KiCad Librarian

Manage footprint and symbol libraries
for the KiCad EDA suite

User Guide & Reference

Acknowledgements

The KiCad Librarian uses the wxWidgets toolkit, by Julian Smart, Robert Roebling et al.

The "Haru PDF Library" is copyright Takeshi Kanno. The KiCad Librarian uses it for the reports that it generates.

The "newstroke" font was created by Vladimir Uryvaev, and converted to CXF format by Brian Bidulock.

The logo of the KiCad Librarian is from icons8, see http://www.icons8.com.

Copyright & license

Contents

Introduction

The KiCad Librarian allows you to view footprints and schematic symbols in KiCad libraries, and move or copy footprints/symbols from one library to another. It allows you to scan all libraries in a list, so that you can check for duplicates or naming conflicts. It supports a repository for central storage (and sharing) of symbols and footprints.
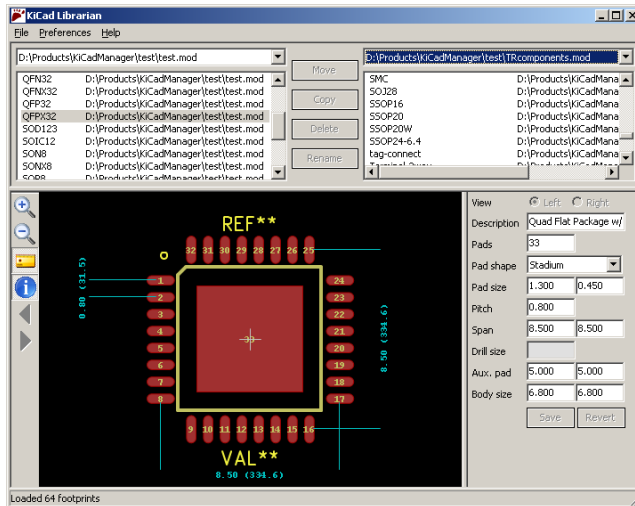
Parameters for a footprint, such as pad sizes, shapes and pitch, can be adjusted by changing values in a property table. This works for new footprints (created from one of several templates) and for *existing* footprints —even those that were not initially created by the Librarian. If a template is present, the Librarian also creates (or adjusts) a 3D model for the package.

With the goal of consistency in a library of footprints, the Librarian calculates essential metrics of pad sizes, spans and pitch. It provides a comparison mode where you can overlay two footprints for a visual comparison of the parameters. (A visual comparison of schematic symbols is also supported.)

For symbols, the pin types, shapes and positions can be adjusted. In the same way as for footprints, you can create new symbols from a template. Pin numbers and descriptions can be imported from the clipboard and/or conveniently edited in a table.

Additionally, the KiCad Librarian can create a report of all parts in a library. The reports contain the essential statistics of the parts. For footprints, this includes a drawing of the footprint and the pitch and pad size parameters; for symbols, it includes alias names and the list of applicable footprints.

With KiCad alone, you can do all of the above, too (except for the reports). This Librarian was developed to make managing the libraries more convenient. For example, instead of needing to "export" a symbol from one library and then to "import" it into the other, the KiCad Librarian allows you to open two libraries at once, and to copy the symbol from one into the other in a single operation. Another example: if you want to make the pads of a 48-pin QFP a tenth of a millimetre narrower, you need only to adjust a single value to change all 48 pads at once.

Footprint library management now also means footprint conversion. KiCad footprints exist in three formats, and all three formats are used interchangeably. When moving a footprint from one library to another, the footprint data may need to be converted. The Librarian does this automatically while copying or moving footprints.

Disclaimer of warranty

The KiCad Librarian is currently considered of "beta" quality. There is no warranty of any kind.

It is advised that you keep your KiCad libraries in a revision control system, so that earlier releases can always be restored. If you currently do not use a revision control system, it is *strongly* advised that you keep back-up files of your KiCad libraries.

Toward a high-quality library

A good footprint/symbol library is worth money —or more accurately, a bad library can cost you dearly. Yet, no library that we are aware of, is both comprehensive and of good quality throughout. What we set out to do is to assemble a library by collecting the best parts of available libraries — combined with our designs and corrections. The Librarian helps with managing the collection and moving footprints into and out of the core collection.

Even when collecting the best footprints from different libraries (or perhaps, *especially* when collecting footprints from different libraries), you may want to adjust the footprints. For example, TSSOP packages with and without "exposed pad" should have the same sizes for the normal pads. While the KiCad Librarian is *not* a footprint editor, it *does* provide a convenient interface to make common adjustments to footprints and schematic symbols. Then, too, moving and copying footprints across libraries is a first step —modifications to footprints should be made only to your private libraries, not to the libraries of the standard distribution. So, first copy, then adjust.
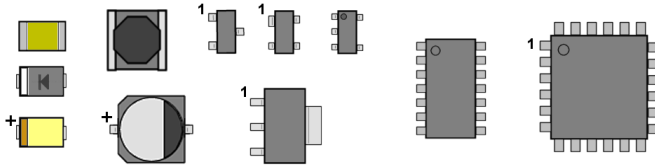
FIGURE 1: *Normal orientations of components, see appendix A*

We advise that footprints adhere to the IPC-7351 standard.[1] A footprint should conform to the pin numbering of this standard. We also recommend that the "normal orientation" of the footprint conforms to the standard as well. See appendix A for detailed information and examples for the normal orientations.

An error to avoid, for example, are the footprints "SOT23EBC", "SOT23_INV" and "SOT23-SPECIAL" in the default library of KiCad: each of these is the SOT23 footprint, but with various permutations of the pin numbers. *Expect confusion*, when the pin numbers in a components data sheet no longer match the pin numbers in the design... The recommended procedure is to create multiple *schematic* symbols, like "NPN-BEC" and "NPN-EBC", and map both of these to a *single* SOT23-3 footprint. The difference in pin labelling is then recorded in the schematic symbols —and it will match the data sheets for the components.

In your schematic, you are unlikely to have components marked as NPN-EBC —instead, you will specify a particular type of transistor, such as "BC817". You can map BC817 to NPN-EBC by adding an *alias*. With aliases properly set up, you can add a BC817 to your schematic, to get the appropriate shape and pin numbering, which in turn maps correctly to the single SOT23-3 footprint.

For polarized components, KiCad uses a pin numbering which is the opposite of the one in IPC-7351. The IPC-7351 standardizes pin 1 to be the cathode on diodes and the "+" pole of capacitors. The footprints in the KiCad libraries have the polarity marker on pin 2. For assembly facilities (board fabrication houses) that adhere to IPC-7351, this is another source of confusion. We advise to adjust the respective footprints and schematic symbols.

