A proposal to include fabrication documentation in Gerber

Rev 2017.03

Draft for review only

Please send your comments to gerber@ucamco.com

The proposal was developed by Karel Tavernier

1 Preface

Since time immemorial, Gerber has been the standard for describing the 2D images – copper layers, solder mask, drills – in PCB fabrication data. Now, attributes added with Gerber X2 provide a standard for defining the layer structure – a 'which layer is which' file – and to ‘add intelligence’ to the image by providing information such as via and SMD identification, pin numbers and reference descriptions.

As Gerber X2 is compatible with X1, an X1 implementation can be extended to X2, so PCB images are completely covered by the X2 standard.

But PCB fabrication data is not just about describing images. It must also include general information about the final PCB; production parameters such as solder mask color, finish, overall thickness, materials and the assembly array definition. These parameters cannot be conveyed effectively by an image and yet they are essential for the quoting, planning, engineering, CAM and fabrication of the bare board as well as for determining the optimal assembly array.

As there is currently no standard for this sort of data, it is typically transferred informally in the form of drawings, PDF files and e-mail messages. None of these are designed for automatic workflow, so the information must necessarily be handled by people, which takes precious time and carries the risk of errors.

This document is a draft for an evolutionary extension of the Gerber format that aims to create a standard way of conveying production parameters that can be put through an automatic workflow. Far from being a full and final draft, it is intended as a discussion document to solicit input and reach a consensus in the user community on how to handle these parameters. The proposal will probably go through several revisions based on this input.

Please send your comments to gerber@ucamco.com

2 **Design considerations**

Rather than attempting to define a standard that will cover any situation imaginable, our aim is to focus on the technical characteristics of a single PCB, bearing in mind that software applications such as ERP and engineering systems, that need to 'know' product parameters, must be able to read them in unequivocally and without operator assistance.

We also aim to maintain Gerber's hallmark simplicity and human-readability – and to do that we need to keep things simple.

PCB data generated by any layout system typically focuses on the image data. It does not need to ‘know’ production parameters such as finishes or standards, and indeed, it rarely 'knows' them anyway, as these parameters are determined separately. It stands to reason: an application that defines an IPC class cannot reasonably be expected to handle Gerber images. The product parameters must therefore be stored in a separate dedicated file. Let us call this file the Gerber job file because it pertains to the whole PCB, not to a single layer.

The standard must be easy to adopt and to implement, it must be compatible with existing workflows, and it must be simple. It must also allow for partial implementations as some applications may only 'know' a few p production parameters. That's ok – better than nothing: the fabrication data user will be grateful if at least some of the data is standardized. Our job is to ensure that the operator can easily and quickly define which of the parameters can be standardized by his or her system.

The new approach must be compatible with existing workflows: legacy software that cannot handle the new standardized parameters must continue to function, albeit in the old manual way, and new software that does read them must still be capable of handling legacy archives.

The draft defines the technical PCB characteristics, *not* commercial parameters such as delivery times and address, pricing, quantities.

We intentionally do not specify default valude. The first reason is that partial implementations must be possibe; with defaults one would not know wether an absent parameter was simply not yet implemented or that the default value is intended. The second is deeper; we do not feel it is up to a mere data transfer format to specify default values for a job paremeter: e.g. it is surely not up to the Gerber format to decree what the default or normal copper thicknesses would be.

The draft does not cover assembly arrays. These are of course important but as they are not single PCBs their definition lies *outside* the scope of this specification. Layout systems are generally designed to generate a single PCB rather than generating arrays, and in the rare case that a layout system can also generate arrays, it typically does a poor job of it. This makes sense: assembly panels are typically not decided by the designer but by a separate company, the assembler. What we can do is to make the assembler's life easier by making sure that the data needed to make the assembly array – the XY size of the PCB, which is stored in the product parameters – is easily found and extracted.

The draft does not cover the full material stackup definition. For the majority of PCBs there is no need to specify the full stackup: it is sufficient to specify overall thickness and outer and inner copper thicknesses. For simplicity's sake we now define just these parameters. In due course we will extend the Gerber job specification to handle the material stackup. However, a full stackup for flex-rigid, special materials, impedance control can be quite complex. The challenge will be to define so that simple things remain simple, but complex structures can also defined unequivocally.

