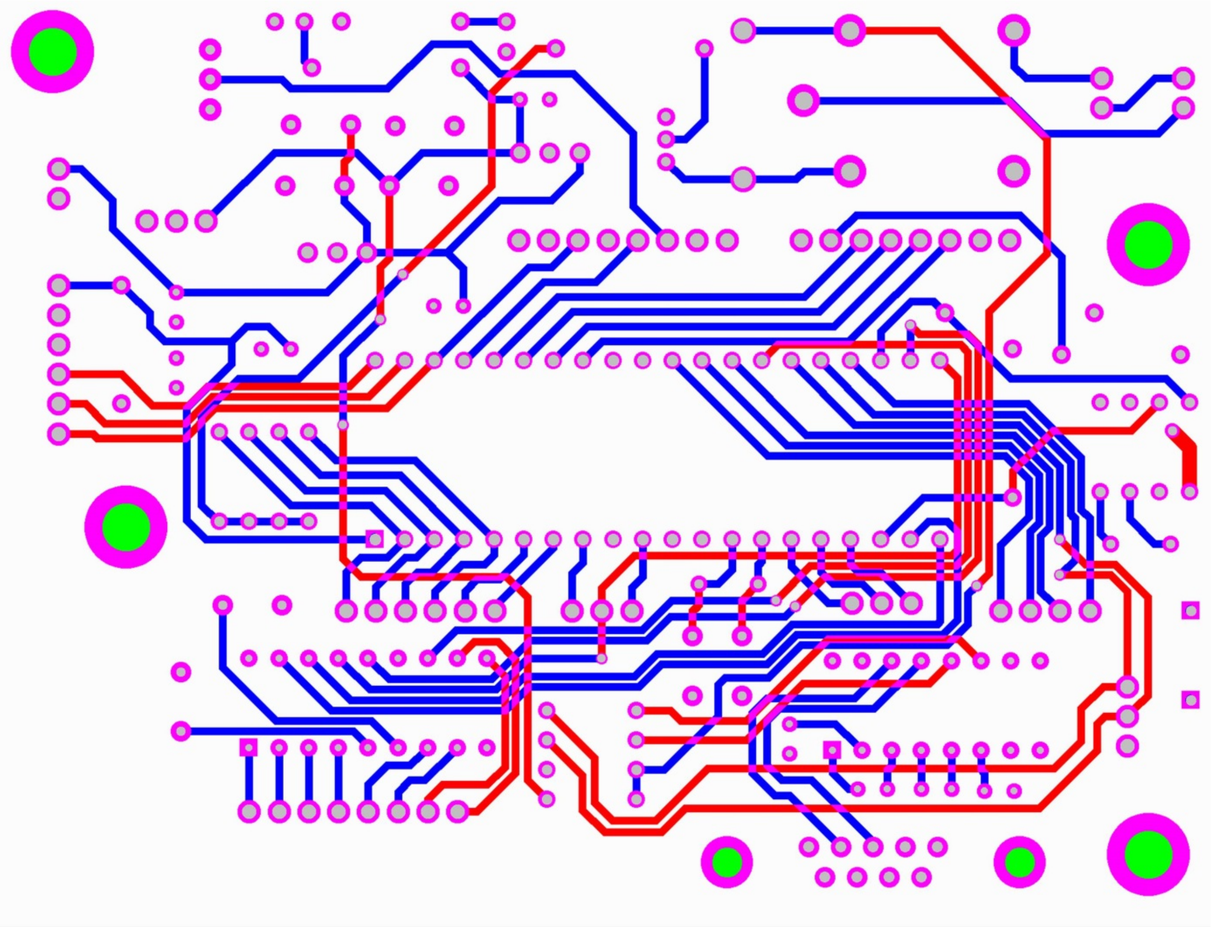


Fig. 4 Job with properly aligned layers and drills

All files representing the physical PCB in the fabrication data – copper, drill, rout, solder mask, legend, outline and peelable - are read into the CAM software to reconstruct a model of the PCB. All these layers must align or stack up correctly – you no doubt want your fabricator to deliver a PCB were copper layers, solder mask, drills do align and are not all over the place. This alignment is very simple to achieve: output all the files in the same coordinate system so that they align. They then align automatically after they are read into the CAM software. Fabrication data is not a set of independent files. There is a relation between these files. This relation must be expressed in the data, by using the same coordinates. PCB fabrication data were layers, drills, routs do not align is simply *wrong*.

Sometimes registration is “solved” by adding alignment targets to the images. This is not the solution. CAM engineers must first manually move the layers until the targets align, and then manually remove the alignment marks. The solution is not to flip in the first place. Targets were needed in the days, now half a century ago, when physical films and not digital data were sent from design to fabrication, and when animals still talked. Eliminate such complications by using the same coordinates, and register your data files.

For the sake of clarity any drawings, though not a physical part of the PCB, should be aligned with the data and use the same coordinates.

Next to intentionally flipping even layers a common cause of wrong alignment is the use of different output formats. (See chapter 14.)

Your favorite Gerber viewer will no doubt allow you to verify that they are aligned.

Output all layers in the same coordinates

6. The PCB profile (or outline)

The profile defines the extent of the PCB. It separates what is the PCB from what is not the PCB and is an essential part of PCB fabrication data. Without knowing the profile the PCB simply cannot be fabricated. The profile must be properly and precisely defined.

The profile defines a simple region in the 2D plane. The proper way to do this is to specify a closed contour: the inside of the contour is the PCB, and the outside is not. Simple.

Note that such a simple region is solid, without holes. By definition then, a profile cannot have holes intentionally placed within it. These are superfluous and represent an unnecessary and complicated duplication given that drill holes are well defined in the drill/rout file. One can view cut-outs in a PCB as still part of the PCB, just as much as the drill holes are.

A contour is defined by the Gerber spec as follows:

A contour is a sequence of connected draw or arc segments. A pair of segments is said to connect only if they are defined consecutively, with the second segment starting where the first one ends. Thus the order in which the segments of a contour are defined is significant. Non-consecutive segments that meet or intersect fortuitously are not considered to connect. A contour is closed: the end point of the last segment must connect to the start point of the first segment.

The Gerber format regions are defined by contours with the G36/G37 commands. This is precise and unequivocal, and is the recommended way to specify the profile. The filled contour exactly covers the PCB.

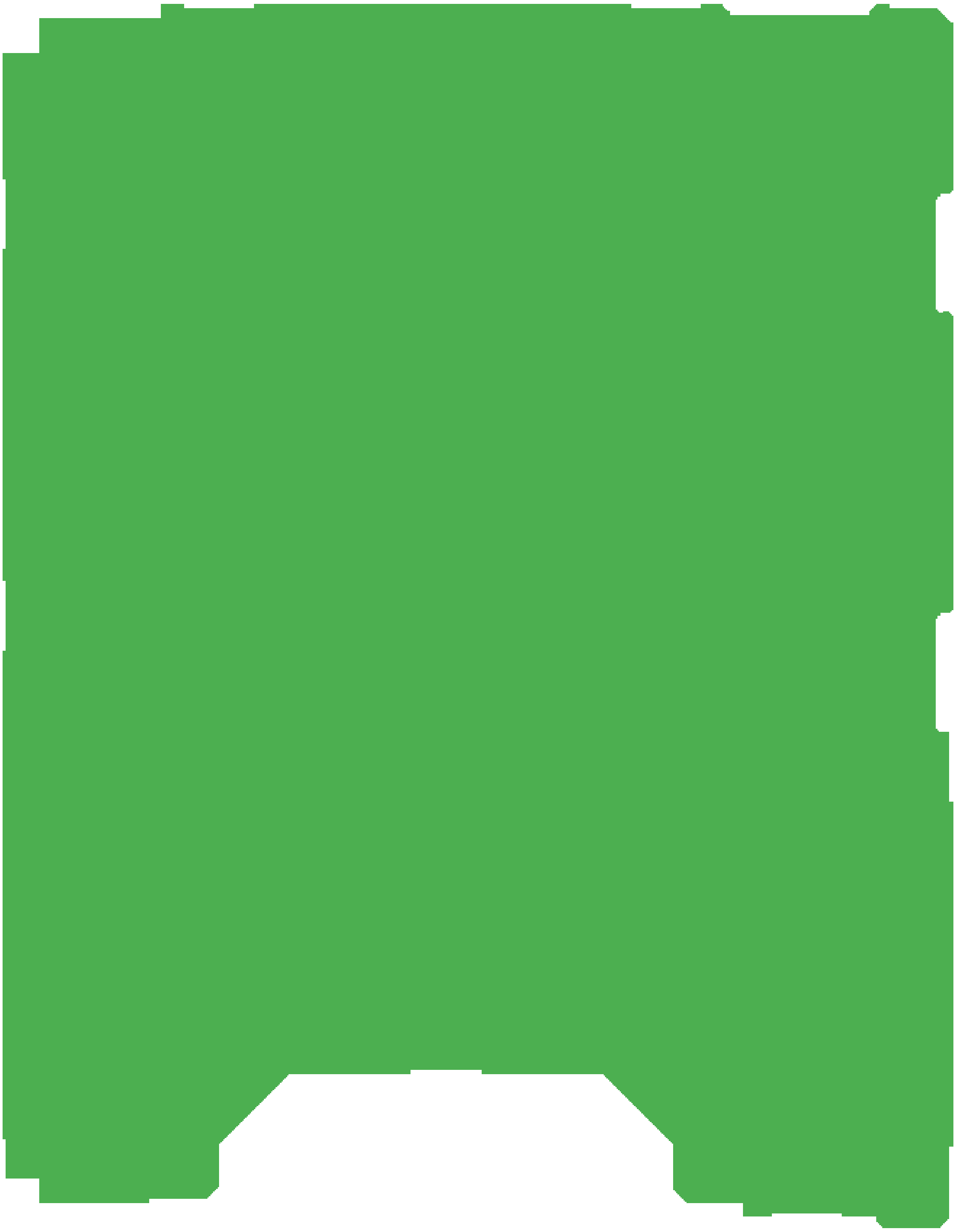


Fig. 5 Profile by G35/G37 region

If this is not possible the profile can be specified by drawing the contour with a zero-size or very small size aperture. If the aperture is not zero size the profile is the center line of the stroked line; in other words do not compensate for aperture size, you are transferring an image, not a production tool for a drill machine.

