

KiCad

The KiCad Team

KiCad

The KiCad Team

Indice

.....	iv
1. Introduzione	1
1.1. Requisiti di sistema	1
1.2. Cartelle e file di KiCad	1
1.2.1. File di progetto	1
1.2.2. File dell'editor degli schemi elettrici	1
1.2.3. File e cartelle dell'editor schede	2
1.2.4. File comuni	2
1.2.5. File di fabbricazione e documentazione	3
1.2.6. Memorizzazione e invio file KiCad	3
2. Installazione e aggiornamento KiCad	4
2.1. Importazione impostazioni	4
2.2. Migrazione di file da versioni precedenti	5
3. Using the KiCad project manager	6
3.1. Standalone mode	7
3.2. Creazione di un nuovo progetto	7
3.3. Importazione di un progetto da un altro strumento EDA	8
3.4. Saving and loading project archives	9
3.5. Git integration	10
4. Configurazione KiCad	13
4.1. Preferenze comuni	13
4.2. Preferenze di mouse e touchpad	15
4.3. Preferenze comandi da tastiera	16
4.4. Configurazione percorsi	16
4.4.1. KiCad path variables	17
4.4.2. Variabili ambiente avanzate	18
4.5. Configurazione librerie	18
5. Jobsets	20
5.1. Defining jobs	21
5.2. Defining jobset destinations	22
5.3. Available job types	23
6. Modelli di progetto	25
6.1. Uso dei modelli	25
6.2. Posizioni modelli	25
6.3. Creazione dei modelli	26
6.3.1. Template file renaming	27
7. Plugin e gestore contenuti	28
8. Elenco di riferimento azioni	29
8.1. Gestore progetti di KiCad	29

Copyright

This document is Copyright © 2010-2024 by its contributors as listed below. You may distribute it and/or modify it under the terms of either the GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), version 3 or later, or the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), version 3.0 or later.

Tutti i marchi registrati all'interno di questa guida appartengono ai loro legittimi proprietari.

Collaboratori

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

Traduzione

Marco Ciampa <ciampix@posteo.net [<mailto:ciampix@posteo.net>]>, 2014-2018.

Commenti

The KiCad project welcomes feedback, bug reports, and suggestions related to the software or its documentation. For more information on how to submit feedback or report an issue, please see the instructions at <https://www.kicad.org/help/report-an-issue/>

Capitolo 1. Introduzione

KiCad è una suite software open source per la creazione di schemi elettrici e circuiti stampati (PCB). KiCad supporta un flusso di lavoro di progettazione integrato in cui uno schema e un C.S. corrispondente sono progettati insieme, nonché flussi di lavoro autonomi per usi speciali. KiCad include anche diverse utilità per aiutare con la progettazione di circuiti e C.S., tra cui una calcolatrice per determinare le proprietà elettriche delle strutture dei circuiti, un visualizzatore di file Gerber per l'ispezione dei file di produzione e un simulatore SPICE integrato per la verifica del comportamento del circuito elettrico.

KiCad funziona su tutti i principali sistemi operativi e su un'ampia gamma di computer. Supporta C.S. con un massimo di 32 strati di rame ed è adatto per creare progetti di ogni complessità. KiCad è sviluppato da un team di volontari di software ed ingegneri in tutto il mondo con la missione di creare software di progettazione elettronica gratuito e open source adatto a progettisti professionisti.

L'ultima versione di questa documentazione è disponibile all'indirizzo <https://docs.kicad.org>.

1.1. Requisiti di sistema

KiCad è in grado di funzionare su un'ampia varietà di macchine e di sistemi operativi, ma alcune attività potrebbero essere più lente o più difficili su dispositivi di fascia bassa. Per un'esperienza ottimale, si consiglia una scheda grafica e uno schermo dedicati con risoluzione 1920x1080 o superiore.

Controllare sul sito Web di KiCad per i requisiti di sistema più recenti: <https://kicad.org/help/system-requirements/>

1.2. Cartelle e file di KiCad

KiCad crea e usa file con le seguenti estensioni (e cartelle) per la modifica di schemi e schede.

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_prl`) or the `fp-info-cache` file. Files that are unnecessary to distribute are noted in the table below.

1.2.1. File di progetto

<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.

1.2.2. File dell'editor degli schemi elettrici

<code>*.kicad_sch</code>	Schematic files, containing all symbol and connection information.
--------------------------	--

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

1.2.3. File e cartelle dell'editor schede

*.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.

1.2.4. File comuni

*.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.

1.2.5. File di fabbricazione e documentazione

*.gbr	File Gerber, per la fabbricazione
*.drl	File di foratura (formato Excellon), per la fabbricazione
*.pos	File di posizionamento (formato ASCII), per le macchine per l'inserzione automatica
*.rpt	File di rapporto (formato ASCII), per documentazione
*.ps	File di tracciatura (formato Postscript), per documentazione
*.pdf	File di tracciatura (formato PDF), per documentazione
*.svg	File di tracciatura (formato SVG), per documentazione
*.dxf	File di tracciatura (formato DXF), per documentazione
*.plt	File di tracciatura (formato HPGL), per documentazione

1.2.6. Memorizzazione e invio file KiCad

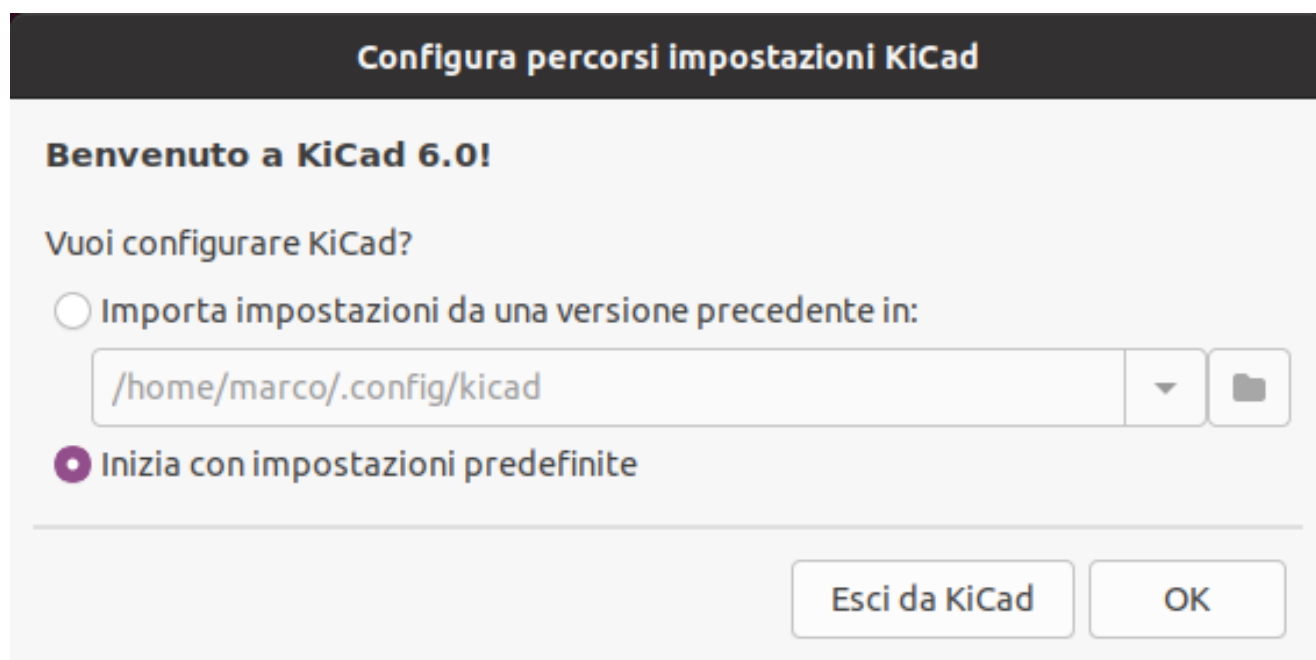
I file di schemi elettrici e C.S. di KiCad contengono tutti i simboli schema e le impronte utilizzati nel progetto, quindi si può eseguire il salvataggio o inviare questi file senza problemi. Alcune importanti informazioni sul progetto sono memorizzate nel file di progetto (.kicad_pro), quindi se si sta inviando un progetto completo, bisogna assicurarsi di includerlo.

