

Kicad/file formats

The latest reviewed version was checked on 12 March 2019. There is 1 pending change awaiting review.

File formats

KiCad creates and uses files of several different formats.^[1]

- Files that end in ".sch" are schematics.
- Files that end in ".lib" are schematic symbols library files.
- Files that end in "-cache.lib" are schematic symbols library files, too. This file is a local copy of symbols used in the current project the file is named for.
- Files that end in ".pro" are project files
- Files that end in ".dcm" add documentation to symbols in the library file with the same name. The ".dcm" file contains the description, keywords and docfilename whereas the ".lib" file contains information about how the symbol is drawn, the pins *et cetera*.
- Files that end in ".000", ".bak", ".bck" are old backup files (don't archive them).
- Files that end in ".brd" are PCB layout files.
- Files that end in ".cmp" are footprint information files that are modified by the PCBNew program
- Files that end in ".erc" are output from the schematic electronic rules check (ERC).
- Files that end in ".gcd" ...
- Files that end in ".lst", ".net" are netlist output from the schematic (don't archive them).
- Files that end in ".kicad_mod", typically in folders with names that end in ".pretty", are the 2014(?) version of modules (a KiCad "module" is called a "footprint" or a "decal" in other CAD software), one footprint per file, lots of files in the entire ".pretty" library.
- Files that end in ".mod" are module libraries (a KiCad "module" is called a "footprint" or a "decal" in other CAD software)
- Files that end in ".mdc" cache a short summary of a few frequently-referenced bits of data from the corresponding ".mod" file of the same name (don't archive them).
- Files that end in ".dsn" are regenerated from the ".kicad_pcb" file every time you hit the "autoroute" button and then hit the "Export a Spectra Design (*.dsn) file" (don't archive them).
- Files that end in ".ses" are session files output from the autorouter (don't archive them).
- The ".git" folder contains data for revision control. (If you use ".git" with KiCad,

you may want to use a ".ignore" file based on <https://github.com/github/gitignore/blob/master/KiCad.gitignore>).

Some people are working on making it very easy for people, when they make improvements to the KiCad footprint libraries and schematic symbol libraries, to push any improvements to GitHub and automatically pull any improvements other people have made from GitHub. Some such libraries include:

- <http://github.com/KiCad> (the repositories that end in ".pretty" are footprint libraries; please submit a pull request if you can make them better)
- <https://github.com/hairymnstr/ndkicadlibrary>

Schematic Files Format

Units

Sizes and coordinates are given in whole numbers of thousandths of an inch (1/1000 inch). Coordinates may be negative by prefixing a hyphen (-) to the numeric value. Note that Y coordinates are positive in a downward direction with respect to the page origin.

Angles are given in whole numbers of tenths of degrees (1/10°), specifying a rotation counter-clockwise.

Examples

| Value | Distance | Angle |
|-------|---------------|---------------------------------|
| 1 | 0.001 inches | 0.1° counter-clockwise |
| 200 | 0.200 inches | 20.0° counter-clockwise |
| 3599 | 3.599 inches | 359.9° counter-clockwise |
| -1234 | -1.234 inches | <i>invalid (negative value)</i> |
| 36000 | 36.000 inches | <i>invalid (over 359.9°)</i> |

Header

| Syntax | Description | Version |
|---|---|---------|
| EESchema Schematic File Version ver [date] | ver is 1 or 2 date is only present in some versions of Version 2 files? | |
| LIBS: library_list | not used, for information only | |
| EELAYER nn mm | nn and mm are not used, reserved | |
| EELAYER END | | |
| \$Descr size w h | size = A4..A0 or A..E | 1 |
| | size = A4..A0, A..E, or User. w = width (mils), h = height (mils). w and h are ignored unless size = User | 2 |
| Sheet m n | m is the current sheet number, n is the total number of sheets. It appears that a sheet will not appear in the project list unless it is m = 1. | |
| "string" | Title field | 1 |
| Title "string" | Title field | 2 |
| Date "string" | Issue Date field | |
| Rev "string" | Revision field | |
| Comp "string" | Company field | |
| Comment1 "string" | Comment1 field | |
| Comment2 "string" | Comment2 field | |
| Comment3 "string" | Comment3 field | |
| Comment4 "string" | Comment4 field | |
| \$EndDescr | | |

