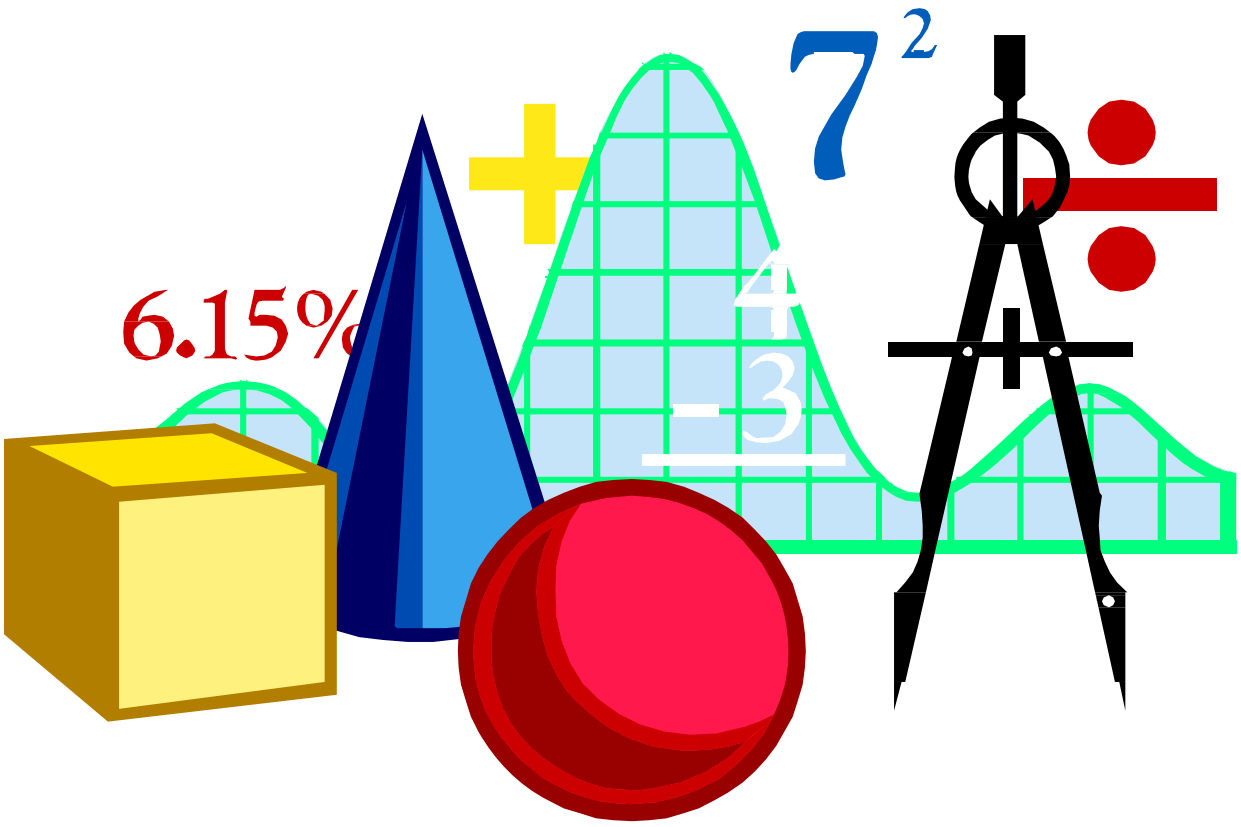
Ucamco
Former Barco ETS

The Gerber Format Specification

RS-274X or Extended Gerber



Preface

Gerber files are the de facto standard for PCB image data transfer. Virtually every PCB design system outputs RS-274X (also known as Extended Gerber or X-Data) and every PCB front-end engineering system inputs it. Implementations are thoroughly field-tested and debugged. Its widespread availability allows PCB professionals to exchange image drill and route securely and efficiently.

The RS-274X format is simple, compact and unequivocal. It is easy to interpret. It describes an image with very high precision. It is complete: one single file describes an image. It is portable and easy to debug by its use of printable 7-bit ASCII characters.

A well-constructed RS-274X file precisely defines the PCB image data and the functions of the different image elements. Unfortunately, poorly constructed or simply erroneous RS-274X files also circulate, sometimes leading to unjustified criticism of the format itself.

Errors may be due to a misunderstanding of the format. With more than 25 years experience in CAM software we at Ucamco know which areas are most often misunderstood. This revision of the RS-274X specification explains these areas more clearly.

Other files are not invalid but poorly constructed. Especially troublesome are painted or stroked pads and copper planes. Poorly constructed files take longer to process, require more manual work and increase the risk of errors. This revision of the RS-274X specification recommends constructions to make RS-274X files safer and more efficient, and hence fabrication more reliable, faster and cheaper.

A few words must be said about RS-274-D or Standard Gerber. This format was developed to drive NC machine tools and was used for Gerber vector plotters in the 1960s and 1970s. It is not an image description format. It is amazing that it is still used. It is like using teletype paper tape to transfer text documents. We call on industry experts and professional organizations to discourage the use of the obsolete RS-274-D format.

Although other data transfer formats have come into the market, they have not displaced RS-274X. The reason is simple. More than 90% of the problems in data transfer are due not to limitations in the RS-274X format but to poor practices and, worse, the use of RS-274-D. To quote a PCB manufacturer: “If we would only receive proper RS-274X files, it would be a perfect world.” The new formats are more complex and less transparent to the user, and new implementations inevitably have bugs. Using the common poor practices in the newer and more complex formats makes matters worse, not better. Fabricators have not adopted the new formats. RS-274X remains the standard.

The emergence of RS-274X as a standard for image exchange is the result of effort by many individuals who developed outstanding software for RS-274X files. Without their dedication the widespread acceptance of RS-274X could not have been achieved. Ucamco thanks these dedicated individuals.

Karel Tavernier

Managing Director,
Ucamco

January 2012
Revision H

Correspondence regarding this publication can be sent to:

Ucamco NV
Bijenstraat 19,
B-9051 Gent,
Belgium

For more information:

Our web site: http://www.ucamco.com

E-mail: info@ucamco.com

About Ucamco

Ucamco (former Barco ETS) is a market leader in PCB CAM software and laser photoplotting systems. We have more than 25 years continuous experience developing and supporting leading-edge front-end tooling solutions for the global PCB industry. We help fabricators world-wide raise yields, increase factory productivity, and cut enterprise risks and costs.

Today we have more than 1000 laser photoplotters and 5000 CAM systems installed around the world with local support in every major market. Our customers include the leading PCB fabricators across the global spectrum. Many of them have been with us for more than 20 years.

Key to this success has been our uncompromising pursuit of engineering excellence in all our products. For 25 years our product goals have been best-in-class performance, long-term reliability, and continuous development to keep each user at the cutting-edge of his chosen technology.

Ucamco
Former Barco ETS

Table of Contents

Preface ..... **iii**

Table of Contents ..... **vi**

1 Introduction **9**

1.1 Who should use this Specification? 9
1.2 About This Document 9
1.3 History of the Gerber Format 9

2 Syntax **10**

2.1 Character Set 10
2.2 Data Blocks 10
2.3 Data Types 10
2.4 Coordinate Data 10
2.5 Function Codes 11
2.6 Parameters 11

3 Image Generation **13**

3.1 Information Layers 13
3.2 Generating the Information Layer Image 13

4 Parameters **14**

4.1 AD - Aperture Definition 15
4.1.1 Standard Apertures 15
4.1.2 Special Apertures 15
4.1.3 Syntax Rules 15
4.1.4 Data Block Format 15
4.1.5 Aperture Definition with Standard Apertures 16
4.1.6 Examples 17
4.2 AM - Aperture Macro 18
4.2.1 Contents 18
4.2.2 Syntax Rules 18
4.2.3 Data Block Format 19
4.2.4 Primitives 20
4.2.5 Examples 22

4.2.6 SM – Symbol Mirror ... 25
4.3 Directive parameters ... 26
4.3.1 AS – Axis Select ... 26
4.3.2 FS – Format Specification ... 27
4.3.3 MI – Mirror Image ... 29
4.3.4 MO – Mode ... 30
4.3.5 OF - Offset ... 31
4.3.6 SF – Scale Factor ... 32
4.4 Image parameters ... 33
4.4.1 IP – Image Polarity ... 33
4.4.2 IR – Image Rotation ... 34
4.5 Layer-specific parameters ... 35
4.5.1 KO – Knock Out ... 35
4.5.2 LN – Layer Name ... 37
4.5.3 SR – Step and Repeat ... 39

5 Function Codes ... 40

5.1 Overview ... 40
5.2 Linear Interpolation (G01 (G1)) ... 42
5.2.1 Data Block Format ... 42
5.3 Circular Interpolation (G02 (G2), G03 (G3), G74, G75) ... 43
5.3.1 Single Quadrant Mode ... 44
5.3.2 Multi Quadrant Mode ... 45
5.3.3 Example ... 46
5.4 Apertures allowed for drawing ... 47
5.4.1 Circle Aperture ... 47
5.4.2 Rectangle or Square Aperture ... 47
5.5 Outline Fill (G36, G37) ... 48
5.5.1 Example 1: Simple outline ... 48
5.5.2 Example 2: Cut In ... 49
5.5.3 Example 3: More complex cut-in ... 51

6 RS-274X Best Practices ... 53

7 Common Syntax Errors ... 55

8 Glossary ... 56

9 Sample Files ... 58

9.1.1 Example 1 ... 58
9.1.2 Example 2 ... 59

1 Introduction

1.1 Who should use this Specification?

This Specification is intended for:

- PCB designers preparing RS-274X files
- PCB fabricators creating or using RS-274X files
- Developers of software applications using RS-274X files

1.2 About This Document

The following conventions are used in this document:

Note:	Provides essential extra information.
Tip:	Provides useful extra information.
Example:	Contains examples of file syntax, commands, settings, etc.
Warning:	Contains an important warning.

1.3 History of the Gerber Format

Gerber Format derives its name from the former Gerber Systems Corporation, a leading supplier of photoplotters.

Originally, Gerber used a subset of the Standard RS-274-D format as input format for its photoplotters. This subset became known as Standard Gerber. It is not an image description format, but a format to drive mechanical machine tools, which photo-plotters originally were, where apertures shapes where physical apertures in a so-called aperture wheel.

In subsequent years, Gerber extended the input format for its range of PCB devices and it actually became a family of formats. The Gerber formats developed into a capable image description format. In 1998 the formats were pulled together and standardized by the publication of the first version of this document. Since then, it has become the de-facto standard for PCB image data.

2 Syntax

2.1 Character Set

An RS-274X file is expressed in printable 7-bit ASCII characters only.

