



kiCad



KiCad

KiCad

6 grudnia 2018

Spis treści

1	Wstęp	1
1.1	KiCad	1
1.2	Pliki i foldery programu KiCad	2
2	Instalacja oraz konfiguracja	4
2.1	Opcje wyświetlania	4
2.2	Inicjalizacja domyślnej konfiguracji	4
2.3	Modyfikacja domyślnej konfiguracji	4
2.4	Paths configuration	5
2.5	Inicjalizacja zewnętrznych narzędzi	6
2.5.1	Wybór edytora tekstu	6
2.5.2	Wybór przeglądarki PDF	6
2.6	Creating a new project	6
2.7	Importing a foreign project	7
3	Using KiCad project manager	8
3.1	Project manager window	9
3.2	Panel uruchomieniowy	9
3.3	Drzewo projektu	10
3.4	Górny pasek narzędzi	10
4	Szablony użytkownika	11
4.1	Using templates	11
4.2	Miejsce przechowywania szablonów:	13
4.3	Tworzenie szablonów	14
4.3.1	Wymagane pliki:	15
4.3.2	Pliki opcjonalne:	16

I	Upgrading from Version 4 to Version 5	17
5	Schematic Symbol Libraries	19
5.1	Global Symbol Library Table.	19
5.2	Symbol Library Table Mapping	19
5.3	Remapping Search Order	20
5.4	Symbol Names and Symbol Library Nickname Limitations	21
6	Symbol Cache Library Availability	22
7	Board File Format Changes	23
7.1	Global Footprint Library Table.	23

Podręcznik użytkownika

Prawa autorskie

This document is Copyright © 2010-2018 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.

Wszystkie znaki towarowe użyte w tym dokumencie należą do ich właścicieli.

Współtwórcy

Jean-Pierre Charras, Fabrizio Tappero.

Tłumaczenie

Kerusey Karyu <keruseykaryu@o2.pl>, 2014-2015.

Kontakt

Wszelkie zauważone błędy, sugestie lub nowe wersje dotyczące tego dokumentu prosimy kierować do:

- W sprawie dokumentacji: <https://github.com/KiCad/kicad-doc/issues>
- W sprawie oprogramowania: <https://bugs.launchpad.net/kicad>
- About KiCad translation: <https://github.com/KiCad/kicad-i18n/issues>

Data publikacji i wersja oprogramowania

21 maja 2015.

Rozdział 1

Wstęp

1.1 KiCad

KiCad to pakiet programów Open Source do rysowania schematów i tworzenia obwodów drukowanych (PCB). Pod jego osobliwą i prostą powierzchnią, KiCad zawiera elegancką spójność następujących samodzielnych narzędzi:

- **KiCad** : Menadżer projektu
- **Eeschema** : Edytor schematów oraz edytor symboli
- **Pcbnew** : Edytor obwodów drukowanych oraz edytor footprintów
- **GerbView** : Przeglądarka plików Gerber

Dołączono też 3 narzędzia:

- **Bitmap2Component**: Do tworzenia graficznych logotypów. Tworzy komponenty lub footprinty z map bitowych.
- **PcbCalculator**: Podręczny kalkulator pomocny w obliczaniu elementów stabilizatorów, szerokości ścieżek na podstawie prądu obciążenia, rozmiarów ścieżek linii transmisyjnych, itp.
- **Pl_Editor**: Edytor do tworzenia spersonalizowanych obramowań arkuszy projektowych.

Narzędzia te są zwykle uruchamiane za pomocą menadżera projektu, ale mogą też być uruchomione jako samodzielne aplikacje.

KiCad nie ma żadnych znaczących ograniczeń co do rozmiaru obwodów drukowanych i może z powodzeniem obsłużyć do 32 warstw sygnałowych, 14 warstw technicznych oraz 4 warstw pomocniczych.

KiCad umożliwia utworzenie wszystkich plików wymaganych do wyprodukowania obwodu drukowanego, czyli:

- pliki Gerber dla fotoploterów
 - pliki wierceń
 - pliki dla maszyn Pick&Place
-

Będąc oprogramowaniem Open Source (licencja GPL), KiCad reprezentuje sobą idealne narzędzie dla projektów zorientowanych na tworzenie elektroniki z gałęzi Open Hardware.

KiCad is available for Linux, Windows and Apple macOS.

1.2 Pliki i foldery programu KiCad

KiCad tworzy oraz używa plików ze specyficznymi rozszerzeniami (oraz folderami) dla poszczególnych plików schematów oraz obwodów drukowanych.

Plik menadżera projektu:

*.pro	Mały plik zawierający parametry dla bieżącego projektu oraz listę bibliotek schematu
-------	--

Pliki edytora schematów:

*.sch	Schematic files, which do not contain the components themselves.
*.lib	Schematic component library files, containing the component descriptions: graphic shape, pins, fields.
*.dcm	Schematic component library documentation, containing some component descriptions: comments, keywords, reference to data sheets.
*_cache.lib	Schematic component library cache file, containing a copy of the components used in the schematic project.
sym-lib-table	Symbol library list (<i>symbol library table</i>): list of symbol libraries available in the schematic editor.

Pliki i foldery edytora obwodów drukowanych:

*.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.
*.brd	Board file in the legacy format. Can be read, but not written, by the current board editor.
*.mod	Footprint library in the legacy format. Can be read by the footprint or the board editor, but not written.
fp-lib-table	Footprint library list (<i>footprint library table</i>): list of footprint libraries (various formats) which are loaded by the board or the footprint editor or CvPcb.