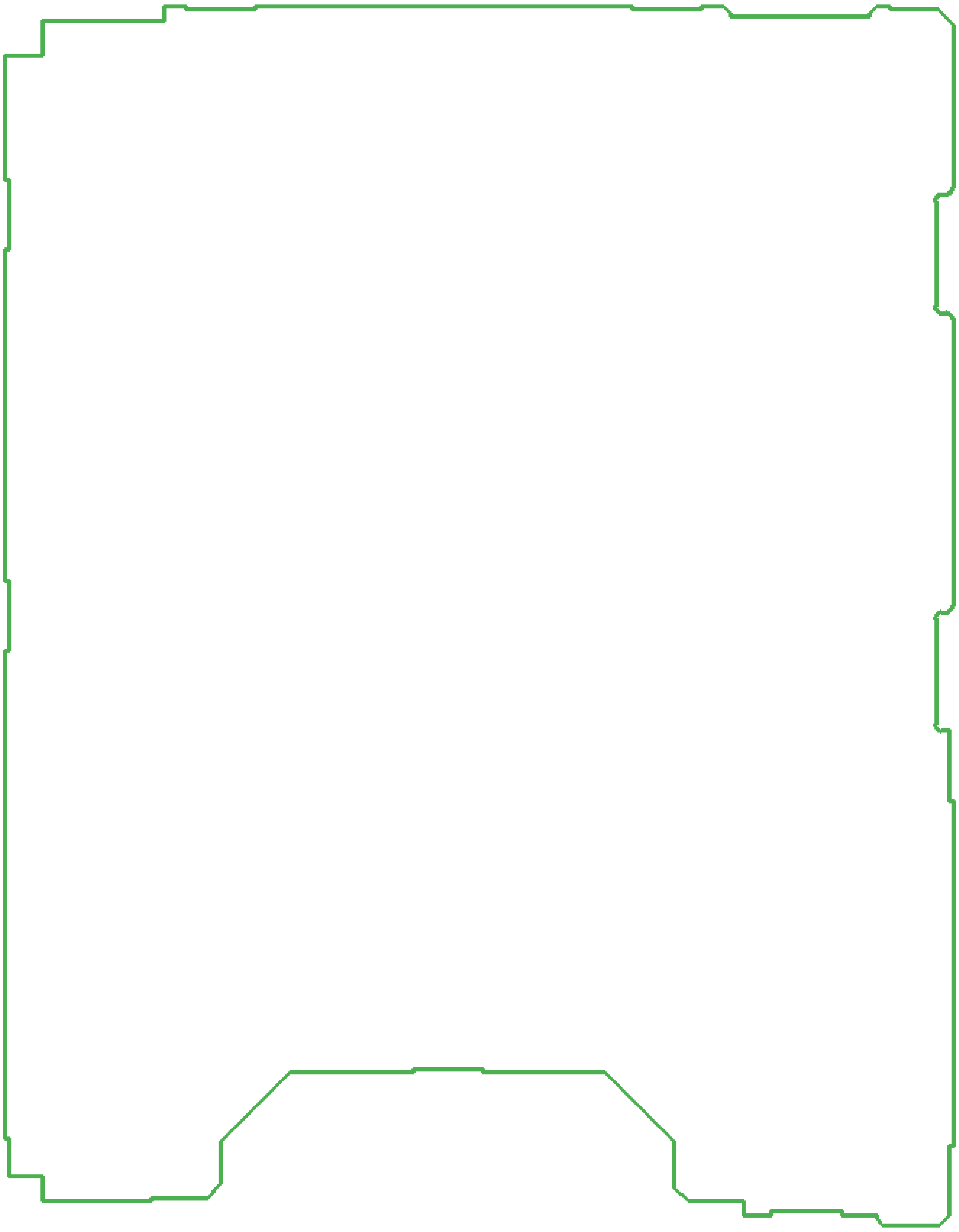


Fig. 6 Profile stroked with a thin aperture

The profile layer is not copper. It is more akin to a drill or rout file as it affects all layers. The outline should therefore be put into a separate file, and not shoved into a copper layer.

It is helpful to provide a mechanical *drawing* with the profile. However, this is not a substitute for *digital data*.

Corner marks are sometimes used to indicate the profile. Again, corner marks are meant for visual interpretation and do not constitute digital data. Therefore they are not a valid specification of the profile.

What is definitely not acceptable is to take a copper layer and add a crude manually-drawn line that, to make matters worse is drawn with the same aperture as the copper tracks.

Include a separate file with a contour defining the true profile.

7. Output drill files in Gerber rather than in an NC format

Incoming drill files in particular are never used directly on the fabricator’s drilling machines. There are plenty of reasons for this, and we've already seen some of these above, but the specific issue with drill files is that incoming CAD files specify the end-diameter of the hole on the finished bare board, while in reality a bigger hole must be drilled to make room for plating. Furthermore, the designer is unlikely to know the origin and orientation of the fabricator's drilling machine, its working parameters and all the rest, which will also affect the way in which the production panels are drilled and routed.

Therefore the CAD designer need not fret about whether the drill machine works in metric or imperial, or can handle the resolution, feeds and speeds: the fabricator's CAM system takes care of all that. The designer needs only to concern him- or herself with how to transfer the design data to CAM, optimally, accurately, completely and safely with a minimum of manual work for all parties.

It may come as a surprise, but the best file format for CAD to CAM drill information transfer is a Gerber file, not an NC or Excellon file. The most important reason for this is *attributes*, which are essential for automated CAD to CAM data transfer. Attributes specify the span and plating of a file, the function of a drill hole – that it is, say, a via hole – and the drill tolerances. Gerber is the only format that supports attributes. NC formats do not.

In addition, CAM often faces the following problems with NC files:

- **Drill files and copper layers are not registered as they use a different datum point**
  This occurs in nearly 50% of all job data sets. Where this happens, the CAM engineer has to work the data, shifting and rotating each drill file to register it with the copper layers. This is a medieval way of working. The only secure solution is for the CAD professional to generate the drill and copper layers in perfect register, preferably using the coordinates of the CAD system on all layers, to facilitate communication between designer and fabricator. The underlying reason for misregistration is that copper layers are output in Gerber and the drill data in an NC format. The NC output in CAD can possibly be set up to register with the Gerber data, but the simplest way to guarantee registration is to output both in Gerber.
- **Drill holes are not centered on the copper pads**
  When this occurs, part of the drill tolerance is gone before drilling even starts, increasing the risk of breakout. The root cause is different rounding and resolution in the Gerber and NC drill files. To overcome this issue, the CAM engineer may “snap” the copper pads to the drill holes. This manual operation takes time and carries risk, forcing the CAM engineer to manipulate the design – which is something the designer does not want. The simplest solution is to output the drill files in Gerber at the same resolution as the copper layers. Then everything matches perfectly.

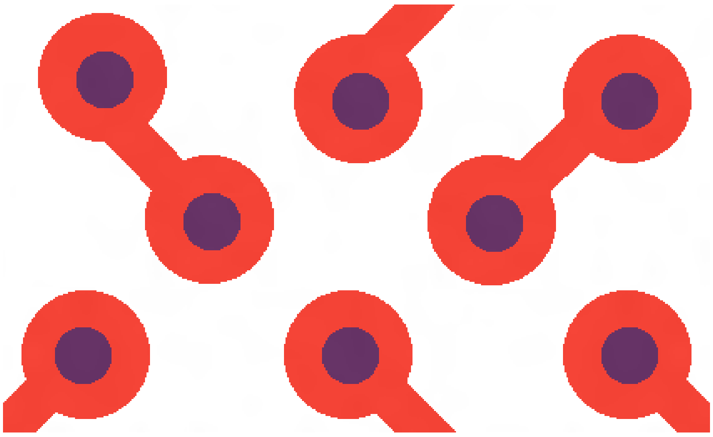


Fig. 7 Drill and copper both in the same format, centered

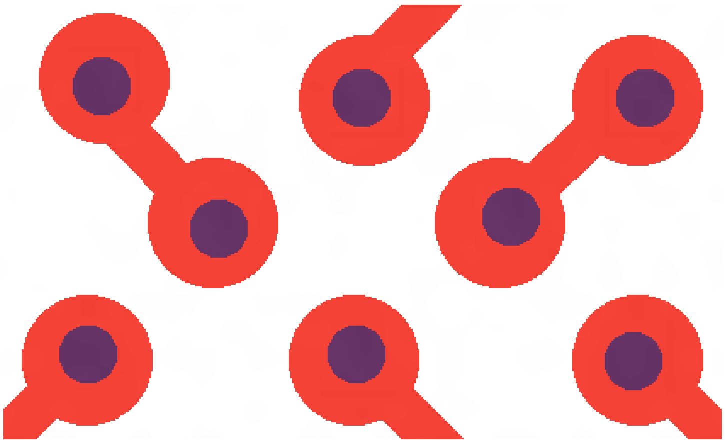


Fig. 8 Drill and copper in different formats, not centered

- **Incomplete NC files**
  All too often – in a whopping 75% of PCB data sets in fact – the NC files are incomplete and require manual input of scale, unit and tool diameters. Such incomplete files relate to a complete NC file as (obsolete) Standard Gerber relate to proper Extended Gerber file. If you use Extended Gerber for image layers it is not terribly consistent to stick to incomplete drill files. See chapter 21. You must use complete drill files. The question is how to do this. Although it may be possible to set up the NC output of your CAM system to generate complete NC files, but the simplest solution is to output in Extended Gerber, which is guaranteed to be complete.
- **Limited resolution of the Excellon files**
  The Excellon format suffers from severe limitations on resolutions, so while it is quite adequate for drilling, it is not recommended for design transfer. There is no such limitation in Gerber.
- **Poor implementation of NC output in some CAD systems**
  Some NC files are really terrible. They can, for example, have binary junk in the middle of the file, or non-standard commands. As images are considered to be more critical than drill files, the quality of Gerber files is typically much better.
- **Needlessly using two formats where one will do**
  Copper layers must be in Gerber as NC files cannot describe a copper image, so Gerber is a given. As Gerber can also describe the simpler drill data, it makes sense to keep things simple by using this format for both copper and drill/rout data. By using an NC format for drill, a second format is needlessly introduced, and the risk of problems is doubled.

There is not a single disadvantage in using Gerber rather than an NC format to express CAD-to-CAM drill data. On the contrary, it brings many advantages in terms of completeness, accuracy, consistency and simplicity.

One may object that we cannot send Gerber files to a drilling machine. True enough, but as outlined above, this is a non-issue. The CAM system will generate the drill files for fabrication in Excellon, Hitachi or Sieb & Meyer or whatever format best suits the drilling machine, and will take care of the offset, rotation, feeds, speeds, tool diameter, resolution, optimisation, size compensation and whatever else is needed for optimal performance on the fabricator’s equipment. Which means that Gerber is hands-down the best format for input in CAM.

