

CvPcb

Narzędzie do przypisywania footprintów komponentom

Spis treści

1. Przeznaczenie
2. Ogólna charakterystyka aplikacji
3. Uruchamianie CvPcb
4. Obsługa CvPcb
 - Ekran główny
 - Główny pasek narzędzi
 - Konfigurowanie CvPcb - Wersja bez obsługi Tabeli Bibliotek
 - Główne okno
 - Wybór bibliotek footprintów
 - Wybór plików skryptów przypisań
 - Wybór domyślnej ścieżki bibliotek footprintów
 - Ścieżki poszukiwań
 - Ścieżki użytkownika
 - Ścieżki ustalone automatycznie przez CvPcb
 - Tabele Bibliotek - Zarządzanie bibliotekami po nowemu
 - Globalna tabela bibliotek footprintów
 - Konfiguracja początkowa
 - Lokalna tabela bibliotek footprintów zależna od projektu
 - Dodawanie nowych wpisów w tabeli
 - Dodawanie nowych wpisów w tabeli z pomocą kreatora
 - Pobieranie wartości ze zmiennych systemowych
 - Używanie wtyczki GitHub
 - Przykładowe wpisy w tabeli
 - Co robić z plikami w COW?
 - Generalne zalecenia przy używaniu tabeli bibliotek
 - Podgląd bieżącego footprintu
 - Wyświetlanie
 - Skróty klawiaturowe
 - Menu kontekstowe
 - Pasek narzędzi

- Podgląd 3D

5. Przypisywanie footprintów do komponentów

- Podstawy
- Operacja przypisywania
- Zmiana bieżącego przypisania
- Filtrowanie listy footprintów
 - Filtrowanie za pomocą wzorców z ustawień komponentów
 - Filtrowanie za pomocą pasującej liczby wyprowadzeń
 - Filtrowanie za pomocą wybranej biblioteki

6. Automatyczne przypisywanie footprintów

- Pliki skryptów przypisań
- Format plików
- Przypisywanie automatyczne

7. Pliki numeracji wstecznej

O dokumencie

Prawa autorskie

Copyright © 2010-2015. 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.

Autorzy

Jean-Pierre Charras, Wayne Stambaugh, Fabrizio Tappero, Kerusey Karyu.

Kontakt

Wszelkie komentarze lub sugestie dotyczące tego dokumentu prosimy kierować na listę dyskusyjną deweloperów programu KiCad <https://launchpad.net/~kicad-developers>

Wersja

22 Wrzesień 2015

Informacja dla użytkowników komputerów Apple Macintosh

Wsparcie programu KiCad dla systemu operacyjnego Apple OS X jest w zaawansowanej fazie eksperymentalnej.

1. Przeznaczenie

CvPcb to narzędzie pozwalające przypisać poszczególnym komponentom na schemacie odpowiadające im footprinty, które odzwierciedlają je na obwodzie drukowanym. Przypisanie to jest później umieszczane na liście sieci, którą należy utworzyć z poziomu programu **Eeschema** przed rozpoczęciem pracy nad płytką.

2. Ogólna charakterystyka aplikacji

Lista sieci generowana przez **Eeschema** może określić, który footprint na obwodzie drukowanym będzie reprezentował dany komponent tylko jeśli specjalne pola z nazwą footprintu zostaną zainicjalizowane.

Taka sytuacja będzie miała miejsce, gdy pole z nazwą zostanie wypełnione już w trakcie tworzenia schematu, albo zostanie wypełnione już podczas projektowania symbolu. Wtedy użycie **CvPcb** nie będzie konieczne.

Użycie do tego celu narzędzia jakim jest **CvPcb** jest jednak znacznie wygodniejsze niż wypełnianie tych pól na bieżąco lub predefiniowanie ich w bibliotekach symboli. **CvPcb** dostarcza kilka pożytecznych opcji jakimi niewątpliwie są: filtrowanie list footprintów, podgląd footprintów - także jako modeli 3D, co znacznie upraszcza cały proces.

Komponenty można przypisywać manualnie albo półautomatycznie z wykorzystaniem **skryptów przypisań** (pliki `.equ`). Skrypty te są prostą tabelą postawną, w której można sparować określony typ komponentu z jednym z footprintów.

Taki interaktywny proces jest znacznie prostszy niż bezpośrednie przypisywanie tych informacji z poziomu schematu a także przyczynia się do zmniejszenia liczby błędów z tym związanych.

3. Uruchamianie CvPcb

CvPcb jest uruchamiany wyłącznie z poziomu programu **Eeschema** za pomocą odpowiedniego polecenia dostępnego na górnym paski narzędzi, zarówno jeśli **Eeschema** jest uruchamiany z poziomu Menadżera projektu lub jako samodzielna aplikacja.

Generalnie, uruchamianie **CvPcb** z **Eeschema**, który został uruchomiony za pomocą Menadżera projektu jest lepszym rozwiązaniem, ponieważ:

- **CvPcb** potrzebuje pliku z projektem, by znać listę dostępnych bibliotek footprintów - zwłaszcza tych, które należą do bibliotek lokalnych projektu.
- **CvPcb** za pomocą wewnętrznych mechanizmów przekazuje informacje o wybranych footprintach bezpośrednio do schematu. Taka operacja jest możliwa **tylko** gdy plik projektu znajduje się w tym samym folderze co otwarty schemat.

Przy uruchomieniu **CvPcb** z **Eeschema** wszystkie powyższe zależności zostaną spełnione.

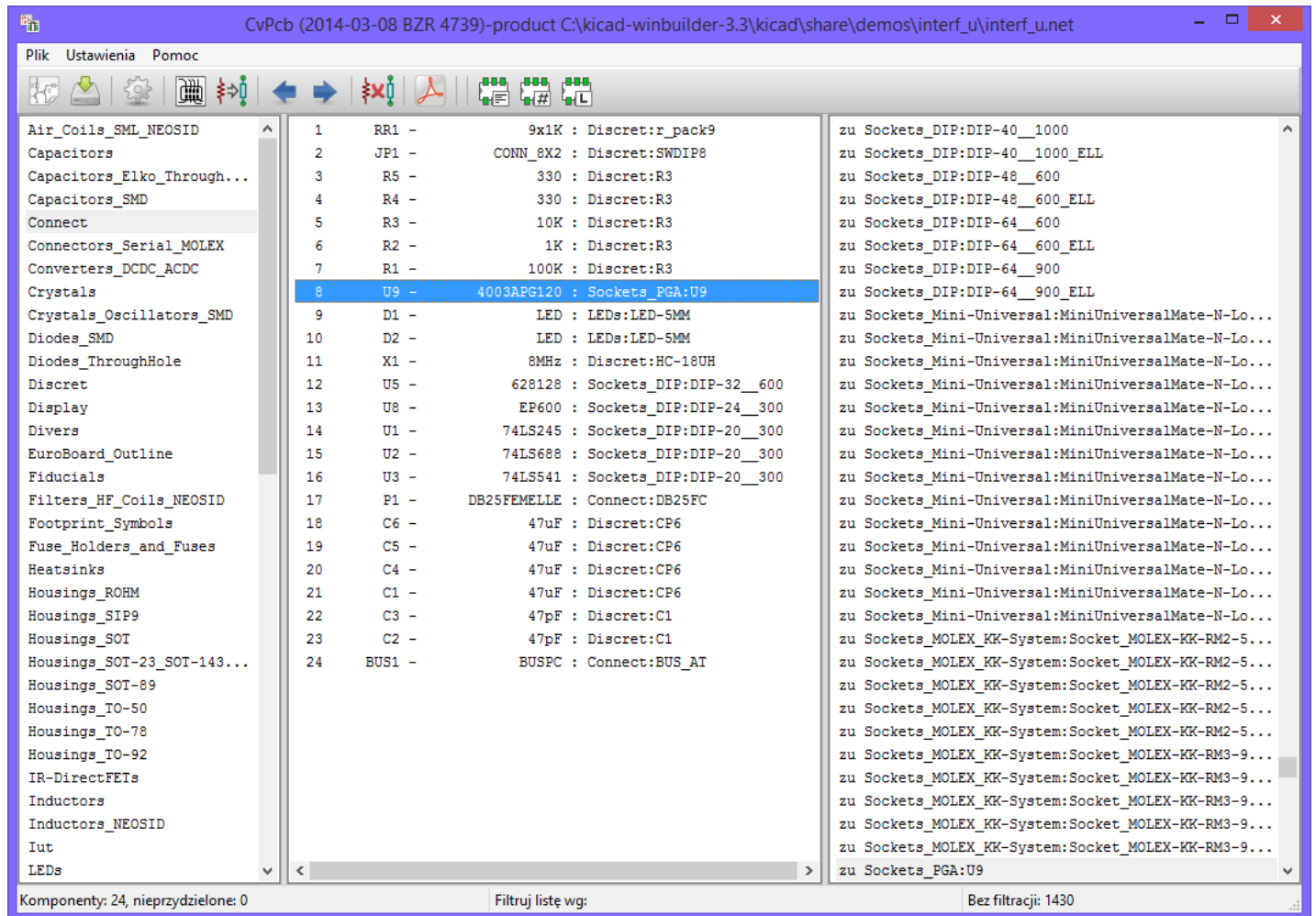
*Można uruchomić program **CvPcb** z samodzielnej sesji programu **Eeschema**, lecz*



należy pamiętać, że dowolny schemat otwarty w ten sposób, który w tej samej ścieżce dostępu nie będzie posiadał pliku projektu może być pozbawiony niektórych komponentów z powodu braku dostępu do plików bibliotek symboli. Brakujące komponenty nie będą ukazywane na liście komponentów w **CvPcb**. Dodatkowo, jeśli zabraknie również pliku **fp-lib-table** żadna z bibliotek powiązanych z projektem również nie będzie dostępna.

4. Obsługa CvPcb

Ekran główny

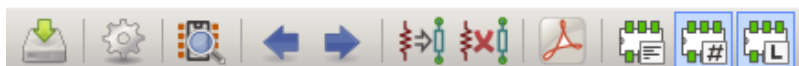


Panel listy bibliotek (z lewej strony) zawiera listę dostępnych w projekcie bibliotek. Panel ten współpracuje z opcją filtrowaniem według bibliotek.

Panel komponentów (w środku) zawiera listę komponentów odczytanych z listy sieci.

Panel footprintów (z prawej strony) zawiera listę modułów odczytanych z dostępnych bibliotek. Zawartość tej listy może być filtrowana.