Some files, such as the project-local settings file (.kicad_prl) 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.

Capitolo 2. Installazione e aggiornamento KiCad

2.1. Importazione impostazioni

Ogni versione principale di KiCad ha la propria configurazione, in modo che si possa eseguire più versioni di KiCad sullo stesso computer senza che le configurazioni interferiscano fra loro. La prima volta che si esegue una nuova versione di KiCad, verrà chiesto come inizializzare le impostazioni:



Se viene rilevata una versione precedente di KiCad, si avrà la possibilità di importare le impostazioni da quella versione. La posizione dei file di configurazione precedenti viene rilevata automaticamente, ma è possibile ignorarla per scegliere un'altra posizione, se lo si desidera.

Si noti che **non** verranno importate le tabelle delle librerie di simboli e di impronte dalle precedenti versioni di KiCad.

Si può anche scegliere di iniziare con le impostazioni predefinite se non si desidera importare le impostazioni da una versione precedente.

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

2.2. Migrazione di file da versioni precedenti

Le versioni moderne di KiCad possono aprire file creati da versioni precedenti, ma possono solo scrivere file nei formati più recenti. Ciò significa che in generale non ci sono passaggi speciali per migrare i file da una versione precedente oltre all'apertura dei file. In alcuni casi, l'estensione di un file viene modificata da una versione di KiCad a quella successiva. Dopo aver aperto questi file, verranno salvati nel nuovo formato con la nuova estensione del file. I vecchi file non verranno eliminati automaticamente.

The schematic editor documentation describes several particular considerations for opening legacy schematics.

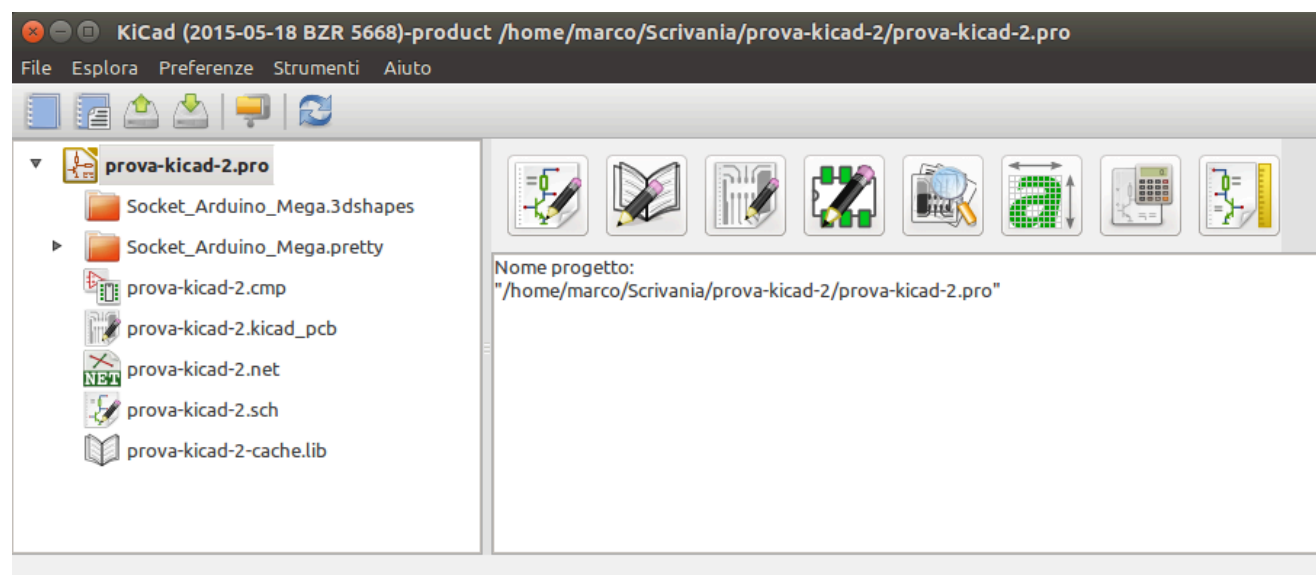
In generale, i file creati o modificati da una versione di KiCad **non** possono essere aperti da versioni precedenti di KiCad. Per questo motivo, è importante conservare copie di salvataggio dei propri progetti durante le prove di una nuova versione di KiCad, finché non si è sicuri di non dover più utilizzare la versione precedente di KiCad.

Nota

Attualmente le configurazioni dei tasti di scelta rapida non vengono importate dalle versioni precedenti. Si può manualmente importare le configurazioni dei tasti di scelta rapida copiando i vari file `*.hotkeys` dalla cartella di configurazione della vecchia versione in quella nuova. Nel caso, tenere presente che KiCad non rileva automaticamente i conflitti come l'assegnazione di un tasto a più azioni.







Capitolo 3. Using the KiCad project manager

The KiCad project manager is a tool which creates and opens KiCad projects and launches the other KiCad tools (schematic and board editors, Gerber viewer, and utility tools).

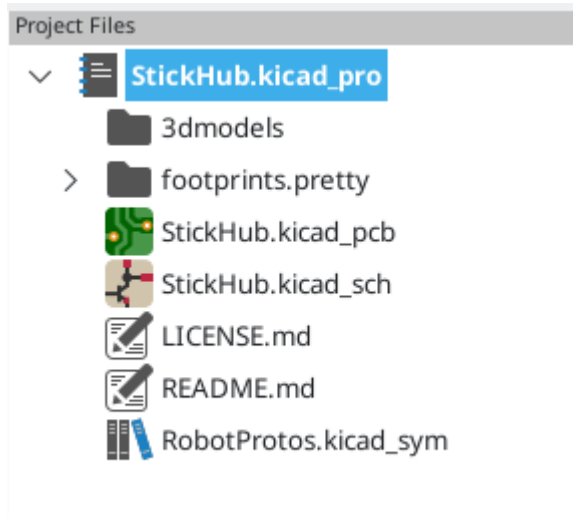


La finestra del gestore progetti di KiCad è composta da una vista ad albero sulla sinistra che mostra i file associati al progetto aperto, e un pannello di avvio sulla destra contenente i pulsanti da usare per eseguire vari editor e strumenti software.

La barra degli strumenti sul lato sinistro della finestra fornisce scorciatoie per le operazioni di progetto comuni:

	Create a new project.
	Open an existing project.
	Create a zip archive of the whole project. This includes schematic files, libraries, PCB, etc.
	Extract a project zip archive into a directory. Files in the destination directory will be overwritten.
	Refresh the tree view, to detect changes made on the filesystem.
	Open the project working directory in a file explorer.

The tree view shows a list of files inside the project folder. 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.



Nota

Solo i file che KiCad sa come aprire vengono visualizzati nella vista ad albero del progetto.

KiCad projects contain at least a project file, a schematic, and a board design. Schematics may contain multiple sheets, each in its own file, but a project can only contain a single board. KiCad expects the project file, schematic root sheet file, and board file to all have the same name.

3.1. Standalone mode

You can also run the KiCad editor tools in *standalone* mode, by launching them directly from your operating system's application launcher rather than from the project manager. It is usually **not recommended** to run the tools in standalone mode, except for some specific situations where it is necessary, such as when importing projects from other EDA tools. When running in standalone mode, some project features are not available, including:

- cross probing between the schematic editor and the board editor
- design synchronization between the schematic and the board

3.2. Creazione di un nuovo progetto

La maggior parte dei progetti KiCad si iniziano con la creazione di un progetto. Esistono due modi per creare un progetto dalla gestione progetti di KiCad: si può creare un progetto vuoto o creare un progetto basato su un modello esistente. Questa sezione tratterà la creazione di un nuovo progetto vuoto. La creazione di progetti da modelli è trattata nella sezione Modelli di progetto.

Per creare un nuovo progetto, usare il comando **Nuovo progetto...** nel menu **File**, fare clic sul pulsante **Nuovo progetto** nella barra degli strumenti in alto o usare la scorciatoia da tastiera (**Ctrl+N** per impostazione predefinita).

