

# KICAD

# CVPCB



# LINUX & WINDOWS

Autorzy: Jean-Pierre Charras  
Wayne Stambaugh

Wersja: Grudzień 2013

# Spis treści

<b><u>1. Przeznaczenie</u></b>	<b>strona 2</b>
<b><u>2. Ogólna charakterystyka aplikacji</u></b>	<b>strona 3</b>
2.1. Charakterystyka aplikacji	
2.2. Pliki wejściowe	
2.3. Pliki wyjściowe	
<b><u>3. Uruchamianie CvPcb</u></b>	<b>strona 3</b>
<b><u>4. Obsługa CvPcb</u></b>	<b>strona 4</b>
4.1. Ekran główny	
4.2. Główny pasek narzędzi	
4.3. Konfigurowanie CvPcb	
4.3.1. Główne okno	
4.3.2. Wybór bibliotek footprintów	
4.3.3. Wybór skryptów przypisać	
4.4. Wybór domyślnej ścieżki bibliotek footprintów	
4.4.1. Ścieżki poszukiwań	
4.4.2. Ścieżki użytkownika	
4.4.3. Ścieżki ustalane automatycznie przez CvPcb	
4.5. Tabele footprintów - Zarządzanie bibliotekami po nowemu	
4.5.1. Globalna tabela bibliotek footprintów	
4.5.2. Lokalna tabela bibliotek footprintów zależna od projektu	
4.5.3. Konfiguracja początkowa	
4.5.4. Dodawanie nowych wpisów w tabeli	
4.5.5. Pobieranie wartości ze zmiennych systemowych	
4.5.6. Używanie wtyczki GitHub	
4.5.7. Przykładowe wpisy w tabeli	
4.5.8. Co robić z plikami w COW?	
4.5.9. Generalne zalecenia przy używaniu tabeli bibliotek	
4.6. Podgląd bieżącego footprintu	
4.6.1. Wyświetlanie	
4.6.2. Skróty klawiaturowe	
4.6.3. Menu kontekstowe	
4.6.4. Pasek narzędzi	
4.6.5. Podgląd 3D	
<b><u>5. Przypisywanie elementom ich footprintów</u></b>	<b>strona 12</b>
5.1. Podstawy	
5.2. Operacja przypisywania	
5.3. Zmiana bieżącego przypisania	
5.4. Filtrowanie listy footprintów	
5.4.1. Filtrowanie za pomocą wzorców z ustawień komponentów	
5.4.2. Filtrowanie za pomocą pasującej liczby wyprowadzeń	
5.4.3. Filtrowanie za pomocą wybranej biblioteki	
<b><u>6. Automatyczne przypisywanie footprintów</u></b>	<b>strona 14</b>
6.1. Skrypty przypisać	
6.2. Format plików	
6.3. Przypisywanie automatyczne	
<b><u>7. Pliki numeracji wstecznej</u></b>	<b>strona 15</b>

## 1. Przeznaczenie

**CvPcb** pozwala na przypisanie każdemu **komponentowi** jaki występuje na liście sieci stworzonej przez program do edycji schematu, nazwy **footprintu**, który będzie go reprezentował na obwodzie drukowanym, a także na dodanie tych informacji do listy sieci. Generalnie, lista sieci nie określa który footprint (fizyczne odzwierciedlenie komponentu) będzie wyświetlany przez oprogramowanie do edycji obwodu drukowanego (**Pcbnew**) podczas tworzenia płytki.

Komponenty mogą mieć manualnie przypisane footprinty. Można też utworzyć skrypty przypisać (ang: *equivalence files*), które stanowią specyficzną bazę danych dla przypisywania komponentom ich footprintów. Jeśli dostępne są skrypty przypisać, jest również możliwa praca automatyczna.

Lista footprintów dostępnych dla oprogramowania PCB przechowywana jest w **bibliotekach footprintów**, których może być więcej niż jedna.

Ten interaktywny proces jest znacznie prostszy niż bezpośrednie przypisywanie tych informacji z poziomu schematu, ponieważ pozwala na jego automatyzację. **CvPcb** pozwala także na przeglądanie list dostępnych footprintów oraz podgląd ich wyglądu na ekranie.

## 2. Ogólna charakterystyka aplikacji

---

### 2.1. Charakterystyka aplikacji

Aplikacja służy do interaktywnego lub automatycznego - w oparciu o pliki skryptów - przypisywania komponentom ich footprintów.

