Headings

Files Format:

1 - Schematic files format:

Schematic Files Format:

1.1 - Units

Sizes and coordinates are given in mils (1/1000 inch)

1.2 - Header

Format :

EESchema Schematic File Version 1
LIBS: *libraries list* (not used, for information only).
EELAYER *nn mm* (*nn mm* not used, reserved)
EELAYER END
$Descr Sheet size *dimx dimy* (sheet size = A4..A0 ou A..E)
Title block description (Texts of the title block)
$EndDescr

```
EESchema Schematic Spins Version 1
LIBS:brooktre, cypress, ttl, power, linear, memory, xilinx, idiot, aaci, INTEL, special, device, dsp
EELAYER 20 0
EELAYER END
$Descr A3 16535 11700
Sheet 1 4
""
Date “28 DEC 1996”
Rev ""
Comp ""
Comment1 ""
Comment2 ""
Comment3 ""
Comment4 ""
$EndDescr
```

1.3 - Description of a component

Format:

$Comp
L name *reference*
U *N mm time_stamp*
P *posx posy*
List of fields:
F *field_number “text” orientation posX posY size Flags* (see below) *hjustify vjustify/italic/bold “name”*
1 posx posy (redundant: not used)

A B C B (orientation matrix with A, B, C, D = - 1, 0 or 1)

$EndComp

Description of the fields:

F n “text” orientation posx posy dimension flags hjustify vjustify/italic/bold “name”
with n = field number (reference field = 0, value field = 1, N = 0..11 or more)
orientation = H (horizontal) or V (vertical).

- n = field number :
  - reference = 0.
  - value = 1.
  - Pcb FootPrint = 2.
  - User doc link = 3. At present time: not used (reserved)
- n = 4..11 = fields 1 to 8 (since January 2009 more than 8 field allowed, so n can be > 11.
- text (delimited by double quotes)
- orientation = H (horizontal) or V (vertical).
- position X and Y
- dimension (default = 50)
- Flags: visibility = 0 (visible) or 1 (invisible)
- hjustify vjustify = L R C B or T
  - L= left
  - R = Right
  - C = centre
  - B = bottom
  - T = Top
- Style: Italic = I or N (since January 2009)
- Style Bold = B or N (since January 2009)
- Name of the field (delimited by double quotes) (only if it is not the default name)

Note: vjustify, Italic and Bold are in the same 3 chars word.
Example:

```
Comp
L CONN_3 JP3
U 1 1 329879E1
P 1200 2000
F 0 “JP3” H 1250 2200 60 0000 C CNN
F 1 “CONN_3” V 1350 2000 50 0000 C CNN
F 4 "example" H 8000 4350 60  0000 C CIB "myfield"
   1 1200 2000
  1 0 0 - 1
$EndComp
```

1.4 - Description of a NoConnect symbol

Format: **NoConn ~** *posx posy*
Example:

```
NoConn ~ 13400 5500
```

1.5 - Description of a hierarchical sheet symbol

Format:

$Sheet
S *posx posy dimx dimy*
List of Sheet Labels
$EndSheet

Format of Sheet Labels

Fn “text” forms side posx posy dimension
With:

n = sequence number (0..x).
n = 0: name of the corresponding schematic file.
n = 1: name of the sheet of hierarchy.
form = I (input) O (output) B (BiDi) T (tri state) U (unspecified)
side = R (right) , L (left)., T (tpo) , B (bottom)

Example:

```
$Sheet
S 1800 1600 1500 1500
F0 “PROGALIM.SCH” 60
F1 “PROGALIM.SCH” 60
F2 “CLK” O R 3300 1800 60
F3 “/RESET” O R 3300 2000 60
F4 “VPWR” O R 3300 2700 60
F5 “/HALT” O R 3300 2100 60
F6 “TRANSF1” I L 1800 1900 60
F7 “TRANSF2” I L 1800 2000 60
F8 “3.84MH” O R 3300 2200 60
$EndSheet
```

1.6 - Description of a text note

Format: **Text Notes** *posx posy orientation dimension ~*
Text
Example:

```
Text Notes 2100 3250 1 60 ~
TOTO
```

1.7 - Description of a Global Label

Format: **Text GLabel** *posx posy orientation dimension shape*
Text
Example:

```
Text GLabel 3100 2500 2 60 UnSpc
TITI
Text GLabel 3150 2700 1 60 3State
3STATES
Text GLabel 2750 2800 0 60 UnSpc
BIDI
Text GLabel 2750 2650 0 60 Output
GLABELOUT
Text GLabel 2750 2400 0 60 Input
RESET
```