Ti verrà chiesto un nome da assegnare al progetto. Per impostazione predefinita, verrà creata una directory per il tuo progetto con lo stesso nome. Ad esempio, se si inserisce il nome `MioProgetto`,

KiCad creerà la cartella `MioProgetto` e il file di progetto `MioProgetto/MioProgetto.kicad_pro` al suo interno.

Se si dispone già di una cartella in cui archiviare i file del progetto, si può deselectare la casella di controllo *Crea una nuova cartella per il progetto* nella finestra di dialogo **Nuovo progetto**.

Nota

È fortemente raccomandato memorizzare ogni progetto KiCad dentro la corrispondente cartella.

Una volta selezionato il nome del progetto, KiCad creerà i seguenti file all'interno della cartella del progetto:

<code>esempio.kicad_pro</code>	File progetto KiCad.
<code>esempio.kicad_sch</code>	File schema elettrico principale.
<code>esempio.kicad_pcb</code>	File del circuito stampato.

3.3. Importazione di un progetto da un altro strumento EDA

KiCad is able to import files created by some other software packages. Some software formats can be imported as complete projects. Others can only be imported as standalone schematics or boards at the moment, and must be manually linked together into a KiCad project. Currently the following types of project are supported:

<code>*.sch, *.brd</code>	Eagle 6.x or newer (XML format)
<code>*.csa, *.cpa</code>	CADSTAR archive format
<code>*.zip</code>	EasyEDA (JLCEDA) Standard Backup
<code>*.epro, *.zip</code>	EasyEDA (JLCEDA) Pro Project

Per importare un progetto da uno di questi strumenti, scegliere l'opzione appropriata dal sottomenu **Importa progetto non KiCad** del menu **File**.

Verrà richiesto di selezionare un file di schema elettrico o di un circuito stampato nella finestra di dialogo della selezione dei file di importazione. I file dello schema e dello stampato devono avere lo stesso nome del file di base (ad es. `progetto.sch` e `progetto.brd`). Una volta selezionati i file richiesti, verrà chiesto di selezionare una cartella in cui memorizzare il progetto KiCad risultante.

Currently the following types of documents can be imported standalone. To import these documents, start the KiCad Schematic Editor or PCB Editor standalone (do not open the KiCad project manager first) and choose **File > Import > Non-KiCad Schematic** or **File > Import > Non-KiCad Board File**. When importing Altium projects, we recommend importing the PCB first, saving the resulting project, and then copying each schematic sheet into the project after importing it in a standalone Schematic Editor window.

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

Nota

KiCad does not support schematics with multiple top-level sheets. When importing designs from other tools that do support this feature, each schematic sheet must be imported, and then the imported sheets must be placed as hierarchical sheets in a new KiCad project.

3.4. Saving and loading project archives

You can archive your project's files into a zip archive with the Archive tool (**File → Archive Project...**).

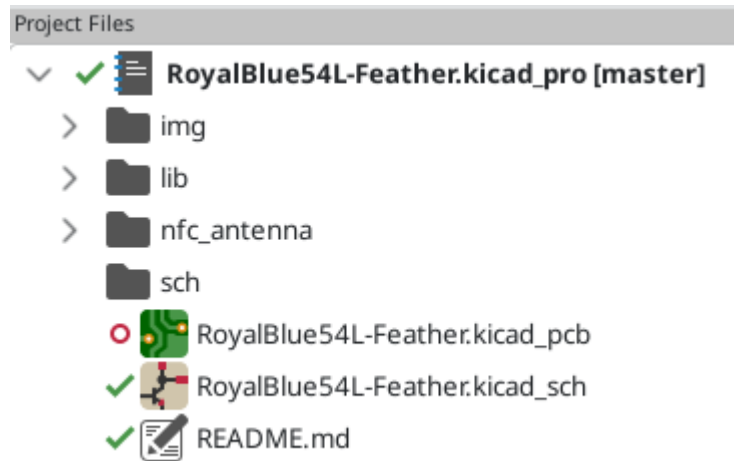
You can also unarchive a project using the Unarchive tool (**File → Unarchive Project...**). When you unarchive a project into the currently loaded project directory, the project will be reloaded automatically to reflect any changes that were in the archived version of the project.

The archive tool saves the following files from your project folder into the archive:

*.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	KiCad design files
*.pro, *.sch, *.lib, *.dcm, *.cmp, *.brd, *.mod	Legacy KiCad design files
*.stp, *.step	3D models
*.g?, *.g??, *.gm??, *.gbrjob	Gerber files
*.pos, *.drl, *.nc, *.xnc, *.d356, *.rpt	Manufacturing files
*.net	Netlists
*.py	Python scripts
*.pdf, *.txt	Documentation files
*.cir, *.sub, *.model	SPICE models
*.ibs, *.pkg	IBIS models

3.5. 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,



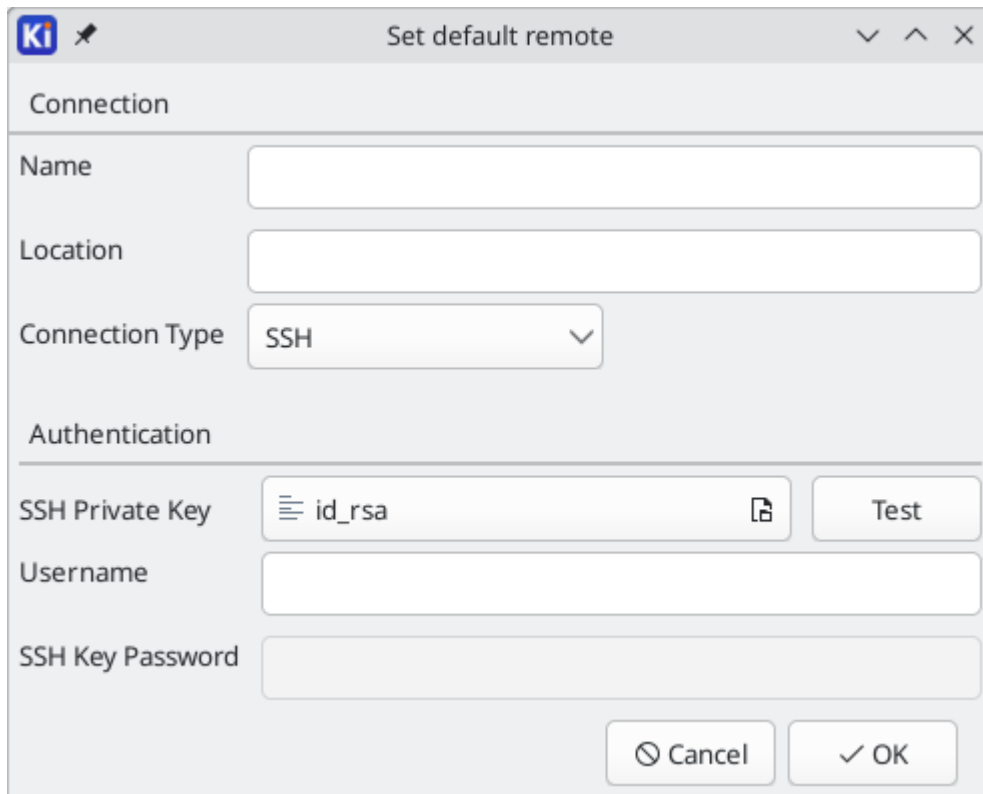
indicates a file has uncommitted changes, and



indicates a file is not tracked. No icons are shown if the project is not part of a Git repository.

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:

- **Name:** A name for the repository. This field can be anything and is not used.
- **Location:** The URL or file path to the remote.
- **Connection Type:** The protocol for connecting to the remote. This can be HTTPS, SSH, or local (file). HTTPS connections use username and password authentication. SSH connections use a username, private key, and an optional password for the keyfile. Local connections do not use authentication. You can check the connection and authentication by clicking the **Test** button.



The screenshot shows a dialog box titled "Set default remote" with the KiCad logo in the top-left corner. The dialog is organized into two main sections: "Connection" and "Authentication".

Connection Section:

- Name:** A text input field.
- Location:** A text input field.
- Connection Type:** A dropdown menu currently showing "SSH".

