



*CvPcb*

*Manuel de référence*

## **CvPcb.**

### Copyright

Ce document est Copyright © 2010–2013 par ses contributeurs ci dessous.

Vous pouvez distribuer et/ou modifier ce document sous les termes de la

« GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), version 3 or later, »

ou la

« Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), version 3.0 or later ».

### Contributeurs

Jean-Pierre Charras.

### Date de publication et version

Version du 21 dec 2013.

## Table des matières

1 - Rôle.....	3
2 - Caractéristiques générales.....	4
2.1 - Caractéristiques.....	4
2.2 - Fichiers d'entrée.....	4
2.3 - Fichiers de sortie.....	4
3 - Appel de CvPcb.....	4
4 - Commandes de CvPcb.....	5
4.1 - Écran principal.....	5
4.2 - Toolbar de l'écran principal.....	5
4.3 - Commandes clavier de la fenêtre principale.....	6
4.4 - Configuration de CvPcb.....	6
5 - Gestion des Librairies d'Empreintes :.....	7
5.1 - Ancien mode de gestion des Librairies:.....	7
5.1.1 - Sélection des Librairies d'empreintes.....	8
5.1.2 - Sélection des fichiers d'équivalence.....	9
5.1.3 - Sélection des répertoires par défaut.....	9
5.1.4 - Chemins de recherche:.....	9
5.1.5 - Chemins définis par l'utilisateur:.....	9
5.1.6 - Chemins automatiquement sélectionnés par CvPcb:.....	10
5.2 - Nouveau mode : Gestion des Librairies par Tables.....	10
5.2.1 - La Table des Empreintes Globale.....	11
5.2.2 - La Table des Empreintes Spécifique au Projet.....	12
5.2.3 - Initial Configuration.....	12
5.2.4 - Ajouter des entrées à la Table.....	12
5.2.5 - Substitution des Variables d'Environnement.....	13
5.2.6 - Utilisation du « Plug-in » GitHub.....	13
5.2.7 - Utilisation des Modèles.....	15
6 - Visualisation du module courant.....	15
6.1 - Affichage bas d'écran.....	16
6.2 - Commandes au clavier.....	16
6.3 - Commandes souris.....	16
6.4 - Menu « PopUp ».....	17
6.5 - Toolbar Horizontal.....	17
6.6 - Toolbar Vertical.....	17
6.7 - Affichage 3D.....	18
6.7.1 - Commandes souris.....	18
6.7.2 - Horizontal Toolbar.....	19
7 - Association composants modules.....	19
7.1 - Principe.....	19
7.2 - Association.....	19
7.3 - Modification d'une association déjà existante.....	19
7.4 - Filtrage de l'affichage des modules:.....	20
8 - Associations automatiques.....	22
8.1 - Fichiers d'équivalence.....	22
8.2 - Format.....	22
8.3 - Association:.....	23

### 1 - Rôle

## **CvPcb.**

**CvPcb** permet de compléter un fichier netliste issu d'un logiciel de Schématique en insérant, pour chaque **composant** apparaissant dans cette netliste, le nom du **module** qui le représentera sur une carte de circuit imprimé.

En effet, en général une telle netliste ne comporte pas d'indications sur ce **module** (c'est à dire le dessin physique du composant) que le logiciel de circuit imprimé (**PCBNEW**) devra placer sur le dessin général de la carte à réaliser.

Cette association entre le composant et son module correspondant se fait de façon interactive, ou/et en mode automatique, si l'on dispose de fichiers d'équivalence, que l'on peut créer soit même, et qui sont en fait des tables de correspondance entre chaque composant et son module.

La liste des modules disponibles pour le logiciel de Circuit Imprimé est contenue dans une ou plusieurs *librairies de MODULES*.

Cette approche interactive est beaucoup plus simple que de placer directement sur le schéma cette indication d'association, car **CvPcb**, outre ses possibilités d'association automatique, permet de visualiser la liste des modules disponibles, et d'afficher à l'écran ces modules.

## **2 - Caractéristiques générales**

### **2.1 - Caractéristiques**

Association interactive des composants avec les modules ou association automatique par l'intermédiaire de fichiers d'équivalence.

Génération (si nécessaire) de fichiers de retour de cette association vers la schématique.

### **2.2 - Fichiers d'entrée**

- Le fichier Netliste **.net** créé par Eeschema (avec ou sans référence aux modules)
- Le fichier auxiliaire association des composants (**.CMP**) précédemment créé par CvPcb (s'il existe).

### **2.3 - Fichiers de sortie**

Deux fichiers sont générés pour **Pcbnew**:

- Le fichier **Netliste** complet (avec référence aux modules)
- Un fichier auxiliaire association des composants (**.CMP**).

## **3 - Appel de CvPcb**

L'appel se fait par **CvPcb** ( le fichier sera alors sélectionné dans CvPcb, par un menu d'accès aux divers fichiers) ou **CvPcb <nomfichier>**, (**nomfichier** étant nom du fichier netliste à traiter, issu de l'outil schématique **Eeschema**).

Le nom du fichier peut être donné avec ou sans extension.

Les extensions seront si nécessaire complétées par celles définies par la configuration de **CvPcb**.

Les deux fichiers générés auront même nom (éventuellement avec une extension différente).

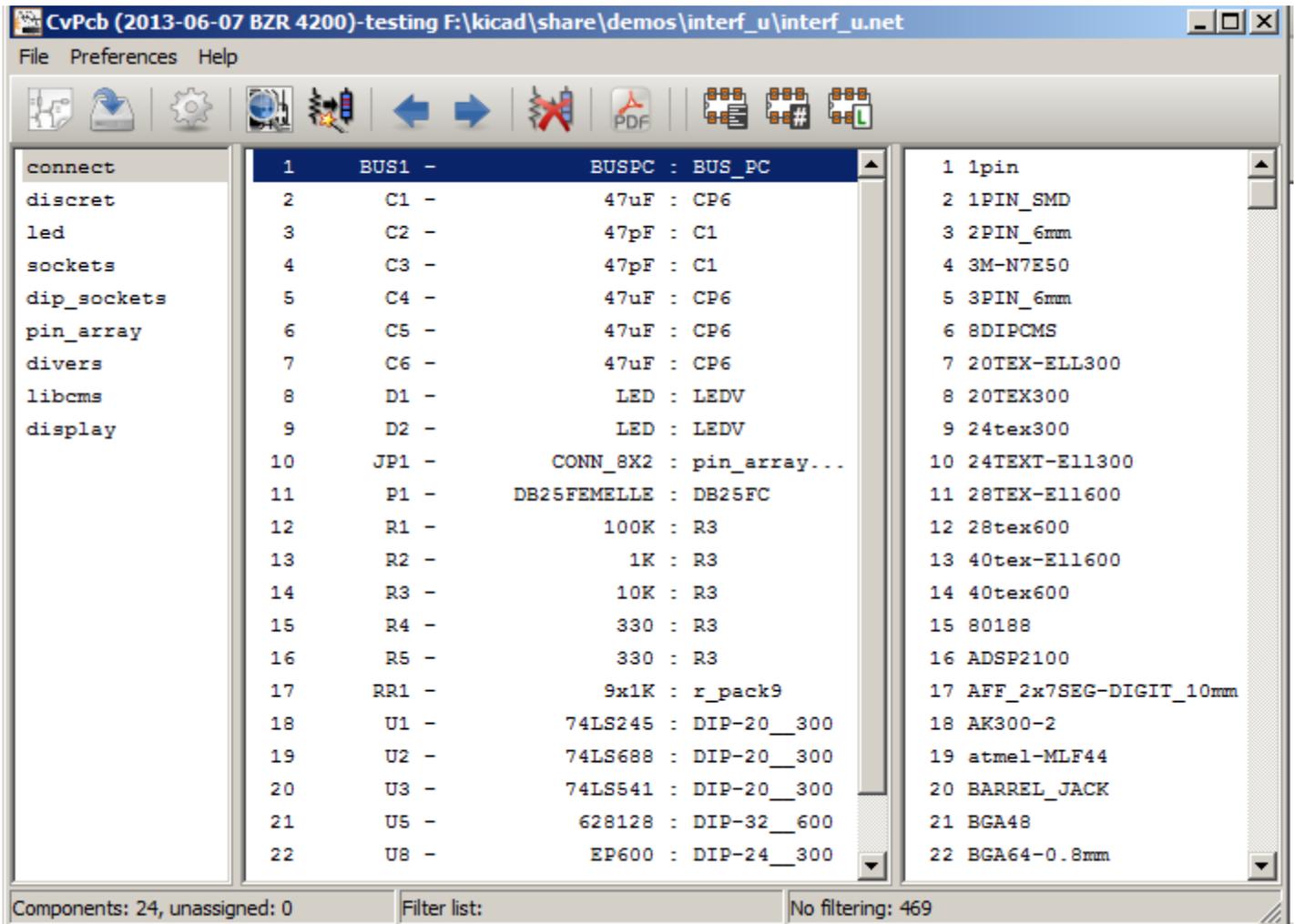
L'extension standard du fichier à traiter est **.net**.