One may object that Gerber is an image description format – true again – and that drill data is not image data. This is actually not true. A Gerber copper file describes where there is copper and where there is no copper. Just as a legend file describes where there is legend ink and where there is no legend ink. These files use images to show the presence and absence of material. So do drill and rout files. In this case they describe where material must be removed: a hole or slot is the absence of material. So a drill file, just like a copper layer, is an image file. Yes, drill holes and copper patterns are produced by different processes, but at design stage, they are images. Copper layers and legends are also produced by different processes for that matter, but no one claims they must be described by different image formats. The only difference between drill and copper layers is that the drill file applies to a range of layers, the span, and not to a single layer, and of course, the span must be clearly specified, in both Gerber and NC formats. Here too, Gerber is better: NC formats have no inherent mechanism for describing spans, while Gerber's attributes allow span to be specified in a standard, machine-readable manner.

Conclusion: Gerber is far superior to the NC formats for transferring drill and rout data from design to fabrication.

Output your drill and rout files in Gerber.

8. The layer structure

When a PCB fabricator receives a data set, the first thing he does is to load it into his CAM system to recreate a model of the PCB. That model which will drive manufacturing. To create this model all image files must be converted and the function of each file in the stack must be clearly defined so it is clear which is the top copper layer, which is the solder mask, etc. The layer structure, this is the position of each layer, must be defined - a PCB is not a pile of unrelated images but a strictly ordered set of layers that are laminated together.

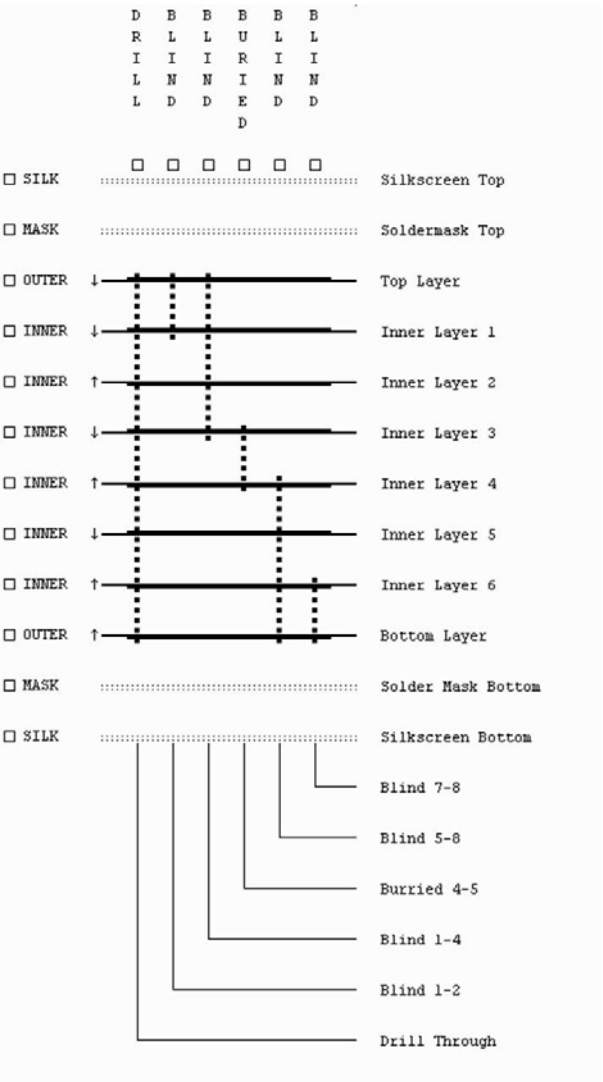


Fig.9 The layer structure

The goal is to define this layer structure in an automatic process following a defined standard to eliminate manual work and subjective interpretation. This is precisely what the Gerber specification's .FileFunction attribute does. To quote from the Gerber specification:

> *5.4.1.1 .FileFunction*
>
> *The .FileFunction file attribute identifies the function of the file in the PCB. The attribute must be set in the file header.*
>
> *Of all the attributes it is the most important.*

When this attribute is present in the data set, the fabricator's CAM software can use it to assign the file to its correct position in the layer structure, automatically and without any need for manual input. Job done. No risk of operator error.

To transfer the layer structure according to the standard you must choose Gerber "X2" output on your layout software, where X2 is shorthand for "Gerber with attributes".

So what should you do if your PCB layout software is not capable of outputting the current Gerber version, with attributes? This is a problem, and indeed the temptation is to suggest that you consider switching to layout software that supports up-to-date Gerber output.

X2 is the Gerber standard to transfer the layer structure. If you cannot output X2 you will need to use an informal method with which to define the layer structure in a human readible form that the CAM operator can use to set the layer structure manually in his CAM system – hoping, naturally, that neither of you make mistakes.

The best option is to express the file function in the file name. The Gerber .Filefunction attribute actually allows defining a file naming convention. The Gerber Standard File Naming (GFSN) convention reflects file function in the file name – it encodes the .FileFunction attribute in the file name. This is the closest you will get to a standard without X2 support.

The GFSN standard file name consists of three parts:

1) A free-to-choose prefix identifying the job. This prefix is common to all files for any one job.
2) A postfix identifying file function. The postfix starts with an underscore ('_') followed by the FileFunction value string where the comma (','), which can create confusion in file names, is replaced by '$'.
3) The standard extension .gbr.

Example:

GJ01911Rev2.1_Legend$Top.gbr
GJ01911Rev2.1_Soldermask$Top.gbr
GJ01911Rev2.1_Copper$L1$Top.gbr
GJ01911Rev2.1_Copper$L2$Inr$Plane.gbr
…
GJ01911Rev2.1_Copper$L10$Inr$Plane.gbr
…

By following the GFNS you clearly identify file function. It is your best option if you cannot use Gerber X2.

Express the layer structure in a standardized way by using Gerber X2. If the standard is not available use clear file names.

9. Drill file structure

Drill files specify geometric information, indicating where material is to be removed and where plated copper is to be added. This geometric data must be specified in a standardized and machine-readable way.

Geometrically, drill holes are characterized by their diameter, their span (start layer and end layer) and whether they are plated or non-plated. The diameter is easily specified by the aperture (tool) diameter in the Gerber drill file - or the NC drill file for that matter. Span and plating are transferred in Gerber with the .FileFunction attribute. To quote from the Gerber format specification:

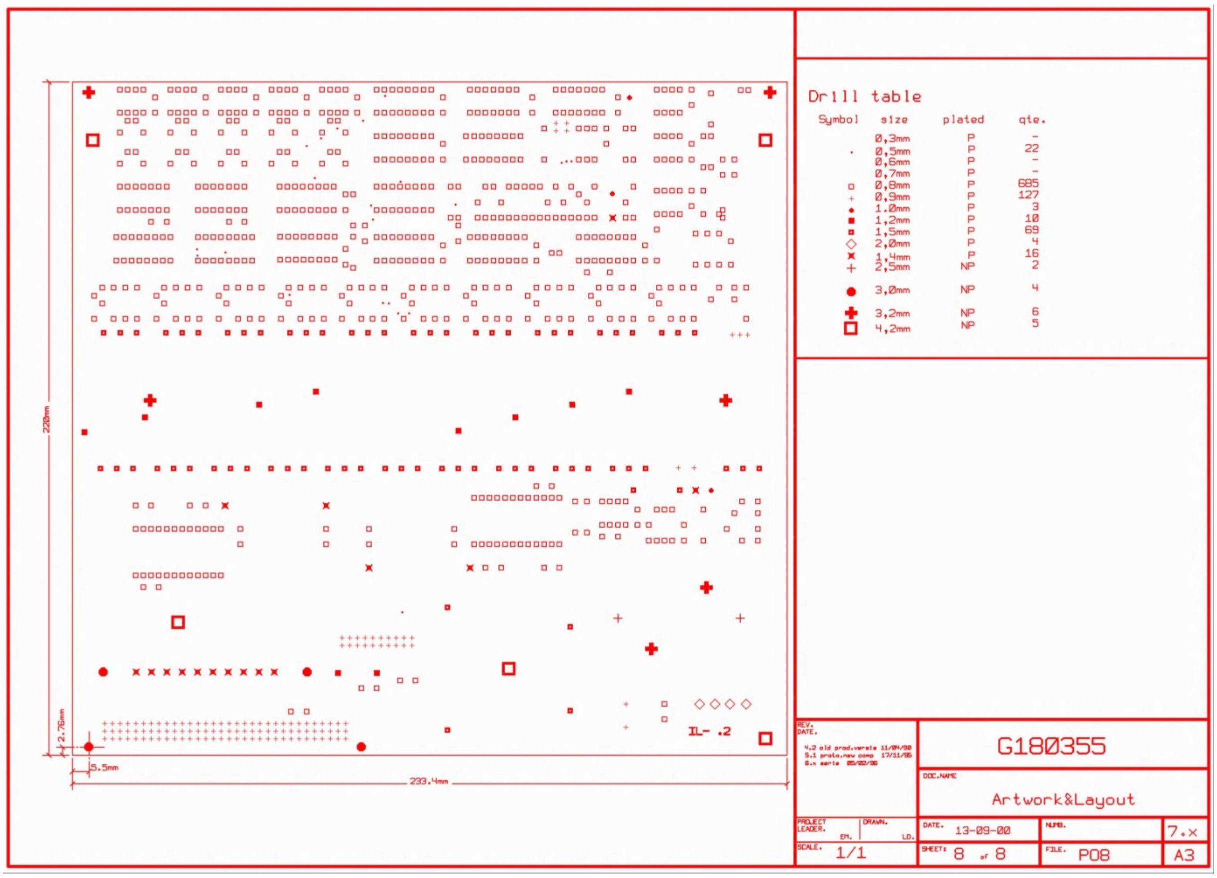
.FileFunction value	**Usage**
`Plated,i,j,<type>[,<label>]`	Plated drill/rout from layer i to layer j. The mandatory type can take the values *PTH, Blind, Buried.* The optional label can take the values *Drill, Route, Mixed*; these labels are not exactly defined; the file creator uses the value he reasonably finds appropriate. .
`NonPlated,i,j,<type>[,<label>]`	Non-plated drill/rout from layer i to layer j. The mandatory type can take the values *NPTH, Blind, Buried.* The optional label is explained in the row above.

As can be seen from this excerpt, the Gerber format provides unequivocal language with which to describe drill span and plating. It should be clear from this that any one file will describe one drill span and one plating instruction – so different spans and plating instructions must be put into separate files. A typical PCB fabrication data set will therefore contain several drill files: one for PTH holes, another for NPTH holes, and others for the different blind and buried spans. By creating data sets in this way, we can ensure that the whole drill file structure is standard and can be read automatically.

If you are not able to add attributes to the file in this way, the CAM operator must manually determine the file function on his CAM system. In this case, you will be providing the information informally, but it should still be as simple and unequivocal as possible. The best way to achieve this is to make the function clear in the file name, e.g. NPTH.GBR. A more indirect method is to list the files and their functions in a text file.