2.2 Data Blocks

The file is composed of data blocks. Each data block ends with the mandatory end-of-block character asterisk "*****". Each data block may contain one or more parameters, codes or coordinates.

For example:
X0Y0D02*
X50000Y0DO1*

For readability it is recommended to:

- Begin independent data blocks on a new line.
- Peep each data block one line.

2.3 Data Types

There are the following data types.

1 Coordinate Data
2 Function Codes
3 Parameters

2.4 Coordinate Data

Coordinate data to define points in the plane. (They were called addresses in the obsolete RS-274-D terminology). Coordinate data can be:

- X, Y data to define the X, Y coordinates of a point
- I,J data to define an offset in the X, Y direction

The FS Parameter (Format Specification) specifies how the digits must be interpreted.

Right handed orthonormal coordinates are used.

Coordinates are modal. If an X is omitted the last X coordinate is used; if the first X in the current layer is omitted, there is no last X coordinate and zero is used as default. Similar for Y coordinates.


Note: Gerber readers sometimes handle the zero default wrongly. For clarity and robustness it is recommend not to count on the zero default, but always to specify the first coordinate explicitly, also if it is zero.

Offsets are *not* modal. If I or J is omitted the default zero is used.

Examples:

```
X200Y200*           Point  (+200, +200)
Y-300*              Point  (+200, -300)
I300J100*           Offset (+300, +100)
X200Y200I50J50*     Point  (+200, +200) and Offset (+50, +50)
X+100I-50*          Point  (+100, +200) and Offset (-50, 0)
```

2.5 Function Codes

Function codes describe how coordinate data associated with them should be interpreted, e.g. draw a line or draw a circle. (Most, but not all, of these codes are inherited from the now obsolete RS-274-D format. They were called words or codes.)

Example:

```
G74*
```

Each code applies to coordinate data located in the same data block as the code and to all subsequent coordinate data until another code of the same type is encountered, or until a new layer is generated. This continuing action is referred to as *modal*.

For example, G02 specifies clockwise circular interpolation. All coordinate data following it will be interpreted as clockwise circular interpolation until another interpolation code is encountered, or until a new layer is generated.

Codes are described in detail later.

2.6 Parameters

Parameters define characteristics applying to an entire image or to a single layer. They are used to interpret the other data types. (Originally, they were called Mass Parameters.)

Parameters operating on the entire image are usually placed at the beginning of the file. Parameters generating a new layer are place at the appropriate place in the file

Parameters consist of two alphabetic characters followed by one or more optional modifiers.

Parameters are delimited by the parameter delimiter “%”. Each parameters is contained in a data block that itself must be ended by an *. The parameter delimiter must immediately follow the end-of-block without intervening spaces.

For example:

```
%FSLAX23Y23*%
```

Parameters may be entered single or grouped between delimiters, up to a maximum of 4096 characters between delimiters.

For example:

```
%SFA1.0B1.0*ASAXBY*%
```

Line breaks are permitted between parameters to improve readability.

For example:

```
%SFA1.0B1.0*
```

ASAXBY*%

For readability and simplicity it is recommended to have one parameter per line.

Use an explicit decimal point with all numerical values associated with a parameter. If the decimal point is omitted, an integer value is assumed.

The syntax for parameters is:

%Parameter code<required modifiers>[optional modifiers]*%

Syntax	Comments
Parameter code	2-character code (AD, AM, FS, etc…)
<required modifiers>	Must be entered to complete the definition
[optional modifiers]	Entry depends on the required modifiers

3 Image Generation

3.1 Information Layers

An information layer is a section of the image consisting of one or more consecutive data blocks. The data blocks of an information layer create the layer image. A polarity (dark or clear) may be assigned to a layer.


Note: An information layer in RS-274X is not the same as a PCB layer.

Information layer images are superposed in the order they appear in the file. A dark layer image marks or *darkens* the image area. A clear layer image clears (unmarks, rubs, erases) its image in *all* the lower layers. In other words, after superposing a clear layer, all objects in the clear layer is white in the image area, whatever objects were there before. Subsequent dark layers may again mark the cleared area. The order of exposing the layers can affect the final image if clear layers are present.

In addition, an individual layer may be repeated and/or rotated individually.

3.2 Generating the Information Layer Image

The RS-274X file defines a *graphics state* at each point in the file; the graphics state is modified by Function Codes and Parameters. The RS-274X file also defines a *stream of* Coordinate Data.

The graphics state operating on the stream of Coordinate Data creates a stream of *graphics objects*, such as a flashes or draws, which are added to the current layer.

The graphics state is reset at the start of the file and at the start of each new layer.

An element of the graphic state is the implicitly defined *current point*. The current point is set to the origin or (0, 0) at the start of the file and at the start of each layer. It is moved by the action of the graphics state. The graphic state typically creates a graphic object, e.g. a draw, starting at the current point, and ending at a point defined by the Coordinate Data; the current point is then set to this endpoint.

Other elements of the graphics state are the *current aperture, interpolation mode,* etc.

4 Parameters

Parameter		Function	Comments	Default
Mandatory	**Optional**			
	AS	*Axis select*	It is recommended to use these parameters only once, at the beginning of the file. These parameters do not generate a new layer.	A=X, B=Y
FS		**Format Statement**		
	MI	*Mirror Image*		A0B0
	MO	**Mode (inch or millimeter units)**		
	OF	*Offset*		A=0.0, B=0.0
	SF	*Scale Factor*		A=1.0, B=1.0
	IP	*Image Polarity*	Use only once at the beginning of the file.	Positive
	IR	*Image Rotation*		0
	AD	**Aperture Definition**	May be used singly or may be layer-specific. Enter these parameters at the beginning of the file or layer.	
	AM	**Aperture Macro**		
	LN	**Layer Name**		
	LP	**Layer Polarity**		Positive
	KO	*Knockout*		Off
	SR	**Step and Repeat**		A=1, B=1
	SM	*Symbol Mirror*		A=0, B=0

Parameter Overview.
Parameters in **bold** are current, those in *italic* are deprecated.

Parameters are usually placed at the beginning of the file in the order shown above. Layer-specific parameters are in the file at the appropriate location.

Gerber writers cannot use the deprecated parameters anymore. As they can be present in legacy files, readers can either implement them or not. If they are present, their meaning is specified below.

4.1 AD - Aperture Definition

The AD parameter defines the shape and size of aperture D-codes used in the RS-274X file. The AD parameter must precede the first use of the associated aperture D-code. A definition remains in effect until redefined.

Two kinds of apertures may be used: *standard apertures* and *special apertures*.

4.1.1 Standard Apertures

The AD parameter identifies standard apertures by D-code number and defines its shape and size.

Standard apertures may be solid or open (with a centered hole). Enter one parameter to define a round hole by its center, two parameters to define rectangular hole by width and height and omit both parameters for a solid aperture. Holes in apertures are transparent. A hole must fully fit within the outside aperture boundary.

4.1.2 Special Apertures

The AD parameter is also used to assign a D-code to a file containing an aperture definition created with the AM (Aperture Macro) parameter. See the AM parameter description for further information on aperture macros.

4.1.3 Syntax Rules

- Begin and end each parameter block with the parameter delimiter %.
- Within the AD parameter block, separate each modifier by an X.
- Dimensions must be positive.
- The range of D-code is from 10 to 999.

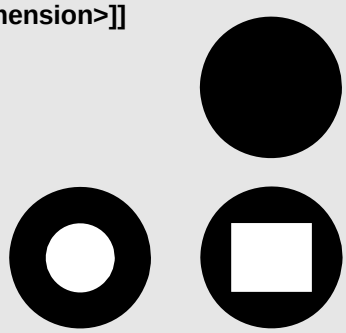
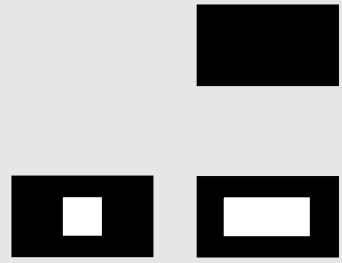
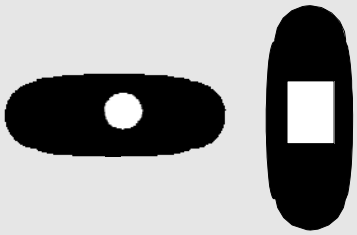
4.1.4 Data Block Format

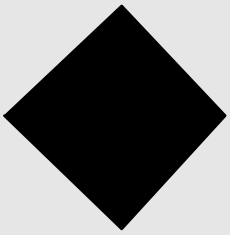
The syntax for the AD parameter is:

%ADD<D-code number><aperture type>,<modifier>[X<modifier>]*%

Syntax	Comments
ADD	**AD** for Aperture Definition and **D** for D-code
<D-code number>	The D-code number being defined (10 - 999)
<aperture type>, <modifier>[X<modifier>]	The Aperture Type has two possible formats: A standard aperture (C, R, O or P) with modifiers separated by X (all dimensions are positive) or an aperture macro name previously defined by the AM parameter.  Notes: Be sure to use the units (inches or millimeters) specified by the MO parameter for all modifiers

4.1.5 Aperture Definition with Standard Apertures

Syntax	Comments
C,<outside diameter>[X<X-axis hole dimension >[X<Y-axis hole dimension>]] 	**Circle**. To define a solid aperture, enter only the diameter. To define a hole, enter one dimension for a round hole, two for a rectangular hole. The hole must fit within the aperture. Both aperture and hole will be centered. For example, **%ADD10C,.05X0.025*%** D-code 10 is a .05 circle with a .025 round hole in the center.
R,<X-axis dimension>X<Y-axis dimension>[X<X-axis hole dimension>X<Y-axis hole dimension>] 	**Rectangle or square**. May be solid or open. If the X axis dimension equals the Y dimension, the aperture will be square. To define a solid aperture, enter only the X and Y dimensions; omit the hole dimensions. To define a hole, enter one dimension for a round hole, two for a rectangle. The hole must fit within the aperture. Both rectangle and hole will be centered. For example, **%ADD22R,0.020X0.040*%** D-code 22 is a .02 x .04 solid rectangle.
O,<X-axis dimension>X<Y-axis dimension>[X<X-axis hole dimension>[X<Y-axis hole dimension>]] 	**Obround (oval).** The smallest side is terminated by half a circle. May be solid or open. If the X dimension is larger than Y, the shape will be horizontal. If the X dimension is smaller than Y, the shape will be vertical. To define a solid aperture, enter only the X and Y dimensions; omit the hole dimensions. To define a hole, enter one hole dimension for a round hole, two for a rectangular or square hole. If open, the hole must fit within the aperture. For example, **%ADD22O,0.020X0.04X0.005X0.010*%** D-code 22 is a vertical obround .02 wide x .04 high with a .05 x .01 rectangular hole.
P,<outer diameter> X<number of sides> [X<degrees of rotation> [X<X-axis hole dimension> X<Y-axis hole dimension>]]	**Regular polygon**. May be solid or open. To define a solid aperture, enter only the outer diameter and number of sides (3 to 12). This aperture can be rotated. Without rotation one vertex is on the X-axis through the center.

