



*Eeschema*  
*Manuel de référence*

## Copyright

Ce document est Copyright © 2010–2011 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.

## Retours

Commentaires ou suggestions relatifs à ce document document sont à communiquer à

kicad mailing list: <https://launchpad.net/~kicad-developers>

## Date de publication et version

Version du 07 octobre 2011. Utilisant LibreOffice 3.4.

## Table des matières

1 - Présentation.....	7
1.1 - Description.....	7
1.2 - Caractéristiques techniques principales.....	7
2 - Commandes générales.....	8
2.1 - Accès aux commandes.....	8
2.2 - Commandes à la SOURIS.....	9
2.2.1 - Commandes de base.....	9
2.2.2 - Opérations sur blocs.....	9
2.3 - Commandes rapides (hot keys).....	10
2.4 - Sélection du pas de grille.....	12
2.5 - Réglage du ZOOM.....	12
2.6 - Affichage des coordonnées du curseur.....	12
2.7 - Barre des Menus.....	13
2.8 - Commandes par icônes du toolbar de haut d'écran.....	13
2.9 - Commandes par icônes du toolbar de droite d'écran.....	14
2.10 - Commandes par icônes du toolbar de gauche d'écran.....	16
2.11 - Menu « Pop Up » et éditions rapides d'éléments.....	17
3 - Menu principal.....	19
3.1 - Menu Files.....	19
3.2 - Menu Préférences.....	20
3.2.1 - Préférences générales.....	20
3.2.2 - Menu des Hotkeys.....	20
3.2.3 - Menu Préférences/Libs et Dir.....	21
3.2.4 - Menu Préférences/Couleurs.....	22
3.2.5 - Menu Préférences/Options.....	23
3.2.6 - Menu Préférences/Langue.....	24
3.3 - Menu Help.....	24
4 - Outils du Toolbar général.....	24
4.1 - Gestion de la feuille de tracé.....	24
4.2 - Options de l'Editeur de schemas :.....	25
4.2.1 - Options générales :.....	25
4.2.2 - Noms des champs modèles :.....	26
4.3 - Outil de recherche.....	27
4.4 - Outil Netliste.....	27
4.5 - Outil Annotate.....	28
4.6 - Outil E.R.C.....	29
4.6.1 - Panneau principal.....	29
4.6.2 - Panneau Options.....	30
4.7 - Outil Liste du Matériel.....	31
4.8 - Outil import pour rétro-annotation.....	33
5 - Création / Edition de Schémas.....	35
5.1 - Quelques définitions.....	35
5.2 - Considérations générales.....	35
5.3 - La chaîne de développement.....	35
5.4 - Placement et Édition des composants.....	36
5.4.1 - Appel et placement d'un composant.....	36
5.4.2 - Placement d'une alimentation.....	37
5.4.3 - Edition/Modification d'un composant déjà placé.....	37
Modification du composant.....	38
Modification d'un champ du composant.....	38
5.5 - Fils, Bus, Labels, Alimentations.....	38
5.5.1 - Les éléments de base.....	39
5.5.2 - Établissement des connexions (Fils et Labels).....	39
5.5.3 - Établissement des connexions (Bus).....	40
Membres d'un bus.....	40
Connexions entre membres d'un bus.....	40
Connexions globales entre bus.....	41
5.5.4 - Connexion des alimentations.....	41
5.5.5 - Utilisation des symboles « NoConnect ».....	42

