

Gent, June 2014

Open Letter to the Gerber User Community

As developer of the Gerber format, Ucamco communicates the following about Standard Gerber. Please feel free to forward it to anyone who may be concerned.

The Standard Gerber specification was revoked in 2014.
Standard Gerber is superseded by Extended Gerber.
Please use **Extended Gerber** for all your operations.

Standard Gerber It was revoked in 2014. Since then, Standard Gerber files do not conform to the Gerber specification. **Standard Gerber files are no longer valid. In fact, they can no longer be called Gerber files.**

Standard Gerber is technically obsolete. It has significant drawbacks vs Extended Gerber and carries inherent risks, explained in the next section. Its main problem is that aperture definitions are *not* standardized. It is Ucamco's position that **the party that chooses to use Standard Gerber rather than Extended Gerber is responsible for any issues resulting from its use.**

It is now used in less than 2% of jobs. Extended Gerber is around for more than 20 years, applications are thoroughly debugged and it is supported by virtually all systems. If you are still using Standard Gerber you are putting your business and that of your clients and business partners at a useless risk, without any benefit. Don't be the last to lay it aside.

Thank you.

Karel Tavernier,
Managing Director,
Ucamco

Further information on Standard Gerber can be found below

Gent, June 2014, Karel Tavernier

Standard Gerber

1. What is Standard Gerber?

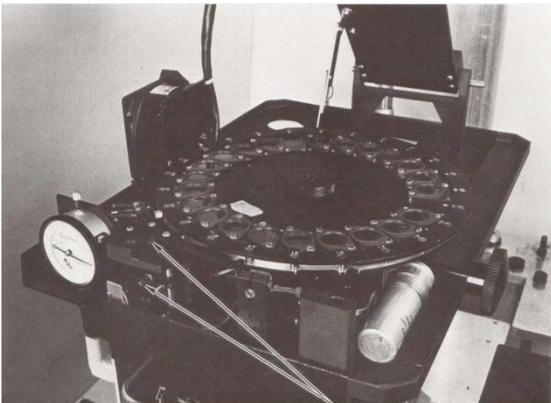
The current Gerber file format is also known as RS-274X or Extended Gerber. There is also a historic format called Standard Gerber or RS-274-D format. This differs from the current Gerber file format (RS-274X), in that it supports neither G36 or G37 codes, nor any parameter codes.

Consequently, Standard Gerber does not allow coordinate format and aperture shapes to be defined, and it lacks the imaging primitives needed to unequivocally transfer information from PCB CAD to CAM, making it incomplete as an image description format.

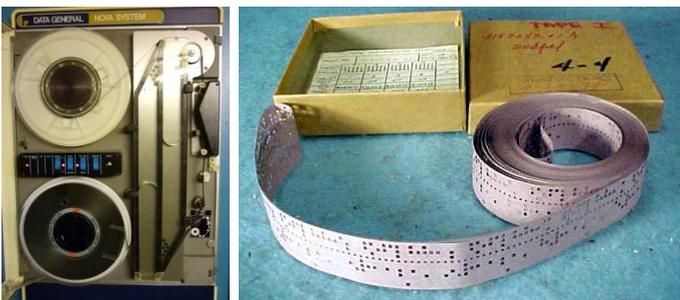
Despite its name, standard Gerber is not a standard image format – it is in fact a standard NC (Numerical Control) format, as explained below.

2. Origin and purpose of Standard Gerber

In the 1960s and 1970s, images were produced on lithographic film by a vector photoplotter; a precision optical NC machine. Images were produced by beaming light from the plotter's light source onto the film through an aperture on a wheel like that shown in the photograph below. This wheel was rotated to select the appropriate aperture, or it could be substituted by another aperture wheel if additional aperture sizes were needed.



The data for the exposure process was contained in a Standard Gerber file, which was typically recorded onto magnetic or paper tape (see pictures), which was in turn mounted onto the vector photoplotter by the operator.



The operator consulted the accompanying notes, typed the coordinate format on a machine console, mounted the appropriate aperture wheel, changed apertures if necessary, and started the plotter. The Standard Gerber file then drove the plotter through the required movements, controlled the aperture wheel and exposure light, and produced the desired image.

Standard Gerber was so well suited to this task that it became the industry standard.

That was decades ago. Vector photoplotters have not been used since, so Standard Gerber has lost its raison d'être. While it deserves a place of honor in the Computer History Museum, Standard Gerber has no place in the 21st century's electronics industry.

