



KiCad

4 listopada 2021

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	Konfiguracja ścieżek dostępu	5
2.5	Inicjalizacja zewnętrznych narzędzi	6
2.5.1	Wybór edytora tekstu	6
2.5.2	Wybór przeglądarki PDF	6
2.6	Tworzenie nowego projektu	7
2.7	Importowanie projektu spoza programu KiCad	7
3	Używanie Menadżera projektu	8
3.1	Okno Menadżera projektu	8
3.2	Panel uruchomieniowy	9
3.3	Drzewo projektu	10
3.4	Górny pasek narzędzi	10
4	Szablony użytkownika	11
4.1	Używanie szablonów	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	Aktualizacja z wersji 4 do wersji 5	17
5	Biblioteki programu Eeschema	19
5.1	Globalna tabela bibliotek symboli.	19
5.2	Mapowanie tabel bibliotek symboli	20
5.3	Reguły przeglądania bibliotek podczas remapowania	21
5.4	Ograniczenia w nazwach symboli i nazwach skrótowych	21
6	Dostępność pamięci podręcznej symboli	22
7	Zmiany w formacie plików PCB	23
7.1	Globalna tabela bibliotek footprintów.	23

Podręcznik użytkownika

Prawa autorskie

Copyright © 2010-2018. Ten dokument jest chroniony prawem autorskim. Lista autorów znajduje się poniżej. Możesz go rozpowszechniać oraz modyfikować na zasadach określonych w GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), wersja 3 lub późniejsza, albo określonych w Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), wersja 3.0 lub późniejsza.

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://gitlab.com/kicad/services/kicad-doc/issues>
- W sprawie oprogramowania: <https://gitlab.com/kicad/code/kicad/issues>
- W sprawie tłumaczeń programu: <https://gitlab.com/kicad/code/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 jest dostępny na platformach Linux, Windows oraz 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	Pliki schematu, które jednak nie zawierają kompletnych symboli.
*.lib	Pliki bibliotek symboli, które zawierają postacie symboli: kształty graficzne, piny, pola.
*.dcm	Pliki dokumentacji symboli, które zawierają niektóre pola pozwalające identyfikować symbole: komentarze, słowa kluczowe, odnośniki do not katalogowych.
*-cache.lib	Plik pamięci podręcznej z użytymi symbolami, zawierający kopie definicji symboli, które użyto w danym projekcie.
sym-lib-table	Lista bibliotek symboli (<i>Tabela bibliotek symboli</i>): lista bibliotek symboli jakie dostępne są w programie do edycji schematów.

Pliki i foldery edytora obwodów drukowanych:

*.kicad_pcb	Plik z obwodem drukowanym zawierający wszystkie informacje oprócz układu strony.
*.pretty	Biblioteki footprintów w postaci oddzielnych folderów. Folder sam w sobie jest taką biblioteką.
*.kicad_mod	Plik z definicją footprintu, zawierający zawsze tylko jeden footprint.
*.brd	Starszy plik z obwodem drukowanym, format Legacy. Można go odczytywać, ale nie ma możliwości jego zapisania.
*.mod	Starszy plik z biblioteką footprintów, format Legacy. Można go odczytywać, ale nie ma możliwości jego zapisania.
fp-lib-table	Lista bibliotek footprintów (<i>Tabela bibliotek footprintów</i>): Lista bibliotek footprintów (w różnych formatach), które są wczytywane przez edytor PCB lub edytor footprintów, lub 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	Plik z zapisanymi przypisaniami pomiędzy komponentami na schemacie i ich footprintami. Można go utworzyć za pomocą Pcbnew i zaimportować do Eeschema. Przeznaczeniem tego pliku jest import danych z Pcbnew do Eeschema, dla użytkowników którzy zmienili footprinty za pomocą Pcbnew (na przykład poleceniem <i>Zamień footprinty</i>) i chcą teraz uaktualnić dane o footprintach na schemacie ideowym.
-------	--

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

Silnik renderowania w Pcbnew oraz GerbView wykorzystujący wspomaganie sprzętowe wymaga by karta grafiki obsługiwała OpenGL v2.1 lub późniejsze.

2.2 Inicjalizacja domyślnej konfiguracji

Domyślny plik konfiguracji zwany **kicad.pro** jest dostarczany w folderze `'share/kicad/template'`. Służy jako plik bazowy dla każdego nowego projektu i jest używany do ustalenia listy bibliotek wczytywanych przez Eeschema. Oprócz tego zawiera również wiele innych parametrów dla pozostałych programów (domyślny rozmiar tekstu, domyślną szerokość linii, itp.).

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*.

Uruchom Eeschema za pomocą menadżera projektu. Zmodyfikuj i uaktualnij konfigurację Eeschema, głównie listę bibliotek, które chciałbyś używać w każdym nowym projekcie.