Tworzenie (jeśli zachodzi taka potrzeba) **plików numeracji wstecznej** (ang: *back-annotation files*) dla edytora schematów z tymi przypisaniami.

### 2.2. Pliki wejściowe

- ♦ Pliki listy sieci \*.net tworzone przez program **Eeschema** (z lub bez odnośników do footprintów),
- ♦ Zewnętrzne skrypty przypisać footprintów \*.cmp stworzone wcześniej przez **CvPcb**, jeśli istnieją.

### 2.3. Pliki wyjściowe

Są generowane dwa pliki dla **Pcbnew**:

- ♦ Rozszerzony plik z listą sieci (z odnośnikami do footprintów),
- ♦ Zewnętrzny skrypt przypisać (.cmp).

## 3. Uruchamianie CvPcb

---

**CvPcb** można uruchomić z poziomu schematu (plik o tej nazwie jest wybierany po naciśnięciu klawisza **CvPcb**, z poziomu menu) lub wsadowo poprzez wpisanie z konsoli polecenia: `cvpcb <filename>`, (filename to nazwa pliku z listą sieci pochodząca z narzędzia do edycji schematów **Eeschema**).

Nazwa pliku może zostać przekazana jako plik z rozszerzeniem lub bez. Rozszerzenie pliku może, jeśli jest potrzebne zostać dostarczone przez zdefiniowaną w **CvPcb** konfigurację.

Dwa tworzone pliki będą posiadać tą samą nazwę (ale z innym rozszerzeniem).

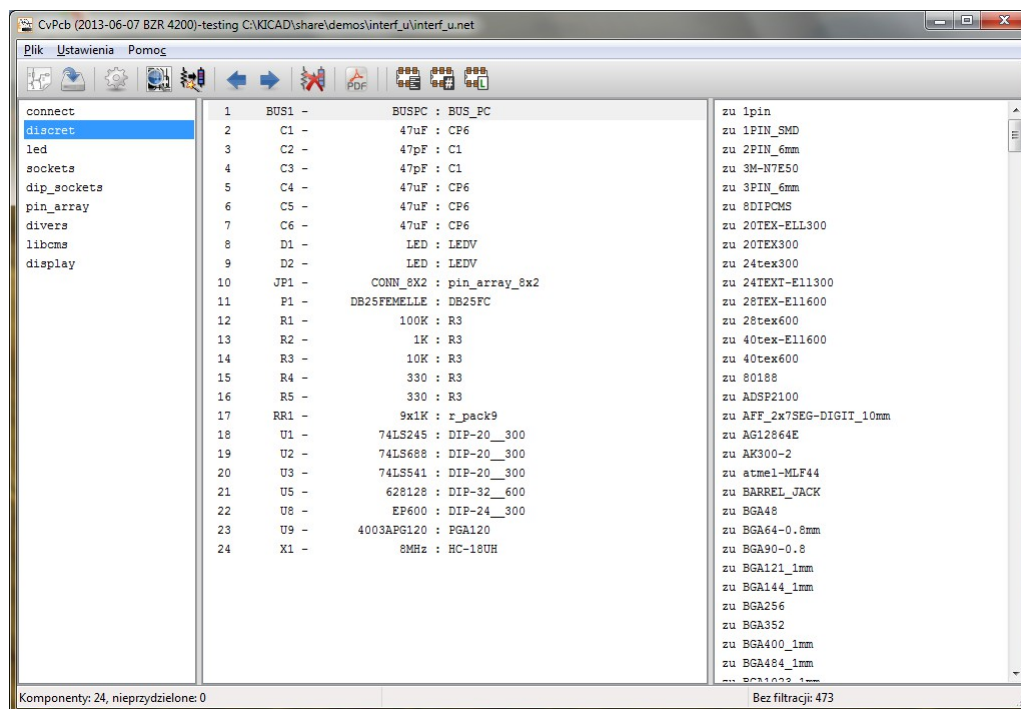
Domyślnym rozszerzeniem pliku wejściowego jest .net. Domyślnym rozszerzeniem pliku wyjściowego również jest .net, i zastępuje poprzedni plik .net.

Standardowym rozszerzeniem pliku zewnętrznego skryptu przypisać do przypisywania footprintów poszczególnym symbolom na schemacie (także generowanym przez **CvPcb**) jest .cmp.