3. Standard Gerber is an NC format, not an image format

From the above, it is clear that Standard Gerber is an NC machine format, and not an image description format. It contains neither the coordinate format definition, so the meaning of coordinate data is undefined, nor aperture definitions, so the meaning of flashes and interpolations is undefined.

Thus if an image is to be defined using Standard Gerber, additional information is essential. This typically comes in the form of a so called "wheel file" consisting of notes in an informal text format, plus drawings that define the more complex apertures. The problem is that there is no standard for this extra information, creating enormous potential for error and misunderstandings. This puts the onus squarely on operators' shoulders to ensure that all of the information is assembled and checked on a workstation – manually and with the help of software tools – in order to be sure that all the necessary image data is present.

As if this were not enough, Standard Gerber also renders the informal description of complex apertures, SMD apertures and areas so difficult that designers give up, and opt instead to paint them. This in turn creates such chaos that there is a very real risk of losing valuable data in both CAD and CAM operations. Thus the CAM engineer must be extremely careful to recover, and piece together, the pads in the design.

So Standard Gerber, despite its name, is not an image definition standard, as it must be supported by a whole lot of extra non-standardized information in order to define an image. That's why Ucamco has defined the new Extended Gerber format. This, unlike its predecessor, *is* a standard, as it standardizes the additional data needed, puts it in the file header, and adds some sorely needed extensions.

4. Why one must use Extended Gerber

The following is sometimes said: "The only difference between Standard Gerber and Extended Gerber is that in Extended Gerber the wheel file is embedded in the file. This is no big deal as software was developed to extract data automatically from the wheel files."

We beg to differ:

- This difference *is* a big deal.
- It is *not* the only difference.

The problem with external wheel files

The use of informal wheel files is a big deal. While it is true that a lot of effort was spent on automating the input of accompanying notes, only a fraction of all data sets can in fact be read in automatically because they are often provided in a free format. While this freedom was perfectly adequate for the vector photoplotter operator of old, it flies in the face of standardization and automation, which consequently becomes a less reliable and higher maintenance process. And what happens if the notes arrive in another language – imagine, for example, automating the input of a wheel file in Japanese. Or of its supporting drawings, for which again, there are no format definitions. It becomes clear pretty quickly that it is not possible to fully and reliably automate the transfer of such informal data, so the operator must carefully check all results for errors. This is particularly important if we consider that a lack of standards can also mean lack of clarity about the intentions of the designer, as well as about where responsibility lies in case of errors.

Unfortunately, these are not theoretical issues, as the following real-life example illustrates. A Standard Gerber file came with the following wheel file (abbreviated):

```
// Units: Inches
// Format: DCode, Shape, Width, Height, Hole diameter, Rotation
D10, Round, 0.007000, 0.007000, 0, 0
...
D51, ObLong, 0.024000, 0.070000, 0.000000, 0
...
```

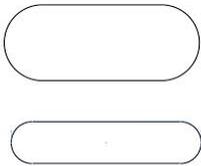
Even though this is one of the better, more explicit wheel files, the manufacturer must nevertheless interpret the term ObLong. A Google search brings up a helpful Wikipedia entry:

- *Oblong, a [rectangle](#) with length greater than its width (i.e. not a square). The word is sometimes used of any shape longer than it is wide.*

This is clear, it is a rectangle – the designer than hardly have meant that he would accept any shape. The manufacturer supplies a board with rectangle shape. And then the wrath of the designer descends upon him, because the designer did not intend a rectangle, but an obround:



To be fair to the designer, when you search Google Images the following comes up:



So the designer has a point. But so does the manufacturer. Recriminations fly, discussions about who pays for the scrap become acrimonious, and in the meantime deadlines are missed.

If the job had been sent in Extended Gerber, the apertures would have been defined unequivocally, according to a public standard, and none of this would have happened. In Extended Gerber, the formal, standardized aperture definitions are clear, so reading them in is straightforward, with no need to pore over the results for errors. And as there is a standard, it is clear what was intended, who is responsible in case of a mistake and what to do to avoid the issue in the future.

So yes, this difference is a big deal. It is the difference between using a published standard format and each individual using his own unspecified format. It is the difference between painstaking, error-prone manual work and inspection, and reliable, automatic data transfer.