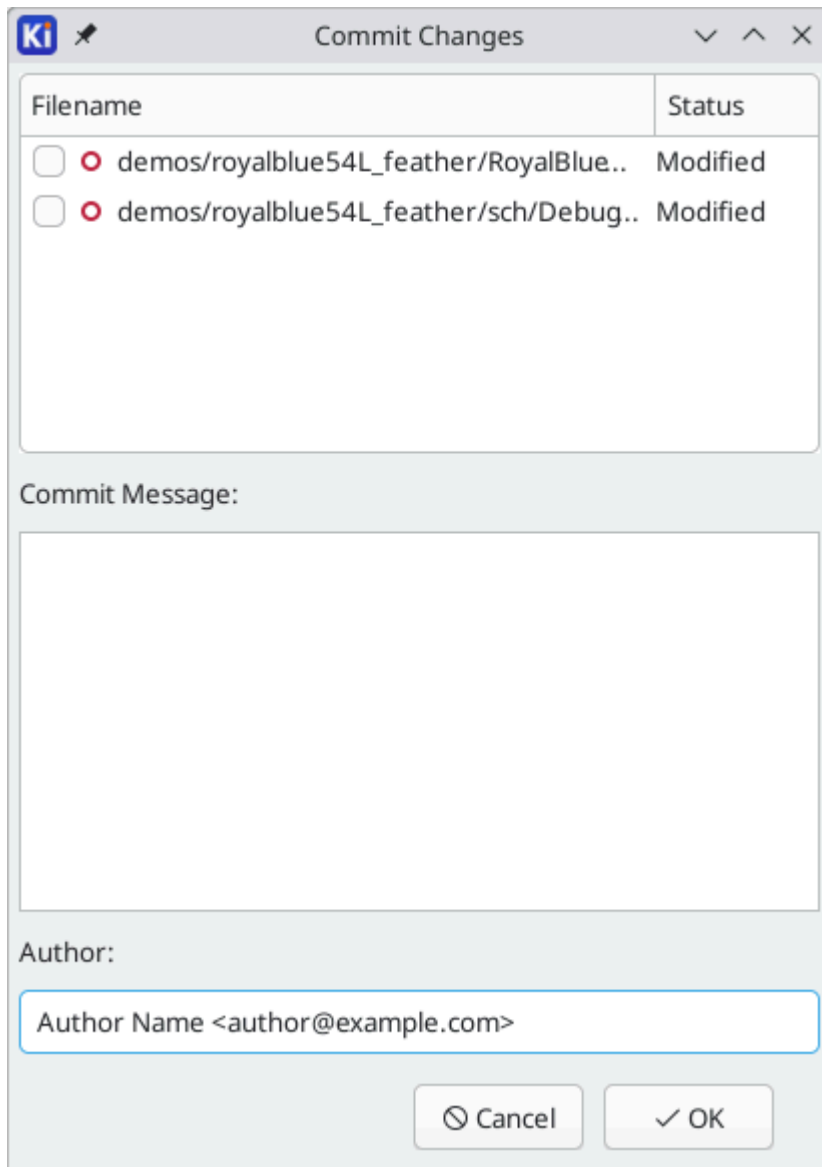
Authentication Section:

- SSH Private Key:** A text field containing "id_rsa" with a file selection icon to its right and a "Test" button.
- Username:** A text input field.
- SSH Key Password:** A text input field.

At the bottom of the dialog are two buttons: "Cancel" and "OK".

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.

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.



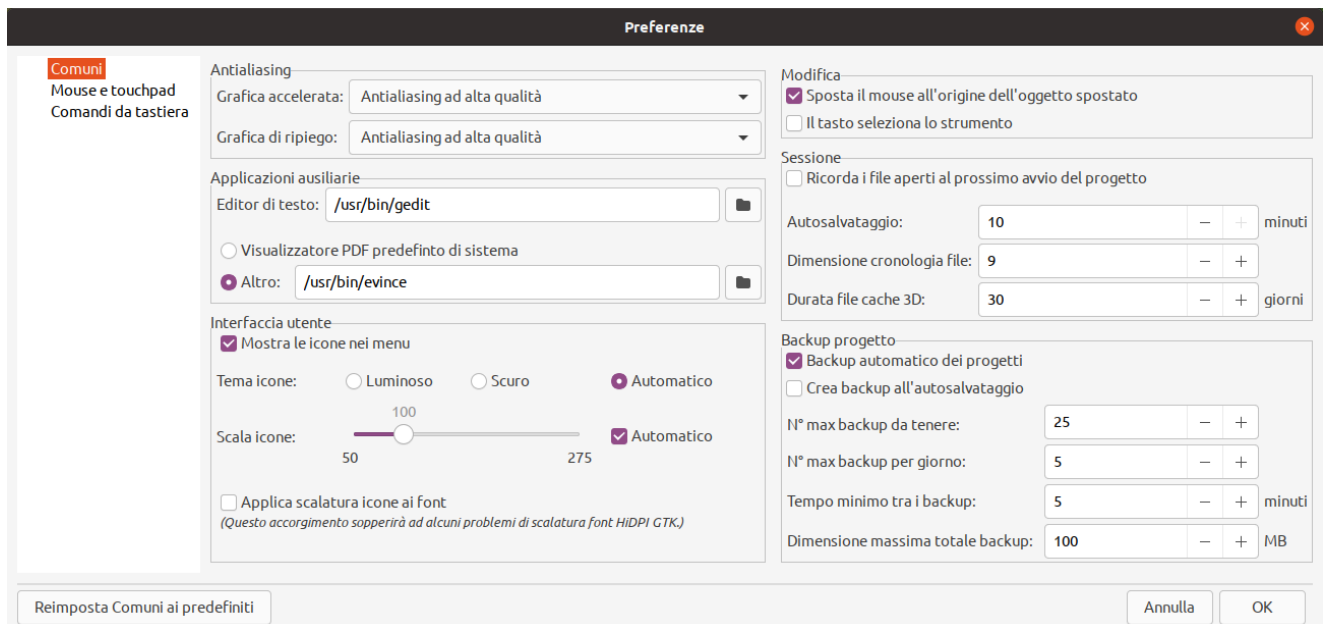
To push changes to the remote, right click in the project files tree and select **Version Control** → **Push**. To pull from the remote, right click and select **Version Control** → **Pull**. You can switch branches by selecting the desired branch from the **Version Control** → **Switch to Branch** menu.

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**.

Capitolo 4. Configurazione KiCad

The KiCad preferences can always be accessed from the **Preferences** menu, or by using the hotkey (default kbd:[Ctrl+,]). The Preferences dialog is shared between the running KiCad tools. Some preferences apply to all tools, and some are specific to a certain tool (such as the schematic or board editor).

4.1. Preferenze comuni



Antialiasing grafico accelerato: KiCad può utilizzare diversi metodi per prevenire l'aliasing (linee frastagliate) durante il rendering utilizzando una scheda grafica. Diversi metodi possono avere un aspetto migliore su hardware diverso, quindi è meglio sperimentare per trovare quello che sembra migliore.

Antialiasing grafica di ripiego: KiCad può fare l'antialiasing anche in modalità grafica di ripiego. L'abilitazione di questa funzione però può comportare prestazioni scadenti su alcuni dispositivi.

Editor di testo: scegli un editor di testo da usare per aprire i file di testo dalla finestra dell'albero del progetto.

Visualizzatore PDF: scegli un programma da usare per aprire i file PDF.

Mostra le icone nei menu: abilita le icone nei menu a discesa in tutta l'interfaccia utente di KiCad.

Nota

Le icone nei menu non vengono comunque visualizzate in alcuni sistemi operativi.

Show scrollbars in editors: When enabled, scrollbars are displayed next to the editing canvases in each tool. When disabled, scrollbars are not shown.

Il focus segue il mouse tra gli editor dello schema e del C.S.: Quando abilitata, la finestra sotto il puntatore del mouse prenderà il focus automaticamente.

Scala icona: imposta la dimensione delle icone usate nei menu e nei pulsanti in tutto KiCad. Scegliere *Automatico* per fare in modo che scelga automaticamente una scala icone appropriata in base alle impostazioni del proprio sistema operativo.