5.6 - Compléments.....	42
5.6.1 - Commentaires.....	42
5.6.2 - Cartouche.....	43
6 - Schémas en Hiérarchies.....	45
6.1 - Présentation.....	45
6.2 - Navigation dans la Hiérarchie.....	45
6.3 - Labels locaux, hiérarchiques et globaux.....	46
6.3.1 - propriétés:.....	46
6.3.2 - Remarques:.....	46
6.4 - Création d'une hiérarchie. Généralités.....	46
6.5 - Placement d'un symbole de hiérarchie.....	47
6.6 - Placement des Connexions : Pins hiérarchiques.....	47
6.7 - Placement des Connexions : Labels Hiérarchiques.....	48
6.8 - Labels, Labels Hiérarchiques, Labels Globaux et Pins Power invisibles.....	49
6.8.1 - Labels simples.....	49
6.8.2 - Labels Hiérarchiques.....	50
6.8.3 - Pins « Power invisibles ».....	50
6.8.4 - Labels Globaux.....	50
6.9 - Hiérarchie Complexe.....	50
6.10 - Hiérarchie "à plat".....	51
7 - Numérotation automatique ( ou Annotation).....	54
7.1 - Rôle.....	54
7.2 - Exemples.....	55
7.2.1 - Sens de Annotation:.....	55
7.2.2 - Choix Annotation:.....	56
8 - Le contrôle E.R.C.....	58
8.1 - Rôle.....	58
8.2 - Utilisation.....	59
8.3 - Exemple de contrôle ERC.....	59
8.4 - Affichage des diagnostics:.....	60
8.5 - Alimentations et Power Flags:.....	61
8.6 - Configuration.....	61
8.7 - Génération du fichier de diagnostic.....	62
9 - Génération de Netlistes.....	64
9.1 - Rôle.....	64
9.2 - Options et exécution (format PCBNEW).....	64
9.3 - Exemple (-format PSPICE).....	65
9.4 - Remarques :.....	67
9.4.1 - Précautions générales.....	67
9.4.2 - Cas de la netliste PSPICE.....	67
9.5 - Autres formats, utilisation de « plugins ».....	68
9.5.1 - Préparation de la fenêtre de dialogue:.....	69
9.5.2 - Format de la ligne de commande:.....	69
9.5.3 - Le convertisseur (plugin).....	70
9.5.4 - Format du fichier netliste intermédiaire:.....	70
10 - Traçage et Impression.....	71
10.1 - Généralités.....	71
10.2 - Commandes communes:.....	71
10.3 - Tracer / Tracé HPGL).....	71
10.3.1 - Sélection de la dimension de la feuille de tracé.....	72
10.3.2 - Réglage des offsets de tracé.....	72
10.4 - Tracer / Tracé Postscript.....	72
10.5 - Tracer / Tracé SVG:.....	73
10.6 - Tracer / Tracé DXF.....	74
10.7 - Imprimer.....	75
11 - Gestion des Composants - Utilisation de LibEdit.....	76
11.1 - Généralités sur les librairies.....	76
11.1.1 - Librairies :.....	76
11.1.2 - Les menus de gestion.....	77
11.2 - Généralités sur les composants.....	77
11.3 - Accès aux composants pour édition.....	77
11.3.1 - Toolbar principal.....	78
11.3.2 - Sélection et maintenance d'une librairie.....	79

11.3.3 - Sélection et sauvegarde d'un composant.....	80
Sélection.....	80
Sauvegarde.....	80
Transfert dans une autre librairie.....	81
Annulation d'une édition de composant.....	81
11.4 - Création d'un composant.....	81
11.4.1 - Création d'un nouveau composant.....	81
11.4.2 - Création à partir d'un autre composant.....	83
11.4.3 - Édition des caractéristiques générales.....	84
11.4.4 - Sélection de l'unité et de la représentation.....	85
11.5 - Édition du graphisme.....	85
11.5.1 - Options d'appartenance des éléments graphiques.....	86
11.5.2 - Les éléments graphiques géométriques.....	87
11.5.3 - Les éléments graphiques type textes.....	87
11.6 - Création et édition des pins.....	87
11.6.1 - Notions générales sur les pins.....	88
11.6.2 - Boîtiers à parts multiples et double représentation.....	88
11.6.3 - Pins : option de base.....	89
11.6.4 - Pins : Définition des caractéristiques.....	89
11.6.5 - Formes des pins.....	90
11.6.6 - Type électrique des pins.....	90
11.6.7 - Modifications globales des pins.....	90
11.6.8 - Pins pour parts multiples et doubles représentations.....	90
11.7 - Édition des champs.....	91
11.8 - Création des symboles d'alimentation.....	93
12 - LibEdit : Compléments.....	95
12.1 - Généralités.....	95
12.2 - Positionnement de l'ancre.....	95
12.3 - Alias.....	96
12.4 - Champs:.....	97
12.5 - Documentation des composants).....	98
12.5.1 - Mots clés (Keywords).....	99
12.5.2 - Documentation du composant (Doc).....	99
12.5.3 - Fichier associé de documentation (DocFileName).....	99
12.5.4 - Filtrage des Modules pour CvPcb.....	99
12.6 - « Librairie » de symboles.....	101
12.6.1 - Exportation/Création de symboles.....	101
12.6.2 - Importation d'un symbole.....	101
13 - Viewlib.....	102
13.1 - Rôle.....	102
13.2 - Écran général.....	102
13.3 - Toolbar de Viewlib.....	103
14 - Customize Netlists and BOM.....	105
14.1 - The Intermediate Netlist:.....	105
14.1.1 - Schematic sample:.....	106
14.1.2 - The Intermediate Netlist file sample:.....	106
14.2 - Conversion to a new format:.....	109
14.3 - XSLT approach:.....	109
14.3.1 - Sample to create a Pads-Pcb netlist file.....	109
14.3.2 - Sample to create a Cadstar netlist file.....	111
The Style-Sheet file:.....	111
The output file:.....	113
14.3.3 - Sample to create a OrcadPCB2 netlist file.....	113
The Style-Sheet file:.....	113
The output file:.....	117
14.3.4 - Using Eeschema plugins interface:.....	117
Init the Dialog window:.....	117
Parameters:.....	119
The command line:.....	119
Command line format:.....	119
14.4 - The Intermediate Netlist structure:.....	119
14.4.1 - General structure:.....	120
14.4.2 - The header section:.....	120