Główny pasek narzędzi



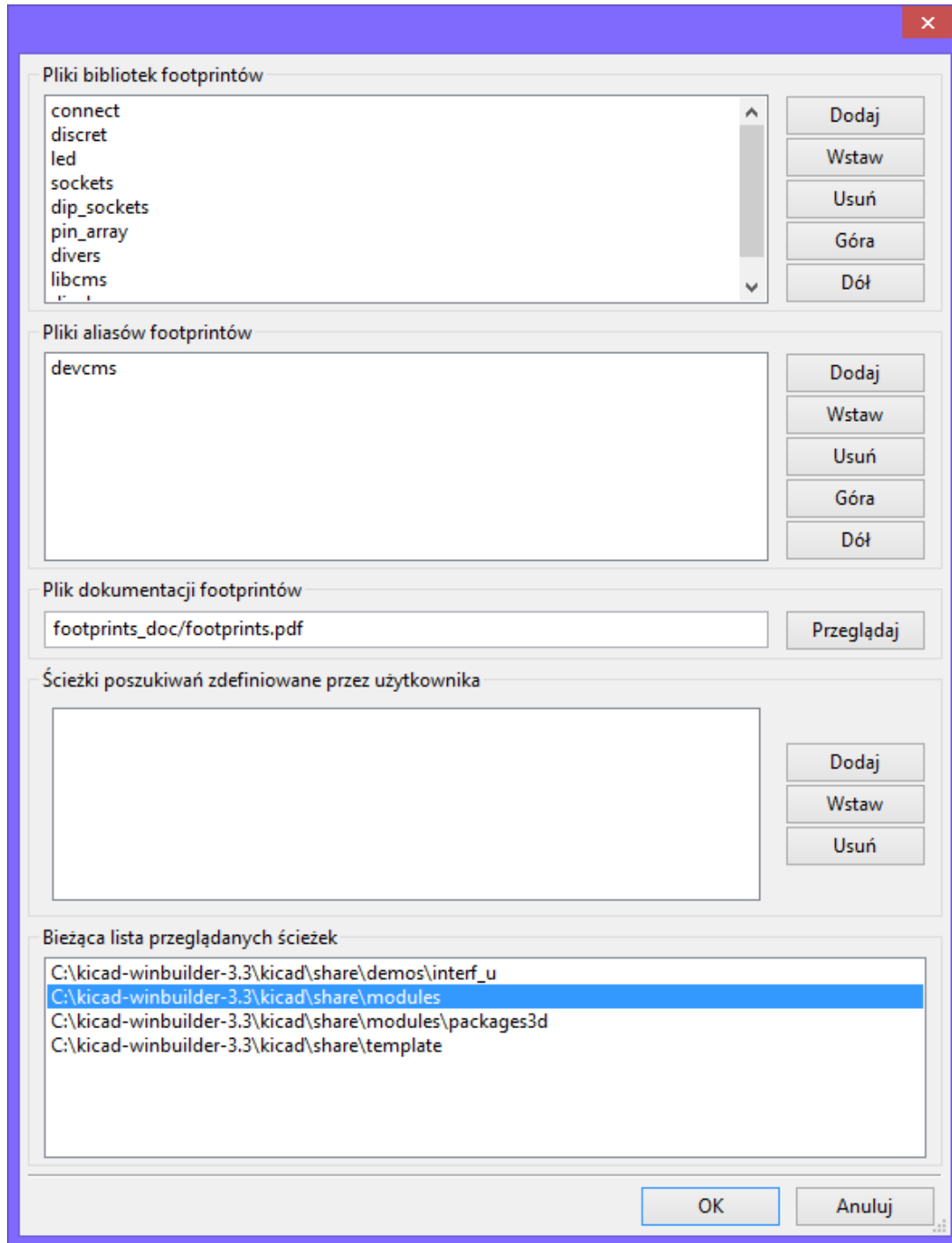
Znaczenie poszczególnych przycisków jest następujące:

	Transferuje przypisania footprintów do poszczególnych komponentów bezpośrednio do schematu.
	Uruchamia menu konfiguracji CvPcb .
	Wyświetla bieżący footprint (czyli ten który obecnie jest wskazany na liście dostępnych footprintów).
	Automatycznie przeskakuje do poprzedniego elementu, któremu jeszcze nie został przypisany żaden footprint.
	Automatycznie przeskakuje do następnego elementu, któremu jeszcze nie został przypisany żaden footprint.
	Automatycznie przypisuje nazwy footprintów korzystając z baz danych .equ o przypisaniach automatycznych. Użycie tego narzędzia domyślnie przyjmuje, że te pliki są dostępne.
	Kasuje wszystkie przypisania.
	Wyświetla dokumentację footprintu, jeśli istnieje.
	Przełącznik poszczególnych filtrów służących do filtrowania listy dostępnych footprintów na panelu footprintów . Możliwa jest filtracja według wzorców ustawionych w ustawieniach poszczególnych komponentów (pierwsza opcja), liczby wyprowadzeń (druga opcja) oraz według wybranej biblioteki (ostatnia opcja). Opcje te można łączyć. Gdy filtrowanie jest włączone, panel footprintów zawiera tylko elementy pasujące do sumy wybranych filtrów.

Konfigurowanie CvPcb - Wersja bez obsługi Tabeli Bibliotek

Główne okno

Uruchomienie menu konfiguracji powoduje otwarcie następującego okna dialogowego:



Wybór bibliotek footprintów



Po wybraniu elementu z listy za pomocą myszy:

- **Dodaj** - dodaje nową bibliotekę do listy na końcu listy.

- **Wstaw** - dodaje nową bibliotekę do listy przed aktualnie wybranym elementem.
- **Usuń** - usuwa wskazaną bibliotekę z listy.
- **Góra** - przesuwa wybraną bibliotekę wyżej na liście (zwiększa priorytet).
- **Dół** - przesuwa wybrany bibliotekę niżej na liście (zmniejsza priorytet).

Zapamiętaj:

Dowolna modyfikacja tej listy przenosi się do **Pcbnew**.

Wybór plików aliasów footprintów (skrypty przypisań)



Po wybraniu nazwy pliku z listy:

- **Dodaj** - dodaje nową nazwę do listy na końcu listy.
- **Wstaw** - dodaje nową nazwę do listy przed aktualnie wybranym elementem.
- **Usuń** - usuwa wskazany element z listy.
- **Góra** - przesuwa wybrany element wyżej na liście (zwiększa priorytet).
- **Dół** - przesuwa wybrany element niżej na liście (zmniejsza priorytet).

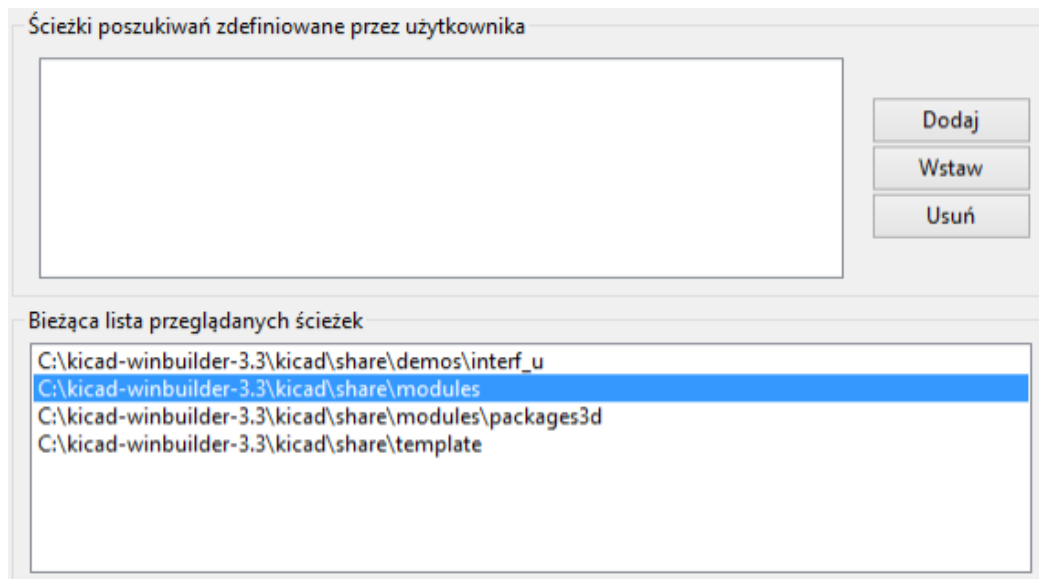
Wybór domyślnej ścieżki bibliotek footprintów

Domyślna ścieżka do bibliotek jest wyświetlana przez **CvPcb**. **CvPcb** używa tych ścieżek do odnajdywania bibliotek footprintów (pliki `.mod`) oraz skryptów przypisań (pliki `.equ`).

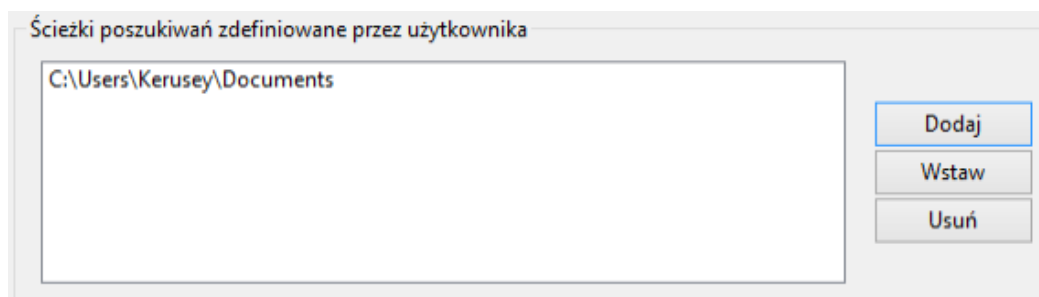
Ścieżki poszukiwań

CvPcb używa dwóch typów ścieżek:

- Ścieżki ustawiane automatycznie przez **CvPcb**.
- Ścieżki dodane przez użytkownika.



Ścieżki użytkownika



Ścieżki ustalone automatycznie przez CvPcb

Zależą one (częściowo) od systemu operacyjnego. Zawsze występuje na liście katalog roboczy. Następnie:

- `kicad/share/modules`.
- `kicad/share/modules/packages3d` (dla obiektów 3D w formacie VRML stworzonych przez Wings3D).
- `kicad/share/template`.

Ścieżka główna programu KiCad to:

- Ścieżka gdzie znaleziono pliki binarne programu **KiCad** (`.../kicad/bin`).