L'extension standard du fichier netliste généré est **.net**, et remplacera l'ancien **.net**.

L'extension standard du fichier affectation des composants aux modules correspondants généré également par **CvPcb**) des affectations est **.cmp**.

## 4 - Commandes de CvPcb

### 4.1 - Écran principal



La fenêtre **Librairies** (à gauche) affiche la liste des librairies associées au projet.

La fenêtre **Composants** (au centre) affiche la liste des composants apparaissant dans la Netliste lue.

La fenêtre **Modules** (à droite) affiche la liste des modules contenus dans les librairies lues.

La fenêtre **Composants** peut être vide si aucun fichier n'a été chargé, et la fenêtre **Modules** peut être aussi vide si aucune librairie de module n'a été trouvée.

### 4.2 - Toolbar de l'écran principal



Les différentes commandes sont:

## CvPcb.

	Sélection du fichier Netliste à traiter.
	Création du fichier <b>.CMP</b> (liste des associations) et du fichier <b>.NET</b> , Netliste modifiée et complète.
	Appel au menu de configuration de CvPcb.
	Affichage du module courant (c'est à dire dont le nom est en surbrillance dans la fenêtre des modules.
	Association automatique composants/modules à partir des fichiers d'équivalence. L'utilisation de cette commande suppose que l'on dispose évidemment de ces fichiers.
	Défilement automatique des composants vers le début de la liste jusqu'au premier composant non affecté à un module.
	Défilement automatique des composants vers la fin de la liste jusqu'au premier composant non affecté à un module.
	Effacement total des affectations déjà effectuées
	Accès à la documentation des modules.
	Active ou désactive le filtrage pour limiter la liste des empreintes à celles du filtre d'empreintes du composant sélectionné.
	Active ou désactive le filtrage pour limiter la liste des empreintes en utilisant le nombre de pins du composant sélectionné.
	Active ou désactive le filtrage pour limiter la liste des empreintes au contenu de la librairie sélectionné.

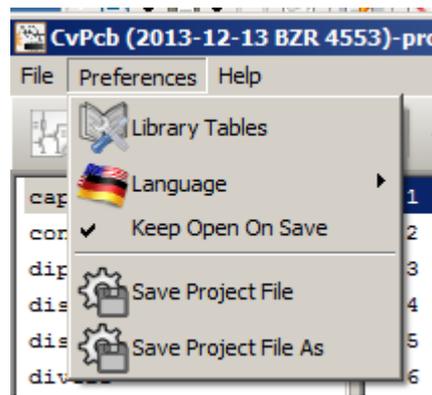
### 4.3 - Commandes clavier de la fenêtre principale

La table suivante liste les commandes au clavier de la fenêtre principale:

Flèche droite Tab	Active la liste suivante à la droite de la liste sélectionnée. Revient à la première liste de gauche après la sélection de la dernière liste
Flèche gauche	Active la liste suivante à la gauche de la liste sélectionnée. Revient à la première liste de droite après la sélection de la dernière liste
Flèche montante	Sélection du précédent élément de la liste sélectionnée.
Flèche descendante	Sélection de l'élément suivant de la liste sélectionnée.
Page Up	Sélection de l'élément de la page précédente de la liste sélectionnée.
Page Down	Sélection de l'élément de la page suivante de la liste sélectionnée.
Début/Home	Sélection du premier élément de la liste sélectionnée.
Fin/End	Sélection du dernier élément de la liste sélectionnée.

### 4.4 - Configuration de CvPcb

## CvPcb.



CvPcb peut être automatiquement fermé après sauvegarde du fichier d'association des empreintes, ou non.

Sélectionner le menu “Librairies” dans le menu “Préférences” affiche le dialogue de configuration des librairie.

Selon la version de CvPcb, il y a 2 différents modes de gestion des libraires:

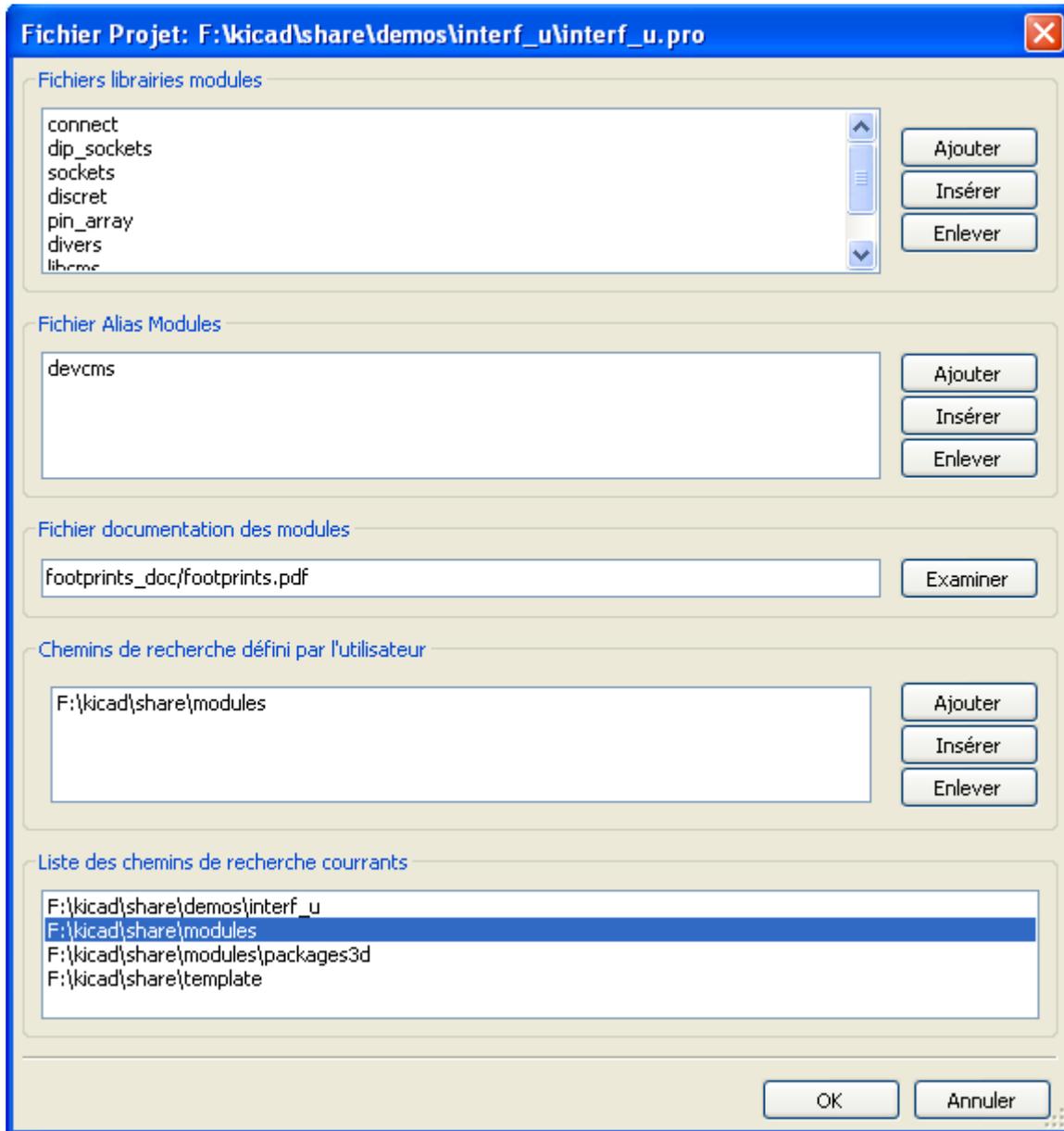
- L'ancienne gestion « legacy management », basé sur les fichiers \*.mod, et une list de ces fichiers librairie
- Le nouveau format “Pretty”, utilisant un fichier par empreinte. Utilisant une liste de répertoires. Chaque répertoire (nom de répertoire « \*.pretty ») est une librairie. En utilisant le nouveau mode, vous pouvez aussi utiliser des librairie natives venant de GEDA/GPCB ou au format Eagle xml.