Pliki wspólne:

*.kicad_wks	Plik z definicją układu strony, dla osób wymagających własnego układu obramowań arkusza.
*.net	Plik listy sieci tworzony na podstawie schematu, i odczytywany przez edytor PCB. Ten plik jest łączony z plikiem .cmp, dla użytkowników, którzy wolą posiadać osobne pliki z przypisaniami komponent-footprint.

Plik specjalny:

*.cmp	Association between components used in the schematic and their footprints. It can be created by Pcbnew and imported by Eeschema. Its purpose is to import changes from Pcbnew to Eeschema, for users who change footprints inside Pcbnew (for instance using <i>Exchange Footprints</i> command) and want to import these changes in schematic.
-------	---

Inne pliki:

Pliki te są generowane przez program KiCad do celów produkcji obwodów drukowanych.

*.gbr	Pliki Gerber, do produkcji płytki.
*.drl	Pliki wierceń (format Excellon), do produkcji płytki.
*.pos	Pliki położeń (format ASCII), dla maszyn układających elementy.
*.rpt	Pliki raportów (format ASCII), do celów dokumentacji.
*.ps	Pliki z rysunkami (Postscript), do celów dokumentacji.
*.pdf	Pliki z rysunkami (format PDF), do celów dokumentacji.
*.svg	Pliki z rysunkami (format SVG), do celów dokumentacji.
*.dxf	Pliki z rysunkami (format DXF), do celów dokumentacji.
*.plt	Pliki z rysunkami (format HPGL), do celów dokumentacji.

Rozdział 2

Instalacja oraz konfiguracja

2.1 Opcje wyświetlania

Hardware accelerated renderer in Pcbnew and Gerbview requires video card with support of OpenGL v2.1 or higher.

2.2 Inicjalizacja domyślnej konfiguracji

The default configuration file named **kicad.pro** is supplied in `kicad/template`. It serves as a template for any new project and is used to set the list of library files loaded by Eeschema. A few other parameters for Pcbnew (default text size, default line thickness, etc.) are also stored here.

Jeśli istnieje drugi z plików konfiguracyjnych *fp-lib-table*, będzie on jednokrotnie użyty by utworzyć listę dostępnych footprintów; inaczej, lista ta będzie utworzona od nowa.

2.3 Modyfikacja domyślnej konfiguracji

Domyślny plik **kicad.pro** może zostać dowolnie zmieniony, jeśli zajdzie taka potrzeba.

Sprawdź czy posiadasz prawa zapisu do `kicad/template/kicad.pro`

Uruchom program KiCad oraz otwórz projekt *kicad.pro*.

Run Eeschema via KiCad manager. Modify and update the Eeschema configuration, to set the list of libraries you want to use each time you create new projects.

Run Pcbnew via KiCad manager. Modify and update the Pcbnew configuration, especially the footprint library list. Pcbnew will create or update a library list file called **footprint library table**. There are 2 library list files (named *fp-lib-table*): The first (located in the user home directory) is global for all projects and the second (located in the project directory) is optional and specific to the project.

2.4 Paths configuration

W programie KiCad, należy zdefiniować parę ścieżek dostępu używając do tego *zmiennych systemowych*. Kilka zmiennych jest definiowanych przez program KiCad, i może być użyta do przekazania do programu odpowiednich ścieżek, takich jak ścieżki do bibliotek, ścieżki do definicji obiektów 3D, itp ...

This is useful when absolute paths are not known or are subject to change (e.g. when you transfer a project to a different computer), and also when one base path is shared by many similar items. Consider the following which may be installed in varying locations:

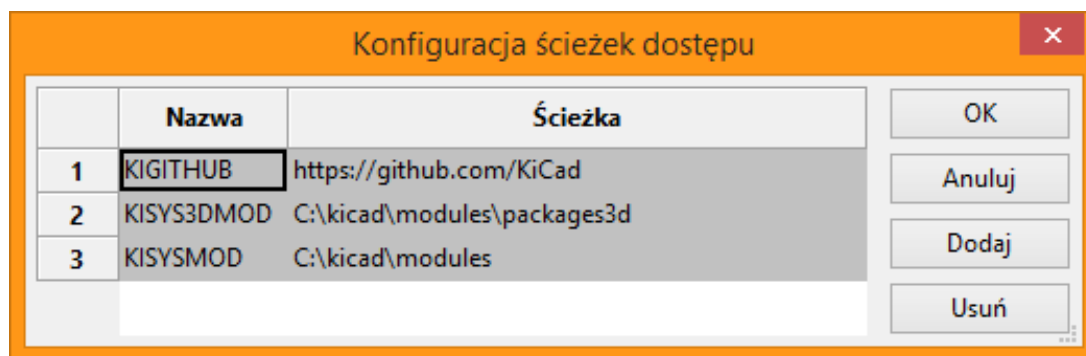
- Biblioteki programu Eeschema
- Biblioteki footprintów programu Pcbnew
- Pliki z kształtami 3D używane w definicjach footprintów

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

This option allows you to define a path using an environment variable, and add your own environment variables to define personal paths, if needed.

Zmienne systemowe programu KICAD:

KICAD_PTEMPLATE	Templates used during project creation (DEPRECATED as of version 5.0.0-rc2, use KICAD_TEMPLATE_DIR instead). If you are using this variable, it must be defined.
KICAD_SYMBOL_DIR	Base path of symbol library files.
KIGITHUB	Frequently used in example footprint lib tables. If you are using this variable, it must be defined.
KISYS3DMOD	Base path of 3D shapes files, and must be defined because an absolute path is not usually used.
KISYSMOD	Base path of footprint library folders, and must be defined if an absolute path is not used in footprint library names.
KICAD_TEMPLATE_DIR	Location of templates installed with KiCad.
KICAD_USER_TEMPLATE_DIR	Location of personal templates.