Sometimes both plated and non-plated holes are lumped together in a single file. Some argue that this is OK because the drill map indicates which holes are plated. The drill map does indeed indicate this (usually). But it is not OK. Drill data must be standardized and machine readable. Drill maps are neither standardized nor machine readable – they must be read offline by CAM operators, who must work out the drill co-ordinates visually, then indicate manually which holes are plated and which are not. This terrible practice is bad enough if the plated and non-plated holes are specified using different tool numbers. But where plated and non-plated holes have the same diameter and the file is 'optimised' by using the same tool for both, it becomes really excruciating work for the CAM engineer, and risky for the successful outcome of CAM and production. This is why plated and non-plated holes should always be put in separate files. If this is not possible, at least use separate tools for them.

One more thing. In some cases, fabrication data comes in with just a drill map, and no digital drill data at all. This is simply unacceptable. Drill machines cannot read a drill map: drill machines work CNC data since decades. Without digital drill data, the CAM operator must pore over the drill map, measuring and manually reconstructing the drill data from it, in a laborious and error-prone process. Don't do this - have mercy on the poor CAM engineer!



Drill table

Symbol	size	plated	qte.
	Ø,3mm	P	-
·	Ø,5mm	P	22
	Ø,6mm	P	-
	Ø,7mm	P	-
□	Ø,8mm	P	685
+	Ø,9mm	P	127
•	1.Ømm	P	3
■	1,2mm	P	1Ø
▣	1,5mm	P	69
◇	2,Ømm	P	4
⌘	1,4mm	P	16
+	2,5mm	NP	2
●	3,Ømm	NP	4
✚	3,2mm	NP	6
□	4,2mm	NP	5

REV. DATE. 4.2 old prod.versie 11/04/98 5.1 proto.new comp 17/11/95 6.x serie 05/02/96	G18Ø355		
	DOC.NAME Artwork&Layout		
PROJECT LEADER. EM. | DRAWN. LD.	DATE. 13-Ø9-ØØ	NUMB.	7.x
SCALE. 1/1	SHEET: 8 of 8	FILE. P08	A3

Fig. 10 A drill map. Useful maybe, but not a substitute for drill data

Note that the considerations in this chapter apply whether the drill files in your fabrication data sets are in an NC or Gerber format. (In chapter 7 we argue that Gerber is by far the best choice). Whatever format you choose, the bare minimum is that you create a separate file for each span, and for plated and non-plated holes, and that you clearly indicate which is which.

These requirements for drill data also hold true for rout data. Even though drilling and routing are very different fabrication processes, the difference between them is largely irrelevant when they are viewed as fabrication image data: both simply indicate where material is removed. Indeed, the fabricator may very well nibble a slot in a rout file, or rout a large hole. In conclusion, the fabrication data must specify what the fabricator must fabricate. The fabricator will decide how to fabricate it.

Put plated and non-plated holes in separate files.

10. Junk, or the difference between data and drawings

The core of a PCB fabrication data set is the *digital data* files. These data files will be read into CAM to recreate a model of the PCB and then used for further digital processing. There must be a digital data file, in Gerber, for each pattern in the PCB: copper layers, drill and route files, solder masks, legends, peelable and whatever other patterned layers are present.

A PCB fabrication data set may contain technical drawings -a mechanical drawing is often absolutely needed. Technical drawings are not intended for digital processing but for a human to look at. A technical drawing is made according to time-hallowed rules. It includes a frame, title block, notes and legends, as in Fig. 11 below.

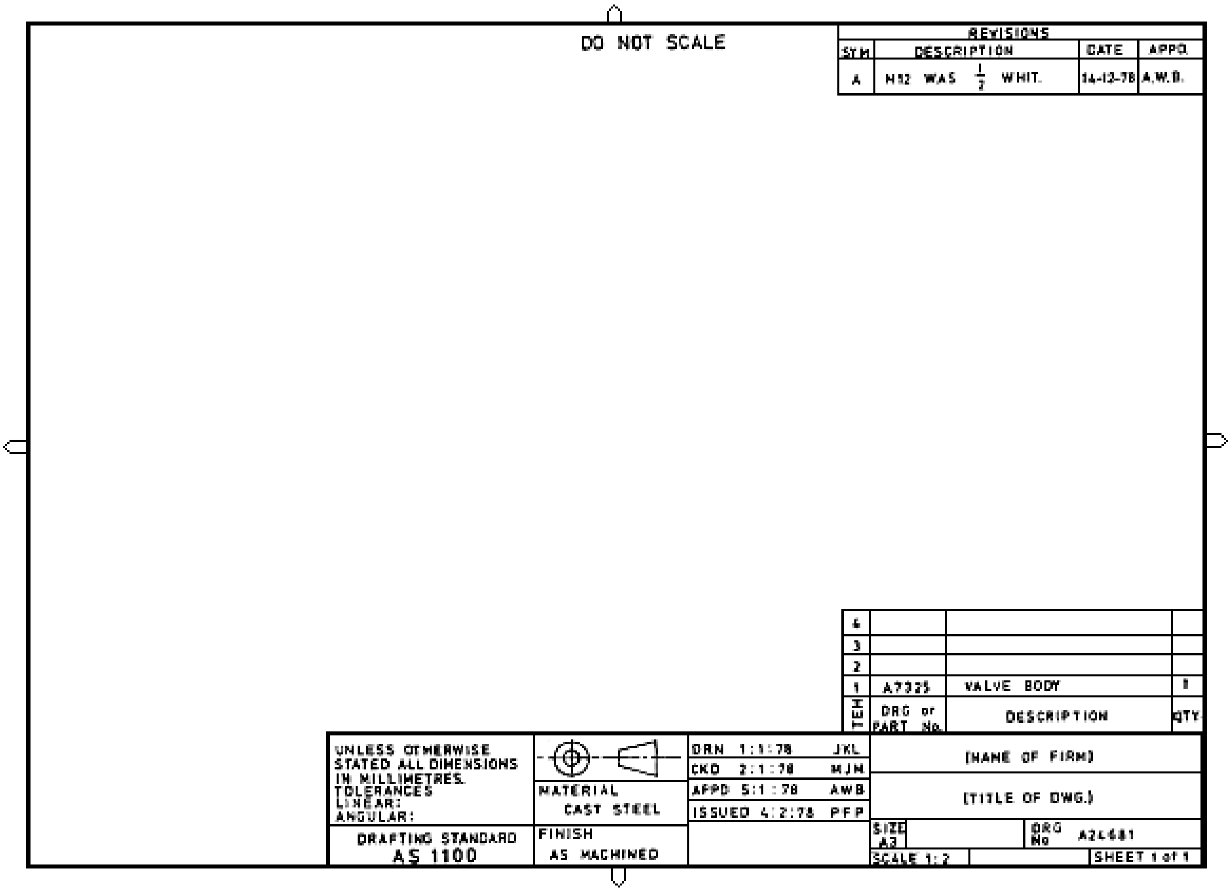


Fig. 11 A typical frame for a technical drawing

As drawings are images they too are best transferred in Gerber – see chapter 14.

Alas sometimes digital data and drawings are confused and a drawing frame is added to digital date. This results in a copper layer as in Fig. 12 below.

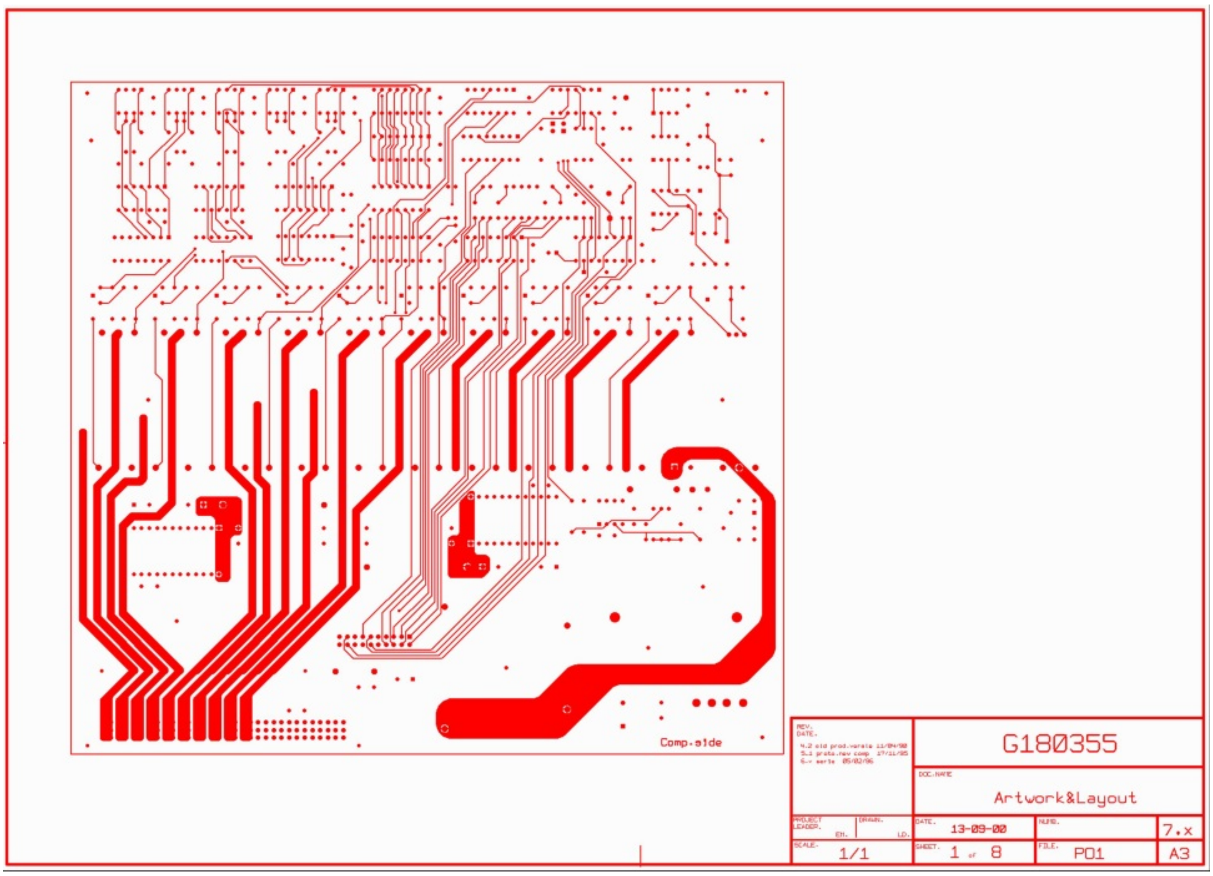


Fig. 12 Confusing a drawing and digital data

While the frame is essential in a drawing, it becomes “junk” when added to a digital data file such as a copper layer. Drawings are intended to be looked at by humans. Digital data is intended for digital processing in CAM software – the frame is then junk that must be manually removed by the CAM operator before the data can be used. It is not because digital data represents graphic information it suddenly becomes a drawing. It is not because drawings and graphical digital data are both expressed in Gerber that are both drawings. Use only pure data files, without junk or embellishments.