## 5 - Gestion des Librairies d'Empreintes :

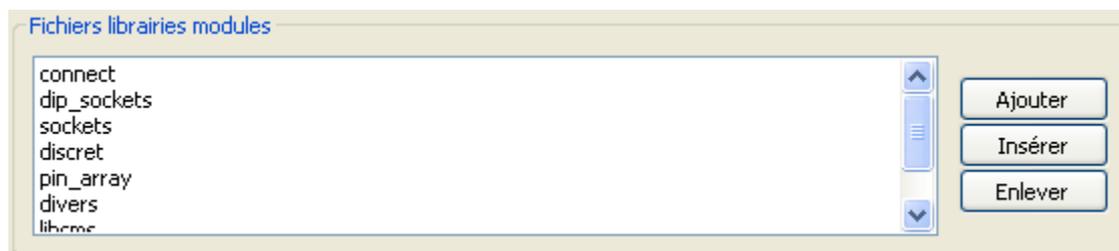
### 5.1 - Ancien mode de gestion des Librairies:

## CvPcb.

L'appel au menu de configuration affiche l'écran suivant :



### 5.1.1 - Sélection des Librairies d'empreintes



Sélectionner à la souris un nom de fichier.

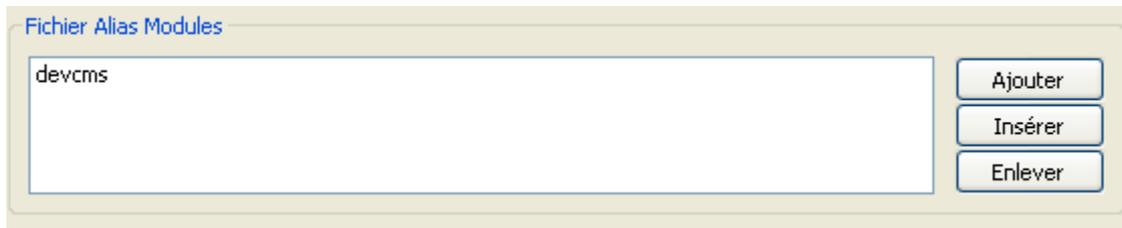
- **Enlever** supprime ce nom de la liste.
- **Ajouter** ajoute un nouveau nom à la liste, a la fin de la liste
- **Insérer** ajoute un nouveau nom à la liste, **avant** le nom sélectionné.

## CvPcb.

### Remarque:

Toute modification de cette liste affecte aussi *Pcbnew*.

### 5.1.2 - Sélection des fichiers d'équivalence



Fichier Alias Modules

devcms

Ajouter

Insérer

Enlever

Sélectionner à la souris un nom de fichier.

- **Enlever** supprime ce nom de la liste.
- **Ajouter** ajoute un nouveau nom à la liste, à la fin de la liste
- **Insérer** ajoute un nouveau nom à la liste, **avant** le nom sélectionné

### 5.1.3 - Sélection des répertoires par défaut

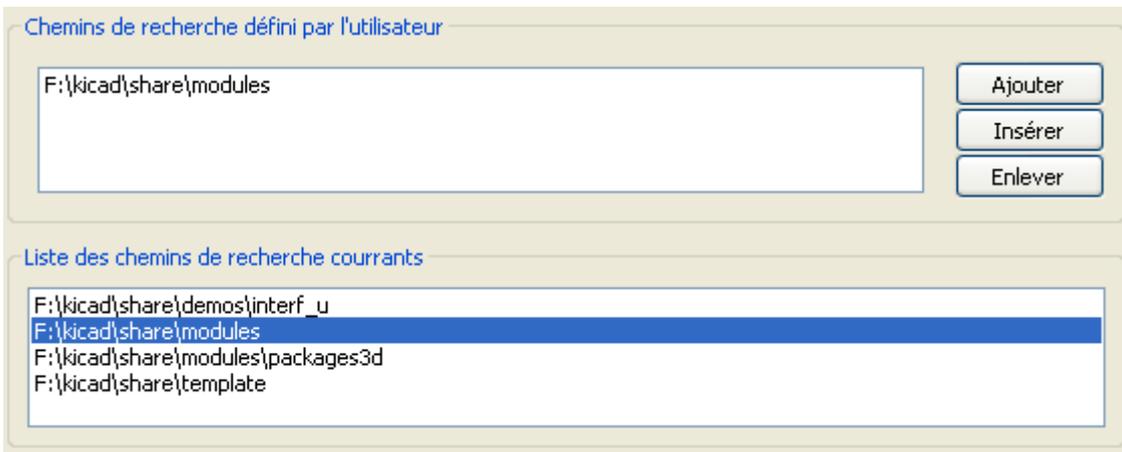
Les chemins de recherche par défaut sont affichés par CvPcb.

C'est dans ces chemins que CvPcb cherche les fichiers librairies de modules (\*.mod) et les fichiers d'équivalence (\*.equ).

### 5.1.4 - Chemins de recherche:

Il y a 2 types de chemins:

- Ceux qui sont toujours automatiquement sélectionnés par CvPcb.
- Ceux qui sont ajoutés par l'utilisateur.



Chemins de recherche défini par l'utilisateur

F:\kicad\share\modules

Ajouter

Insérer

Enlever

Liste des chemins de recherche courants

F:\kicad\share\demos\interf\_u

F:\kicad\share\modules

F:\kicad\share\modules\packages3d

F:\kicad\share\template

### 5.1.5 - Chemins définis par l'utilisateur:



Chemins de recherche défini par l'utilisateur

F:\kicad\share\modules

Ajouter

Insérer

Enlever

**CvPcb.**

### **5.1.6 - Chemins automatiquement sélectionnés par CvPcb:**

Ils dépendent en partie du système d'exploitation utilisé.

Il y a toujours le **répertoire de travail courant**.

Ensuite:

- ***kiCAD/share/modules***.
- ***kiCAD/share/modules/packages3d*** (pour les fichiers de formes 3D au format vrmf généré par Wings3D).
- ***kiCAD/share/template***.

Le chemin de base *kiCAD* est

- Le chemin où l'exécutable *kiCAD.exe* est trouvé (.../kiCAD/bin.

Si non trouvé:

*Sous Windows:*

- c:\kiCAD
- d:\kiCAD

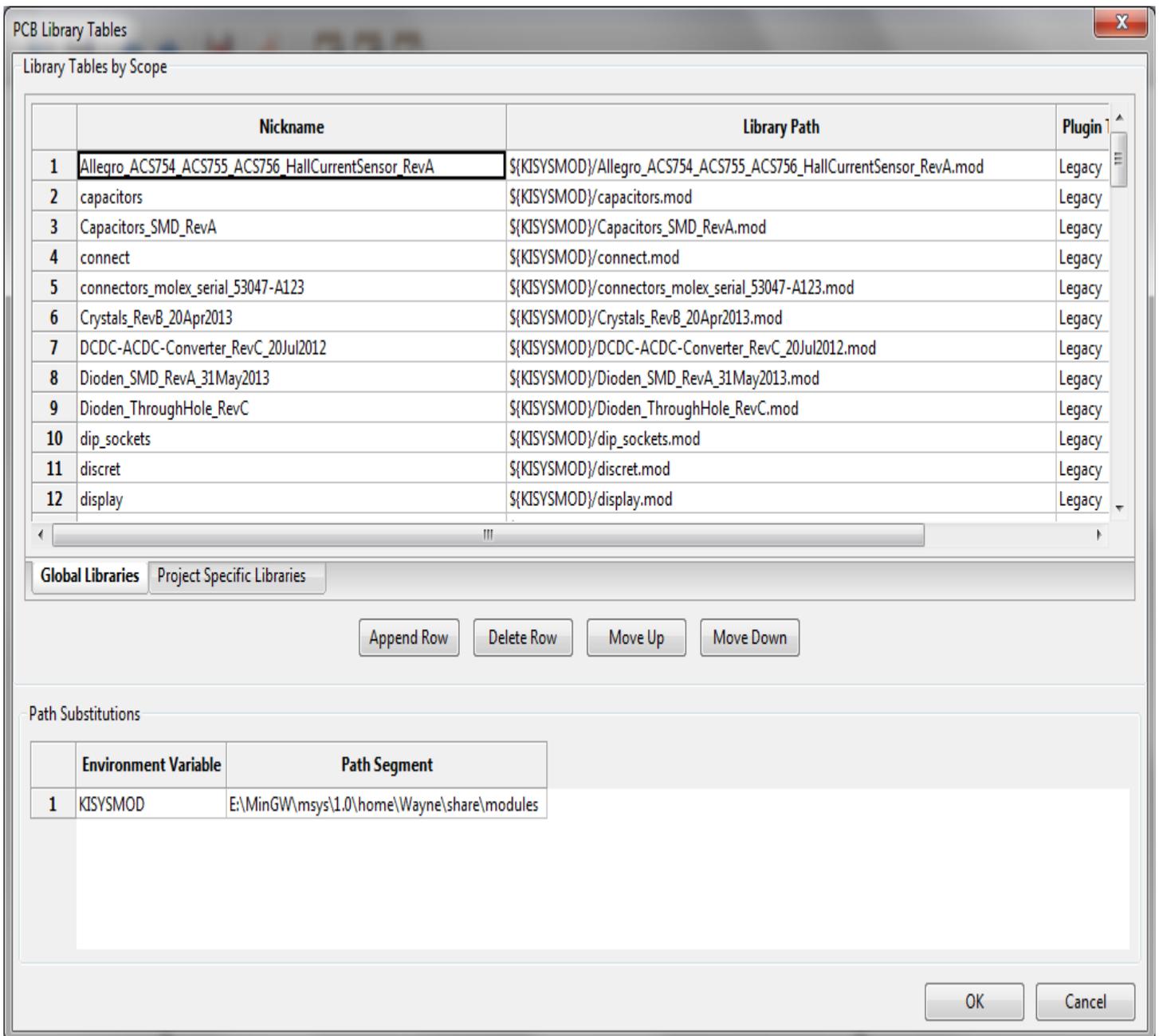
*Sous un système Unix:*

- /usr/local/kiCAD
- /usr/share/kiCAD

## **5.2 - Nouveau mode : Gestion des Librairies par Tables**

Depuis Décembre 2013, CvPcb n'utilise plus l'outil de gestion de librairies décrit dans la [section 5.1](#). La nouvelle méthode utilise une table donnant la méthode de gestion des librairies.

L'image ci-dessous montre le dialogue d'édition de la table des empreintes qui peut être ouvert par le menu "Table des Librairies" dans le menu "Préférences".



La table des librairies est utilisée pour associer une librairie physique de n'importe quel type supporté (actuellement, Kicad, Geda pcb, Eagle) à un nom logique ("nickname", dans la version anglaise).

**Ce nom logique est utilisé pour chercher les empreintes dans une librairie précise**, au lieu de la méthode précédente qui cherchait l'empreinte dans une liste ordonnée de librairies, et qui chargeait la première empreinte trouvée.

Ceci permet à CvPcb d'accéder à des empreintes de nom identique, dans différentes librairies en étant sûr que l'empreinte correcte est chargée de la librairie appropriée.

Cela permet aussi à CvPcb de supporter le chargement de librairies venant de différents éditeurs de PCB tels que Eagle et GEDA.

### 5.2.1 - La Table des Empreintes Globale

La table des empreintes globale contient la liste des librairies qui sont toujours disponibles pour tout projet, donc quel que soit le projet chargé.

Cette table est sauvegardée dans le fichier **fp-lib-table** dans le répertoire d'accueil de l'utilisateur. Cet endroit dépend du système d'exploitation.

### 5.2.2 - La Table des Empreintes Spécifique au Projet

La table des empreintes spécifique au projet contient la liste des librairies qui sont disponibles pour le projet qui est chargé.

La table des empreintes spécifique au projet peut être éditée seulement lorsque elle est chargée en même temps que le fichier netliste du projet.

Si aucun fichier projet n'est chargé, ou s'il n'y a aucune table des librairies dans le répertoire du projet, une table vide est créée (et pourra être éditée plus tard), en même temps que le fichier de lien des empreintes (.cmp).

### 5.2.3 - Initial Configuration

La première fois que CvPcb or Pcbnew est lancé et si la table des empreintes globale fp-lib-table n'est pas trouvée, dans le répertoire d'accueil de l'utilisateur, CvPcb tentera de copier la table globale par défaut **fp-lib-table** stockée dans le répertoire **template** de **KiCad** dans le répertoire d'accueil de l'utilisateur.

Si **fp-lib-table** ne peut être trouvé, une table des empreintes vide sera créée dans le répertoire d'accueil de l'utilisateur.

Si cela arrive, l'utilisateur peut soit copier une table manuellement, soit la configurer manuellement.

Le fichier table des empreintes par défaut **fp-lib-table** qui inclut toutes les librairies d'empreintes standard qui sont installées fait partie de la distribution **KiCad**.

### 5.2.4 - Ajouter des entrées à la Table

Pour utiliser une librairie d'empreintes, il faut d'abord l'ajouter à la table globale, ou à la table spécifique au projet.

**La table spécifique au projet n'est accessible que si vous avez chargé un fichier netliste.**

Chaque librairie doit avoir un nom logique unique. Il n'a pas de relation avec le nom physique ou le chemin de la librairie réelle. Le caractère `:` est un caractère réservé, et ne peut être utilisé dans le nom logique.

Chaque librairie doit avoir un chemin et/ou un nom de fichier valide, selon le type de librairie.

Les chemins peuvent être définis de façon absolue, relative ou par substitution de variables d'environnement. (voir paragraphe suivant).

Le « plug in » approprié (type de librairie) doit être sélectionné de façon à ce que la librairie puisse être lue correctement.

CvPcb supporte actuellement les formats :

- KiCad legacy (.mod files)
- KiCad Pretty (répertoires contenant des fichiers .kicad\_mod , un par empreinte)
- Eagle,
- GEDA (répertoires contenant des fichiers .fp , un par empreinte)

Il y a aussi un champ description pour ajouter un commentaire pour l'entrée de la table.

Le champ Option n'est pas encore utilisé, et n'a donc pas d'effet lors du chargement des librairies.

Notez également que vous ne pouvez pas avoir des noms logiques dans la même table.

Cependant vous pouvez avoir des noms dupliqués entre la table globale et la table locale spécifique.

La table locale spécifique a la priorité sur la table globale lorsque des noms dupliqués sont trouvés.

Lorsque des entrées sont définies dans la table spécifique au projet , un fichier **fp-lib-table** contenant ces entrées sera écrit dans le répertoire de la netliste ouverte.

## 5.2.5 - Substitution des Variables d'Environnement

Une des plus puissantes caractéristiques de la table des empreintes est la substitution des variables d'environnement.

Ceci vous permet de définir les chemins « utilisateur » où vos bibliothèques sont stockées par des variables d'environnement.

La substitution des variables d'environnement est faite par l'utilisation de la syntaxe

« `#{ENV_VAR_NAME}` »

dans le chemin des bibliothèques.

Par défaut, lorsque CvPcb est lancé, il définit la variable d'environnement « **KISYSMOD** » .

Cette variable définit le chemin où les bibliothèques d'empreintes qui sont installées avec KiCad sont placées.

Vous pouvez redéfinir vous même \$KISYSMOD pour substituer vos propres bibliothèques à celles des bibliothèques par défaut de KiCad.

Lorsque une netliste est chargée, CvPcb définit aussi **KIPRJMOD**.

Ceci vous permet de créer des bibliothèques dans le répertoire projet sans avoir à définir le chemin absolu des bibliothèques dans la table des bibliothèques spécifiques au projet.

## 5.2.6 - Utilisation du « Plug-in » GitHub

GitHub est un Plug-in spécial qui fournit une interface pour un accès distant en lecture seule à un dépôt GitHub composé de fichiers d'empreintes « pretty » (« Pretty » est le nom du format des nouveaux fichiers d'empreintes KiCad) et fournit un support optionnel « Copy On Write » (COW) pour l'édition des empreintes lues à partir du dépôt GitHub pour les sauver localement.