Jeśli nie znaleziono:

W systemie Windows:

- `c:\kicad`
- `d:\kicad`

W systemach Linux/Unix:

- `/usr/local/kicad`

- /usr/share/kicad

Tabele Bibliotek - Zarządzanie bibliotekami po nowemu

Począwszy od **wersji z dnia 2013-12-08 BZR4535-product**, CvPcb nie używa narzędzia do konfiguracji opisanego w poprzednim punkcie. Nowa implementacja tego narzędzia opiera się na **tabeli bibliotek footprintów**, która pozwala na bezpośrednie użycie następujących typów bibliotek:

- KiCad Legacy (Pliki .mod ze starszych wersji programu)
- KiCad Pretty z dysku lokalnego (foldery z rozszerzeniem .pretty, zawierające pliki .kicad_mod)
- KiCad Pretty umieszczone zdalnie na serwerach Github
- Biblioteki gEDA (foldery zawierające pliki .fp)
- Biblioteki programu EAGLE zapisane w formacie XML (wersje 6 lub późniejsze)



Pomimo obsługi kilku innych formatów, można zapisywać tylko biblioteki KiCad Pretty na dysku lokalnym (oraz pliki .kicad_mod wewnątrz tych folderów). Pozostałe formaty są przeznaczone **tylko do odczytu**.

Poniższy rysunek pokazuje okno dialogowe z wspomnianą tabelą. Aby go wywołać należy użyć polecenia **Tabela bibliotek**.

Tabela bibliotek według zasięgu

Tabela: C:\Users\Kerusey Karyu\AppData\Roaming\kicad\fp-lib-table

	Nazwa skrótowna	Ścieżka	Typ wtyczki	Opcje	Opis
1	Air_Coils_SML_NEOSID	\${KIGITHUB}/Air_Coils_SML_NEOSID.pretty	Github		HAMxx31A_HDMxx31A
2	Buttons_Switches_SMD	\${KIGITHUB}/buttons_switches_smd.pretty	Github		
3	Buzzers_Beepers	\${KIGITHUB}/Buzzers_Beepers.pretty	Github		The way you like them.
4	Capacitors_Elko_ThroughHole	\${KIGITHUB}/Capacitors_Elko_ThroughHole.pretty	Github		The way you like them.
5	Capacitors_SMD	\${KIGITHUB}/Capacitors_SMD.pretty	Github		The way you like them.
6	Capacitors_Tantalum_SMD	\${KIGITHUB}/Capacitors_Tantalum_SMD.pretty	Github		The way you like them.
7	Capacitors_ThroughHole	\${KIGITHUB}/Capacitors_ThroughHole.pretty	Github		The way you like them.
8	Choke_Axial_ThroughHole	\${KIGITHUB}/Choke_Axial_ThroughHole.pretty	Github		The way you like them.
9	Choke_Common-Mode_Wurth	\${KIGITHUB}/Choke_Common-Mode_Wurth.pretty	Github		The way you like them.
10	Choke_Radial_ThroughHole	\${KIGITHUB}/Choke_Radial_ThroughHole.pretty	Github		The way you like them.
11	Choke_SMD	\${KIGITHUB}/Choke_SMD.pretty	Github		The way you like them.

Biblioteki globalne | Biblioteki własne projektu

Dołącz bibliotekę | Dołącz za pomocą kreatora | Usuń bibliotekę | Przesuń w górę | Przesuń w dół | Edycja Opcji

Odpowiedniki ścieżek

	Zmienna środowiskowa	Ścieżka
1	KIGITHUB	https://github.com/KiCad
2	KIPRJMOD	C:\Users\Kerusey Karyu\Documents\MojeProjekty\LibraryMaking
3	KISYS3DMOD	C:\KiCad\share\modules\packages3d
4	KISYSMOD	C:\KiCad\share\modules\legacy_mod

OK | Anuluj

Tabela bibliotek footprintów jest używana do mapowania plików bibliotek obsługiwanych przez program do ich **nazw skrótych**. Nazwa skrótna jest używana do wyszukiwania footprintów

zamiast poprzedniej metody z wyszukiwaniem plików zgodnie z ustalonym układem ścieżek dostępu. Pozwala to programowi **CvPcb** na dostęp do footprintów za pomocą tej samej nazwy w różnych bibliotekach gwarantując tym samym, że właściwy footprint zostanie załadowany z odpowiedniej biblioteki. Pozwala to również na obsługę bibliotek pochodzących z innych programów (z pomocą wtyczek) EDA, takich jak np. Eagle czy gEDA.

Globalna tabela bibliotek footprintów

Globalna tabela bibliotek footprintów zawiera listę biblioteki, które są dostępne zawsze, niezależnie od obecnie wczytanego projektu. Tabela ta jest zapisana w pliku `fp-lib-table` w katalogu domowym użytkownika. Jego rzeczywista lokalizacja zależy użytego systemu operacyjnego.

Lokalna tabela bibliotek footprintów zależna od projektu

Lokalna tabela bibliotek footprintów zależna od projektu zawiera listę bibliotek, które są dostępne wyłącznie w obecnie wczytanym projekcie. Lokalna tabela może być modyfikowana tylko wtedy, gdy zostanie ona załadowana razem z listą sieci tego projektu. Gdy projekt nie został załadowany lub gdy taka lokalna tabela nie istnieje, tworzona jest pusta tabela, którą będzie można wypełnić i później zapisać razem z plikiem przypisać footprintów (z rozszerzeniem `.cmp`).

Konfiguracja początkowa

Gdy **CvPcb** lub **Pcbnew** zostanie uruchomiony i globalna tabela bibliotek `fp-lib-table` nie zostanie znaleziona w katalogu domowym użytkownika, **CvPcb** będzie próbował skopiować domyślną tabelę bibliotek `fp_global_table` zapisaną w folderze `template` do pliku `fp-lib-table` w katalogu domowym użytkownika. Jeśli plik `fp_global_table` nie został znaleziony, to zamiast operacji kopiowania zostanie utworzona pusta tabela. Gdyby taka sytuacja miała miejsce użytkownik ma też możliwość skopiowania `fp_global_table` samodzielnie lub "ręczne" skonfigurowania tabeli. Domyślna tabela bibliotek zawiera wszystkie standardowe biblioteki jakie zostały zainstalowane razem z programem **KiCad EDA Suite**.

Dodawanie nowych wpisów w tabeli

By móc używać biblioteki najpierw należy dodać globalną lub lokalną tabelę. Lokalna tabela ma zastosowanie tylko gdy istnieje otwarta lista sieci projektu.

Każda pozycja tabeli musi posiadać **unikalną nazwę skrótową**. Nie musi ona mieć jakiegokolwiek związku z bieżącą nazwą pliku lub ścieżki do niego. Znak dwukropka `:` nie może być używany w nazwach skrótowych. Każda pozycja musi również odnosić się do prawidłowej ścieżki/nazwy pliku w zależności od typu biblioteki. Ścieżki do plików mogą być bezpośrednie, względne lub pochodzić ze specjalnych zmiennych systemowych - opisanych dalej.

Aby biblioteka została wczytana przez **CvPcb** musi być także wybrana właściwa **wtyczka** obsługująca dany format pliku. **CvPcb** obecnie wspiera następujące formaty plików bibliotek: **KiCad Legacy**, **KiCad Pretty**, **Eagle** oraz **gEDA**.

Istnieje również pole przeznaczone do wpisania opisu dla danego wpisu w tabeli. Pole z opcjami nie jest w tej chwili używane, zatem umieszczanie jakichkolwiek opcji nie ma znaczenia przy ładowaniu bibliotek.

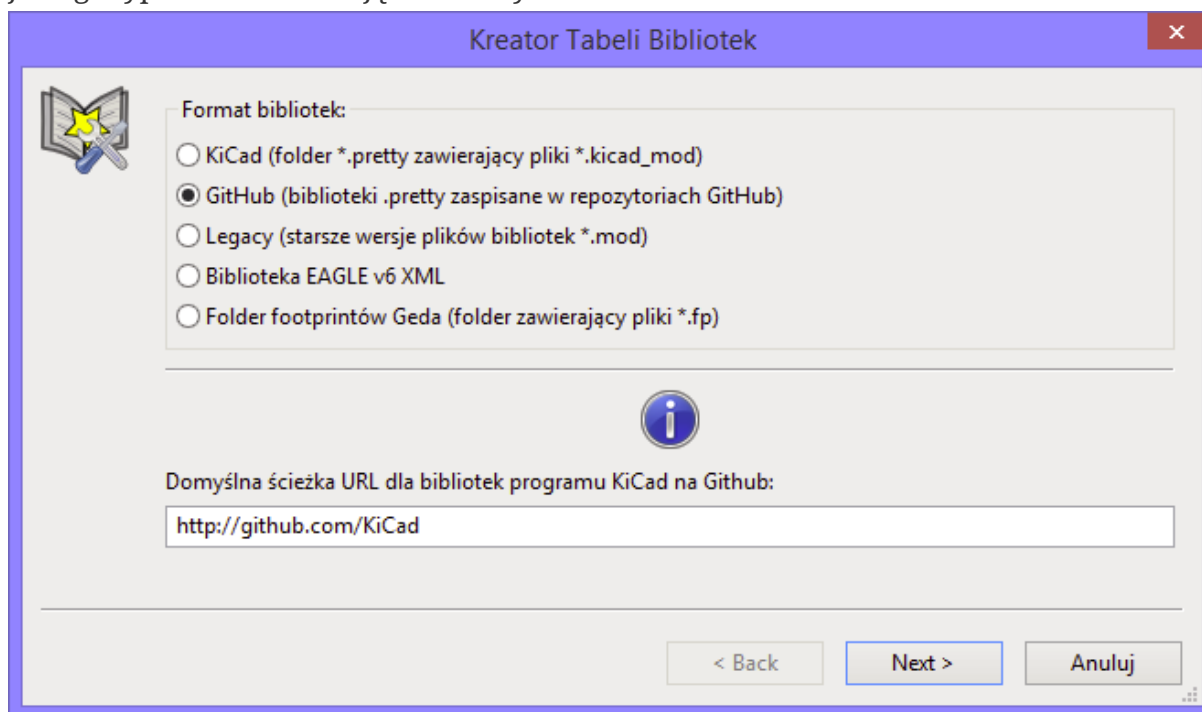
Proszę zauważyć, że nie można umieścić dwóch takich samych nazw skrótowych w jednej tabeli. Jednakże, można wpisać tę samą nazwę skrótową w globalnej i lokalnej tabeli bibliotek, ponieważ tabela lokalna ma większy priorytet niż tabela globalna w takim przypadku.

Gdy wpisy zostaną zdefiniowane w lokalnej tabeli bibliotek, to plik `fp-lib-table` zawierający te wpisy zostanie umieszczony w folderze skąd pochodzi lista sieci.