You may object that the title block contains useful information. This may well be. The solution is to put that information in a separate text file or a true drawing. The data is then pure and can be used without manual cleanup and the information intended for the human operator is conveniently available in a separate file.

Put only data in data files.

11. Always include the netlist

Basically, a netlist is a set of nets, where each net has a name and a set of nodes identified by their XY coordinates. Nodes on the same net must be electrically connected. Nodes on different nets must be isolated.

Including the netlist in the PCB fabrication data set increases the security of data transfer by an order of magnitude. The first thing a fabricator does after reading a fabrication data set into his CAM system is to generate a netlist from the image, called the reference netlist. During the CAM process the CAM engineer will regularly check the job data against the reference netlist to protect against operator or software errors. When a netlist is present in the incoming fabrication data set, he will also check his reference netlist against the supplied netlist. Any serious errors in the images or drill files will inevitably result in netlist differences and set off an alarm. The presence of the netlist in the fabrication data sets protects against mistakes in data transfer, whether these are due to software or operator error, in CAD output or CAM input. Adding the netlist to the fabrication data sets extends the regular netlist check performed by the CAM operator to encompass not only the CAM process, but the complete CAD to fabrication data transfer.

Now, errors in the transfer of image data from CAD to CAM are rare, but they do happen. And they are very costly when they happen. Without the netlist the fabricator cannot know he is working from a wrong image. He will faithfully manufacture the wrong PCB, which will pass his electrical test as it is tested against the netlist created from the wrong data. The error will only become apparent after the PCB is loaded with components, at which point the costs are staggering. All deadlines are missed, recriminations fly, and the search for the guilty starts. Simply including a netlist largely protects against such a rare but dramatic event – it is like installing smoke detectors and sprinklers. Fires do not occur often either but everyone takes steps to protect against them.

The netlist is a powerful check on the image data – it is akin to the checksums that are widely used to make data transfer reliable.

Including the netlist is simple. It is sufficient to include an IPC-D-356A file in the data set. IPC-D-3356A contains all the necessary information, and more. Virtually all manufacturers can read IPC-D-356A and most CAD systems can output it.

It is sometimes claimed that comparing netlist and image data throws up many false errors. This is largely a problem of the past when implementations were new and buggy. Nowadays most netlist files are OK.

There is another way to view this: the netlist is the basis of any PCB layout, and the essential function of a PCB is to physically implement that netlist. The PCB fabricator is expected to electrically test the fabricated boards and to guarantee that the shipped boards' netlist is correct. It is therefore of paramount importance that the fabricator works from the correct netlist, so it is an obvious requirement that the netlist be supplied to the fabricator rather than leaving him to reverse engineer it from the images.

A professional PCB fabrication data set must include a netlist. Omitting it amounts to a self-inflicted competitive disadvantage.

Because the inclusion of a netlist is so simple and is such a powerful security check on the data, Ucamco's position is that if a data transfer error occurs that would have been flagged by checking back to a netlist, the responsibility lies at least partially with the party that has omitted to include a netlist, or that has omitted to use a supplied netlist.

Always include an IPC-D-350A netlist file in the PCB fabrication data set.

12. Pads on copper layers

The main function of copper is to conduct. However, pads also have other functions. SMD, component, connector and test pads on outer layers provide electrical access to the circuit. They will carry solder, paste, platings. Thermal relief pads confine heat. Via pads too are mainly there for conduction but still, they not only electrically connect the track the barrel but also provide mechanical support for the barrel and provide room for the tolerance on drill positioning.

Tolerances on pads are often tighter than on conductive copper – pads are treated differently from mere conductive copper when compensating for production tolerances. In a solder mask via pads are treated completely differently from SMD pads. To gold plate connector pads a gold mask must be created. The position the probes of an electrical test machine the location, size and shape of the pads must be known.

CAM must not only ‘know’ where and what the pads are, CAM must also be able to edit them efficiently. Consequently, pads of different types must be separate classes of objects so that they can be modified separately.

The way to do this is in Gerber is simple: represent all pads by a flash of an aperture, and only use flashes for pads. Different types of pads are identified by different apertures. Even of a component pad and via pad have the same size and shape they must still have a different aperture number. Thus is easy to select all pads with a given function.

Painted (stroked) pads are evil because they are so hard to edit. If a Gerber file just served as an image painted pads would be fine. However, a Gerber file must also be editable, and painted pads are not.

All pads must be flashes. All flashes must be pads.

Two other remarks about copper pads.

Embedded pads are pads fully included in a bigger copper area, e.g. a via pad in an SMD pad, or an SMD pad on a copper pour. Image wise these pads have no effect – whether they are present or absent, the image remains the same. However, CAM must know where the pads are. The Gerber file does not only transfer the image, but also pad information. Thus, the embedded pads must be present. Don't 'optimize' your output by removing embedded pads!

Sometimes pads on a plated hole are represented by a donut rather than a solid pad. Presumably this is to make room for the drill hole or to indicate that copper will be removed. And it is true, copper will be removed. However the donut is very impractical because it does not fit how the copper is removed. The copper is not removed by imaging. First a solid pad is created by imaging, and then the pad is drilled. Consequently CAM must laboriously replace all the donuts with solid pads to prepare for imaging. Furthermore with non-plated pads which typically truly are donuts plated donuts are confusing. Drill plated pads must be solid. Whatever you want to express by using a donut, express it in another way.

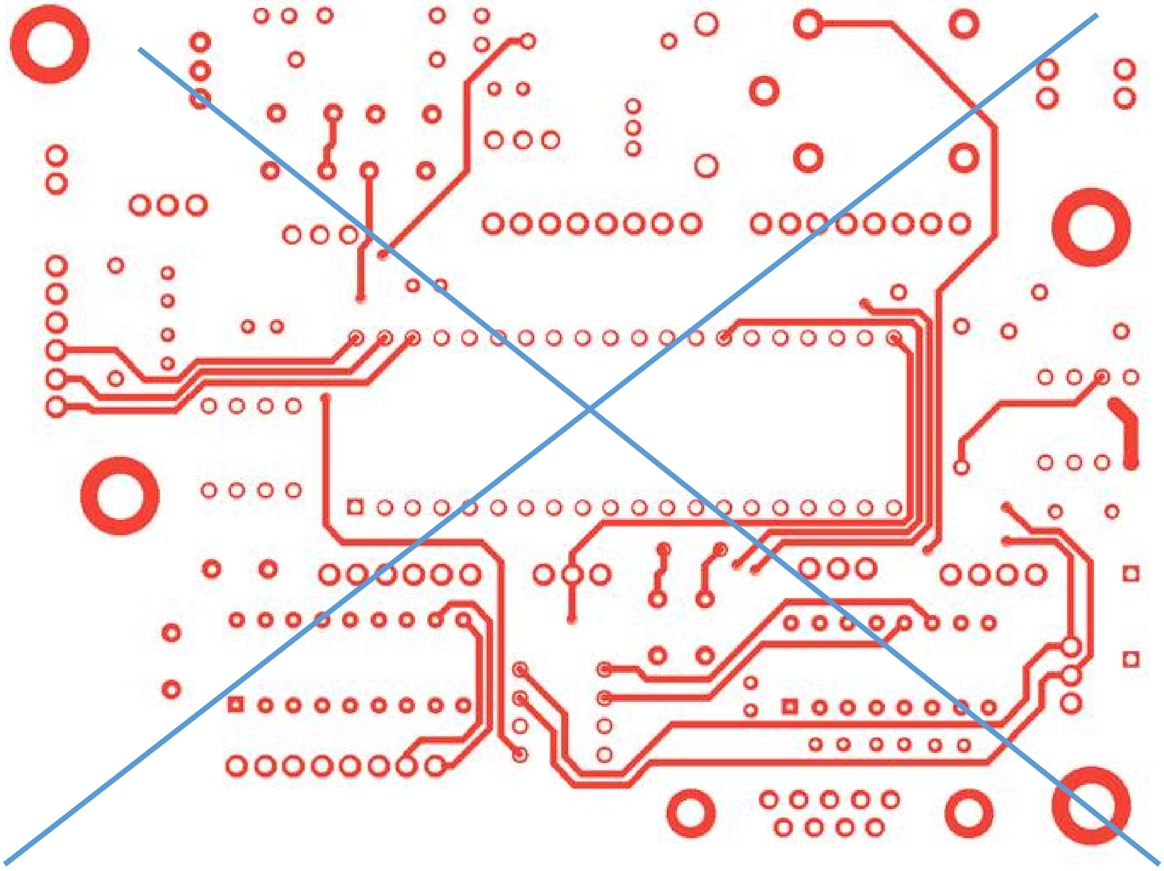


Fig. 13 All donut pads, very confusing.

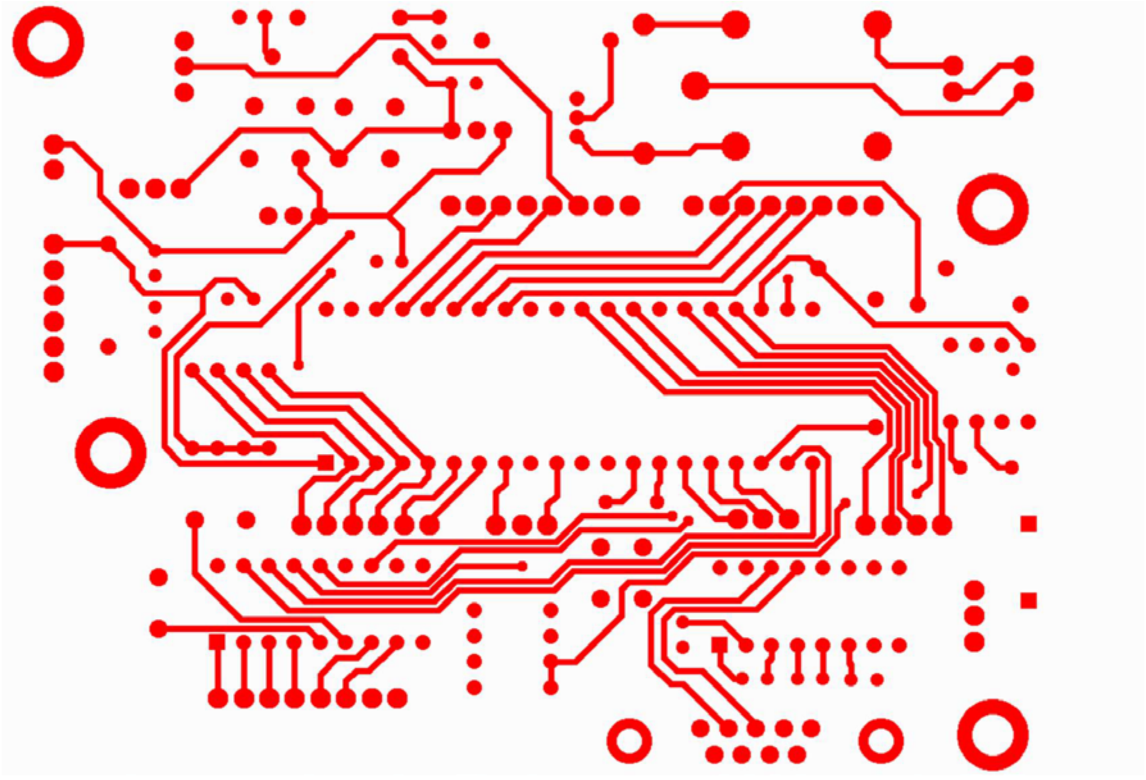


Fig. 14 Proper pads, everything is clear.