Uruchom Pcbnew za pomocą programu KiCad. Zmodyfikuj i uaktualnij konfigurację Pcbnew, zwłaszcza listę dostępnych bibliotek. Pcbnew utworzy plik z listą bibliotek zwaną **tabelą bibliotek footprintów**. Istnieją dwa pliki takiej listy (nazwane *fp-lib-table*). Pierwszy (umieszczony w folderze domowym danego użytkownika) jest plikiem globalnym dostępnym dla wszystkich projektów. Drugi z nich, jeśli istnieje (umieszczony w folderze z projektem) dotyczy tylko danego projektu.

2.4 Konfiguracja ścieżek dostępu

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 ...

Jest to pomocne w przypadku gdy ścieżki absolutne nie są znane lub zmieniają się (np. gdy projekt jest przenoszony z jednego komputera na inny), a także gdy ścieżka bazowa jest wspólna dla wielu podobnych elementów. Rozważmy następujące elementy, które mogą być instalowane w różnych miejscach:

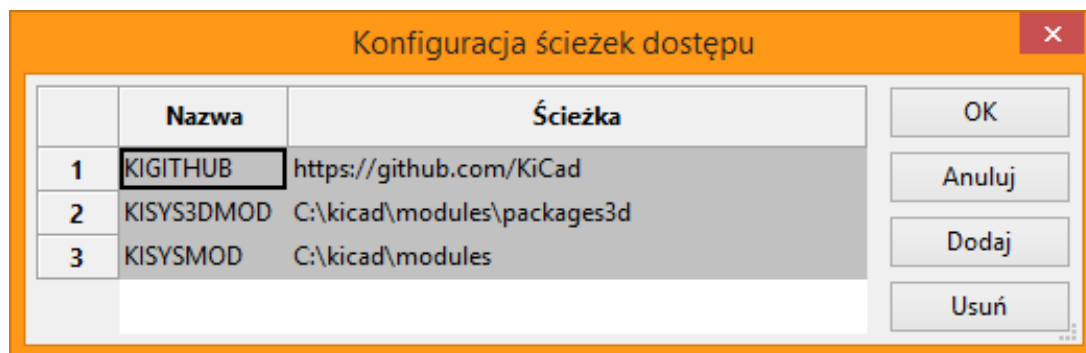
- Biblioteki programu Eeschema
- Biblioteki footprintów programu Pcbnew
- Pliki z kształtami 3D używane w definicjach footprintów

Przykładowo, pełna ścieżka do biblioteki *Connect.pretty* gdy wykorzystuje się zmienną systemową **KISYSMOD** powinna być zapisana jako **`${KISYSMOD}/Connect.pretty`**

Opcja ta pozwala na zdefiniowanie kilku ścieżek dostępu za pomocą zmiennych systemowych, oraz dodanie własnych zmiennych określających na przykład ścieżki użytkownika, jeśli są używane.

Zmienne systemowe programu KICAD:

KICAD_PTEMPLATES	Szablony przenośne używane podczas tworzenia projektów. (WYCOFANA razem z wersją 5.0.0-rc2, zamiast niej użyj KICAD_TEMPLATE_DIR) Jeśli jest nadal używana musi zostać zdefiniowana.
KICAD_SYMBOL_DIR	Ścieżka bazowa dla bibliotek zawierających symbole schematowe.
KIGITHUB	Jest często używana w przykładach tabel bibliotek footprintów. Jeśli zmienna jest używana musi zostać zdefiniowana.
KISYS3DMOD	Jest domyślną, bazową ścieżką do plików z kształtami 3D, i musi zostać zdefiniowana, ponieważ ścieżka absolutna nie jest zwykle używana.
KISYSMOD	Jest domyślną, bazową ścieżką do folderów z bibliotekami footprintów, i musi zostać zdefiniowana, jeśli w nazwach bibliotek nie jest używana ścieżka absolutna.
KICAD_TEMPLATE_DIR	Określa lokalizację szablonów instalowanych razem z programem KiCad.
KICAD_USER_TEMPLATE_DIR	Określa lokalizację szablonów użytkownika.



Należy pamiętać, że zmienna **KIPRJMOD** jest **zawsze** wewnętrznie definiowana przez program KiCad, i wskazuje

na absolutną ścieżkę dostępu do bieżącego projektu.

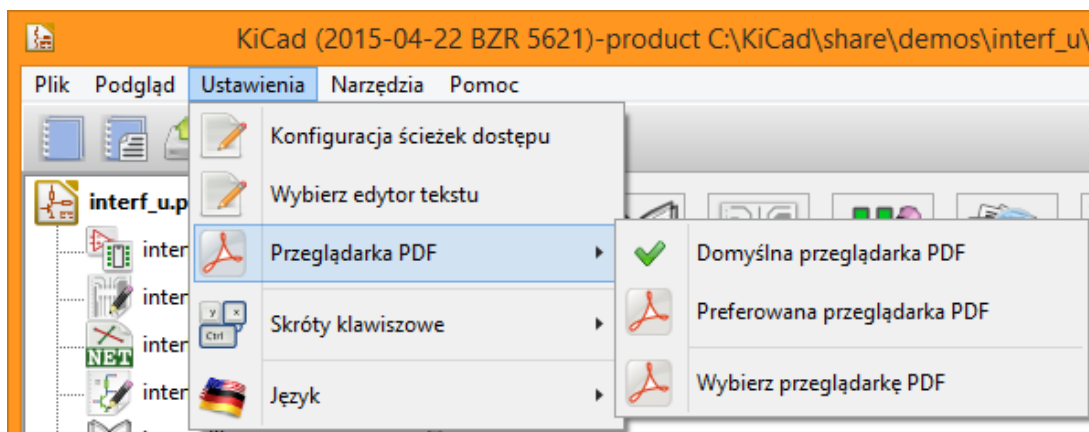
Przykładowo, `${KIPRJMOD}/Connect.pretty` jest zawsze odnośnikiem do folderu *Connect.pretty* (Biblioteka footprintów typu Pretty) *wewnątrz folderu bieżącego projektu*.