Należy pamiętać, że zmienna **KIPRJMOD** jest **zawsze** wewnętrznie definiowana przez program KiCad, i wskazuje na **absolutną ścieżkę dostępu do bieżącego projektu**.

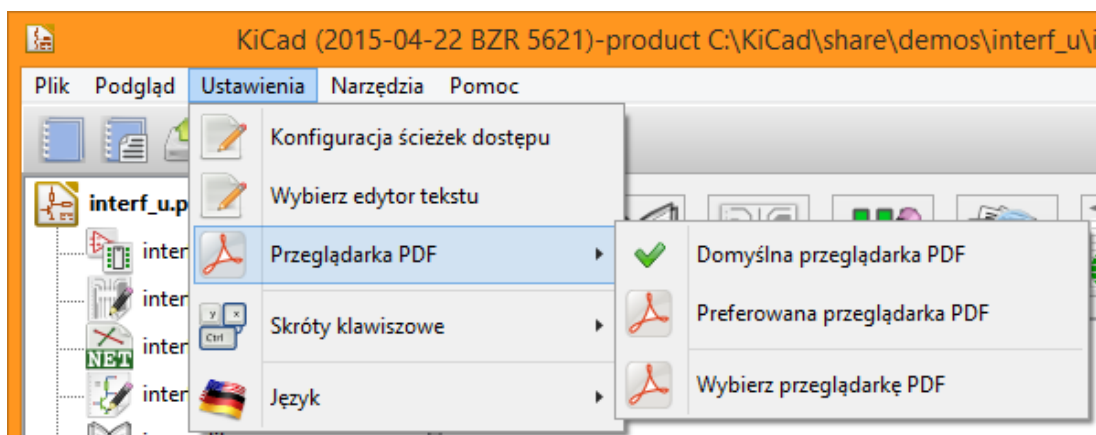
Przykładowo, `/${KIPRJMOD}/Connect.pretty` jest zawsze odnośnikiem do folderu *Connect.pretty* (Biblioteka footprintów typu Pretty) *wewnątrz folderu bieżącego projektu*.

Jeśli zmodyfikujemy konfigurację ścieżek, należy zawsze zamknąć program KiCad, oraz ponownie go uruchomić, by zapobiec błędom związanym z użyciem niepoprawnych ścieżek dostępu.

2.5 Inicjalizacja zewnętrznych narzędzi.

You may define your favorite text editor and PDF viewer. These settings are used whenever you want to open a text or PDF file.

Opcje te są dostępne przez menu Preferencje:



2.5.1 Wybór edytora tekstu

Przed użyciem edytora tekstu do przeglądania lub modyfikacji plików bieżącego projektu, należy wybrać odpowiedni edytor, który będzie używany do tego celu.

Wybierz polecenie *Ustawienia* → *Wybierz edytor tekstu* by wskazać plik wykonywalny edytora jaki chcesz używać.

2.5.2 Wybór przeglądarki PDF

Można użyć domyślnej przeglądarki plików PDF lub innej wybranej przeglądarki.

Aby zmienić domyślną przeglądarkę PDF użyj polecenia *Ustawienia* → *Przeglądarka PDF* → *Wybierz przeglądarkę PDF* by wskazać plik wykonywalny przeglądarki PDF, następnie wybierz *Ustawienia* → *Przeglądarka PDF* → *Preferowana przeglądarka PDF*.

W systemach Linux znany jest problem z domyślną przeglądarką PDF, dlatego użycie opcji *Preferowana przeglądarka PDF* po wyborze właściwej przeglądarki jest obowiązkowe.

2.6 Creating a new project

In order to manage a KiCad project consisting of schematic files, printed circuit board files, supplementary libraries, manufacturing files for photo-tracing, drilling and automatic component placement files, it is recommended to create a project as follows:

- **Utworzyć katalog roboczy dla projektu** (używając narzędzi dostępnych z poziomu centrum programu lub narzędzi dostępnych z poziomu systemu operacyjnego).
- **W katalogu tym przy użyciu KiCad-a utworzyć plik projektu** (generowany jest plik `.pro`) za pomocą ikony *Utwórz nowy projekt* lub *Utwórz nowy za pomocą szablonu*.

**Ostrzeżenie**

Należy używać unikalnych folderów dla każdego projektu programu KiCad. Nie należy łączyć wielu projektów w jednym folderze.

KiCad tworzy plik z rozszerzeniem `.pro` który zawiera listę parametrów, zależnych od danego projektu (np. listę użytych bibliotek w schematach, oraz ustawienia programu). Nazwa domyślna pliku ze schematem i PCB jest oparta na nazwie projektu. Dlatego, jeśli projekt nazwany `example.pro` został stworzony w folderze nazwanym `example`, domyślnie będą znajdować się tam pliki:

<code>example.pro</code>	Plik projektu.
<code>example.sch</code>	Plik schematu.
<code>example.kicad_pcb</code>	Plik z obwodem drukowanym.
<code>example.net</code>	Lista sieci.
<code>example.*</code>	Inne pliki tworzone przez narzędzia, w tym pliki tymczasowe i kopie zapasowe.
<code>example-cache.lib</code>	Podręczna pamięć z biblioteką elementów użytych na schemacie zawierający kopie użytych komponentów ze schematu.