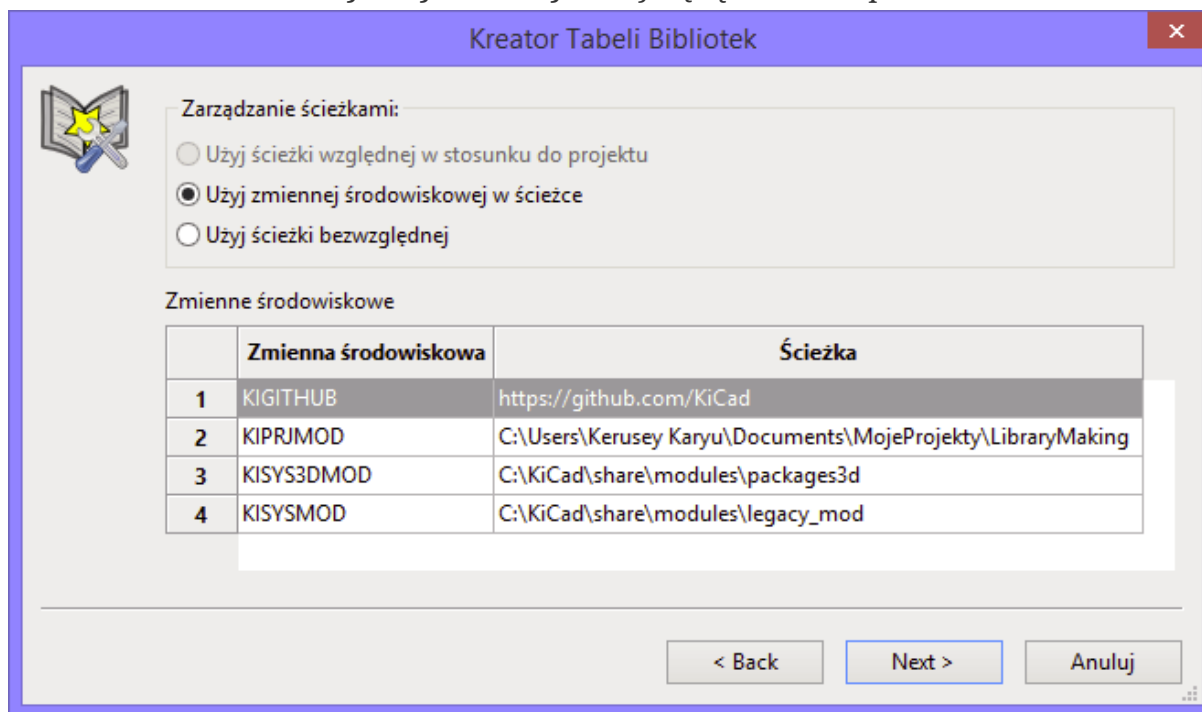
Dodawanie nowych wpisów w tabeli z pomocą kreatora

Aby ułatwić proces dodawania poszczególnych bibliotek do tabeli bibliotek przewidziano możliwość skorzystania z prostego kreatora uruchamianego za pomocą przycisku **Dołącz za pomocą kreatora**. Przykład jego zastosowania zostanie pokazany z wykorzystaniem bibliotek zdalnych przechowywanych w repozytoriach GitHub.

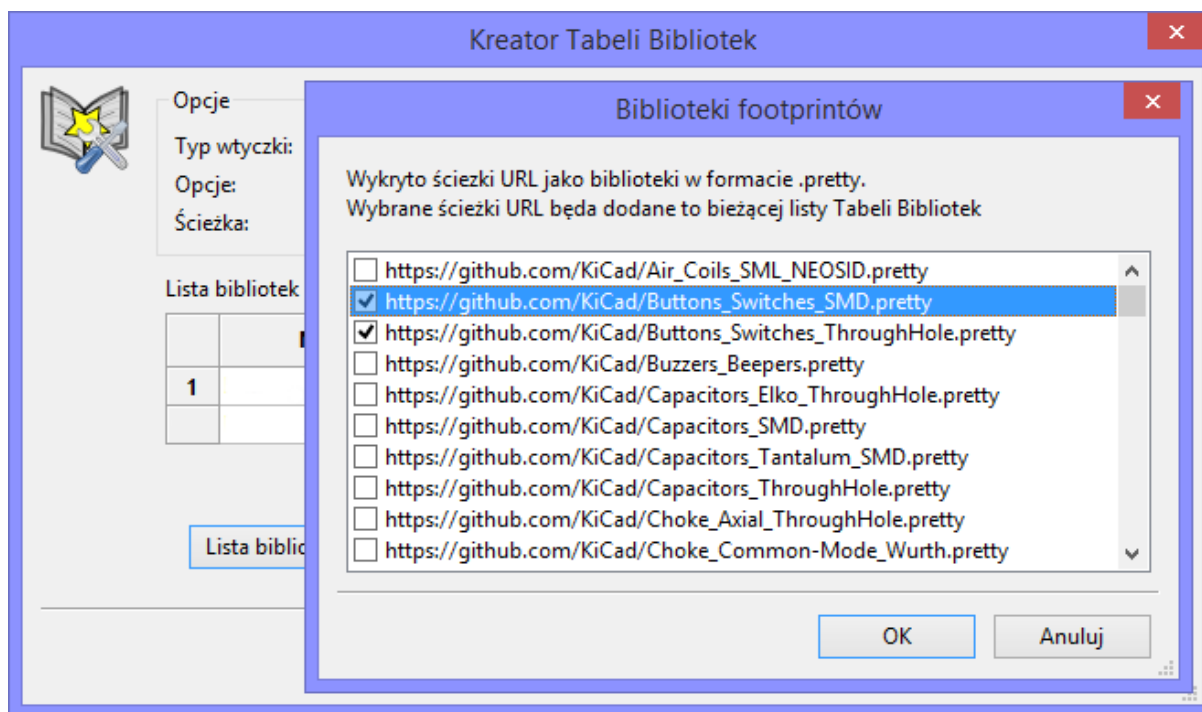
Po uruchomieniu kreatora pojawi się pierwsze w kolejności okno dialogowe, w którym należy określić jakiego typu biblioteki mają zostać wyszukane i dodane.



W następnym kroku należy zdecydować się czy ścieżki dostępu do bibliotek będą zawierać **odnośników do zmiennych systemowych** czy będą to ścieżki pełne.

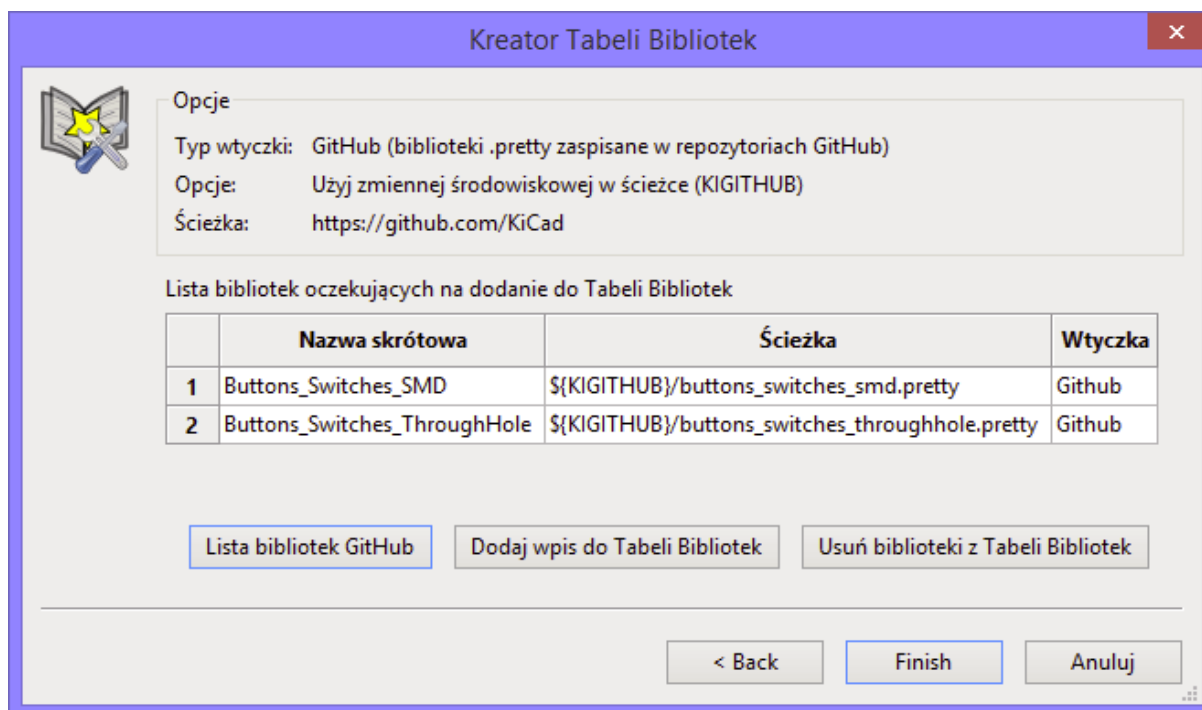


Gdy te dwie rzeczy zostaną już wstępnie skonfigurowane będzie można przystąpić do wyszukania dostępnych bibliotek. W tym celu należy utworzyć listę wyboru za pomocą przycisku **Lista bibliotek GitHub**. Po chwili powinno się pojawić okno z możliwością zaznaczenia wyszukanych bibliotek.



Z listy powinno się zaznaczyć tylko te biblioteki, których nie ma w tabeli bibliotek lub te które chcielibyśmy z niej usunąć.

Po zatwierdzeniu listy w poprzednim oknie, w ostatnim kroku można przystąpić do połączenia tabeli tymczasowej z bieżącą tabelą bibliotek.



Pobieranie wartości ze zmiennych systemowych

Jednym z największych zalet tabeli bibliotek footprintów jest możliwość używania **odnośników do zmiennych systemowych**. Pozwala to na zdefiniowanie własnych ścieżek do bibliotek w zmiennych systemowych i używanie ich w projektach.

Odnośniki do zmiennych systemowych można wpłacać w treść pól zawierających ścieżkę do pliku używając powszechnie znanego formatu **`${nazwa_zmiennej}`**. Domyślnie **CvPcb** definiuje zmienną środowiskową **KISYSMOD**. Wskazuje ona na miejsce, gdzie zainstalowane zostały biblioteki instalowane razem z programem **KiCad EDA Suite**. Można ją re-definiować samodzielnie, co pozwala na zastąpienie standardowych bibliotek ich własnymi odpowiednikami. Gdy wczytana zostanie lista sieci, **CvPcb** automatycznie definiuje również zmienną **KIPRJMOD**. Pozwala to na tworzenie bibliotek w miejscu wskazywanym przez projekt bez konieczności definiowania bezwzględnej ścieżki do biblioteki w lokalnej tabeli footprintów projektu.