Par conséquent le plug-in "GitHub" fournit un **accès distant en lecture seule adès depots de bibliothèques au format « pretty »** à <https://github.com>.

Pour ajouter une entrée GitHub à la table des empreintes, le « Chemin Bibliothèque » dans ligne correspondante de la table des empreintes doit être une URL GitHub valide.

Par exemple:

[https://github.com/liftoff-sr/pretty\\_footprints](https://github.com/liftoff-sr/pretty_footprints)

or

<https://github.com/KiCad>

Typiquement GitHub URLs sont de la forme:

[https://github.com/user\\_name/repo\\_name](https://github.com/user_name/repo_name)

The « Plugin Type » doit être « Github ». Pour activer le mode « **Copy On Write** » (**COW**) l'option **allow\_pretty\_writing\_to\_this\_dir** doit être ajoutée dans les »Options » de la table des empreintes pour cette entrée.

Cette option est le "Chemin Bibliothèque" pour le stockage local des copies modifiées des empreintes lues du depot GitHub.

Les empreintes sauvées dans ce repertoire sont combinées avec la partie enlecture seule du dépôt GitHub pour créer la bibliothèque complète d'empreintes.

Si cette option est manquante, alors la bibliothèque GitHub library est en lecture seule.

Si cette option est présente pour une bibliothèque GitHub, alors toute écriture dans cette bibliothèque hybride se fera dans le répertoire local \*.pretty.

## CvPcb.

Noter que la partie de cette « librairie hybride COW » résidente sur github.com est toujours en lecture seule, ce qui signifie que vous ne pouvez pas supprimer ou modifier une empreinte directement dans le dépôt GitHub spécifié.

Le type de la librairie composite reste "Github" dans ce qui suit, mais il consiste à la fois de la portion locale en écriture/lecture et de la portion distante en lecture seule.

La table ci dessous montre une entrée dans la table des librairies sans l'option **allow\_pretty\_writing\_to\_this\_dir**:

Nickname	Library Path	Plugin Type	Options	Description
github	<a href="https://github.com/liftoff-sr/pretty_footprints">https://github.com/liftoff-sr/pretty_footprints</a>	Github		Liftoff's GH footprints

La table ci dessous montre une entrée dans la table des librairies avec l'option COW.

Noter que l'utilisation de la variable d'environnement `${HOME}` est just un exemple.

Le répertoire `github.pretty` est dans le chemin `${HOME}/pretty/`.

Chaque fois que vous utilisez l'option **allow\_pretty\_writing\_to\_this\_dir**, vous devez créer ce répertoire manuellement à l'avance et le nom doit finir par l'extension **.pretty**.

Nickname	Library Path	Plugin Type	Options	Description
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 GH footprints

Le chargement des empreintes donne la priorité aux empreintes locale trouvées dans le répertoire donné dans l'option **allow\_pretty\_writing\_to\_this\_dir**.

Une fois que vous avez sauvé une empreinte dans le répertoire librairie COW locale par la commande **footprint save** de l'éditeur d'empreintes, aucune mise à jour sur GitHub ne sera vue lors du chargement de l'empreinte qui a le même nom que celle sauvée localement.

Toujours garder un répertoire local `*.pretty` séparé pour chaque librairie GitHub, ne jamais les combiner par utilisation du même répertoire pour plusieurs librairies GitHub.

De plus ne pas utiliser ces répertoires COW (`*.pretty`) dans une autre entrée de la table des librairies. Cela pourrait créer des problèmes.

La valeur donnée dans l'option **allow\_pretty\_writing\_to\_this\_dir** peut utiliser n'importe quelle variable d'environnement, en utilisant la notation `${}` pour créer le chemin de la même manière que dans le paramètre « Library Path ».

Quel est l'intérêt de l'option COW?

Il est de faciliter le partage des empreintes entre utilisateurs.

Si périodiquement vous envoyez COW les empreintes modifiées au mainteneur du dépôt GitHub, vous pouvez aider à la mise à jour des copies sur GitHub.

Simplement envoyez les fichiers individuels `*.kicad_mod` que vous trouvez dans les répertoires COW au mainteneur du dépôt GitHub. Après réception de la confirmation que vos changements ont été pris en compte, vous pouvez sans problème supprimer le(s) fichier(s) COW file(s) et les empreintes mises à jour pour la partie lecture seule la librairie GitHub seront utilisées.

Votre but doit être d'avoir aussi peu de fichiers COW que possible en contribuant fréquemment aux copies partagées à <https://github.com>.

(voir <https://github.com/KiCad>)

### 5.2.7 - Utilisation des Modèles

Les librairies d'empreintes peuvent être définies soit globalement soit spécifiquement au projet.

Les librairies d'empreintes définies dans la table globale sont toujours disponibles et sont stockées dans le fichier **fp-lib-table** dans le répertoire d'accueil de l'utilisateur.

Les librairies d'empreintes globales peuvent toujours être accédées même si aucune netliste du projet n'est chargée.

La table des empreintes spécifique au projet n'est active que pour la netliste chargée.

La table des empreintes spécifique au projet est sauvee dans le fichier **fp-lib-table** dans le répertoire de la netliste chargée.

Vous êtes libre de définir les librairies à charger dans l'une ou l'autre des tables.

Il y a des avantages et des inconvénients dans chaque méthode.

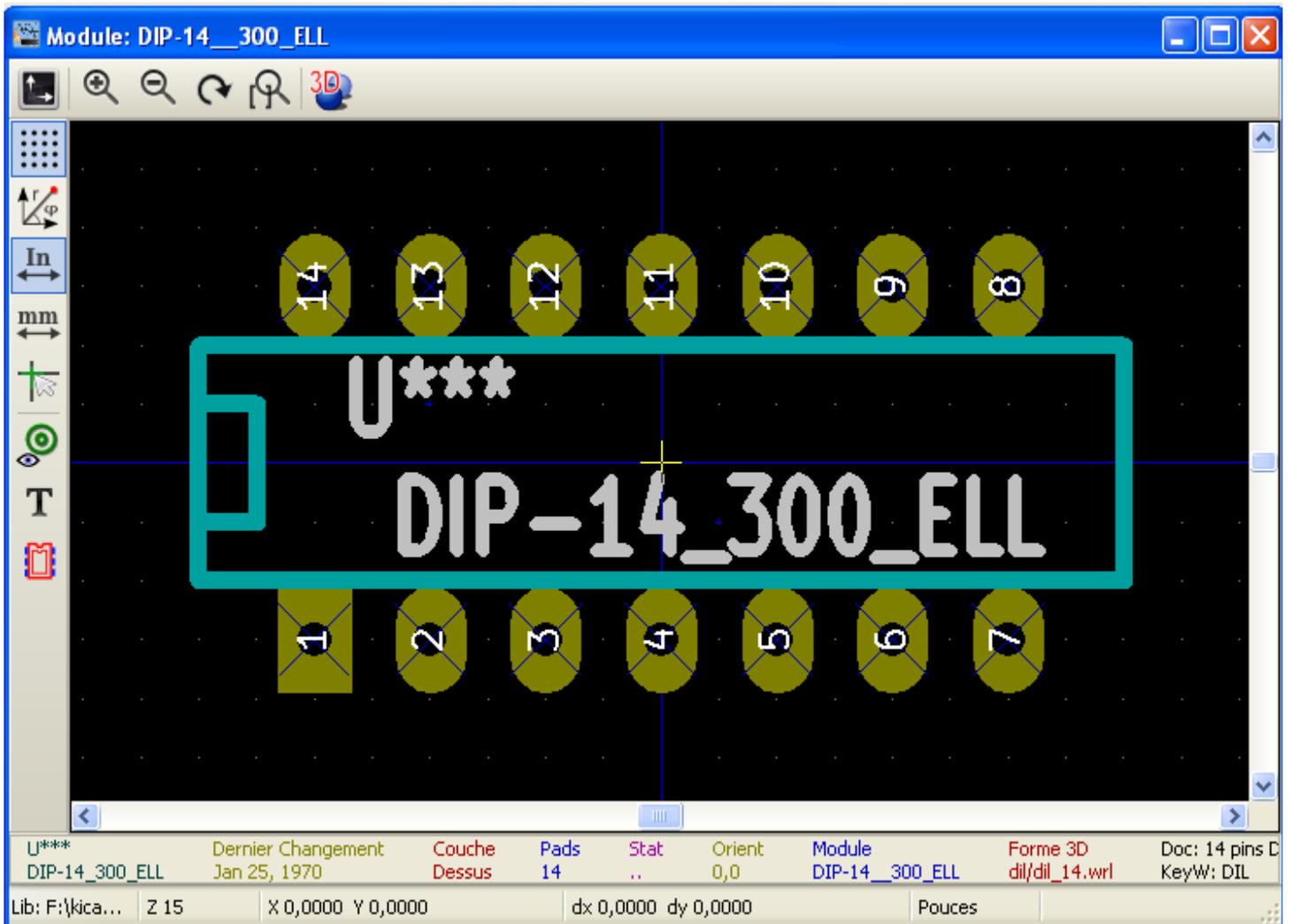
- Vous pouvez définir toutes vos librairies dans la table globale, ce qui signifie qu'elles seront toujours disponibles quand vous en aurez besoin  
L'inconvénient est que vous devrez chercher dans une longue liste d'empreintes inutiles celle dont vous aurez besoin.
- Vous pouvez définir toutes vos librairies sur la base d'un projet spécifique.  
L'avantage est que vous définissez uniquement les librairies utiles au projet, ce qui réduit considérablement la recherche.  
L'inconvénient est que vous devez vous souvenir de définir la liste des librairies pour chaque nouveau projet.
- Vous pouvez aussi définir à la fois des librairies globalement et spécifiquement au projet.  
Un modèle pourrait être de définir les librairies les plus communément utilisées globalement et les quelques librairies plus spécifiques exigées par le projet dans la table du projet.  
Il n'y a pas de restrictions à la façon de définir la liste des librairies.