13. Drawings are no substitute for data

Drawings may be a useful part of a PCB fabrication data set, but they are not a substitute for digital data. . There must be a digital data file for each pattern in the PCB: copper layers, drill and route files, solder masks, legends, peelable and whatever other patterned layers are present.

For example adding a drill map may be useful but it is not a substitute for a proper drill file. Drilling is done with a CNC drill machine. This machine needs CNC data, generated by CAM. If the fabricator only has a drill map, how do you expect him to generate the drill data? Visually, by measuring the drill map on an XY table and typing in the coordinates?

Another example is the excellent profile drawing below. This drawing defines and illustrates the profile perfectly. The profile drawing may still be useful as a check, but it does not replace digital data specifying the profile – see chapter 6. Digital data is read in automatically. Note the drawing does not exactly specify the position of the profile versus the copper layout – profile digital data does.

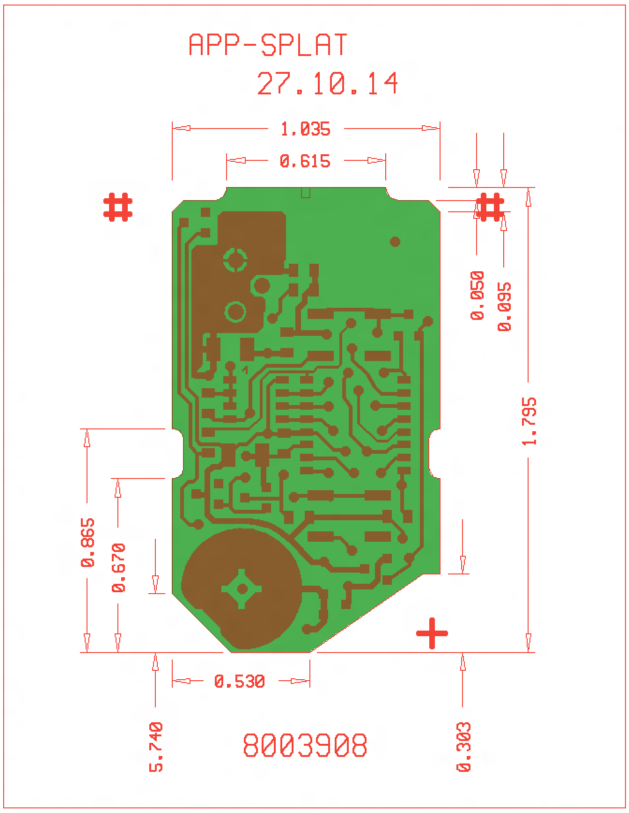


Fig. 15 A profile drawing

Drawings are no substitute for digital data.

14. Use Gerber format only for your image data

Keep data formats to a minimum. Use only those that are truly needed. Every extra format an additional adds output and input processor is, increasing the risk of bugs and version problems. Mixing formats increases the risk of misalignment between files – see chapter 5.

Copper layers must be expressed in Gerber. So Gerber is a given. Consequently, what that reasonably *can be* expressed in Gerber *must be* expressed in Gerber.

Drawings are images, can and therefore must be transferred in Gerber. Other formats are often used for drawings: PDF, HPGL, DXF, DWG etc. These may be fine formats – and DPF is definitely a first rate data exchange format - for drawings in PCB fabrication data Gerber is the better choice: your fabricator needs to read your drawings into his Gerber-capable CAM system to relate it to and shed light on the image data – which is, after all, the very reason why you sent the drawings at all. His CAM system is definitely proficient in handling Gerber data but it was not designed to handle other formats. So although PDF may be a better choice for other workflows, Gerber is the format for drawings in the PCB workflow.

Avoid complicated formats such as DXF and DWG like the plague. You cannot expect your professional CAM operator, who is highly skilled in the Gerber format, to have access to, and be familiar with, the high-end professional software that would be necessary to handle such formats. Simpler software is often of mixed quality, and not safe or reliable enough to faithfully transfer your professional work.

Definitely do not use DXF or DWG for data files such as copper layers. These formats were not designed for PCB data and are not suitable for it. Such files are loathed by CAM operators.

In fact, the number of formats needed is very limited. Copper, drill, rout, solder mask, legend must all be expressed in Gerber, and so do the drawings. The netlist cannot be expressed in Gerber, IPC-D-356A must be used. The informal data intended for human eyes – delivery info for example can be expressed in plain text or PDF files. More formal data is expressed in structured text files such as CSV, XML or YAML. Consequently the only formats you need are:

- Gerber
- IPC-D-356A
- Text files
- PDF possibly

Output *all* your drawings in Gerber.

15. Non-image data

A PCB Fabrication set contains general PCB specifications such as thickness, finishes, ROHS compliance etc. There is also commercial data: number ordered, addresses, delivery time.

There is currently no common standard to govern this type of data. They are typically put in informal or formal data plain text of in PDF files intended for human eyes. PDF is a fine format and widely supported but for PCB fabrication data remarkably enough the best choice is plain text: many fabricators have software that automatically extracts information from this sort of text format. PDF is too complex for automatic data extraction. Formal information is best expressed structured text files: comma-separated-values (CSV), XML, JSON or YAML. Such files are easiest and safest to automate.

Often, some of this general PCB data is put in drawings. This is an excellent and time-honored practice in all fabrication industries. However, also take into account automatic data extraction. Data cannot realistically be extracted from drawings. Automation protects against operator error. Therefore it is advisable put all your general data in text files, even if it is already in a drawing.

Excel, Word and Power Point formats are also used. This is harmful. This introduce yet more formats, and complicated on top. These are proprietary *application formats* and not *data exchange formats*. They are *not open* formats but their specification is rightfully a closely guarded secret. Automated data extraction is well-nigh impossible. They require dedicated interactive applications and cannot be built into automated workflows. Version problems abound, e.g. between open source and Microsoft software. These formats were not designed for data exchange and must not be used for it. If you have a choice between such formats and PDF choose PDF without hesitation, but plain text files are best.

Put all your general data in human readable text files, informal or structured as in CSV, XML and YAML.

Do not use application formats such as Word or Excel format.

16. The content of fabrication data set

All files in the fabrication data set is stored in an industry standard archive format, e.g. rar, zip, 7z. The archive contains the data of *one single* PCB. Putting more than one PCB in an archive is confusing – no clever file naming scheme or comprehensive explanatory note can be as clear to identify which files belong together as putting only the files that belong together in the archive. One PCB, one archive.

The question is what must be, and what must not be, in the fabrication data set. Obviously,the archive must contain all data needed to fabricate the board, in a standard and unequivocal manner. Less obviously, the archive must not contain other files, no superfluous data nor duplicates. The reason is that the manufacturer must check each file to see if it contains relevant instructions. Superfluous files waste his time and increase the risk of missing something essential. A typical superfluous file is an aperture list – not needed as a Gerber file itself contains all the required aperture information. CAD data is also useless is requires the CAD software to handle it. Duplicate information is even worse, the fabricator must compare the different files to check whether they contain conforming instructions, and conflicts raise the question what is now valid. Duplicate image information, e.g. in Gerber and ODB++ format is especially aggravating as images are complex and hard to compare – which tolerances apply?

Mandatory is *one single* Gerber file for *each* patterned layers (copper, solder mask, legend, etc.) and for *each* drill sequence present. (Supplying only one solder mask if top and bottom is equal violates this elementary rule. This obnoxious habit save a few bytes but then begs the questions whether a mask was forgotten and which masks must now effectively must be on the board. Confusion. The space saving is illusory as the first thing the CAM operator must do is to create two masks from the single one. A bigger archive is better than a more confusing archive. When both masks are present, even if they are identical, everything is clear.) Mandatory is also all fabrication drawings – in Gerber format -and fabrication instructions such as finishes, ROHS etc. And what is not mandatory must not be there.

One seeming exception to avoiding duplicate data is the rule to include a netlist – see chapter 11. Actually it is not an exception as the CAD netlist is genuinely original data– and essential – information as the netlist generated from the image data is only a reverse-engineered approximation. Adding it is a powerful checksum on the data.

Make the fabrication data set as simple as possible, but no simpler.

17. Negative copper layers

Negative layers are a relic of the 1960s and 1970s and the age of the vector photoplotter, now as obsolete as the mechanical typewriter. The vector photoplotter was similar to a pen plotter, but instead of using ink and paper, it wrote onto photosensitive film using a stationary light “pen”. The film was held firm on a flat table that moved in the X-Y plane under the pen's light beam which was switched on and off as the image dictated. Every movement was governed by commands in input Gerber files.