1.8 - Description of a Hierarchical label

Format: **Text HLabel** *posx posy orientation dimension shape*
Text
Example:

```
Text HLabel 3400 2000 0 60 Input
/RESET
```

1.9 - Description of a label

Format: **Text Label** *posx posy orientation dimension ~*
Text
Example:

```
Text Label 3400 2000 0 60 ~
/RESET
```

1.10 - Description of a junction

Format: **Connection ~** *posx posy*
Example:

```
Connection ~ 13300 6500
```

1.11 - Description of a wire segment (Wire)

Format:

Wire Wire Line

startx starty endx endy

Example:

```
Wire Wire Line
3300 1800 3900 1800
```

1.12 - Description of a Bus segment

Format:

Wire Bus Line

startx starty endx endy

Example:

```
Wire Bus Line
3900 5300 4500 5300
```

1.13 - Description of a dotted line segment

Format:

Wire Notes Line

startx starty endx endy

Example:

```
Wire Notes Line
2850 3350 2850 3050
```

1.14 - Description of a bus entry

Format:

- For an entry wire/bus :

  Wire Wire Bus

  startx starty endx endy

- For an entry bus/bus :

  Wire Bus Bus

  startx starty endx endy

Example:

```
Entry Wire Bus
4100 2300 4200 2400
Entry Bus Bus
4400 2600 4500 2700
```

1.15 - Description of a Bitmap Image

Bitmaps are considered to be 300x300 pixels per inch.

A scaling factor is applied by Eeschema to adjust the actual bitmap size on screen.

Format:

$Bitmap

Pos *posx posy*

Scale *scale value (float). This is the user scalin factor used to display the bitmap.*

Data

Bitmap data , PNG format, in hexadecimal.

Each byte is coded by 2 hexadecimal digits.

Bytes are separated by a space.

EndData

$EndBitmap

Example:

```
$Bitmap
Pos 7450 5600
Scale 1,000000
Data
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 01 00 00 00 01 00 08 06 00 00 00 5C 72 A8
66 00 00 00 04 73 42 49 54 08 08 08 08 7C 08 64 88 00 00 00 09 70 48 59 73 00 00 1B 58 00 00 1B
….
EndData
$EndBitmap
```

2 - Schematic Libraries Files Format:

Schematic Libraries Files Format:

2.1 - Units

Sizes and coordinates are given in mils (1/1000 inch)

2.2 - Heading

format:

```
EESchema-LIBRARY Version 2.0 24/1/1997-18:9:6
description of the components
# End Library
```

2.3 - Description of a component

The format is as follows :

DEF *name reference unused text_offset draw_pinnumber draw_pinname unit_count units_locked option_flag*
ALIAS *name1 name2…*
fields list
DRAW
list graphic elements and pins
ENDDRAW
ENDDEF

*Parameters for **DEF** :*

- ***name*** = component name in library (74LS02 ...)
- ***référence*** = Reference (U, R, IC .., which become U3, U8, R1, R45, IC4...)
- ***unused*** = 0 (reserved)
- **t*ext_offset*** = offset for pin name position
- ***draw_pinnumber*** = Y (display pin number) ou N (do not display pin number).
- ***draw_pinname*** = Y (display pin name) ou N (do not display pin name).
- ***unit_count*** = Number of part (or section) in a component package.
- ***units_locked*** = = L (units are not identical and cannot be swapped) or F (units are identical and therefore can be swapped) (Used only if unit_count > 1)
- ***option_flag*** = N (normal) or P (component type "power")

Example:

```
DEF BNC P 0 40 Y NR 1 L NR
F0 “P” 10.120 60 H V L C
F1 “BNC” 110 - 60 40 V V L C
```

```
DRAW
C 0 0 70 0 1 0
C 0 0 20 0 1 0
X Ext. 2 0 - 200 130 U 40 40 1 1 P
X In 1 - 150 0.130 R 40 40 1 1 P
ENDDRAW
ENDDEF
```

2.3.1 - Description of Alias

This line exists only if the component has alias names.

format:

ALIAS *name1 name2 name3…*

2.3.2 - Description of the fields

format:

F n "*text" posx posy dimension orientation visibility hjustify vjustify/italic/bold "name"*

with:

- n = field number :
  - reference = 0.
  - value = 1.
  - Pcb FootPrint = 2.
  - User doc link = 3. At present time: not used
- n = 4..11 = fields 1 to 8 (since January 2009 more than 8 field allowed, so n can be > 11.
- text (delimited by double quotes)
- position X and Y
- dimension (default = 50)
- orientation = H (horizontal) or V (vertical).
- Visibility = V (visible) or I (invisible)
- hjustify vjustify = L R C B or T
  - L= left
  - R = Right
  - C = centre
  - B = bottom
  - T = Top
- Style: Italic = I or N (since January 2009)
- Style Bold = B or N (since January 2009)
- Name of the field (delimited by double quotes) (only if it is not the default name)

Note: vjustify, Italic and Bold are in the same 3 chars word.

