

KiCad

The KiCad Team

Table of Contents

Einleitung	2
Systemanforderungen	2
Installieren und Aktualisieren von KiCad	3
Einstellungen importieren	3
Migration von Dateien aus früheren Versionen	3
Benutzung der KiCad-Projektverwaltung	5
Standalone-Modus	7
Erstellen eines neuen Projektes	7
Importieren eines Projekts aus einem anderen EDA- Programm	7
Speichern und Laden von Projektarchiven	8
Git integration	9
KiCad Dateien und Ordner	12
Projektdateien	12
Schaltplaneditor-Dateien	12
Leiterplatteneditor Dateien und Ordner	13
Gemeinsame Dateien	13
Fertigungs- und Dokumentationsdateien	14
Speichern und Senden von KiCad-Dateien	14
KiCad Konfiguration	15
Allgemeine Einstellungen	15
Maus und Touchpad Einstellungen	17
Tastaturbefehle	18
Pfadeinstellungen	19
Konfiguration der Bibliotheken	21
Jobsets	22
Defining jobs	23
Defining jobset destinations	23
Available job types	25
Projektvorlagen	28
Verwendung von Vorlagen	28
Speicherorte von Vorlagen	29
Erstellung von Vorlagen	29
Plugin- und Content-Verwaltung	32
Browsing packages	32
Installing packages	34
Managing repositories	36
Creating packages and repositories	36
Referenz der Aktionen	38
KiCad-Projektverwaltung	38

KiCad 9.0 Reference Manual

Copyright

Dieses Dokument unterliegt dem Copyright © 2010-2024 der unten aufgeführten Mitwirkenden. Sie dürfen es unter den Bedingungen der GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), Version 3 oder höher, oder der Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), Version 3.0 oder höher, verbreiten und/oder verändern.

Alle in diesem Leitfaden genannten Marken sind Eigentum ihrer rechtmäßigen Inhaber.

Mitwirkende

Jean-Pierre Charras, Fabrizio Tappero, Jon Evans, Graham Keeth.

Übersetzung

Lorenz Bewig <robotaxi@arcor.de>, 2024; Carsten Schönert <c.schoenert@t-online.de>, 2016

Feedback

Das KiCad-Projekt freut sich über Rückmeldungen, Fehlerberichte und Vorschläge in Bezug auf die Software oder ihre Dokumentation. Weitere Informationen zum Einreichen von Feedback oder zum Melden eines Problems finden Sie in den Anweisungen unter <https://www.kicad.org/help/report-an-issue/>

Software and Documentation Version

This user manual is based on KiCad 9.0.6. Functionality and appearance may be different in other versions of KiCad.

Documentation revision: 90da21fb .

Einleitung

KiCad ist ein Open-Source-Softwarepaket zur Erstellung von elektronischen Schaltplänen und Leiterplatten (PCBs). KiCad unterstützt einen integrierten Design-Workflow, bei dem ein Schaltplan und die dazugehörige Leiterplatte zusammen entworfen werden, sowie eigenständige Workflows für spezielle Anwendungen. KiCad enthält auch mehrere Dienstprogramme, die bei der Schaltungs- und PCB-Konstruktion helfen, einschließlich eines PCB-Rechners zur Bestimmung der elektrischen Eigenschaften von Schaltkreisstrukturen, eines Gerber-Viewers zur Prüfung von Fertigungsdateien und eines integrierten SPICE-Simulators zur Prüfung des Schaltungsverhaltens.

KiCad läuft auf allen gängigen Betriebssystemen und einer breiten Palette von Computerhardware. Es unterstützt Leiterplatten mit bis zu 32 Kupferebenen und eignet sich für die Erstellung von Designs aller Komplexitätsgrade. KiCad wird von einem freiwilligen Team von Software- und Elektroingenieuren auf der ganzen Welt entwickelt, die es sich zur Aufgabe gemacht haben, freie und quelloffene Elektronikdesignsoftware für professionelle Designer zu erstellen.

Die neueste Version dieser Dokumentation finden Sie unter <https://docs.kicad.org>.

Systemanforderungen

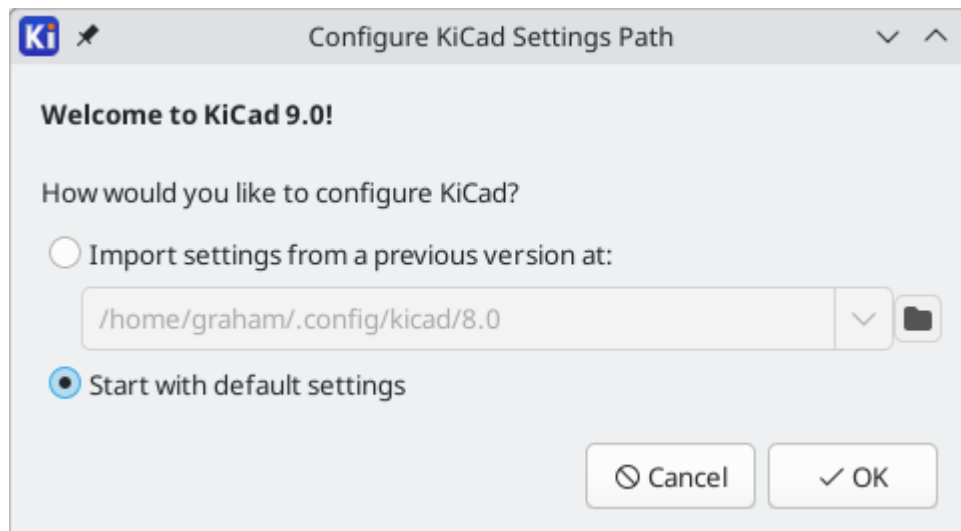
KiCad ist in der Lage, auf einer Vielzahl von Hardware und Betriebssystemen zu laufen, aber einige Aufgaben können auf weniger leistungsfähiger Hardware langsamer oder schwieriger sein. Für ein optimales Erlebnis wird eine eigene Grafikkarte und ein Bildschirm mit einer Auflösung von 1920x1080 oder höher empfohlen.

Die aktuellen Systemanforderungen finden Sie auf der KiCad-Website: <https://kicad.org/help/system-requirements/>

Installieren und Aktualisieren von KiCad

Einstellungen importieren

Jede Hauptversion von KiCad hat ihre eigene Konfiguration, so dass Sie mehrere KiCad-Versionen auf demselben Computer ausführen können, ohne dass sich die Konfigurationen gegenseitig stören. Wenn Sie eine neue Version von KiCad zum ersten Mal ausführen, werden Sie gefragt, wie Sie die Einstellungen initialisieren möchten:



Wenn eine frühere Version von KiCad erkannt wird, haben Sie die Möglichkeit, die Einstellungen aus dieser Version zu importieren. Der Speicherort der vorherigen Konfigurationsdateien wird automatisch erkannt, aber Sie können bei Bedarf einen anderen Speicherort wählen.

Bitte beachten Sie, dass die Schaltplansymbol- und Footprint-Bibliothekstabellen aus der vorherigen KiCad-Version **nicht** importiert werden.

Sie können auch mit den Standardeinstellungen beginnen, wenn Sie keine Einstellungen aus einer früheren Version importieren möchten.

KiCad stores the settings files in a folder inside your user directory. Each KiCad version will use a different versioned subfolder. For KiCad 9, those folders are:

Windows	%APPDATA%\kicad\9.0
Linux	~/.config/kicad/9.0
macOS	/Users/<username>/Library/Preferences/kicad/9.0

Migration von Dateien aus früheren Versionen

Moderne Versionen von KiCad können Dateien öffnen, die in früheren Versionen erstellt wurden. Sie können aber nur Dateien in den neuesten Formaten schreiben. Das bedeutet, dass es grundsätzlich keine weiteren Schritte gibt, um Dateien aus einer früheren Version zu migrieren, außer die Dateien zu öffnen. In einigen Fällen hat sich die Dateierweiterung für eine Datei von einer KiCad-Version zur nächsten geändert.

Nach dem Öffnen dieser Dateien werden sie in dem neuen Format mit der neuen Dateierweiterung gespeichert. Die alten Dateien werden nicht automatisch gelöscht.

In der Dokumentation des Schaltplaneditors werden einige besondere Aspekte beim Öffnen von [Legacy-Schaltplänen](#) beschrieben.

Im Allgemeinen können Dateien, die mit einer Version von KiCad erstellt oder geändert wurden, nicht mit älteren Versionen von KiCad geöffnet werden. Aus diesem Grund ist es wichtig, beim Testen einer neuen KiCad-Version Sicherungskopien Ihrer Projekte aufzubewahren, bis Sie sicher sind, dass Sie die ältere KiCad-Version nicht mehr verwenden müssen.

NOTE

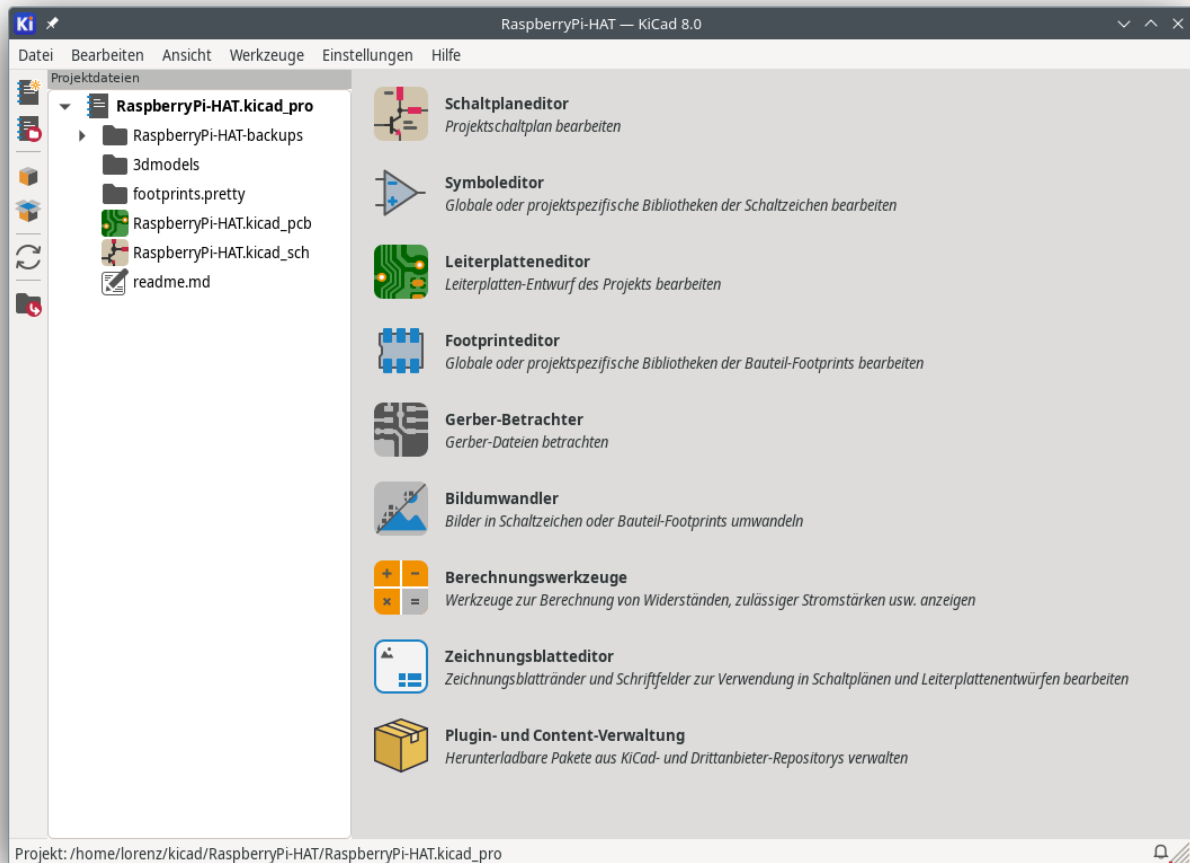
Konfigurationen von Tastaturkürzeln werden zur Zeit nicht aus früheren Versionen importiert. Sie können diese Konfigurationen manuell importieren, indem Sie die verschiedenen *.hotkeys Dateien in das Konfigurationsverzeichnis der neuen Version kopieren. Dabei ist zu beachten, dass KiCad nicht automatisch Konflikte erkennt, z. B. wenn eine Taste mehreren Aktionen zugewiesen ist.

Benutzung der KiCad-Projektverwaltung

The KiCad project manager is the window that opens when you start KiCad. The project manager creates and opens KiCad projects and launches the other KiCad tools:

- the [Schematic Editor](#)
- the [Symbol Editor](#)
- the [PCB Editor](#)
- the [Footprint Editor](#)
- the [Gerber Viewer](#)
- the Image Converter
- the [Calculator Tools](#)
- the [Drawing Sheet Editor](#)
- the [Plugin and Content Manager](#)

Das Fenster der KiCad-Projektverwaltung besteht aus einer Baumansicht auf der linken Seite, in der die Dateien angezeigt werden, die mit dem geöffneten Projekt verbunden sind, und einem Startmenü auf der rechten Seite, das Verknüpfungen zu den verschiedenen Editoren und Werkzeugen enthält.

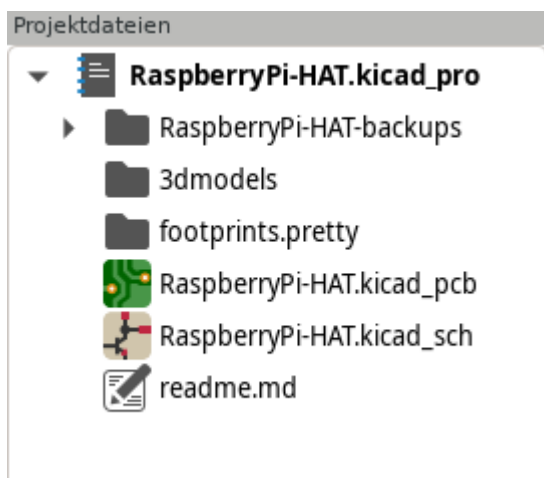


The tree view shows a list of the files inside the project folder. This includes the KiCad schematic and board design files as well as other files and folders in the project folder. If the project contains multiple schematic sheets, only the root sheet is shown in the tree view.