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

Można zdefiniować swój ulubiony edytor tekstu lub przeglądarkę PDF. Ustawienia te są używane gdy zachodzi potrzeba użycia edytora tekstu lub podczas podglądu plików PDF.

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 Tworzenie nowego projektu

Aby utworzyć i zarządzać prostym projektem z jego wszystkimi elementami, tj. schemat, PCB, potrzebne biblioteki, pliki produkcyjne: Gerber, pliki wierceń, pliki położeń elementów; zalecane jest utworzenie nadrzędnego projektu w następujący sposób:

- **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 Importowanie projektu spoza programu KiCad

KiCad umożliwia import plików utworzonych w innych narzędziach EDA. Obecnie obsługiwane są tylko pliki programu EAGLE 6.x lub nowszego (z zawartością XML).

By zaimportować obcy projekt, należy za pomocą standardowego okna wyboru pliku wybrać albo plik ze schematem, albo plik z obwodem drukowanym. Importowany schemat i płytkę powinna posiadać taką samą nazwę bazową (np. `project.sch` i `project.brd`). Jak tylko pliki zostaną wybrane, program zapyta do folder docelowy, gdzie zostaną zapisane zaimportowane pliki, które zostaną przekształcone w pliki formatu KiCad.

Rozdział 3

Używanie Menadżera projektu

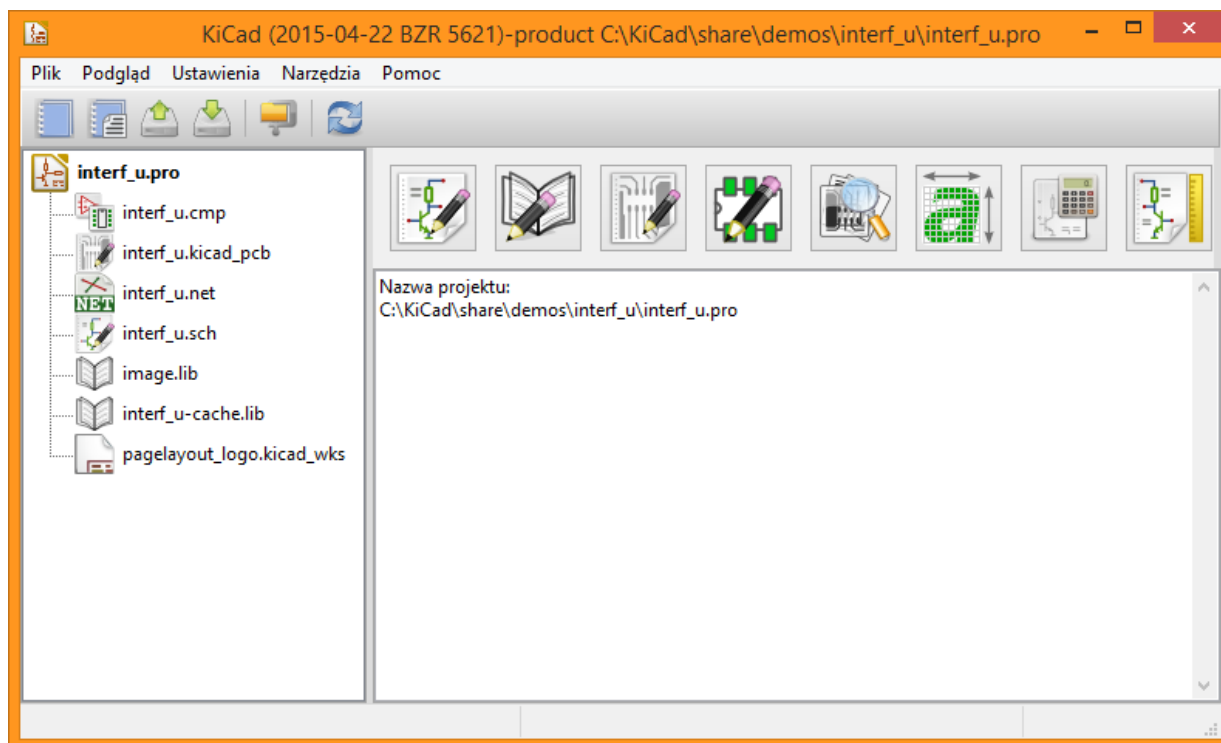
KiCad Menadżer Projektu (plik `kicad` lub `kicad.exe`) to narzędzie do zarządzania całym projektem, z którego można łatwo uruchomić pozostałe aplikacje wchodzące w skład KiCad EDA Suite (edytory, przeglądarkę Gerber i inne dodatkowe narzędzia).