Example:

```
DEF DIODE D 0 40 Y NR 1 0 NR
F0 "D" 0.100 50 H V L CNN
F1 "DIODE" 0 -100 50 H V L CIB
F5 "2euros" 0 -200 50 H V L CIB "PRICE"
```

2.3.2.1 - Important Note 1:

The **F1** field is the default component value **and** the component name in library.
So the F1 field text should be the same as the name.

2.3.2.2 - Important Note 2:

F0 is the reference prefix.
If the prefix starts b **#** (like **#U**) the component is not output to netlist or Bill Of Material.
This is a "virtual" component.
Mainly power symbols must have the prefix starting by **#**.

2.3.3 - Description of graphic elements

There are of 5 types:

- Polygon (succession of segments), filled or normal.
- Rectangle.
- Circle.
- Arc of circle.
- Text.

2.3.3.1 - Polygon :

Format:

P *Nb parts convert ltrait x0 y0 x1 y1 xi yi cc*

With:

- Nb = a number of points.
- unit = 0 if common to the parts; if not, number of part (1. .n).
- convert = 0 if common to the 2 representations, if not 1 or 2.
- *trait* = line thickness.
- xi yi coordinates of end i.
- cc = N F or F (F = filled polygon; f = . filled polygon, transparent background)

Example:

```
P 3 0 1 0 - 50 50 50 0 - 50 - 50 F
P 2 0 1 0 50 50 50 – 50 N
```

2.3.3.2 - Rectangle

Format:

S *startx starty endx endy unit convert ltrait cc*

With

- unit = 0 if common to the parts; if not, number of part (1. .n).
- convert = 0 if common to the representations; if not, 1 or 2.
- ltrait = thickness.
- cc = N F or F (F = filled Rectangle,; f = . filled Rectangle, transparent background)

Example:

```
S 0 50.900.900 0 1 0 f
```

2.3.3.3 - Circle

Format:

C *posx posy radius unit convert ltrait cc*

With

- unit = 0 if common to the parts; if not, number of part (1. .n).
- convert = 0 so common to the representations, if not 1 or 2.
- ltrait = thickness.
- cc = N F or F (F = filled Rectangle,; f = . filled Rectangle, transparent background)

Example:

```
C 0 0 70 0 1 0 F
C 0 0 20 0 1 0 N
```

2.3.3.4 - Arc of circle

Format:

With *posx posy radius start end part convert ltrait start_pointX start_pointY end_pointX end_pointY cc*

With:

- start = angle of the starting point (in 0,1 degrees).
- end = angle of the end point (in 0,1 degrees).
- unit = 0 so common to the parts; if not, number of part (1. .n).
- convert = 0 if common to the representations, if not 1 or 2.
- ltrait = thickness.
- start_pointX start_pointY = coord of the starting point (role similar to start)
- end_pointX end_pointY = coord of the point of arrival (role similar to end)
- cc = N F or F (F = filled Rectangle,; f = . filled Rectangle, transparent background)

Example:

```
To 0.148 48 - 889 889 0 1 0 N
To 0 51 51 - 889 889 0 1 0 N
```

2.3.3.5 - Text field

Format:

T orientation *posx posy dimension unit convert Text*

With:

- orientation = horizontal orientation (=0) or vertical (=1).
- type = always 0.
- unit = 0 so common to the parts, if not number of part (1. .n).
- convert = 0 if common to the representations, if not 1 or 2.

Example:

```
T 0 - 320 - 10 100 0 0 1 VREF
```

2.3.4 - Description of pins

Format:

X *name number posx posy length orientation Snum Snom unit convert Etype [shape]*.

With:

- orientation = U (up) D (down) R (right) L (left).
- name = name (without space) of the pin. if ~: no name
- number = n pin number (4 characters maximum).
- length = pin length.
- Snum = pin number text size.
- Snom = pin name text size.
- unit = 0 if common to the parts; if not, number of part (1. .n).
- convert = 0 if common to the representations, if not 1 or 2.
- Etype = electric type (1 character)
- shape = if present: pin shape (clock, inversion…).