	Rotation is expressed in integer degrees; positive is counterclockwise, negative is clockwise. Note: Holes do not rotate. To enter hole dimension modifiers you need to enter a rotation angle first. Enter rotation angle 0 if you do not want to rotate. For example, **%ADD17P,.030X4X0.0*%** D-code 17 is a regular polygon with an outer diameter of .03 and 4 sides, without hole.

4.1.6 Examples

Syntax	Comments
%ADD10C,.025*%	Define D-code 10: 25 mil round
%ADD22R,.050X.050X.027*%	Define D-code 22: 50 mil square with 27 mil round hole
%ADD57O,.030X.040X.015*%	Define D-code 57: obround 30 x 40 mil with 15 mil round hole
%ADD30P,.016X6*%	Define D-code 30: polygon with 16 mil outside dimension and 6 sides
%ADD15CIRC*%	Define D-code 15: a special aperture described by aperture macro CIRC defined previously by an aperture macro

4.2 AM - Aperture Macro

The AM parameter is used to define special apertures consisting of building blocks called primitives. The special aperture macros may be used in AD parameter definitions just like the standard apertures (that is, circle, rectangle, obround, polygon and thermal). Every special aperture must be described before the D-code associated with it.

Special apertures offer two advantages over standard apertures:

- They allow multiple shapes called primitives to be combined in a single aperture, which permits creation of unusual aperture shapes.
- Aperture macro modifiers may be variable. Variable modifiers are supplied by the AD parameter that references the aperture macro. An aperture macro variable may be an arithmetic function of other macro variables.

4.2.1 Contents

An aperture macro contains the following elements:

- Aperture macro name
- One or more of the seven aperture primitives. (See table below)
- Primitive modifiers specifying exposure, position, dimensions, etc.
- Variable primitive modifiers supplied by the AD parameters
- Optional embedded comment block
- Arithmetic operators

4.2.2 Syntax Rules

- Like other parameters, enclose each parameter block with the delimiter %.
- Within the AM parameter block, separate each primitive and modifier group by the end-of-block character *.
- Separate modifiers by commas within each primitive group.
- Modifiers may be number, such as 0, 1, 2, or 9.05, or they may be variable modifiers supplied by the AD parameter using the special aperture.
- Identify variable modifiers to be supplied by the AD parameter as $n where n indicates the order in which the modifier is expected in the AD parameter. $1 would be the first variable modifier expected in the AD parameter, $2 the second, and so on, numbering sequentially from left to right. If an absolute value is entered instead of a variable, the variables shift right. For example, if an absolute value is entered for the first variable, the next variable becomes $1 even though it is the second modifier of the primitive.
- The interpretation of each modifier depends on the primitive. See table below.
- Do not begin a variable primitive modifier with a minus sign (for example, -$1). To indicate negative, precede the variable with 0 (for example, 0-$1).
- Start optional comment strings with a leading 0 (for example, *0 is a comment*).
- Position and dimensions are expressed in the units specified by the MO parameter. Decimal points are permitted.

- The following arithmetic operators can be used with variable modifiers:

Operator	Function
+	Add
-	Subtract
/	Divide
x	Multiply
=	Equate
n	Number, with or without decimal point

The standard arithmetic precedence rules apply.

- Exposure can be *on* or *off*. Exposure on creates a solid part of the aperture. Exposure off creates a hole in it. (Note: A hole is transparent. One sees the objects below it. This is *not* the same as *clear* Layer Polarity, where all object below are cleared or erased.)

  Exposure is set with the exposure modifier:

  0 = off

  1 = on

  2 = toggle exposure mode (0 if exposure mode was not yet set)

- Rotation angle is expressed in integer degrees; positive for counterclockwise rotation, negative for clockwise rotation.

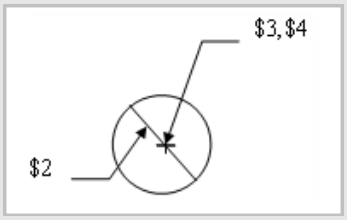
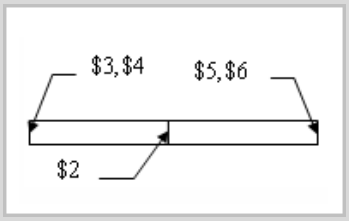
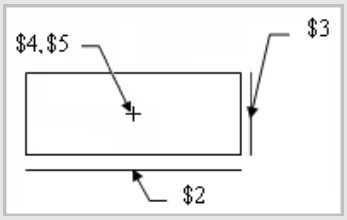
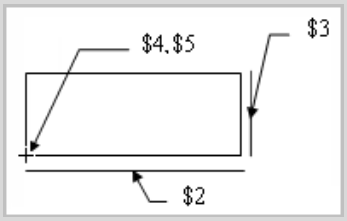
4.2.3 Data Block Format

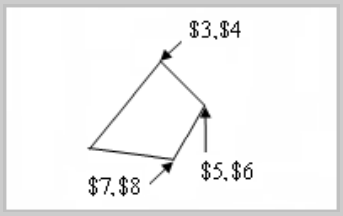
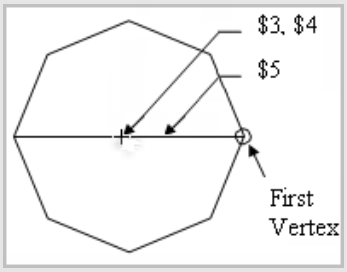
The syntax for the AM parameter is:

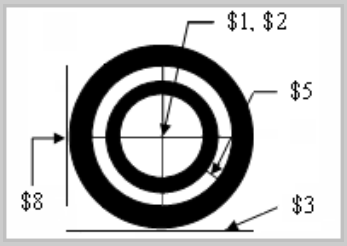
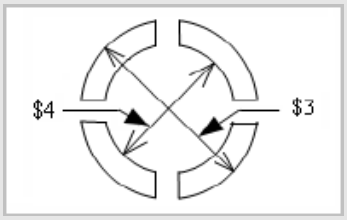
%AM<aperture macro name>*<primitive number>,<modifier$1>,<modifier$2>,[<...>]*[<primitive number>[<modifiers>]]*...*%

Syntax	Comments
AM	**AM** for Aperture Macro
<aperture macro name>	The name of the Aperture macro
<primitive number>, <modifier$1>, <modifier$2>, [<...>]	The primitive number with modifiers. The primitive number specifies the shape (e.g. polygon) and the modifier specify parameters (.e.g. diameter). The modifiers depend on the primitive. Use either a value (e.g. 0.050 for a diameter) or a variable placeholder (for example, $1 for exposure).

4.2.4 Primitives

Prim` Num	Graphical Description	Var Mod	Description
1	**Circle** 	**$1**	Exposure on/off/toggle
		$2	Diameter
		$3	X center position
		$4	Y center position
2 or 20	**Line (vector):** a line defined by the line width and the beginning and end points. The line ends are rectangular. 	**$1**	Exposure on/off/toggle
		$2	Line width
		$3	X start point
		$4	Y start point
		$5	X end point
		$6	Y end point
		$7	Rotation angle around the origin.
21	**Line (center):** a rectangle defined by width, height, and center point. 	**$1**	Exposure on/off/toggle
		$2	Rectangle width
		$3	Rectangle height
		$4	X center point
		$5	Y center point
		$6	Rotation angle around the origin.
22	**Line (lower left):** a rectangle defined by width, height, and the lower left point. 	**$1**	Exposure on/off/toggle
		$2	Width
		$3	Height
		$4	X lower left point
		$5	Y lower left point
		$6	Rotation angle around the origin.
4	**Outline**: the area enclosed	**$1**	Exposure on/off/toggle

	by an n point polygon, defined by its start point and n subsequent points. The outline must be closed, i.e. the last point must be equal to the start point. Self-intersecting outlines are not allowed.  The X and Y coordinates are *not* modal, both the X and the Y coordinate must be specified for all points. Older versions of this document gave a minimum implementation limit for n of 50. This has proven to be too restrictive, and the limit is now increased to 4000.	**$2**	n, the number of subsequent points
		$3, $4	X and Y of the start point
		$5, $6	X and Y of subsequent point #1
		$...	X and Y of further susbsequent points
		$(3+2n), $(4+2n)	X and Y of subsequent point #n
		$(5+2n)	Rotation angle around the origin.
5	**Polygon**: an n-sided regular polygon (n 3 to 12 inclusive), a center point, diameter, and rotation.  	**$1**	Exposure on/off/toggle
		$2	Number of sides
		$3	X center point
		$4	Y center point
		$5	Outer diameter
		$6	Rotation angle around the origin. Without rotation, one vertex is on the X-axis through the center.

6	**Moiré:** a cross hair centered on concentric rings (annuli). The outer circle of first ring is equal outer diameter $3. Moving further towards the center there is a gap $5, and then the second ring. The process stops when there are $6 rings are or the center is reaches, whichever comes first. If there is not enough room for the last ring it becomes a full disc centered on the origin.  Exposure is always on.	**$1**	X center point
		$2	Y center point
		$3	Outer diameter
		$4	Ring thickness
		$5	Gap between rings
		$6	Maximum number of rings
		$7	Cross hair thickness
		$8	Cross hair length
		$9	Rotation angle around the center point
7	**Thermal**: a ring (annulus) interrupted by four gaps.  Exposure is always on. Note: $5 must be smaller than $3. The inner circle disappears when $5 is bigger than $4,	**$1**	X center point
		$2	Y center point
		$3	Outer diameter
		$4	Inner diameter
		$5	Gap thickness
		$6	Rotation angle around the center point of the thermal. Without rotation, the gaps are on the axes through the center.

4.2.5 Examples

All units are inch

Example 1

Defines an aperture macro named DONUTFIX consisting of two concentric circles with fixed diameter sizes.