Używanie wtyczki GitHub

GitHub to specjalna wtyczka pozwalająca na łączenie się ze zdalnym repozytorium GitHub zawierającym footprinty w formacie `.pretty` (nowa wersja formatu zapisu footprintów przez program **KiCad**). Repozytorium to jest tylko do odczytu, ale wtyczka umożliwia również dostęp do technologii "Copy On Write" (COW) wspierającej możliwość edycji footprintów odczytanych z repozytorium GitHub i zapisanie ich nowych wersji na dysku lokalnym, które później można wysłać z w celu ich aktualizacji. Sama wtyczka nie umożliwia zapisu do repozytorium.

By dodać wpis GitHub do tabeli bibliotek, pole *Ścieżka* musi zostać wypełniona ważnym adresem URL do repozytorium GitHub.

Przykładowo:

https://github.com/liftoff-sr/pretty_footprints , https://github.com/liftoff-sr/pretty_footprints

Zwykle poprawna ścieżka URL jest tworzona wg następującego schematu:

https://github.com/nazwa_uzytkownika/nazwa_repozytorium

Pole *Typ Wtyczki* musi być ustawione jako `Github`. Aby włączyć funkcję "Copy On Write" należy w polu *Opcje* dodać parametr `allow_pretty_writing_to_this_dir` który zawierał będzie ścieżkę na dysku lokalnym gdzie zapisywane będą pliki z modyfikacjami. Jeśli ta opcja zostanie pominięta to biblioteka GitHub jest tylko do odczytu.

Footprinty tam zapisane są połączeniem części tylko do odczytu repozytorium GitHub i treści lokalnych zmian by utworzyć zmodyfikowaną bibliotekę footprintów. Każda modyfikacja biblioteki GitHub będzie trafiać do tej lokalnej biblioteki hybrydowej COW umieszczonej w odpowiednim folderze `*.pretty`. Należy w tym miejscu nadmienić, iż część rezydentna COW pochodząca z repozytorium GitHub jest zawsze tylko do odczytu, co oznacza, że nie można niczego samodzielnie usunąć lub zmodyfikować bezpośrednio w samym repozytorium GitHub. Niezależnie czy biblioteka będzie hybrydowa, czyli połączona z lokalnej części tylko do odczytu i zapisu, czy tylko część zdalną przeznaczoną tylko do odczytu, będzie ona dalej zwana biblioteką "Github" w dalszych rozważaniach.

Przykładowe wpisy w tabeli

Poniższa tabela pokazuje wpis z tabeli bibliotek, której nie została przypisana opcja `allow_pretty_writing_to_this_dir`:

Nazwa skrókowa	Ścieżka	Typ wtyczki	Opcje	Opis
github	<code>https://github.com/liftoff-sr/pretty_footprints</code>	Github		Liftoff's GH footprints

Następna tabela pokazuje wpis z tabeli bibliotek z opcją dotyczącą COW. Zmienna `#{HOME}` jest tylko przykładowa. Folder `github.pretty` jest umieszczony w folderze do którego prowadzi ścieżka `#{HOME}/pretty/`. W każdym przypadku użycia opcji `allow_pretty_writing_to_this_dir`, wymagane jest samodzielne utworzenie tego folderu i musi on posiadać rozszerzenie `.pretty`.

Nazwa skrókowa	Ścieżka	Typ wtyczki	Opcje	Opis
github	<code>https://github.com/liftoff-sr/pretty_footprints</code>	Github	<code>allow_pretty_writing_to_this_dir= #{HOME}/pretty/github.pretty</code>	Liftoff's GH footprints

Footprinty pobierane z repozytorium mają zawsze pierwszeństwo przed tymi umieszczonymi w folderze na który wskazuje opcja `allow_pretty_writing_to_this_dir`. Po zapisaniu footprintu do lokalnego folderu przechowującego hybrydowe pliki COW, np. poprzez zapisanie zmian w edytorze footprintów, żadne aktualizacje GitHub nie będą widoczne podczas ładowania footprintów o tej samej nazwie, niż te, które zostały zapisane lokalnie.

Zawsze należy korzystać z odrębnego folderu `*.pretty` dla poszczególnych bibliotek GitHub i nigdy nie powinno się łączyć folderów przez przypisywanie tego samego folderu do innych bibliotek GitHub, gdyż mogłoby to doprowadzić do bałaganu nad którym nie byłoby można zapanować.

Wartości symboliczne w zmiennych systemowych zapisane w notacji `${nazwa_zmiennej}` przypisane do opcji `allow_pretty_writing_to_this_dir` będą rozwijane automatycznie by utworzyć właściwą ścieżkę, tak samo jak to ma miejsce w polu *Ścieżka*.

Co robić z plikami w COW?

System COW to element przyspieszający współużytkowanie footprintów. Jeśli zawartość COW będzie regularnie przesyłana do zarządcy repozytorium GitHub, będzie można pomóc w uaktualnianiu kopii znajdujących się w repozytorium zdalnym.

Całość jest bardzo prosta. Za pomocą poczty elektronicznej należy wysłać pliki `*.kicad_mod` znajdujące się w folderach systemu COW do osoby zarządzającej repozytorium. Po otrzymaniu potwierdzenia, że zmiany zostały zaakceptowane i wprowadzone, można skasować wysłane pliki z COW. Nowe wersje plików zostaną pobrane z repozytorium GitHub.

Głównym celem jest utrzymywanie jak najmniejszego zestawu plików systemu COW jak tylko jest to możliwe poprzez regularne przesyłanie zawartych w niej plików.

Generalne zalecenia przy używaniu tabeli bibliotek

Biblioteki footprintów mogą być zdefiniowane globalne lub lokalnie dla obecnie wczytanego projektu. Biblioteki umieszczone w globalnej tabeli bibliotek użytkownika są zawsze dostępne i są zapisane w pliku `fp-lib-table` w katalogu domowym użytkownika. Globalne biblioteki będą dostępne nawet jeśli nie została otwarta lista sieci danego projektu. Inaczej sprawa się ma w przypadku lokalnych bibliotek, które są aktywne wyłącznie dla bieżącej listy sieci. Lokalna tabela bibliotek jest zapisywana w pliku `fp-lib-table` umieszczonym w tej samej ścieżce co lista sieci.

Nie ma przeszkód co do definiowania odnośników do bibliotek w obu tabelach. Dlatego też nie zostało odgórnie określone w jaki sposób użytkownik będzie wykorzystywał możliwości jakie dają globalne i lokalne tabele. Są jednak zalety i wady każdego z rozwiązań, które należy rozważyć.

Można zdefiniować wszystkie biblioteki w globalnej tabeli bibliotek, co oznacza, że będą one zawsze dostępne gdy będą potrzebne. Wadą takiego rozwiązania będzie szybkość wyszukiwania w nich odpowiedniego footprintu.

Można zdefiniować wszystkie biblioteki w lokalnej tabeli bibliotek. Zaletą takiego rozwiązania będzie możliwość zdefiniowania tylko tych bibliotek, które będą w danej chwili potrzebne oraz skrócenie czasu ich przeszukiwania. Wadą tego rozwiązania będzie zaś to, że będzie trzeba zawsze pamiętać, by dodać odpowiednie biblioteki dla każdego nowego projektu.

Można zdefiniować biblioteki w obu tabelach jednocześnie. Sensowne staje się wtedy wpisanie bibliotek, które są wykorzystywane prawie we wszystkich projektach do tabeli globalnej, a w lokalnych tabelach umieszczać tylko te, które są przydatne tylko w tym konkretnym projekcie. Będzie to rozwiązanie, które będzie posiadało największą elastyczność kosztem zmniejszenia szybkości wyszukiwania.

Podgląd bieżącego footprintu

Polecenie **Podgląd footprintu** pozwala na wyświetlenie bieżącego footprintu, czyli tego który aktualnie jest wskazany na liście w panelu footprintów.

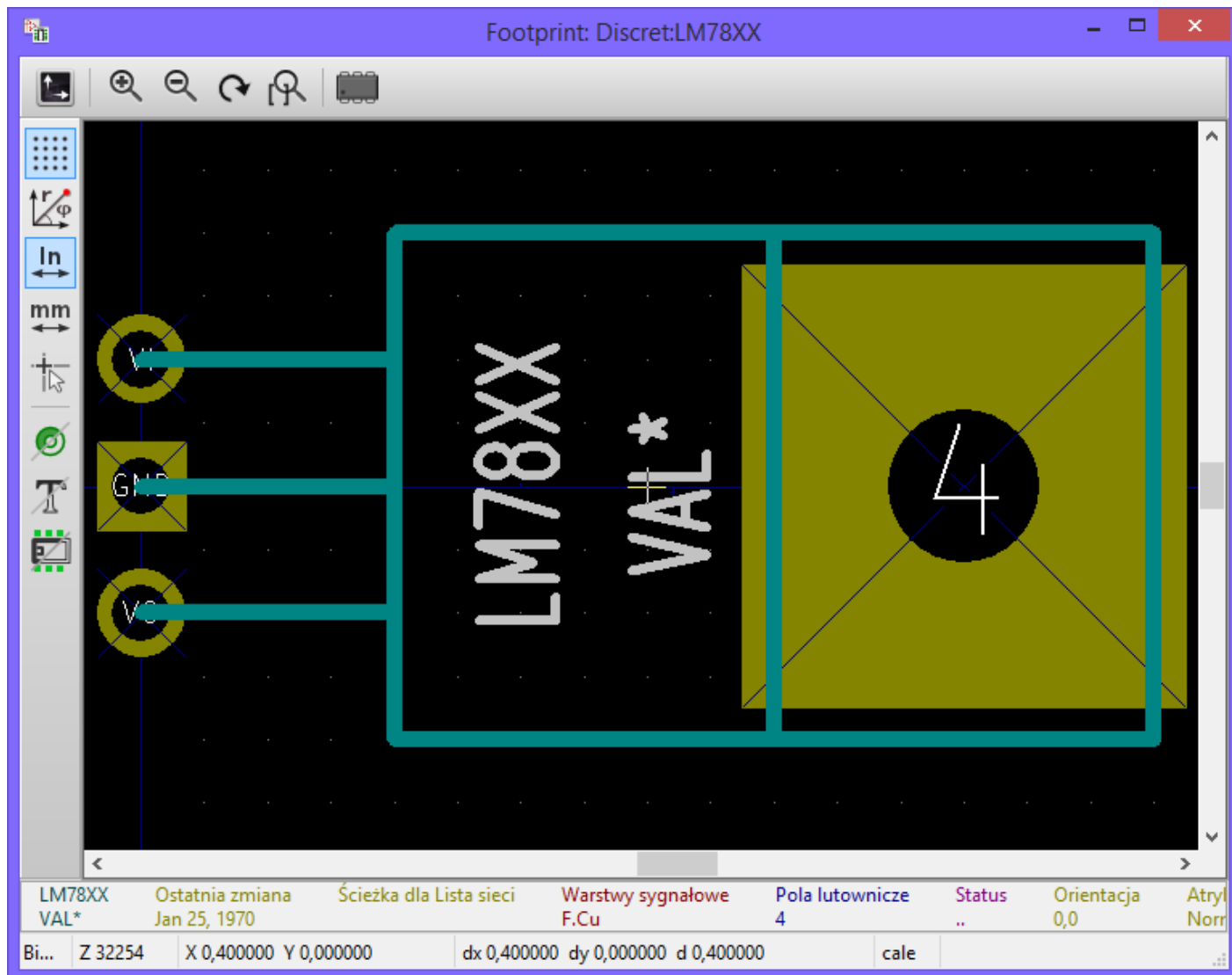
Można przeglądać w ten sposób listę footprintów klikając na ich nazwy przy pozostawionym oknie

podglądu footprintów.