Example:

```
X TO 1 - 200 0.150 R 40 40 1 1 P
X K 2.200 0.150 L 40 40 1 1 P
X 0 1 0 0 0 R 40 40 1 1 W NC
X ~ 2 0 - 250 200 U 40 40 1 1 P
```

Etype list:

INPUT	**I**
OUTPUT	**O**
BIDI	**B**
TRISTATE	**T**
PASSIVE	**P**
UNSPECIFIED	**U**
POWER INPUT	**W**
POWER OUTPUT	**w**
OPEN COLLECTOR	**C**
OPEN EMITTER	**E**
NOT CONNECTED	**N**

Shape list:

- If invisible pin, the shape identifier starts by **N**
- Next character is:

Line	None (default)
Inverted	**I**
Clock	**C**
Inverted clock	**CI**
Input low	**L**
Clock low	**CL**
Output low	**V**
Falling edge clock	**F**
Non Logic	**X**

Example:
A clock is coded **C** if visible, and **NC** if invisible.

3 - Board File Format

Board Files Format:

3.1 - General Informations:

- Board file (*.brd files) are in ASCII format.
- Dimensions are in 1/10000 inch, except for the page size (in 1/1000 inch).

First line is something as:

PCBNEW-BOARD Version 0 date 5/1/2005-14:45:23

All the following descriptions are like this:

$DESCRIPTION
some data
...
$endDESCRIPTION

Example:

$GENERAL
Ly 1FFF8001
Links 66
NoConn 0
Di 24940 20675 73708 40323
Ndraw 16
Ntrack 267
Nzone 1929
Nmodule 29
Nnets 26
$EndGENERAL

$SHEETDESCR
Sheet A4 11700 8267
Title ""
Date "23 feb 2004"
Rev ""
Comp ""
Comment1 ""
Comment2 ""
Comment3 ""
Comment4 ""
$EndSHEETDESCR

3.2 - Layer numbering:

Tracks and other items (texts, drawings ...) use one layer.
Pads and vias use several layers.
There are 16 copper layers and 13 technical layers.
The *layer* parameter used in descriptions has the value:

value	layer name	
0	Copper layer	"Copper" layers
1 to 14	Inner layers	
15	Component layer	
16	Copper side adhesive layer	Technical layers
17	Component side adhesive layer	
18	Copper side Solder paste layer	
19	Component Solder paste layer	
20	Copper side Silk screen layer	
21	Component Silk screen layer	
22	Copper side Solder mask layer	
23	Component Solder mask layer	
24	Draw layer (Used for general drawings)	
25	Comment layer (Other layer used for general drawings)	
26	ECO1 layer (Other layer used for general drawings)	
27	ECO2 layer (Other layer used for general drawings)	
28	Edge layer. Items on Edge layer **are seen on all layers**	
29	Not yet used	
30	Not yet used	
31	Not yet used	

Mask layer:

Sometimes, a *mask layer* parameter is used.
It is a 32 bits mask used to indicate a layer group usage (0 up to 32 layers).
A *mask layer* parameter is given in *hexadecimal form*.
Bit 0 is the copper layer, bit 1 is the inner 1 layer, and so on...(Bit 27 is the Edge layer).
Mask layer is the ORed mask of the used layers

3.3 - First line of description:

Format:

PCBNEW-BOARD Version <version number> date <date>-<time>
Date and time are useful only for information (not used by pcbnew).

3.4 - $GENERAL

This data is useful only when loading file.
It is used by Pcbnew for displaying activity when loading data.

$GENERAL	Start description
Ly 1FFF8001	Obsolete (used for old pcbnew compatibility)
Links 66	Total number of connections
NoConn 0	Remaining connections
Di 24940 20675 73708 40323	Bounding box coordinates: X_start Y_start X_end Y_end
Ndraw 16	Number of draw items like eged segments, texts...
Ntrack 267	Number of track segments
Nzone 1929	Number of zone segments
Nmodule 29	Number of modulss
Nnets 26	Number of nets
$EndGENERAL	End description

3.5 - $SHEETDESCR

This the page size and texts.

$SHEETDESCR	Start description
Sheet A4 11700 8267	<Page size> X_size Y_size **in mils (1/1000 inch)**
Title ""	Title text
Date "23 feb 2004"	Date text
Rev ""	Revision text
Comp ""	Company name text
Comment1 ""	Comment text, line 1
Comment2 ""	Comment text, line 2
Comment3 ""	Comment text, line 3
Comment4 ""	Comment text, line 4
$EndSHEETDESCR	End description

3.6 - $SETUP block:

This data bock is used for design settings
This is useful only for board edition.
Example:
$SETUP
InternalUnit 0.000100 INCH
Layers 2
Layer[0] Cuivre signal
Layer[15] Composant signal
TrackWidth 250
TrackWidthHistory 25
TrackWidthHistory 170
TrackWidthHistory 250
TrackClearence 110
ZoneClearence 150
DrawSegmWidth 150
EdgeSegmWidth 50
ViaSize 600
ViaDrill 250