%AMDONUTFIX*1,1,0.100,0,0*1,0,0.080,0,0*%

Syntax	Comments
AMDONUTFIX	Define an aperture macro named DONUTFIX

1,1,0.100,0,0	Circle (1), exposure on (1), diameter (0.100), X center (0), Y center (0)
1,0,0.080,0,0	Circle (1), exposure off (0), diameter (0.080), X center (0), Y center (0)

The AD parameter using this aperture macro will look like the following:

%ADD33DONUTFIX*%

Example 2

Defines an aperture macro named DONUTVAR consisting of two concentric circles with variable diameter sizes:

%AMDONUTVAR*1,1,$1,$2,$3*1,0,$4,$2,$3*%

Syntax	Comments
%AMDONUTVAR*	Define an aperture macro named DONUTVAR
1,1,$1,$2,$3	Circle (1), exposure on (1), diameter ($1), X center ($2), Y center ($3) all to be supplied by AD parameter
1,0,$4,$2,$3	Circle (1), exposure off (0), diameter ($4), X center and Y center ($2 and $3, same as first circle)

The AD parameter using this aperture macro might look like the following:

%ADD34DONUTVAR,0.100X0X0X0.080*%

Example 3

Defines an aperture macro named DONUTCAL consisting of two concentric circles with diameter of the second circle defined as a function of the diameter of the first:

%AMDONUTCAL*1,1,$1,$2,$3*$4=$1x0.75*1,0,$4,$2,$3*%

Syntax	Comments
AMDONUTCAL	Define an aperture macro named DONUTCAL
1,1,$1,$2,$3	Circle (1), exposure on (1), diameter ($1), X center ($2), Y center ($3) all to be supplied by AD parameter
$4=$1x0.75	Define variable $4 to be used to calculate the diameter of the inner circle. The diameter of this circle is 0.75 times the diameter of the outer circle.
1,0,$4,$2,$3	Circle (1), exposure off (0), diameter ($4), and center point

	X, Y ($2, $3, same as the first circle).

The AD parameter using this aperture macro might look like the following:

%ADD35DONUTCAL,0.020X0X0*%

This defines a donut with outer circle 0.02 and inner circle 0.015 inch

Example 4

Defines an aperture macro named TRIANGLE_15 rotated 15 degrees around the origin.

%AMTRIANGLE_15*4,1,3,1,-1,1,1,2,1,1,-1,15*%

Syntax	Comments
AMTRIANGLE_15	Define an aperture macro named TRIANGLE_15
4,1,3	Outline (4), exposure on (1), 3 subsequent points (3),
1,-1	Start point ([1,-1])
1,1,2,1,1,-1	Subsequent points ([1,1], [2,1], [1,-1])
15	Rotation (15) degrees

4.2.6 SM – Symbol Mirror

The SM parameter is used to turn mirror symbol imaging either on or off. When on, all A- and/or B-axis symbol images are mirrored until another SM command is used. Notice that mirroring A-axis data flips the image about the B axis. Mirroring B-axis data flips the image about the A axis. SM does not mirror special apertures.

Note: It is recommended *not* to use the SM parameter. Avoid it like the plague. The exception for special apertures is confusing and leads to mistakes.

4.2.6.1 Data Block Format

The syntax for the S**M** parameter is:

%SM[A<0 or 1>][B<0 or 1>]*%

Syntax	Comments
SM	**SM** for Symbol Mirror
A<0 or 1>	Use A0 to disable mirroring Use A1 to invert the A-axis. The image will be flipped over the B-axis.
B<0 or 1>	Use B0 to disable mirroring Use B1 to invert the B-axis. The image will be flipped over the A-axis.

4.2.6.2 Examples

Syntax	Comments
%SMA0B0*%	No mirroring of A or B axis (default)
%SMA0B1*%	No mirroring of A-axis data Invert B-axis data, flipping the image over the A-axis
%SMB1*%	Keep A mirror and Invert B-axis data

4.3 Directive parameters

AS	Axis Select
FS	Format Statement
MI	Mirror Image
MO	Mode of units
OF	Offset
SF	Scale Factor

Directive parameters control overall file processing.

It is recommended to place directive parameters at the beginning of the file. Each directive parameter controls processing until another one is encountered It is recommended to use each directive parameter only once in a file.

Directive parameters do not generate a new layer.

4.3.1 AS – Axis Select

The AS parameter assigns any two data axes to the output device's A and B axes.

4.3.1.1 Data Block Format

The syntax for the AS parameter is:

AS A<X or Y>B<X or Y>*

Syntax	Comments
AS	**AS** for Axis Select
AX or AY	Assign output device axis A to data axis X or Y Default is AX
BX or BY	Assign output device axis B to date axis X or Y Default is BY

4.3.1.2 Example

Syntax	Comments
%ASAYBX*%	Assign output device axis A to data axis Y and output device axis B to data axis X

4.3.2 FS – Format Specification

The FS parameter specifies the format of the coordinate data. It is a mandatory parameter. The FS parameter can only be used once (1) in the file. It must be specified before the first use of coordinate data. It is recommended to use it as the very first line, maybe after some general comments.

The FS parameter specifies the following format characteristics:

- Number of integer and decimal places in coordinate data (coordinate format)
- Zero omission (leading or trailing zeroes omitted)
- Absolute or incremental coordinate notation


Note: Explicit decimal points in coordinates are not allowed.

4.3.2.1 Coordinate Format

The coordinate format specifies the number of integer and decimal places in the coordinate data. For example, the 2.3 format specifies 2 integer and 3 decimal places. A maximum of 7 integer and 7 decimal places may be specified (nnnnnnn.nnnnnnn). The same format must be defined for X and Y. Signs are allowed. The "+" sign is not mandatory.


Note: The implementation limit on integer and decimal places was 6 in the past. However, some applications started to generate 7 decimal places because they needed the accuracy. To cater for this need we have raised this limit to 7. However, there are probably still a number of Gerber readers in use that can only handle 6. Therefore we recommend using 7 decimal places only if the extra accuracy is needed.

4.3.2.2 Zero Omission

Zero omission compresses data by omitting *either* leading or trailing zeroes from the coordinate values.

With *leading zero omission* some or all leading zeroes may be omitted but all trailing zeroes must be retained. To interpret the coordinate string, it is first padded with zeroes in front until its length fits the coordinate format. For example, with 2.3 coordinate format, "015" is padded to "00015" and therefore represents 0.015.

With *trailing zero omission* some or all trailing zeroes may be omitted but all leading zeroes must be retained. To interpret the coordinate string, it is first padded with zeroes at the back until its length fits the coordinate format. For example, with 2.3 coordinate format, "15"is padded to "15000" therefore represents 15.000.

With *no zeroes omitted* all zeroes are present, the format length fits and there is no padding. It is conventional to indicate leading zero omission.

4.3.2.3 Absolute or Incremental Notation

Coordinate values may be expressed as either absolute distances from the origin or incremental distances from the preceding coordinate position.


Note: It is recommended to use absolute notation only. With incremental notation rounding errors can accumulate and become problematic. Incremental notation is a relic from a distant past, when it was of paramount importance to save bytes on punched

paper tape. The small pleasure of saving a few bytes is quickly forgotten when the first data error happens. We will deprecate incremental notation in future revisions of this document.

4.3.2.4 Data Block Format

The syntax for the FS parameter is:

%FS<L or T><A or I>[Nn][Gn]<Xn><Yn>[Dn][Mn]

Syntax	Comments
FS	**FS** for Format Specification
<L or T>	Use L to omit leading zeroes. Use T to omit trailing zeroes.
<A or I>	Use A for absolute coordinate values. Use I for incremental coordinate values.
<Xnm> and <Ynm>	Enter X or Y and the number n of integer and m decimal places in the coordinate data for each axis. The value of n and m ranges from 0 to 7. The X and Y coordinates must have the same format!

4.3.2.5 Example

Syntax	Comments
%FSLAX25Y25*%	Coordinate data will have leading zeros omitted and be expressed as absolute positions with 2 integer and 5 decimal places in both axes.

4.3.3 MI – Mirror Image

The MI parameter is used to turn axis mirroring on or off. When on, all A- and/or B-axis data following the parameter will be mirrored (that is, inverted or multiplied by -1) until another MI command is used. Mirroring A-axis data flips the image about the B axis and mirroring B-axis data flips the image about the A axis. **MI does not mirror special apertures!**

Note: It is recommended *not* to use the MI parameter. Avoid it like the plague. The exception for special apertures is confusing and leads to mistakes. Use the AS parameter to correlate data axes with output device axes.

4.3.3.1 Data Block Format

The syntax for the **MI** parameter is:

%MI[A<0 or 1>][B<0 or 1>]*%

Syntax	Comments
MI	**MI** for Mirror image
A<0 or 1>	Use A0 to disable mirroring Use A1 to invert the A-axis. The image will be flipped over the B-axis.
B<0 or 1>	Use B0 to disable mirroring Use B1 to invert the B-axis. The image will be flipped over the A-axis.

4.3.3.2 Examples

Syntax	Comments
%MIA0B0*%	No mirroring of A or B axis (default)
%MIA0B1*%	No mirroring of A-axis data Invert B-axis data, flipping the image over the A-axis.
%MIB1*%	Keep A mirror and Invert B-axis data

4.3.4 MO – Mode

The MO parameter specifies that dimension data should be interpreted as inches or millimeters.


Note: Use the FS parameter to specify the integer and decimal place format.

4.3.4.1 Data Block Format

The syntax for the **MO** parameter is:

%MO<IN or MM>*%

Syntax	Comments
MO	**M** for Mode
<IN or MM>	Use IN to specify units in inches Use MM to specify units in millimeters

4.3.4.2 Examples

Syntax	Comments
%MOIN*%	Dimensions in inches
%MOMM*%	Dimensions in millimeters

4.3.5 OF - Offset

The OF parameter is used to offset the final image up to plus or minus 99999.99999 units from the imaging device 0,0 point. The data may be offset along the imaging device A or B axis, or both. Values used with the OF parameter are expressed in units specified by the MO parameter, are always absolute, and are used primarily with absolute coordinate data. Incremental coordinate data may be offset simply by moving the imaging device to the desired offset position before starting the image. The FS parameter specifies whether the data is absolute or incremental.

4.3.5.1 Data Block Format

The syntax for the **OF** parameter is:

OF[A<offset>][Boffset>]

Syntax	Comments
OF	**OF** for Offset
A<sn>	The offset along the output device A axis s = optional sign (+ or -) n = offset value
B<sn>	The offset along the output device B axis s = optional sign (+ or -) n = offset value

4.3.5.2 Examples

Syntax	Comments
%OFA0B0*%	No offsets (default)
%IOA1.0B-1.5*%	Offset of 1 unit along the A axis Offset of -1.5 units along the B axis
%IOB5.0*%	Offset of 0 units along the A axis Offset of 5 units along the B axis

4.3.6 SF – Scale Factor

The SF parameter is used to specify a scale factor from 0.0001 to 999.99999 for the output device A- and/or B-axis data. The factor may be different for each axis. All data following the parameter will be multiplied by the factor until another SF parameter is encountered.