A component has a "centroid position", which is the centre of mass. A pick-&-place machine needs to pick up the component at the centroid position, for best assurance that the component will not slide or rotate on the nozzle. For most components, the centroid position is also the centre of the component.

---

[1] The IPC-7351 conforms to the IEC 61188-7 "Level A" standard.

The KiCad Librarian displays the centroid position, but currently does not allow you to edit it.

Pads for surface-mount components are traditionally rectangular, but due to surface tension, solder does not reflow well to the into the corners on small pads. Especially on the narrow pads of fine-pitch components, better *wetting* is achieved on pads with an obround shape.

Many EDA suites keep the schematic symbol and the associated footprint as a unit: the footprint is linked with the schematic symbol. In KiCad, though, that link is made with the `CvPcb` program, per project —it is not in the libraries. The advantage of the procedure used by KiCad is that it avoids redundancy: a chip resistor is available in many packages; sizes from 0402 to 1206 are all commonly used, but ceramic capacitors and ferrite beads also use these footprints. KiCad allows you to have a single schematic symbol for a resistor, another symbol for a capacitor and one more for a ferrite bead, plus a single footprint for the 0603 package.

On the other hand, though integrated circuits are often available in multiple packages, the pin lay-out may be different for each package. In those cases, the choice of symbol mandates a particular footprint. KiCad uses a "footprint filter" to help mapping symbols to the correct footprint. You may optionally assign to each schematic symbol, a list of footprints that is appropriate for the symbol. Each name in the list may also contain the wild-card characters "*" and "?". If a 16-pin part is available in SSOP, SOIC, and DIL packages, the footprint filter could be set to "*SSOP16 SOIC16 DIL16", where the "*" wild-card makes sure that the TSSOP shape is also matched.[2]

---

[2] Beware, though, that the SSOP and TSSOP footprints are not identical: SSOP has a pin pitch of 0.635mm and TSSOP has a pitch of 0.65mm.

Configuring the Librarian

Before you can use the KiCad Librarian, you will need to configure it. On first launch, you will at least need to configure the paths where the KiCad libraries can be found.

The KiCad Librarian stores its settings in an "INI" file. By default, this INI file is in the home directory of the current user. For the purpose of running the Librarian as a "portable app" (for example, directly from an USB stick without requiring installation or configuration), the Librarian allows you to move the INI file to the main directory of where the application is installed itself. If the Librarian finds the INI file in that location on start-up, it will use that INI file (instead of writing in the user's home directory). For example, if the KiCad Librarian is installed in /Tools/KiCadLibrarian (meaning that the executable file is in /Tools/KiCadLibrarian/bin, the Librarian will look for the file /Tools/KiCadLibrarian/KiCadLibrarian.ini first.

Search paths

The Librarian collects data from module libraries that it finds. The paths that the Librarian looks in, for these libraries, must first be configured. See the dialog under Preferences / Search paths.

In contrast to earlier releases, the Librarian *does not* recurse into the subdirectories of each directory that you add. If you have your libraries arranged in a tree with subdirectories, each of these subdirectories should be added to the search path. There are separate search paths for footprint and for symbol libraries.

When the paths are set, the two drop-down lists are filled with all libraries that are collected from the search paths. Both the left and right drop-down lists contain the same set of libraries.

Application settings

To configure the user interface, open the dialog under Preferences / Application settings.

The font size and the length of the dimension lines, are options for the the text and dimensions drawn into the symbols or footprints. The length of the dimension lines is the minimum length of the lines for indicating the pitch or span dimensions in a footprint. Since the dimensions (as text) are typeset between the line end-points, the text is at the given distance from the pads

as well. The most appropriate offset depends on personal preference, but also on the zoom level.

Most footprints include texts for the reference designator and the component value, to be printed on the silk screen. These texts, the *value* and *reference* labels, may be optionally hidden in the footprint viewport. To hide the labels, remove the checkmark in the option Draw footprint/symbol labels.

The symbol/footprint lists show the library name for each part as well, which is particularly useful when the Librarian shows a combined list of modules from *all* collected libraries. You can choose whether to display only the base filename of the library, or its full path and filename. The latter is needed when two libraries with the same name exist in different locations.

To help create the enclosure for a PCB design, it is convenient to be able to extract a 3D model from the EDA suite. This requires that a 3D model is available for each module in the footprint libraries. KiCad optionally links a VRML file to a footprint. When moving/copying footprints across libraries, you can choose whether to copy the associated VRML files as well.

Since several footprints from various libraries may reference to the same VRML file, the KiCad Librarian never deletes VRML files. So when moving a footprint from one library to another, the associated VRML file is copied to the target path, but it is not removed from the source path.

The KiCad Librarian can modify parameters of symbols and footprints. It is able to do so for both those parts that it created itself (the Librarian includes a generator to create symbols and footprints from templates), as well as for existing parts. However, it uses a different approach for each type: when editing existing (non-generated) parts, the KiCad Librarian is conservative in what it changes and it will not alter or delete information that it does not manage itself. When editing generated parts, though, the Librarian may decide to simply rebuild the part from the template. If you have modified the generated part in the KiCad module editor, you may loose those changes through the rebuild. For this reason, the function to rebuild parts may be disabled.

Connecting to a repository

For the operation of the KiCad Librarian, it is not needed to configure a repository. However, configuring a repository will allow you to share symbols and footprints between different users and/or different locations.

The first step is to sign up for an account. A repository may allow read-only access without an account, but for write access, an account is always

needed. To sign up to a repository, open the dialog below Preferences / Repository and click on the button Sign up at the bottom row.

In this second dialog, fill in the required information. The URL to the repository, the user name and the email address are *required*. If the repository is on a server that requires user authentication, the user name and the password *for the server* are needed as well. The user name for the server need not be the same as the user name for the repository. If you are connecting to the server via HTTPS, you may need to disable the option to verify the certificate in the "chain of trust" of the root CAs. Especially in the case of self-signed certificates, the certificate will not validate against the root CA. Any certificate also contains the name of the host for which the certificate was issued. In some cases, you may need to disable the check whether the name in the certificate matches the host name.

Viewing and comparing

The KiCad Librarian always displays the selected footprint or symbol. To view a footprint/symbol, the first step is to select a library from one of the drop-down lists. There is no difference between the left and the right library lists —both contain the same set of libraries. There are three special items at the top of the drop-down lists: "(None)", "(All)" and "(Repository)". When selecting "(None)" for either side, the respective library is closed and the part list is cleared. When selecting "(All)", the Librarian collects all parts (of all *local* libraries) into a single list. The option "(Repository)" lists all parts available in a remote repository. The repository must be configured before it can be used, see .