Te standardowe rozszerzenia plików mogą być zmodyfikowane poprzez zmianę konfiguracji **CvPcb**.

## 4. Obsługa CvPcb

### 4.1. 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.

**Panel komponentów** będzie pusty jeśli żaden plik nie zostanie załadowany. **Panel footprintów** również może być pusty jeśli nie odnaleziono żadnych footprintów w bibliotekach lub też żadne z nich nie pasowały do wybranego filtra.

### 4.2. Główny pasek narzędzi



Znaczenie poszczególnych przycisków jest następujące:



Wybiera listę sieci, która ma być przetworzona.



Tworzy skrypt przypisać komponentów .cmp oraz plik .net, czyli zmodyfikowaną, rozszerzoną listę sieci.



Uruchamia menu konfiguracji **CvPcb**.



Wyświetla bieżący footprint (czyli ten który obecnie jest wskazany na liście dostępnych footprintów).



Automatycznie przypisuje nazwy footprintów korzystając z baz danych .equ o przypisaniach automatycznych. Użycie tego narzędzia domyślnie przyjmuje, że te pliki są dostępne.



Automatycznie przeskakuje do poprzedniego elementu, któremu jeszcze nie został przypisany żaden footprint.



Automatycznie przeskakuje do następnego komponentu, któremu jeszcze nie został przypisany żaden footprint.



Kasuje wszystkie przypisania.



Wyświetla dokumentację footprintu, jeśli istnieje.

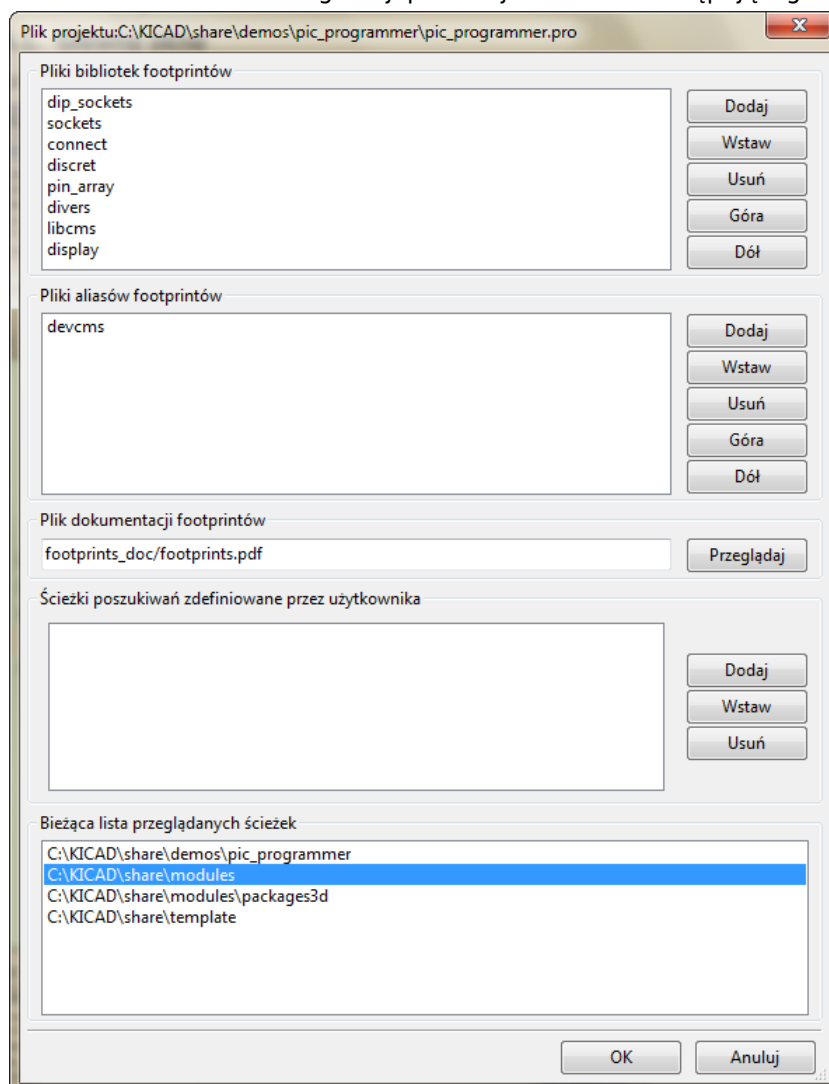


Przełącznik poszczególnych filtrów służących do filtrowania listy dostępnych footprintów na **panelu footprintów**. Możliwa jest filtracja według wzorców ustawionych w ustawieniach poszczególnych komponentów (pierwsza opcja), liczby wyprowadzeń (druga opcja) oraz według wybranej biblioteki (ostatnia opcja). Opcje te można łączyć. Gdy filtrowanie jest włączone, **panel footprintów** zawiera tylko elementy pasujące do sumy wybranych filtrów.