Tema icone: imposta se usare il tema icone progettato per gli sfondi finestre chiari o per quelli scuri. L'impostazione predefinita automatica sceglierà il tema in base alla luminosità del tema delle finestre del sistema operativo.

Fattore di attenuazione modalità alto contrasto: imposta la quantità di oscuramento degli elementi non a fuoco nella modalità di visualizzazione ad alto contrasto.

Sposta il mouse all'origine dell'oggetto spostato: se abilitata, il cursore del mouse verrà riposizionato all'origine di un oggetto quando si avvia un comando di spostamento su quell'oggetto.

Il tasto seleziona la strumento: se disabilitata, eseguendo un comando da tastiera come *Aggiungi filo* inizierà immediatamente il comando alla posizione corrente del puntatore. Se abilitata, premendo il comando la prima volta selezionerà solamente lo strumento *Aggiungi filo* ma non comincerà immediatamente un filo.

Ricorda i file aperti per il prossimo lancio del progetto: se abilitata, KiCad riaprirà automaticamente tutti i file che erano stati aperti in precedenza alla riapertura di un progetto.

Salvataggio automatico: durante la modifica di file schemi e schede, KiCad può salvare il lavoro periodicamente in automatico. Impostare su 0 per disabilitare questa funzione.

Dimensione cronologia file: configura il numero massimo di voci nell'elenco dei file aperti recentemente.

Durata file cache 3D: KiCad crea una cache di modelli 3D per velocizzare il visualizzatore 3D. Si può configurare per quanto tempo conservare questa cache prima di eliminare i vecchi file.

Backup automatico dei progetti: se abilitata, i progetti KiCad verranno archiviati automaticamente in file ZIP in base alle impostazioni di seguito. Gli archivi verranno memorizzati in una sottocartella della cartella del progetto. I backup vengono creati quando si effettua il salvataggio dei file nel progetto.

Crea backup all'autosalvataggio: se abilitata, verrà creato un backup ogni volta che si verifica un salvataggio automatico file (se il backup è consentito dalle impostazioni di seguito). Questa impostazione non ha effetto se l'intervallo di salvataggio automatico è impostato su 0 (disabilitato).

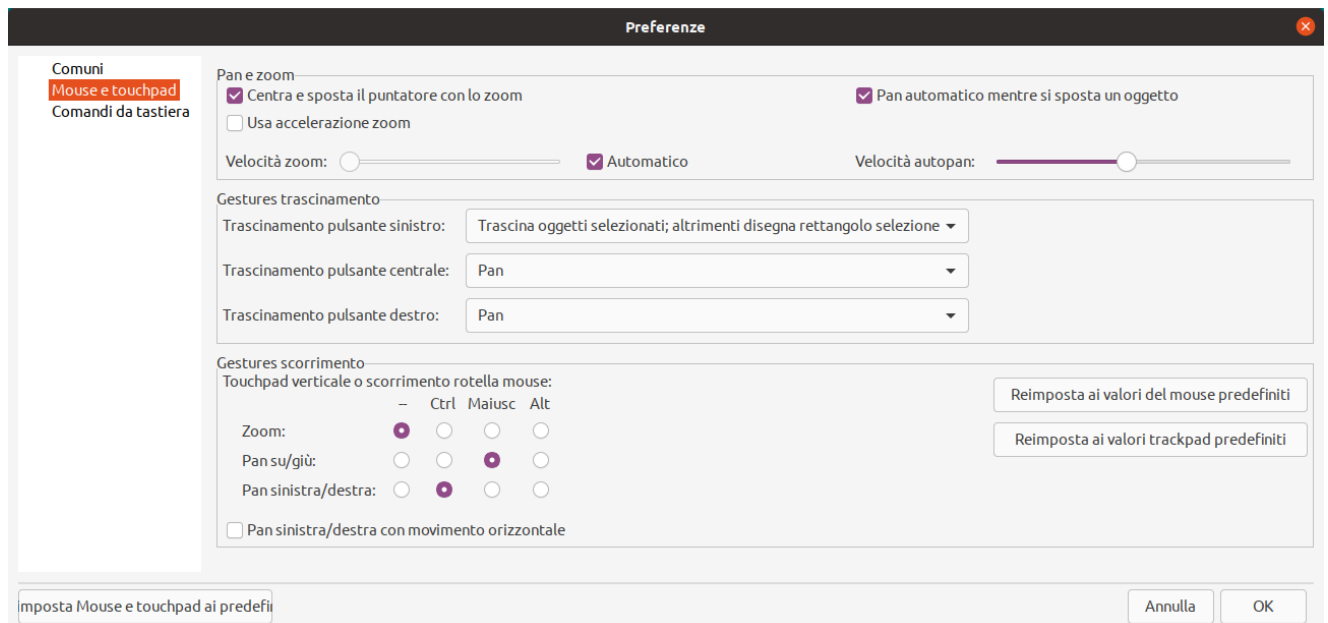
Numero max backup da tenere: quando si crea un nuovo backup, il file di backup più vecchio verrà eliminato per mantenere il numero totale di file di backup al di sotto di questo limite.

Numero max backup per giorno: quando si crea un nuovo backup, il file di backup più vecchio creato nel giorno corrente verrà eliminato per rimanere al di sotto di questo limite.

Tempo minimo tra i backup: Se il backup viene attivato (ad esempio, salvando un file scheda), il backup non verrà creato se un file di backup esistente è più recente di questo limite.

Dimensione massima totale backup: quando si crea un nuovo backup, i file di backup meno recenti verranno eliminati per mantenere la dimensione totale della cartella dei file di backup al di sotto di questo limite.

4.2. Preferenze di mouse e touchpad



Centra e sposta il puntatore con lo zoom: se abilitata, facendo zoom usando i comandi da tastiera o la rotellina del mouse, la vista verrà centrata sulla posizione del puntatore.

Usa accelerazione zoom: se abilitata, lo scorrimento più veloce della rotellina del mouse o del touchpad farà cambiare lo zoom più velocemente.

Velocità zoom: controlla quanto cambia lo zoom per una determinata quantità di scorrimento della rotellina del mouse o del touchpad. Utilizzare *Automatico* per impostare un valore predefinito a seconda del sistema operativo.

Pan automatico mentre si sposta un oggetto: se abilitata, la vista può essere spostata (pan) mentre si sta spostando un oggetto in prossimità del bordo dell'area di disegno.

Velocità autopan: controlla la velocità di spostamento dell'area di disegno durante lo spostamento di un oggetto.

Pulsanti del mouse: è possibile impostare il comportamento del trascinamento dei pulsanti centrale e destro del mouse per ingrandire la vista, eseguire una panoramica della vista o non avere alcun effetto. Si può anche impostare il comportamento del trascinamento del pulsante sinistro del mouse a seconda che nell'area di modifica siano già selezionati o meno oggetti.