NOTE | Nur Dateien, die KiCad lesen kann, werden in der Projektstrukturansicht angezeigt.

Double-clicking on a file in the tree view will open it in the associated editor. Right-clicking on a file will open a context menu with some file manipulation commands.

If the project is part of a Git repository, the tree shows icons indicating the [version control status](#) of each file and lists the active branch next to the project name. While normally the tree view only shows the schematic root sheet, and not any subsheets, all sheets are shown when the project is in a Git repository so that modifications to each sheet can be observed individually.



KiCad-Projekte enthalten mindestens eine Projektdatei, einen Schaltplan und einen Platinenentwurf. Schaltpläne können aus mehreren Blättern bestehen, jedes in seiner eigenen Datei, aber ein Projekt kann nur eine einzige Platine enthalten. KiCad erwartet, dass die Projektdatei, die Schaltplan-Stammblattdatei und die Leiterplattendatei denselben Namen haben.

Die Symbolleiste am linken Rand des Fensters bietet Kurzbefehle für die gängigen Projektfunktionen:

	Ein neues Projekt erstellen.
	Ein bestehendes Projekt öffnen.
	Zip-Archiv des gesamten Projekts erstellen. Dies beinhaltet Schaltplan Dateien, Bibliotheken, PCB, etc.
	Zip-Archiv eines Projektes in ein Verzeichnis entpacken. Dateien im Zielverzeichnis werden überschrieben.
	Baumansicht aktualisieren, um Änderungen im Dateisystem zu erkennen.
	Das Arbeitsverzeichnis des Projekts im Dateexplorer öffnen.

Standalone-Modus

Sie können die KiCad-Editor-Tools auch im *Standalone-Modus* ausführen, indem Sie sie direkt aus dem Anwendungsprogramm Ihres Betriebssystems und nicht aus dem Projektmanager starten. Es wird gewöhnlich **nicht empfohlen**, die Werkzeuge im Standalone-Modus auszuführen, außer in einigen speziellen Situationen, in denen dies notwendig ist, z. B. beim Importieren von Projekten aus anderen EDA-Werkzeugen. Bei der Ausführung im Standalone-Modus sind einige Projektfunktionen nicht verfügbar, darunter:

- Cross-Probing zwischen dem Schaltplan-Editor und dem Platinen-Editor
- Entwurfssynchronisierung zwischen dem Schaltplan und der Leiterplatte

Erstellen eines neuen Projektes

Die meisten KiCad-Entwürfe beginnen mit der Erstellung eines Projekts. Es gibt zwei Möglichkeiten, ein Projekt über die KiCad-Projektverwaltung zu erstellen: Sie können ein leeres Projekt erstellen oder ein Projekt auf der Grundlage einer vorhandenen Vorlage erstellen. In diesem Abschnitt wird die Erstellung eines neuen leeren Projekts behandelt. Das Erstellen von Projekten aus Vorlagen wird im Abschnitt [Projektvorlagen](#) behandelt.

Um ein neues Projekt zu erstellen, verwenden Sie den Befehl **Neues Projekt...** im Menü **Datei**, klicken Sie auf die Schaltfläche **Neues Projekt** in der Symbolleiste am linken Fensterrand oder verwenden Sie das Tastaturkürzel (standardmäßig **Strg** + **N**).

Sie werden aufgefordert, einen Namen für Ihr Projekt zu vergeben. Standardmäßig wird für Ihr Projekt ein Verzeichnis mit demselben Namen erstellt. Wenn Sie z. B. den Namen **Mein Projekt** eingeben, erstellt KiCad das Verzeichnis **Mein Projekt** und darin die Projektdatei **Mein Projekt/Mein Projekt.kicad_pro**.

Wenn Sie bereits ein Verzeichnis haben, in dem Sie Ihre Projektdateien speichern, können Sie das Kontrollkästchen *Ein neues Verzeichnis für das Projekt erstellen* im Dialogfeld **Neues Projekt erstellen** deaktivieren.

NOTE

Es wird dringend empfohlen, jedes KiCad-Projekt in seinem eigenen Verzeichnis zu speichern.

Sobald Sie den Namen Ihres Projekts ausgewählt haben, erstellt KiCad die folgenden Dateien im Projektverzeichnis:

beispiel.kicad_pro	KiCad-Projektdatei.
beispiel.kicad_sch	Hauptschaltplandatei.
beispiel.kicad_pcb	Datei für gedruckte Leiterplatten.

Importieren eines Projekts aus einem anderen EDA- Programm

KiCad kann Dateien importieren, die von einigen anderen Softwarepaketen erstellt wurden. Einige Softwareformate können als komplette Projekte importiert werden, andere können im Moment nur als

eigenständige Schaltpläne oder Platinen importiert werden und müssen manuell zu einem KiCad-Projekt zusammengefügt werden. Derzeit werden die folgenden Projekttypen unterstützt:

*.sch, *.brd	Eagle 6.x oder neuer (XML-Format)
*.csa, *.cpa	CADSTAR-Archivformat
*.zip	EasyEDA (JLCEDA) Standard Backup
*.epro, *.zip	EasyEDA (JLCEDA) Pro Project

Um ein Projekt aus einem dieser Tools zu importieren, wählen Sie die entsprechende Option aus dem Untermenü **Nicht-KiCad-Projekt importieren...** des Menüs **Datei**.

Sie werden aufgefordert, im Datei-Import-Dialog entweder eine Schaltplan- oder eine Platinen-Datei auszuwählen. Die importierten Schaltplan- und Platinen-Dateien sollten denselben Basis-Dateinamen haben (z. B. project.sch und project.brd). Sobald die gewünschten Dateien ausgewählt sind, werden Sie aufgefordert, ein Verzeichnis zum Speichern des resultierenden KiCad-Projekts auszuwählen.

Derzeit können die folgenden Dokumenttypen eigenständig importiert werden. Um diese Dokumente zu importieren, starten Sie den KiCad Schaltplan-Editor oder PCB-Editor eigenständig (öffnen Sie nicht zuerst die KiCad-Projektverwaltung) und wählen Sie **Datei > Importieren > Nicht-KiCad-Schaltplan** oder **Datei > Importieren > Nicht-KiCad-Platinendatei**. Wenn Sie Altium-Projekte importieren, empfehlen wir, zuerst die Platine zu importieren, das resultierende Projekt zu speichern und dann jedes Schaltplanblatt in das Projekt zu kopieren, nachdem es in einem eigenständigen Schaltplaneditor-Fenster importiert wurde.

*.SchDoc	Altium Designer, Circuit Studio, Circuit Maker Schaltplan-Dokumente
*.PcbDoc	Altium Designer PCB
*.CMPcbDoc	Altium Circuit Maker PCB
*.CSPcbDoc	Altium Circuit Studio PCB
*.pcb	P-Cad 200x ASCII PCB
*.txt, *.fab	Fabmaster PCB

NOTE

KiCad unterstützt keine Schaltpläne mit mehreren Blättern auf oberster Ebene. Beim Importieren von Entwürfen aus anderen Programmen, die diese Funktion unterstützen, muss jedes Schaltplanblatt importiert werden, dann müssen die importierten Blätter als hierarchische Blätter in einem neuen KiCad-Projekt angeordnet werden.

Speichern und Laden von Projektarchiven

Sie können die Dateien Ihres Projekts mit dem Archivierungstool (**Datei → Projektdateien archivieren...**) in einem Zip-Archiv speichern.

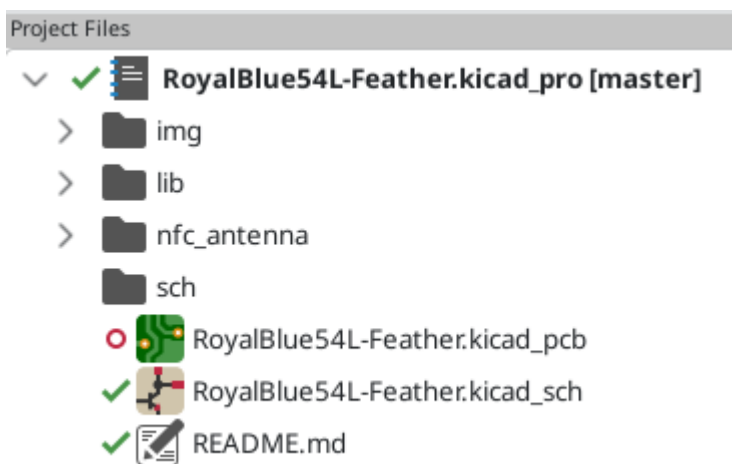
Sie können ein Projekt auch aus einem Archiv entpacken (**Datei → Projektdateien aus Archiv entpacken...**). Wenn Sie ein Projekt in das aktuell geladene Projektverzeichnis entpacken, wird das Projekt automatisch neu geladen, um alle Änderungen in der archivierten Version des Projekts zu berücksichtigen.

Das Archivierungswerkzeug speichert die folgenden Dateien aus Ihrem Projektordner in das Archiv:

<code>*.kicad_prl, *.kicad_pro, *.kicad_sch, *.kicad_sym, *.kicad_pcb, *.kicad_mod, *.kicad_dru, *.kicad_wks, *.kicad_jobset, *.wbk, *.json, fp- lib-table, sym-lib-table, design- block-lib-table</code>	KiCad design files
<code>*.pro, *.sch, *.lib, *.dcm, *.cmp, *.brd, *.mod</code>	Legacy KiCad design files
<code>*.stp, *.step</code>	3D models
<code>*.g?, *.g??, *.gm??, *.gbrjob</code>	Gerber files
<code>*.pos, *.drl, *.nc, *.xnc, *.d356, *.rpt</code>	Manufacturing files
<code>*.net</code>	Netlists
<code>*.py</code>	Python scripts
<code>*.pdf, *.txt</code>	Documentation files
<code>*.cir, *.sub, *.model</code>	SPICE models
<code>*.ibs, *.pkg</code>	IBIS models

Git integration

The KiCad Project Manager integrates with the Git version control tool for tracking changes in your projects. It can work with an existing local Git repository, clone a project from a remote repository, or create a new repository in an existing project. You can use the tool to commit changes from your project, push and pull from a remote repository, and switch branches.



If you open a project that is already under version control with Git, i.e. it is part of an existing Git repository, you can use KiCad's version control features to track changes in the project without any additional

configuration. The active branch is displayed next to the project name, and the version control status of each file in your project is shown graphically in the project files tree. For example, the ✓ icon indicates a file is unchanged, ○ indicates a file has uncommitted changes, and + indicates a file is not tracked. No icons are shown if the project is not part of a Git repository.

NOTE

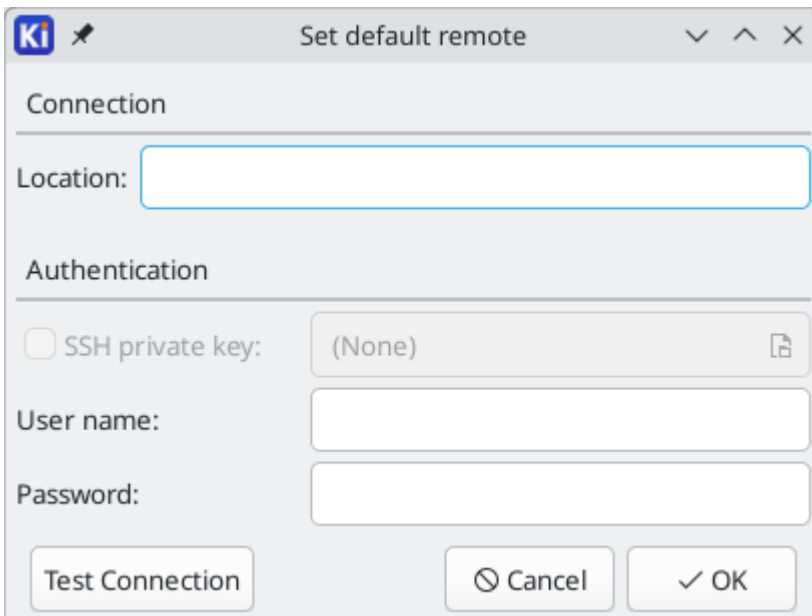
You can disable KiCad's Git integration features in the **Version Control** page of the Preferences dialog. The preferences also let you configure the remote update interval and the default commit author information.

Adding version control to an existing project

If an existing project is not already under version control, you can initialize a new Git repository in the project by right clicking on one of the files in the project files tree and clicking **Version Control** → **Add Project to Version Control....** You must configure a remote when initializing a repository in this way. Configuring the repository requires the following information:

- **Location:** The URL or file path to the remote. HTTPS, SSH, and local (file) connections are supported. The format of the URL is used to automatically determine the type of connection and set the authentication options accordingly.
- **SSH private key:** The SSH private keyfile to use for authentication (SSH connections only).
- **User name:** The user name to use for authentication. For SSH connections, the user name is often `git`.
- **Password:** The password to use for authentication (HTTPS connections only).
- **SSH key password:** The passphrase for your SSH key, if you have configured one (SSH connections only).

You can check the connection and authentication by clicking the **Test Connection** button.