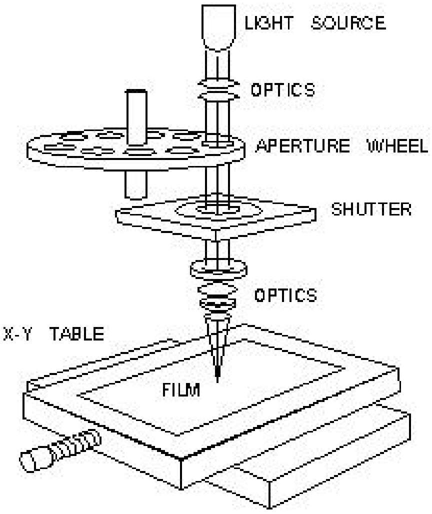


Fig. 16 Vector photo plotter

This was fine for drawing tracks. The problem started with planes, or anything with large copper pours such as that shown in the image below:

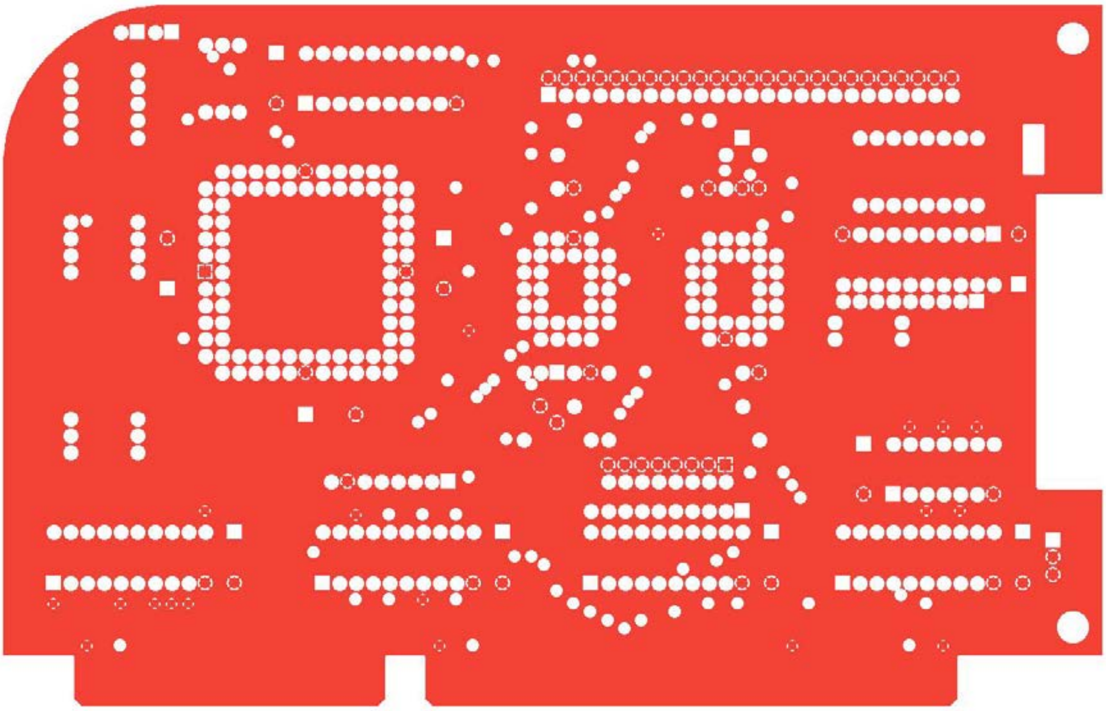


Fig. 17 A plane layer

This is because vector plotters created these large copper pours using a technique called 'vector-fill', 'painting' or 'stroking'. This involved repeatedly moving the table back and forth under the pen; just as a child, when colouring a drawing, moves a crayon back and forth over an area until it is completely filled. See the image below:

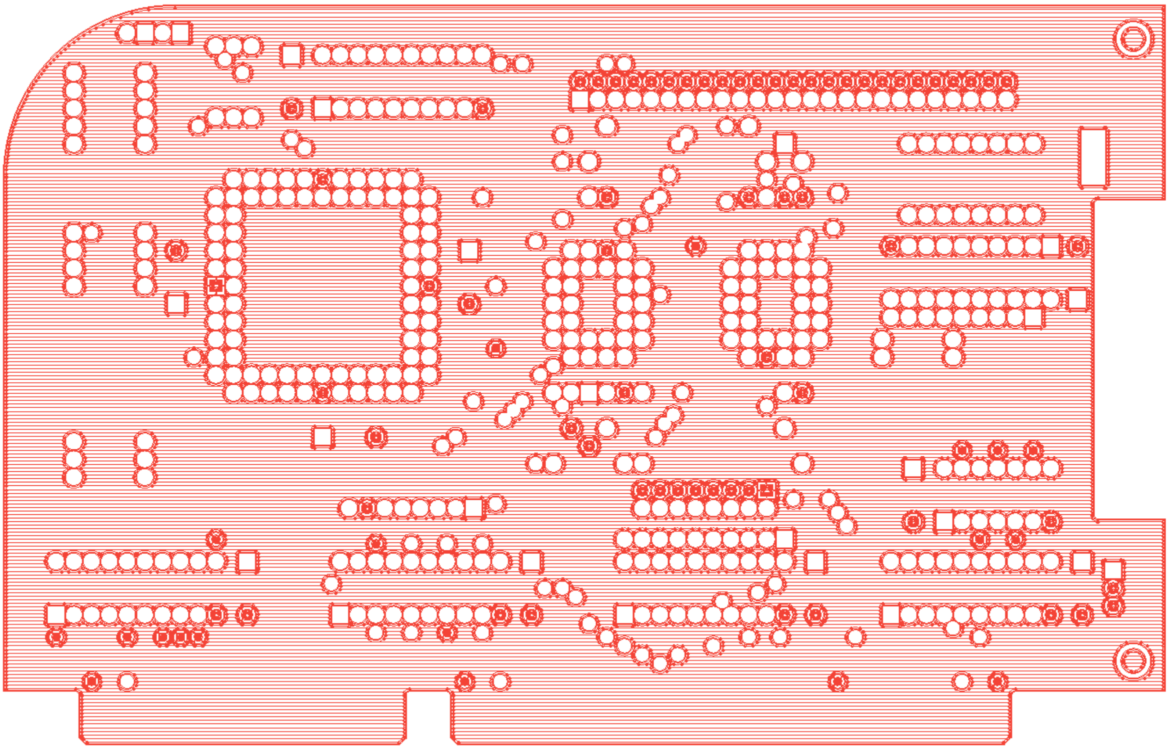


Fig. 18 A stroked plane layer

In principle this worked. But in practice, the input Gerber file, containing the zillions of draws that were needed for the vector fill, was huge. More importantly, it took ages – easily an entire shift – to plot a plane, so it was highly impractical.

The solution was to plot in negative, in other words, creating the clearances rather than the copper pours.

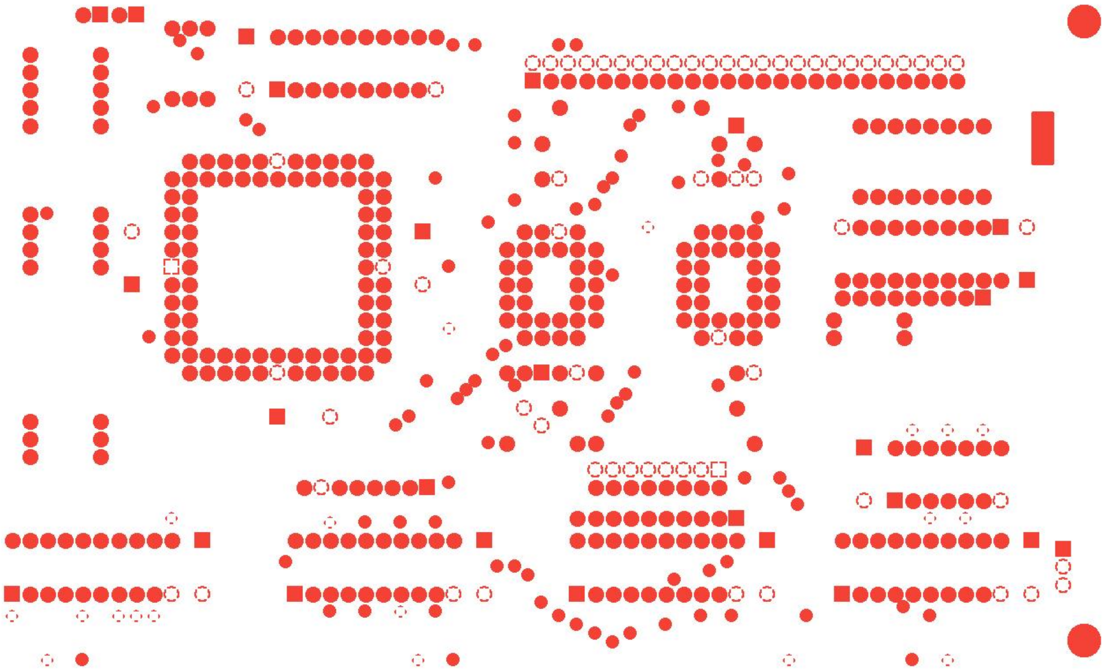


Fig. 19 A plane layer in negative

This eliminated the need for painting, and the Gerber file size and plotting times were kept to a minimum. The negative film thus created was then used in the photolab as a phototool, to generate the positive film that was necessary for downstream production processes. Although this added a step and involved manual work, it was infinitely better than blocking the expensive photoplotter for a whole shift with impatient customers breathing down one's neck and demanding their plots. Negative films made a lot of sense in the days of vector photoplotters.

But time has moved on since then, and so have printing devices. For decades, plotters and direct imagers have used raster-scan technology, whose speed depends soley on image area and resolution. Image content no longer dictates throughput so same-size planes and signal layers take exactly the same time to plot.

So negative files no longer offer benefits. On the contrary, they introduce some serious disadvantages:

While positive layers have clearly defined limits, negative layers do not, so arbitrary limits must be imposed.
In a mixed data set, there is no standardized method by which to define which layers are negative, so manual reverse engineering is necessary. When all layers are positive there is no problem.

Most importantly, negative layers do not contain copper pour outlines, so these must be created. This can be done by maintaining specific clearances from the profile, for example, but this is still guesswork and reverse engineering. In positive layers, on the other hand, copper outline is clearly defined.

How to define a positive copper plane:

```
G04 We define the antipads
%TF.AperFunction,AntiPad*%
%AD11C....*%
....
G04 We now define the extent of the copper pour*
LPD*
G36*
X...Y...D02*
X...Y...D01*
...
G37*
G04 And now we flash clearances
%LPC*%
D11*
X...Y...D03*
....
```

This gives CAM clear and unequivocal information which is robust, numerically accurate, and the anti-pads register with the drill files. And it's as compact as a negative file, and clearly defines the extent of the copper. Perfect.

There is no longer any benefit in transferring the data in negative but it adds confusion, manual work and risk. It is relic of the bad old days.

Always use positive copper layers.

18. Number of digits used for co-ordinates

Use the same resolution and number of digits (resolution) for *all* data layers. If you use different resolutions for different layers, the position of elements will be rounded differently. The result is that, say, via pads and drill pads end up being out of alignment. You can only hope then that the CAM engineer notices this and ‘snaps’ the via pads to the drill holes. This of course changes the copper image, which is something the CAM engineer is not supposed to do, but he is also supposed to deliver a perfectly aligned PCB. So he's damned if he does, and damned if he doesn’t.