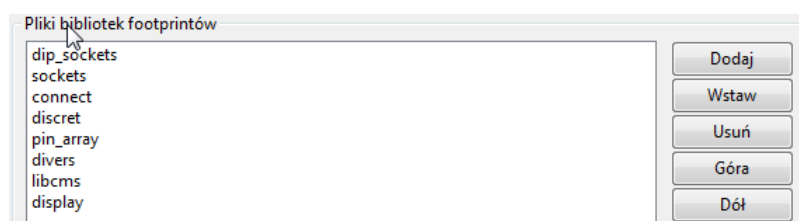
## 4.3. Konfigurowanie CvPcb

### 4.3.1. Główne okno

Uruchomienie menu konfiguracji powoduje otwarcie następującego okna dialogowego:



### 4.3.2. 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 wybraną bibliotekę niżej na liście (zmniejsza priorytet).

#### **Zapamiętaj:**

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

#### **4.3.3. Wybór skryptów przypisać**



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).

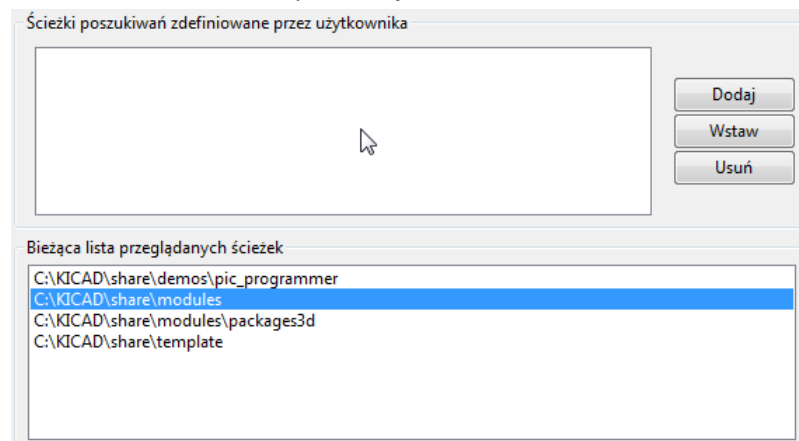
### **4.4. Wybór domyślnej ścieżki bibliotek footprintów**

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

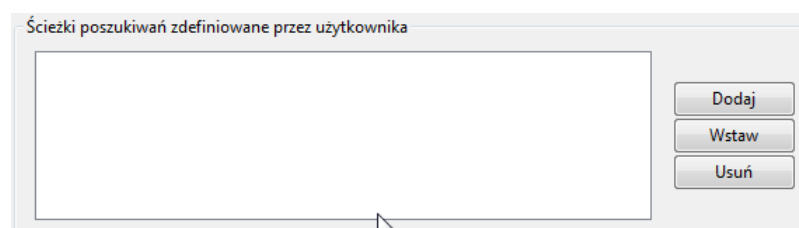
#### **4.4.1. Ścieżki poszukiwań**

**CvPcb** używa dwóch typów ścieżek:

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



#### **4.4.2. Ścieżki użytkownika**



#### **4.4.3. Ścieżki ustalane 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

## 4.5. Tabele footprintów - Zarządzanie bibliotekami po nowemu

Począwszy od wersji z dnia **2013-12-08 BZR4535-product**, **CvPcb** nie używa narzędzia do konfiguracji opisanego w punkcie 4.4. Nowa implementacja tego narzędzia opiera się na **tabeli bibliotek footprintów**.