Można również podglądać widok 3D (jeśli moduły mają przypisane kształty 3D).

Wyświetlanie

Pozycja kursora jest wyświetlana na dolnym pasku ekranu:

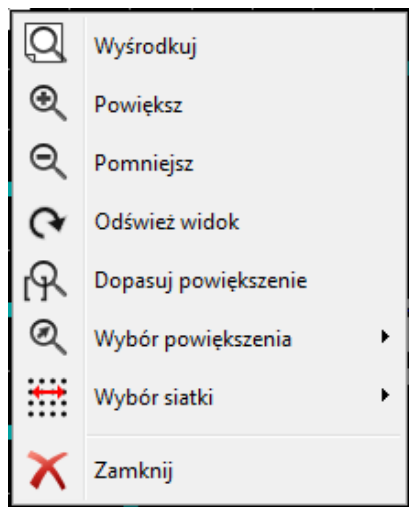


Pozycja absolutna (X nnnn Y nnnn) oraz relatywna (dx nnnn dy nnnn). Punkt odniesienia dla pozycji relatywnej można zmieniać klawiszem spacji.

Skróty klawiaturowe

F1	Przybliżenie
F2	Oddalenie
F3	Odświeża obraz
<spacja>	Ustawia punkt odniesienia

Menu kontekstowe



Wyświetlane jest klikając prawy klawisz myszy. Pierwsze cztery polecenia służą do bezpośredniego dostosowywania powiększenia. Dwa kolejne wyświetlają dodatkowe podmenu:

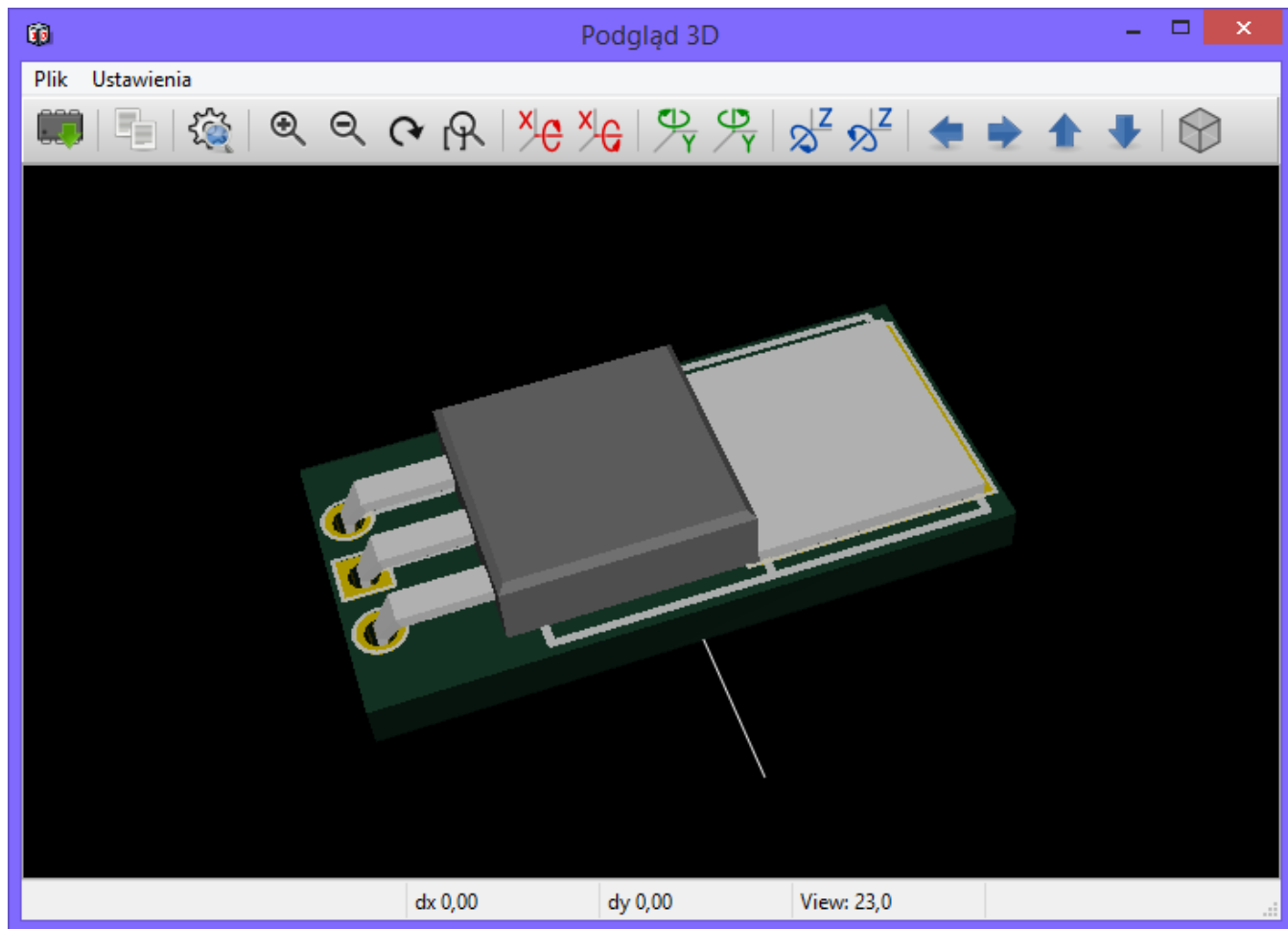
Wybór powiększenia	Ustawienie wybranego współczynnika powiększenia z listy dostępnych.
Wybór siatki	Wybór ustawienia skoku siatki z listy dostępnych.

Pasek narzędzi



	Opcje wyświetlania
	Opcje powiększania
	Wyświetlanie kształtów 3D

Podgląd 3D



5. Przypisywanie footprintów do komponentów

Podstawy

Aby przypisać wybranemu **komponentowi** (podświetlony element) z **panelu komponentów** określony footprint, należy w **panelu footprintów** dwukrotnie kliknąć nazwę wybranego footprintu.

Następny komponent z listy komponentów jest podświetlany:

- Automatycznie po poprzednim przypisaniu.
- Ręcznie z pomocą myszy lub klawiszy kursora.


Operacja przypisywania

Wystarczy kliknąć dwukrotnie lewym klawiszem myszy na wybranym footprintie.

Zmiana bieżącego przypisania

Jest wykonywana na zasadzie nowego przypisania, jak wyżej.

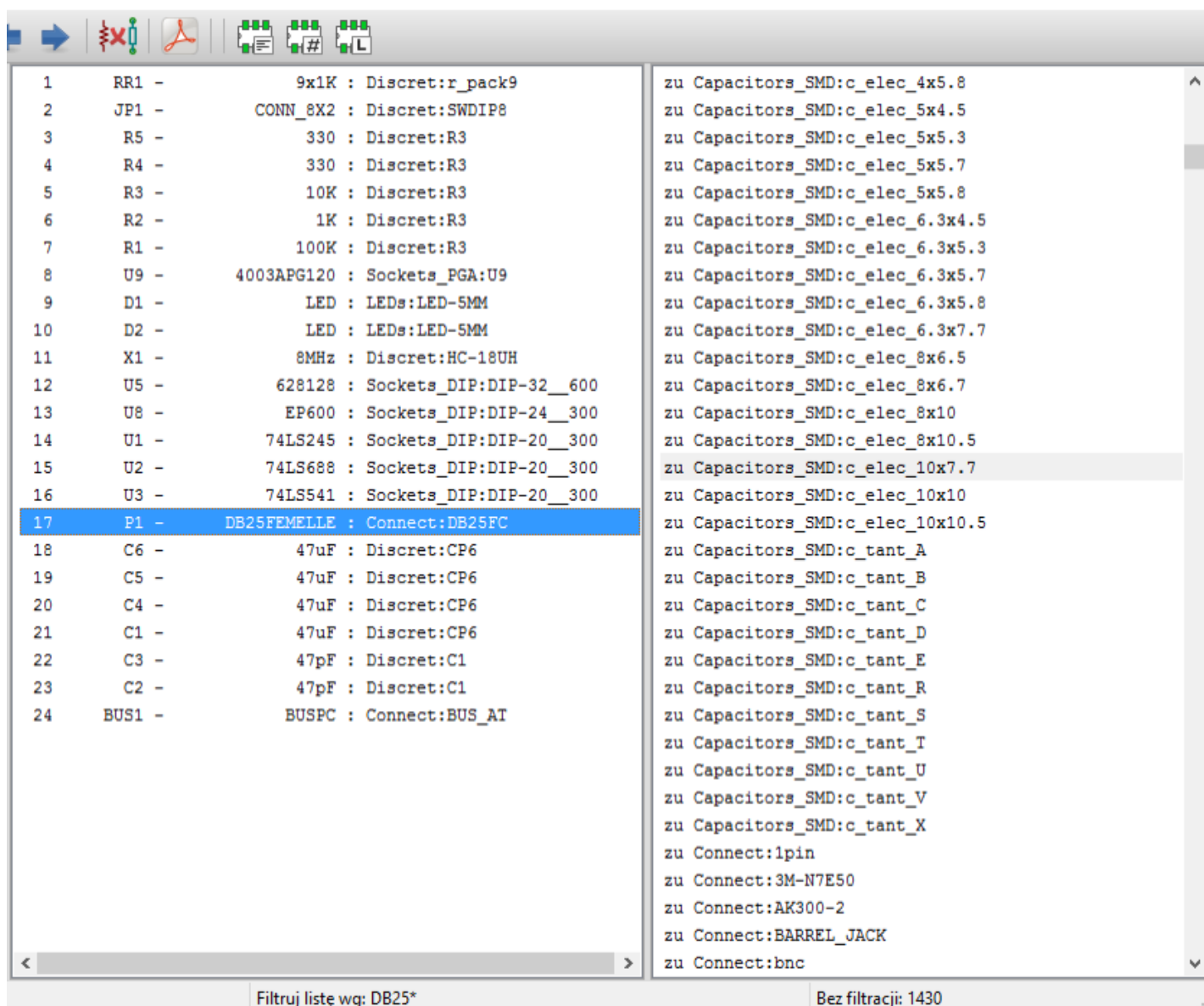
Filtrowanie listy footprintów

Polecenia z paska narzędzi  pozwalają na włączenie/wyłączenie poszczególnych sposobów filtracji. Poszczególne filtry można włączać niezależnie by bardziej zawęzić wynik filtracji.