## 6 - Visualisation du module courant

La commande Visu permet l'affichage du module courant, c'est à dire celui qui apparaît en surbrillance sur la ligne centrale de la fenêtre *Modules*.

On peut afficher les différents modules en cliquant sur le module désiré (dans la liste des modules), tant que cette fenêtre est affichée

On peut aussi l'afficher en représentation 3D (si elle a été créée et associée au module)



### 6.1 - Affichage bas d'écran

En bas de l'écran sont affichées les coordonnées curseur :

Coordonnées absolues (X nnnn Y nnnn) et coordonnées relatives (dx nnnn dy nnnn)

Les coordonnées relatives sont remises à 0 par la barre d'espace.

### 6.2 - Commandes au clavier

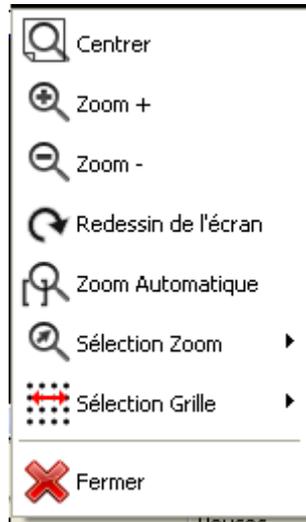
<b>F1</b>	Agrandissement.(Zoom +)
<b>F2</b>	Réduction (Zoom-)
<b>F3</b>	Rafraichissement de l'affichage.
<barre <b>espace</b> > :	Remise à zéro des coordonnées relatives.

### 6.3 - Commandes souris

Scroll Wheel	Zoom + et - centré autour du curseur
Ctrl + Scroll Wheel	Pan horizontal
Shift + Scroll Wheel	Pan vertical

Right Button Click	Ouvre un menu « pop up »
--------------------	--------------------------

## 6.4 - Menu « PopUp »



Affiché par la souris sur action sur le bouton de droite:

<b>Sélection Zoom</b>	Sélection directe du zoom pour affichage.
<b>Sélection grille</b>	Sélection directe de la grille.

## 6.5 - Toolbar Horizontal



	Accès aux options d'affichage
	Zoom +
	Zoom -
	Rafraîchissement écran
	Meilleur zoom
	Ouvrer la fenêtre d'affichage en 3D

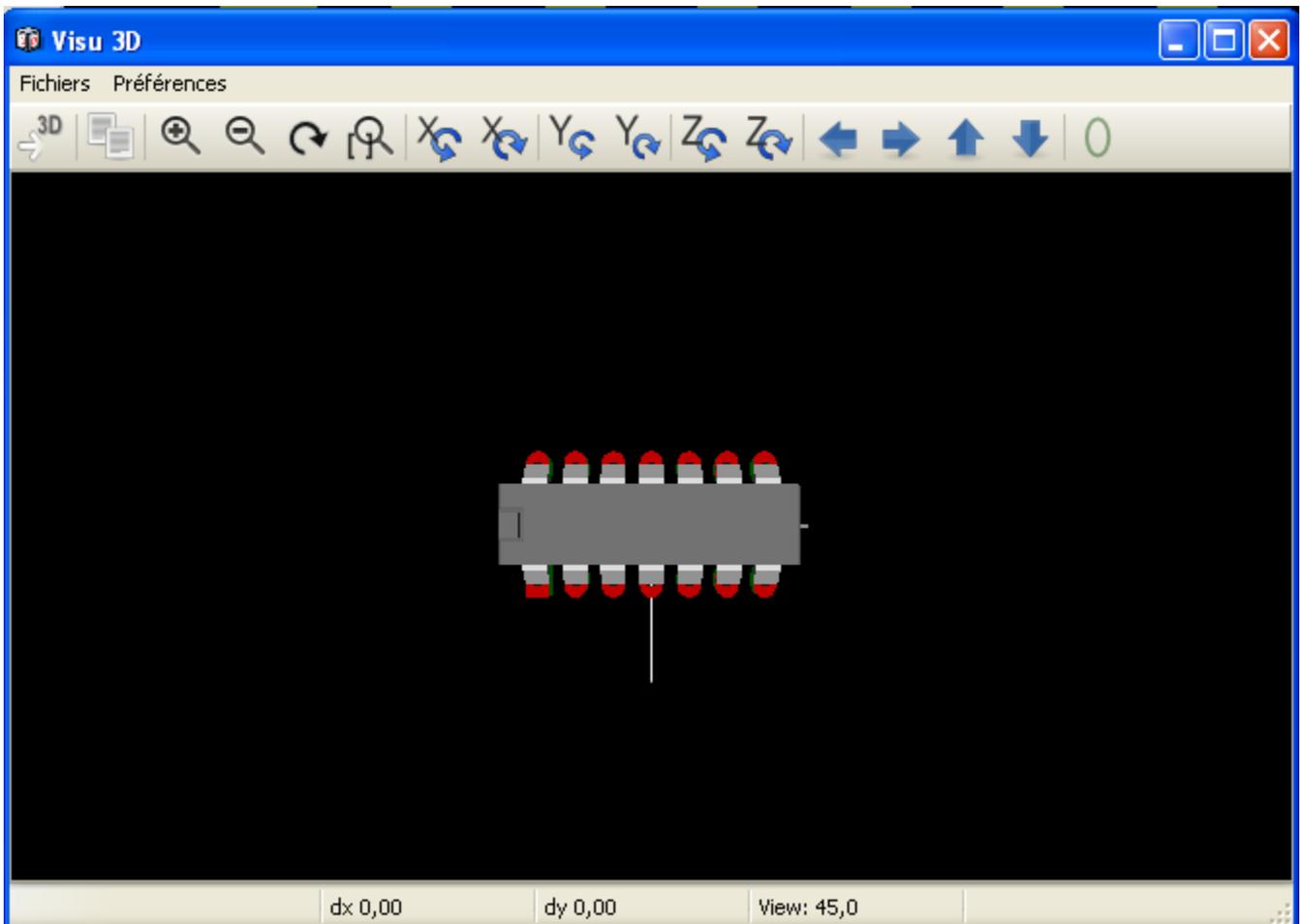
## 6.6 - Toolbar Vertical

	Affiche ou non la grille
	Affiche les coordonnées en mode polaire ou cartésien

## CvPcb.

	Affiche les coordonnées en pouces
	Affiche les coordonnées en millimètres
	Change le style du curseur
	Bascule l'affichage des pads du mode contour au mode normal
	Bascule l'affichage des textes du mode contour au mode normal
	Bascule l'affichage des contours du mode contour au mode normal

## 6.7 - Affichage 3D



### 6.7.1 - Commandes souris

Scroll Wheel	Zoom in and out at the current cursor position
Ctrl + Scroll Wheel	Pan right and left
Shift + Scroll Wheel	Pan up and down

## 6.7.2 - Horizontal Toolbar

	Reload the 3D model
	Copy 3D image to clipboard
	Zoom in
	Zoom out
	Redraw
	Fit drawing in display area
	Rotate backward along the X axis
	Rotate forward along the X axis
	Rotate backward along the Y axis
	Rotate forward along the Y axis
	Rotate backward along the Z axis
	Rotate forward along the Z axis
	Pan left
	Pan right
	Pan up
	Pan down
	Toggle orthographic projection mode on and off

## 7 - Association composants modules

### 7.1 - Principe

Dans fenêtre des modules il suffit de double-cliquer sur le nom du **module** voulu (Ce nom est affiché en **surbrillance**), pour l'associer avec le **composant** dont le nom est **en surbrillance sur la ligne centrale** de la fenêtre des Composants.