ViaSizeHistory 600
MicroViaSize 200
MicroViaDrill 80
MicroViasAllowed 0
TextPcbWidth 170
TextPcbSize 600 800
EdgeModWidth 150
TextModSize 600 600
TextModWidth 120
PadSize 1500 2500
PadDrill 1200
AuxiliaryAxisOrg 29500 55500
$EndSETUP

$SETUP	Start block "SETUP"
InternalUnit 0.000100 INCH	Internal unit for Pcbnew, all coordinates are in this unit
Layers 2	Number of layers (2 = double sided board) must be 1 to 16
Layer[0] Cuivre signal	layer name and type name = name given to the layer by the user (here: "cuivre" type = signal (not current used in Pcbnew)
Layer[15] Composant signal	
TrackWidth 250	Current track width
TrackWidthHistory 170	Last used track widths
TrackWidthHistory 250	
TrackWidthHistory 400	
TrackClearence 100	Isolation for DRC (Design rules check)
ZoneClearence 200	Isolation used in zone filling
DrawSegmWidth 120	Current segment width for drawings on technical layers
EdgeSegmWidth 120	Current segment width for drawings on "edge layer"
ViaSize 700	Current via size
ViaDrill 250	Via drill for this board
ViaSizeHistory 450	Last used via sizes
ViaSizeHistory 650	
ViaSizeHistory 700	
TextPcbWidth 120	Current text width for texts on copper or technical layers. This is not for text on footprints
TextPcbSize 600 600	Current text X Y size
EdgeModWidth 120	Current Segment width for footprint edition
TextModSize 120 600	Current text XY size for texts for footprint edition
TextModWidth 120	Current text width for texts for footprint edition
PadSize 700 700	Current X Y pad size (footprint edition)
PadDrill 320	Current pad drill
AuxiliaryAxisOrg 0 0	Auxiliary axis position (Auxiliary axis is the reference coordinate (0 0 coordinate) for EXCELLON drilling files
$EndSETUP	End block "SETUP"

3.7 - $EQUIPOT

$EQUIPOT describes a net name.

$EQUIPOT	Start block
Na 2 "N-000026"	**Na** <internal net number> « net name »
St ~	
$EndEQUIPOT	End block

Note1:

Internal net number is an arbitrary number.
It is computed by Pcbnew when compiling netlist.

Note2:

Net 0 is not a real net.
Net 0 is the net number used internally by Pcbnew for all the no connected pads.

Example:
$EQUIPOT;
Na 0 ""
St ~
$EndEQUIPOT$EQUIPOT
Na 1 "DONE"
St ~
$EndEQUIPOT
$EQUIPOT
Na 2 "N-000026"
St ~
$EndEQUIPOT
$EQUIPOT
Na 3 "TD0/PROG"
St ~
$EndEQUIPOT

3.8 - $MODULE

Description =start by:
$MODULE <module name>
And ends with
$EndMODULE <module name>

Module description has four sections:

1. General description (fixed size)
2. Field description (variable size)
3. Drawing description (variable size)
4. Pad description. (variable size)
5. 3D shape informations.

Note:

All coordinates are relative to the module position.
Its means the coordinates of segments, pads, texts ... are given for a module in position 0, rotation 0.
If a module is rotated or mirrored, real coordinates must be computed according to the real position and rotation.

3.8.1 - General description:

$MODULE bornier6	**$MODULE** <module lib name>
Po 62000 30500 2700 15 3EC0C28A 3EBF830C ~~	**Po** Xpos Ypos Orientation(0.1deg) Layer TimeStamp Attribut1Attribut2 Attribut1 = ~or 'F' for autoplace (F = Fixed, ~= moveable) Attribut2 = ~or 'P' for autoplace (P = autoplaced)

$MODULE bornier6	**$MODULE** <module lib name>
Li bornier6	**Li** <module lib name>
Cd Bornier d'alimentation 4 pins	**Cd** comment description (displayed when browsing libraries)
Kw DEV	**Kw** Keyword1 Keyword2 ... (for footprint selection by keywords)
Sc 3EBF830C	**Sc** TimeStampOp
Op 0 0 0	**Op** <rotation cost 90 deg> <rotation cost 180 deg> for auto place. rotation cost = 0 (no rotation allowed) to 10 (null cost)

Note:

Usually, components are on layer 15 (*component layer*) or 0 (*copper layer*).
If the component is on layer 0, it is"mirrored". The "mirror axis is the X axis

3.8.2 - Field Description:

There are 2 to 12 fields
Field 0 = component reference (U1, R5 ...) (required)
Field 1 = component value (10K, 74LS02 ...) (required)
Other fields (optional) are comments.