2.7 Importing a foreign project

KiCad is able to import files created using other software packages. Currently only Eagle 6.x or newer (XML format) is supported.

To import a foreign project, you need to select either a schematic or a board file in the import file browser dialog. Imported schematic and board files should have the same base file name (e.g. `project.sch` and `project.brd`). Once the requested files are selected, you will be asked to select a directory to store the imported files, which are going to be saved as a KiCad project.

Rozdział 3

Using KiCad project manager

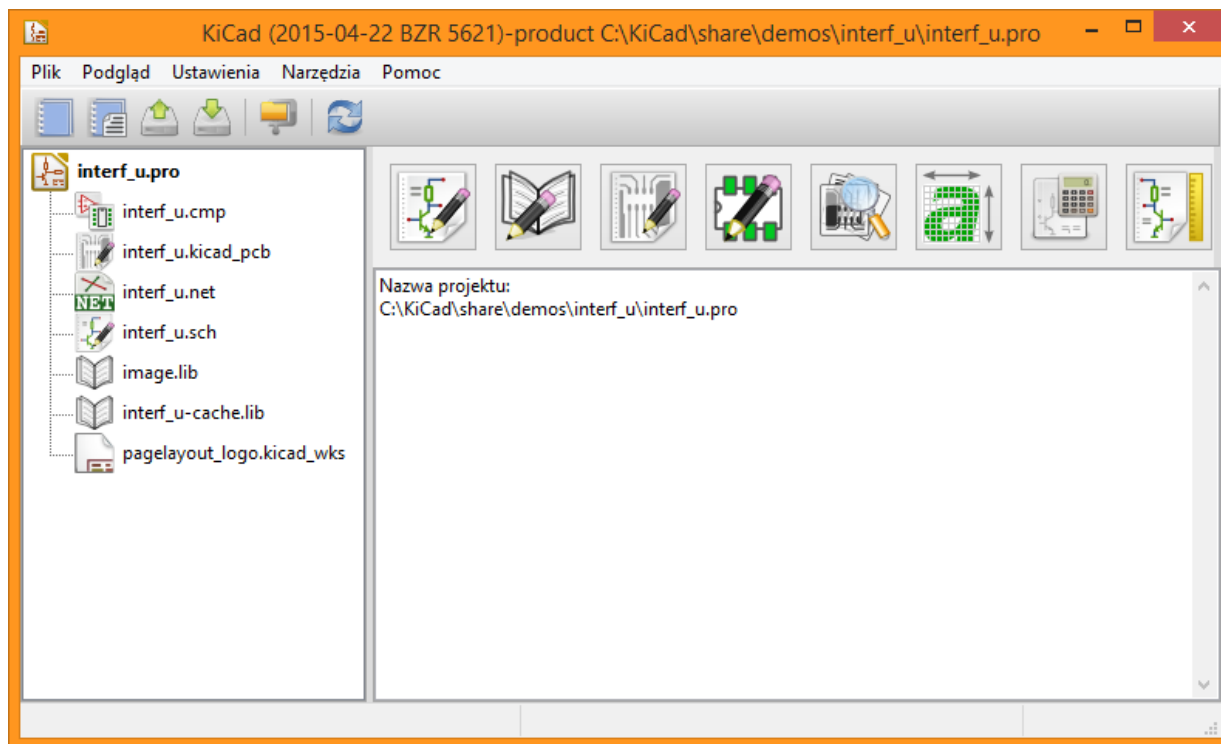
KiCad project manager (kicad or kicad.exe) is a tool which can easily run the other tools (schematic and PCB editors, Gerber viewer and utility tools) when creating a design.

Uruchamianie pozostałych aplikacji z poziomu Menadżera Projektu ma swoje zalety:

- “cross probing” pomiędzy edytorem schematów a edytorem obwodów drukowanych.
- “cross probing” pomiędzy edytorem schematów a narzędziem do przypisywania footprintów (CvPcb).

Jednakże, można modyfikować tylko bieżące pliki projektu. Gdy poszczególne aplikacje są uruchamiane jako *samodzielne aplikacje* można otworzyć dowolny plik z dowolnego projektu ale “cross probing” może dać niespodziewane rezultaty.

3.1 Project manager window



Okno główne składa się z listy o strukturze drzewa (po lewej) zawierającą pliki projektu, panelu uruchomieniowego (po prawej, na górze) pozwalającego na uruchomienie poszczególnych narzędzi oraz okna z wiadomościami. Główne menu oraz pasek narzędzi może być użyte do utworzenia, odczytania, zapisania pliku projektu (*.pro), a także do zarchiwizowania całości projektu do pliku archiwum ZIP.

3.2 Panel uruchomieniowy

KiCad allows you to run all standalone software tools that come with it.

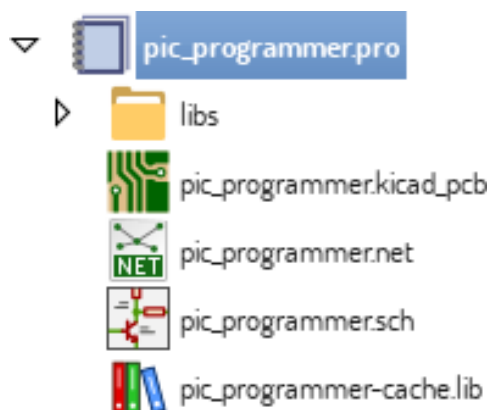
Panel uruchomieniowy posiada 8 przycisków, które odpowiadają poszczególnym narzędziom (1 do 8, od lewej):



1	EESchema	Zaawansowany, hierarchiczny edytor schematów.
2	LibEdit	Narzędzie do tworzenia bibliotek symboli oraz zarządzania nimi.
3	Pcbnew	Edytor obwodów drukowanych.
4	Footprint Editor	Narzędzie do tworzenia bibliotek footprintów oraz zarządzania nimi.
5	Gerbrview	Przeglądarka plików GERBER. Może również pokazywać pliki wierceń.
6	Bitmap2component	Narzędzie do tworzenia footprintów lub symboli schematów z czarno-białych obrazów bitmapowych mogących służyć jako logotyp.

