Gerber X3:

Including assembly information in Gerber data

Revision 2020.03

This specification was developed by Karel Tavernier, in extensive discussion with Wim De Greve, Jean-Pierre Charras, Thiadmer Riemersma, Rafal Powierski and Bruce McKibben.

Preface

The purpose of this Gerber standard is to transfer the component information held in CAD which is needed in manufacturing to:

- visualize the component placement to check for errors and set up the assembly,
- generate the manufacturing tools, such as the paste stencils and the pick and place machine programs,
- assist in the procurement of the components.

The bare board design is transferred as Gerber files, while component information is currently transferred as separate, non-standardized drawings, pick & place and BOM files. This works fine for the bare board, but not for the components, neither location nor identification. The standard will include component data in the Gerber data. Component location is geometrical data: the centroid, outline, fiducial locations and the footprints; geometric data fits naturally in a Gerber image file. Combining bare board and component data in Gerber files allows a holistic review of the final board.

The intended workflow is that the CAD Gerber output data is read into an assembly CAM system, which analyzes the incoming data, allows for visual inspection and generates the outputs tuned to the specific assembly equipment.

The scope of this specification is intentionally limited to *technical* product information of a single physical PCB. It is not mixed with commercial information such as order quantities - the same physical PCB can be ordered or sold under different commercial terms, and this must not affect the product model.

It is natural to put the components in two new Gerber files: the top and bottom component file. By placing the component data in separate files, full compatibility is maintained: if one does not like the new files, simply ignore them. The new standard is therefore compatible with existing workflows. Legacy software handles the CAD data with the new assembly information without change, of course also without benefiting from it. Great care is taken to minimize the development effort by keeping the existing fabrication outputs in place and sticking to an existing syntax. Benefits:

  - Compatible with installed base
  - Minimal implementation work.
  - Any decent legacy viewer displays the new component layers.

Please send your comments to gerber@ucamco.com

1 Component Layers

The new top and bottom component layers specify component information needed for the assembly process. A set of attributes identify each component's location, orientation, identification and its properties such as manufacturer part number.

1.1 Identifying the new component layer

The component layer is identified by a.FileFunction value:

.FileFunction value	Usage
`Component,L<p>,(Top|Bot)`	A component layer. L<p> The integer p is the copper layer number to which the components described in this file are attached. (*Top*|*Bot*) indicates if the components are on top, upwards, or on the bottom, downward, of the layer to which they are attached. This syntax caters for embedded components. For jobs without embedded components there is an intentional redundancy.

An example, the component layers of a four-layer board.

```
    %TF.FileFunction,Component,L1,Top*%

    %TF.FileFunction,Component,L4,Bot*%
```

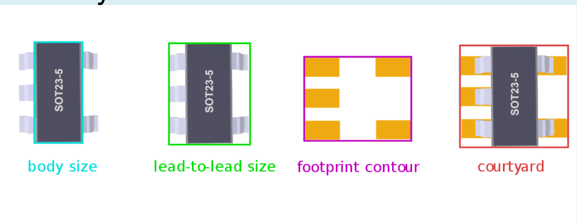
The coordinate system of the component layers as for all layers: a right-handed system when the PCB is viewed from the top.

1.2 The overall component properties

These are identified by .AperFunction values and object attributes. These can only be used in the component layers.

.AperFunction value	Usage

`ComponentMain`	This aperture is flashed at the centroid of a component. The flash carries the object attributes with the main characteristics of the component.
	The following aperture must be used:
	`%ADDnnC,0.300*%    (mm)`
	`%ADDnnC,0.012*%    (in)`
	nn is an aperture number, an integer ≥ 10.

`ComponentOutline,<type>`
`<type>=(Body|Lead2Lead|Footprint|Courtyard)`
	This attribute is used to draw the outline of the component. An outline is a sequence of connected draws and arcs. They are said to connect only if they are defined consecutively, with the second starting where the first one ends. Thus, the order in which they are defined is significant. A contour is closed: the end point of the last draw/arc must coincide with the start point of the first. Outlines cannot self-intersect.
	Four different types of outlines are defined. See drawing, courtesy Thiadmer Riemersma:
	 SOT23-5
	body size
	lead-to-lead size
	footprint contour
	courtyard
	Outlines of different types on the same component are allowed.
	The following aperture must be used:
	`%ADDnnC,0.100*%    (mm)`
	`%ADDnnC,0.004*%    (in)`

Object attributes	Usage

`.C,<refdes>`
`<refdes>=<field>`
	This is an already existing attribute. See section 5.6.15 in the main specification for more information. It identifies the component reference descriptor.