After choosing a library (or "(All)"), the list below it is filled with all symbols in that library. When you select one of these symbols, it will be shown in the viewport below the lists. There is no difference between the left and the right symbol lists either. You can select a footprint from either one. In standard mode, only a single footprint is shown at any time. To view two footprints together, see section "comparing footprints".

You can zoom in and out with two buttons in the toolbar at the left of the viewport. Alternatively, you can use the key combinations Ctrl + and Ctrl - to zoom in and out. You can pan the view port by dragging it with the left mouse button pressed. A double-click centres the viewport.

The footprint/symbol preview is created completely independently from the KiCad code. The Librarian aims at being accurate in drawing the pads of a footprint. However, the (silk-screen) text on a footprint is considered non-critical; the Librarian does not necessarily show you the information in *exactly* the same way that KiCad does. Also note that the texts for reference designators and the component value can be disabled —see the user-interface configuration on .

In *footprint mode*, see , the viewport shows a cross in the centre. This is the origin of the footprint. It is also used as the centroid position of the component on pick-&-place machines. For most surface-mount components, the cross should be in the centre of the footprint.

The viewport displays some dimensions of the footprint, when in *footprint mode*. The dimensions are in millimetres, with the value in mil between parentheses. In the side panel (at the right of the view port), the dimensions are in millimetres only. Both the dimensions and the side panel can be individually enabled or disabled, through buttons in the toolbar (at the left of the viewport).

The dimensions are collected from the footprint —provided that the footprint adheres to a few conventions. For example, pads need to be numbered sequentially, from 1 to the total pad count. Pads that have a *name* rather than a number, are not taken into account in the calculations of the pitch and spans.

Librarian Mode

The KiCad Librarian can operate in two modes. While the user interface is largely the same for both modes, *Footprint mode* only operates on footprint libraries and *Schematic mode* only operates on symbol libraries.

To select the mode, select menu View, and then either of the options Footprint mode or Schematic mode.

Viewing synchronized lists

When a library is selected on both the left and the right sides, the lists with the symbol or footprint names may optionally be synchronized. When the lists are synchronized, both lists contain the same parts. However, parts that are not present in a library are printed in red and these cannot be viewed.

For example, assume that you have two footprint libraries opened on the left and right sides, and assume that only the library on the right side includes the SOT23-5 footprint. In synchronization mode, that part name will also be in the left part list, but in red. You can view the SOT23-5 footprint when you click on it in the right list, and you can copy or move it from right to left, but clicking on the SOT23-5 name in the left list does nothing.

Synchronization mode also synchronizes the scroll positions of the left and right lists, but only after a selection changes. That is, if you click on a part in one list, the other list scrolls so that it shows the same part name at the same position.

Comparing footprints

The KiCad Librarian allows you to overlay two footprints by selecting Compare mode from the View menu. When compare mode is active, you can select a footprint from the left and from the right list box and both will be shown.

When compare mode is active, two buttons on the toolbar allow you to show or hide the left & right footprints individually. If both footprints are set to

be visible (which is the default setting), they are drawn as semi-transparent overlays. The left & right footprints are coloured differently in compare mode.

The side panel contains the details of both footprints, but shows only one at a time. You can click on the buttons Left or Right to select the part of which you want to see the details.

You cannot move and copy footprints in compare mode; the buttons are disabled. The reason is that *both* libraries are active at the same time, and therefore the direction to copy or move is indeterminate.

You cannot change pad shapes or sizes (or other dimensions) in compare mode either. To change a footprint, you need to disable compare mode first.

Compare mode works in *Schematic mode* too.

Filtering the library

Especially for large libraries, it may be useful to reduce the list, so that it shows only the parts that match a keyword. To set a filter, select View / Filter… from the menu, or double-click in the right field of the status bar. The edit field for the filter appears in the status bar.

When changing the text of a filter, the edit field turns red. This signals that the modified filter is *not yet active*. When typing Enter in the edit field, the library lists are refreshed and the edit field for the filter turns white again.

To remove a filter, either clear the entire edit field and type Enter, or toggle the edit field off through the menu (View / Filter…).

The keyword in the filter is matched against the name of the part (symbol or footprint), the description of the part and the list of aliases. You can only filter on a single keyword.

Note that the application of a filter slows down the listing of the libraries, because each symbol or footprint has to be loaded individually to match it against the filter.

Moving, copying, deleting

To move or copy footprints/symbols, you need to open a library in both the left and the right lists. The source of the move or copy operation may be "(All)", but the destination must always be a single library.

Note that "compare mode" must be deselected, to move, copy, delete or rename items. See the menu View, option Compare mode (see also).

There is no "undo" operation for any of the file operations. Most operations can be reverted —when accidentally moving a footprint to the wrong library, you can move it to the correct one in a subsequent action. However, when deleting a footprint, the footprint is gone and cannot be recovered (unless you have kept a backup file or are using revision control).

In *footprint mode*, the copy/move/delete operations only copy/move/delete the footprints. The 3D drawing and the footprint documentation are not changed or moved. In *schematic mode*, the link to a data-sheet is copied (if present), but the data-sheet itself is not.

Note that when copying a footprint in s-expression format to a legacy library, some data may get lost. For example, the s-expressions library supports Bezier curves for outlines, but these cannot be stored in the legacy format.

Creating a new library

To create a new library, open the dialog File / New library. In the dialog, you can chose the name and path for the new library.

In *footprint mode*, you can choose between three formats for the library:
◇ a legacy format with units in decimil (tenth-of-a-mil, the true legacy format);
◇ a legacy format with units in millimetres;
◇ and the s-expression format.

In *schematic mode*, there is no choice for the file format; only a single file format for schematic symbol libraries is in use today.

When confirming the library name, a new empty library is created. You can then select that new (empty) library on the left or the right side, and copy footprints or symbols into it.

Changing parameters

Both footprints and symbols can be adjusted with the KiCad Librarian. For footprints, the available adjustments include the values for the pitch and pad spans, as well as pad sizes and shape. For schematic symbols, there are only a few parameters that can be edited, notably aliases for the component name and labels for the pins.

Changing footprint parameters

Many footprints have a regular, systematic lay-out: the pads all have the same size and they are aligned in two or four rows/columns with the pads being numbered in counter-clockwise order. For such shapes, the KiCad Librarian allows you to change these parameters.

The way to change pad and footprint parameters is to edit them in the side panel. Therefore, you must activate the side panel; see the menu View and then the item Details panel. Also make sure that "compare mode" is off (also in the View menu, and see page 9).