Note: Use the AS parameter to correlate data axes with output device axes

4.3.6.1 Data Block Format

The syntax for the **SF** parameter is:

%SF[A<factor>][B<factor>]*%

Syntax	Comments
SF	**SF** for Scale Factor
A<factor>	The A-axis data scale factor
B<factor>	The B-axis data scale factor

4.3.6.2 Example

Syntax	Comments
%SFA1B1*%	Scale factor 1 (default)
%SFA.5B3*%	Multiply A-axis data by 0.5 Multiply B-axis data by 3

4.4 Image parameters

4.4.1 IP – Image Polarity

The IP parameter sets the positive or negative polarity of the entire file image. With positive polarity the image is exposed as is. With negative polarity dark areas are displayed as clear, and clear areas displayed as dark. The default polarity is positive.

The IP parameter affects the entire image. If it occurs more than once, the last one is valid for the whole file. It is recommended to use it only at the beginning of the file.

(*Image polarity* is not the same as *layer polarity,* see the LP parameter).

4.4.1.1 Data Block Format

The syntax for the **IP** parameter is:

%IP<POS or NEG>*%

Syntax	Comments
IP	**IP** for Image Polarity
<POS>	Output the image with positive polarity. (Default)
<NEG>	Output the image with negative polarity

4.4.1.2 Examples

Syntax	Comments
%IPNEG*%	Output the image with negative polarity Note: IPNEG does not toggle the polarity

4.4.2 IR – Image Rotation

The IR parameter is used to rotate the entire image counterclockwise in 90° increments around the 0, 0 coordinate. All objects are rotated. If there is no IR parameter there is no rotation.

The IR parameter affects the entire image. If it occurs more than once, the last one is valid for the whole file. It is recommended to use it only at the beginning of the file.

4.4.2.1 Data Block Format

The syntax for the **IR** parameter is:

%IR<n>*% with n 0, 90, 180 or 270

Syntax	Comments
IR	**IR** for Image Rotation
<0>	Rotate the image over 0°; no rotation
<90>	Rotate the image over 90°
<180>	Rotate the image over 180°
<270>	Rotate the image over 270°

4.4.2.2 Examples

Syntax	Comments
%IR0*%	No rotation (default)
%IR90*%	Rotate the image over 90° counterclockwise
%IR270*%	Rotate the image over 270° counterclockwise

4.5 Layer-specific parameters

KO Knockout

LN Layer Name

LP Layer Polarity

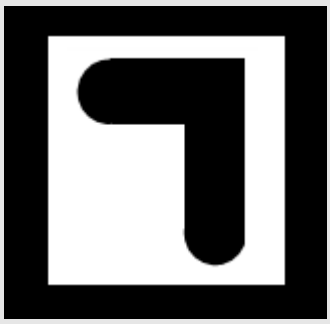
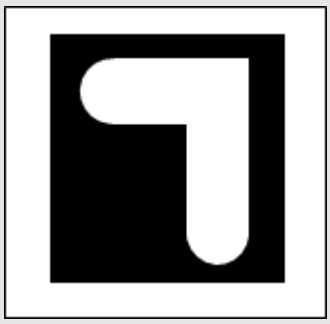
SR Step and Repeat

Layer-specific parameters specify the processing of information layers. (Information layers are not to be confused with board layers). These parameters may be used more than once in a file. Layer-specific parameters *always* generate a new layer.

Note: Generating a new layer resets the graphic state, including the current point.

4.5.1 KO – Knock Out

The KO parameter specifies that a rectangular region of the image will have the opposite layer polarity (dark or clear) of the image or information layer in which it occurs, making it a "knockout" from the surrounding region.

	
Positive image polarity Dark layer polarity Clear knockout	Negative image polarity Clear layer polarity Dark knockout

KO is typically used to:

- Set the initial background polarity of the final image
- Knock out a region
- Disable knockout on the current information layer

The knockout area is defined by its lower left point and height and width.

The knockout is applied to all data following the KO parameter until the knockout is disabled. To disable a previously defined knockout, enter KO without modifiers.

4.5.1.1 *Data Block Format*

The syntax for the **KO** parameter is:

%KO[C or D][X<coordinate>Y<coordinate>I<width>J<height>]

Syntax	Comments
KO	**KO** for Knock Out
[C or D]	Use C for clear polarity of the background Use D for dark polarity of the background To disable a previously enabled knockout enter neither C nor D.
X<coordinate>	Lower left X coordinate
Y<coordinate>	Lower left Y coordinate
I<width>	Width of the knockout area
J<height>	Height of the knockout area

4.5.1.2 *Examples*

Syntax	Comments
%KODX0I0I20J26*%	Create a dark knockout that extends from 0,0 to 20,26.
%KO*%	Disable a previously enabled knockout.

4.5.2 LN – Layer Name

The LN parameter is used to assign a name of up to 77 alphanumeric characters to the information layer that follows the parameter in the RS-274X file.

4.5.2.1 Data Block Format

The syntax for the **LN** parameter is:

%LN<character string>*%

Syntax	Comments
LN	**LN** for Layer Name
<character string>	Up to 77 alphanumeric characters.

4.5.2.2 Examples

Syntax	Comments
%LNInternal VCC*%	Layer name “Internal VCC”

LP – Layer Polarity

The LP parameter is used to specify the dark or clear polarity of the information layer or layers following it. The layer polarity applies to all data following the LP parameter until another LP parameter is encountered. The default polarity is dark.

Note: *Layer polarity* is not the same as *image polarity*, see the IP parameter.

4.5.2.3 *Data Block Format*

The syntax for the **LP** parameter is:

%LP<C or D>*%

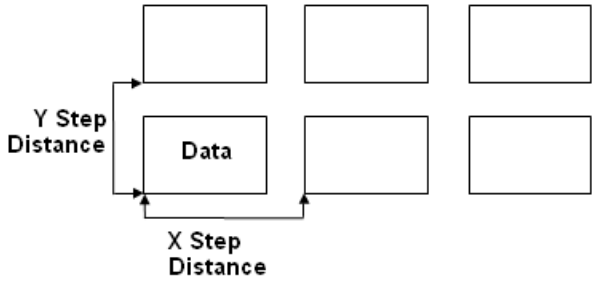
Syntax	Comments
LP	**LP** for Layer Polarity
<C or D>	Use C for clear polarity Use D for dark polarity

4.5.2.4 *Examples*

Syntax	Comments
%LPD*%	Use dark polarity (default)
%LPC*%	Use clear polarity for all succeeding data

4.5.3 SR – Step and Repeat

The SR copies the data following it a number of time with a specified step (distance) between the repeats (instances). The number of repeats and the steps can be different in X and Y. An SR parameter without repeat and step values disables the previous SR parameter.



4.5.3.1 Data Block Format

The syntax for the **SR** parameter is:

%SR[X<repeats>][Y<repeats>][I<X-axis step>][J<Y-axis step>]

Syntax	Comments
SR	**SR** for Step and Repeat
X<repeats>	The number of times the data is repeated along the X-axis
Y<repeats>	The number of times the data is repeated along the Y-axis
I<X-axis step>	The step between the repeats along the X-axis
J<Y-axis step>	The step between the repeats along the Y-axis

4.5.3.2 Example

Syntax	Comments
%SRX1Y1I0J0*%	Step and Repeat with 1 repeat, i.e. not repeated (default)
%SRX2Y3I2.0J3.0*%	Repeat the image 2 times along the X axis and 3 times along the Y axis. X-axis repeats are spaced 2.0 units apart. Y-axis repeats are spaced 3.0 units apart.
%SRX4I5.0J2*%	Repeat the image 4 times along the X axis with 5.0 units from one step to the next. The J modifier is ignored because no Y repeats are specified.
%SR*%	Disable the previous SR parameter.

5 Function Codes

Function codes specify how coordinate data should be interpreted. Function codes apply to Coordinate Data in the same block as well as to subsequent Coordinate Data. They do not affect Coordinate Data preceding the block in which they occur.

Functions Codes and Coordinate Data normally follow the Parameters.

Codes are grouped as follows:

1 **G-codes** (general functions) specify how to interpolate and move to the coordinate locations following the code until changed or until a new layer is generated (modal).

2 **D-codes** (drawing functions) select and control tools, specify line type, etc.

3 **M-codes** (miscellaneous functions) perform the program end.

The next section lists the D-codes, G-codes and M-codes. Subsequent sections explain the more intricate codes in detail.

5.1 Overview

Code	Function	Comments
D01	Exposure and Draw mode on.	Draws a line using the current aperture. (There are strict limitations on the apertures that can be used for drawing.) The current point is moved to the end point.
D02	Exposure and Draw mode off.	Generates no image. The current point is moved. This is the default state
D03	Set Flash mode.	Flash the current aperture. The current point is moved to the flash point.
D10-D999	Select an aperture defined by an AD parameter	

D-codes

Code	Function	Comments
G01	Set linear interpolation mode (default)	A modifier of the draw operator. For clarity and robustness it is recommended not to use the default but always to set the interpolation mode explicitly.
G02	Set clockwise circular interpolation mode	
G03	Set counterclockwise circular interpolation mode	
G04	Ignore data block	Used for comments
G36	Turn on Outline Fill	See below.
G37	Turn off Outline Fill	

G54	Select aperture	This historic code optionally precedes an aperture D-code. It is superfluous and deprecated.
G70	Specify inches	
G71	Specify millimeters	
G74	Set Single quadrant mode. (default)	A modifier of the circular interpolation.
G75	Set Multi quadrant mode	
G90	Specify absolute format	See also FS parameter. It is recommended to use absolute format only.
G91	Specify incremental format	

G-codes

Code	Function	Comments
M00	Program stop	This historic code has no effect on the image. It is deprecated.
M01	Optional stop	This historic code has no effect on the image. It is deprecated.
M02	End of program	Ends image generation. Every file must end in a M02

M-codes

5.2 Linear Interpolation (G01 (G1))

Linear interpolation generates a straight line from the current point to the X, Y coordinate specified by the data block. The current point is set to the X, Y coordinate.

5.2.1 Data Block Format

The syntax for the **Linear Interpolation code**s is:

Gn[Xsn][Ysn][Dn]*

Syntax	Comments
Gn	Linear interpolation mode
Xsn	s = optional sign (+ or -) n = X coordinate of line endpoint
Ysn	s = optional sign (+ or -) n = Y coordinate of line endpoint
Dn	Exposure n = 01 => On n = 02 => Off

5.3 Circular Interpolation (G02 (G2), G03 (G3), G74, G75)

Circular interpolation generates a circular arc from the current point to the X, Y to the coordinate in the data block. The current point is then set to the X, Y coordinate.