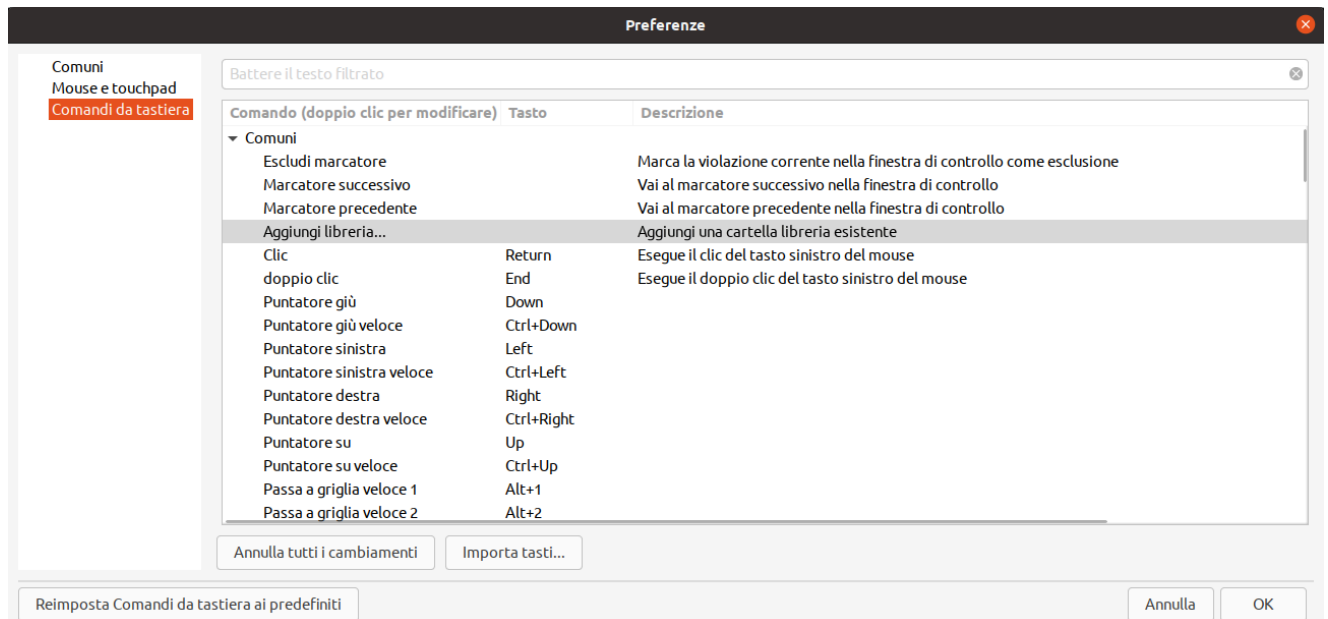
Nota

Il pulsante sinistro del mouse viene usato sempre per selezionare e gestire oggetti.

Rotellina del mouse e scorrimento touchpad: si può impostare il comportamento della rotellina del mouse o del movimento verticale del touchpad durante la pressione di certi tasti modificatori.

Pan sinistra/destra con movimento orizzontale: se abilitata, si può ottenere lo scorrimento orizzontale della vista usando il touchpad o la rotellina di scorrimento orizzontale (se presente nel mouse).

4.3. Preferenze comandi da tastiera



Si può usare questa finestra di dialogo per personalizzare i comandi da tastiera usati per controllare KiCad. I comandi da tastiera nella sezione *Comuni* sono condivisi tra ogni programma KiCad. I comandi da tastiera per ogni programma KiCad specifico vengono visualizzati quando quel programma è in esecuzione. Si può assegnare lo stesso tasto ad un'azione diversa in diversi programmi KiCad (ad esempio, l'editor di schemi e l'editor di schede), ma non si può assegnare lo stesso tasto a più di un'azione nello stesso programma.

Ci sono molti comandi disponibili e quindi non tutti hanno un tasto assegnato per impostazione predefinita. È possibile assegnare un tasto a un qualsiasi comando facendo doppio clic sul comando nell'elenco. Se si sceglie un tasto già assegnato a un comando diverso, si può scegliere di continuare a usare quel tasto sul comando scelto, rimuovendo l'assegnazione del tasto dal comando in conflitto.

Le modifiche apportate alle assegnazioni dei tasti vengono visualizzate con un carattere * alla fine del nome del comando. Si può annullare le modifiche a un comando specifico facendo clic con il pulsante destro del mouse su quel comando e selezionando **Annulla cambiamenti**, oppure si può annullare tutte le modifiche con il pulsante sotto l'elenco dei comandi.

Importazione comandi da tastiera

Le preferenze dei comandi da tastiera sono memorizzate in file `.hotkeys` nella cartella delle impostazioni di KiCad (consultare la sezione Impostazioni per informazioni su dove si trova la cartella delle impostazioni sul proprio sistema operativo). Se si ha personalizzato i comandi da tastiera di KiCad come si preferisce su un computer, si può trasferire questa configurazione su un altro computer importando i file `.hotkeys` appropriati.

4.4. Configurazione percorsi

In KiCad, one can define paths using a **path variable**. A few path variables are internally defined by KiCad, and can be used to define paths for libraries, 3D shapes, etc.

Ciò è utile quando i percorsi assoluti non sono conosciuti o sono soggetti a cambiamenti (per es. quando si trasferisce un progetto su un altro computer), o anche quando un percorso base viene condiviso con altri simili. Si consideri per esempio i seguenti che possono essere installati in posizioni variabili:

- Librerie di simboli di schema elettrico
- Librerie di impronte
- File modelli 3D usati nelle definizioni delle impronte

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`.

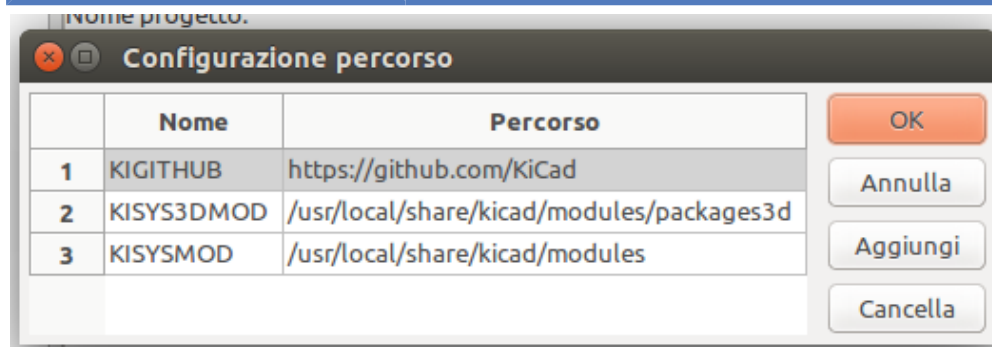
The **Preferences** → **Configure Paths...** menu allows you to define paths for some built-in KiCad path variables, and add your own path variables to define personal paths, if needed.

Nota

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.

4.4.1. KiCad path variables

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.



I percorsi impostati nella finestra di dialogo Configura percorsi sono interni a KiCad e non sono visibili come variabili ambiente al di fuori di KiCad. Sono memorizzati nei file di configurazione utente di KiCad.

Paths can also be set as system environment variables outside of KiCad, which will override any settings in the user's configuration.

Nota

You cannot override a system environment variable that has been set outside of KiCad by using the Configure Paths dialog. Any variable that has been set externally will be shown as read-only in the dialog.

Note also that the path variable `KIPRJMOD` is **always** internally defined by KiCad, and expands to the **current project absolute path**. For instance, `${KIPRJMOD}/connect.pretty` is always the `connect.pretty` folder (the footprint library) inside **the current project folder**. The `KIPRJMOD` variable cannot be changed in the Configure Paths dialog or overridden by an external environment variable.

4.4.2. Variabili ambiente avanzate

Some advanced environment variables can be set to customize where KiCad expects certain files to be located. By default, these locations are set based on your platform, but they can be overridden by system environment variables. These variables are not shown in the Configure Paths dialog and cannot be used in path substitutions.

La modifica di queste variabili non comporterà lo spostamento di alcun file da parte di KiCad dalla posizione predefinita alla nuova posizione, quindi se si modificano queste variabili sarà necessario copiare manualmente le impostazioni o i file desiderati.

<code>KICAD_CONFIG_HOME</code>	Base path of KiCad configuration files. Subdirectories will be created within this directory for each KiCad minor version.
<code>KICAD_DOCUMENTS_HOME</code>	Base path of KiCad user-modifiable documents, such as projects, templates, Python scripts, libraries, etc. Subdirectories will be created within this directory for each KiCad minor version. This directory is provided as a suggested user data location, but does not need to be used.
<code>KICAD_STOCK_DATA_HOME</code>	Base path of KiCad stock data, including default libraries. The data in this directory is managed by the KiCad installer or system package manager, and is not intended for user-writable data.

Avvertimento

Se si modifica la configurazione dei percorsi, uscire e riavviare KiCad, per evitare qualsiasi problema nella gestione dei percorsi.

4.5. Configurazione librerie

The **Preferences** → **Manage Symbol Libraries...** menu lets you manage the list of symbol libraries (symbol library table).

Likewise, use the **Preferences** → **Manage Footprint Libraries...** menu to manage the list of footprint libraries (footprint library table).

Per ogni tipo di libreria (simboli e impronte), ci sono 2 tabelle di librerie: globali e specifiche del progetto. La tabella delle librerie globali si trova nella cartella di configurazione dell'utente e contiene l'elenco delle librerie disponibili per tutti i progetti. La tabella delle librerie specifiche del progetto è facoltativa, contiene appunto l'elenco delle librerie specifiche del progetto, e si trova nella cartella del progetto.