## Eeschema -

14.4.3 - The components section:.....	121
Note about time stamps for components:.....	121
14.4.4 - The libparts section:.....	121
14.4.5 - The libraries section:.....	122
14.4.6 - The nets section:.....	122
14.5 - More info about xsltproc:.....	123
14.5.1 - Introduction.....	123
14.5.2 - Synopsis.....	123
14.5.3 - Command Line Options.....	123
14.5.4 - Return values.....	124
14.5.5 - More Information.....	124

## 1 - Présentation

# Table des matières

1 - Présentation.....	1
1.1 - Description.....	1
1.2 - Caractéristiques techniques principales.....	1

## 1.1 - Description

**Eeschema** est un puissant logiciel de schématisation de circuits électroniques disponible sous les systèmes d'exploitation

- LINUX
- Windows XP/2000/ W7

Quel que soit le système utilisé, les fichiers générés sont totalement compatibles d'un système à l'autre.

**Eeschema** est un logiciel « intégré » car toutes les fonctions de dessin, de contrôle, de tracé, de gestion de bibliothèques et d'accès au logiciel de circuit imprimé sont exécutées depuis Eeschema, sans sortir de celui-ci.

Il permet la réalisation de dessins sous forme **hiérarchique**, pour la gestion de **schémas multi-feuilles**.

Eeschema supporte les hiérarchies :

- A plat.
- Simples.
- Complexes.

Il est destiné à travailler associé à un logiciel de réalisation de circuits imprimés tel que **PCBNEW**, à qui il fournira le fichier **Netliste** décrivant le schéma de la carte de circuit imprimé à réaliser.

Eeschema intègre également un éditeur de composants qui permet la création et l'édition de ces composants, leur visualisation et la manipulation des bibliothèques de composants( Import, export, ajout, effacement de composants dans les bibliothèques).

Eeschema intègre aussi toutes les fonctions annexes (mais pourtant indispensables ) à un logiciel de schématisation électronique moderne:

- Contrôle des règles électriques (**D.R.C.**)pour la détection automatique des connexions incorrectes, des entrées « en l'air » de composants...
- Génération des fichiers de tracé au format POSTSCRIPT ou HPGL
- Génération des fichiers de tracé sur imprimante locale.
- Génération de la liste du matériel.
- Génération du fichier **Netliste** pour le logiciel de circuits imprimés, ou pour un simulateur.

## 1.2 - Caractéristiques techniques principales

Ce logiciel fonctionnant en mode 32bits, sa capacité en traitement de circuits n'est limitée que par la capacité mémoire disponible.

Il n'y a pas donc de limitation réelle dans le nombre de composants, de pins par composants, de connexions, de feuilles...

Eeschema travaille sur des schémas simple feuille ou multi feuilles.

Dans le cas de schémas multi feuilles, la représentation est alors dite « en hiérarchie », et l'accès à chaque feuille est alors immédiat.

Eeschema pour les schémas multi feuilles supporte :

- Les hiérarchies simples ( chaque fichier ou sous-schéma est utilisé une seule fois)
- Les hiérarchies complexes ( un fichier ou schéma est utilisé plusieurs fois: multiples instances)
- Les hiérarchies "à plat" ( plusieurs schémas sans connexions par une feuille maître)

La taille maximum des dessins est ajustable à tout moment du format A4 à A0 et du format A à E .

## 2 - Commandes générales

### Table des matières

2 - Commandes générales.....	1
2.1 - Accès aux commandes.....	1
2.2 - Commandes à la SOURIS.....	2
2.2.1 - Commandes de base.....	2
2.2.2 - Opérations sur blocs.....	2
2.3 - Commandes rapides (hot keys).....	3
2.4 - Sélection du pas de grille.....	5
2.5 - Réglage du ZOOM.....	5
2.6 - Affichage des coordonnées du curseur.....	5
2.7 - Barre des Menus.....	6
2.8 - Commandes par icônes du toolbar de haut d'écran.....	6
2.9 - Commandes par icônes du toolbar de droite d'écran.....	7
2.10 - Commandes par icônes du toolbar de gauche d'écran.....	9
2.11 - Menu « Pop Up » et éditions rapides d'éléments.....	10

#### 2.1 - Accès aux commandes

On accède aux différentes commandes par:

- Action sur la barre des menus ( haut d'écran).
- Action sur les icônes de haut d'écran (commandes générales)
- Action sur les icônes sur la droite de l'écran (commandes particulières ou « **outils** »)
- Action sur les icônes sur la gauche de l'écran (Options d'affichage)
- Action sur les boutons de la souris (importantes commandes complémentaires); En particulier le bouton de droite permet de faire apparaître un menu « Pop Up » dont le contenu dépend de l'élément sous le curseur (Zoom, grille et édition des éléments).
- Touches de fonction du clavier (**F1**, **F2**, **F3**, **F4**, touche **Inser** et barre « **espace** »).