7	Pcb Calculator	Narzędzie do obliczania szerokości ścieżek, oraz wielu innych parametrów obwodów drukowanych.
8	Pl Editor	Narzędzie do tworzenia i edycji obramowań arkuszy projektowych.

3.3 Drzewo projektu



Double-clicking on the schematic file runs the schematic editor, in this case opening the file **pic_programmer.sch**.







Double-clicking on the board file runs the layout editor, in this case opening the file **pic_programmer.kicad_pcb**.

Kliknięcie prawym klawiszem wywołuje podręczne menu, które pozwala na dodatkowe operacje na plikach.

3.4 Górny pasek narzędzi



Górny pasek narzędzi programu KiCad pozwala na wykonanie niektórych podstawowych operacji na plikach (od lewej):

	Create a new project. If the default template file (kicad.pro) is found in kicad/template , it is copied into the working directory.
	Create a new project from an existing template.
	Open an existing project.
	Update and save the current project tree.
	Create a zip archive of the whole project. This includes schematic files, libraries, PCB, etc.
	Refresh the tree view, sometimes needed after a tree change.

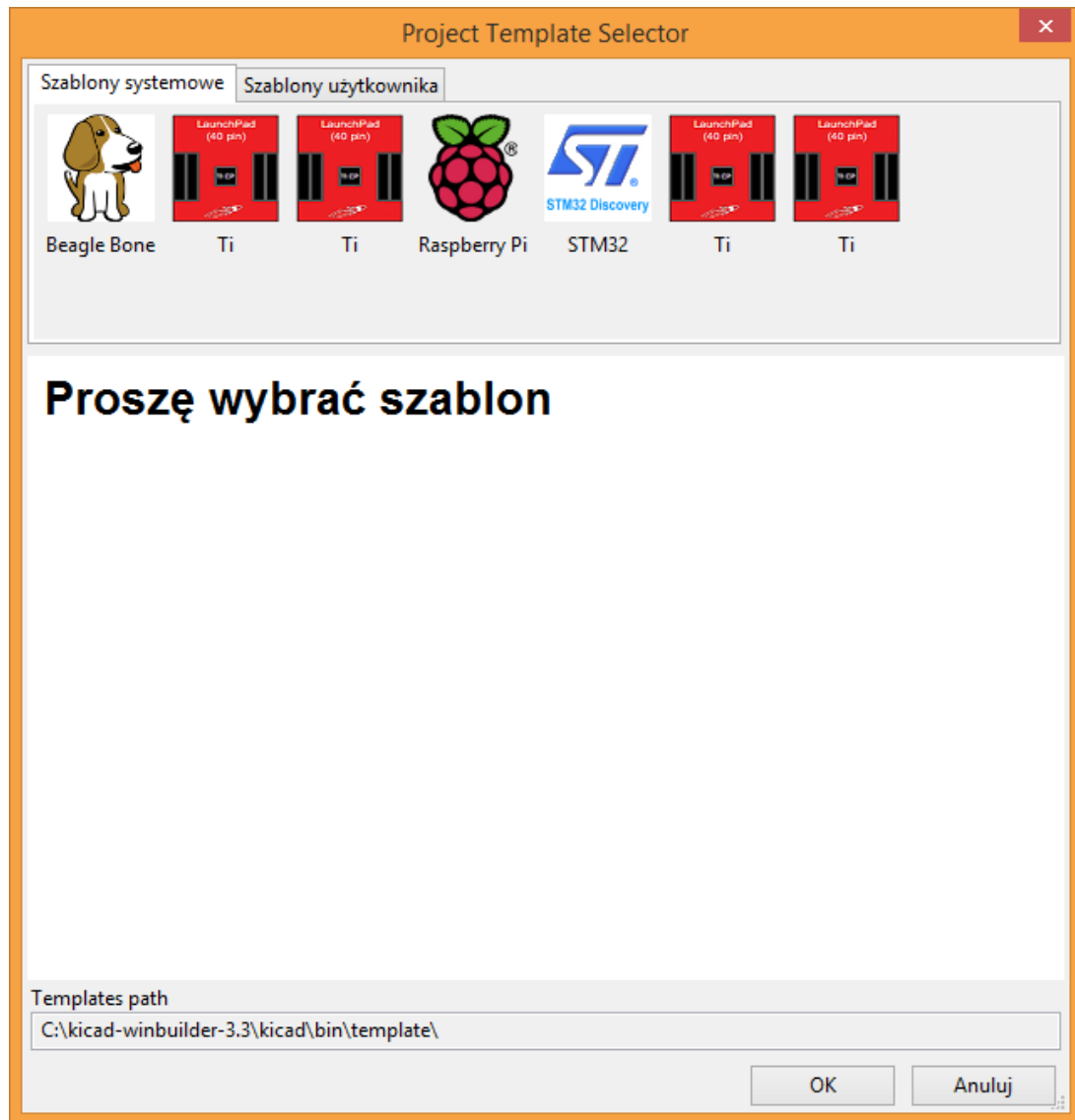
Rozdział 4

Szablony użytkownika

Using a project template facilitates setting up a new project with predefined settings. Templates may contain predefined board outlines, connector positions, schematic elements, design rules, etc. Complete schematics and/or PCBs used as seed files for the new project may even be included.