Capitolo 5. Jobsets

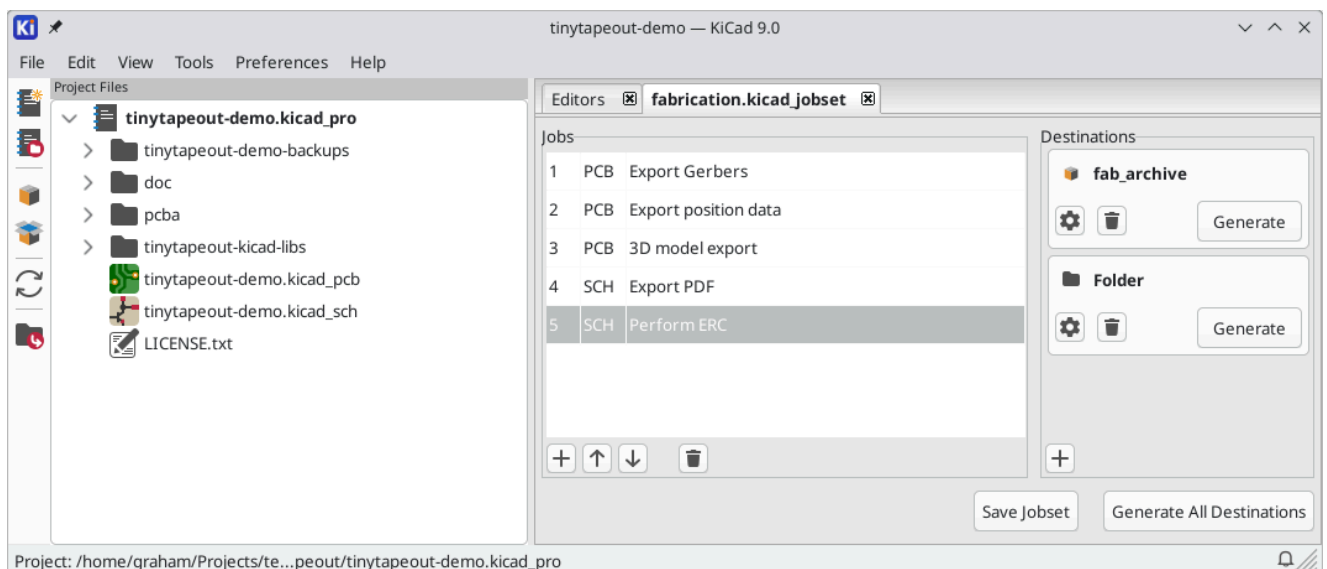
KiCad lets you configure a list of outputs that are all generated with a single click. The list of output jobs and the destinations where they will be saved is called a *jobset*. For example, a jobset might contain jobs to generate Gerber files, assembly data, a bill of materials, PDF plots of the schematic 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.

5.1. 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.

Nota

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.

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.

5.2. 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.

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.

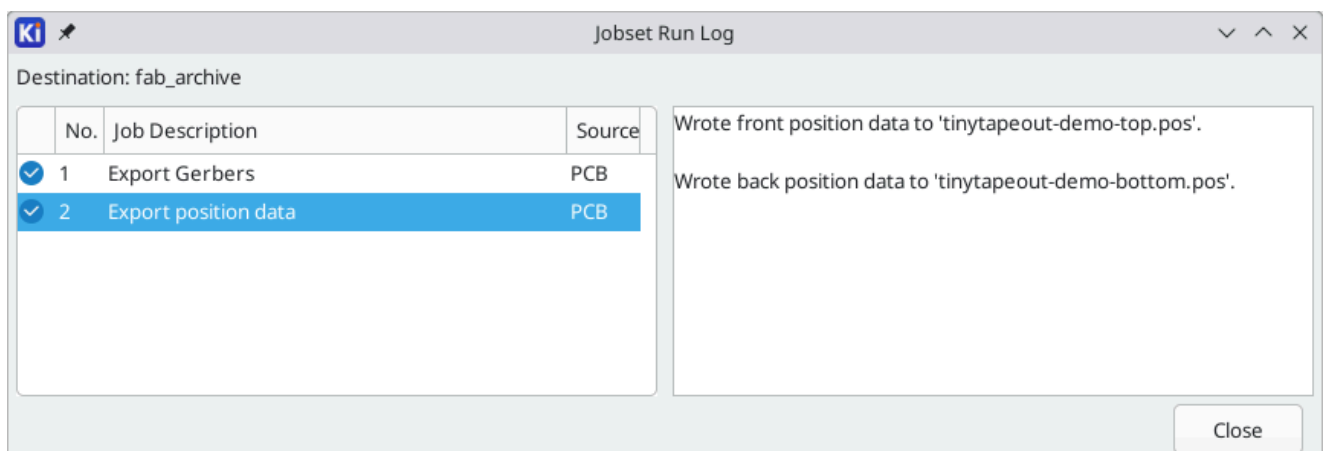
When you click **OK** in this dialog, the new jobset destination is added to the Destinations list. You can modify an existing jobset destination by clicking its



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.



5.3. 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 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	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.