In the side panel, a field may have a light-grey background colour. This means that the field is read-only. A field may be read-only because it is irrelevant for the shape —for example, a "dual in-line" shape (DIL or DIP) has only two parallel "lines" of pads and therefore only *either* a horizontal span (between these lines) *or* a vertical span, but not both. Also when the footprint is irregular in one of its parameters, the respective fields become read-only. For example, if the footprint uses multiple pad sizes, the fields to set the pad size becomes non-changeable.

You cannot edit footprints or symbols directly in the repository. To change the parameters of a part in the repository, you must first copy it to a local library. You can copy the part back in the repository afterwards —if you have a suitable account for the repository.

See figure 2 for how the descriptions of the dimensions. The "width" parameters are in the horizontal direction and the "length" parameters are in the vertical direction. For a 2-pin package, the *body width* is therefore usually greater than its *body length*.

When you have entered a new value in a field, the footprint is redrawn to the new parameters, but the data is not yet saved. The background of changed (non-saved) fields is set to red. When clicking on the Save button at the bottom of the parameter list, the field colours are reset. There is no undo in editing one or more fields, but you can revert all changes since the last save, by clicking on the Revert button.
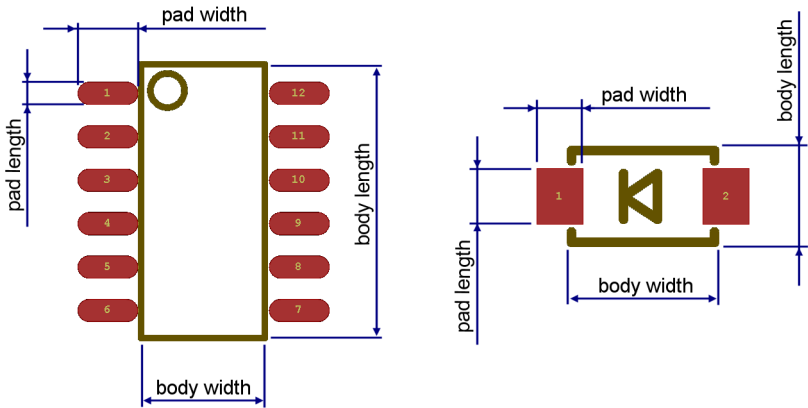
FIGURE 2: *Component dimensions)*

Note that when you set the drill size of the pads to zero, the pads will automatically toggle to SMD pads.

The pad shapes supported by the Librarian are the rectangle, circle, obround and a shape called "round + square". The "round + square" shape means that all pads are circular, except for the pad for pin 1, which will be square.

Pad size and span guidelines

Based on the IPC-7351 calculations for "nominal" density and land pattern recommendations in manufacturer data sheets (most of which comply with IPC-7351 nominal density), below is a set of guidelines for common packages.

Package	Note	Pad	Span
chip	0805 and smaller	0.5·body width × 1.2·body length	0.95·body width
	1206 and larger	0.35·body width × 1.1·body length	0.9·body width
MELF		0.3·body width × 1.15·diameter	0.95·body width
QFN	0.5 mm pitch	0.75 × 0.25 mm	body size + 0.15 mm
QFP	0.5 mm pitch	1.4 × 0.26 mm	body size + 1.4 mm
	0.65 mm pitch	1.4 × 0.35 mm	body size + 1.4 mm
	0.8 mm pitch	1.4 × 0.4 mm	body size + 1.4 mm
SOIC	1.27 mm pitch, 3.9 mm wide	1.6 × 0.7 mm	5.6 mm
	1.27 mm pitch, 7.5 mm wide	1.6 × 0.7 mm	9.6 mm
SOT23	3.0 × 1.5 mm, 3 pins	1.2 × 0.9 mm	2.4 mm
	3.0 × 1.5 mm, 5–6 pins	1.2 × 0.6 mm	2.4 mm

SOT3*3	2.0 × 1.25 mm, 3 pins	1.0 × 0.6 mm	2.0 mm
	2.0 × 1.25 mm, 5–6 pins	1.0 × 0.4 mm	2.0 mm
SSOP	0.635 mm pitch, 3.9 mm wide	1.3 × 0.35 mm	5.4 mm
sTSOP	0.5 mm pitch, 11.8 mm wide	1.4 × 0.26 mm	body size + 1.4 mm
TSOP-I	0.5 mm pitch, 18.4 mm wide	1.4 × 0.26 mm	19.8 mm
TSSOP	0.65 mm pitch, 4.4 mm wide	1.4 × 0.35 mm	5.6 mm
	0.65 mm pitch, 5.3 mm wide	1.4 × 0.35 mm	6.8 mm

Creating a new footprint

To create a new footprint with the KiCad Librarian, you first need to open a library. This will be the library that the new footprint is stored in. Footprints can only be created in a local library; to add a footprint in the repository, create it in a local library first and move it to the repository thereafter.

First open the dialog below File / New footprint. In the dialog, select a template for the footprint, and a name for the part. For example, when you wish to create a footprint for a surface mount linear power regulator, you may select the template "SOT" and type in the footprint name "SOT353". After confirming the selections in the dialog, a new footprint will be drawn in the viewport. The footprint is then already inserted into the library.

The initial footprint is drawn with the default parameters (as specified in the template itself). In most cases, these parameters will be incorrect. For example, the default parameters in the SOT template result in a (3-pin) SOT23 footprint. For a SOT353, the number of pins must be changed from 3 to 5 and the pad size, pitch and span must all be adjusted.

Note that for a grid array (PGA or BGA), the columns are numbered and rows are indicated with the letters A, B, C, D, E, F, G, H, J, K, L, M, N, P, R, T, U, V, W and Y —there are no I, O, Q, S, X and Z.

After adjusting the parameters and clicking Save, the new footprint is completed.

Footprints created by the Librarian can, of course, be edited by the KiCad module editor. With the module editor, you can fine-tune the footprint, for example to set a pad-specific solder-paste reduction. For the 3D model, you can adjust its height by changing the z-axis scaling factor in the KiCad module editor.

Conversely, footprints that have been edited by the KiCad module editor can still be adjusted with the Librarian. However, if you change the body size, any text or drawings that you have added in the module editor, will be lost, because the body is redrawn from the template. If you change the number of

pads, the footprint is completely rebuilt from the template, and any drawings or pad-specific settings that you have added in the module editor, will be lost. One some footprints, a redraw also happens when changing the pad size or pitch.

To avoid loosing edits that you made in the KiCad module editor, the rebuild of the footprint (partial or complete) from the template can be disabled in the options.

Changing symbol parameters

Discrete devices often share the same symbol. For example, a standard bi-junction NPN transistor commonly has the base on pin 2, emitter on pin 1 and collector on pin 3. In a schematic, all transistors that match this description, can be represented by one and the same schematic symbol.