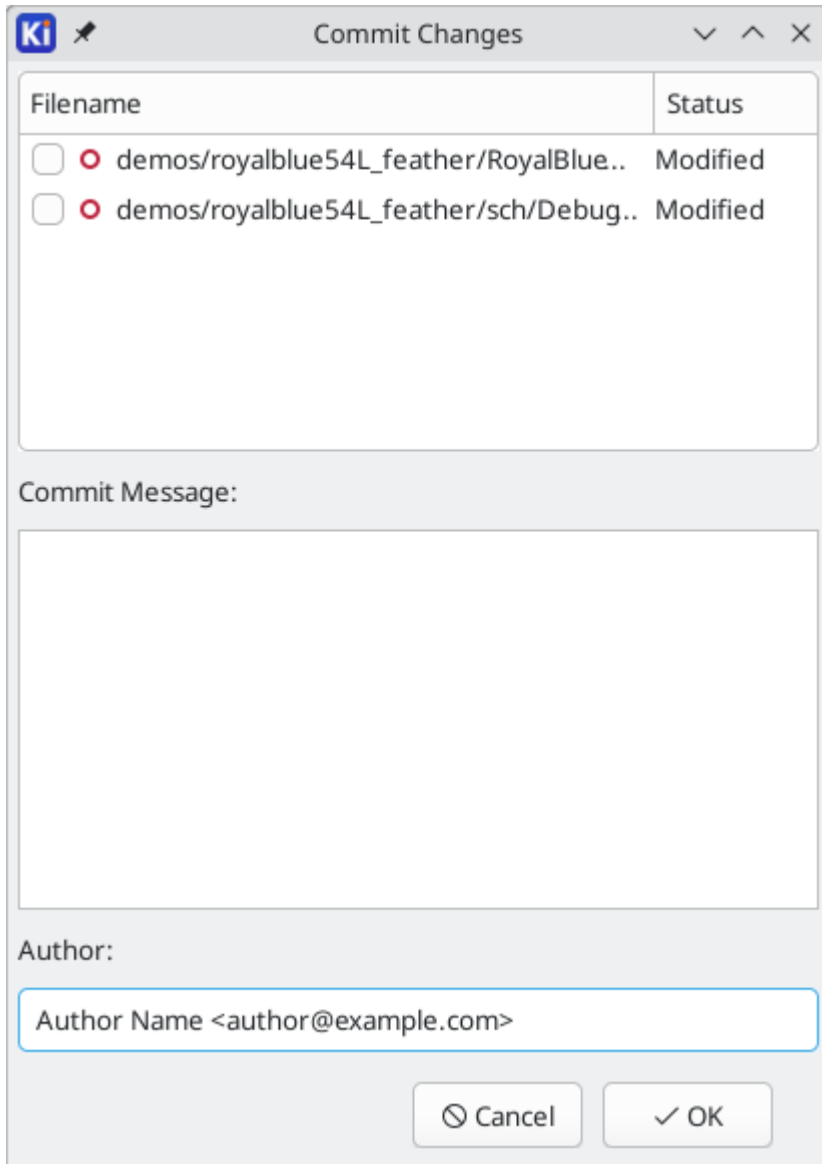


Cloning an existing repository

To clone an existing repository and open the cloned project, use **File** → **Clone Project from Repository....** You can clone a remote repository using SSH or HTTPS, or clone a local repository. The configuration settings for cloning are the same as the settings for configuring a new repository and remote for an existing project.

Tracking project changes with Git

When you have made changes that you want to commit, you can commit either the entire project (right click → **Version Control** → **Commit Project...**) or a specific file (right click the file → **Version Control** → **Commit File...**). Both actions open the Commit Changes dialog, but the Commit Project action shows all changed files in the repository, while the Commit File action shows only the file that was right clicked. The Commit Changes dialog lets you select the changed files you want to include in the commit, provide a commit message and author, and commit the changes.



Filename	Status
<input type="checkbox"/> ○ demos/royalblue54L_feather/RoyalBlue..	Modified
<input type="checkbox"/> ○ demos/royalblue54L_feather/sch/Debug..	Modified

Commit Message:

Author:

Author Name <author@example.com>

Cancel OK

To push changes to the remote, right click in the project files tree and select **Version Control** → **Push**. To pull from the remote, right click and select **Version Control** → **Pull**. You can switch branches by selecting the desired branch from the **Version Control** → **Switch to Branch** menu.

Finally, you can remove version control entirely, deleting all tracked history from the local repository, by right clicking and selecting **Version Control** → **Remove Version Control**.

KiCad Dateien und Ordner

KiCad erstellt und verwendet Dateien mit den folgenden spezifischen Dateierweiterungen (und Ordnern) für die Schaltplan- und Leiterplattenbearbeitung.

Many of these files include important design information, especially the project file (`.kicad_pro`), the schematic file(s) (`.kicad_sch`), and the board file (`.kicad_pcb`). Other files may also be necessary. Such files should always be included when distributing the project. Some files are not necessary to distribute with the project, such as the project-local settings file (`.kicad_pr1`) or the `fp-info-cache` file. Files that are unnecessary to distribute are noted in the table below.

Projektdateien

<code>*.kicad_pro</code>	Project file, containing settings that are shared between the schematic and PCB
<code>*.pro</code>	Legacy (KiCad 5.x and earlier) project file. Can be read and will be converted to a <code>.kicad_pro</code> file by the project manager.

Schaltplaneditor-Dateien

<code>*.kicad_sch</code>	Schematic files, containing all symbol and connection information.
<code>*.kicad_sym</code>	Schematic symbol library file, containing the symbol descriptions: graphic shape, pins, fields.
<code>*.kicad_blocks</code>	Schematic design block library folders. The folder itself is the library.
<code>*.kicad_block</code>	Schematic design block folder for defining a reusable schematic design. The folder is the design block, and contains a <code>.kicad_sch</code> file defining the design block's schematic and a <code>.json</code> file defining the design block's metadata.
<code>*.wbk</code>	Simulator workbook file containing SPICE simulation setup information.
<code>*.sch</code>	Legacy (KiCad 5.x and earlier) schematic file. Can be read and will be converted to a <code>.kicad_sch</code> file on write.
<code>*.lib</code>	Legacy (KiCad 5.x and earlier) schematic library file. Can be read but not written.
<code>*.dcm</code>	Legacy (KiCad 5.x and earlier) schematic library documentation. Can be read but not written.
<code>*-cache.lib</code>	Legacy (KiCad 5.x and earlier) schematic component library cache file. Required for proper loading of a legacy schematic (<code>.sch</code>) file.
<code>sym-lib-table</code>	Symbol library table: list of symbol libraries available in the schematic editor.
<code>design-block-lib-table</code>	Design block library table: list of design block libraries available in the schematic editor.

Leiterplatteneditor Dateien und Ordner

*.kicad_pcb	Board file containing all info but the page layout.
*.pretty	Footprint library folders. The folder itself is the library.
*.kicad_mod	Footprint files, containing one footprint description each.
*.kicad_dru	Design rules file, containing custom design rules for a certain .kicad_pcb file.
*.brd	Legacy (KiCad 4.x and earlier) board file. Can be read, but not written, by the current board editor.
*.mod	Legacy (KiCad 4.x and earlier) footprint library file. Can be read by the footprint or the board editor, but not written.
fp-lib-table	Footprint library table: list of footprint libraries available in the board editor.
fp-info-cache	Cache to speed up loading of footprint libraries. Does not need to be distributed with the project or put under version control.

Gemeinsame Dateien

*.kicad_prl	Local settings for the current project; helps KiCad remember the last used settings such as layer visibility or selection filter. Does not need to be distributed with the project or put under version control.
*.kicad_wks	Page layout (drawing border and title block) description file.
*.kicad_jobset	Jobset definition file containing output jobsets.
*.net	Netlist file created from the schematic, and read by the board editor. Note that the recommended workflow for transferring information from the schematic to the board does not require the use of netlist files.
*.cmp	Association between components used in the schematic and their footprints. It can be created by the Board Editor and imported by the Schematic Editor. Its purpose is to import changes from the board to the schematic, for users who change footprints in the Board Editor (for instance using Exchange Footprints command) and want to import these changes back to the schematic. Note that the recommended workflow for transferring information from the board to the schematic does not require the use of .cmp files.

Fertigungs- und Dokumentationsdateien

*.gbr	Gerber files, for fabrication.
*.drl	Drill files (Excellon format), for fabrication.
*.pos	Position files (ASCII format), for automatic insertion machines.
*.rpt	Report files (ASCII format), for documentation.
*.ps	Plot files (Postscript), for documentation.
*.pdf	Plot files (PDF format), for documentation.
*.svg	Plot files (SVG format), for documentation.
*.dxf	Plot files (DXF format), for documentation.

Speichern und Senden von KiCad-Dateien

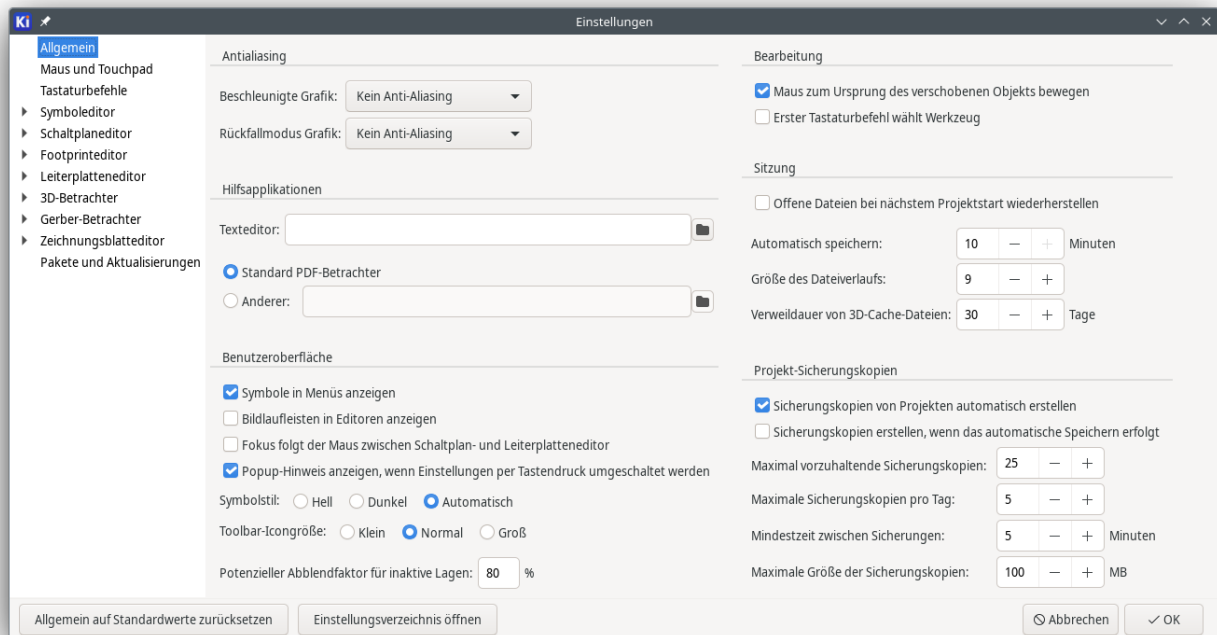
KiCad-Schaltplan- und Plattendateien enthalten alle im Entwurf verwendeten Schaltplansymbole und Footprints, so dass Sie diese Dateien problemlos selbst sichern oder versenden können. Einige wichtige Entwurfsinformationen sind in der Projektdatei (`.kicad_pro`) gespeichert. Wenn Sie also einen vollständigen Entwurf senden, sollten Sie diese Datei unbedingt mitsenden.

Some files, such as the project-local settings file (`.kicad_pr1`) and the `fp-info-cache` file, are not necessary to send with your project. If you use a version control system such as Git to keep track of your KiCad projects, you can add these files to the list of ignored files so that they are not tracked.

KiCad Konfiguration

Die KiCad-Einstellungen können jederzeit über das Menü **Einstellungen** oder über die Tastenkombination (standardmäßig **Strg + ,**) aufgerufen werden. Das Einstellungsfenster wird von den laufenden KiCad-Werkzeugen gemeinsam genutzt. Einige Einstellungen gelten für alle Werkzeuge, andere sind spezifisch für ein bestimmtes Werkzeug (z. B. den Schaltplan- oder Platineneditor).

Allgemeine Einstellungen



Beschleunigte Grafik: KiCad kann verschiedene Methoden verwenden, um Aliasing (gezackte Linien) beim Rendern mit einer Grafikkarte zu verhindern. Unterschiedliche Methoden können auf unterschiedlicher Hardware besser aussehen, daher sollten Sie experimentieren, um die Methode zu finden, die Ihnen am besten gefällt.

Rückfallmodus Grafik: KiCad kann auch Antialiasing anwenden, wenn der Rückfallmodus verwendet wird. Das Aktivieren dieser Funktion kann zu einer schlechten Leistung auf einiger Hardware führen.

Texteditor: Wählen Sie einen Texteditor aus, der beim Öffnen von Textdateien in der Projektstrukturansicht verwendet werden soll.

Standard PDF-Betrachter: Wählen Sie ein Programm, mit dem Sie PDF-Dateien öffnen können.

Symbole in Menüs anzeigen: Aktiviert Symbole in Dropdown-Menüs in der gesamten KiCad-Benutzeroberfläche.

NOTE | Auf einigen Betriebssystemen werden die Symbole in den Menüs nicht angezeigt.

Bildlaufleisten in Editoren anzeigen: Wenn diese Option aktiviert ist, werden in jedem Werkzeug Bildlaufleisten neben den Bearbeitungsflächen angezeigt. Wenn sie deaktiviert sind, werden die

Bildlaufleisten nicht angezeigt.

Fokus folgt der Maus zwischen Schaltplan- und Leiterplatteneditor: Ist diese Funktion aktiviert, wird das Fenster unter dem Mauszeiger automatisch fokussiert.

Symbolgröße: Legt die Größe der Symbole fest, die in den Menüs und Schaltflächen in KiCad verwendet werden. Wählen Sie *Automatisch*, um automatisch einen geeigneten Symbolmaßstab basierend auf den Einstellungen Ihres Betriebssystems auszuwählen.

Symbolstil: Legt fest, ob der Symbolstil für helle oder für dunkle Fensterhintergründe verwendet werden soll. Bei der Standardeinstellung *Automatisch* wird der Stil auf der Grundlage der Helligkeit des Fensterschemas des Betriebssystems festgelegt.