`.CRot,<decimal>`	The rotation angle of the component. The rotation angle is consistent with the one for graphics objects. Positive rotation is counterclockwise viewed from the top side, even if the component is on the bottom side. The zero-rotation orientation of a top side component as in IPC-7351. The base orientation of a bottom side component is the one on the top side, mirrored around the X axis.
`.CMfr,<field>`	Manufacturer
`.CMPN,<field>`	Manufacturer part number
`.CVal,<field>`	Value, e.g. 220nF
`.CMnt,(TH|SMD|Fiducial|Other)`	Mount type
`.CFtp,<field>`	Footprint name
`.CPgN,<field>`	Package name
`.CPgD,<field>`	Package description
`.CHgt,<decimal>`	Height, in the unit of the file.
`.CLbN,<field>`	Name of the component library from where the component information originates
`.CLbD,<field>`	Description of the component library from where the component information originates
`.CSup,<SN>,<SPN>,{<SN>,<SPN>}`	<SN> is a field with the supplier name. <SPN> is a field with a supplier part name

All ComponentMain and ComponentOutline objects must have a .C object attribute. The <RefDes> value for each component should be unique, since it links the ComponentOutline objects to their ComponentMain object. The other .Cxxx attributes are only valid for ComponentMain objects. If attached to other objects, they are ignored.

1.3 The pin locations

The pin locations are included in the component layer to unequivocally specify location and rotation of the components, using the new aperture function ComponentPin and a re-use of the .P attribute which already exists for pads on the top and bottom copper layers. The .P attribute is mandatory for each ComponentPin flash. Any .Cxxx attributes attached to a pin are ignored.

.AperFunction value	Usage
ComponentPin	The coordinates in the flash command (D03) indicate the location of the component pins (leads). The .P object attribute must be attached to each flash to identify the reference descriptor and pin. For the key pin, typically pin "1" or "A1", the following diamond shape aperture must be used: %ADDnnP,0.360X4X0.0*% (mm) %ADDnnP,0.017X4X0.0*% (in) The key pin is then visible in the image. For all other pins the following zero size aperture must be used: %ADDnnC,0*%...(both mm and in) These pins are not visible which avoids cluttering the image.
Object attribute	**Usage**
.P,<refdes>,<number>[,<function>] <refdes>=<field> <number>=<field> <function>=<field>	This is an already existing attribute. See section 5.6.14 in the main specification for more information. It identifies the pin reference descriptor, pin number or name, and optionally the pin function. It is strongly recommended to include *both* pin number and pin function. The pin function is less prone to error than the pin number, and the redundancy provides extra security This is especially important for diodes and transistors.

It is optional to add local fiducials in the component layer. If they are added a zero-size aperture must be used, with .AperFunction value the existing value FiducialPad,Local, and object attribute .C to identify the component.

1.4 The character set

The basic Gerber character set, used in command syntax, is limited to the 7-bit ASCII codes 32 to 126. This character set is too restrictive for user-defined strings and fields, which are defined in other applications with wider character sets, can be in a local language etc.

A fully compatible way to include such out-of-range characters in user-defined strings is with the escape sequence \uXXXX where XXXX is the four-digit hexadecimal number representing the Unicode character code. For example, \u00a9 represents the copyright symbol '©'. When a character code is less than four digits it must be padded with leading zeroes. The Unicode escape sequence syntax therefore conforms to \\u[a-fA-F0-9]{4}

Support for out-of-range characters becomes more and more important as the functionality of Gerber grows. To maintain human readability and streamline software applications it is now allowed to use the full Unicode character set with UTF-8 encoding, *in user-defined strings and fields only*. Please note that poorly implemented legacy applications might struggle with this.

The reserved characters asterisk *, percent %, comma , and backslash \ must always be escaped with a sequence \uXXXX.

To enhance human readability, it is recommended to enclose complicated fields and strings between quotes.

The example below contains UTF-8 encoded characters and is between quotes for clarity.

```
    %TO.CLbN,"Condensateurs électrolytiques"*%
```

In the example below the comma , in 4,7uF is escaped. (2C is the Unicode for comma.)

```
    %TO.CVal,"4\u002C7uF"*%
```

⚠ **Warning:** Be careful always to escape * and %. These are the terminator characters in the Gerber syntax since times immemorial, and their presence in strings will throw input parsers completely off track.