There are two orientation modes:

- clockwise mode, specified by G02
- counterclockwise mode, specified by G03

The orientation is defined around the center of the arc, moving from begin to end.

There are two quadrant modes:

- single quadrant mode
- multi quadrant mode

In single quadrant mode the arc is not allowed to extend over more than 90°. The following relation must hold: 0° =< |arc angle| =<90°. If the start point of the arc is equal to the end point, the arc has length zero, i.e. it covers 0°. A data block is required for each quadrant. A minimum of four data blocks is required for a full circle.

In multi quadrant mode the arc is allowed to extend over more than 90°. To avoid ambiguity between 0° and 360° arcs the following relation must hold: 0° < |arc angle|. If the start point of the arc is equal to the end point, the arc is a full circle of 360°.

The default mode is single quadrant. The G74 and G75 command allow switching between the two modes. A data block containing G75 enables multi quadrant mode. Every block following it will be interpreted as multi quadrant, until cancelled by a G74. A data block containing G74 code turns off multi quadrant mode, reverting to single quadrant mode.


Some Gerber writers erroneously take G75 as a default. This leads to obscure errors when readers try deal with these wrong files by guessing the default used. If you specify G74/G75 explicitly the mode is unequivocal for both reader and writer. This is safer and more robust than relying on the default. **We strongly recommend always specifying the G74/G75 explicitly when creating a Gerber file.**

5.3.1 Single Quadrant Mode

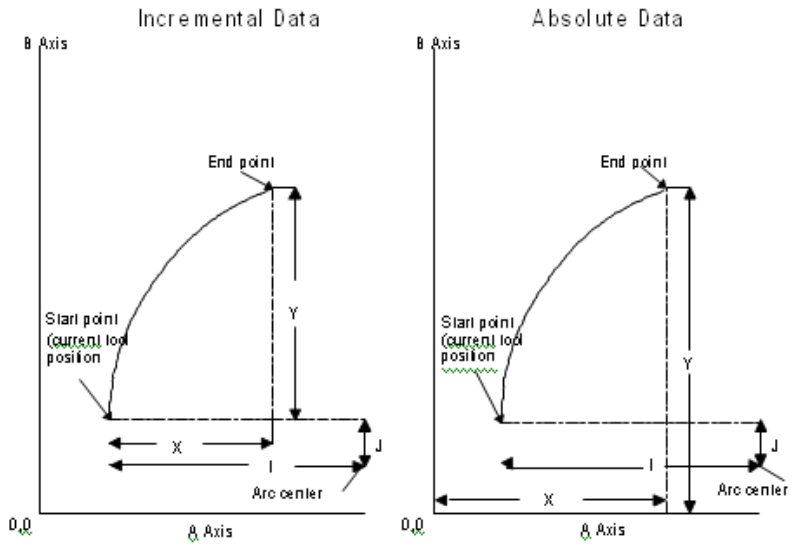
5.3.1.1 Data Block Format

The syntax in **Single Quadrant Circular Mode** is:

Gn[Xsn][Ysn][In][Jn][Dn]*

Syntax	Comments
Gn	Specifies direction of arc n = 02 => clockwise n = 03 => counterclockwise
Xsn	s = optional sign (+ or -) n = X coordinate of arc endpoint
Ysn	s = optional sign (+ or -) n = Y coordinate of arc endpoint
In	n = The offset between the arc start point and the center parallel to the X axis. The value is always positive. A sign is not allowed. The direction to the center is determined implicitly.
Jn	n = The offset between the arc start point and the center parallel to the Y axis. The value is always positive. A sign is not allowed. The direction to the center is determined implicitly.
Dn	Exposure n = 01 => On n = 02 => Off

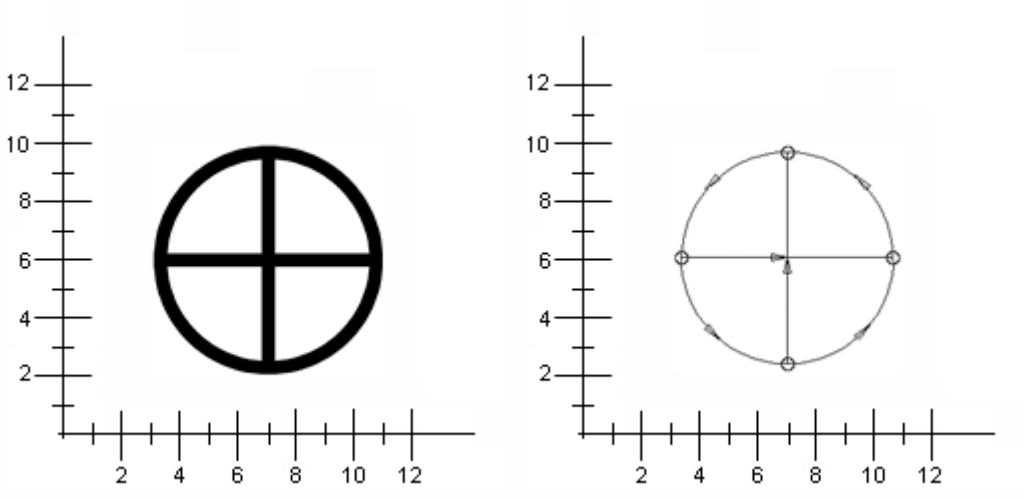
5.3.1.2 Image



5.3.1.3 Example

Syntax	Comments
G74* G54D10* G01X1100Y600D02* G03X700Y1000I400J0D01* X300Y600I0J400* X700Y200I400J0* X1100Y600I0J400* G01X300D02* X1100D01* X700Y200D02* Y1000D01*	Single quadrant mode Use aperture D10 Start from 11,6 Quarter arc (radius 4) to 7,10 Quarter arc (radius 4) to 3,6 Quarter arc (radius 4) to 7,2 Quarter arc (radius 4) to 11,6 Start from 3,6 Line to 11,6 Start from 7,2 Line to 7,10

Resulting Image



5.3.2 Multi Quadrant Mode

5.3.2.1 Data Block Format

The syntax in **Multi Quadrant Mode** is:

Gn Xsn Ysn Isn Jsn Dn

Note: The I and J offsets are signed. If no sign is present, the offset is positive.

Syntax	Comments
Gn	Specifies direction of arc (see also table 5) n = 02 => clockwise n = 03 => counterclockwise

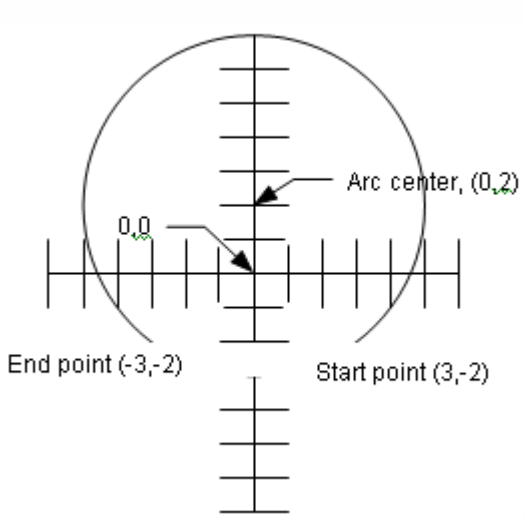
Xsn	s = optional sign (+ or -) n = X coordinate of arc endpoint
Ysn	s = optional sign (+ or -) n = Y coordinate of arc endpoint
Isn	s = optional sign (+ or -) n = Defines the incremental distance between the arc start point and the center measured parallel to the X axis.
Jsn	s = optional sign (+ or -) n = Defines the incremental distance between the arc start point and the center measured parallel to the Y axis.
Dn	Exposure (see table 4) n = 01 => On n = 02 => Off

5.3.3 Example

Syntax	Comments

`G75*`
`G01X300Y-200D02*`
`G03X-300Y-200I-300J400D01`
`G01*`
	`Multi quadrant mode`
	`Start from 3,-2`
	`Arc counterclockwise (rel. center -3,4) to -3,-2`
	`Back to linear interpolation mode`

Resulting Image



5.4 Apertures allowed for drawing

The solid circle and the solid rectangle or square standard apertures are the only ones that can be used for drawing. No other standard apertures can be used for drawing. No special aperture can be used for drawing, whatever its final shape.

(Of course, any aperture can be flashed.)

5.4.1 Circle Aperture

The solid circle aperture can be used for both linear and circular interpolations.

It's meaning is obvious, The line or arc is stroked with a round pen or aperture. The interpolation creates a line or arc with thickness equal to diameter of the circle and with a circular line ending.

5.4.2 Rectangle or Square Aperture

The solid rectangle aperture - rotated or not - can be used for linear interpolation only. It cannot be used for circular interpolation.

The line is stroked with the aperture as defined. An example:

If the aperture is aligned with the draw the result is a line with a straight line ending:

Circular interpolation is not allowed because it would result in a complex shape without general use

Note: The rectangle is *not* automatically rotated to be aligned with the draw.

5.5 Outline Fill (G36, G37)

The commands G36 and G37 create a filled area defining its closed outline.

G36 turns on outline fill, G37 turn it off. There are no variables or apertures. Following a G36 and before G37, all lines drawn with D01 are considered edges of the outline; D02 closes and fills the outline; D03 is not allowed.

Self-intersecting outlines are not allowed because their interpretation is not obvious. Outline edges can coincide, allowing cut-ins to create holes in solid areas.


Care must be taken that rounding errors, especially with arcs, does not turn a proper outline into a self-intersecting one, with unpredictable results; construct outlines defensively and round carefully.

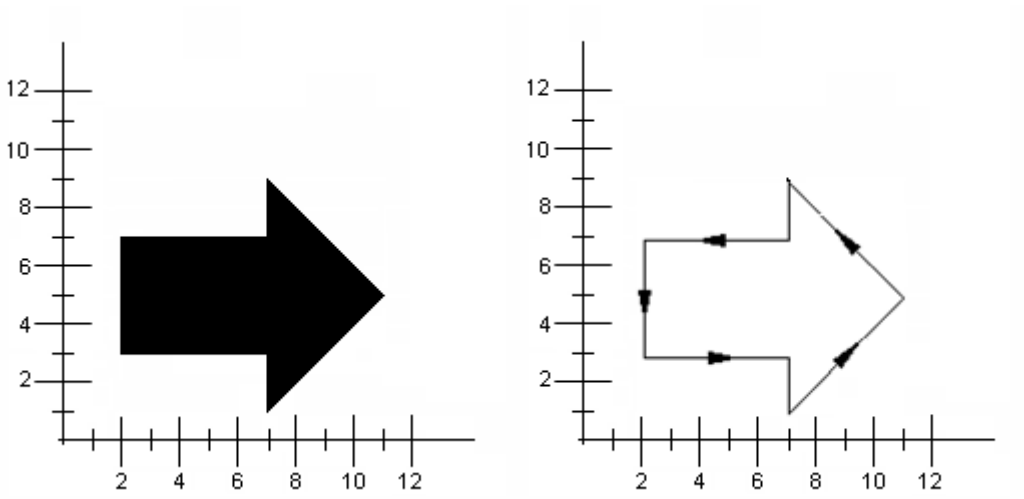
Outline area fill is much more efficient than the obsolete method of stroke fill.