Potenzieller Abblendfaktor für inaktive Lagen: Legt fest, wie stark nicht fokussierte Elemente im Modus "Hoher Kontrast" abgeblendet werden.

Maus zum Ursprung des verschobenen Objekts bewegen: Wenn diese Option aktiviert ist, wird der Mauszeiger zum Ursprung eines Objekts bewegt, wenn Sie einen Bewegungsbefehl für dieses Objekt starten.

Erster Tastaturbefehl wählt Werkzeug: Wenn deaktiviert, wird beim Drücken des Tastaturbefehls für einen Befehl wie z.B. *Elektrische Verbindung hinzufügen* der Befehl sofort an der aktuellen Cursorposition gestartet. Wenn aktiviert, wird durch das erste Drücken des Tastaturbefehls nur das Werkzeug *Elektrische Verbindung hinzufügen* ausgewählt, aber es wird nicht sofort eine Verbindung hergestellt.

Offene Dateien beim nächsten Projektstart wiederherstellen: Wenn diese Option aktiviert ist, öffnet KiCad automatisch alle zuvor geöffneten Dateien, wenn ein Projekt erneut geöffnet wird.

Automatisch speichern: Beim Bearbeiten von Schaltplänen und Platinendateien kann KiCad Ihre Arbeit automatisch und regelmäßig speichern. Stellen Sie das Intervall auf 0, um diese Funktion zu deaktivieren.

Größe des Dateiverlaufs: Konfigurieren Sie die Anzahl der Einträge in der Liste der zuletzt geöffneten Dateien

Verweildauer von 3D-Cache-Dateien: KiCad erstellt einen Cache für 3D-Modelle, um den 3D-Viewer zu beschleunigen. Sie können konfigurieren, wie lange dieser Cache gehalten werden soll, bevor alte Dateien gelöscht werden.

Sicherungskopien von Projekten automatisch erstellen: Wenn diese Option aktiviert ist, werden KiCad-Projekte gemäß den unten aufgeführten Einstellungen automatisch in ZIP-Dateien archiviert. Die Archive werden in einem Unterordner des Projektordners gespeichert. Backups werden beim Speichern von Dateien im Projekt erstellt.

Sicherungskopien erstellen, wenn das automatische Speichern erfolgt: Wenn diese Option aktiviert ist, wird bei jedem automatischen Speichern einer Datei eine Sicherungskopie erstellt (sofern die unten aufgeführten Einstellungen dies zulassen). Diese Einstellung hat keine Auswirkung, wenn das Intervall für die automatische Speicherung auf 0 (deaktiviert) eingestellt ist.

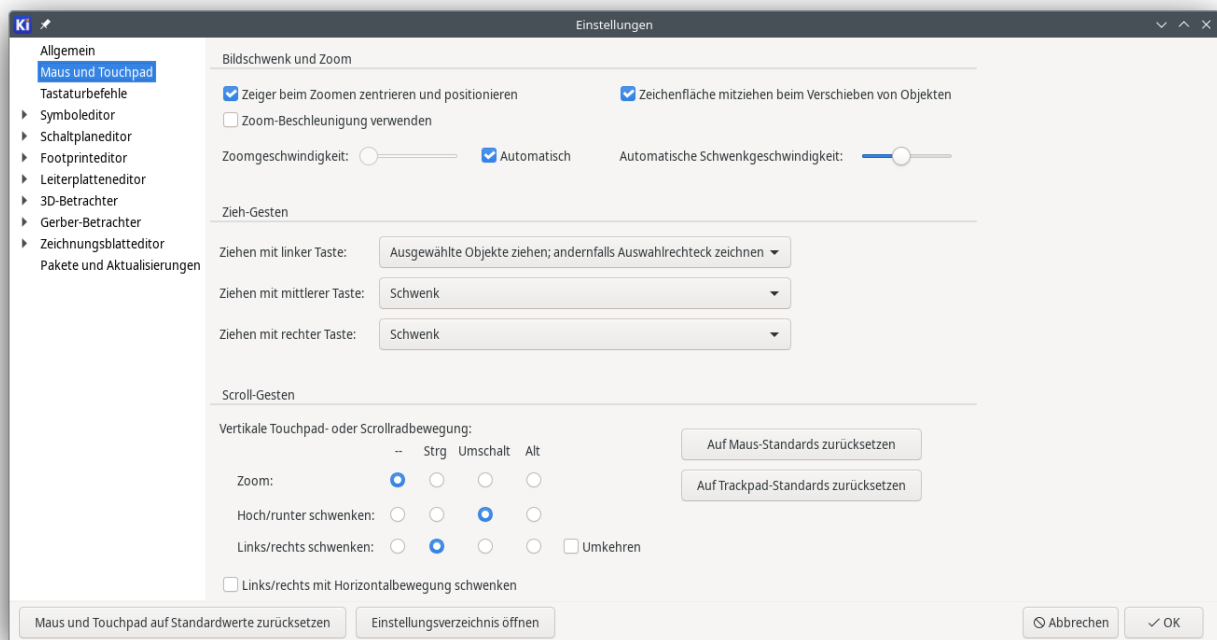
Maximal vorzuhaltende Sicherungskopien: Bei der Erstellung einer neuen Sicherung wird die älteste Sicherung gelöscht, um die Gesamtzahl der Sicherungsdateien unter dieser Grenze zu halten.

Maximale Sicherungskopien pro Tag: Wenn eine neue Sicherung erstellt wird, wird die älteste Sicherung, die am Tag erstellt wurde, gelöscht, um unter dieser Grenze zu bleiben.

Mindestzeit zwischen Sicherungen: Wenn eine Sicherung ausgelöst wird (z. B. durch das Speichern einer Platinendatei), wird die Sicherung nicht erstellt, wenn eine vorhandene Sicherungsdatei neuer ist als diese Grenze.

Maximale Größe der Sicherungskopien: Bei der Erstellung einer neuen Sicherung werden die ältesten Sicherungsdateien gelöscht, um die Gesamtgröße des Verzeichnisses der Sicherungsdateien unter dieser Grenze zu halten.

Maus und Touchpad Einstellungen



Zeiger beim Zoomen zentrieren und positionieren: Wenn diese Funktion aktiviert ist, wird beim Zoomen mit den Tastenbefehlen oder dem Mausexplorer die Ansicht auf die Position des Mauszeigers zentriert.

Zoom-Beschleunigung verwenden: Wenn diese Funktion aktiviert ist, führt ein schnellerer Bildlauf mit dem Mausexplorer oder Touchpad zu einer schnelleren Änderung des Zooms.

Zoom-Geschwindigkeit: Steuert, wie stark sich der Zoom bei einem bestimmten Bildlauf mit dem Mausexplorer oder Touchpad ändert. Verwenden Sie *Automatisch*, um einen Standardwert abhängig von Ihrem Betriebssystem festzulegen.

Zeichenfläche mitziehen beim Verschieben von Objekten: Wenn diese Funktion aktiviert ist, kann die Ansicht beim Verschieben eines Objekts mitgezogen werden, indem es nahe an den Rand der Zeichenfläche bewegt wird.

Automatische Schwenkgeschwindigkeit: Steuert, wie schnell die Zeichenfläche beim Bewegen eines Objekts verschoben wird.

Maustasten: Sie können festlegen, ob das Ziehen der mittleren und rechten Maustaste die Ansicht zoomen, schwenken oder keine Wirkung haben soll. Sie können auch das Verhalten des Ziehens mit der linken Maustaste davon abhängig machen, ob bereits Objekte in der Zeichenfläche ausgewählt sind oder nicht.

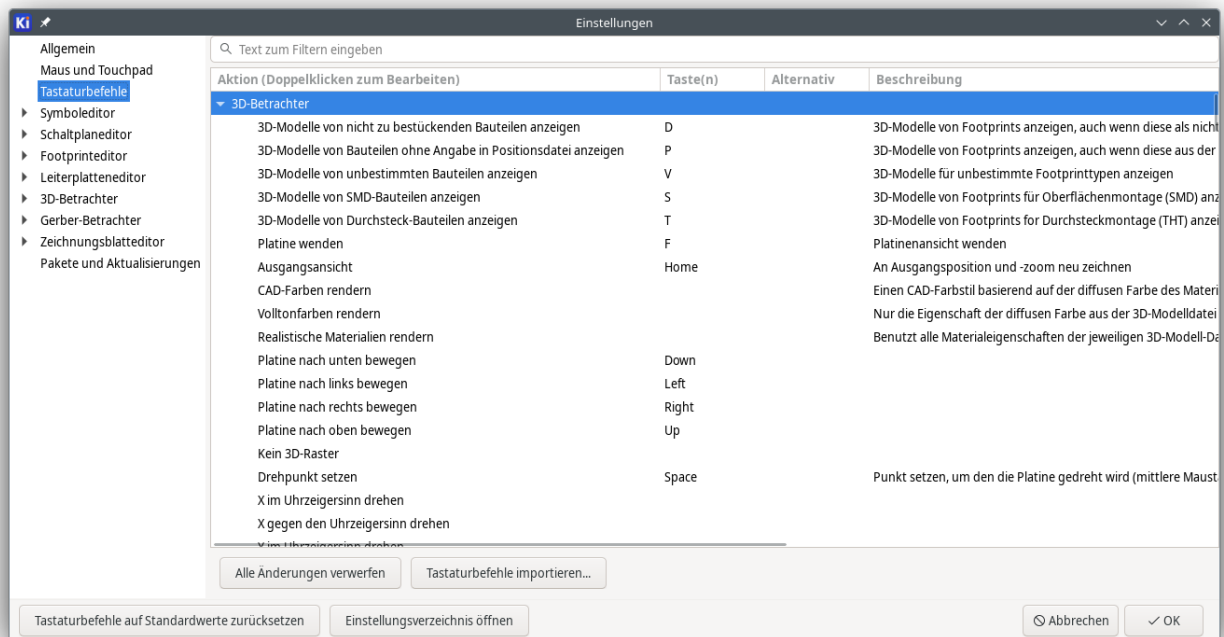
NOTE

Die linke Maustaste wird immer zum Auswählen und Verändern von Objekten verwendet.

Mausrad- und Touchpad-Bildlauf: Sie können das Verhalten des Mausrads oder der vertikalen Bewegung des Touchpads beim Drücken bestimmter Modifizierungstasten einstellen.

Links/rechts schwenken mit horizontaler Bewegung: Wenn diese Option aktiviert ist, können Sie die Ansicht mit dem Touchpad oder dem horizontalen Scrollrad (falls an Ihrer Maus vorhanden) schwenken.

Tastaturbefehle



In diesem Dialogfeld können Sie die zur Steuerung von KiCad verwendeten Tastaturbefehle anpassen. Die Tastaturbefehle im Abschnitt *Allgemein* werden von allen KiCad-Programmen gemeinsam genutzt. Tastaturbefehle für die einzelnen KiCad-Programme werden angezeigt, wenn das jeweilige Programm ausgeführt wird. Sie können denselben Tastaturbefehl einer anderen Aktion in verschiedenen KiCad-Programmen zuweisen (z.B. dem Schaltplaneditor und dem Platineneditor), aber Sie können einen Tastaturbefehl nicht mehr als einer Aktion im selben Programm zuweisen.

Es gibt viele verfügbare Befehle, so dass nicht allen standardmäßig ein Tastaturbefehl zugewiesen ist. Sie können jedem Befehl einen Tastaturbefehl hinzufügen, indem Sie in der Liste auf den Befehl doppelklicken. Wenn Sie einen Tastaturbefehl wählen, der bereits einem anderen Befehl zugewiesen ist, können Sie diesen Tastaturbefehl für den gewählten Befehl verwenden, wodurch die Zuweisung des Tastaturbefehls für den ursprünglichen Befehl aufgehoben wird.

Änderungen, die Sie an Tastaturbefehlen vorgenommen haben, werden mit einem * -Zeichen am Ende des Befehlsnamens angezeigt. Sie können Änderungen an einem bestimmten Befehl rückgängig machen, indem Sie mit der rechten Maustaste auf diesen Befehl klicken und **Änderungen rückgängig machen** wählen, oder Sie können alle Änderungen mit der Schaltfläche unterhalb der Befehlsliste rückgängig machen.

Tastaturbefehle importieren

Die Einstellungen für Tastaturbefehle werden in `.hotkeys`-Dateien im KiCad-Einstellungsverzeichnis gespeichert (siehe den Abschnitt [Einstellungen](#) für Informationen darüber, wo sich das Einstellungsverzeichnis auf Ihrem Betriebssystem befindet). Wenn Sie KiCad Tastaturbefehle auf einem Computer nach Ihren Wünschen konfiguriert haben, können Sie diese Konfiguration auf einen anderen Computer übertragen, indem Sie die entsprechende(n) `.hotkeys`-Datei(en) importieren.

Pfadeinstellungen

In KiCad können manche Pfade durch **Umgebungsvariablen** definiert werden. Einige der Umgebungsvariablen werden intern durch KiCad definiert und können verwendet werden, um Pfade für Bibliotheken, 3D-Formen usw. zu definieren.

Dies ist nützlich, wenn die absoluten Pfade nicht bekannt sind oder sich ändern können (z. B. wenn Sie ein Projekt auf einen anderen Computer übertragen), und auch, wenn ein Basispfad von vielen ähnlichen Elementen gemeinsam genutzt wird. Beachten Sie die folgenden Objekte, die an unterschiedlichen Orten installiert sein können:

- Schaltplan Symbolbibliotheken
- Footprint Bibliotheken
- 3D-Modelldateien, die in Footprint-Definitionen verwendet werden

For instance, the path to the `connect.pretty` footprint library, when using the `KICAD9_FOOTPRINT_DIR` path variable, would be defined as `${KICAD9_FOOTPRINT_DIR}/connect.pretty`.