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 *samo-dzielne aplikacje* można otworzyć dowolny plik z dowolnego projektu ale “cross probing” może dać niespodziewane rezultaty.

3.1 Okno Menadżera projektu



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

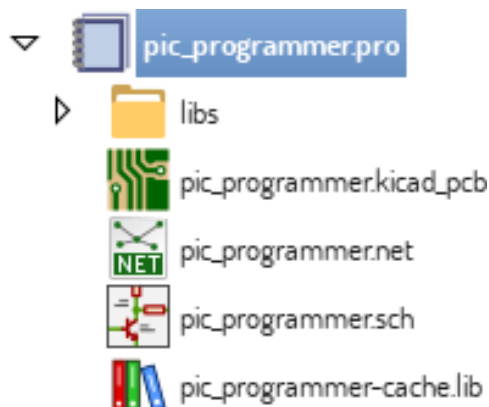
KiCad pozwala na uruchamianie wszystkich składników z jakimi został dostarczony.

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	Gerbview	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



Kliknięcie podwójne na ikonę Eeschema uruchomi edytor schematów, w tym wypadku otwierając automatycznie plik *pic_programmer.sch*.







Kliknięcie podwójne na ikonę Pbcnew uruchomi edytor obwodów drukowanych PCB, w tym wypadku otwierając automatycznie plik *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):

	Tworzy plik projektu. Jeśli domyślny szablon (kicad.pro) został znaleziony w kicad/template , jest on kopiowany do katalogu roboczego projektu.
	Tworzy projekt na podstawie gotowego szablonu.
	Otwiera istniejący projekt.
	Aktualizuje i zapisuje bieżące drzewo projektu.
	Tworzy archiwum ZIP całego projektu. Archiwum zawiera pliki schematów, biblioteki, obwód drukowany, itd.
	Odświeża i przerysowuje drzewo projektu, czasem wymagane przy zmianach w drzewie projektu.