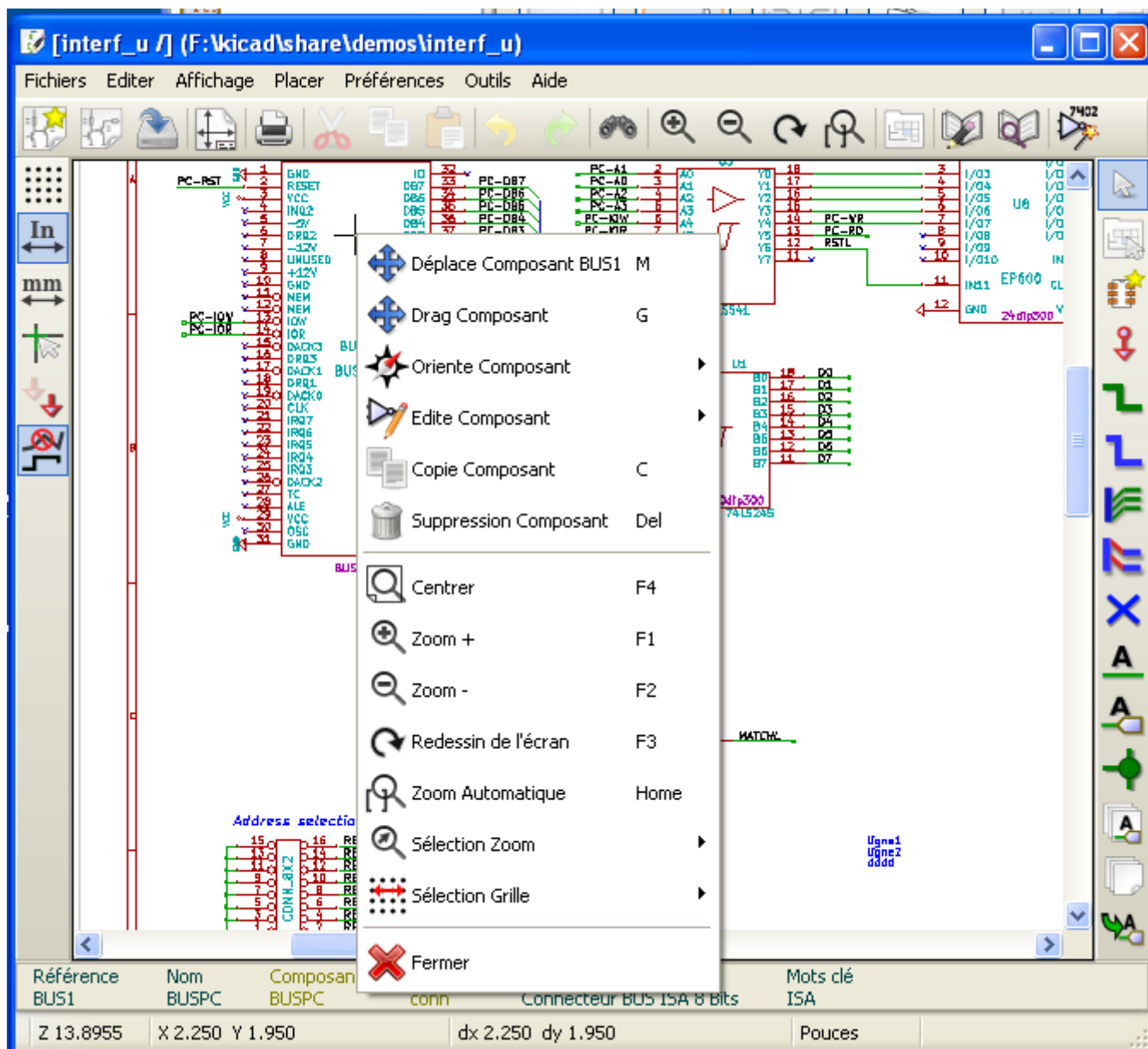
**En particulier:**

- La touche « **Escape** » (ou « **Echap** ») permet souvent d'annuler une commande en cours.
- La touche **Inser** permet de dupliquer le dernier élément créé.



## Eeschema - Commandes générales

Voici les différents accès possibles aux commandes.



## 2.2 - Commandes à la SOURIS

### 2.2.1 - Commandes de base

**Bouton de gauche :**

- ◆ Simple clic: affichage des caractéristiques du composant ou texte sous le curseur souris.
- ◆ Double clic: édition (si l'élément est éditable) de ce composant ou texte.

**Bouton de droite :**

- ◆ Affichage d'un menu Pop Up..

### 2.2.2 - Opérations sur blocs

Les commandes de déplacement, déplacement en mode « drag », copie, et effacement de blocs sont possibles dans tous les menus schématiques.

Le cadre du bloc est tracé en maintenant appuyé le bouton gauche de la souris.