In previous versions of this document “outline fill” was also called “polygon fill” This was a misnomer as arcs can be used in the outline, whereas a polygon is usually understood to have only straight edges.

5.5.1 Example 1: Simple outline

Syntax	Comments
G36* X200Y300D02* X700D01* Y100D01* X1100Y500D01* X700Y900D01* Y700D01* X200D01* Y300D01* G37*	Outline fill mode Start at 2,3 Line to 7,3 Line to 7,1 Line to 11,5 Line to 7,9 Line to 7,7 Line to 2,7 Line to 7,1 End of outline fill mode

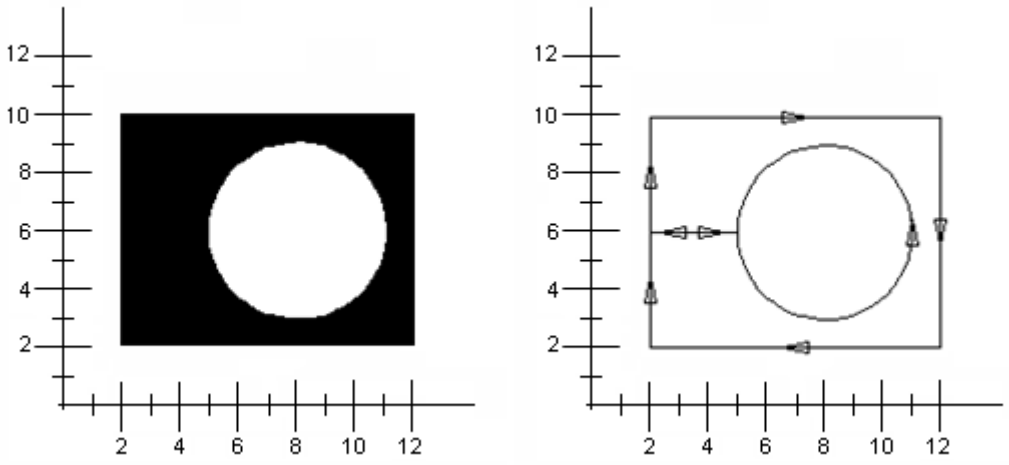
Resulting Image



5.5.2 Example 2: Cut In

Syntax	Comments
G36* X200Y1000D02* X1200D01* Y200* X200* Y600* X500* G75* G3X500Y600I0J300D01* G74* G1X200D01* Y1000* G37*	Outline fill mode Start at 2,10 Line to 12,10 Line to 12,2 Line to 2,2 Line to 2,6 Line to 5,6 Multi quadrant mode Full arc ccw (radius = 300) Single quadrant mode Line to 2,6 Line to 2,10 End of outline fill mode

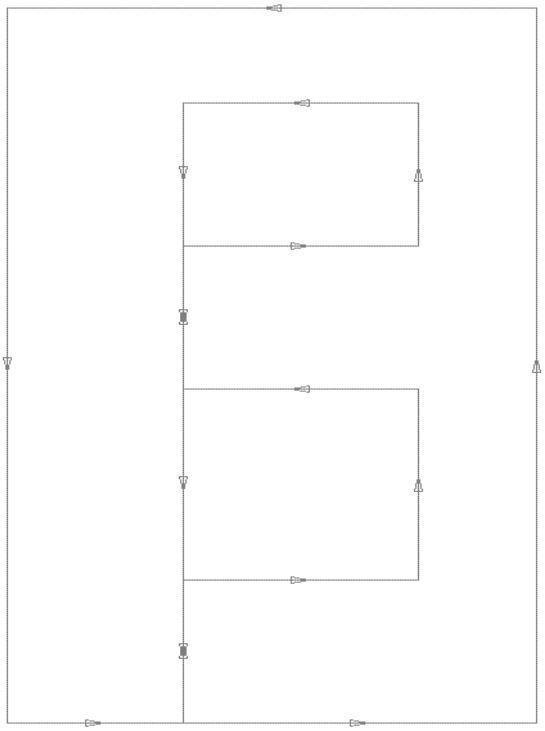
Resulting Image



5.5.3 Example 3: More complex cut-in

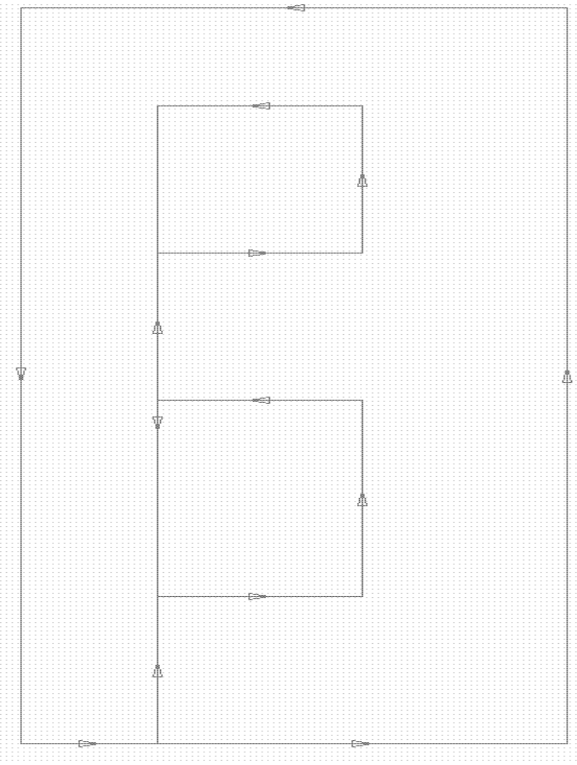
Cut-ins are susceptible to rounding problems. When the vertices move due to rounding the contour may become self-intersecting, with unpredictable results. Construct your cut-ins defensively, as in this example. Draws that are on top of one another have the *same* end vertices. When the vertices move under rounding, the draws will remain exactly on top of one another, and no self-intersections are created.

```
G36*
X1220000Y2570000D02*
X1250000D01*
Y2600000*
X1290000*
Y2640000*
X1250000*
Y2670000*
X1290000*
Y2700000*
X1250000*
Y2670000*
Y2640000*
Y2600000*
Y2570000*
X1310000*
Y2720000*
X1220000*
Y2570000*
G37*
```



The example below creates the same image, but with a less robust construction. The number of draws has been reduced by eliminating vertices between collinear draws. When the vertices move slightly due to rounding, the draws that were on top of one another may become intersecting, with unpredictable results. When a RS-274X file moves from system to system, numerical rounding must be expected. Therefore this construction is bad practice.

```
G36*
X1110000Y2570000D02*
Y2600000D01*
X1140000*
Y2640000*
X1110000*
Y2670000*
X1140000*
Y2700000*
X1110000*
Y2570000*
X1170000*
Y2720000*
X1090000*
Y2570000*
X1110000*
G37*
```



6 RS-274X Best Practices

Some RS-274X files produce the desired image but are needlessly cumbersome to work with or error-prone.
Some common poor practices:

Poor Practice	Problems	Best Practice
Low numerical precision.	Poor registration of objects between PCB layers. Rounding when writing the file can results in outlines to self-intersect, invalid arcs, zero-arcs, with unexpected result downstream.. Software processing the file will unavoidably add further numerical rounding, aggravating the problem.	**Always use high numerical precision.** Do not sacrifice precision to save a few bytes. Do not throw away the precision when writing the file
G74 and G75 not specified	Some Gerber readers use the wrong default. If you specify explicitly there can be no misunderstanding.	Always specify G74 or G75%%
Multi quadrant mode and rounding errors.	The begin- and end-point of an arc will move somewhat due to rounding. For a very small are they can happen to move on top of another. Under G75 mode the small arc suddenly becomes a full circle. Under G74 it remains small, actually zero size.	Use G74 single quadrant mode unless you are very careful with rounding on small arcs.
Long file with lots of identical X and Y coordinate values.	File needlessly long	Modal usage of coordinates within the same layer-specific parameter.
Painted or stroked pads.	Painted pads produce the correct image but are very awkward and time consuming for CAM software, e.g. for DRC checks, electrical test and so on. Stroking was needed for vector photoplotters in the 1960's and1970's, but these devices are as outdated as the mechanical typewriter Using them today is like sending your file on paper tape.	**Never use stroked pads**. Define pads, including SMD pads, with the AD and AM parameters.
Painted or stroked areas	Painted areas produce the correct image, but the files are needlessly large and the data is very confusing for CAM software. Stroking was needed for vector photoplotters in the 1960's and1970's, but these devices are as outdated as the mechanical typewriter.	**Never use stroked areas.** Use the Outline Fill commands;

7 Common Syntax Errors

Some RS-274X files do not produce the desired image due to incorrect interpretation of the specifications. Some common errors:

Symptom	Cause and Correct Usage
Polygons are smaller than expected	The inside diameter of a Regular Polygon is used instead of the outside diameter
Outline fill defined in a Layer Polarity Clear (%LPC) section erases a previously defined object on that location.	Outline fill as any other object in an %LPC section indeed *clears* (erases) the underlying objects. Clear does not mean transparent. Use outline fill commands with negative polarity before all objects you want to keep at that location.
A Mirror command is used to mirror a macro definition but the result is not as expected.	The Mirror command is not applied to aperture definitions. It is recommended not to use the Mirror command. Apply the transformation directly in the aperture definition and object coordinates.
Negative image is only visible if a positive background image is used.	Polarity in Image Polarity and Layer Polarity confusingly has different meanings. Use Layer Polarity to clear (make holes in objects).

8 Glossary

ABSOLUTE POSITION: Position expressed in Cartesian coordinates from the *origin* 0,0.

APERTURE: A shape that is used for drawing lines or flashing. (The name is historic; one used to expose shapes on film by shining light through an *aperture* in an aperture wheel.)

APERTURE MACRO: A parameter describing the geometry of a special aperture and assigns it to a D-code.

APERTURE PARAMETER: A parameter (AD or AM) that assigns an aperture definition to a D-code.

CIRCULAR INTERPOLATION: Drawing a circular arc.