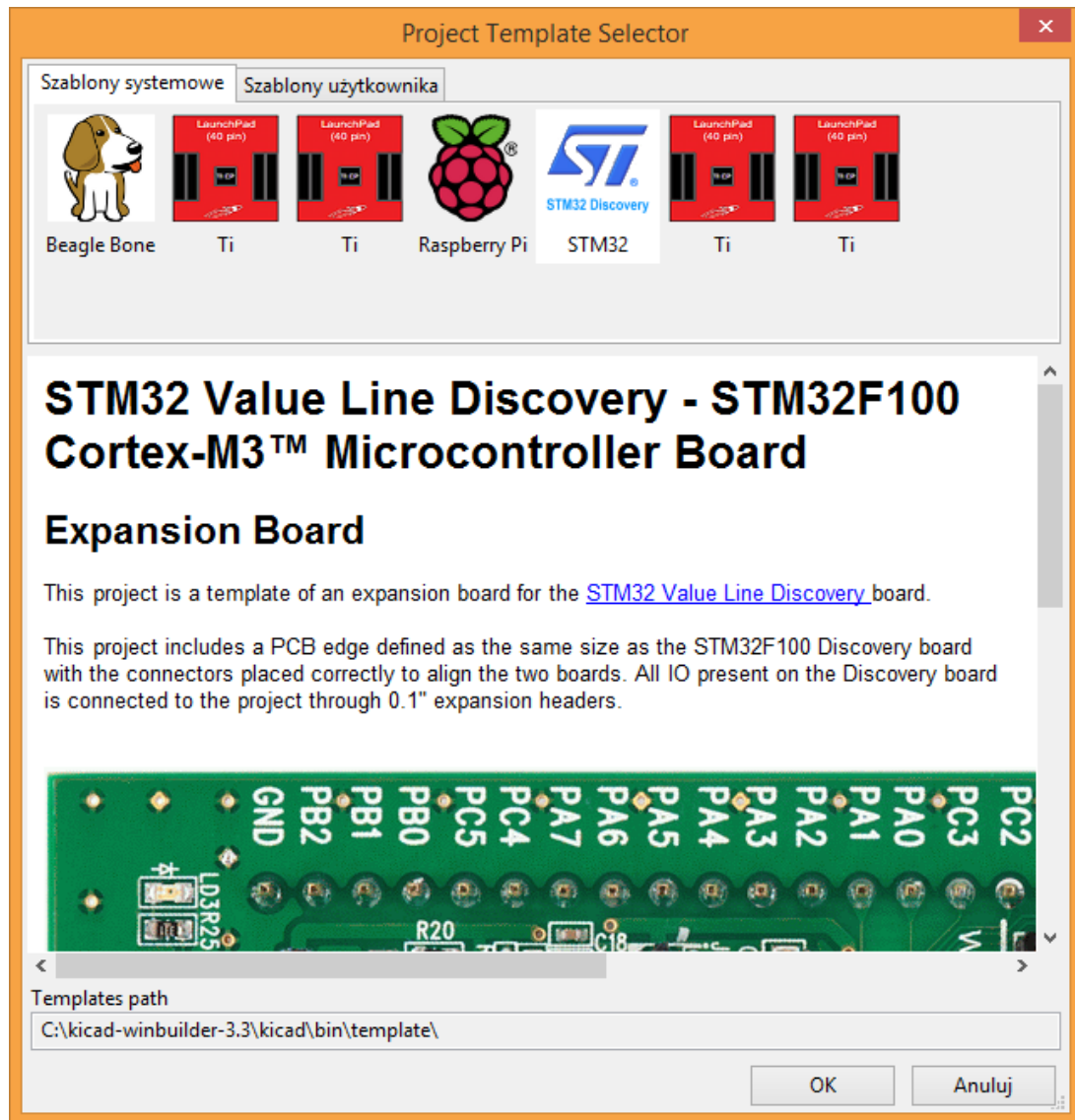
4.1 Using templates

Polecenie *Plik* → *Nowy projekt* → *Nowy za pomocą szablonu* otwiera okno dialogowe służące do wyboru szablonu:



A single click on a template's icon will display the template information, and a further click on the OK button creates the new project. The template files will be copied to the new project location and renamed to reflect the new project's name.

Po wybraniu jednego z szablonów:



4.2 Miejsce przechowywania szablonów:

KiCad looks for template files in the following paths:

- path defined in the environment variable `KICAD_USER_TEMPLATE_DIR`
- path defined in the environment variable `KICAD_TEMPLATE_DIR`
- Szablony systemowe: `<kicad bin dir>/../share/template/`
- Szablony użytkownika:
 - Unix: `~/kicad/templates/`
 - Windows: `C:\Documents and Settings\username\My Documents\kicad\templates`
 - Mac: `~/Documents/kicad/templates/`
- When the environment variable `KICAD_PTEMPLATES` is defined there is a third tab, Portable Templates, which lists templates found at the `KICAD_PTEMPLATES` path (DEPRECATED).