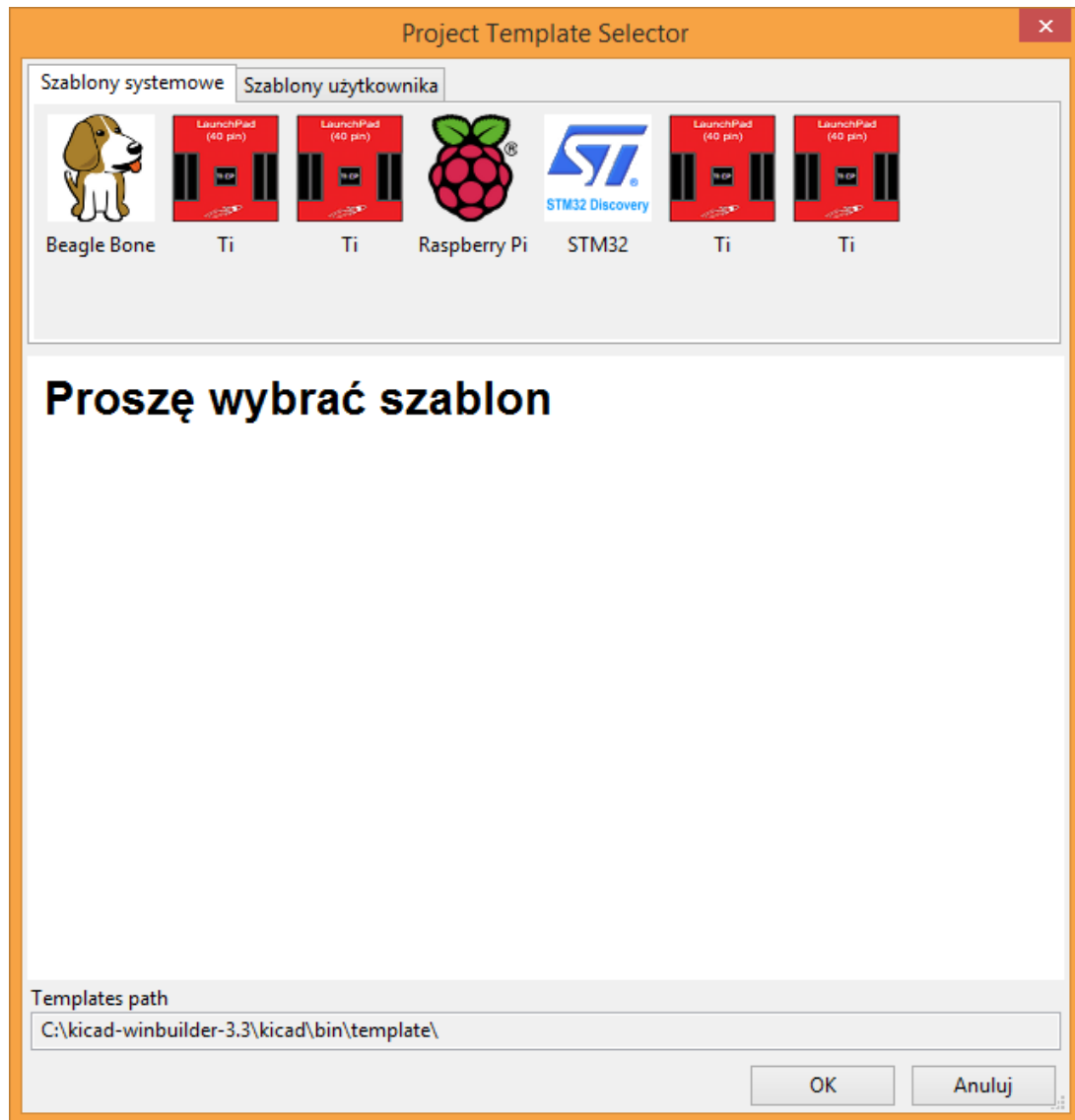
Rozdział 4

Szablony użytkownika

Użycie szablonu ułatwia konfigurację nowego projektu bazując na predefiniowanych ustawieniach. Szablony mogą zawierać wstępnie zdefiniowane kontury obwodu drukowanego, położenia specyficznych złącz, wstępnie narysowane fragmenty schematu, wstępnie ustalone zasady projektowe, itp. Możliwe jest również skorzystanie z kompletnych schematów jak i PCB jako źródeł plików szablonu.

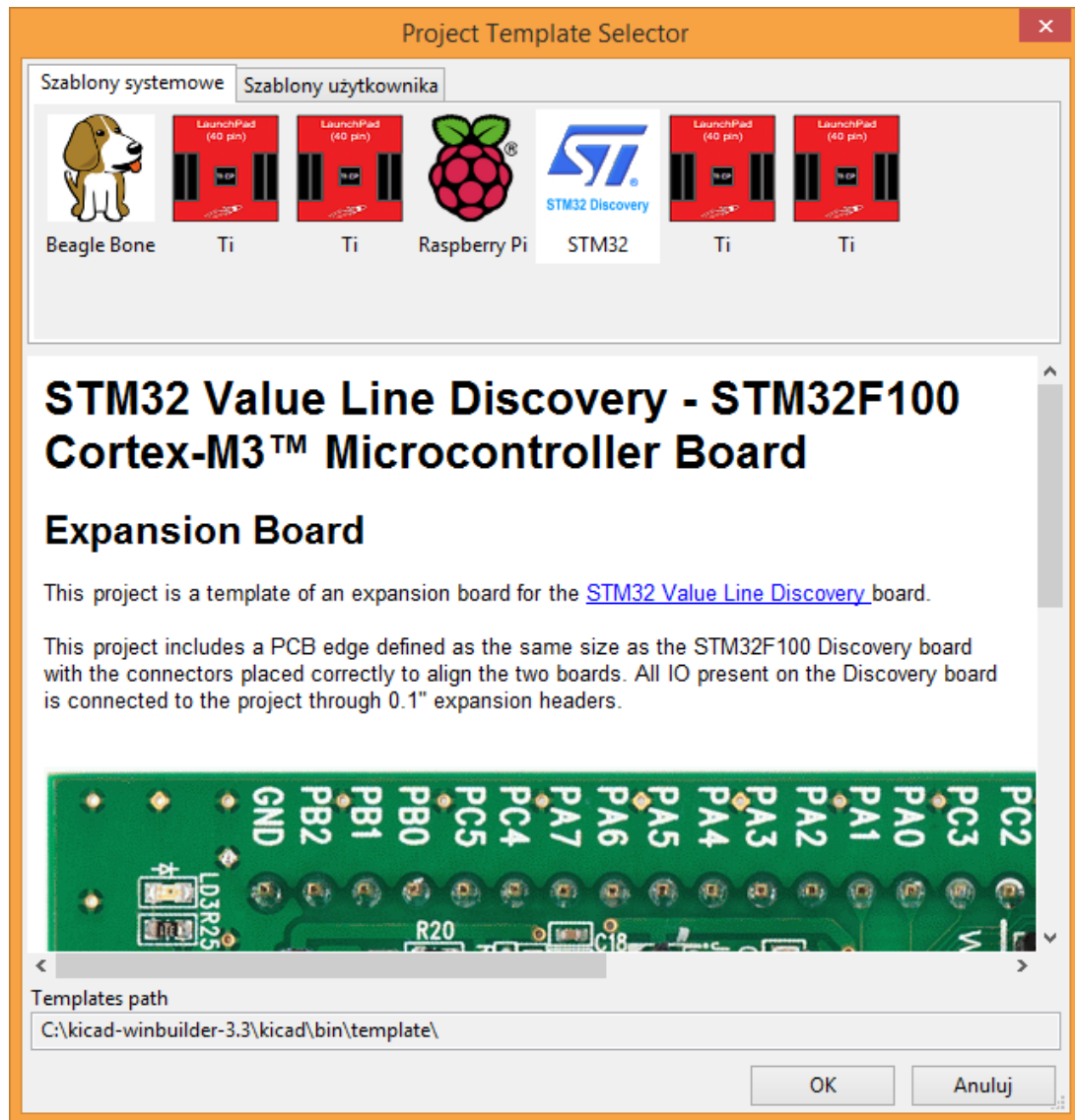
4.1 Używanie szablonów

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



Pojedyncze kliknięcie na ikonę szablonu powoduje pokazanie informacji o zawartości szablonu. Późniejsze kliknięcie przycisku OK utworzy nowy projekt. Pliki szablonu zostaną skopiowane do nowej lokalizacji projektu oraz zostaną im zmienione nazwy plików na zgodne z nazwą projektu.