Możliwa jest filtracja **według wzorców** ustawionych w ustawieniach poszczególnych komponentów

(pierwsza ikona), **liczby wyprowadzeń** (druga ikona) oraz według **wybranej biblioteki** (ostatnia ikona).

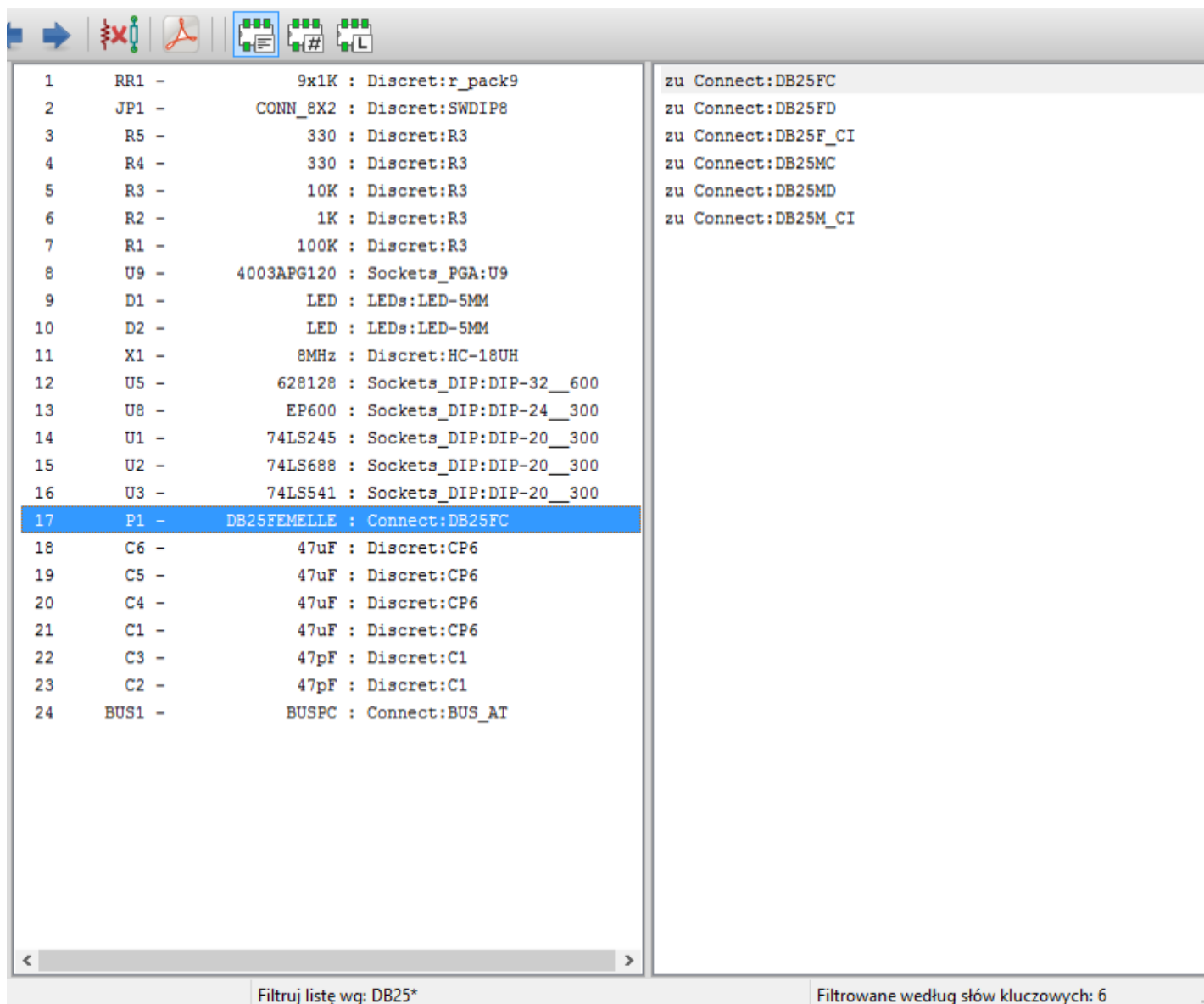
Jeśli nie ma ustalonej filtracji wyświetlana jest pełna lista footprintów. Poniższy rysunek ukazuje przykład listy **bez filtracji**:



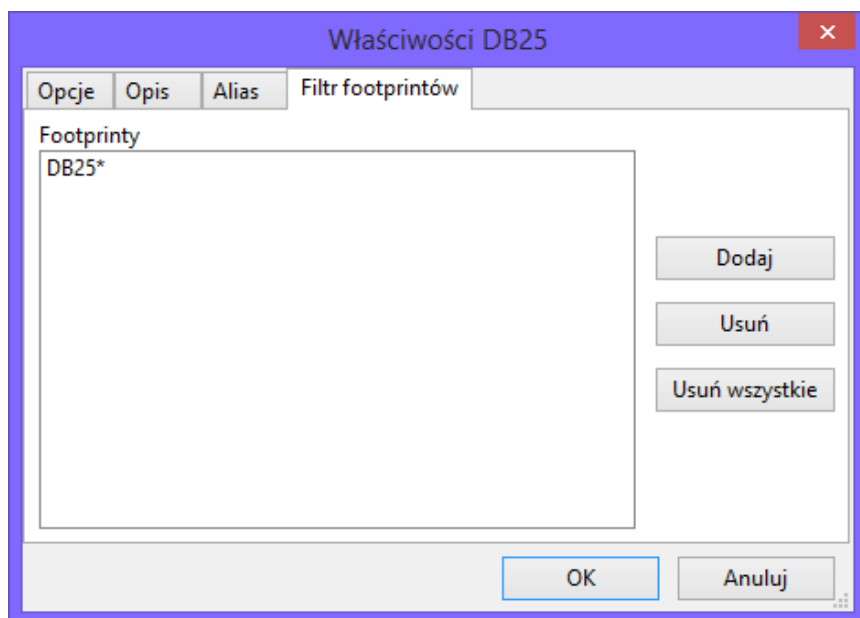
Filtrowanie za pomocą wzorców z ustawień komponentów

Jeśli wybrany komponent posiada zdefiniowane wzorce nazw dozwolonych footprintów, lista footprintów w **CvPcb** może być według niej filtrowana.

Po włączeniu **filtracji według pasujących nazw** lista footprintów ulegnie znacznemu skróceniu. W tym konkretnym przypadku wyświetlonych zostanie tylko 8 footprintów:

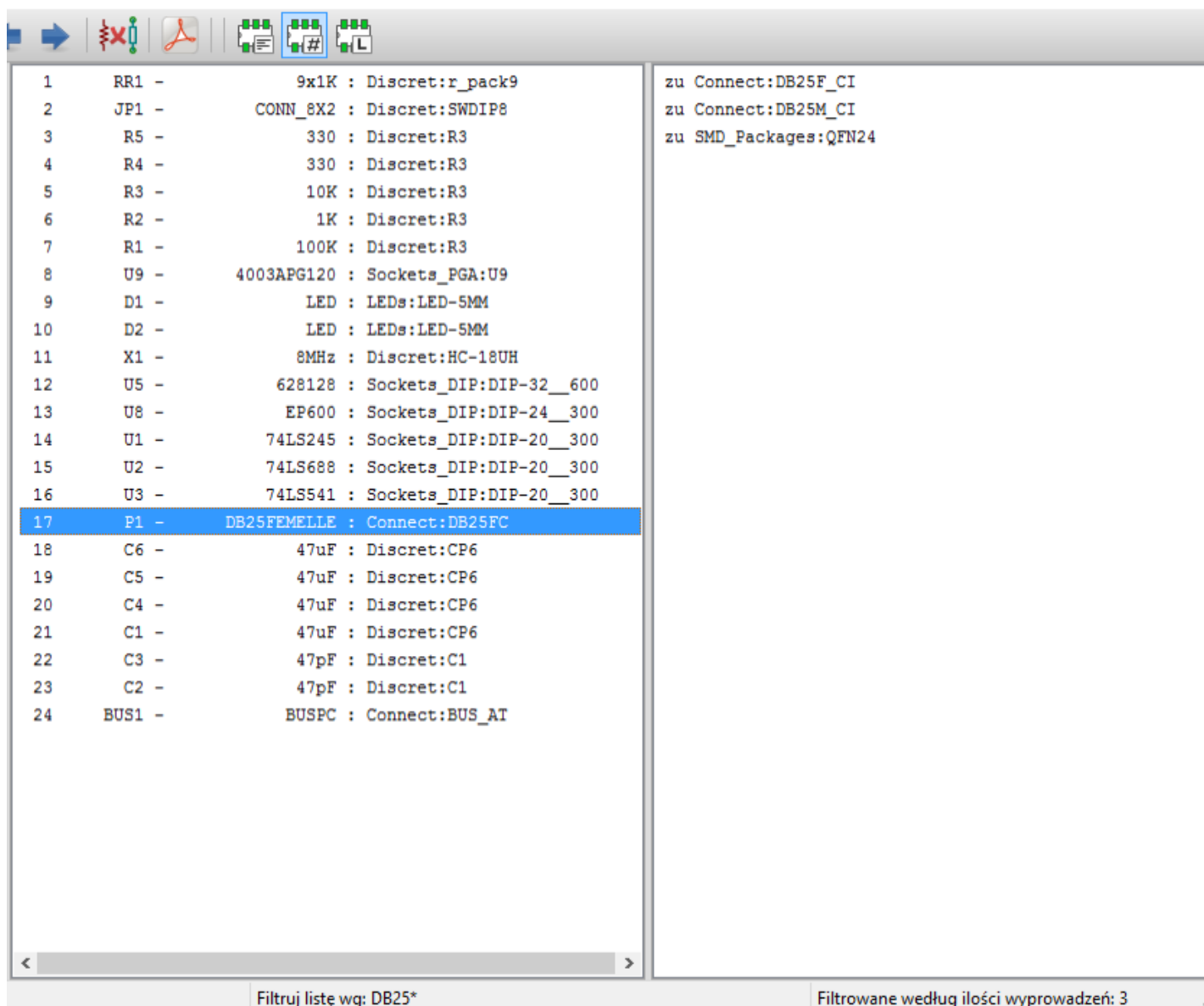


Dzieje się tak, gdyż w **Eeschema** podczas tworzeniu symbolu zostały uzupełnione informacje o dozwolonych footprintach dla tego komponentu w zakładce **Filtr footprintów**:



Filtrowanie za pomocą pasującej liczby wyprowadzeń

W przypadku włączenia **filtracji według liczby wyprowadzeń**, widok prawego panelu zmieni się i zostaną wyświetlone tylko te footprinty, które posiadają odpowiednią liczbę wyprowadzeń (tu: 9).