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

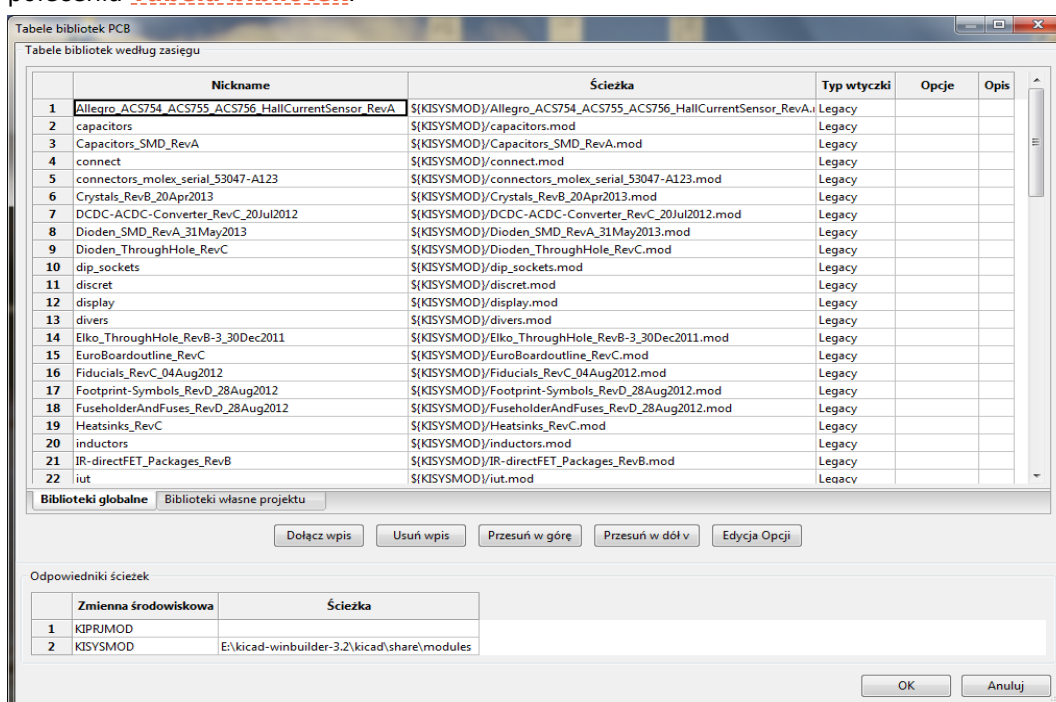


Tabela bibliotek footprintów jest używana do mapowania plików bibliotek obsługiwanych przez program do ich **nazw skrótowych**. Nazwa skrókowa jest używana do wyszukiwania footprintów zamiast poprzedniej metody z wyszukiwaniem plików zgodnie z ustalonym układem ścieżek dostępu. Pozwala to programowi **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.

### 4.5.1. 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.

### 4.5.2. 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`).



#### 4.5.3. 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**.

#### 4.5.4. 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 ścieżką 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 tą 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.

#### 4.5.5. 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 wplatać 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.

#### 4.5.6. 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](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.

#### 4.5.7. 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	<a href="https://github.com/liftoff-sr/pretty_footprints">https://github.com/liftoff-sr/pretty_footprints</a>	Github		Liftoff's footprints GH

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	<a href="https://github.com/liftoff-sr/pretty_footprints">https://github.com/liftoff-sr/pretty_footprints</a>	Github	<code>allow_pretty_writing_to_this_dir=\${HOME}/pretty/github.pretty</code>	Liftoff's footprints GH

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*.