Po wybraniu jednego z szablonów:



4.2 Miejsce przechowywania szablonów:

KiCad przy wyszukiwaniu szablonów przeszukuje następujące ścieżki:

- ścieżka zdefiniowana w zmiennej KICAD_USER_TEMPLATE_DIR
- ścieżka zdefiniowana w zmiennej 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/

- Jeśli w systemie zdefiniowano specjalną zmienną środowiskową `KICAD_PTEMPLATES`, zostaje dodana trzecia zakładka „Szablony Przenośne”, w której znajdują się wszystkie szablony odnalezione w ścieżce do której odnosi się `KICAD_PTEMPLATES` (Opcja PRZESTARZAŁA, obecnie zaleca się używania zmiennej `KICAD_USER_TEMPLATE_D`).

4.3 Tworzenie szablonów

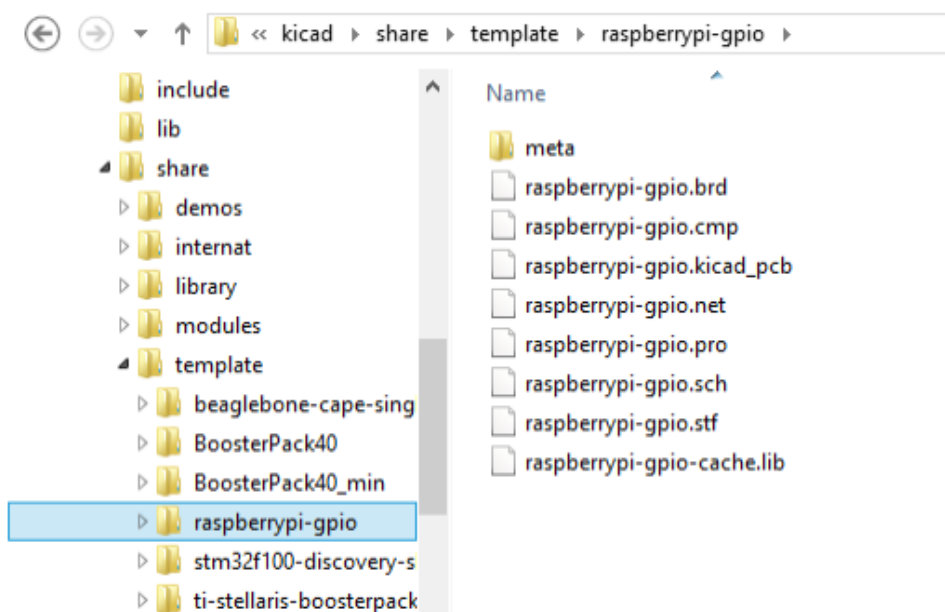
Nazwa szablonu jest nazwą katalogu, w którym zapisano poszczególne pliki wchodzące w jego skład. Wewnętrzny katalog metadanych, w podfolderze zwanym **meta**, zawiera specyficzne pliki dostarczające podstawowych informacji o szablonie.

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

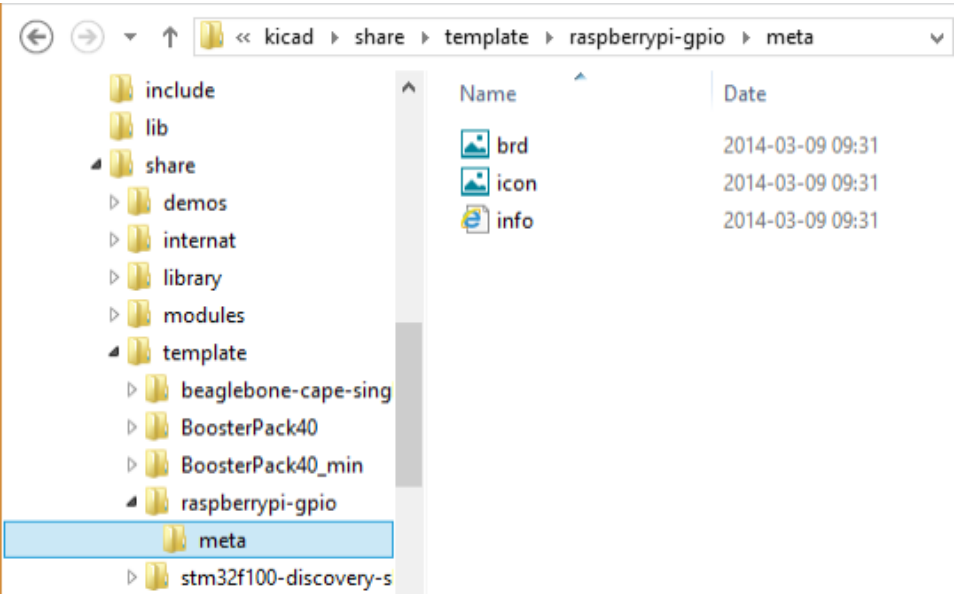
Podczas kopiowania plików do miejsca docelowego (ścieżki docelowej), wszystkim plikom oraz katalogom, które pasują nazwą do nazwy szablonu zostaną zmienione nazwy na zgodne z nazwą nowego projektu. Pozostałe pliki i rozszerzenia plików pozostaną nienaruszone.