KiCad supports *aliases* to map component names to generic symbols, or to give alternate names for specific symbols. For example, a generic shape, such as NPN-EBC may list a component like the BC817 as one of its aliases. When inserting a component in the schematic, you can then select the BC817, which KiCad will map to NPN-EBC. A single generic symbol may have potentially many aliases.

As explained in the section "Toward a high-quality library" on , the footprint filter allows you to match one or more footprints to a symbol. The CvPcb program from the KiCad suite can use these filters to present only the matching footprint(s). The footprint filter may contain wild-card characters, see for details.

Pin labels can be edited, for those pins that have a label. In the table, when the pin name is set to a "~", that pin is unnamed. When a pin name starts with a "~" (but more text follows), it indicates an inverted signal. The pin numbers can also be edited, for the purpose of re-numbering the pins. Only existing pins can be altered (pin label and/or pin number). The number of pins can only be adjusted when the symbol is based on a template; i.e. when the symbol was originally created by the KiCad Librarian (see the next section).

Pins can be re-arranged by selecting a row and pressing the key combination Alt+Up or Alt+Down. Pins can also be moved to another side of the component, by changing the field in the column side. These functions are only available for symbols that are based on a template.

The table with the pin information allows multiple selections (press Ctrl while clicking on the cells). A pop-up menu that appears on a right click then allows you to set the type, style or position of all selected pins to the same value.

After entering a new value in a field, the background of that field switches to red, to indicate that the adjusted data is not yet saved. Until you save the data, by clicking on the Save button at the bottom of the parameter list, the changes can be reverted by clicking on the Revert button.

Creating a new symbol

Similar to footprint creation, the generation of schematic symbols is based on templates. To create a new symbol, make sure that the Librarian is in schematic mode and that a library is selected. Then open the dialog below File / New symbol. In this dialog, select a template from the list, give the name of the symbol and optionally a prefix letter, and confirm with OK.

The symbol is initially drawn with default parameters. You can now adjust the symbol by setting the number of pins, the size of the outline, and the electrical types, labels and shapes of all pins.

Most templates allow you to put a pin to each of four sides: left, right, top or bottom. Some templates have additional sections. A communications line-driver IC may, for example, have galvanically separated input and output sections. The purpose of these sections is purely visual; the symbol is drawn with those sections and pins associated with a section are visually aligned to be in that section.

The symbol generator offers only little control over the graphic lay-out and design of the schematic symbols. But you can, of course, fine-tune a symbol with the KiCad library editor after having generated the bulk of the data from the template (using the Librarian).

Changing pin types, shapes and order

You can change the information in the pin table by directly editing the table. The fields that allow only a fixed set of choices, present these choices in a drop-down list.

The table presents pop-up menu (under the right mouse click) with additional operations. The content of the menu depends on the column in which you (right-) clicked. For the first two columns, the pop-up menu allows to move the row up or down, or to import a pin description list —this is covered in the next section. Moving pin descriptions in the table also changes their visual order in the drawing of the symbol.

For the other columns, the pop-up menu repeats the choices in the drop-down list for the respective column. However, using the menu allows you to

select multiple pins and set the type, shape or section of these pins to the same value (by choosing that value from the pop-up menu).

Importing a list of pin descriptions

When you right-click on a field in the second column of the pin table (the column "tensans label"), you can select the option Paste pin list from the pop-up menu. This function inserts (or overwrites) pin labels in the table. Before using the function, the pin data must have been copied onto the clipboard.

The Paste pin list function assumes the same format that spreadsheet programs use when these programs copy data to the clipboard. You can therefore first create a pin list in a spreadsheet, for example by copying the pin labels from a data-sheet and using the spreadsheet's "range" functions to (semi-)automatically assign pin numbers. Then, you select the appropriate range in the spreadsheet, copy it onto the clipboard, and paste it into the pin table in the KiCad Librarian.

You should copy onto the clipboard at least two columns. One of the columns must hold the pin numbers (numeric) and one must hold the pin labels. There may be columns with other data between the pin numbers and the pin labels; these are ignored. The KiCad Librarian only looks at the first and the last columns. It determines which of these two holds the pin numbers by looking which of the two is numeric.

Specifically, the data on the clipboard must have the format:
◇ Text format, with one pin description per row; rows separated by newlines.
◇ Fields (columns) on the row must be separated by TABs, commas or semi-colons (as space character is used if all of these separators fail).

Creating a library report

A library report lists all symbols or all footprints in that library. A symbol library report also serves as a cross-reference that maps alias names to their respective symbols. A footprint report shows the basic information on the footprints, such as the pitch and the pad size, plus a drawing at an accurate scale. The reports are in PDF format (Adobe Acrobat).

You can currently only create a report for a local library; that is, you cannot create a report for the remote repository.

To generate the report, select either Create footprint report or Create symbol report from the File menu (only one of these options will be enabled, depending on the mode that the Librarian is in). If you have chosen libraries in both the left and the right lists, the Librarian will first ask which of the two libraries to use. Then it will ask for the filename (and path) for the generated report. After confirming the output file selection, the Librarian starts creating the report, and opens it in the default PDF file viewer as soon as it is done.

There are a few options related to the library report, notably the paper size and orientation. For a correct print-out, you need to avoid that the PDF viewer scales the "logical paper size" to the physical paper. It is therefore important to choose the correct paper size for the library report.

To access the settings, open the dialog below Preferences / Report options. . . from the menu.

Creating templates

The templates to create new footprints are essentially files that contain a single footprint, but with a special header and with expressions instead of coordinates. The templates are currently based on the legacy footprint format, but it will switch to the s-expression format in a future revision. A footprint template is a text file with the extension ".mt".

Symbol templates (for schematic symbols) are likewise text file that define a single symbol, with a special header. A symbol template has the extension ".st".

The format of the KiCad symbol and footprint definitions is *not* in this chapter, please see the relevant documentation of the KiCad project itself.

Next to the template definition file, you should also create an image that gives a preview of the component. This image must have the size of 128 by 128 pixels and be in uncompressed "bitmap" format. Optionally, a footprint template may contain a 3D model in a separate file; this file is covered near the end of this chapter.

The template definition file should declare only a single pad or pin; this pad/pin definition gets repeated for the total number of pads that the footprint or symbol has. When pads need to have specific shapes, or when the position of a pad cannot be expressed in a simple arithmetic rule, you can use conditional expressions. The conditional expressions should be exclusive, so that for each iteration over all pads/pins, only one of the conditional rules is active. Conditional expressions are described near the end of this chapter.

A template starts with a header. Each line of the header starts with a "#", followed by a field name. Long lines in the header may be split over multiple lines with the "\" character. When a line ends with a trailing "\", it will be continued with the next line. That next line, however, must still start with a "#".