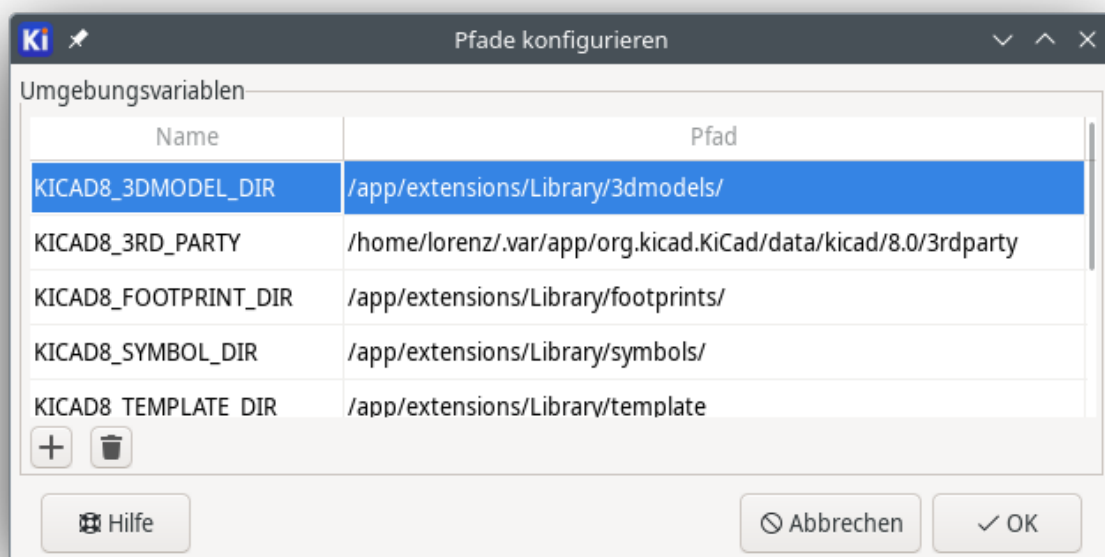
Das Menü **Einstellungen** → **Pfade konfigurieren...** ermöglicht es Ihnen, Pfade für einige eingebaute KiCad-Pfadvariablen zu definieren und Ihre eigenen Pfadvariablen hinzuzufügen, um bei Bedarf persönliche Pfade zu definieren.

NOTE

KiCad will automatically resolve versioned path variables from older versions of KiCad to the value of the corresponding variable from the current KiCad version, as long as the old variable is not explicitly defined itself. For example, `${KICAD8_FOOTPRINT_DIR}` will automatically resolve to the value of `${KICAD9_FOOTPRINT_DIR}` if there is no `KICAD8_FOOTPRINT_DIR` variable defined.

KiCad Umgebungsvariablen

KICAD9_3DMODEL_DIR	Base path of KiCad's standard 3D footprint model library files.
KICAD9_3RD_PARTY	Location for plugins, libraries, and color themes installed by the Plugin and Content Manager .
KICAD9_FOOTPRINT_DIR	Base path of KiCad's standard footprint library files.
KICAD9_SYMBOL_DIR	Base path of KiCad's standard symbol library files.
KICAD9_TEMPLATE_DIR	Location of KiCad's standard project template library files.
KICAD_USER_TEMPLATE_DIR	Location of personal project templates.
SPICE_LIB_DIR	Location of personal simulation model libraries . This variable is not defined by default.
KIPRJMOD	Absolute path to the current project directory. This variable is set automatically and cannot be redefined.



Pfade, die im **Pfade konfigurieren**-Dialog festgelegt werden, sind KiCad-intern und nicht als Umgebungsvariablen außerhalb von KiCad sichtbar. Sie werden in den [KiCad Benutzer-Konfigurationsdateien](#) gespeichert.

Pfade können auch als Systemumgebungsvariablen außerhalb von KiCad festgelegt werden, die alle Einstellungen in der Benutzerkonfiguration außer Kraft setzen.

NOTE Sie können über den Dialogfeld 'Pfade konfigurieren' keine Umgebungsvariable überschreiben, die außerhalb von KiCad festgelegt wurde. Jede Variable, die extern gesetzt wurde, wird im Dialogfeld als schreibgeschützt angezeigt.

Beachten Sie auch, dass die Pfadvariable `KIPRJMOD` **immer** intern von KiCad definiert wird und sich auf den **absoluten Pfad des aktuellen Projekts** ausdehnt. Zum Beispiel ist `${KIPRJMOD}/connect.pretty` immer der Ordner `connect.pretty` (die Footprint-Bibliothek) innerhalb **des aktuellen Projektordners**. Die Variable `KIPRJMOD` kann nicht im Dialogfeld 'Pfade konfigurieren' geändert oder durch eine externe Umgebungsvariable überschrieben werden.

Erweiterte Umgebungsvariablen

Einige erweiterte Umgebungsvariablen können eingestellt werden, um festzulegen, wo KiCad bestimmte Dateien erwartet. Standardmäßig werden diese Orte auf der Grundlage Ihrer Plattform festgelegt, aber sie können durch Systemumgebungsvariablen überschrieben werden. Diese Variablen werden im Dialogfeld 'Pfade konfigurieren' nicht angezeigt und können nicht in Pfadsubstitutionen verwendet werden.

Das Ändern dieser Variablen führt nicht dazu, dass KiCad Dateien vom Standardspeicherort an den neuen Speicherort verschiebt. Wenn Sie also diese Variablen ändern, müssen Sie alle gewünschten Einstellungen oder Dateien manuell kopieren.

KICAD_CONFIG_HOME	Basispfad der KiCad-Konfigurationsdateien. Innerhalb dieses Verzeichnisses werden für jede KiCad-Unterversion Unterverzeichnisse angelegt.
KICAD_DOCUMENTS_HOME	Basispfad der vom Benutzer modifizierbaren KiCad-Dokumente, wie Projekte, Vorlagen, Python-Skripte, Bibliotheken, usw. Innerhalb dieses Verzeichnisses werden für jede KiCad-Unterversion Unterverzeichnisse erstellt. Dieses Verzeichnis wird als Speicherort für Benutzerdaten vorgeschlagen, muss aber nicht verwendet werden.
KICAD_STOCK_DATA_HOME	Basispfad der KiCad-Bestandsdaten, einschließlich Standardbibliotheken. Die Daten in diesem Verzeichnis werden vom KiCad-Installationsprogramm oder dem Systempaketmanager verwaltet und sind nicht für Daten vorgesehen, die vom Anwender geschrieben werden können.

WARNING

Wenn Sie die Konfiguration der Pfade ändern, beenden Sie bitte KiCad und starten Sie es neu, um Probleme bei der Pfadbehandlung zu vermeiden.

Konfiguration der Bibliotheken

Über das Menü **Einstellungen** → **Symbolbibliotheken verwalten...** können Sie die Liste der Symbolbibliotheken verwalten ([Symbolbibliotheken verwalten](#)).

Verwenden Sie ebenfalls das Menü **Einstellungen** → **Footprintbibliotheken verwalten...**, um die Liste der Footprintbibliotheken zu verwalten ([Footprintbibliotheken verwalten](#)).

Für jede Art von Bibliothek (Symbol und Footprint) gibt es 2 Bibliothekstabellen: eine globale und eine projektspezifische. Die globale Bibliothekstabelle befindet sich im [Benutzer-Konfigurationsverzeichnis](#) und enthält eine Liste der für alle Projekte verfügbaren Bibliotheken. Die projektspezifische Bibliothekstabelle ist optional und enthält eine Liste der projektspezifischen Bibliotheken. Sie befindet sich im Projektverzeichnis.

Jobsets

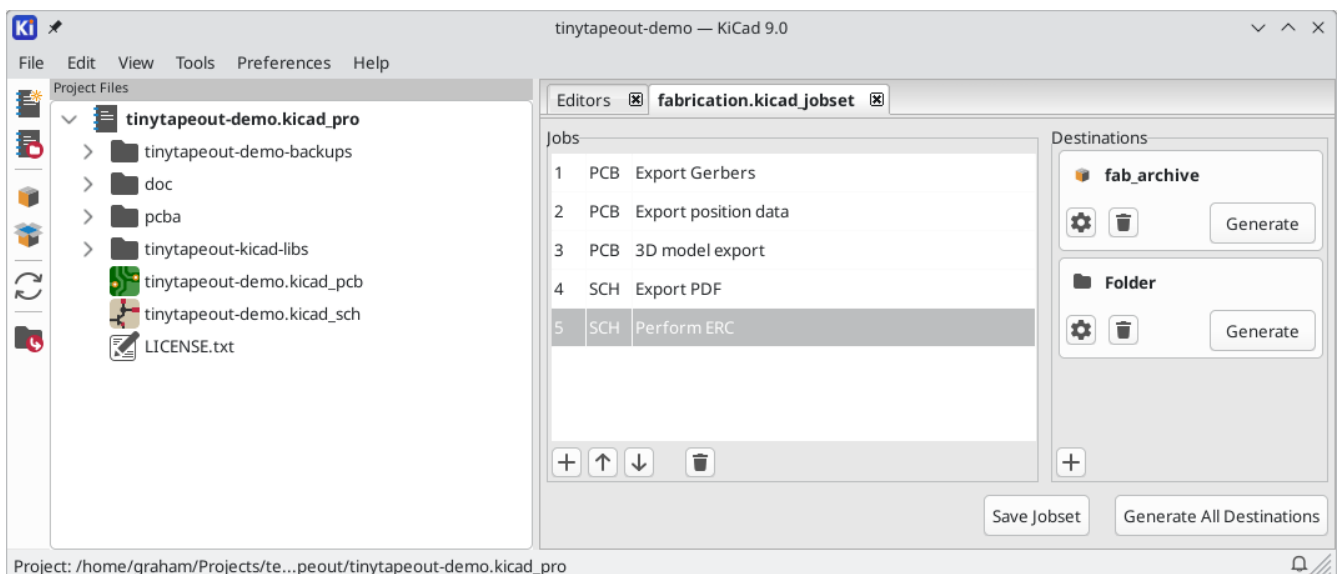
KiCad lets you configure a list of outputs that are all generated with a single click. The list of output jobs and the destinations where they will be saved is called a *jobset*. For example, a jobset might contain jobs to generate Gerber files, assembly data, a bill of materials, PDF plots of the schematic and PCB, while also running ERC and DRC checks, with all of the outputs saved to a compressed archive. The full list of available jobs is given [below](#).

Each *job* in a jobset defines a single type of generated output, such as a bill of materials or a set of Gerbers. A job can be configured in the same way as if the output was manually generated from the schematic or board editor. The configuration for each job is stored in the jobset and remembered when you load the jobset later. Jobs are configured individually, so if you include the same type of job multiple times in a single jobset, each job will have its own independent configuration. For example, this lets you generate PDF outputs in color as well as black and white.

In addition to the jobs, jobsets also contain *destinations*, which define a list of jobs to run and how to store their outputs. A jobset destination can simply store the chosen jobs' output files in a specified location, or it can add the output files to a compressed archive. Each jobset destination can select a different subset of jobs from the full list of jobs in the jobset. You can run each jobset destination individually or run all jobset destinations at once. As an example, you could set up one jobset destination that generates PDFs of the board and schematic and copies them to an external location, while another destination generates the fabrication files and compresses them in a zip archive to send to the board manufacturer.

Projects can have multiple jobsets, with each jobset defining a different list of jobs and output configurations. Each jobset is stored in a `.kicad_jobset` file, which can be specific to a single project, copied between projects, or even stored in a central location and shared between projects.

To use a jobset, first create a new jobset file in the KiCad project manager (**File** → **New Jobset File...**) and choose a name and location for it. Alternatively, you can open an existing jobset file with **File** → **Open Jobset File...** Jobset files that are stored in the project directory are considered part of the project and are displayed in the project file tree. You can open a jobset file in the project file tree by double clicking on it.



Once you create or open a jobset, it is displayed in a new tab in the project manager. The list of jobs is shown in the middle and the list of jobset destinations is shown on the right. New jobsets will not contain any jobs, but a destination is automatically created to save outputs to a folder. When you make changes to a jobset, you can save the changes by clicking the **Save Jobset** button.

Defining jobs

To add a new job, click the **+** button under the Jobs list. In the Add New Job dialog that appears, select the desired type of job. You can filter which types of jobs are shown in the list by typing in the **Filter** textbox at the bottom.

When you select a job and press **OK**, the settings dialog for that type of output will appear. Each job settings dialog provides the same options you would have if you manually generated that type of output from the schematic or board editor.

NOTE

Output filenames and paths specified in job settings are relative to the [jobset destination](#) folder or archive root. You can use certain [text variables](#), like `${PROJECTNAME}`, `${CURRENT_DATE}`, and [project text variables](#).

Export PDF Job Settings

Output file: `${PROJECTNAME}_r${REVISION}_${ISSUE_DATE}.pdf`

Options

Page size: Schematic size

☒ Plot drawing sheet

Output mode: Color

Color theme: KiCad Classic

☒ Plot background color

Minimum line width: 0 mils

HPGL Options

Position and units: Content fit, user units

Pen width: 0.004 mils

PDF Options

☒ Generate property popups

☒ Generate clickable links for hierarchical elements

☒ Generate metadata from AUTHOR & SUBJECT variables

Other Options

Cancel OK

When you accept the job settings dialog, the job is added to the list of jobs, where you can optionally change the new job's description from its default. To change a job's description or settings later, right click the job in the list and select **Edit Job Description** or **Edit Job Settings....** Double clicking on a job also edits its settings. To remove a job, select the job and click the **🗑** button. To reorder the list, select a job and move it up or down using the **↑** or **↓** buttons.