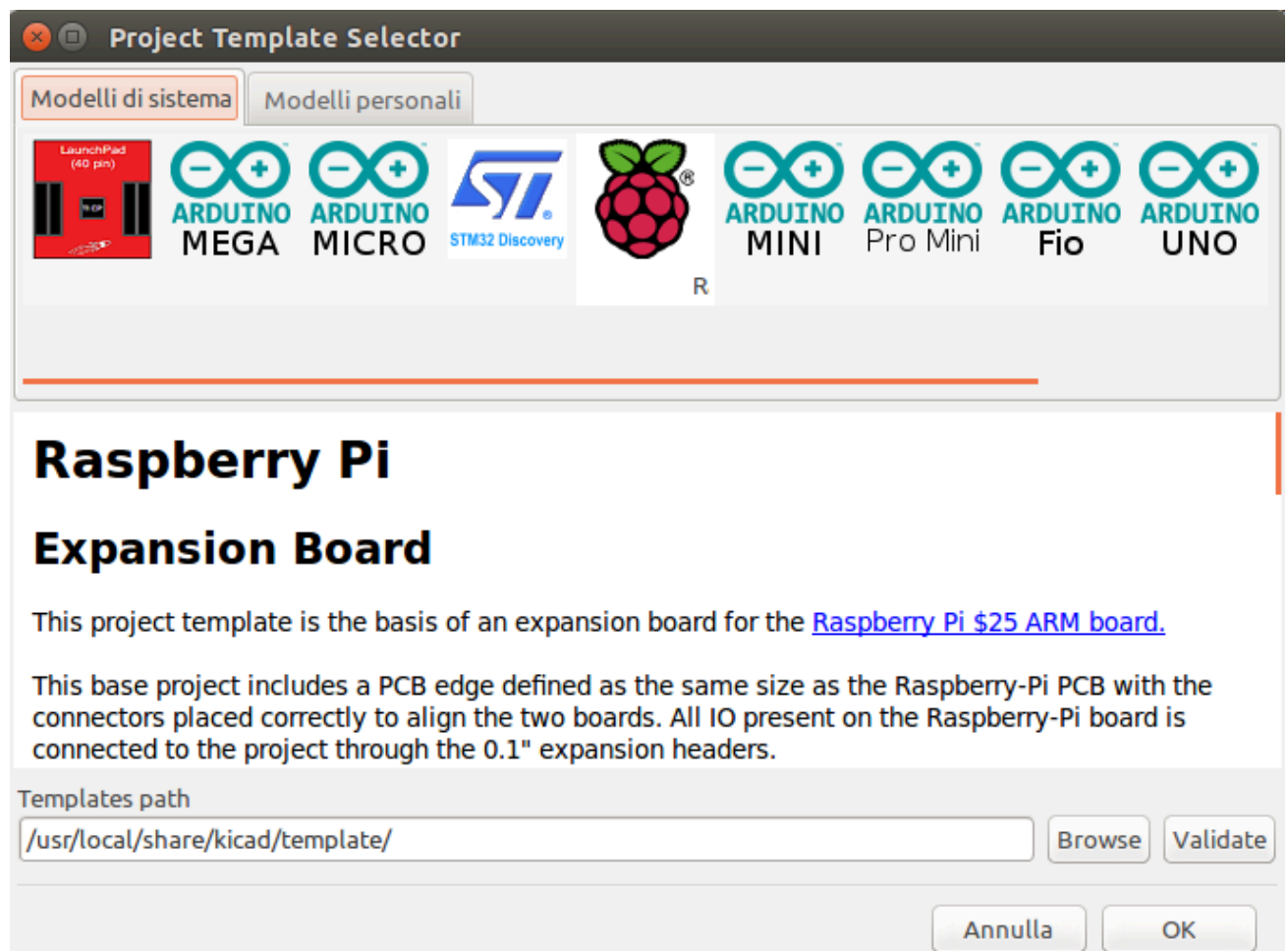
Capitolo 6. Modelli di progetto

L'uso di un modello di progetto facilita la creazione di un nuovo progetto, includendo alcune preimpostazioni. I modelli possono contenere: profili di scheda predefiniti, posizioni di connettori, elementi dello schema, regole di progettazione, ecc. Persino schemi elettrici o circuiti stampati completi possono venire inclusi come spunti per il nuovo progetto.

6.1. Uso dei modelli

Il menu **File # Nuovo progetto da modello** aprirà la finestra di dialogo di selezione dei modelli:

Un singolo clic sull'icona di un modello mostrerà le informazioni del modello stesso, mentre un ulteriore clic sul pulsante OK farà partire la creazione del nuovo progetto. I file del modello verranno copiati nella posizione del nuovo progetto e rinominati per riflettere il nome del nuovo progetto.



6.2. Posizioni modelli

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.

6.3. Creazione dei modelli

A KiCad template is simply a directory containing the template project files, as well as some required metadata for the template in a subdirectory named `meta`. The name of the directory containing the template files determines the name of the template. When you create a project from a template, KiCad copies the template files to the new project directory, renaming them to match the new project name as described below.

All files in the template are copied, with two exceptions:

- Files with names beginning with the `.` character (dotfiles) are not copied. There is a special case for files named `.gitignore` or `.gitattributes`, which are copied if they exist.
- The `meta` directory is not copied

The `meta` directory must contain an HTML file named `info.html`, which is displayed in the KiCad template browser and should contain basic information describing the template. Basic HTML features are supported, including images. Any images referenced by `info.html` should also be stored in the `meta` directory.

The `<title>` tag determines the name of the template that is displayed during template selection. Note that the project template name will be cut off if it's too long. This display name does not need to be the same as the template directory name.

Ecco un file `info.html` di esempio:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=windows-1252">
<TITLE>Raspberry Pi - Expansion Board</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>This project template is the basis of an expansion board for the
<A HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25
ARM board.</A> <BR><BR>This base project includes a PCB edge defined
as the same size as the Raspberry-Pi PCB with the connectors placed
correctly to align the two boards. All IO present on the Raspberry-Pi
board is connected to the project through the 0.1" expansion
headers. <BR><BR>The board outline looks like the following:
</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>
```

Finally, `meta` can optionally contain an image named `icon.png`, which will be used as the template's icon in the template selection dialog. The icon should be a 64 x 64 pixel PNG image.

6.3.1. Template file renaming

All files and directories in a template are copied to the new project path when a project is created using a template, except `meta` and any dotfiles. Files and directories containing the template directory name will be renamed with the new project file name.

For example, using a template named `example` (left) to create a project named `newproject` (right), with renamed files shown in **bold**:

File presenti nella cartella modello <code>esempio</code>	File creati nella cartella progetto <code>nuovoprogetto</code>
<code>esempio.kicad_pro</code>	<code>*nuovoprogetto.kicad_pro*</code>
<code>esempio.kicad_sch</code>	<code>*nuovoprogetto.kicad_sch*</code>
<code>esempio.kicad_pcb</code>	<code>*nuovoprogetto.kicad_pcb*</code>
<code>esempio-primo.kicad_sch</code>	<code>*nuovoprogetto-primo.kicad_sch*</code>
<code>secondo-esempio.kicad_sch</code>	<code>*secondo-nuovoprogetto.kicad_sch*</code>
<code>terzo.kicad_sch</code>	<code>terzo.kicad_sch</code>
<code>terzo.kicad_pcb</code>	<code>terzo.kicad_pcb</code>

A template does not need to contain a complete project. If a required project file is missing, KiCad will create the file using the default create project behavior:

File nella cartella modelli <code>esempio</code>	File creati nella cartella <code>nuovoprogetto</code>
<code>esempio.kicad_sch</code>	<code>*nuovoprogetto.kicad_sch*</code>
<code>primo-esempio.kicad_sch</code>	<code>*primo-nuovoprogetto.kicad_sch*</code>
<code>primo-esempio.kicad_pcb</code>	<code>*primo-nuovoprogetto.kicad_pcb*</code>
<code>secondo-esempio.kicad_sch</code>	<code>*secondo-nuovoprogetto.kicad_sch*</code>
<code>secondo-esempio.kicad_pcb</code>	<code>*secondo-nuovoprogetto.kicad_pcb*</code>
	<code>*nuovoprogetto.kicad_pro* (predefinito)</code>
	<code>*nuovoprogetto.kicad_pcb* (predefinito)</code>

As an exception to the template name renaming rule, if the template contains one project file (`.kicad_pro`) and its name doesn't match the template name, KiCad will do the renaming based on that project file name instead:

File nella cartella <code>esempio</code>	File creati nella cartella <code>nuovoprogetto</code>
<code>esempio.kicad_sch</code>	<code>esempio.kicad_sch</code>
<code>esempio.kicad_pcb</code>	<code>esempio.kicad_pcb</code>
<code>*primo-esempio.kicad_pro*</code>	<code>*nuovoprogetto.kicad_pro*</code>
<code>primo-esempio.kicad_sch</code>	<code>*nuovoprogetto.kicad_sch*</code>
<code>primo-esempio.kicad_pcb</code>	<code>*nuovoprogetto.kicad_pcb*</code>
<code>secondo-esempio.kicad_sch</code>	<code>secondo-esempio.kicad_sch</code>
<code>secondo-esempio.kicad_pcb</code>	<code>secondo-esempio.kicad_pcb</code>

Nota

Non è consigliabile creare un modello con più di un file progetto.

Capitolo 7. Plugin e gestore contenuti

Nota

This section of the KiCad documentation has not yet been written. We appreciate your patience as our small team of volunteer documentation writers work to update and expand the documentation.

Capitolo 8. Elenco di riferimento azioni

Di seguito è riportato un elenco di tutte le **azioni** disponibili nella Gestione progetti di KiCad: un comando che può essere assegnato a un tasto comando.

8.1. Gestore progetti di KiCad

Le azioni seguenti sono disponibili nella Gestione progetti di KiCad. I tasti comando possono essere assegnati a una qualsiasi di queste azioni nella sezione **Tasti comando** delle preferenze.

Action	Default Hotkey	Description
Close Project		Close the current project
Image Converter	kbd:[Ctrl+B]	Convert bitmap images to schematic or PCB components
Drawing Sheet Editor	kbd:[Ctrl+Y]	Edit drawing sheet borders and title block
Footprint Editor	kbd:[Ctrl+F]	Edit PCB footprints
PCB Editor	kbd:[Ctrl+P]	Edit PCB
Schematic Editor	kbd:[Ctrl+E]	Edit schematic
Symbol Editor	kbd:[Ctrl+L]	Edit schematic symbols
Clone Project from Repository...		Clone a project from an existing repository
New Project from Template...	kbd:[Ctrl+T]	Create new project from template
New Project...	kbd:[Ctrl+N]	Create new blank project
Open Demo Project...		Open a demo project
Open Project...	kbd:[Ctrl+O]	Open an existing project
Open Text Editor		Launch preferred text editor
Plugin and Content Manager	kbd:[Ctrl+M]	Run Plugin and Content Manager
Calculator Tools		Run component calculations, track width calculations, etc.
Gerber Viewer	kbd:[Ctrl+G]	Preview Gerber output files