Field	Description
#brief	A brief description of the package/symbol type
#model	The base name of a template of a 3D model; if not set, the footprint template base name is used
#note	Additional information, e.g. the sub-type of the package
#param	A RPN expression that assigns default values into variables
#pins	The number of pins, a list of space-separated values, optionally terminated with "..."
#prefix	A prefix for the footprint or symbol; if not set, the template name is used for the prefix
#section	An additional section for a schematic symbol
#version	The version of the template format, must be set to 1

The number of pins may be a single value, such as "2" for a capacitor, or a list of values separated by spaces, such as "3 5 6 8" for a shape like SOT23. The list of values may also end with an ellipsis (a triple dot, like "..."); this indicates that the list of pins extends indefinitely, with the increment given by the last values in the list. For example, for a SSOP shape with a thermal pad, the specification for the #pins may be "17 19 ...". Pin counts of 21, 23, etc. will then be allowed. Note that there *must* be a space before the ellipsis.

The schematic symbol of a part sometimes has a block or section with a specific purpose, and the pins associated with that section ought to be grouped together. For example, part of an IC may be galvanically isolated from the rest of the chip. The template may specify extra sections for this purpose, and allow the user to assign pins to those sections. The syntax is "#section *label side criterion*", where *label* is the name of the section, *side* must be either left or right and *criterion* is a fraction of the size (height) of the body. When the criterion is positive, the section takes the bottom part of the body, when it is negative, the section takes the top part of the body.

The #param field holds an expression in RPN syntax (Reverse Polish Notation). In the particular case of the #param line, this expression should hold only variable assignments. These variables will be used in the template as the default values. See below for a list of these variables. When you change parameters in the side panel of the Librarian, it will overwrite the predefined variables with the values from the side panel. A few of the predefined variables are are *not* in the side panel; these can be set in the dialog below Preferences / Template variables.

The presence of variables in the #param line also affects the behaviour of the side panel. If the variables BW and BL (for body width and body length) are absent, the respective fields in the side panel are disabled. When the variable STP (for "Silk-To-Pad" clearance) is set in the #param line, a change of the pad size, pitch or span, will also cause the body to be recalculated. The rationale is that when the silk-screen drawing of the body must remain at a minimum distance of the pads, any change in the pads may affect that clearance, and therefore require a re-parsing of the template.

At any location in the template, a value may be replaced by an expression between curly brackets ("{" and "}"). This expression must use the RPN syntax and the expression must be on a single line (the closing bracket must be on the same line as the opening bracket). Briefly, in RPN syntax for a calculation, you specify first the operands and then the operator or function. To add two values, you would say "32 8 +" (instead of "32 + 8"). You can find several RPN tutorials on the internet; WikiPedia is a good start.

Numbers in the RPN expression must start with a digit: "`0.123`" is a valid number, but "`.123`" is not. A "`-`" may be prefixed to a number to indicate a negative value; no space should appear between the "`-`" and the first digit. Variable names may have a length of up to 16 characters and must start with an it upper case letter, they are case sensitive.

The RPN expressions use the common arithmetic operators +, -, * and / for addition, subtraction, multiplication and division. To toggle the sign of a value, you need to use the ~ operator —so the - is for subtracting one value from a second, and the ~ flips a positive value to negative, or vice versa. Other operators are <, >, = and <> for comparisons: they result in the value 0 if the comparison fails (0 = false) and 1 if it holds (1 = true). Finally, & and | are logical operators, also resulting in 0 for false and 1 for true.

The ? operator allows to select between two values, based on a condition. The operator takes three parameters: the first is the value for the case that the condition is *true*, the second the value for the case that the condition is *false* and the third is the condition itself.

A final symbol is @ for variable assignment. The @-symbol must be prefixed to a variable name, without white-space between the @ and the name. You may assign values to both the predefined variables or to new variables. Generally, you should avoid changing the predefined variables; if you wish to make sure that a predefined variable has a proper default, you can use an *optional assignment* by putting a "?" immediately after the "@". An optional assignment will assign the value to the variable only if that variable does not already have a value. See below for a list of predefined variables.

Variable	Description
NAME	The name of the footprint/symbol (this is a "text" variable)
DESCR	A description of the footprint/symbol (this is a "text" variable)
REF	The reference prefix for the symbol, e.g. 'R' for a resistor (this is a "text" variable)
BL	Body Length, the vertical dimension of the component body shape, in mm
BW	Body Width, the horizontal dimension of the component body shape, in mm
BP	Body Pen, the pen size for drawing the body, in mm
TRS	Text Reference Size, the size of the reference label, in mm; if this value negative, the text is hidden
TVS	Text Value Size, the size of value label, in mm; when negative, the text is hidden
TW	Text Weight (boldness), in percent of the text size (normal = 15, max. = 25)
PN	Pad/Pin Number, the number of the pad currently being drawn (only valid inside pad definition)
PT	Pad/Pin Total, the number of pads defined for the footprint/symbol
PL	Pad Length, the vertical dimension of the pad, in mm (only for footprints)
PW	Pad Width, the horizontal dimension of the pad, in mm (only for footprints)
PP	Pad Pitch, the pitch between two pads, in mm (only for footprints)

SH	Horizontal Span, the distance between opposing columns, in mm (only for footprints)
SV	Vertical Span, the distance between opposing rows, in mm (only for footprints)
PLA	Auxiliary Pad Length, the vertical dimension of the pad, in mm (only for footprints)
PWA	Auxiliary Pad width, the horizontal dimension of the pad, in mm (only for footprints)
PSRA	Auxiliary Pad Solder Reduction, the reduction of the aperture in the stencil, in percent
DS	Drill Size, in mm (or zero for an SMD pad, only for footprints)
STP	Silk-To-Pad clearance, the offset of silk-screen drawings from the pads, in mm
PS	Pin Section, the current section/side that the pin is in/on (only for symbols)
PNS	Pin Number in Section, sequence number in the current section/side (only for symbols)
PTS	Pin Total in Section, the number of pins in the current section/side (only for symbols)
PT:0	Pin Total for section/side 0; there are also PT:1, PT:2 and PT:3 (only for symbols)
PLB	Pin Label, the functional description of the pin (only for symbols)
PTY	Pin Type, the electrical function of the pin (only for symbols)
PSH	Pin Shape, the graphic shape or style of the pin (only for symbols)
PSC:4	Pin Section Criterion for the left section; PSC:5 for the right section (only for symbols)

A few predefined variables are "text" variables (or "string variables") —for example NAME and DESCR. These non-numeric variables cannot be used in an expression, but they can be assigned to variables. Literal strings can also be assigned to variables. A literal string must be enclosed in double quotes. The PN variable for the pin number is a numeric variable for footprint templates, but a text variable for symbol templates. To calculate the position of a pin, symbol templates can use the PNS variable, along with PS and PTS (these three are numeric variables). The sections are numbered 0 to 3 for left, right, top and bottom, and the section numbers 4 and 5 are for custom sections at the left and at the right.