#### 4.5.8. 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 przysyłanie zawartych w niej plików.

#### 4.5.9. 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.

### 4.6. Podgląd bieżącego footprintu

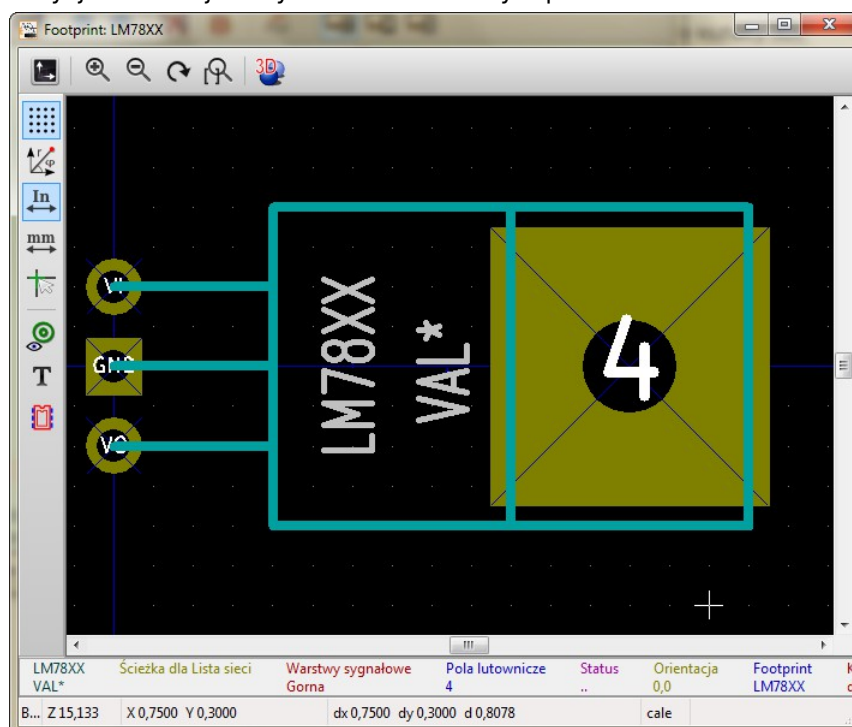
Polecenie **Podgląd footprintu** pozwala na wyświetlenie bieżącego footprintu, czyli tego który aktualnie jest wskazany na liście w panelu footprintów.

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).

#### 4.6.1. Wyświetlanie

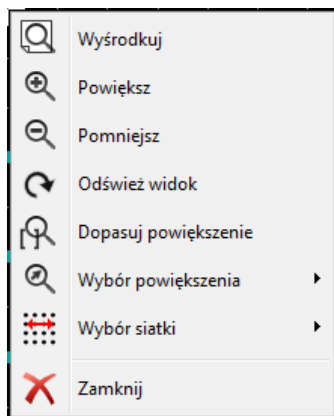
Pozycja kursora jest wyświetlana na dolnym pasku ekranu:



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

**4.6.2. Skróty klawiaturowe**




<b>F1</b>	Przybliżenie
<b>F2</b>	Oddalenie
<b>F3</b>	Odświeża obraz
<b>&lt;spacja&gt;</b>	Ustawia punkt odniesienia

**4.6.3. Menu kontekstowe**

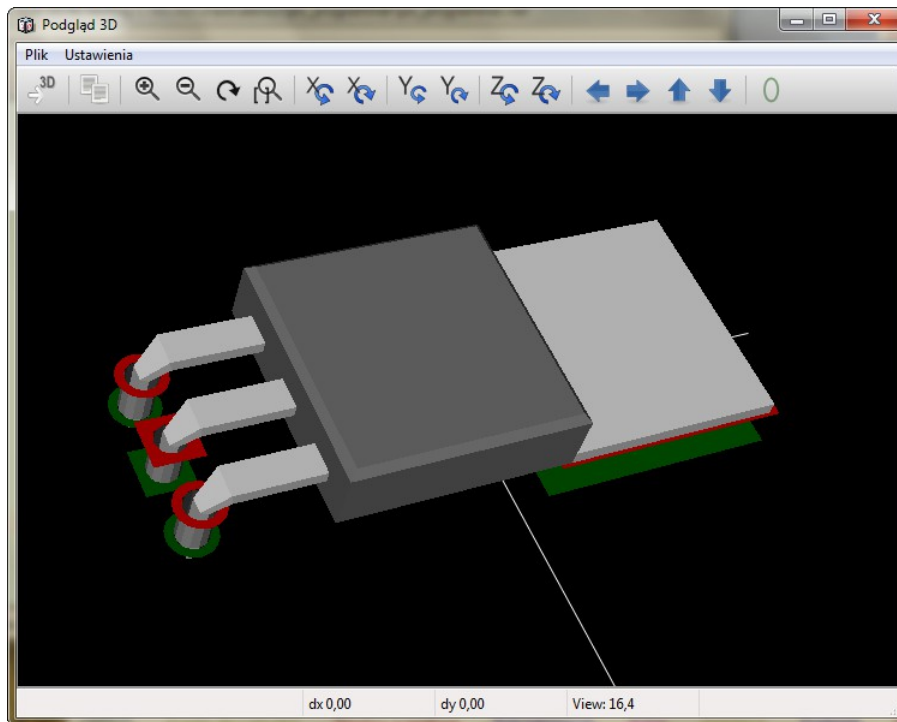
Wyświetlane jest klikając prawy klawisz myszy. Pierwsze cztery polecenia służą do bezpośredniego dostosowywania powiększenia. Dwa kolejne wyświetlają dodatkowe podmenu:

<b>Wybór powiększenia</b>	Ustawienie wybranego współczynnika powiększenia z listy dostępnych.
<b>Wybór siatki</b>	Wybór ustawienia skoku siatki z listy dostępnych.