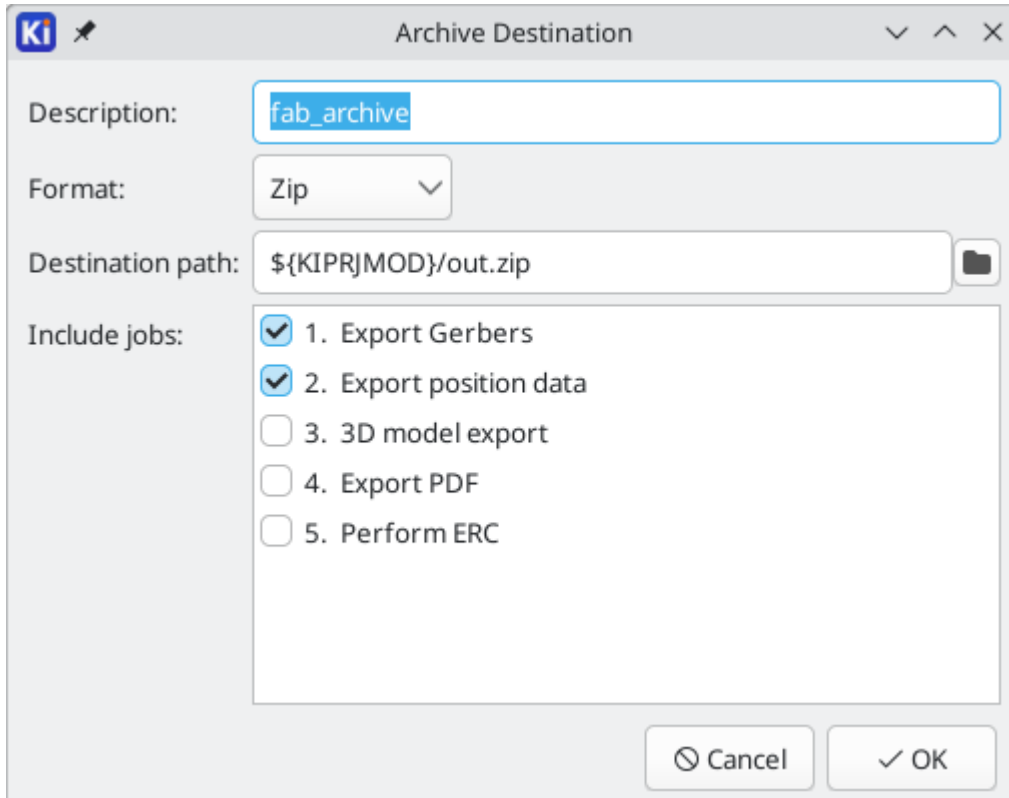
Defining jobset destinations

You cannot generate any outputs from a jobset until you add a jobset destination. One destination is created automatically when a jobset is created, but you can add as many destinations as you need.

To add a jobset destination, click the **+** button under the Destinations list. When the Add New Destination dialog appears, select a type of destination:

- **Archive** saves the outputs generated by the jobs in a compressed zip archive.
- **Folder** saves the outputs generated by the jobs uncompressed in a folder.

Once you have selected a type of output, the Destination options dialog appears.





The screenshot shows the 'Archive Destination' dialog box. It has a title bar with the KiCad logo and window controls. The dialog contains the following fields and options:

- Description:** A text field containing 'fab_archive'.
- Format:** A dropdown menu set to 'Zip'.
- Destination path:** A text field containing '\${KIPRJMOD}/out.zip' with a folder icon button to the right.
- Include jobs:** A list of five jobs with checkboxes:
 - ☒ 1. Export Gerbers
 - ☒ 2. Export position data
 - ☐ 3. 3D model export
 - ☐ 4. Export PDF
 - ☐ 5. Perform ERC

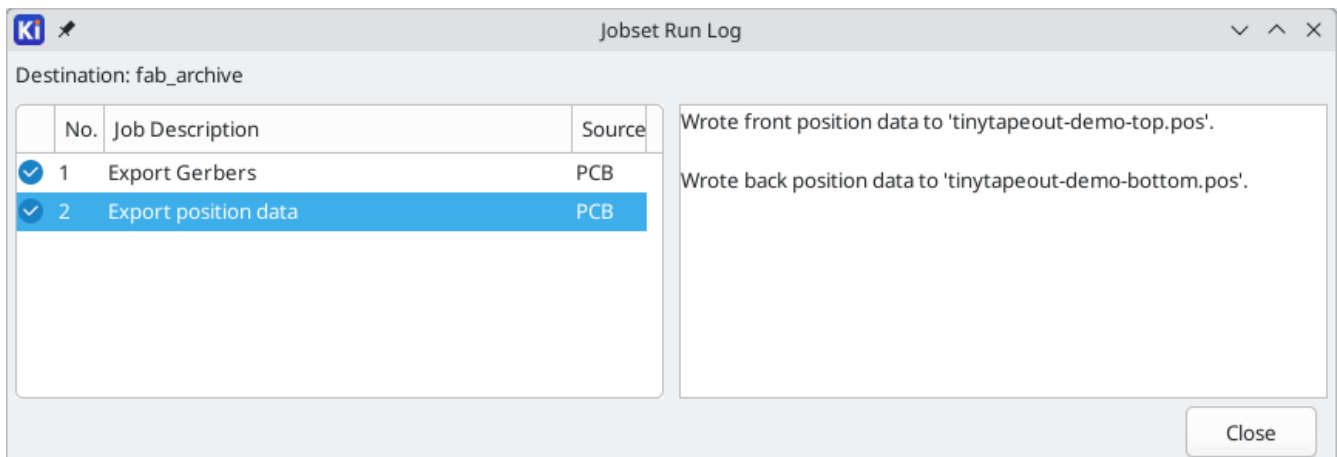
At the bottom right are 'Cancel' and 'OK' buttons.

Here you can select which jobs will be run as part of this jobset destination, as well as the folder or archive name that will be used to store them. By default all jobs are enabled. You can also set a description for the destination to be displayed in the Destinations list. The output path controls where the files generated by the jobs will be saved. The path here can be absolute or relative to the project directory, and it can use [path variables](#) or certain text variables (`${PROJECTNAME}`, `${CURRENT_DATE}`, and [project text variables](#)). Filenames defined in job configurations are relative to the jobset destination directory or archive root.

When you click **OK** in this dialog, the new jobset destination is added to the Destinations list. You can modify an existing jobset destination by clicking its  button, or remove it by clicking its  button.

After configuring your jobs and destinations, you can generate an individual set of outputs by clicking the **Generate** button for the desired destination. You can run all destinations at once by clicking the **Generate All Destinations** button.

If a jobset destination runs and generates its outputs successfully, a blue check is shown that indicates the last run was successful. If a jobset destination fails to complete successfully, a red exclamation point is shown to indicate the run was not successful. Clicking on the success/failure indicator will display the Jobset Run Log dialog, which displays the status of each job in the jobset destination. Clicking on a specific job will display the logged output from that job, if there is any.



Jobset destination details

When jobs run, output files are initially generated in a temporary folder. After all jobs in a jobset destination are completed, the output files are moved from the temporary folder to the folder or archive specified by the jobset destination. None of the outputs are moved until all of the jobs finish. Therefore, if a job needs to access files that are generated by another job in the same jobset destination, these files will be located in the temporary folder, not the final jobset destination.

KiCad defines the `${JOBSET_OUTPUT_WORK_PATH}` environment variable while jobs are running and sets it to the temporary path for the current jobset destination. You can use this environment variable if a job needs to be aware of the temporary folder's actual location. For example, if an Execute Command job runs a script that renames the outputs of another job, the script needs to know the temporary location of the outputs being renamed, before they are moved to the jobset destination. `${JOBSET_OUTPUT_WORK_PATH}` provides this location.

Available job types

The following types of jobs are available:

Job	Description
PCB: Export 3D Model	Exports a 3D model of the board. The model format can be STEP, GLB (binary glTF), XAO, BREP (OCCT), PLY, or STL.
PCB: Export Drill Data	Exports a drill file from the board.
PCB: Export DXF	Exports the board design to a DXF file .
PCB: Export Gerbers	Exports the board design to Gerber files , with one file per selected layer.
PCB: Export IPC-2581	Exports the board design in IPC-2581 format .
PCB: Export ODB++	Exports the board design in ODB++ format .
PCB: Export PDF	Exports the board design to PDF files , with one file per selected board layer. You can also generate a single PDF with multiple layers depending on the plot configuration.
PCB: Export Position Data	Exports a position (component placement) file from the board.

PCB: Export SVG	Exports the board design to a SVG file .
PCB: Perform DRC	Performs a Design Rule Check on the board and generates a report. If DRC violations are found, this job can optionally report a job failure.
PCB: Render	Generates a raytraced rendering of the 3D model of the board as a PNG or JPG file.
Schematic: Export DXF	Exports the schematic to a DXF file .
Schematic: Export HPGL	Exports the schematic to a HPGL file .
Schematic: Export Netlist	Exports a netlist from the schematic, with various formats available.
Schematic: Export PDF	Exports the schematic to a PDF file .
Schematic: Export Postscript	Exports the schematic to a PostScript file .
Schematic: Export SVG	Exports the schematic to a SVG file .
Schematic: Generate Bill of Materials	Exports a bill of materials from the schematic.
Schematic: Perform ERC	Performs an Electrical Rule Check on the schematic and generates a report. If ERC violations are found, this job can optionally report a job failure.
Special: Copy Files	Copies the specified file to the specified location. A failure to copy the files can optionally cause the output job to fail. You can control whether files in the output location should be overwritten or not.

Special: Execute Command	<p>Executes an arbitrary command. Output from the command can optionally be logged to a file. You can either ignore non-zero output codes or cause them to fail the output job.</p> <p>Note: Job output files are generated in a temporary folder, then moved to the location specified by the jobset destination after all jobs for that destination are executed. In other words, when an Execute Command job runs, the output from other jobs in the same jobset destination are still in a temporary location and not yet in the ultimate destination folder or archive. You can use the <code>\${JOBSET_OUTPUT_WORK_PATH}</code> environment variable if you need to refer to the temporary location in an Execute Command job (for example, in a script that renames files generated by another job).</p>
--------------------------	--

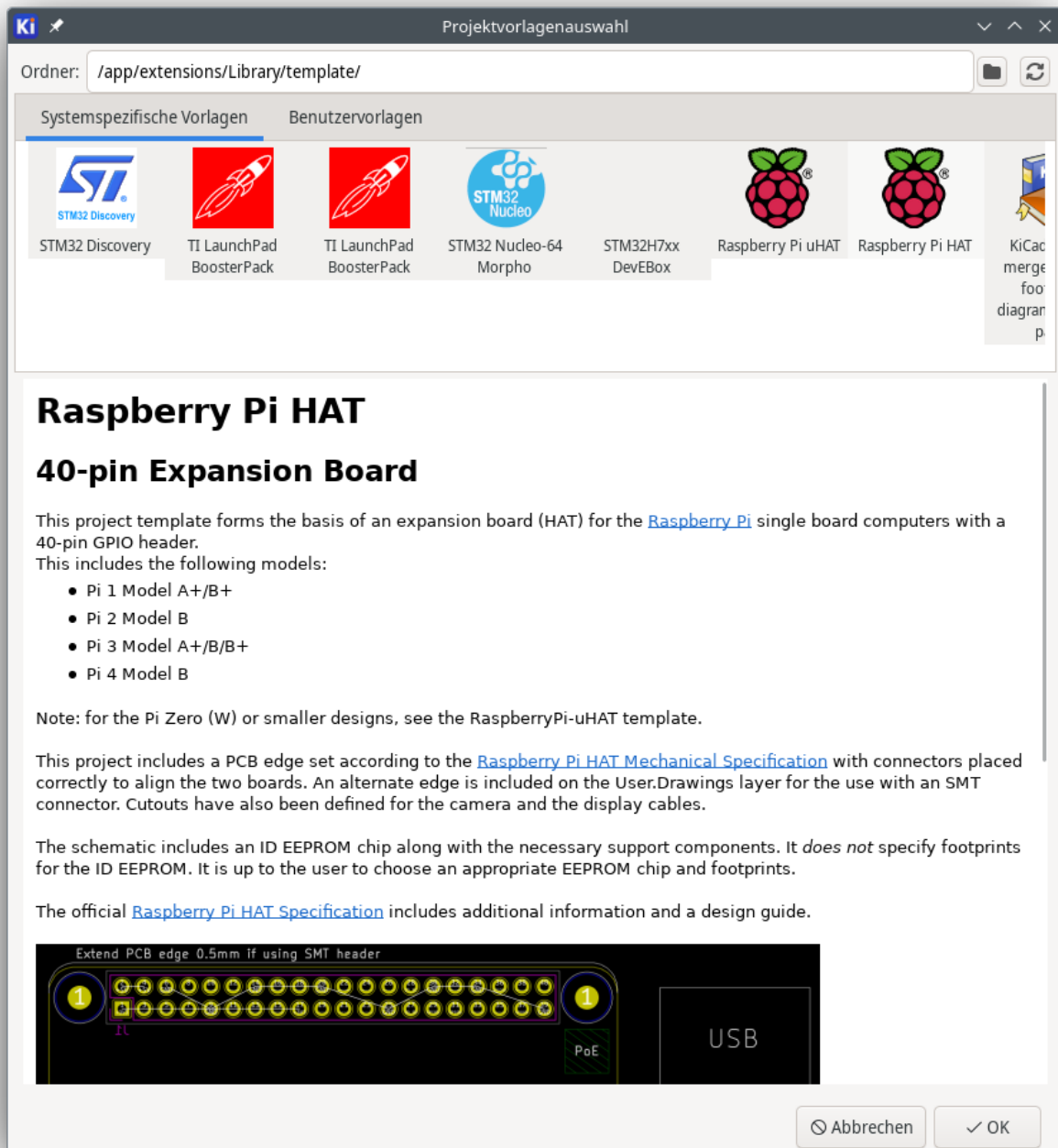
Projektvorlagen

Die Verwendung einer Projektvorlage erleichtert das Einrichten eines neuen Projekts mit vordefinierten Einstellungen. Vorlagen können vordefinierte Platinenumrisse, Steckerpositionen, Schaltplanelemente, Designregeln usw. enthalten. Es können sogar komplette Schaltpläne und/oder Leiterplatten enthalten sein, die als Startdateien für das neue Projekt dienen.