In addition to operators, there is a select set of functions (in fact, in an RPN expression, there is no real difference between operators and functions). All functions start with a lower case letter (the names are case sensitive).

Function	Description
abs	the absolute value of the parameter
atan	takes two parameters (y and x) and returns the arctangent of y/x
ceil	round the parameter upward to a whole number
chr	convert a number to an ASCII character, in a string type
cos	cosine of the parameter (which must be in degrees)
even	if (after rounding) the parameter is an even number, the result is 1; otherwise it is 0
floor	round the parameter downward to a whole number (truncate to integer)
gacol	convert a pin column of a grid array to an ASCII character, in a string type
max	the maximum of 2 parameters
mil	converts the parameter from mm to mil, rounded to a while number
min	the maximum of 2 parameters
odd	if (after rounding) the parameter is an odd number, the result is 1, otherwise it is 0

round	round the parameter to the nearest whole number
sin	sine of the parameter (which must be in degrees)
sqrt	square root of the parameter
tan	tangent of the parameter (which must be in degrees)

When the expressions in the template become complex, it is useful to break them up and assign intermediate results to temporary variables. A line in the template that contains only one or more expressions that each end with a variable assignment, the line will be parsed, but not be copied to the footprint.

Comments in the template must start with the "#" symbol and run up to the end of the line. Comments may appear on a line of their own, or at the end of lines that contain data.

A special syntax exists to copy a template line *conditionally* to the footprint. This is frequently needed inside the pad definitions. A template may only defined a single pad, but the pads themselves often require different parameters, depending on their position or number. For example, a QFP package with a thermal pad often has an obround shape for the standard pads, but the thermal pad is rectangular.

When a line is prefixed with a condition, the line is only parsed and copied if the condition evaluates to a non-zero value. If the expression in the condition is false, the remainder of the line is ignored.

You create a conditional expression, by putting "{?" at the start of a line, followed by the RPN expression and then end it with a "}". Conditions may only occur at the start of a line.

For conditional blocks that span multiple lines, you can add a label immediately after the "{?"; a label is any word that starts with a colon. The conditional block spans up to the line that contains the label. The label should be on a line of its own and start in the left column. For example, the next snippet creates a one-line condition (for the normal pins) and a conditional block (for the thermal pad).

LISTING Conditional block

```
$PAD
{? PN PINS <=}Sh "{PN}" O {PW} {PL} 0 0 0
{?:THERMAL PN PINS >}
Sh "{PN}" R {PWA} {PLA} 0 0 0
Po 0 0
.SolderPasteRatio {PSRA 2 /}
:THERMAL
Dr 0 0 0
At SMD N 00888000
$EndPAD
```

3D model

A 3D model may be added to the template. The 3D model must be in a separate file, with the extension ".vt" and using the VRML 2.0 syntax. Similar to the main template, numbers in this file can be replaced by expressions between curly brackets ("{" and "}").

Note that KiCad supports only a very limited subset of the VRML 2.0 syntax; it essentially restricts itself to what the "WINGS3D" program generates. As with any VRML file, the first line must be "#VRML V2.0 utf8". On the second line, the template should contain the line "#model "*name*, where *name* is the base name of the template.

The footprint template (in the ".mt" file) must also contain a reference to the 3D model. Just before the "EndMODULE" line, the following section should be present.

LISTING Conditional block
```
$SHAPE3D
Na "{NAME}.wrl"
Sc 0.3937 0.3937 0.3937
Of 0 0 0
Ro 0 0 0
$EndSHAPE3D
```

The scale is set to map millimetres to a unit of 0.1 inch — which is the unit that KiCad assumes for VRML models. The KiCad Librarian expresses all sizes in millimetres.

Setting up a repository

The KiCad Librarian can interface with a central repository with symbols and footprints. This interface is based on HTTP "POST" queries. For every query to the repository, the Librarian sends a command and a set of parameters. All parameters should be URL-encoded.

The Librarian assumes that it is the primary interface to the repository. From signing up to a repository to copying parts and deleting/renaming symbols and footprints, all can be handled from the Librarian. A repository does not need to have a web interface.

The fields that can be present in a query are:

Function	Description
alias	a text string with a list of aliases ("putinfo")
author	the name of the author of the part
cat	either "symbols", "footprints" or "models"
cmd	one of "user", "list", "get", "put", "putinfo" or "del"
data	the data of the symbol, footprint or 3D model; a multi-line string ("put")
descr	a single line with a description of the part ("putinfo")
fplist	a text string with a list of suitable footprints for the symbol ("putinfo")
filter	a text string with a keyword to filter on ("list")
part	the (current) name of the part
pins	the number of pins of the part ("putinfo")
pitch	the main pin pitch of the part ("putinfo")
pwd	the password for access to the repository
thumb	a Base64-encoded byte stream of a thumbnail image ("putinfo")
user	the user name for access to the repository

Not all parameters are present for every command. For example, the `data` is only present in a `put` query. For `put` and `del` queries, the `author` parameter is always the same as the `user` parameter; in a `get` query, the `author` parameter is set to the name that was returned as the "author" in a previous `list` query.

Several fields with general information of the part (a description, the pin count, . . .) are set with `putinfo` command. The `alias` and `fplist` fields are only relevant for symbols; the `pitch` field is only relevant for footprints. The information set with `putinfo` is for the purpose of a web interface or filtering the database with other utilities; the KiCad Librarian does not use the data itself —there is no "`getinfo`" command. When creating a repository only for use with the Librarian, your repository may ignore the `putinfo` command.

The KiCad Librarian furthermore uploads a bitmap with a thumbnail of the symbol or footprint to the repository. This bitmap is in the "PNG" image

format. This bitmap is likewise only for use by a web interface or simple browser. The Librarian does not use the bitmap.

Note that "user" is both a command (one of the valid values for the cmd parameter) and a parameter itself. When signing up to a repository, the cmd parameter is set to "user" and the user parameter is set to the user name.