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.

Poniżej znajduje się przykładowa zawartość głównego katalogu z szablonem płytki rozszerzającej Raspberry-Gpio:



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>
</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

Aktualizacja z wersji 4 do wersji 5

Wprowadzono zmiany w zachowaniu programu KiCad podczas opracowywania wersji 5, które mogą wpłynąć na projekty utworzone za pomocą starszych wersji programu KiCad. Ta sekcja służy jako przewodnik zapewniający płynne przejście podczas aktualizacji do wersji 5 programu KiCad.

Rozdział 5

Biblioteki programu Eeschema

Biblioteki symboli nie są już dostępne za pomocą listy symboli (określanego jako komponenty w wersji 4). Biblioteki symboli są teraz zarządzane przez tabelę bibliotek symboli, która zachowuje się podobnie do tablicy biblioteki footprintów. Zmiana ta jest znaczną poprawą, ale niektóre schematy mogą wymagać ręcznej interwencji podczas konwersji do wersji 5.

W wersji 4.0 i również wcześniejszych, biblioteki symboli były dostępne poprzez listę aktywnych bibliotek dla danego projektu. Można było określić listę bibliotek wchodzących w skład danego projektu i ustawić im priorytet. Lista była przeglądana z góry na dół i pierwsze wystąpienie danego symbolu było uznawane jako te właściwe.

Od wersji 5 nazwy symboli KiCad są poprzedzane nazwą skrótową i aby zlokalizować bibliotekę zawierającą symbol używana jest **lista porównująca pasujące nazwy skrótowe do ścieżek bibliotek**. Tabela nosi nazwę *tabeli bibliotek symboli* i została zbudowana z plików konfiguracyjnych przechowywanych w katalogu konfiguracyjnym KiCada użytkownika i aktualnie załadowanym katalogu projektu.

Aby zaktualizować projekt KiCad z wersji 4 do wersji 5, należy utworzyć nazwy skrótowe dla wszystkich plików bibliotecznych, by następnie nazwy symboli zostały poprzedzone poprawnym prefiksem.

5.1 Globalna tabela bibliotek symboli.

Eeschema w wersji 5 automatycznie tworzy globalną tablicę symboli przy pierwszym uruchomieniu. Można ten proces pominąć i ręcznie stworzyć swoją własną globalną tabelę bibliotek symboli. Trzeba to zrobić wyłącznie gdy w ogóle nie korzystamy z domyślnych bibliotek symboli programu KiCad. W przeciwnym razie łatwiej jest zmodyfikować automatycznie wygenerowaną globalną tablicę symboli.

Notatka

Jeśli na bieżąco obserwujesz **repozytorium bibliotek symboli**, to musisz pamiętać, że zmiany dokonane w domyślnej tabeli globalnej biblioteki symboli nie są śledzone przez KiCad. Będziesz musiał ręcznie aktualizować tabelę globalnej biblioteki symboli gdy zauważysz zmiany.

5.2 Mapowanie tabel bibliotek symboli

Automatyczne remapowanie symboli będzie wykonywane po każdym otwarciu schematu, który nie został jeszcze zmapowany. Jest kilka kroków, które powinieneś zrobić z wyprzedzeniem, aby zmiana przypisania była najbardziej efektywna.

Notatka

Jeśli korzystasz z wersji rozwojowej KiCada, skopiuj pełny domyślny plik tabeli globalnej biblioteki symboli (sym-lib-table) z folderu szablonu zainstalowanego w bibliotekach KiCad lub z [repozytorium bibliotek programu KiCad](#) do folderu konfiguracji użytkownika programu KiCad. To zastąpi pustą (najprawdopodobniej) tabelę stworzoną przez Eeschema. Jeśli tego nie zrobisz, najprawdopodobniej skończy się zepsutymi łączami do poszczególnych symboli.



Ostrzeżenie

Zmienione schematy nie będą kompatybilne ze starszymi wersjami programu KiCad. Okno dialogowe Remapowanie Symboli stworzy kopię zapasową plików schematów i powinieneś zrobić to samo, jeśli remapujesz je ręcznie.