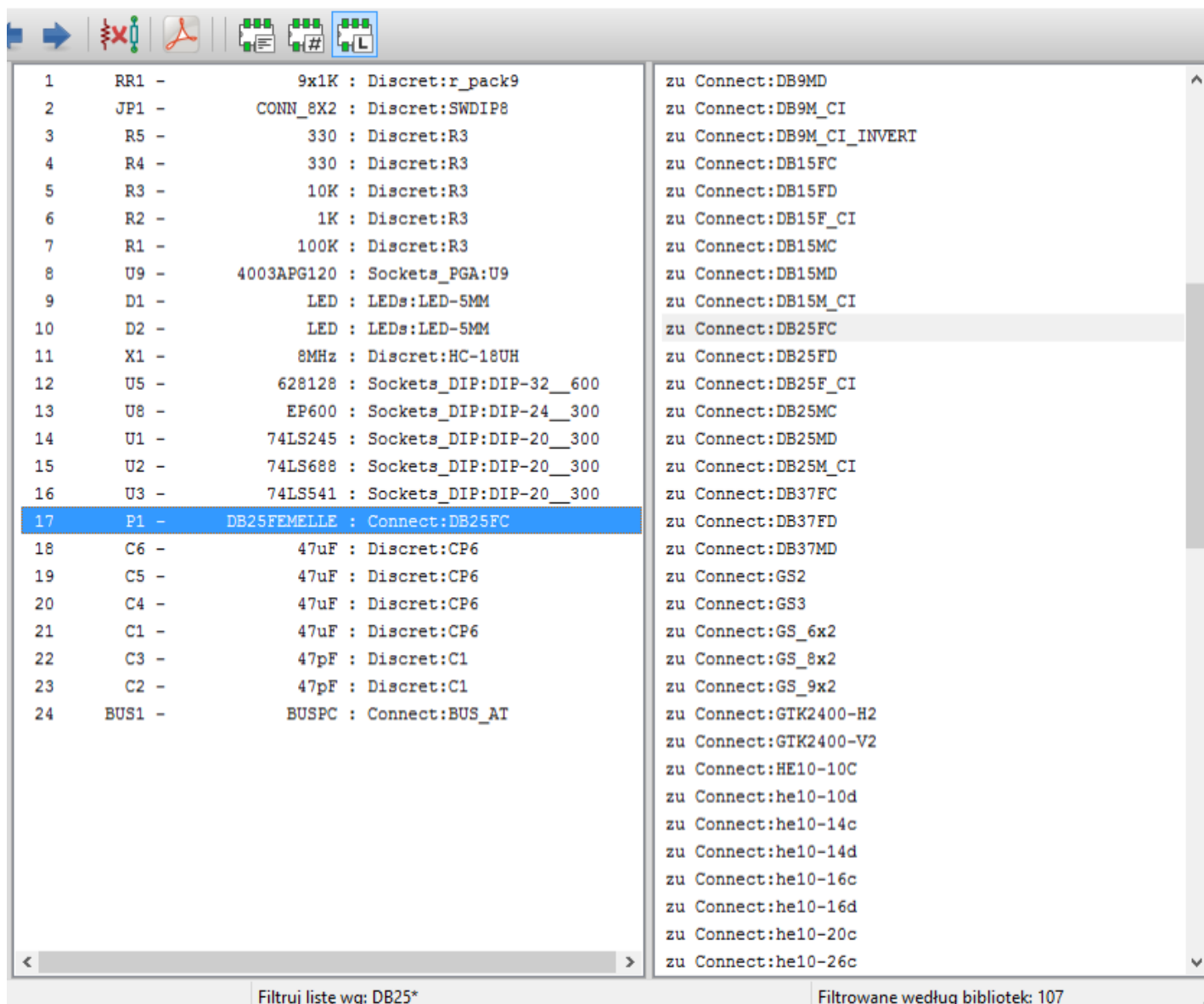


1	RR1 -	9x1K	: Discret:r_pack9	zu Connect:DB25F_CI
2	JP1 -	CONN_8X2	: Discret:SWDIP8	zu Connect:DB25M_CI
3	R5 -	330	: Discret:R3	zu SMD_Packages:QFN24
4	R4 -	330	: Discret:R3	
5	R3 -	10K	: Discret:R3	
6	R2 -	1K	: Discret:R3	
7	R1 -	100K	: Discret:R3	
8	U9 -	4003APG120	: Sockets_PGA:U9	
9	D1 -	LED	: LEDs:LED-5MM	
10	D2 -	LED	: LEDs:LED-5MM	
11	X1 -	8MHz	: Discret:HC-18UH	
12	U5 -	628128	: Sockets_DIP:DIP-32_600	
13	U8 -	EP600	: Sockets_DIP:DIP-24_300	
14	U1 -	74LS245	: Sockets_DIP:DIP-20_300	
15	U2 -	74LS688	: Sockets_DIP:DIP-20_300	
16	U3 -	74LS541	: Sockets_DIP:DIP-20_300	
17	P1 -	DB25FEMELLE	: Connect:DB25FC	
18	C6 -	47uF	: Discret:CP6	
19	C5 -	47uF	: Discret:CP6	
20	C4 -	47uF	: Discret:CP6	
21	C1 -	47uF	: Discret:CP6	
22	C3 -	47pF	: Discret:C1	
23	C2 -	47pF	: Discret:C1	
24	BUS1 -	BUSPC	: Connect:BUS_AT	

Filtruj listę wg: DB25* Filtrowane według ilości wyprowadzeń: 3

Filtrowanie za pomocą wybranej biblioteki

W przypadku włączenia **filtracji według wybranej biblioteki**, widok prawego panelu będzie podobny i zostaną wyświetlone tylko te footprinty, które pochodzą z wybranej w **panelu bibliotek** biblioteki.



6. Automatyczne przypisywanie footprintów

Pliki skryptów przypisać

Pliki te pozwalają na proces automatycznego przypisywania footprintów.

Pobierane są z nich nazwy footprintów na podstawie wartości (pole *Value*) komponentów.

Pliki te mają standardowe rozszerzenie `.equ`

Po wybraniu odpowiedniego pliku w zależności od przeznaczenia projektu, można w prosty sposób dostosowywać technologię jego wykonania, np. wersja SMD, wersja THT...

Zobacz też Wybór plików skryptów przypisać.

Format plików

Pliki `.equ` zawierają zwykły tekst, gdzie każda linia odpowiada jednemu komponentowi.

Każda z linii posiada strukturę:

```
'component value' 'footprint name'
```


Każda nazwa musi być zamknięta w apostrofach, obie nazwy muszą być rozdzielone przynajmniej jednym znakiem spacji.

Przykładowo:

Jeśli element **U3** to układ scalony **14011** a jego odpowiedni footprint to **14DIP300**, to linia powinna wyglądać tak:

```
'14011' '14DIP300'
```

TEXT

Linie rozpoczynające się od znaku *hash* (#) są traktowane jako komentarz.

Przykładowa zawartość pliku:

```
#integrated circuits (smd):  
'74LV14' 'S014E'  
'74HCT541M' 'S020L'  
'EL7242C' 'S08E'  
'DS1302N' 'S08E'  
'XRC3064' 'VQFP44'  
'LM324N' 'S014E'  
'LT3430' 'SSOP17'  
'LM358' 'S08E'  
'LTC1878' 'MSOP8'  
'24LC512I/SM' 'S08E'  
'LM2903M' 'S08E'  
'LT1129_S08' 'S08E'  
'LT1129CS8-3.3' 'S08E'  
'LT1129CS8' 'S08E'  
'LM358M' 'S08E'  
'TL7702BID' 'S08E'  
'TL7702BCD' 'S08E'  
'U2270B' 'S016E'  
#Xilinx  
'XC3S400PQ208' 'PQFP208'  
'XCR3128-VQ100' 'VQFP100'  
'XCF08P' 'BGA48'  
  
#upro  
'MCF5213-LQFP100' 'VQFP100'  
  
#regulators  
'LP2985LV' 'SOT23-5'
```

TEXT

Przypisywanie automatyczne

Proces automatycznego przypisywania jest uruchamiany przez wybranie ikony:



Wszystkie komponenty jakie zostały znalezione (na podstawie ich wartości) w pliku `.equ` będą miały przypisany automatycznie wskazany tam footprint.

7. Pliki numeracji wstecznej

Pliki te mogą być użyte do przeprowadzenia procesu numeracji wstecznej na schemacie. Są one generowane przez **Pcbnew** podczas zamiany footprintów.

Zawierają zwykły tekst, gdzie każda linia odpowiada jednemu komponentowi, podając nazwę komponentu (identyfikator) na schemacie i przypisany jemu footprint.

Przykładowo:

Jeśli komponentowi oznaczonemu jako **U3** został zmieniony footprint na **14DIP300**, to wygenerowana linia będzie zawierać tekst:

```
comp "U3" = footprint "14DIP300"
```

TEXT

Utworzony plik będzie się nazywał tak samo jak plik z obwodem drukowanym, ale z rozszerzeniem `.cmp`, i zostanie umieszczony w folderze projektu.