With the exception of the queries get and list, all queries must return "ok" on success or an error message on failure. The replies should be in plain text (not as HTML). When a get or list query is *unsuccessful*, it also returns an error message, but if these commands are successful, they return data instead.
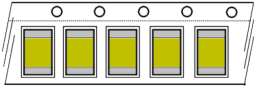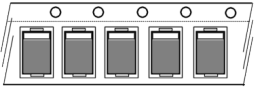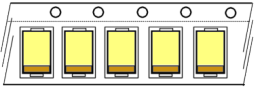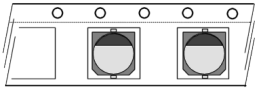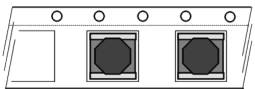
The return data for a get command is the data for the symbol or footprint as it was put in the repository earlier. The output of a successful get command must be plain text; it is not URL-encoded or with HTML-formatting. The return data for a list command is in comma-separated values format, with two fields per line: the part name and the name of the author. Each field may optionally be enclosed in double quotes.

Normal orientations

There are two predominant standards for the *normal orientation* (or "zero-orientation") of components on a PCB. The normal orientation is the orientation that the component has when its rotation is zero degrees. Fortunately, these two standards, IPC-7351 and IEC 61188-7 "Level A", are compatible with each other.[1]

Most surface mount components components come packaged in tape, on reel. A separate standard exists for how the components are packaged in carrier tape, EIA-481. Manufacturers generally adhere to the EIA-481 standard, but many also specify *exceptions* for specific component packages; for example, many manufactures have continued to package QFP and QFN packages according to the rules in EIA-481**C** instead of adopting the changes of EIA-481**D**.

The table below shows the normal orientations according to IPC-7351 and the packaging orientation according to the latest EIA-481 standard (EIA-481D).

IPC-7351	EIA-481D	Notes
		Non-polarized chip or moulded packages, e.g. resistors, capacitors, inductors, crystals.
		Polarized chip or moulded packages (except capacitors, see below). For diodes, pin 1 is the cathode
		Tantalum capacitors or other capacitors in moulded packages. Pin 1 (with a bar) is the positive pole.
		Electrolytic capacitors. Pin 1 is the positive pole. (The black bar marks the negative pole.)
		Power inductors (non-polarized).
		PLCC2 package (for LEDs). See also the notes below.

---

[1] There also exists an IEC 61188-7 "Level B" standard, but "Level A" is predominant.
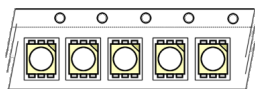
PLCC4 and PLCC6 packages (for multi-colour LEDs). See also the notes below.
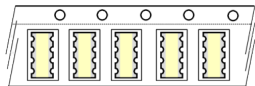
Resistor arrays and capacitor arrays in a chip package.
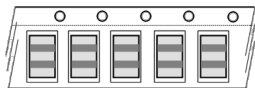
Ceramic resonators with 3 pins. See the notes below for crystals and oscillators with two or four pins.
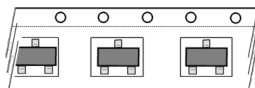
3-pin SOT style transistor, e.g. SOT23, SOT323, SOT523, SC70, TO-236.
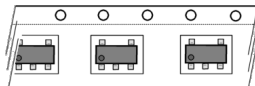
5-pin SOT style packages, e.g. SOT23-5, SC70-5, SSOP-5.
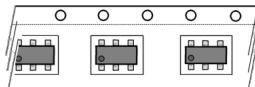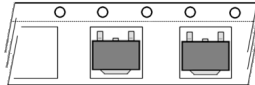
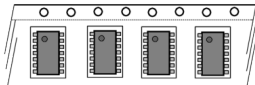6-pin SOT style packages, e.g. SOT23-6, SC70-6, SSOP-6.
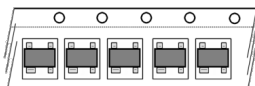
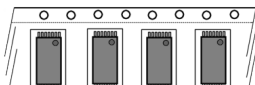SOT143 and SOT343.
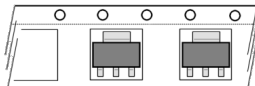
SOT223, SC73, TO-89, TO-261. (See below for similar packages.)
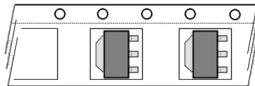
SOT89, TO-293. (See above and below for similar packages.)

DPAK D2PAK, TO-220, TO-252, TO-263. (See above for similar packages.)

Rectangular IC packages: SOIC, SSOP, TSSOP, TSOP type 2 (but see below for TSOP type 1).

TSOP type 1.
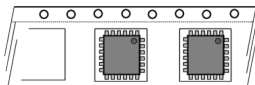
Square IC packages: QFP, QFN (but not BGA and most PLCC, see below).
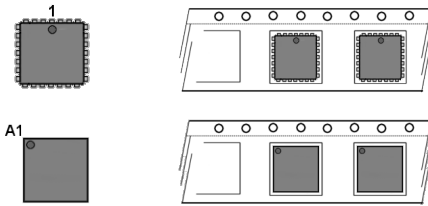
Square IC packagesthat have pin 1 in the middle of the row, which is common for PLCC packages.

Square Grid array IC packages (BGA, PGA).

Notes

• PLCC2, PLCC4 and PLCC6

The low-pin PLCC packages for LEDs have a chamfered corner to mark pin 1. For PLCC2, pin one is usually the cathode (conforming to IPC-7351); for RGB LEDs in PLCC4 and PLCC6 packages, pin 1 is typically the common anode.

The PLCC4 and PLCC4 packages use a clockwise pin numbering scheme, starting from pin 1 in the upper left corner. The zero-orientation is not explicitly defined in the IPC-7351 standard (because IPC-7351 assumes a counter-clockwise pin numbering).

• Crystals, ceramic resonators, oscillators

Crystals in a 2-pin chip package follow the orientation of 2-pin non-polarized chip packages. Crystals and crystal oscillators with four pins (or more) follow the orientation of PLCC4 (like PLCC4, 4-pin crystals and oscillators typically use a clockwise pin numbering). Ceramic resonators with three aligned pins are oriented as shown in the table.

• Tube and tray orientations

Components are oriented sideways in tube (sticks), such that the pins of consecutive components do not touch. The end of the tube that "pin 1" of the component points to is often marked with a green stop or pin.

JEDEC Trays have a chamfered corner. If the tray is oriented such that the chamfered corner is on the upper left, the components in the tray are oriented according to IPC-7351.

License

The Work "KiCad Librarian" is copyright © 2013–2014 by ITB CompuPhase, and distributed under the "Apache License" version 2.0, which is reproduced below.

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions**

    "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

    "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

    "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

    "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

    "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

    "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

    "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.

    "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

    "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

    "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sub-license, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any

entity (including a cross-claim or counter-claim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   a.   You must give any other recipients of the Work or Derivative Works a copy of this License; and

   b.   You must cause any modified files to carry prominent notices stating that You changed the files; and

   c.   You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   d.   If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Index