Format:

T<field number> <Xpos> <Ypos> <Xsize> <Ysize> <rotation> <penWidth> N <visible> <layer> "text"

Field	***Units***	***Meaning***
field number	enumeration	0=>reference, 1=>value, etc.
Xpos	tenths of mils (.0001 inches)	The horizontal offset relative to the module's overall position
Ypos	tenths of mils (.0001 inches)	The vertical offset relative to the module's overall position
Xsize	tenths of mils (.0001 inches)	The horizontal size of the character 'M'
Ysize	tenths of mils (.0001 inches)	The vertical size of the character 'M'
rotation	tenths of degrees	Angular rotation from horizontal, counterclockwise
penWidth	tenths of mils (.0001 inches)	Width of the pen used to draw characters
N	none	flag for the parser?
visible	boolean	I=> invisible, V=> visible
layer	enumeration	see layer numbers above

Examples:

T0 500 -3000 1030 629 2700 120 N V 21 "P1"	T0 => reference
T1 0 3000 1201 825 2700 120 N V 21 "CONN_6"	T1 => value

3.8.3 - Drawings:

Tells how to draw module shape.
Drawings are segment, circle, arc.

DS -6000 -1500 -6000 1500 120 21	DS is a **D**raw **S**egment DS Xstart Ystart Xend Yend Width Layer
DS 6000 1500 6000 -1500 120 21	An other Draw Segment

Other Drawings are:

DC ox oy fx fy w	DC is a **D**raw **C**ircle DC Xcentre Ycentre Xpoint Ypoint Width Layer
DA x0 y0 x1 y1 angle width layer	DA is a **D**raw **A**rc X0,y0 = Start point x1,y1 = end point

3.8.4 - Pad Descriptions:

All the pads of this footprint are listed here (Many $PAD/$EndPAD sections here)..
See $PAD description.

3.8.5 - $SHAPE3D

3D shape informations:
The real shape description is a vrml file, build by ***Wings3d***.
This shape can be scaled, moved and rotated.
This is because a single 3D shape can be used for many footprints (for instance, we use the shape resistor.wrl for several resistor footprints, by tuning the X, Y, Z scale of the 3D shape according to the different size of resistor footprints).
Some smd footprints are using this feature.
For the same reasons, the 3D shape can be moved (by the move factor) and/or rotated.

Real shape unit is 0.1 inch (1 unit vrml = 0.1 inch = 2.54 millimeter).

An other reason exists: when a footprint is very big (a big connector) or very small (a small SMD resistor) we must create a 3D shape small or bigger than real size, in order to use easily the 3D modeler.

$SHAPE3D	Start description
Na "device/bornier_6.wrl"	*File**Na**me* (default path is kicad/modules/packages3d/)
Sc 1.000000 1.000000 1.000000	X Y Z ***s**cale **f**actor*
Of 0.000000 0.000000 0.000000	X Y Z ***off**set (move vector, in 3D units (0.1 inch))*
Ro 0.000000 0.000000 0.000000	X Y Z ***ro**tation (in degree)*
$EndSHAPE3D	End description

The 3D shape coordinates are relative to the footprint coordinates.
The 3D shape must be scale, moved and rotated according to the parameters Sc Of and Ro,
and after moved and rotated according to the footprint coordinates and rotation.
If the footprint is « inverted » (that is, located on copper side) the 3D shape must be « inverted » too.

Note:

A footprint may have several 3D shapes (for instance an integrated circuit and his socket).

3.8.6 - $PAD

Pads have different shapes and attributes.

Pad shapes are:

Circle.
Oblong(or oval).
Rectangular (Square is like a rectangle).
Trapeze.

Pad attributes are:

- Normal (Has usually a hole)
- Smd (used for Surface Mounted Devices). Has no hole.
- Connector (used for connectors like a PC Board Bus connector)
- Mechanical. (Like a hole for mechanical use)

And shape can be draw with an offset related to the drilling hole.

The hole shale is round or oblong

$PAD	Start description
Sh "2" C 1500 1500 0 0 2700	***Sh**ape*: <pad name> shape Xsize Ysize Xdelta Ydelta Orientation
Dr 600 0 0 or (oblong hole) Dr 600 0 0 O 600 650	***Dr**ill* <Pad drill> Xoffset Yoffset (round hole) or (oblong hole) ***Dr**ill* <Pad drill.x> Xoffset Yoffset <Hole shape> <Pad drill.x> <Pad drill.y>
At STD N 00E0FFFF	***At**tributs*: <Pad type> N <layer mask>
Ne 8 "GND"	***Ne**t reference of the pad:* <netnumber> <net name>
Po -3000 0	X_pos Y_pos (relative to the module position)
$EndPAD	End description

Note:
<Pad type> is the Pad Attribute. It is one of: "STD" "SMD" "CONN" "HOLE" "MECA".
Shape is one of:

- C (circle)
- R (Rectangular).
- O (Oblong)
- T (Trapèze)

Hole shape = O (O for ***O**blong*)

Example:
$PAD
Sh "3" C 1500 1500 0 0 2700
Dr 600 0 0
At STD N 00E0FFFF
Ne 10 "TD0_1"
Po -1000 0
$EndPAD

3.9 - Graphic items:

There are drawing items like segments, circles, texts, targets and cotations.