Verwendung von Vorlagen

Über das Menü **Datei** → **Neues Projekt aus einer Vorlage...** wird das Dialogfeld Projektvorlagenauswahl geöffnet:

Ein einzelner Klick auf das Symbol einer Vorlage zeigt die Vorlageninformationen an, und ein weiterer Klick auf die Schaltfläche OK erstellt das neue Projekt. Die Vorlagendateien werden in das neue Projekt kopiert und umbenannt, um den Namen des neuen Projekts zu übernehmen.



Speicherorte von Vorlagen

KiCad looks for system templates in the path defined in the `KICAD9_TEMPLATE_DIR` path variable, and user templates in the path defined in `KICAD_USER_TEMPLATE_DIR`. However, you can browse for templates in an arbitrary directory using the **Folder** control at the top of the dialog.

Erstellung von Vorlagen

Eine KiCad-Vorlage ist einfach ein Verzeichnis, das die Projektdateien der Vorlage sowie einige erforderliche Metadaten für die Vorlage in einem Unterverzeichnis namens `meta` enthält. Der Name des Verzeichnisses, das die Vorlagendateien enthält, bestimmt den Namen der Vorlage. Wenn Sie ein Projekt aus einer Vorlage erstellen, kopiert KiCad die Vorlagendateien in das neue Projektverzeichnis und benennt sie um, damit sie dem neuen Projektnamen entsprechen, wie unten beschrieben.

Alle Dateien der Vorlage werden kopiert, mit zwei Ausnahmen:

- Dateien, deren Namen mit dem Zeichen `.` beginnen (Punktdateien), werden nicht kopiert mit Ausnahme von Dateien mit dem Namen `.gitignore` oder `.gitattributes`. Diese werden kopiert, wenn sie existieren.
- Das Verzeichnis `meta` wird nicht kopiert

Das Verzeichnis `meta` muss eine HTML-Datei namens `info.html` enthalten, die im KiCad-Vorlagenbrowser angezeigt wird und grundlegende Informationen zur Beschreibung der Vorlage enthalten sollte. Grundlegende HTML-Funktionen werden unterstützt, einschließlich Bilder. Alle Bilder, auf die in der Datei `info.html` verwiesen wird, sollten ebenfalls im Verzeichnis `meta` gespeichert werden.

Der `<title>`-Tag bestimmt den Namen der Vorlage, der bei der Vorlagenauswahl angezeigt wird. Beachten Sie, dass der Name der Projektvorlage abgeschnitten wird, wenn er zu lang ist. Dieser Anzeigename muss nicht mit dem Namen des Vorlagenverzeichnisses übereinstimmen.