La commande sera exécutée au relâchement du bouton.

- En maintenant appuyée l'une des touches « Shift », « Ctrl », ou les 2 touches « Shift et Ctrl », au moment où l'on appuie sur le bouton droit de la souris, la commande copie, déplacement en mode « drag », effacement

## Eeschema - Commandes générales

sera sélectionnée.

- On peut aussi sélectionner la commande après avoir déterminé le cadre du bloc.

Commandes résumées :

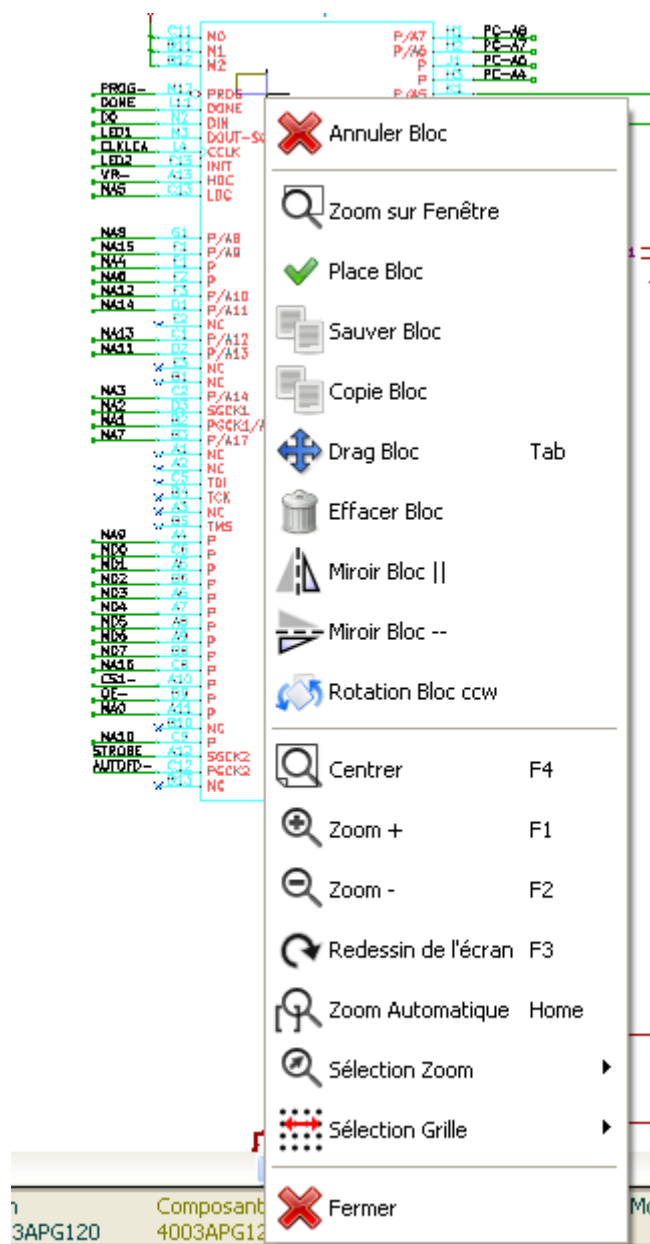
bouton gauche + souris appuyé	Tracé du cadre pour déplacement de bloc
Shift + bouton gauche souris appuyé	Tracé du cadre pour copie de bloc
Ctrl + bouton gauche souris appuyé	Tracé du cadre pour drag de bloc
Shift+Ctrl + bouton gauche souris appuyé	Tracé du cadre pour effacer le bloc

Relâche du bouton : exécution.

En déplacement :

- Cliquer à nouveau sur le bouton pour placer les éléments.
- Cliquer sur le bouton droit pour annuler.

**Si on a débuté une commande de déplacement de bloc, on peut aussi sélectionner une autre commande sur ce bloc par le menu Pop Up (appelé par le bouton droit de la souris):**