La liste des composants peut aussi défiler:

- Automatiquement après une association.
- Volontairement par commande de défilement (ascenseur).

### 7.2 - Association

Double-cliquer par le **bouton gauche de la souris** sur le **module** désiré

### 7.3 - Modification d'une association déjà existante

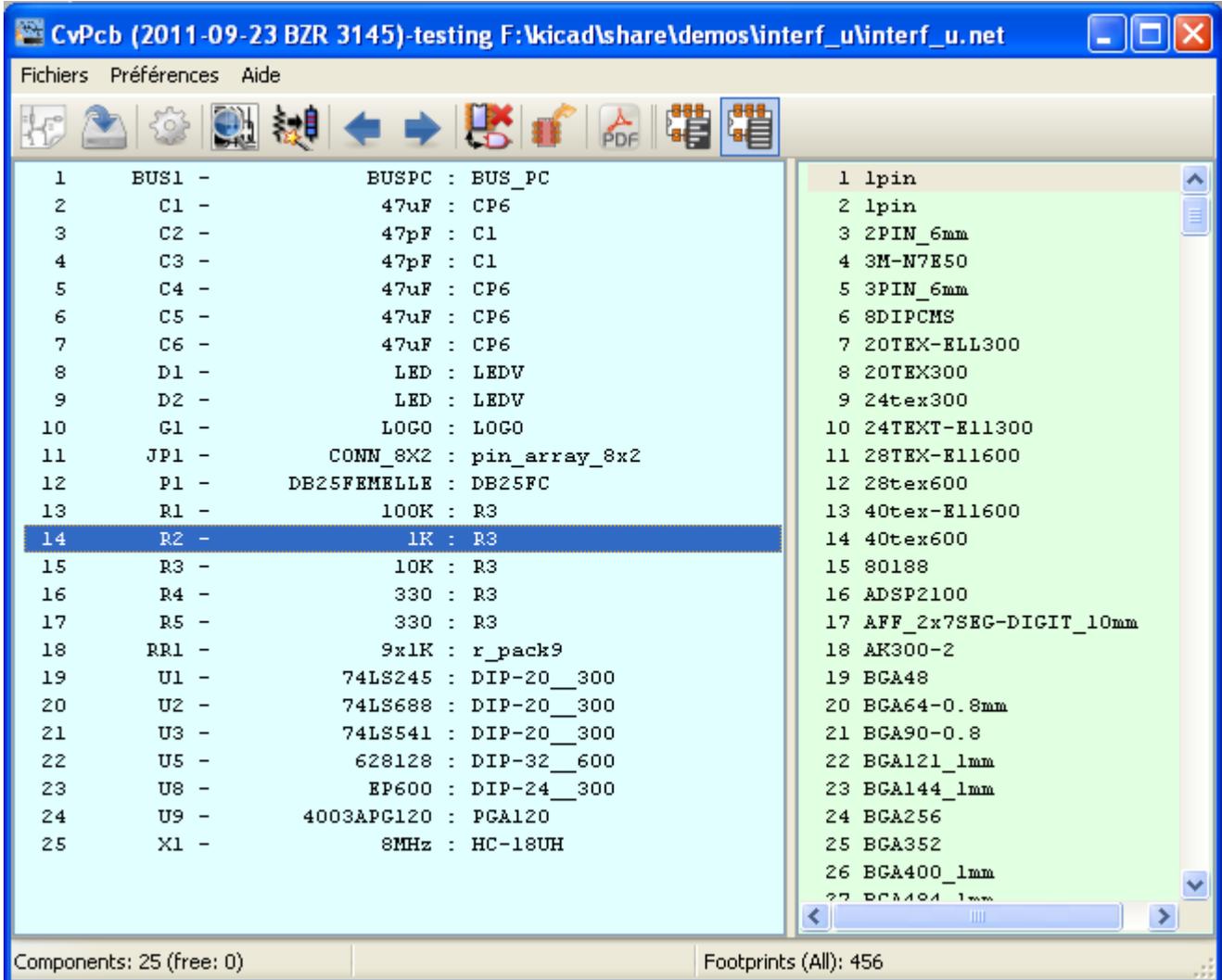
Se fait comme pour une association nouvelle :

Double-cliquer par le **bouton gauche de la souris** sur le nouveau **module** désiré

## 7.4 - Filtrage de l'affichage des modules:

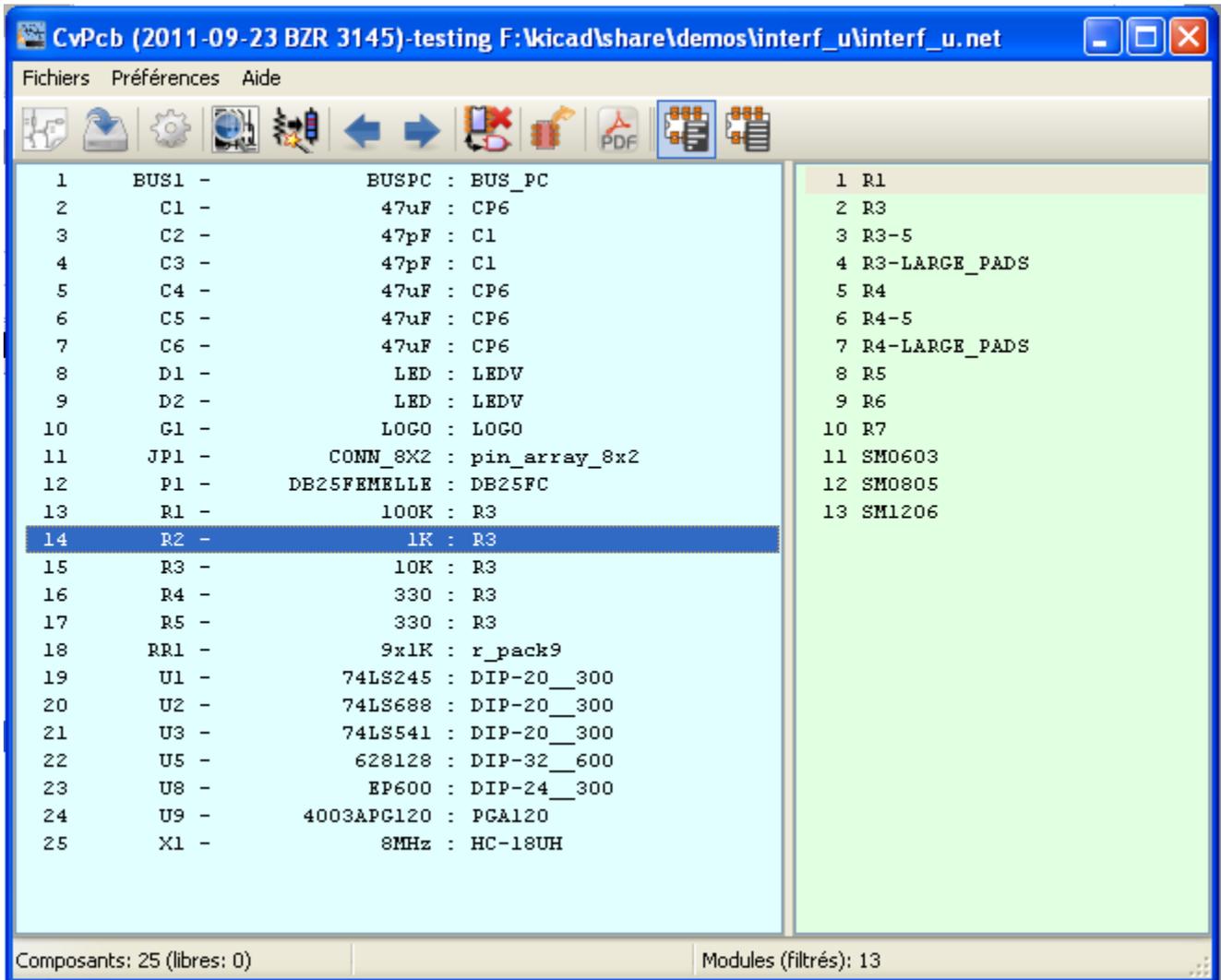
Si le composant sélectionné possède une liste de modules autorisés, l'affichage des modules est filtré selon cette liste.