Hier ist ein Beispiel für die Datei `info.html` :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//DE">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=utf-8">
<TITLE>Raspberry Pi - Erweiterungsplatine</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>Diese Projektvorlage ist die Basis für eine Erweiterungsplatine für das <A
HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25 ARM board.</A> <BR>
<BR>Dieses Basisprojekt enthält die Form einer Leiterplatte, die die gleiche Größe hat wie
die Raspberry-Pi-Platine, wobei die Anschlüsse entsprechend richtig platziert sind, um die
beiden Platinen aufeinander abzustimmen. Alle IOs, die auf der Raspberry-Pi-Platine
vorhanden sind, werden mit dem Projekt über die 0.1" expansion headers angeschlossen
<BR><BR>Die Platine sieht wie folgt aus:</P>
<P><IMG SRC="brd.png" NAME="brd" ALIGN=BOTTOM WIDTH=680 HEIGHT=378
BORDER=0><BR><BR><BR><BR>
</P>
<P>(c)2012 Brian Sidebotham<BR>(c)2012 KiCad Developers</P>
</BODY>
</HTML>
```

Schließlich kann `meta` optional ein Bild mit dem Namen `icon.png` enthalten, das als Symbol für die Vorlage im Auswahldialog verwendet wird. Das Symbol sollte ein 64 x 64 Pixel großes PNG-Bild sein.

Umbenennen von Vorlagen

Alle Dateien und Ordner in einer Vorlage werden in den neuen Projektordner kopiert, sofern ein neues Projekt unter Verwendung dieser Vorlage erstellt wird. Davon ausgenommen ist das Verzeichnis **meta**. Dateien und Verzeichnisse, die den Verzeichnisnamen der Vorlage enthalten, werden mit dem neuen Projektdateinamen umbenannt.

Beispiel: Verwendung einer Vorlage mit dem Namen `example` (links) zur Erstellung eines Projekts mit dem Namen `newproject` (rechts), wobei die umbenannten Dateien in **fett** angezeigt werden:

Dateien im Vorlagenordner <code>example</code>	Dateien im neu erstellten Projektordner <code>newproject</code>
<code>example.kicad_pro</code> <code>example.kicad_sch</code> <code>example.kicad_pcb</code> <code>example-first.kicad_sch</code> <code>second-example.kicad_sch</code> <code>third.kicad_sch</code> <code>third.kicad_pcb</code>	<code>newproject.kicad_pro</code> <code>newproject.kicad_sch</code> <code>newproject.kicad_pcb</code> <code>newproject-first.kicad_sch</code> <code>second-newproject.kicad_sch</code> <code>third.kicad_sch</code> <code>third.kicad_pcb</code>

Eine Vorlage muss nicht unbedingt ein komplettes Projekt enthalten. Wenn eine erforderliche Projektdatei fehlt, erstellt KiCad die Datei mit dem Standardverhalten für das Erstellen von Projekten:

Dateien im Vorlagenordner <code>example</code>	Dateien im neu erstellten Projektordner <code>newproject</code>
<code>example.kicad_sch</code> <code>first-example.kicad_sch</code> <code>first-example.kicad_pcb</code> <code>second-example.kicad_sch</code> <code>second-example.kicad_pcb</code>	<code>newproject.kicad_sch</code> <code>first-newproject.kicad_sch</code> <code>first-newproject.kicad_pcb</code> <code>second-newproject.kicad_sch</code> <code>second-newproject.kicad_pcb</code> <code>newproject.kicad_pro</code> (default) <code>newproject.kicad_pcb</code> (default)

Wenn die Vorlage eine Projektdatei (`.kicad_pro`) enthält und deren Name nicht mit dem Vorlagenamen übereinstimmt, führt KiCad die Umbenennung stattdessen mit dem Namen der Projektdatei durch:

Dateien im Vorlagenordner <code>example</code>	Dateien im neu erstellten Projektordner <code>newproject</code>
<code>example.kicad_sch</code> <code>example.kicad_pcb</code> <code>first-example.kicad_pro</code> <code>first-example.kicad_sch</code> <code>first-example.kicad_pcb</code> <code>second-example.kicad_sch</code> <code>second-example.kicad_pcb</code>	<code>example.kicad_sch</code> <code>example.kicad_pcb</code> <code>newproject.kicad_pro</code> <code>newproject.kicad_sch</code> <code>newproject.kicad_pcb</code> <code>second-example.kicad_sch</code> <code>second-example.kicad_pcb</code>

NOTE

Es wird nicht empfohlen, eine Vorlage mit mehreren Projektdateien zu erstellen.

Plugin- und Content-Verwaltung

KiCad has a Plugin and Content Manager that lets you browse, install, and manage packages submitted by other users and organizations. Packages can be plugins offering specialized functionality, tools for ordering PCBs from specific manufacturers, libraries of symbols and footprints, or new editor color schemes. You can access the PCM by launching it from the main KiCad Project Manager window.

The PCM downloads packages from repositories on the internet. Each repository is a collection of packages that is managed by an individual or an organization. By default, the PCM uses a single repository managed by the KiCad organization. Users and organizations are free to create their own repositories, which other users can then add as additional repositories in their own KiCad installation. Third-party repositories can be public or private.

NOTE

Packages may install code that runs on your computer. Packages are not developed by the KiCad developers. The KiCad organization does not make any guarantees about the quality or safety of packages installed through the PCM. Make sure you only install packages that you trust.

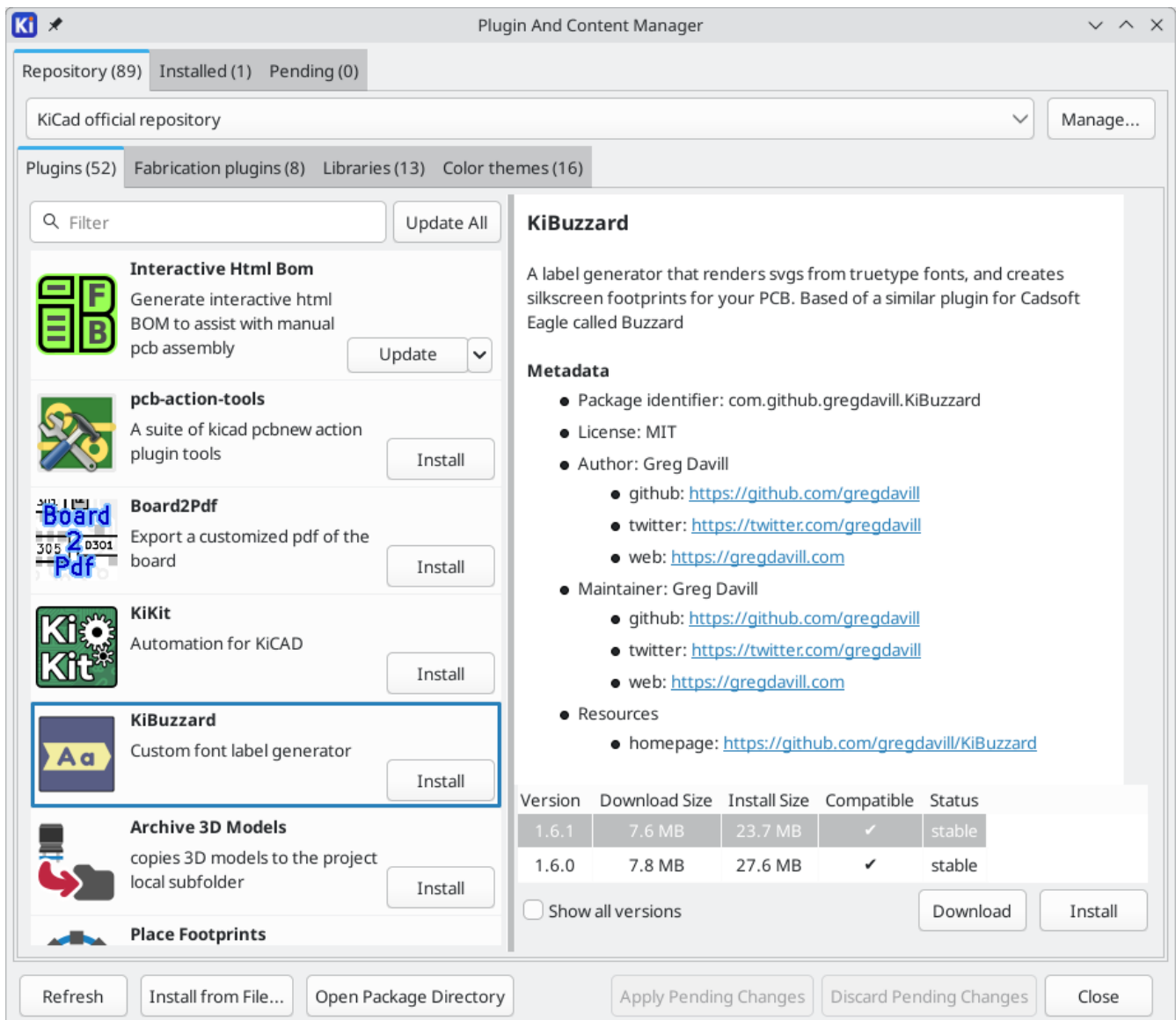
NOTE

If you have feedback regarding a package, please submit it to the package's developer, not the KiCad team. Contact information for each package is shown in the package's description.

To share a package that you have developed, you can submit it to the KiCad official repository. If it is accepted, it will then be available to other users browsing the KiCad repository. You can also create your own repository or submit your package to a repository managed by another person or group, but in that case users will not see your package unless they have configured that repository in their own installation. More information about creating your own packages and repositories is available [below](#).

Browsing packages

All of the packages in a repository are shown in the PCM's **Repository** tab. You can choose which repository to browse by selecting the repository in the dropdown at the top of the tab. By default, only the KiCad official repository is enabled.



Packages are grouped into four categories, each with its own tab within the **Repository** tab:

- **Plugins** are additional tools that can be launched from the PCB Editor. Plugins can have many purposes, for example modifying a board design or generating specific outputs. Footprint wizards can also be distributed as plugins.
- **Fabrication plugins** are a sub-category of plugins for ordering your PCBs from specific fabricators. These plugins may be a convenient way to order from a manufacturer, but they are typically not required; you can usually provide manufacturers with normal fabrication outputs instead. Consult with your manufacturer to find out the best way to order from them.
- **Libraries** contain symbols, footprints, and/or 3D models. By default, libraries installed by a library package are automatically added to the appropriate symbol and/or footprint library tables when the package is installed, and removed from the table when the package is uninstalled. Libraries installed by a package have configurable library name prefix (`PCM_` by default). These settings are configurable in the **Packages and Updates** section of Preferences.
- **Color themes** are color themes for the Schematic, Symbol, Board, and Footprint editors. You can select an installed theme in the **Colors** section of the Preferences for each editor.

The list on the left of the window shows the packages within each category. You can filter the list of packages by typing in the filter box at the top of the package list. Press the **Refresh** button to reload the list of packages from the online repository.

When you select a package in the list, information about the package is shown on the right. This includes a description of the package, the package's license, and contact information for the package's developer, including a place to report bugs and other feedback for the package.

The package information also includes a table of the versions of the package that are currently available. For each version, the table displays the package size, its status, and whether it is compatible with your version of KiCad.

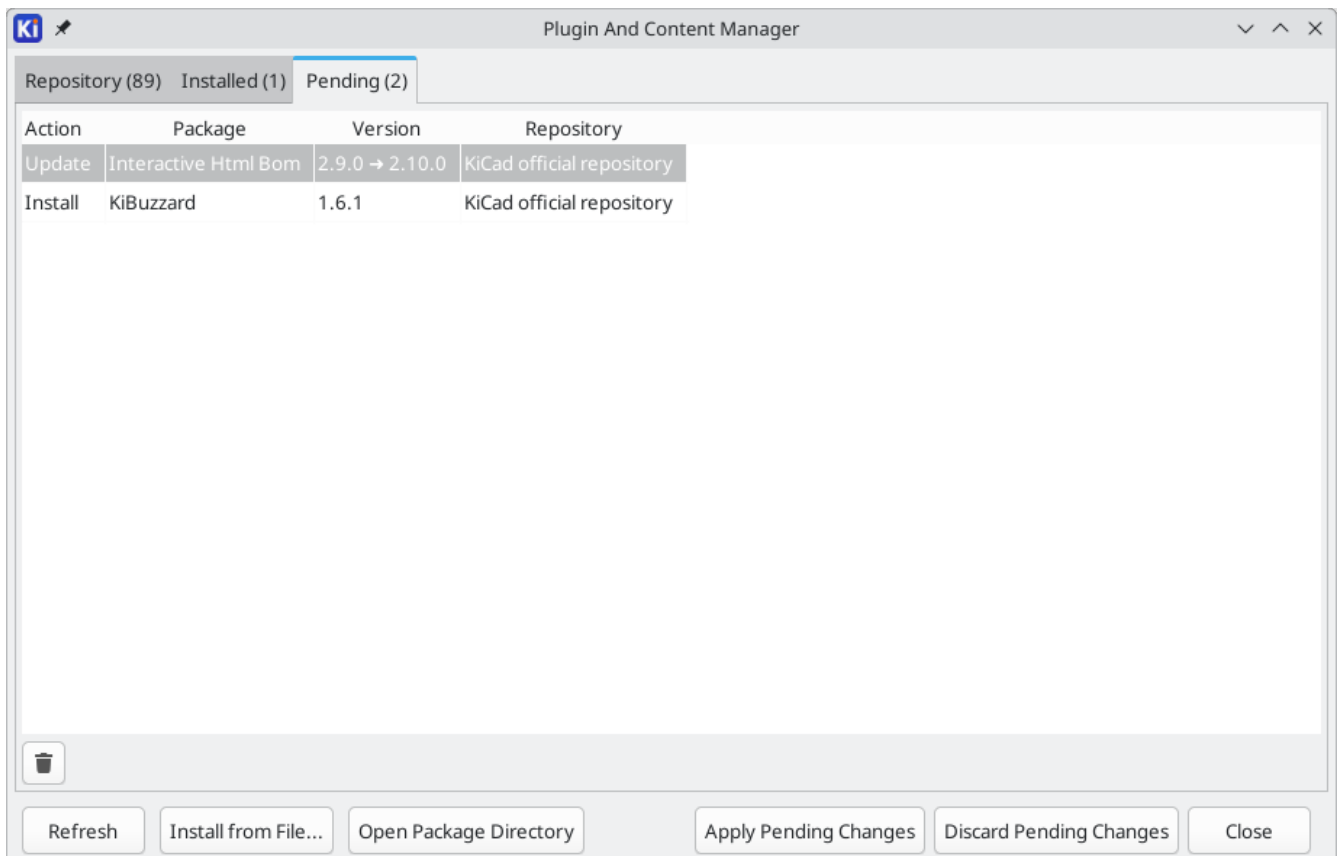
A package's status can be one of the following, as indicated by the package's developer:


- **Stable:** the package version is suitable for general use.
- **Testing:** the package version is in a testing phase; users should be cautious and report any issues they encounter to the package's developers.
- **Development:** the package version is in a development phase; users should not expect it to work fully.
- **Deprecated:** the package is no longer maintained.

By default, the version table only shows versions of the package that are compatible with your version of KiCad. You can show all versions of the package, even those that will not work with your KiCad version, by checking the **Show all versions** box. If multiple versions of a package are available, you can choose which to install.

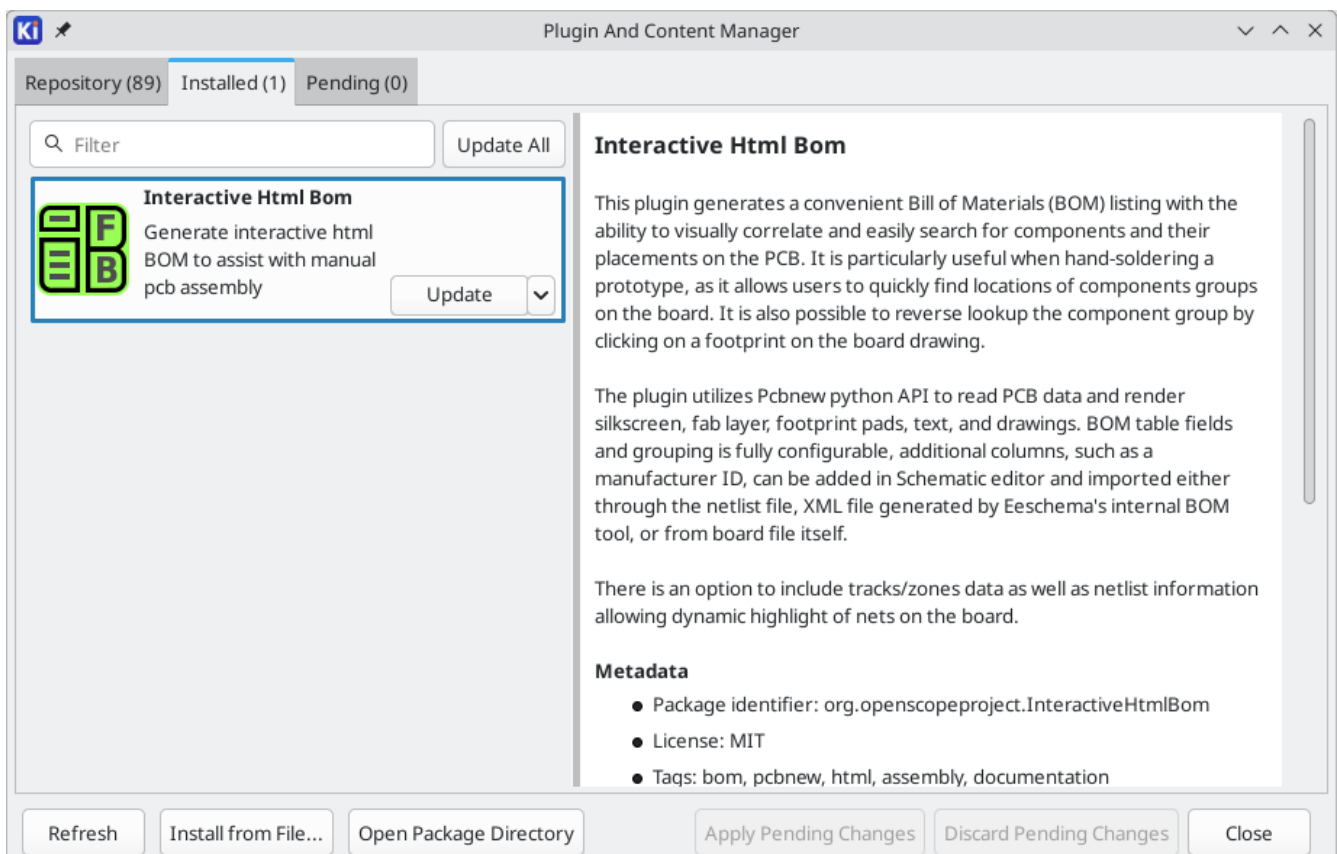
Installing packages

When you decide to install a package, click the package's **Install** button in the package's description page. If there are multiple versions of the package available, select the desired version in the version table before installing. You can also install the latest version by clicking the **Install** button in the list of packages. This does not immediately install the package; the package is instead queued for later installation and is shown in the **Pending** tab. The package will not be installed until you click the **Apply Pending Changes** button, which installs all pending packages at once.

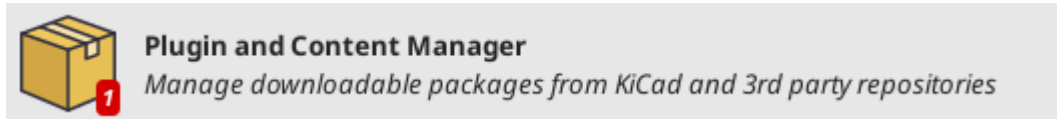


To remove a single package from the pending installation list, select it in the **Pending** tab and press the  button. To cancel the installation of all pending packages, click the **Discard Pending Changes** button.

Once a package is installed, it is listed in the **Installed** tab. In this tab you can see the list of installed packages and update or uninstall any of them.



If any packages have new versions available, the PCM icon in the Project Manager displays an indicator showing the number of available updates.



To update to a different version of a package, select the new version in its version table and click the **Update** button. You can also update a package to the newest version by clicking **Update** in the dropdown menu next to its **Uninstall** button in the list of installed packages. To update all installed packages, click the **Update All** button at the top of the installed package list. A package will not be updated if **Pin Package** is selected in its dropdown menu.

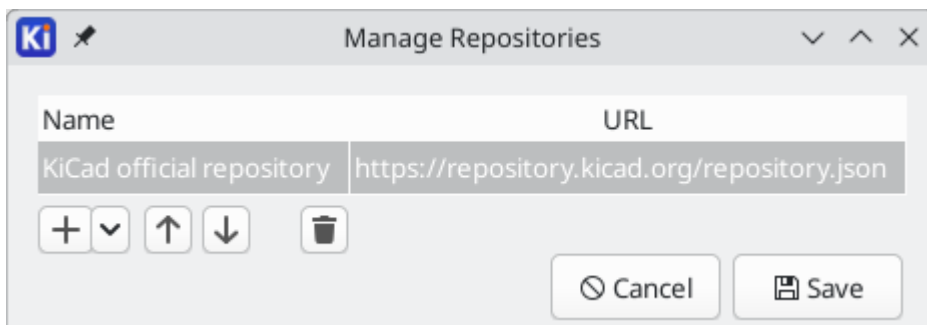
To uninstall a package, click its **Uninstall** button in the list of installed packages or in the package's description page.

As with installation, packages are not updated or uninstalled immediately when you click the **Update** or **Uninstall** button. Instead, the operation is queued in the **Pending** tab until you apply each change individually or click the **Apply Pending Changes** button.

Finally, you can download a package without installing it by clicking the **Download** button at the bottom of the package's description and selecting a location to save the package. This lets you inspect the files in the package before installing it. To install a package that has been downloaded but not yet installed, click the **Install from file...** button and select the package. The **Open Package Directory** button opens a file browser in the folder where KiCad installs packages.

Managing repositories

By default, only the KiCad official repository is used by the PCM. You can add a third party repository, or remove existing repositories, by clicking the **Manage...** button at the top of the **Repository** tab.



To add a repository, click the **+** button and specify the full URL to the repository. To remove a repository, select it and press the **🗑** button. Use the **↑** and **↓** buttons to reorder repositories in the list.

If you remove the default KiCad official repository, you can easily re-add it by clicking **Add Default Repository** in the dropdown menu next to the **+** button.

Creating packages and repositories

To create a package for the PCM, follow the instructions at <https://dev-docs.kicad.org/en/addons/index.html>. These instructions explain how to create PCM packages for any repository, including but not limited to the KiCad official repository. They also explain the rules for packages included in the KiCad repository and how

to submit packages to this repository. Third-party repositories use the same package format but may have different rules and procedures for submitting packages.

To create your own repository, publish a repository JSON file following the schema published at <https://go.kicad.org/pcm/schemas/v1>.

Referenz der Aktionen

Im Folgenden finden Sie eine Liste aller verfügbaren **Aktionen** im KiCad-Projektmanager: ein Befehl, der einer Tastenkombination zugewiesen werden kann.

KiCad-Projektverwaltung

Die folgenden Aktionen sind im KiCad-Projektmanager verfügbar. Tastenkombinationen können jeder dieser Aktionen im Abschnitt **Tastenkombinationen** der Voreinstellungen zugewiesen werden.

Aktion	Standard-Tastenkombination	Beschreibung
Projekt schließen		Das aktuelle Projekt schließen
Bildumwandler	Strg + B	Konvertiert Bitmapdateien zu Elementen des Schaltplans oder der Leiterplatte
Zeichnungsblatteditor	Strg + Y	Zeichnungsblattränder und Schriftfeld bearbeiten
Footprint-Editor	Strg + F	Bauteil-Footprints bearbeiten
Leiterplatteneditor	Strg + P	Leiterplatte bearbeiten
Schaltplaneditor	Strg + E	Schaltplan bearbeiten
Symboleditor	Strg + L	Schaltplansymbole bearbeiten
Projekt von Repository klonen...		Projekt von einem bestehenden Repository klonen
Neues Projekt aus einer Vorlage...	Strg + T	Ein neues Projekt aus einer Vorlage erstellen
Neues Projekt...	Strg + N	Neues leeres Projekt erstellen
Demoprojekt öffnen...		Ein Demoprojekt öffnen
Projekt öffnen...	Strg + O	Ein vorhandenes Projekt öffnen
Texteditor öffnen		Bevorzugten Texteditor starten
Plugin- und Content-Verwaltung	Strg + M	Plugin- und Content-Verwaltung starten
Berechnungswerkzeuge		Berechnen von Bauteilen, Leiterbahnbreiten und vieles mehr
Gerber-Viewer	Strg + G	Gerber-Ausgabedateien anzeigen