Example of Version 1

```
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
```

A later example:

```
EESchema Schematic File Version 2  date 4/15/2011 3:59:54 PM
LIBS:mylib
LIBS:transistors
LIBS:someotherlib
EELAYER 25 0
EELAYER END
$Descr A4 11700 8267
Sheet 1 1
Title "DC Supply"
Date "15 apr 2011"
Rev "1"
Comp "Circuits R Us"
Comment1 ""
Comment2 ""
Comment3 ""
Comment4 ""
$EndDescr
```

Example from KiCad version 4.0.6:

```

EESchema Schematic File Version 2
LIBS:74xgxx
LIBS:74xx
LIBS:ac-dc
EELAYER 25 0
EELAYER END
$Descr A4 11693 8268
encoding utf-8
Sheet 1 1
Title ""
Date ""
Rev ""
Comp ""
Comment1 ""
Comment2 ""
Comment3 ""
Comment4 ""
$EndDescr

```

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 hor_justify style* <"*field_name*">(see below)

1 posx posy (redundant: not used (hmm... used in 4.0.6 maybe P seems not to be used. Best to keep these in sync))

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

\$EndComp

Description of the fields:

F n "*text*" *orientation posX posY size flags hor_justify style* <"*field_name*">

with n = field_number (reference_field = 0, value_field = 1, footprint_field = 2, datasheet_field = 3, user_defined_fields = 4..12)

orientation = H (horizontal) or V (vertical).

posX posY = text position in mils

size = character size in mils (0,001")

flags = abcd

a=
b=
c=
d= Visibility 0=Visible 1=Invisible

hor_justify = L (left), C (center), R (right)

style = xyz

x=vertical justify [T (top), C (center), B (bottom)
y=text_style_1 N (Normal), I (cursive)
z=text_style_2 N (normal), B (bold)

field_name = only used for user defined fields (field_number > 4)

Example:

```

$Comp
L CONN 3 JP3
U 1 1 329879E1
P 1200 2000
F 0 "JP3" H 1250 2200 60 0000 L CNN
F 1 "CONN 3" V 1350 2000 50 0000 L CNN
F 2 "" H 1450 1800 60 0000 C CNN
F 3 "" H 1550 1600 60 0000 C CNN
F 4 "20%" H 1650 1400 60 0000 C CNN "Tolerance"
    1 1200 2000
    - 1 0 0 - 1
$EndComp

```

Description of a NoConnect symbol

Format:

NoConn ~ posx posy

Example:

```
NoConn ~ 13400 5500
```

Description of a hierarchical sheet symbol

Format:

\$Sheet

S posx posy dimx dimy

List of Sheet Labels

\$EndSheet

Format of Sheet Labels:

F_n "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)

side = R (right) or L (left).

Example:

```

$Sheet
S 1800 1600 1500 1500
F0 "PROGALIM.SCH" 60
F1 "PROGALIM.SCH" 60
F2 "CLK" 0 R 3300 1800 60
F3 "/RESET" 0 R 3300 2000 60
F4 "VPWR" 0 R 3300 2700 60
F5 "/HALT" 0 R 3300 2100 60
F6 "TRANSF1" I L 1800 1900 60
F7 "TRANSF2" I L 1800 2000 60
F8 "3.84MH" 0 R 3300 2200 60
$EndSheet

```

Description of a text note

Format:

Text Notes posx posy orientation dimension ~ *Text*

Example:

Text Notes 2100 3250 1 60 ~
TOTO

Description of a Global Label

Format:

Text GLabel posx posy orientation dimension shape

Text

Example:

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

Description of a Hierarchical label

Format:

Text HLabel *posx posy orientation dimension shape*

Text

Example:

Text HLabel 3400 2000 0 60 Input
/RESET

Description of a label

Format:

Text Label *posx posy orientation dimension ~*

Text

Example:

Text Label 3400 2000 0 60 ~
/RESET

Description of a junction

Format:

Connection ~ *posx posy*

Example:

Connection ~ 13300 6500

Description of a wire segment (Wire)

Format:

Wire Wire Line

startx starty endx endy

Example:

Wire Wire Line
3300 1800 3900 1800

Description of a Bus segment

Format:

Wire Bus Line

startx starty endx endy

Example:

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

Description of a dotted line segment

Format:

Wire Notes Line

startx starty endx endy

Example:

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

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:

Wire/Bus:

```
Entry Wire Bus
4100 2300 4200 2400
```

Bus/Bus:

```
Entry Bus Bus
4400 2600 4500 2700
```

Schematic Libraries Files Format

Units

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

Headings

Format:

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

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

F0 reference posx posy text_size text_orient visibility htext_justify vtext_justify

F1 name posx posy text_size text_orient visibility htext_justify vtext_justify

F2 ???

F3 ???

\$FPLIST

footprint list

\$ENDFPLIST

ALIAS name1 name2 name3 fields list

DRAW

list graphic elements and pins

ENDDRAW

ENDDEF

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
```

Description of DEF

This is the component definition line.

Format:

DEF name reference unused text_offset draw_pinnumber draw_pinname unit_count units_locked option_flag

- name = component name in library (74LS02 ...), write insert preceding '~' in front of the name, in the case of it does not have any unit in sch library. the preceding '~' has to be ignored when reading the name.
- reference = Reference (U, R, IC ..., which become U3, U8, R1, R45, IC4...)
- unused = 0 (reserved)
- text_offset = offset for pin name position
- draw_pinnumber = Y (display pin number) or N (do not display pin number).
- draw_pinname = Y (display pin name) or N (do not display pin name).
- unit_count = Number of part (or section) in a component package. Limit is 26 (shown as chars form A to Z).
- 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")

Description of F0 and F1

F0 is the component reference line. F1 is the component name line.

Format:

F0 *reference posx posy text_size text_orient visible htext_justify vtext_justify*

F1 *name posx posy text_size text_orient visibility htext_justify vtext_justify*

- reference = Reference (U, R, IC ..., which become U3, U8, R1, R45, IC4...)
- name = component name in library (74LS02 ...)
- posx, posy = position of the text label
- text_size = Size of the displayed text
- text_orient = Displayed text orientation (V=Vertical, H=Horizontal(default))
- visible = Is label displayed (I=Invisible, V=Visible(default))
- htext_justify = Horizontal text justify (L=Left, R=Right, C=Centre(default))
- vtext_justify = Vertical text justify (T=Top, B=Bottom, C=Centre(default))

Description of \$FPLIST

This line exists if one or more footprints are specified. Footprint names can have wildcards.

Description of ALIAS

This line exists only if the component has alias names.

Format:

ALIAS *name1 name2 name3...*

Description of DRAW

Lists graphic elements and pins. Each line defines a single element. The line starts with a single character indicating the type e.g. P indicates a polygon. The following items are commonly used in some of the elements:

- posx, posy = Position of the graphic element
- unit = unit no. in case of multiple units
- convert = In case of variations in shape for units, each variation has a number. 0 indicates no variations. For example, an inverter may have two variations - one with the bubble on the input and one on the output.
- thickness = line thickness
- fill = fill colour (F=filled with foreground colour, f=filled with background colour, N=Not filled(default))

A record (Arc)

A *posx posy radius start_angle end_angle unit convert thickness fill startx starty endx endy*

- posx, posy = centre of the circle part of which is the arc
- radius = radius of the lost arc
- start_angle = start angle of the arc in tenths of degrees
- end_angle = end angle of the arc in tenths of degrees
- startx, starty = coordiantes of the start of the arc
- endx, endy = coordinates of the end of the arc

C record (Circle)

C *posx posy radius unit convert thickness fill*

- posx, posy = centre of the circle
- radius = radius of the circle

P record (Polyline)

The polyline has a series of points. It need not described a closed shape i.e. a polygon. To do this make the first pair the same as the last pair.

P *point_count unit convert thickness (posx posy)* fill*

- point_count = no. of coordinate pairs. posx and posy are repaeated these many times.

S record (Rectangle)

S startx starty endx endy unit convert thickness fill

- startx, starty = Starting corner of the rectangle
- endx, endy = End corner of the rectangle

T record (Text)

T direction posx posy text_size text_type unit convert text text_italic text_hjustify text_vjustify

- direction = Direction of text(0=Horizontal, 90=Vertical(default))
- text_size = Size of the text
- text_type = ???
- text = Text to be displayed. ~~All characters are replaced with spaces.~~ in case having one or more spaces in the text, double quote enclosed like "some thing".
- text_italic = "Italic" or "Normal"
- text_bold = 0 to normal 1 to bold