Misalignment sometimes arises even when the same resolution is applied to all layers. This is usually because the output processors for Gerber and for Excellon round differently. As for other issues mentioned above, the simplest solution is to output the drill/rout layers in Gerber. See chapter 5.

Ideally, Gerber files use 6 decimal places in inches and 5 or 6 decimal places in mm. If your software cannot produce these resolutions then get as close as you can. Some recommend the use of lower resolutions, probably to save a few bytes. Who cares about saving a few bytes? Don’t follow this recommendation: lower resolution increases rounding errors. This not only affects alignment, but, more importantly, it increases the risk of serious problems because these rounding errors can make contours self-intersect, which is invalid – and result in invalid arcs. Low resolution is the root cause of most of the few image errors in Gerber files; it does not happen often but it makes no sense to take this risk to save a few bytes. Another reason given to use low resolution is that the drilling machine may not be able to handle big files. This may well be true but it is irrelevant: as we have already seen, the CAM system will output drill files exactly as the driller needs them.

Use the maximum precision for coordinates.

Output all layers in the same precision.

19. The file extension

Wikipedia states: “A filename extension is a suffix (separated from the base filename by a dot or space) to the name of a computer file applied to indicate the encoding (file format) of its contents or usage. Examples of filename extensions are `.png, .jpeg, .exe, .dmg` and `.txt.`” Microsoft states: “A file name extension is a set of characters added to the end of a file name that determine which program should open it.” The advantage of this rule is that the file format is clear without first opening the file, and, consequently, so is the preferred application for the file.

To quote from the Gerber format specification:

> *The Gerber Format has a standard file name extension a registered mime type and a UTI definition.*
>
> ***Standard file extension:*** *.gbr or .GBR*

Use the standard file extension `.gbr` or `.GBR.` on all Gerber files.

All too often names such as pn674847.top are used, meaning the top layer of job PN674846. This idea dates from the 1980s – the days of MS-DOS, the first Microsoft operating system. The file names in MS-DOS were restricted to a measly 8 characters. Abusing the file extension to gain a few extra characters may have made sense in those days, but it makes no sense today.

File extensions such as .ger, .pho, etc. are also sometimes used. They do indeed express the format, but in a proprietary, non-standard way. Always use the standard extension `gbr` or `.GBR.`

One may object that the supply chain is used to a file name convention such as in pn674847.top and it is not possible to simply throw it overboard. Fine: give this file its full name, in this case, pn674847.top.gbr. This is a valid and the old file name is still prominent.

Always use the standard file extension “.gbr” for Gerber files.

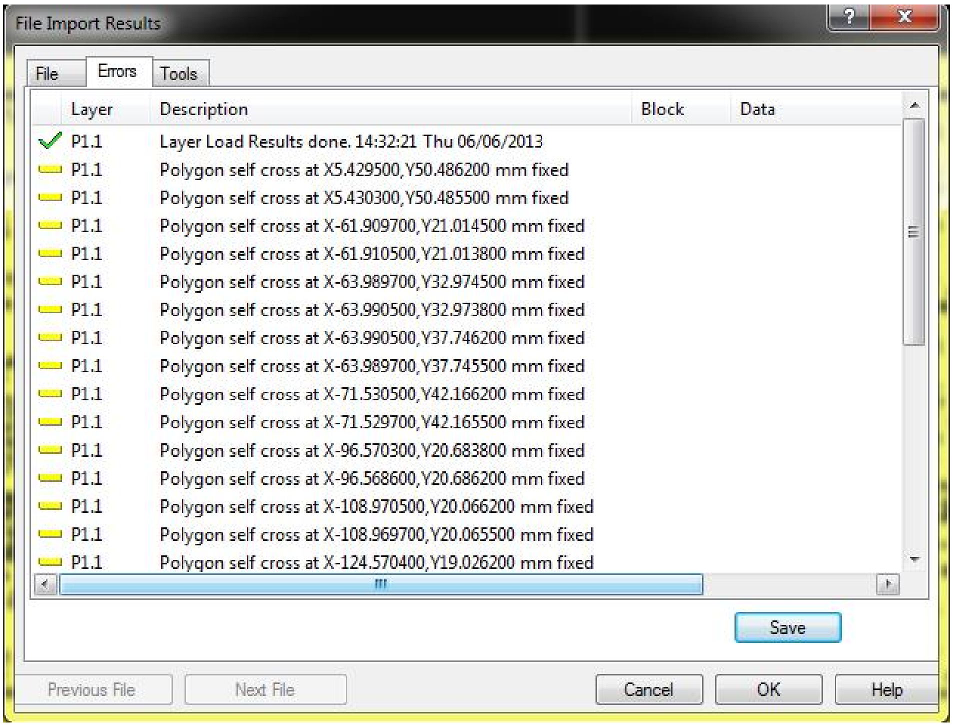
20. The use of Gerber viewers

Before sending your Gerber files off to your fabricator, you are often advised to check them using a reputable Gerber viewer such as GC-Prevue. This is excellent advice.

Note that this involves more than just verifying that the viewer displays your intended image: it is important that you check too that the file is valid. Even when handling invalid data, viewers typically try to reverse engineer the intended image by 'reading between the lines'. This is perfectly OK, but the file is still invalid and, according to Gerber specification: *An invalid Gerber file is meaningless and does not represent an image.*

A file with errors must not be sent to the fabricator as if all is well, even if the intended image is shown. This is because even if *your reader* has reverse engineered the intended image from the invalid data, *another reader may not be so successful*. And that reader may be your fabricator's CAM, which will result in scrap. Should this happen, the fault lies squarely with the file. To quote from the Gerber specification: *The responsibilities are obvious and plain. Writers must write valid and robust files and readers must process such files correctly. Writers are not responsible to navigate around problems in the readers, nor are readers responsible to solve problems in the writers.*

It is therefore extremely important that you check that your files are valid. Invalid files can cause viewers to throw error messages like the one below, taken from GC-Prevue:



	Layer	Description	Block	Data
✓	P1.1	Layer Load Results done. 14:32:21 Thu 06/06/2013		
▬	P1.1	Polygon self cross at X5.429500,Y50.486200 mm fixed		
▬	P1.1	Polygon self cross at X5.430300,Y50.485500 mm fixed		
▬	P1.1	Polygon self cross at X-61.909700,Y21.014500 mm fixed		
▬	P1.1	Polygon self cross at X-61.910500,Y21.013800 mm fixed		
▬	P1.1	Polygon self cross at X-63.989700,Y32.974500 mm fixed		
▬	P1.1	Polygon self cross at X-63.990500,Y32.973800 mm fixed		
▬	P1.1	Polygon self cross at X-63.990500,Y37.746200 mm fixed		
▬	P1.1	Polygon self cross at X-63.989700,Y37.745500 mm fixed		
▬	P1.1	Polygon self cross at X-71.530500,Y42.166200 mm fixed		
▬	P1.1	Polygon self cross at X-71.529700,Y42.165500 mm fixed		
▬	P1.1	Polygon self cross at X-96.570300,Y20.683800 mm fixed		
▬	P1.1	Polygon self cross at X-96.568600,Y20.686200 mm fixed		
▬	P1.1	Polygon self cross at X-108.970500,Y20.066200 mm fixed		
▬	P1.1	Polygon self cross at X-108.969700,Y20.065500 mm fixed		
▬	P1.1	Polygon self cross at X-124.570400,Y19.026200 mm fixed		

Fig. 20 Error messages in a viewer

These messages clearly indicate that there is something very wrong with the file. The question is, what you do if you see such errors? Not easy. Low resolution is often the root cause of problems, so it is worth trying to output the file at the resolution recommend in chapter

The only safe solution is to fix the bugs in the Gerber output software. It is therefore essential that you provide detailed information of the problem to your software supplier so that the bug can be fixed for the future. That said, the chances are that your board cannot wait for this fix and you have no way to output a valid file. This is then a conundrum. You could send the invalid data with the necessary caveats and hope that your fabricator's software, like your reader, will reverse engineer the intended image correctly. If it does, all is well. But this is a risk, so if you decide to do this, always include a netlist as a safeguard, see chapter 11. You can also ask your fabricator to send you the images he generates in CAM, so that you can check them for errors.

Check the error messages of your Gerber viewer.

21. Standard Gerber

To quote from the Gerber format specification:

> *Standard Gerber is revoked and superseded by Extended Gerber, which is the current Gerber format. Consequently, Standard Gerber no longer complies with the Gerber specification. Files in that format can no longer be correctly called Gerber files. Standard Gerber files are not only deprecated, they are no longer valid.*

Please use Extended Gerber for all your operations. Standard Gerber is technically obsolete. If you are still using it, you are putting your business and that of your clients and business partners at a useless risk, without benefit:
Despite its name, Standard Gerber is not a defined standard for PCB data transfer: Units and aperture definitions, rather than being governed by a recognizable standard, are in an informal document, the interpretation of which is unavoidably subjective. As a result, Standard Gerber files cannot be machine-read in a standardized, reliable way. Standard Gerber requires aperture painting and copper pours, both of which create manual work in CAM, adding cost, delay and risk to the PCB manufacturing process. Standard Gerber does not support attributes.

Extended Gerber files *are* machine readable, they do *not* require painting, and they *do* support attributes. Virtually all software read Extended Gerber and many new implementations no longer support Standard Gerber. There is not a single good reason left to use Standard Gerber. Using of Standard rather than Extended Gerber is a self-inflicted competitive disadvantage.

To quote from the specification once more:

> ⚠ ***Warning:*** *The responsibility of errors or misunderstandings about the wheel file when processing a Standard Gerber file rests solely with the party that decided to use revoked Standard Gerber, with its informal and non-standardized wheel file, rather than Extended Gerber, which is unequivocally and formally standardized.*

More information can be found it the Open Letter on Standard Gerber at the download page of www.ucamco.com

Do *not* use Standard Gerber.

22. References

The Gerber File Format Specification, www. ucamco.com

PCB Fabrication Data in Gerber Example 1, www.ucamco.com

PCB Fabrication Data in Gerber Example 2, www.ucamco.com

Open Letter on Standard Gerber, www.ucamco.com

Painting Considered Harmful, Karel Tavernier,
The PCB Design Magazine • November 2013

Kick-Starting a Revolution, IPC-2581 meets Gerber, Karel Tavernier,
The PCB Design Magazine • January 2013

Gerber Grows Attributes, Karel Tavernier,
Printed Circuit Design & Fab Magazine, August 2013

Improving CAD to CAM data transfer: a practical approach, Karel Tavernier
Journal of the HKPCA, nr. 40, 2011 Q2, 2011