Affichage non filtré:



## CvPcb.

### Affichage filtré



The screenshot shows the CvPcb software interface. The title bar reads "CvPcb (2011-09-23 BZR 3145)-testing F:\kicad\share\demos\interf\_u\interf\_u.net". The menu bar includes "Fichiers", "Préférences", and "Aide". The toolbar contains various icons for file operations, navigation, and printing. The main window is divided into two panes. The left pane displays a list of 25 components, with the 14th component, "R2 - 1K : R3", highlighted in blue. The right pane displays a filtered list of 13 components, with the 1st component, "R1", highlighted in green. The status bar at the bottom indicates "Composants: 25 (libres: 0)" and "Modules (filtrés): 13".

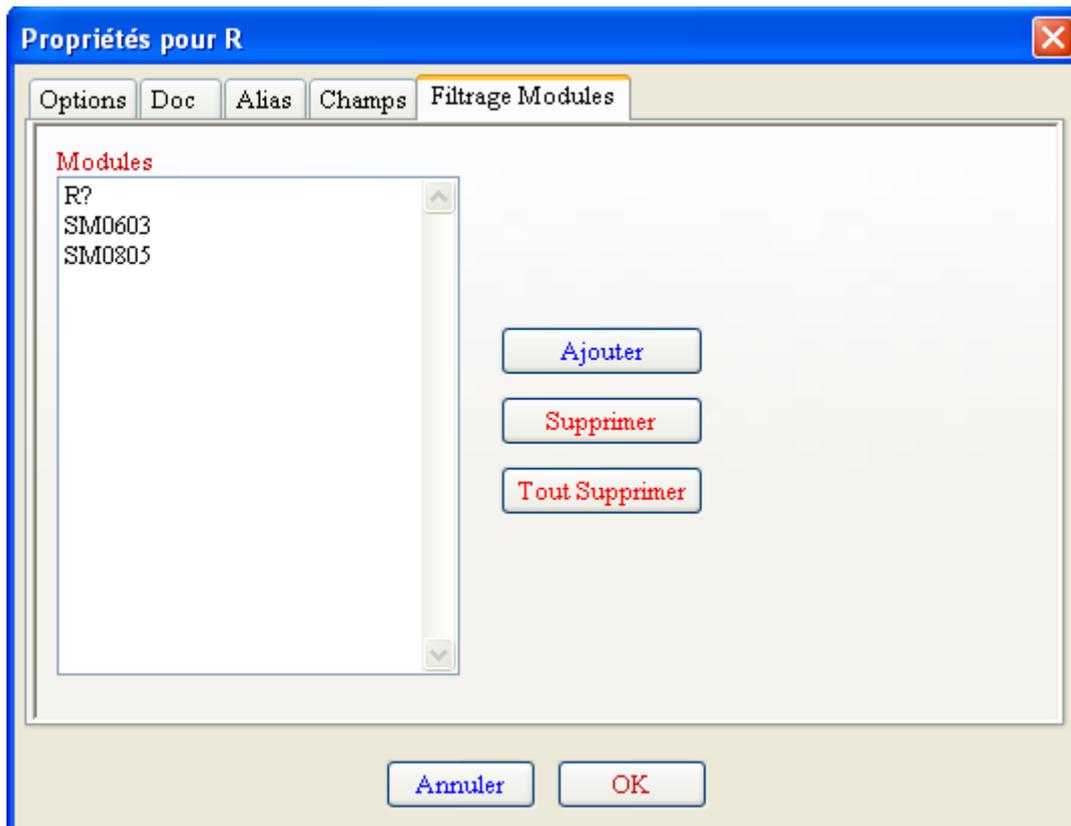
Index	Component	Value	Footprint
1	BUS1	-	BUSPC : BUS_PC
2	C1	-	47uF : CP6
3	C2	-	47pF : C1
4	C3	-	47pF : C1
5	C4	-	47uF : CP6
6	C5	-	47uF : CP6
7	C6	-	47uF : CP6
8	D1	-	LED : LEDV
9	D2	-	LED : LEDV
10	G1	-	LOGO : LOGO
11	JP1	-	CONN_8X2 : pin_array_8x2
12	P1	-	DB25FEMELLE : DB25FC
13	R1	-	100K : R3
14	R2	-	1K : R3
15	R3	-	10K : R3
16	R4	-	330 : R3
17	R5	-	330 : R3
18	RR1	-	9x1K : r_pack9
19	U1	-	74LS245 : DIP-20_300
20	U2	-	74LS688 : DIP-20_300
21	U3	-	74LS541 : DIP-20_300
22	U5	-	628128 : DIP-32_600
23	U8	-	EP600 : DIP-24_300
24	U9	-	4003APG120 : PGA120
25	X1	-	8MHz : HC-18UH

Index	Component
1	R1
2	R3
3	R3-5
4	R3-LARGE_PADS
5	R4
6	R4-5
7	R4-LARGE_PADS
8	R5
9	R6
10	R7
11	SM0603
12	SM0805
13	SM1206

## CvPcb.

Sous Eeschema, la liste des modules autorisés avait été initialisée comme ceci:



Le filtrage est automatique. L'icone  permet d'autoriser ou interdire ce filtrage. Lorsque le filtrage est interdit, la liste complète des modules est affichée.

## 8 - Associations automatiques

### 8.1 - Fichiers d'équivalence

Ces fichiers permettent l'association automatique.

Ils donnent le nom du module adéquat en fonction du nom (valeur) du composant.

Leur extension standard est **.equ**

Il est ainsi très facile avec un jeu de fichiers \*.equ de sélectionner les empreintes, pour un projet donné, selon que l'on utilise des composants standards, ou cms ... en sélectionnant le ou les fichiers \*.stf adéquats.

Voir « **Sélection des fichiers d'équivalence** »

### 8.2 - Format

Ils sont constitués de une ligne par composant.

Chaque ligne a la structure suivante :

**'valeur composant' 'nom module'**

Chaque nom étant encadré par la lettre ',et les 2 noms sont séparés par un ou plusieurs espaces.

*Exemple:*

Si le composant U3 est le circuit 14011 et son module est 14DIP300, la ligne est :

## CvPcb.

'14011' '14DIP300'

Une ligne commençant par # est un commentaire.

Voici un exemple de fichier:

```
#integrated circuits (smd):  
'74LV14' 'SO14E'  
'74HCT541M' 'SO20L'  
'EL7242C' 'SO8E'  
'DS1302N' 'SO8E'  
'XRC3064' 'VQFP44'  
'LM324N' 'SO14E'  
'LT3430' 'SSOP17'  
'LM358' 'SO8E'  
'LTC1878' 'MSOP8'  
'24LC512I/SM' 'SO8E'  
'LM2903M' 'SO8E'  
'LT1129_SO8' 'SO8E'  
'LT1129CS8-3.3' 'SO8E'  
'LT1129CS8' 'SO8E'  
'LM358M' 'SO8E'  
'TL7702BID' 'SO8E'  
'TL7702BCD' 'SO8E'  
'U2270B' 'SO16E'  
#Xilinx  
'XC3S400PQ208' 'PQFP208'  
'XCR3128-VQ100' 'VQFP100'  
'XCF08P' 'BGA48'  
  
#upro  
'MCF5213-LQFP100' 'VQFP100'  
  
#regulators  
'LP2985LV' 'SOT23-5'
```

### 8.3 - Association:

Pour associer automatiquement les composants avec les empreintes, cliquer sur:



Tous les composants dont les valeurs sont trouvées dans un fichier d'équivalence auront les empreinte automatiquement sélectionnée.

**CvPcb.**