Production parameters must be stored in a simple dedicated file. This draft uses a standard, using a Gerber-like syntax. This has benefits – clear association with the current Gerber spec, familiar syntax. However, it technically not the most elegant or general. An alternative is to use XML or JSON. For complex structures as the full stackup or a BOM the Gerber syntax may become awkward. Of course, we could still use XML for the complex structures.

This draft currently contains only the most important parameters. Its purpose is not to offer a complete definition but to provide a framework to discuss this proposal. Once we agree on a concept we can complete the list.

3 Draft Specification

3.1 Job Attributes

Job attributes provide information about the PCB job of which the Gerber file containing them is one of the files. They are defined with the TJ command. They follow the syntax of the other attributes.

Example, to define the board size in X:

```
%TJ.B.Size.X,160*%
```

3.2 New File Function: Job file

The single PCB job attributes are concentrated in a separate file, the Gerber job file. The job file is identified with the new file function `JobInfo`. This indicates the file does not describe a single layer or entity in the PCB job, but manufacturing that belong to the job as a whole, such as the finish or overall thickness. The Gerber job file can only contain job attributes, not image data.

Example:

```
%TF.FileFunction,JobInfo*%
```

To find the job file quickly in an archive the file name must end with "_job.gbr". An example job file name is

```
Controller54382rev4_job.gbr
```

3.3 Single PCB job attributes

Question. For example, drill tool tolerance, is this a job parameter (definable externally) or an aperture parameter (can be different per tool). Or both – a general one that can be overruled for a specific tool. This would be an instance of successive refinements.

Attribute name	Usage, attribute values
Overall Board Parameters	
`.B.Owner`	Reference of the design owner as used by himself
`.B.ID`	Board identification or reference as used by the design owner
`.B.Size.X`	Size of the enclosing rectangle of the board outline. Tolerances are positive numbers. Decimals in the unit of the MO command
`.B.Size.Y`	
`.B.Size.Tol+`	

`.B.Size.Tol-`	
`.B.LayerNum`	Number of copper layers.
`.B.OverallThickness`	Overall board thickness. Tolerances are positive numbers. Tolerances are positive numbers. Decimals in the unit of the MO command. .Over stands for measured *over* e.g. mask, substrate etc.
`.B.Overall.Thickness.Tol+`	
`.B.Overall.Thickness.Tol-`	
`.B.Overall.Thickness.Over`	
`.B.Notes`	A free string
`.B.Copper.Thickness.Outer`	Final copper thicknesses. Decimals in the unit of the MO command. B.CuThickness.Holes is the plating thickness.
`.B.Copper.Thickness.Inner`	
`.B.Copper.Thickness.Holes`	
`.B.SolderMask.Present`	(Yes|No|Both)
`.B.Legend.Present`	
`.B.PeelableMask.Present`	
`.B.CarbonMask.Present`	
`.B.SolderMask.Color`	The value is either (Red|Green|Blue|White|Black|R<n>G<n>B<n>) <n> is an integer from 0 to 255 Can be over ruled by the values in the stackup details.
`.B.Legend.Color`	
`.B.Laminate`	FR4, Metal, ….### Ideally defined formally
`.B.IPC-600-Class`	(1|2|3|NA)
`.B.Standard`	-MIL, JSS, IPC6012B, PCA600,… We need formal pre-defined strings and a free string for other cases. TBD.
`.B.Finish`	HAL, OSP, Immersion gold, Chem NiAu… ### We need formal pre-defined strings and a free string for other cases. TBD.
`.B.ViaFilling`	(Open|CoverdTop|CoveredBot|CoveredBoth|FilledCopper|FilledResin|Other)
`.B.Logo`	String. ### Try formal definition.
`.B.ROHS`	(Yes|No)
`.B.UL`	(Yes|No)
`.B.ITAR`	(Yes|No).