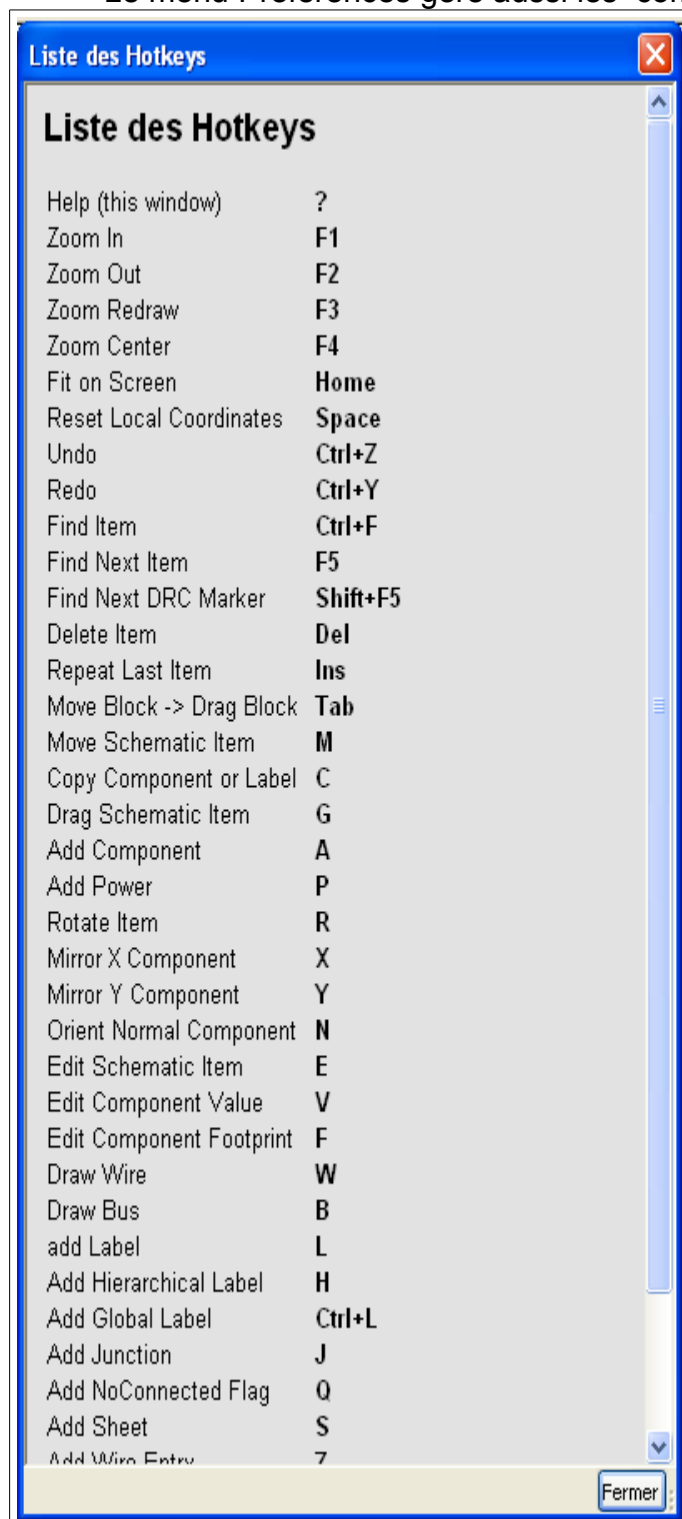


### 2.3 - Commandes rapides (hot keys)

Les majuscules et les minuscules sont indifférenciées.

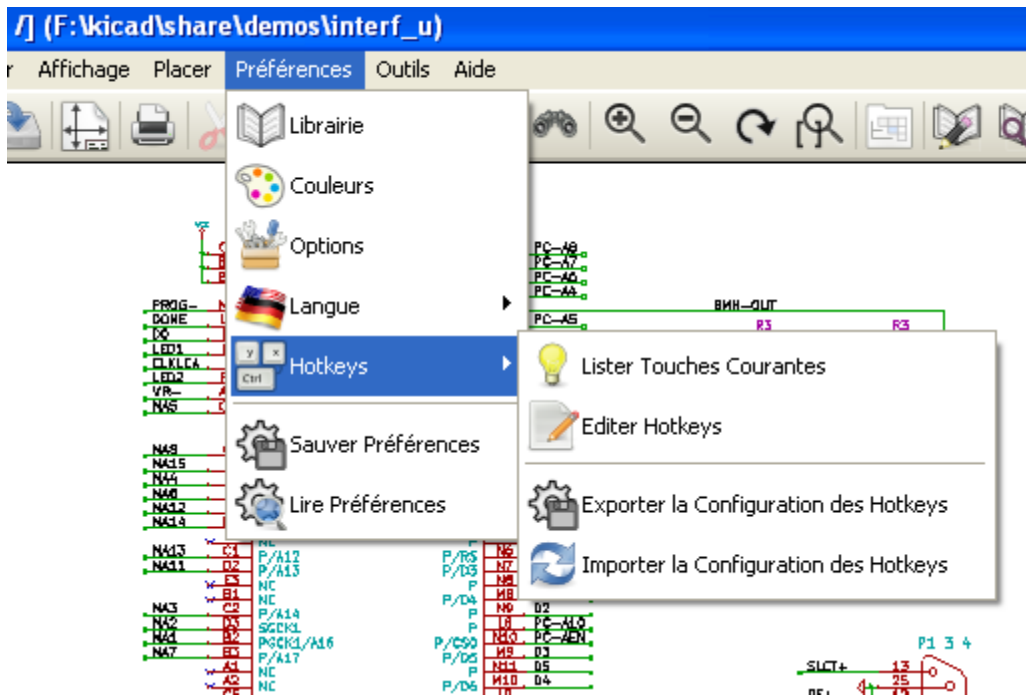
## Eeschema - Commandes générales

- la touche ? permet d'avoir la liste courante des commandes rapides:
- Le menu Préférences gère aussi les commandes rapides.



voici un exemple

Les touches sont programmables.  
Pour cela il y a un éditeur intégré:



## 2.4 - Sélection du pas de grille

Le curseur de tracé se déplace sur une grille, qui peut être affichée ou non (cette grille est toujours affichée dans les menus de gestion des librairies).