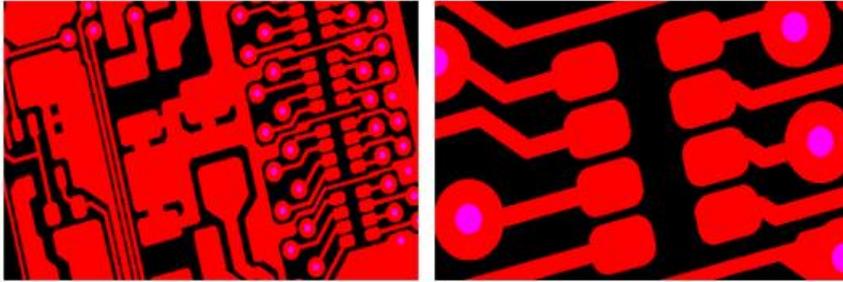
The problem with the limited features of Standard Gerber

Standard Gerber has very limited imaging features. It requires painting (or stroking) to describe planes and all but the simplest pads. Extended Gerber is a far richer format than Standard Gerber. It has all the features necessary for describing a PCB image efficiently. It has regions, positive/negative levels and powerful aperture macros. Planes and anti-pads can be described without painting, and pads are properly described as flashes, ensuring that no data is lost.

One of the most troublesome practices is one whereby pads, and sometimes other features, are “painted”. This is otherwise known as “stroking”, “paint-fill”, “stroke-fill” and “vector-fill”. This practice, born decades ago in the heyday of the Standard Gerber format, was then a necessary design practice. The vector photoplotters used at that time were driven by computers that were nowhere near as sophisticated as those we use today, so the shapes they could plot were very limited indeed. Thus designers would build pads using separate elements: curves or circles for the rounded corners, lines, or strokes, for the edges, and more strokes to fill the outline.

Things have of course moved on since then. In today's CAD systems the pad shape is described by a geometric primitive or its outline but instead of outputting this shape properly, too many system are still “improving” the output by filling the outline with numerous filling strokes. The result, to the naked eye, is a clear, nicely generated pad. The image is correct. (See fig. 1) But to a CAM system, it's chaos. Rather than reading a single pad, the system sees the digital reality: a hodgepodge of disconnected curves and straight lines, which it simply cannot recognise as anything particularly meaningful (see Fig. 2). Consider this happening not just once, but maybe hundreds of times, and in different sizes, across a densely populated PCB, and it quickly becomes obvious that painted features are every CAM engineer's nightmare.

Fig. 1: Painted features are visually clean...



Of course, the image is correct, and one may think there is therefore no big problem, just the nuisance that the files are bigger than they could be. However, the manufacturer needs more than the correct image. As we will demonstrate below, he needs to know the exact location and shape of all pads, areas and tracks.

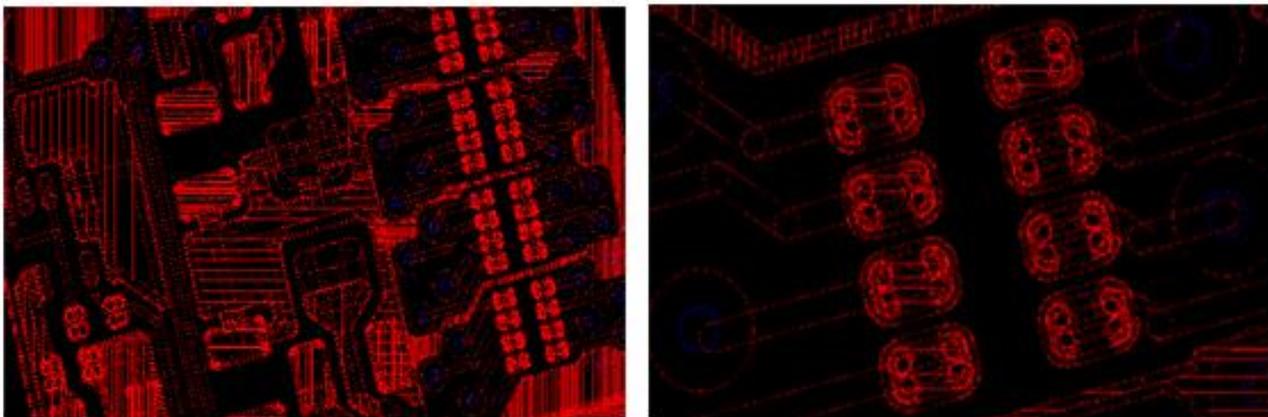
Of course, CAM systems have tools that aid the search for painted features and their conversion into a CAM-legible structure, but this is still incredibly labour-intensive. Everything has to be verified manually, slowing the CAM phase considerably, potentially creating delays in delivery, or worse, giving rise to PCB errors further down the line. It also adds unnecessary complexity to the Gerber file, and as each feature is made up of a large number of objects, the files are humongous and slow.