<table>
<tr><td>.B.EdgePlating</td><td>(Yes|No)</td></tr>
<tr><td>.B.ImpedanceControlled</td><td>(Yes|No)</td></tr>
<tr><td>.B.HardGold</td><td>(Yes|No)</td></tr>
<tr><td>.B.EdgeConnector</td><td>(Yes|No)</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Intended design rules

These parameters state the main design rules used to create the layout. These parameters may seem superfluous as these values are reflected in the image. However, it can be convenient to know these values without analyzing the Gerber image files and knowing the design intent is useful in CAM when there are problems.

This probably must be split between inners and outers</td></tr>
<tr><td>.D.PadToPad</td><td></td></tr>
<tr><td>.D.PadToTrack</td><td></td></tr>
<tr><td>.D.PadToRegion</td><td>Regions are here used to represent copper pours.</td></tr>
<tr><td>.D.TrackToTrack</td><td></td></tr>
<tr><td>.D.TrackToRegion</td><td></td></tr>
<tr><td>.D.RegionToRegion</td><td></td></tr>
<tr><td>.D.MinLineWidth</td><td></td></tr>
<tr><td>.D.MinHoleSize</td><td></td></tr>
<tr><td colspan="2">Materials Stackup
(rigid boards only, for now)

The material stackup must either be complete or not present at all. Consequently, if any .S line is present, the absence of e.g. a bottom legend specifies that no bottom legend must be present, even if there is an image for it in layer structure.</td></tr>
<tr><td>.S.Legend.(Top|Bot).Color</td><td>The value is either
(Red|Green|Blue|White|Black|R<n>G<n>B<n>)
<n> is an integer from 0 to 255</td></tr>
<tr><td>.S.SolderMask.(Top|Bot).Color</td><td></td></tr>
<tr><td>.S.SolderMask.(Top|Bot).Thickness</td><td></td></tr>
<tr><td>.S.SolderMask.(Top|Bot).Constant</td><td>Dielectric constant</td></tr>
</table>

<table>
<tr><td><code>.S.PeelableMask.(Top|Bot).Color</code></td><td></td></tr>
<tr><td><code>.S.PeelableMask.(Top|Bot).Thickness</code></td><td></td></tr>
<tr><td><code>.S.Copper.<Ln>.Thickness</code></td><td>The integer n indicates the Cu layer number</td></tr>
<tr><td><code>.S.Dielectric.<n>.Constant</code></td><td>Dielectric constant. The integer n indicates the dielectric layer number; it is the same as the Cu layer number on top of the dielectric.</td></tr>
<tr><td><code>.S.Dielectric.<n>.Thickness</code></td><td>This is the net isolation distance between copper layers in the final PCB. This will be thinner than the base material, and thinner than the finished thickness between the substrate. <Ask if we can use Polar drawing.></td></tr>
<tr><td><code>.S.Notes</code></td><td>A free string.</td></tr>
<tr><td colspan="2">Layer Structure
These attributes give an overview of the data files present. It indicates their function in the layer structure. This function is already expressed in the .FileFunction file attribute inside the file. The benefit that this information is available without opening the individual Gerber file. It makes it easier to load the needed files in an application.
In fact, this has nothing to the with the board itself, but about how the data is structured. So the question is if this must be in the job file. On the other hand, it seems a useless complication to create a separate file for it; or not? Maybe more stuff will appear to fit in such a structure file. TBD.</td></tr>
<tr><td><code>.L.”<.FileFunction value>”</code></td><td>Path to the file</td></tr>
</table>

There are no default values. If a parameter is not present it is not defined in Gerber. The parameter is then defined in other ways; typically the board owner will accept reasonably fabricator defaults.

Of course CAD systems 'know' board size and layer count, both of which are required to define assembly panels. Ideally, CAD systems will output a simple job file with board size and layer count (and other parameters they might know), allowing assemblers to define their panels automatically, without having to process Gerber image files.