Le changement du pas de la grille se fait dans le menu PopUp, ou dans le menu **Preferences/Options** de la barre des menus.

Le pas de la grille est normalement de 50 mils (0.050") ou 1,27 mm.

On peut également travailler sur la grille moyenne (pas = 20 mils) ou fine (pas = 10 mils).

Ceci n'est pas recommandé pour le travail usuel.

Cette grille moyenne ou fine est surtout prévue pour être utilisée dans la création du dessin des composants ou pour la manipulation de composants à très grand nombre de pins (plusieurs centaines) .

## 2.5 - Réglage du ZOOM

Pour changer le "ZOOM" :

- Activer le menu Pop Up (bouton de droite de la souris) et sélectionner le zoom voulu (ou le pas de grille voulu).
  - Ou utiliser les touches de fonction :
    - ◆ **F1** : Grossissement
    - ◆ **F2** : Réduction
    - ◆ **F3** : Rafraîchissement de l'affichage
    - ◆ **F4** : Recentrage autour du curseur.
- Egalement: Simple clic sur le bouton milieu de la souris (sans déplacement)
- ◆ **Zoom sur fenêtre**: « Drag » de la souris, avec le bouton milieu maintenu appuyé.
  - ◆ **Roulette de la souris**: Grossissement/Réduction
  - ◆ **SHIFT+Roulette de la souris**: Panning vertical
  - ◆ **CTRL+Roulette de la souris**: Panning horizontal

## 2.6 - Affichage des coordonnées du curseur

L'unité d'affichage est le pouce (inch ou ") ou le millimètre.

Cependant, Eeschema, de façon interne, travaille toujours en 1/1000 de pouce.

L'affichage en bas et à droite de l'écran donne :

- Le zoom.
- La position absolue du curseur .

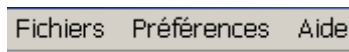
## Eeschema - Commandes générales

- La position relative du curseur.
- Les coordonnées relatives (x,y) peuvent être remises à 0 par la barre d'espace.
- Les coordonnées affichées ensuite seront alors relatives à ce point de remise à 0.



### 2.7 - Barre des Menus















Elle permet l'accès aux lectures et sauvegardes des schémas, aux menus de configuration et à l'aide en ligne












### 2.8 - Commandes par icônes du toolbar de haut d'écran

Ce toolbar permet l'accès aux principales fonctions de Eeschema.




	Création d'un nouveau schéma.
	Ouverture d'un ancien schéma.
	Sauvegarde du schéma complet (avec toutes les feuilles de la hiérarchie).
	Sélection de la taille de la feuille de dessin et <b>modification</b> du contenu du <b>cartouche</b> .
	Accès au menu de gestion de l'impression des schémas.
	Suppression des éléments sélectionnés lors d'un <b>move block</b> .
	Copie des éléments sélectionnés lors d'un <b>move block</b> dans une mémoire de sauvegarde.
	Copie du dernier élément ou bloc effacé ou sauvegardé dans le schéma en cours.
	Annulation du dernier changement (jusqu'à 10 niveaux)
	Restaurer la dernière annulation (jusqu'à 10 niveaux)
	Appel au menu de localisation de composants et textes.
	Zoom plus et Zoom moins, autour du centre d'écran.
	Rafraichissement de l'écran et Zoom optimal.
	Appel au « navigateur » permettant la visualisation de l'arborescence de la hiérarchie du schéma (s'il contient des sous feuilles) et la sélection immédiate de n'importe quel schéma de la hiérarchie.

## Eeschema - Commandes générales

	Appel de l'éditeur de composants <b>Libedit</b> (Examen, modification, édition des composants en librairies).
	Appel du visualisateur de librairies ( <b>Viewlib</b> ).
	Auto numérotation des composants.
	ERC (Electrical Rule Check) : contrôle automatique des connexions.
	Création de la netliste (format Pcbnew, Spice ...).
	Génération de la liste des composants et/ou labels hiérarchiques.
	Appel à <b>CVPCB</b> .
	Appel à <b>PCBNEW</b> .
	Importe un fichier .stf de Cvpcb ( Initialise le champ modules des composants)








### 2.9 - Commandes par icônes du toolbar de droite d'écran
















Ce toolbar permet l'accès aux **outils** pour:

- Le placement de composants, fils et bus, jonctions labels, textes ...
- La navigation dans la hiérarchie pour les schémas multi-feuilles.
- La création des sous-feuilles de la hiérarchie et des symboles de connexion.
- L'effacement de composants.