**4.6.4. Pasek narzędzi**

	Opcje wyświetlania
	Opcje powiększania
	Wyświetlanie kształtów 3D

#### 4.6.5. Podgląd 3D



## 5. Przypisywanie elementom ich footprintów

### 5.1. 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.

### 5.2. Operacja przypisywania

Wystarczy kliknąć dwukrotnie lewym klawiszem myszy na wybranym footprintie.

### 5.3. Zmiana bieżącego przypisania

Jest wykonywana na zasadzie nowego przypisania, jak wyżej.

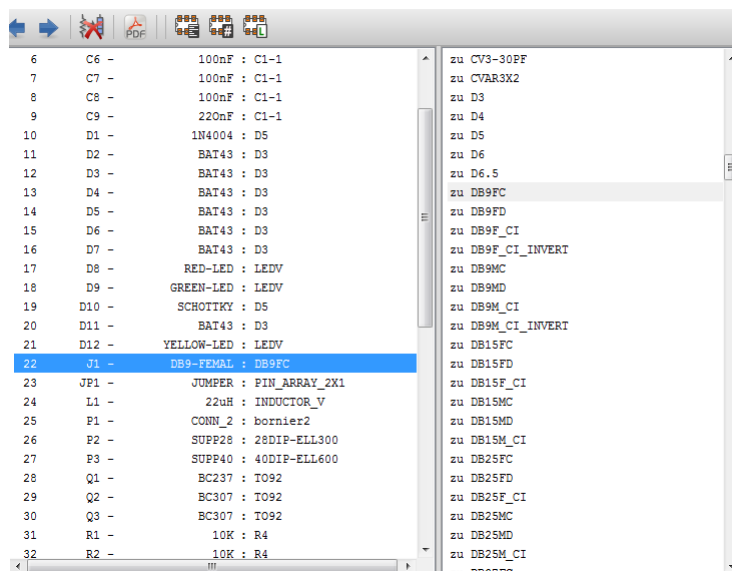
### 5.4. Filtrowanie listy footprintów



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).

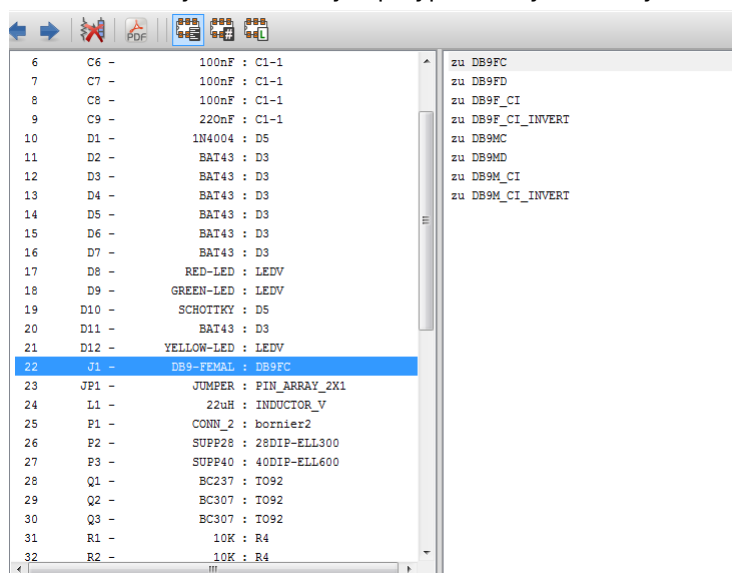
Jeśli nie ma ustalonej filtracji wyświetlana jest pełna lista footprintów. Poniższy rysunek ukazuje przykład listy **bez filtracji**:



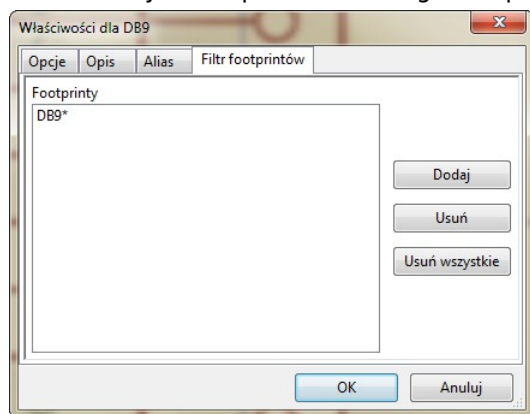
#### 5.4.1. Filtrowanie za pomocą wzorców z ustawień komponentów

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:

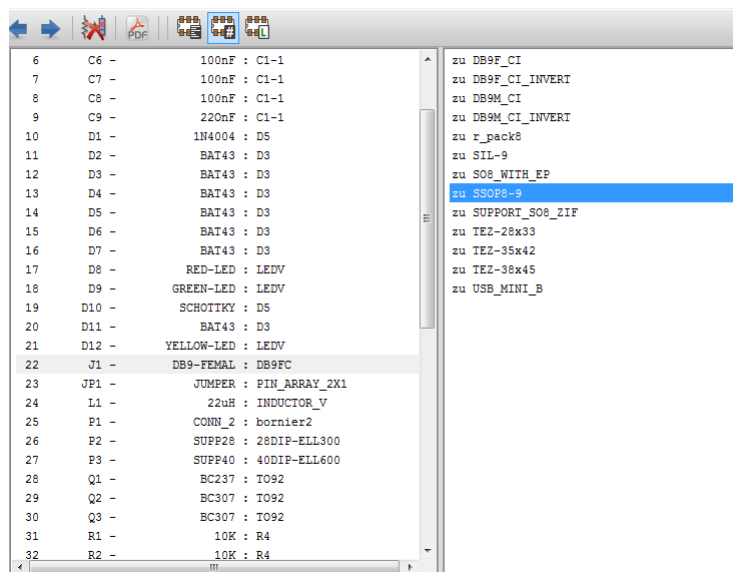


Dzieje się tak, gdyż w **Eeschema** podczas tworzenia symbolu zostały uzupełnione informacje o dozwolonych footprintach dla tego komponentu w zakładce **Filtr footprintów**:



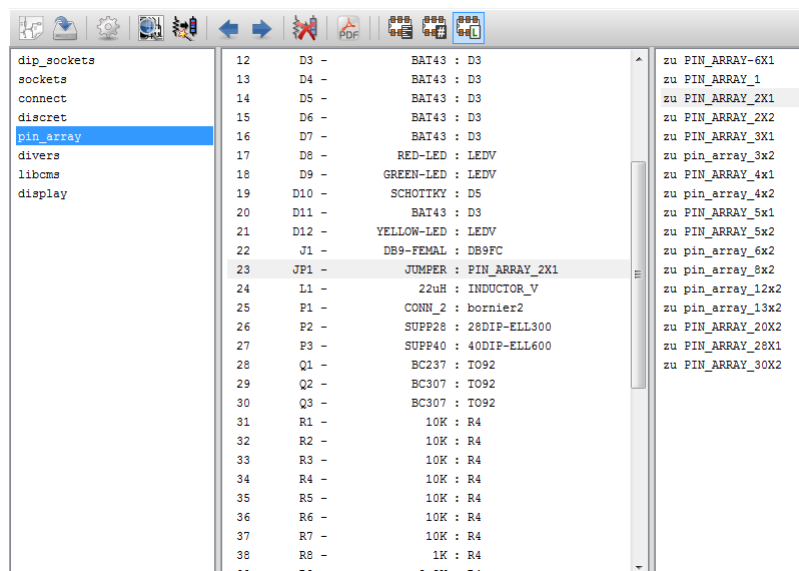
### 5.4.2. Filtrowanie za pomocą pasującej liczby wyprowadzeń

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).



### 5.4.3. 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

### 6.1. Skrypty przypisań

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 skryptów przypisań**.

### 6.2. 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'
```

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'
```

### **6.3. 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. Nie są one używane przez **Pcbnew**.

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 przypisany footprint **14DIP300**, to wygenerowana linia będzie zawierać tekst :

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

Utworzony plik będzie się nazywał tak samo jak plik wejściowy **CvPcb**, ale z rozszerzeniem .stf, i zostanie umieszczony w tym samym folderze gdzie zostanie wygenerowana nowa lista sieci.