3.9.1 - $DRAWSEGMENT

Draw segments are :

- segments (strait line)
- circles
- arcs

3.9.1.1 - Line:

$DRAWSEGMENT	Start description
Po 0 67500 39000 65500 39000 120	***Po**sition* shape Xstart Ystart Xend Yend width
De 28 0 900 0 0	***De**scription* layer type angle timestamp status
$EndDRAWSEGMENT	End description

Note:

- shape = 0
- Angle is used only for arc segments (unused for line, left for compatibility).

3.9.1.2 - Circle:

$DRAWSEGMENT	Start description
Po 1 67500 39000 65500 39000 120	***Po**sition* shape Xcentre Ycentre Xend Yend width
De 28 0 900 0 0	***De**scription* layer type angle timestamp status
$EndDRAWSEGMENT	End description

Note:

- shape = 1
- Angle is used only for arc segments (unused for circle, left for compatibility).

- End is a point of this circle. (If Xend or Yend is 0, the other coordinate is the radius)

3.9.2 - Arc:

$DRAWSEGMENT	Start description
Po 2 67500 39000 65500 39000 120	***Po**sition* shape Xstart Ystart Xend Yend width
De 28 0 900 0 0	***De**scription* layer type angle timestamp status
$EndDRAWSEGMENT	End description

Note:

- shape = 2
- *start* and *end* are the 2 points of the arc. *angle* is the arc angle (in 0.1 degree). Center coordinates are computed by pcbnew from *start*, *end* and *angle*.

Currently, only 90 degrees arcs are supported.(thereby, angle = 900)

Example:

$DRAWSEGMENT
Po 0 67500 34000 67500 39000 120
De 28 0 900 0
$EndDRAWSEGMENT

3.9.3 - $TEXTPCB

Example: 

$TEXTPCB	Start description
Te "TDI"	***Te**xt* "string"
Po 57250 35750 600 600 150 0	***Po**sition* Xstart Ystart Xsize Ysize Width rotation
De 15 1 B98C Normal	***De**scription* layer normal timestamp style normal = 0 : text is mirrored. normal = 1 : text is normal. style = Normal or Italic
$EndTEXTPCB	End description

***Example**:*

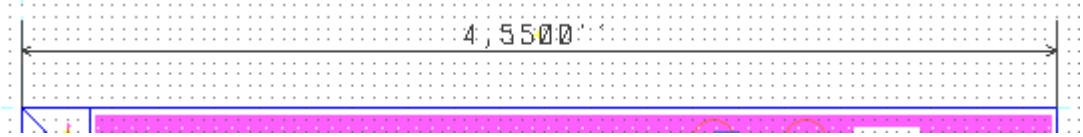
$TEXTPCB
Te "TCK"
Po 57250 33500 600 600 150 0
De 15 1 B98C Normal
$EndTEXTPCB

3.9.4 - $MIRE

	shape 1
	shape 0

$MIREPCB	Start description
Po 0 28 28000 51000 5000 150 00000000	***Po**sition* shape Xpos Ypos size width timestamp
$EndMIREPCB	End description

3.9.5 - $COTATION



$COTATION	Start description
Ge 0 24 0	***Ge**neral* shape layer timestamp currently, shape = 0.
Te "4,5500""	***Te**xt* "string" string is the cotation value in inches or millimeters
Po 50250 5791 600 800 170 0 1	***Po**sition (for text)* Xpos Ypos Xsize Ysize width orient normal
Sb 0 27500 6501 73000 6501 150	Coordinates of segments (axis, arrows...)
Sd 0 73000 9000 73000 5081 150	
Sg 0 27500 9000 27500 5081 150	
S1 0 73000 6501 72557 6731 150	
S2 0 73000 6501 72557 6271 150	
S3 0 27500 6501 27943 6731 150	
S4 0 27500 6501 27943 6271 150	
$EndCOTATION	End description

3.10 - Track, vias and Zone section:

3.10.1 - $TRACK

Track section decribes tracks and vias on copper layers.
Each track (or via) has a two line description:
For a track segment:
***Po**sition* shape Xstart Ystart Xend Yend width
***De**scription* layer 0 netcode timestamp status
Shape parameter is set to 0 (reserved for future changes).