1. Jeśli to możliwe, utrzymuj wersję 4 programu KiCad zainstalowaną w systemie, chyba że nigdy nie używałeś żadnej biblioteki symboli dystrybuowanej z programem KiCad.
2. Jeśli pojawi się ostrzeżenie o brakujących bibliotekach po uruchomieniu wersji 4 programu Eeschema, pamiętaj o naprawieniu brakujących bibliotek jeśli zawierają one symbole, które są na schemacie zanim spróbujesz zmienić schemat. W przeciwnym razie właściwy symbol nie zostanie znaleziony, a na jego miejscu pojawią się symbole uszkodzonego łącza. Możesz to sprawdzić klikając lewym przyciskiem na symbol na schemacie, sprawdzając czy symbol nie jest ładowany z biblioteki pamięci podręcznej. Jeśli symbol jest ładowany z biblioteki pamięci podręcznej, Eeschema nie będzie mógł znaleźć twojego symbolu w systemowych bibliotekach symboli lub w bibliotekach projektu. Jeśli potrzebujesz zburowanej części, aby była dostępna dla innych projektów w twoim systemie, musisz ręcznie zintegrować ją z biblioteką systemu lub projektu.
3. Jeśli odzyskiwanie symbolu jest wymagane podczas procesu remapowania nie należy go odrzucać. Nieodzyskanie symboli spowoduje zerwanie łącz symbolicznych lub przypisanie niewłaściwego symbolu na schemacie.
4. Podczas procesu ponownego mapowania, biblioteki symboli które nie zostały znalezione w tabeli globalnej biblioteki symboli zostaną użyte do utworzenia tabeli bibliotek symboli specyficznych dla projektu. Możesz przenieść je ręcznie do tabeli globalnej biblioteki symboli, jeśli preferujesz takie rozwiązanie.
5. Aby uzyskać najdokładniejszy rezultat mapowania, utwórz bibliotekę projektu, kopiując plik pamięci podręcznej projektu (`project-name-cache.lib`) do innego pliku i dodaj go na początek listy bibliotek symboli. Musisz użyć wersji programu KiCad sprzed implementacji tabeli bibliotek symboli aby to zrobić.

Notatka

Program dostarcza narzędzia by spróbować naprawić problemy z ponownym mapowaniem. Jeśli na schemacie brakuje linków do bibliotek symboli, można je naprawić, otwierając menu **Narzędzia** → **Edycja odnośników w bibliotece symbol...**, a następnie kliknąć przycisk *Mapuj osierocone symbole*

5.3 Reguły przeglądania bibliotek podczas remapowania

Podczas ponownego mapowania symboli, KiCad wykorzystuje następującą kolejność wyszukiwania by przypisać bibliotekę do symbolu:

1. Tabela globalnej biblioteki symboli: symbole są mapowane na pierwszym miejscu do tabeli globalnej biblioteki symboli, jeśli taka istnieje.
2. Biblioteki specyficzne dla projektu: biblioteki wymienione na liście bibliotek projektów, które nie znajdują się w tabeli globalnych bibliotek symboli, są przeszukiwane jako następne.
3. Plik pamięci podręcznej projektu: Jeśli symbol nie istnieje w wymienionych bibliotekach powyżej, jest on najpierw odzyskiwany - kopia jest tworzona z pamięci podręcznej i umieszczana w *proj-rescue.lib* - zanim symbol zostanie odwzorowany do nowej biblioteki z symbolami odzyskanymi.

5.4 Ograniczenia w nazwach symboli i nazwach skrótowych

Nazwy symbolu nie mogą zawierać znaków `<SPACE>`, `' : '`, `' / '`.

Nazwy skrótowe nie mogą zawierać znaków `<SPACE>`, `' : '`.

Istniejące nazwy symboli z tymi znakami muszą zostać zmienione poprzez ręczną edycję odpowiednich plików schematów i bibliotek.

Rozdział 6

Dostępność pamięci podręcznej symboli

Biblioteka pamięci podręcznej nie jest już wyświetlana ani w przeglądarce bibliotek symboli, ani w edytorze bibliotek symboli. Pamięć podręczna nie powinna być nigdy edytowana, ponieważ wszelkie zmiany są nadpisywane przez kolejne zapisanie schematu.

Rozdział 7

Zmiany w formacie plików PCB

Kilka nowych funkcji zostało dodanych do Pcbnew, które mają wpływ na format pliku PCB. Korzystanie z tych nowych funkcji w projektach PCB uniemożliwi ich otwarcie przy użyciu poprzednich wersji programu Pcbnew.

- Pola lutownicze w formie zaokrąglonego prostokąta.
- Pola lutownicze o kształcie zdefiniowanym przez użytkownika.
- Nazwy pól dłuższe niż 4 znaki.
- Strefy chronione na więcej niż jednej warstwie.
- Przesunięcia modeli 3D są zapisywane w milimetrach a nie w calach.
- Blokowanie tekstów footprintów.

7.1 Globalna tabela bibliotek footprintów.

Jeśli samodzielnie obserwujesz [repozytorium bibliotek footprintów](#), to musisz pamiętać, że zmiany dokonane w domyślnej tabeli bibliotek footprintów nie są śledzone przez program KiCad. Będziesz musiał ręcznie je aktualizować..