4.3 Tworzenie szablonów

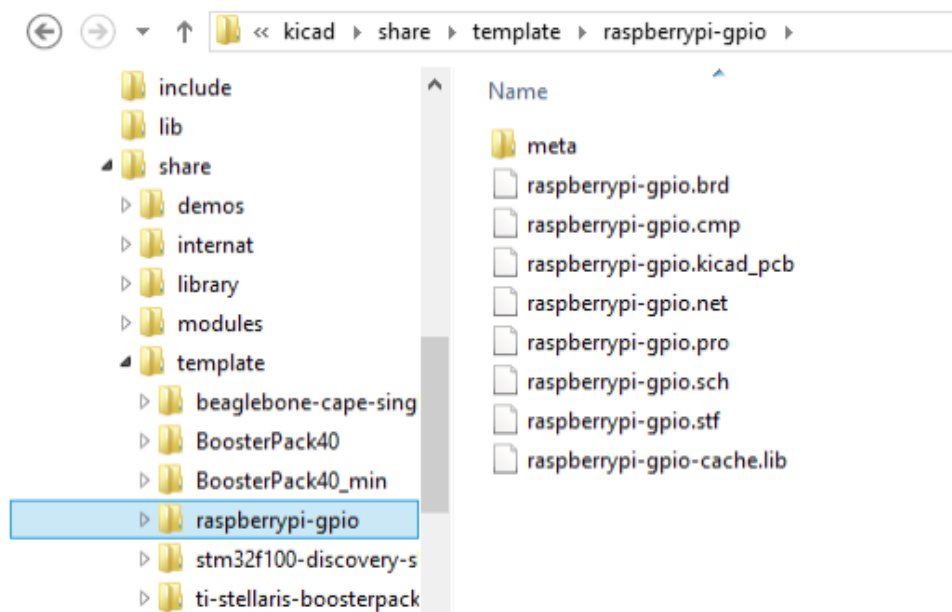
The template name is the directory name where the template files are stored. The metadata directory is a subdirectory named **meta** containing files describing the template.

Wszystkie pliki oraz katalogi składające się na definicję szablonu są kopiowane do nowego projektu podczas jego tworzenia z wykorzystaniem szablonu, z wyjątkiem metadanych.

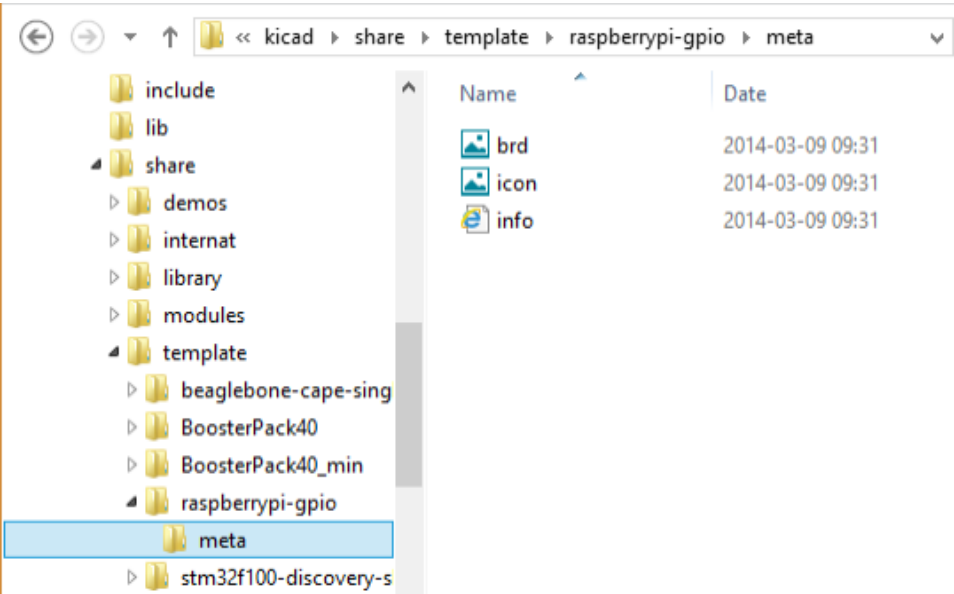
When a new project is created from a template, all files and directories starting with the template name will be renamed with the new project file name, excluding the file extension.

Folder metadanych musi zawierać jeden obowiązkowy plik, ale może zawierać także dodatkowe pliki. Wszystkie pliki muszą być utworzone przez użytkownika za pomocą edytora tekstu lub z pomocą gotowych projektów, oraz umieszczone w odpowiednio przygotowanej strukturze plików.

Here is an example showing project files for **raspberrypi-gpio** template:



Oraz jego katalog z metadanymi:



4.3.1 Wymagane pliki:

meta/info.html	Sformatowana w HTML informacja opisująca szablon.
----------------	---

Tag `<title>` określa właściwą nazwę szablonu, która jest pokazywana użytkownikowi podczas wyboru szablonu. Należy pamiętać, że zbyt długa nazwa zostanie obcięta. Z uwagi na rozmiar czcionki zwykle od 7 do 8 znaków może być wyświetlonych.

Użycie HTML oznacza, że można bardzo łatwo dodać ilustracje w opisach. Jednak trzeba mieć na uwadze, że podzbiór dozwolonych tagów HTML jest ograniczony.

Poniżej znajduje się przykład pliku `info.html`:

```
<!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>
<META NAME="GENERATOR" CONTENT="LibreOffice 3.6 (Windows)">
<META NAME="CREATED" CONTENT="0;0">
<META NAME="CHANGED" CONTENT="20121015;19015295">
</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>
```

4.3.2 Pliki opcjonalne:

meta/icon.png	Ikona o rozmiarze 64x64 pikseli, w formacie PNG, która zostanie użyta jako ikona na pasku wyboru szablonu w oknie dialogowym wyboru szablonu.
---------------	---

Każda inna ilustracja użyta w `meta/info.html`, tak jak przykładowa ilustracja płytki w oknie dialogowym, jest ta samo umieszczona w tym folderze.