For a via:

***Po**sition* shape Xstart Ystart Xend Yend diameter
***De**scription* layer 1 netcode timestamp status
For a via, layer parameter gives :
On the 4 less significant bits: the starting layer of the via
On the 4 next bits: the ending layer.
For instance, a via starting at copper kayer (layer 0) end ending at component layer (layer 15 has the layer parametre set to F0 hexadecimal or 240 decimal.
Shape parameter is the via type (*through = 3, blind = 2, buried = 1*)
Timestamp parameters are set to 0 (reserved for future changes).
Status parameter can be set to 0 (Used internally for routing infos)..

$TRACK	Start description
Po 0 36750 37000 36550 37000 250	***Po**sition* shape Xstart Ystart Xend Yend width width = diameter for a via
De 15 0 1 0 400	***De**scription* layer type netcode timestamp status type = 0 for a track segment. type = 1 for a via
Po 0 39000 36750 38750 37000 250	An other track
De 15 0 1 0 0	
Po 3 53500 27000 53500 27000 650	This is a via (via "through") from layer 15 (component) to layer 0 (copper)
De 15 1 14 0 0	
$EndTRACK	End description

3.10.2 - $ZONE

Zone section is like track section. (There is no via in Zone section).
It is used to handle a zone filling, from a zone outline.

$ZONE	Start description
Po 0 67100 33700 67100 38600 100	Same as track description
De 0 0 2 3EDDB09D 0	
$EndZONE	End description

3.10.3 - $CZONE_OUTLINE

Describes the main outlines of a zone and the outlines of filled areas (solid polygons) inside the zone main outlines.
Outlines of filled areas can be missing (if the zone is not currently filled)
Because a zone handles thermal reliefs, there are options to describe pads in zones options and thermal reliefs parameters.
Example:

```
$CZONE_OUTLINE
ZInfo 47868246 1 "GND"
ZLayer 0
ZAux 4 E
ZClearance 150 T
ZMinThickness 190
ZOptions 0 32 F 200 200
ZCorner 74750 51750 0
ZCorner 74750 13250 0
ZCorner 29750 13250 0
ZCorner 29750 51750 1
....
$POLYSCORNERS
74655 51655 0 0
74655 13345 0 0
...
$endPOLYSCORNERS
$endCZONE_OUTLINE
```

$CZONE_OUTLINE	Start description
ZInfo 478E3FC8 1 "/aux_sheet/INPUT"	<Time stamp> <internal netcode> "net name"
ZLayer 0	Layer (0 = copper, 15 = component, 1 ..14 = inner layers)
ZAux 4 E	<corners count> <zone hatching option> zone hatching option = N (none), E (edge hatching) or F (full hatching)
ZClearance 200 T	<Zone clearance> <pads option = I, T or X> I = pads in zone T = Thermal reliefs X = pads not in zone.
ZMinThickness 190	<Zone min thickness (for copper zone)>
ZOptions 0 32 F 200 200	<fill mode> <arc approx> <antipad thickness> <thermal stubs width> fill mode = 0 (use solid polygons) or 1 (use segments) arc approx = 16 or 32 (segments count to approximate a 360 arc)
ZCorner 49450 19150 0	First corner (external outline)
ZCorner 40600 19150 0	Next corner
ZCorner 40600 22850 0	

ZCorner 49450 22850 1	End corner (flag = 1)
$POLYSCORNERS	Start of filled areas outlines
74655 51655 0 0	First corner (first filled area outline)
74655 13345 0 0	Next corner
$endPOLYSCORNERS	
$endCZONE_OUTLINE	End description

Other example:

$CZONE_OUTLINE	Start description of an other outline
ZInfo 47B3E800 3 "VCC"	
ZLayer 1	
ZAux 8 F	
ZClearance 200 T	
ZMinThickness 190	Zone min thickness (for copper zone)
ZOptions 0 32 F 200 200	
ZCorner 49704 23032 0	First corner (external outline)
ZCorner 49704 18940 0	
ZCorner 46140 19024 0	
ZCorner 46148 20000 0	
ZCorner 45250 20000 0	
ZCorner 44750 21250 0	
ZCorner 43750 22250 0	
ZCorner 46176 23068 1	End corner (flag = 1)
ZCorner 48450 19900 0	First corner (this is a hole)
ZCorner 48450 20800 0	
ZCorner 47350 20800 0	
ZCorner 47250 19900 1	End corner (flag = 1)
$endCZONE_OUTLINE	End description

3.11 - $EndBOARD

$EndBOARD terminates the whole board description.
Must be the last line.