Pads

PCB manufacturers need to know exactly where every single SMD, component and via pad is on the board. This information is used for netlist creation and electrical test, to ensure that the solder mask is applied precisely where necessary, for via plugging, and to ensure that all clearances are within spec. And of course the assembler needs to know where pads are in order to apply paste.

Instant recognition and selection of all pads is also important where feature dimensions must be modified to comply with designers' specs, e.g., to compensate for the fabricator's specific etching process parameters.

Fig 2: ...but digitally chaotic



Because pad data is so important, any fabricator that receives images with painted pads must scan the whole image, *guess* where the pads are and typically replace all painted pads by proper flashed pads (see Fig. 3) prior to working with that file. This is a lengthy process that can give rise to errors in the product, because the manufacturer ends up having to gauge the designer's intentions rather than dealing with clear data.

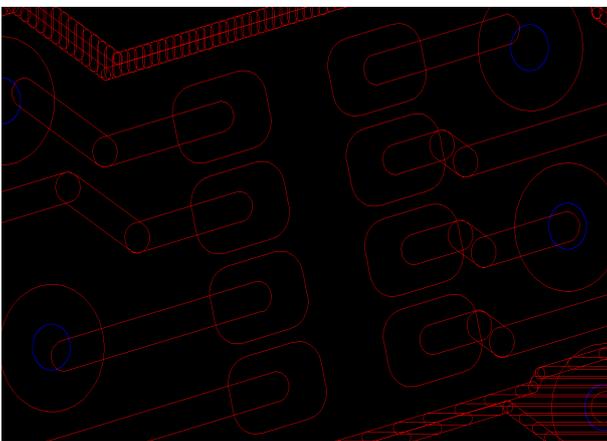
To obviate all of these problems, pads should be generated properly. This is easy with Extended Gerber which has a number of built-in pad shapes, and a powerful and unique macro language that makes it easy to create any shape, easier in fact than with any other PCB image format. There is no need to use painting. Shapes are defined using the %AD and %AM commands, and are then flashed wherever a pad should be – one flash, one pad.

Areas

Many PCB layers – e.g. power and ground layers – contain large copper areas, some of which can be extremely complex. The areas contain holes: clearances to allow non-connecting vias to go through the planes, also known as *anti-pads*. Here too, some designers will paint these areas, carefully filling in around the anti-pads. The problems thus created are similar to those mentioned above, but whereas a pad is relatively small, an area can take up a significant part of a layer's surface area. The least of the CAM engineer's resulting problems is that the incoming data file is huge. The real issue here is that the inside of an area will accommodate so many draws that it becomes extremely difficult for the CAM system to differentiate between draws that should be within the area, and draws that define neighbouring tracks. This creates all sorts of problems. For instance, when track width must be modified to compensate for etching parameters, it is crucially important to know which draws represent tracks and which represent painted pads or areas. The picture becomes even muddier when designers place embedded painted pads within painted areas.

Here too, in a lengthy, complicated, error-prone, manual operation, the manufacturer must replace the painted area by a properly constructed one, separating out the pads, tracks and areas from a messy jumble of draws.

Fig 3: Painted pads are replaced manually with properly flashed structures, in this case using flashmaker



This can all be avoided easily in Extended Gerber. The outlines can be directly stored in the Gerber file using the G36/G37 commands. This supports areas of any any shape, size and complexity using concise, clear language. Anti-pads and their positions are defined precisely and efficiently by using the %LP parameter to make a negative layer containing all the holes.

This is why there is a world of difference between Standard and Extended Gerber, and it's also why Extended Gerber is today's standard for CAD-to-CAM image data transfer.

Information about the Extended Gerber File Format, including the File Format Specification, can be found on the [Downloads](#) page of the Ucamco website.

© Copyright Ucamco NV, Gent, Belgium

All rights reserved. This material, information and instructions for use contained herein are the property of Ucamco. The material, information and instructions are provided on an AS IS basis without warranty of any kind. There are no warranties granted or extended by this document. Furthermore Ucamco does not warrant, guarantee or make any representations regarding the use, or the results of the use of the software or the information contained herein. Ucamco shall not be liable for any direct, indirect, consequential or incidental damages arising out of the use or inability to use the software or the information contained herein.

The information contained herein is subject to change without prior notice. Revisions may be issued from time to time to advise of such changes and/or additions.

No part of this document may be reproduced, stored in a data base or retrieval system, or published, in any form or in any way, electronically, mechanically, by print, photoprint, microfilm or any other means without prior written permission from Ucamco.

This document supersedes all previous dated versions.

Trademarks. All product names cited are trademarks or registered trademarks of their respective owners.