Część I

Upgrading from Version 4 to Version 5

Changes were made to the behavior to KiCad during the version 5 development that can impact projects created with older versions of KiCad. This section serves as a guide to ensure the smoothest possible path when upgrading to version 5 of KiCad.

Rozdział 5

Schematic Symbol Libraries

Schematic symbol libraries are no longer accessed using a symbol (referred to as components in version 4) look up list. Symbol libraries are now managed by a symbol library table that behaves similarly to the footprint library table. This change is a significant improvement, but some schematics may need manual intervention when being converted to version 5.

In previous versions, KiCad used a list of library files to search when locating symbols in the Eeschema file. When locating a symbol, each path would be searched and the first library that held the symbol name would be used.

From v5, KiCad symbol names are prefixed with a nickname, and a [lookup table matching nicknames to library paths](#) is used to locate the library which holds the symbol. The table is called the *symbol library table* and built from configuration files stored in the user's KiCad configuration directory and the currently loaded project directory.

To upgrade a KiCad project from v4 to v5, nicknames for all of the library files need to be created and then schematic symbol names need to be prefixed with the correct nickname.

5.1 Global Symbol Library Table.

Eeschema v5 will automatically create a global symbol table when first started. You will be given a chance to skip this and create your own global symbol table by hand. You only need to do this if don't use KiCad symbol libraries at all. Otherwise it is easier to modify the automatically generated global symbol table.

Notatka

If you track the [symbol library repository](#), changes made to the default global symbol library table are not tracked by KiCad. You will have to manually keep the global symbol library table up to date.

5.2 Symbol Library Table Mapping

Automatic remapping of symbols will be executed whenever a schematic is opened that has not been remapped. There are a few steps you should take ahead of time in order for the remapping to be the most effective.

Notatka

If you have been using a development build of KiCad, copy the full default global symbol library table file (sym-lib-table) from the template folder installed with the KiCad libraries or from the [KiCad library repo](#) to your KiCad user configuration folder. This will replace the empty one (most likely) created by Eeschema. If you do not do this, you will most likely end up with a bunch of broken symbol links.

**Ostrzeżenie**

Remapped schematics will not be compatible with older versions of KiCad. The Remap Symbols dialog will make a backup of your schematic files and you should do the same if you remap manually.

1. If possible, keep version 4 of KiCad installed on your system unless you have never used any of the symbol libraries distributed with KiCad.
2. If you get warning about missing libraries when you start version 4 of Eeschema, make sure to fix the missing libraries if they contain symbols that are in the schematic before you attempt to remap your schematic. Otherwise, the correct symbol will not be found and you will end up with broken symbol links in your schematic. You can test this by left clicking on a symbol in the schematic and verifying that the symbol is not being loaded from the cache library. If a symbol is being loaded from the cache library, Eeschema cannot find your part in the system or project symbol libraries. If you need a cached part to be available to other projects on your system, you will need to integrate it into a system or project library manually.
3. If symbol recovery is required during the remapping process, do not dismiss it. Failure to recover symbols will result in broken symbol links or the wrong symbol being linked in the schematic.
4. During the remapping process, symbol libraries not found in the global symbol library table will be used to create a project specific symbol library table. You can move them manually to the global symbol library table if that is your preference.
5. For the most accurate remapping, create a project library by copying the project cache file (project-name-cache.lib) to a different file and add it to the top of the symbol library list. You must use a version of KiCad prior to the symbol library table implementation in order to do this.

Notatka

A tool has been provided to attempt to fix remapping issues. If there are missing symbol library links in a schematic, they can be fixed by opening the "Tools→Edit Symbol Library References..." menu entry and clicking on the "Map Orphans" button.

5.3 Remapping Search Order

When remapping symbols, KiCad proceeds in the following order to assign the library to a symbol:

1. Global Symbol Library Table: Symbols are preferentially mapped to the global symbol library table, if one exists.
-

2. Project specific libraries: Libraries listed in the project library list that are not in the global symbol library table are searched next.
3. Project cache file: If a symbol doesn't exist in the listed libraries above, it is first rescued — a copy is made from the cache and placed in the *proj-rescue.lib* — before the symbol is mapped to this new, rescue library.

5.4 Symbol Names and Symbol Library Nickname Limitations

Symbol names may not contain `<SPACE>`, `'`, `:`, `'`, `/`.

Library nicknames may not contain `<SPACE>`, `'`, `:`.

Existing symbol names with these characters must be renamed by manually editing the relevant schematic and library files.

Rozdział 6

Symbol Cache Library Availability

The cache library is no longer shown in either the symbol library viewer or the symbol library editor. The cache should never be edited because any changes are overwritten by the next schematic save.

Rozdział 7

Board File Format Changes

Several new features have been added to Pcbnew which impact the board file format. Using these new features in board designs will prevent them from being opened with previous versions of Pcbnew.

- Rounded rectangle footprint pads.
- Custom shape footprint pads.
- Footprint pad names longer than four characters.
- Keep out zones on more than a single layer.
- 3D models offset saved as millimeters instead of inches.
- Footprint text locking.

7.1 Global Footprint Library Table.

If you track the [footprint library repository](#), changes made to the default global footprint library table are not tracked by KiCad. You will have to manually keep the global footprint library table up to date.