DIRECTIVE PARAMETER: A parameter that controls overall file processing.

FILE IMAGE: The entire image, including all information layers.

INCREMENTAL POSITION: Position expressed as a distance in X and Y from the current point.

IMAGE PARAMETER: A parameter that applies to the entire file image.

INFORMATION LAYER. See layer.

KNOCKOUT: A rectangular region around an information layer whose polarity is the opposite of the layer polarity.

LAYER: An information section of RS-274X data that is treated as a unit, for example, rotated or repeated. (In the RS-274X context it does *not* mean the layer of a PCB.)

LAYER-SPECIFIC PARAMETER: A parameter that applies to a single information layer (KO, LN, LP, and SR).

LINEAR INTERPOLATION: Drawing a straight line.

,

MULTI QUADRANT MODE: The arc is allowed extend over more than 90°. If the start point of the arc is equal to the end point the arc is a full circle of 360°.

NUMERICAL PRECISION: The number of integer and decimal places used to express a number.

PARAMETERS: Commands that specify how the data should be processed.

POLARITY: When applied to the file image, positive polarity means the image is exposed black on white, and negative that it is exposed white on black. When applied to a layer, *dark* means that the object exposes the image area in dark (black) and *clear* means that the object clears or erases everything underneath it.

POLYGON FILL: See outline fill.

OUTLINE FILL: A feature to create solid (filled) areas. (Previously called polygon fill.)

SINGLE QUADRANT MODE: The arc cannot extend over more than 90°. If the start point of the arc is equal to the end point, the arc has length zero, i.e. covers 0°.

STEP AND REPEAT: A method by which successive exposures of a single image are made to produce a multiple image production master.

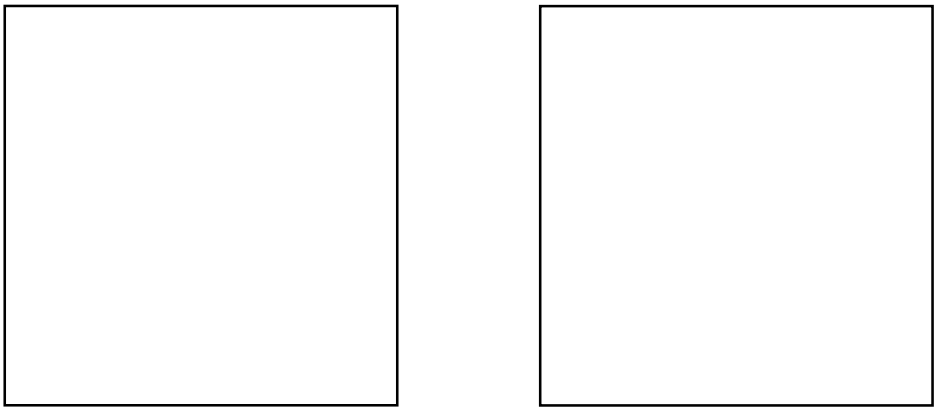
TRANSPARANT. : Part of an object that has no effect on the image, typically a hole. Any objects under the transparent part remain visible.

9 Sample Files

The examples on these pages illustrate the use of both Parameters and Function codes.

9.1.1 Example 1

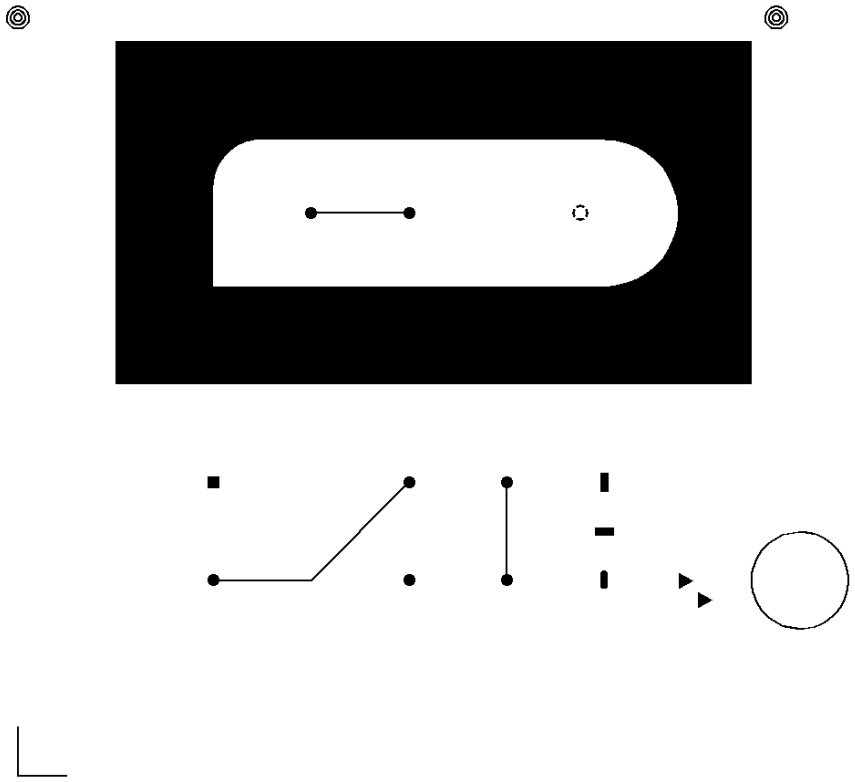
Example 1 is a single layer image with two square boxes.



G04 EXAMPLE 1: 2 BOXES*	
%FSLAX23Y23*%	Format statement - leading zeroes omitted, absolute coordinates, X2.3, Y2.3.
%MOIN*%	Set units to inches.
%OFA0B0*%	No offset
%SFA1.0B1.0*%	Scale factor is A1, B1
%ADD10C,0.010*%	Define aperture with D-code 10 as a 10 mil circle
%LNBOXES*%	Name layer “BOXES”.
G54D10*	
X0Y0D02*	Start from 0,0
X5000Y0D01*	Line to 5,0
Y5000*	Line to 5,5
X0*	Line to 0,5
Y0*	Line to 0,0
X6000D02*	Start from 6,0
X11000D01*	Line to 11,0
Y5000*	Line to 11,5
X6000*	Line to 6,5
Y0*	Line to 6,0
M02*	End of program

9.1.2 Example 2

Example 2 illustrates RS-274X image showing various shapes.



%FSLAX23Y23*%	Format Statement, Leading zeros omitted, absolute data, 2 integer digits and 3 fractional digits
%MOIN*%	Mode inches
%IJALBL*%	Image Justify
G04 Define Apertures*	Comment
%AMTARGET125*%	Aperture Macro
6,0,0,0.125,.01,0.01,3,0.003,0.150,0*%	Moire definition
%AMTHERMAL80*%	Aperture Macro
7,0,0,0.080,0.055,0.0125,45*%	Thermal definition
%ADD10C,0.01*%	Aperture definition, D10 is a circular aperture with 0.01” diameter
%ADD11C,0.06*%	Aperture Definition, D11 is a circular aperture with 0.06” diameter
%ADD12R,0.06X0.06*%	Aperture Definition, D12 is a rectangular aperture, 0.06” X 0.06”
%ADD13R,0.04X0.100*%	Aperture Definition, D13 is a rectangular

	aperture, 0.04” X 0.100”
%ADD14R,0.100X0.04*%	Aperture Definition, D14 is a rectangular aperture, 0.100” X 0.04”
%ADD15O,0.04X0.100*%	Aperture Definition, D15 is a obround aperture, 0.04” X 0.100”
%ADD16P,0.100X3*%	Aperture Definition, D16 is a 3 sided polygon 0.100” overall size
%ADD17P,0.100X3*%	Aperture Definition, D17 is a 3 sided polygon 0.100” overall size
%ADD18TARGET125*%	Aperture Definition, D18 is a special aperture called “TARGET125”
%ADD19THERMAL80*%	Aperture Description, D19 is a special aperture called "THERMAL80”
%LNXTEST1*%	Layer Name XTEST1
%LPD*%	Layer Polarity Dark
%SRX1Y1I0J0*%	Step and Repeat set to 1 X 1 (Not Required)
G54D10*	Aperture select
G01X0Y250D02*	Linear move with light off
X0Y0D01*	Linear move with light on
X250Y0D01*	Linear move with light on
X1000Y1000D02*	Linear move with light off
X1500D01*	Linear move with light on
X2000Y1500*	Notice since D01 is modal it does not need to be repeated
X2500D02*	Notice since the X & Y commands are modal, Y is not repeated
Y1000D01*	X is not repeated and uses its previous value of 2.500”
D02*	Light off no move
G54D11*	New aperture selected
G55X1000Y1000D03*	G55 prepares for flash It is not necessary. D03 is the flash command.
X2000D03*	Y value does not change
X2500D03*	This method reduces the size of the file
Y1500D03*	Here, X does not change from previous value
X2000D03*	Flash
G54D12*	New aperture select
X1000Y1500D03*	Move to (1.0, 1.5) and flash
G54D13*	New aperture select
X3000Y1500D03*	Move and flash
G54D14*	New aperture select
Y1250D03*	Move and flash
G54D15*	New aperture select
Y1000D03*	Move and flash

G54D10*	New aperture select
G01X3750Y1000D02*	Linear move, light off. Start point of the following arc command
G75*	Sets multi quadrant mode
G03X3750Y1000I250J0D01*	Move from start point above to end point drawing a complete circle
G54D16*	New aperture select
G55X3400Y1000D03*	Flash
G54D17*	New aperture select
G55X3500Y900D03*	Flash
G54D10*	New aperture select
G36*	Start Outline fill
G01X500Y2000D02*	
Y3750D01*	
X3750*	
Y2000*	
X500*	
X500Y2000D02*	
G37*	End Outline fill
G54D18*	New aperture select
G55X0Y3875D03*	Flash
X3875Y3875D03*	Flash
%LNXTEST2* LPC*%	Layer Name
G36*	Start Outline fill
G01X1000Y2500D02*	
Y3000D01*	
G74*	Single Quadrant mode
G02X1250Y3250I250J0D01*	Clockwise arc move with radius .25”
G01X3000*	Complete 90° arc
G75*	Sets multi quadrant mode
G02X3000Y2500I0J-375D01*	Clockwise arc move with radius .375”
G01X1000*	Linear move light on
X1000Y2500D02*	Linear move light off
G37*	End Outline fill
%LNXTEST3*%	Layer Name
%LPD*%	Layer Polarity Dark
G54D10*	New aperture select
X1500Y2875D02*	
X2000D01*	
D02*	

G54D11*	
X1500Y2875D03*	
X2000D03*	
G54D19*	New aperture select
X2875Y2875D03*	
M02*	End of Program