- text_hjustify = C(Center), L(Left) or R(Right)
- text_vjustify = C(Center), B(Bottom) or T(Top)

X record (Pin)

X name num posx posy length direction name_text_size num_text_size unit convert electrical_type [pin_type]

- name = name displayed on the pin
- num = pin no. displayed on the pin
- posx = Position X same units as the length
- posy = Position Y same units as the length
- length = length of pin
- direction = R for Right, L for left, U for Up, D for Down
- name_text_size = Text size for the pin name
- num_text_size = Text size for the pin number
- unit_num = Unit number reference (see REF 'unit_count')
- convert = (0 if common to the representations, if not 1 or 2)
- electrical_type = Elec. Type of pin (I=Input, O=Output, B=Bidi, T=tristate,P=Passive, U=Unspecified, W=Power In, w=Power Out, C=Open Collector, E=Open Emitter, N=Not Connected)
- [pin_type] = Type of pin or "Graphic Style" (N=Not Visible, I=Invert (hollow circle), C=Clock, IC=Inverted Clock, L=Low In (IEEE), CL=Clock Low, V=Low Out (IEEE), F=Falling Edge, NX=Non Logic). Optional : when not specified uses "Line" graphic style.

Board File Format

Note : this section describes the "old" .brd file format (file version 1 or 2).

General Information

Layer numbering

0. Back - Solder

1. Inner_back

2. Inner_front

3. Inner

5. Inner

6. Inner

7. Inner

8. Inner

9. Inner

- 10. Inner
- 11. Inner
- 12. Inner
- 13. Inner
- 14. Inner
- 15. Front - Component
- 16. Adhesive/glue Back
- 17. Adhesive/glue Front
- 18. Solder Paste Back
- 19. Solder Paste Front
- 20. SilkScreen Back
- 21. SilkScreen Front
- 22. SolderMask Back
- 23. SolderMask Front
- 24. Drawings
- 25. Comments
- 26. ECO1
- 27. ECO2
- 28. Edge Cuts

First line of description

\$GENERAL

\$SHEETDESCR

\$SETUP block

\$EQUIPOT

\$MODULE

General description

Field Description

Drawings

All physical units are in mils (1/1000th inch) unless otherwise noted. The default layer number for graphic segments is 21, which corresponds to SilkS_Front.

DS *x1 y1 x2 y2 width layer*

Draws a line segment from (x1, y1) to (x2, y2) with width *width* on the *layer* number specified.

DC *x1 y1 x2 y2 width layer*

Draws a circle whose center is (x1, y1), and whose radius is specified by the segment (x1, y1) - (x2, y2) with line width *width* on the *layer* number specified.

DA *x1 y1 x2 y2 angle width layer*

Draws a circular arc. Center is at (x1, y1). The arc's starting point is (x2, y2). The length of the arc sweeps clockwise (for positive angles) from here by the number of degrees specified by (*angle* / 10).

T*type x y height width angle stroke layer mirror visible layer italic "Text"*

Draws the text *Text* as either reference text (*type*=0), value text (*type*=1), or user text (*type*=2) at position (x, y), rotated counterclockwise (*angle* / 10) degrees, on layer number *layer*. Each character will be *height* high, *width* wide, and strokes will be *stroke* thick. Text will be mirrored (*mirror*=M) or not (*mirror*=N), italic (*italic*=I) or not (*italic*=N), and visible by default (*visible*=V) or invisible by default (*visible*=I).

Pad Descriptions

\$SHAPE3D

\$PAD

A pad is usually a copper area for electrical connection to an electrical component. It has an optional hole for through-hole components, or may be defined as an area on a single copper layer for surface-mount components. It can also be used as a thermal connection for heat distribution, or as a hole for mounting or other uses.

Sh "*padNum*" *shape xSize ySize yBaseIncrease xBaseIncrease angle*

Defines the pad's dominant shape. *padNum* defines the pad number. The shape of the pad (*shape*) can be circular (**C**), ovate (**O**), rectangular (**R**), or trapezoidal (**T**) with its size specified by *xSize* and *ySize*. (Note that for circular pad shapes *xSize* and *ySize* must be equal.) The pad is rotated at an angle of *angle*. For trapezoidal shapes, *yBaseIncrease* specifies how much taller the pad's left edge is from its right, and *xBaseIncrease* specifies how much wider the pad's bottom is from its top; *xSize* and *ySize* then specify the size of the pad at its center and the trapezoidal effect increases one edge and decreases the other.

Dr *dia xOffset yOffset*

Defines the pad's drilled hole offset from the pad's position by (*xOffset*,*yOffset*) with a diameter of *dia*. To specify no hole, specify *dia* as **0**. Note that the drilled hole can be located offset from the center of the pad's shape (**Sh**) although pcbnew requires the drilled hole be located on the pad itself.

At *type flag layers*

Defines the pad's attributes. The pad type is specified by *type* and can be **STD** for a standard pad with a hole, **SMD** for a surface-mount pad, **CONN** for a connector, or **HOLE** for a hole. *flag* is **N** (unknown function). *layers* specifies the active layers as a 32-bit hexadecimal number with leading zeroes such that active layers are indicated by a 1 bit and inactive by 0.

Ne *unknown "netName"*

Defines the net name as *netName*. There are other options specified by the *unknown* flag: it appears that *unknown* is 0 in a module library and has a numeric value other than 0 when placed and connected in a board file.

Po *x y*

Defines the position of the pad as (x,y). This is the point that traces must terminate for pcbnew to confirm connection to a pad.

Graphic items

\$DRAWSEGMENT

Comes in Po, De pairs. Example:

```
Po 0 73000 59250 63250 59250 150
De 0 0 900 0 0
```

Po *function x1 y1 x2 y2 width*

De *?? ?? ?*

Line

In a Po line, function is 0.

Circle

In a Po line, function is 1. (x1,y1) defines the center of the circle and (x2,y2) is a point on the circumference.

Arc

In a Po line, function is 2. (x1,y1) defines the center of the circle for the arc. (x2,y2) is the start point for a 90° clockwise arc.

Arc**\$TEXTPCB**

Te "text"

Defines *text* as the string to render.

nl "newLineText"

If present after **Te**, renders *newLineText* on the line after *text*. This can be repeated multiple times for multiple new lines. It is also common to have no **nl** entries if *text* is to fit on one line only.

Po x y height width thickness angle

Defines the position of the text as (x,y) with a height of *height*, a width of *width*, a thickness of *thickness*, and an angle of *angle*. Although the user interface only supports *angles* of 0, 900, 1800, and 2700, other angles can be entered in the board file.

De layerNum mirror 0 style

Defines the options for the text. The text is rendered on layer *layerNum*. If *mirror* is 1, the text is rendered normal; if it is 0 it is mirrored. Setting *style* to **Normal** renders the text normal and **Italic** renders the text *italic*.

\$MIRE**\$COTATION****Track, vias and Zone section**

\$TRACK

Comes in **Po**, **De** pairs. Example:

```
Po 0 38900 95200 39500 95800 80 -1
De 0 0 1 0 80000
```

Po function x1 y1 x2 y2 width ?

- *function* must be 0 (only straight line segments, no arcs or circles as in **\$DRAWSEGMENT**)

De layer ? net ? flags

flags bitfield: Unknown length (probably 32 bit), printed as hex with leading 0's truncated.

Binary: ???? ???? a??? ???? ???? ???? ???? ???? ?

a = Autorouted Flag

l = (Segment?) Locked Flag

Example: (open a new .brd file, add a track, save it - and then insert/replace the changes below in a text editor; no nets are assigned)

(*FIXME: add an example of "(kicad_pcb (version 3))."*)

```
PCBNEW-BOARD Version 1 date 2012-03-18T07:15:54 CET
# Created by Pcbnew(2010-00-09 BZR 23xx)-stable
$GENERAL
LayerCount 6
Ly 1FFF801F
EnabledLayers 1FFF801F
....
$TRACK
# gray track (Inner4 - jumper layer):
Po 0 32000 25250 32000 23250 80 -1
De 3 0 0 0 0
#
# red track (front layer):
Po 0 24250 10750 24250 25250 80 -1
De 15 0 0 0 0
#
# green track (back layer):
Po 0 24250 25250 30000 25250 80 -1
De 0 0 0 0 0
#
# via between red and green track:
Po 3 24250 25250 24250 25250 350 -1
De 15 1 0 0 0
#
$EndTRACK
...
```

\$ZONE

\$CZONE_OUTLINE

\$EndBOARD

Historical notes

Version 3 and later

As of 2013, the PCBnew application creates ".kicad_pcb" files that begin with "(kicad_pcb (version 3))", and for files from KiCad 4.0.x, "(kicad_pcb (version 4))".

All distances are in millimeters. If the distance is not an integer number of millimeters, the distance will be indicated with a decimal point. For example, the vast majority of PCBs have a board thickness of 1/16 inch \approx 1.6 mm,^[2] so Version 3/4 board files often have a line

```
(thickness 1.6002)
```

(The internal PCBnew unit of length is now an integer multiple of 1 nanometer, which enables representation of metric units and imperial units down to 1/100 mil = 1/100,000 inch.^{[3][4]})

Version 2

Earlier versions of the PCBnew application create ".brd" files that begin with "PCBNEW-BOARD Version 2". Such files typically have a line:

```
Units mm
```

indicating that all distances are in millimeters. If the distance is not an integer number of millimeters, the distance will be indicated with a decimal point. For example, the vast majority of PCBs have a board thickness of 1/16 inch \approx 1.6 mm,^[2] so Version 2 board files often have a line:

```
BoardThickness 1.6002
```

Version 1

The earliest (?) versions of the PCBnew application created ".brd" files that begin with "PCBNEW-BOARD Version 1". Such files have all distances in integer multiples of some tiny reference unit. Typically such files have a line:

```
InternalUnit 0.000100 INCH
```

indicating that all distances are in integer multiples of 1/10,000 of an inch, which was once the internal PCBnew unit of length.^[5] For example, the vast majority of PCBs have a board thickness of 1/16 inch \approx 0.063 inch,^[2] so Version 1 board files often have a line:

BoardThickness 630

1. "KiCad File Formats" (<http://kicad-pcb.org/help/file-formats/>).
2. [Practical Electronics/PCB Layout#Board Thickness and Layers](#)
3. "KiCad: Internal unit system" (<https://blueprints.launchpad.net/kicad/+spec/unit-system>).
4. "All dimensions are stored as integer nanometers."--"Pcbnew reference manual" (<http://docs.kicad-pcb.org/stable/en/pcbnew.html>).
5. "KiCad: Convert Internal CPB Units to 1nm" (<https://blueprints.launchpad.net/kicad/+spec/internal-units-to-1nm>).

Retrieved from "https://en.wikibooks.org/w/index.php?title=Kicad/file_formats&oldid=3559980"

This page was last edited on 26 July 2019, at 16:28.

Text is available under the [Creative Commons Attribution-ShareAlike License](#).; additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#).