2 Annotated Example File

This is an example of a top component layer. It assumes a general knowledge of the Gerber format as only the new, component related, commands are annotated.

Commands	Annotation
%TF.GenerationSoftware,KiCad,Pcbnew,(5.99.0-190-g0fd48dd4f-dirty)*%	
%TF.CreationDate,2019-10-02T18:52:55+02:00*%	
%TF.ProjectId,kit-dev-coldfire-xilinx_5213,6b69742d-6465-4762-9d63-6f6c64666972,2*%	
%TF.SameCoordinates,PX3e22018PY8d89728*%	
%TF.FileFunction,Component,L1,Top*%	**This file is the top component layer**
%TF.FilePolarity,Positive*%	
%FSLAX46Y46*%	
%MOMM*%	
%LPD*%	
G04 Aperture begin list*	
%TA.AperFunction,ComponentMain*%	
%ADD10C,0.3*%	**The aperture for the flash with the main component information**
%TA.AperFunction,ComponentOutline,Courtyard*%	
%ADD11C,0.1*%	**The aperture to draw the outline**
%TA.AperFunction,ComponentPin*%	
%ADD12P,0.36X4X0.0*%	**The aperture to flash the pin 1 location**
%ADD13C,0*%	**The aperture for the other pin locations**
%TD*%	
G04 Aperture end list*	
G04 Begin component info*	
D10*	**Select main component aperture**
%TO.C,R301*%	**Attach reference descriptor R301**
%TO.CFtp,R_0805_2012Metric*%	**Attach footprint**
%TO.CVal,4K7*%	**Attach value**
%TO.CMnt,SMD*%	**Attach mount type**
%TO.CRot,-90*%	**Attach rotation**

Commands	Annotation
X218250000Y-73000000D03*	Flash at reference point
D11*	Select outline aperture
X219250000Y-71310000D02*	Draw outline
X217250000Y-71310000D01*	Draw outline
X217250000Y-74690000D01*	Draw outline
X219250000Y-74690000D01*	Draw outline
X219250000Y-71310000D01*	Draw outline
D12*	Select key pin aperture
%TO.P,R301,1*%	Attach ref. desc and pin number
X218250000Y-72045000D03*	Flash at key pin location
D13*	Select subsequent pin aperture
%TO.P,R301,2*%	Attach pin 2 ident
X218250000Y-73955000D03*	Flash at key pin location
%TD*%	Clear attributes before next component
G04 Next component*	
…	

3 **Revisions**

Rev 2020.03

This is the first official version of the Gerber specification to include component information in Gerber fabrication data, known as X3.

Karel Tavernier developed a prototype specification which was circulated privately with Wim De Greve, Jean-Pierre Charras, Thiadmer Riemersma, Bruce McKibben and Rafal Powierski in December 2018. In intense discussion among this group the draft went through five revisions until a first public draft was published in October 2019, calling for input from the user community. The review process was closed in February 2020 with the publication of this specification.

4 Copyright

The information contained herein is subject to change without prior notice. Revisions may be issued from time to time. This document supersedes all previous versions. Users of the Gerber Format®, especially software developers, must consult www.ucamco.com to determine whether any changes have been made.

Ucamco developed the Gerber Format®. The Gerber Format®, this document and all intellectual property contained in it are solely owned by Ucamco. Gerber Format® is a Ucamco registered trademark. By publishing this document Ucamco does not grant a license to the intellectual property contained in it. Ucamco encourages users to apply for a license to develop Gerber Format® based software.

By using this document, developing software interfaces based on this format or using the name Gerber Format®, users agree not to (i) rename the Gerber Format®; (ii) associate the Gerber Format® with data that does not conform to the Gerber file format specification; (iii) develop derivative versions, modifications or extensions without prior written approval by Ucamco; (iv) make alternative interpretations of the data; (v) communicate that the Gerber Format® is not owned by Ucamco or owned by anyone other than Ucamco. Developers of software interfaces based on this format specification commit to make all reasonable efforts to comply with the latest specification.

The material, information and instructions are provided AS IS without warranty of any kind. There are no warranties granted or extended by this document. Ucamco does not warrant, guarantee or make any representations regarding the use, or the results of the use of 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 information contained herein. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Ucamco. All product names cited are trademarks or registered trademarks of their respective owners.