3.4 Minimal CAD Job File

Below is the minimal job file that CAD must include in the fabrication data. It contains the essential information needed to define the assembly panel. Other applications can then read and extend this file with more information.

```
G04 Mininal CAD Gerber job file*
%TF.FileFunction,JobInfo*%
%TF.Part,SinglePCB*%
%MOMM*%
%TJ.B.Size.X,160*%
%TJ.B.Size.Y,50.8*%
%TJ.B.LayerNum,6*%
M02*
```

3.5 Example: Basic Job File

This example contains the main overall board parameters.

```
G04 Gerber job file with the basic overall board parameters*
%TF.FileFunction,JobInfo*%
%TF.Part,SinglePCB*%
%TF.GenerationSoftware,Ucamco,UcamX,2016.12*%
%TF.CreationDate,2017-01-02T16:58:41+01:00*%
%TF.ProjectId,Sample file,6B69742D6465762D636F6C6466697265,1*%
%MOMM*%
G04 Single PCB fabrication instructions*
%TJ.B.ID,AZ2375EM*%
%TJ.B.Size.X,160*%
%TJ.B.Size.Y,50.8*%
%TJ.B.LayerNum,6*%
%TJ.B.Overall.Thickness,1.6*%
%TJ.B.Copper.Thickness.Outer,0.035*%
%TJ.B.Copper.Thickness.Inner,0.012*%
%TJ.B.SolderMask.Color,Green*%
%TJ.B.Legend.Color,White*%
%TJ.B.ROHS,Yes*%
M02*
```

3.6 Example: More Complete Job File

This example also contains the material stackup and layer strucuter

```
G04 Gerber job file with some stackup info and the layer structure.*
%TF.FileFunction,JobInfo*%
%TF.Part,SinglePCB*%
%TF.GenerationSoftware,Ucamco,UcamX,2016.12*%
%TF.CreationDate,2017-02-20T20:58:41+01:00*%
%TF.ProjectId,Sample file,6B69742D6465762D636F6C6466697265,1*%
```

```
%MOMM*%
G04 Single PCB fabrication instructions*
G04*
G04 Overall board parameters*
G04 ------------------------*
%TJ.B.Owner,Galactic Corporation*%
%TJ.B.ID,AZ2375EM*%
%TJ.B.Size.X,160*%
%TJ.B.Size.Y,50.8*%
%TJ.B.LayerNum,2*%
%TJ.B.IPC-600-Class,2*%
%TJ.B.Finish,Immersion Gold*%
%TJ.B.Overall.Thickness,1.6*%
G04*
G04 Material Stackup*
G04 ----------------*
%TJ.S.Legend.Top.Color,White*%
%TJ.S.SolderMask.Top.Thickness,0.01016*%
%TJ.S.SolderMask.Top.Color,Green*%
%TJ.S.Copper.L1.Thickness,0.03556*%
%TJ.S.Dielectric.1.Thickness, 1.48336mm*%
%TJ.S.Copper.L2.Thickness,0.03556*%
%TJ.S.SolderMask.Bot.Thickness,0.01016*%
%TJ.S.SolderMask.Bot.Color,Green*%
G04*
G04 Layer Structure*
G04----------------*
%TJ.L.”Paste,Top”,          AZ2375EM_Top_SMT_Paste.gbr*%
%TJ.L.”Legend,Top”,         AZ2375EM_Top_Silk.gbr*%
%TJ.L.”SolderMask,Top”,     AZ2375EM_Top_Solder.gbr*%
%TJ.L.”Copper,L1,Top”,      AZ2375EM_Top_Copper.gbr*%
%TJ.L.”Copper,L2,Bot”,      AZ2375EM_Bot_Copper.gbr*%
%TJ.L.”SolderMask,Bot”,     AZ2375EM_Bot_Solder.gbr*%
%TJ.L.”Plated,1,2,PTH”,     AZ2375EM_Drill_Top_Bot_Plated.gbr*%
%TJ.L.”NonPlated,1,2,NPTH”,AZ2375EM_Drill_Top_Bot_Unplated.gbr*%
%TJ.L.”AssemblyDrawing,Top”,AZ2375EM_Top_Assembly.gbr*%
M02*
```

4 Revisions

Rev 2017.01

Initial version

Rev 2017.03

Error corrections and improvements suggested by Paul Wells-Edwards, Remco Poelstra, Rik Breemeersch. Added parameters suggested by Ken Caluwaerts.

Made a more complete list of board parameters, added intended design rules, material stackup and layer structure definition.

5 **Copyright**