L'utilisation détaillée de ces outils est décrite dans le chapitre « **Création / Edition de Schémas** »  
 Un aperçu de cette utilisation est donnée ci dessous.

	Arrêt de la commande en cours, annulation de l'outil en cours.
	Navigation dans la hiérarchie: cet outil permet d'entrer dans une sous feuille du schem afficher (cliquer dans le symbole de cette sous feuille), ou de remonter d'un cran dans la hiérarchie (cliquer n'importe où sauf dans un symbole de sous feuille )
	appel au menu de placement d'un composant.
	appel au menu de placement d'alimentations (« <b>powers</b> »).
	Placement de fils de connexion (Wires).
	Placement de Bus.
	Placement de raccords de fils à bus. Ces éléments n'ont qu'un rôle décoratif et n'ont pas de pouvoir de connexion; Ils ne doivent donc pas être utilisés pour des connexions entre deux fils.

## Eeschema - Commandes générales

	Placement de raccords de bus à bus. Ils ne peuvent connecter que deux bus entre eux.
	Placement de symbole de « Non connexion ». Ceux-ci sont à placer sur des pins de composant que l'on ne veut pas connecter. Ceci sert dans la fonction E.RC. Pour vérifier si il est normal d'avoir telle ou telle pin non connectée, ou si c'est un oubli.
	Placement d'un label (label local). Deux fils peuvent être connectés entre eux pas deux labels identiques <b>dans la même feuille</b> . Pour des connexions entre deux feuilles différentes, il faut utiliser des symboles globaux.
	Placement d'un label global. Tous les labels globaux (même sur des feuilles différentes) sont connectés entre eux.
	Placement d'une jonction. A placer sur le point d'intersection de deux fils, ou d'un fil et une pin, lorsque il peut y avoir ambiguïté. (c'est à dire si une extrémité du fil ou de la pin n'est pas raccordée à une des extrémité de l'autre fil).
	Placement d'un label hiérarchique. Ceci permet d'assurer une connexion entre la feuille où l'on place ce label et la feuille racine qui contient le symbole de cette feuille.
	Placement d'un symbole de sous feuille de hiérarchie (rectangle de dimension ajustable). Il faudra préciser le nom du fichier pour la sauvegarde des données de ce « sous schéma ».
	Importation des labels hiérarchiques de la sous feuille pour créer des point de connexion (pins hiérarchiques ) sur un symbole de feuille hiérarchique. Cela suppose que l'on a déjà placé dans cette sous feuille des labels hiérarchiques. Pour ce symbole de hiérarchie, les points de connexion ainsi créés sont équivalents aux pins d'un composant classique, et doivent être connectés par des fils.
	Création de pins hiérarchiques pour la feuille. C'est une fonction analogue à la précédente qui ne nécessite pas d'avoir déjà les symboles globaux définis.
	Tracé de lignes. N'a qu'une valeur décorative, et n'assure aucune connexion.
	Placement de textes de commentaire. N'a qu'une valeur décorative.
	Insertion d'images « bitmap »
	Effacement de l'élément schématique pointé par le curseur Si plusieurs éléments superposés sont pointés, la priorité est donnée au plus petit ( soit dans l'ordre de priorité décroissante jonction, NoConnect, fil, bus, texte, composant). Les feuilles de hiérarchie ne sont pas effaçables par cette commande. Remarque: la fonction « Undelete » du toolbar général permet annulation des derniers effacements.

### 2.10 - Commandes par icônes du toolbar de gauche d'écran

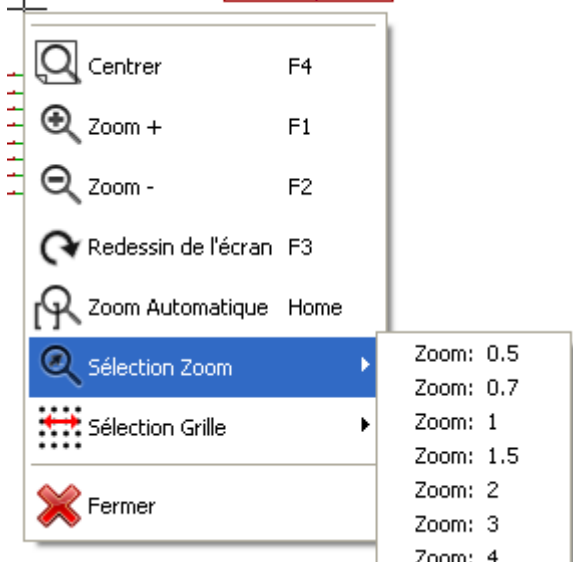
	<p>Ce toolbar gère les options d'affichage:</p> <ul style="list-style-type: none"> <li>De la grille</li> <li>Des unités</li> <li>Du curseur</li> <li>Des pins « invisibles »</li> <li>Et les directions autorisées des fils et bus.</li> </ul>
--	--



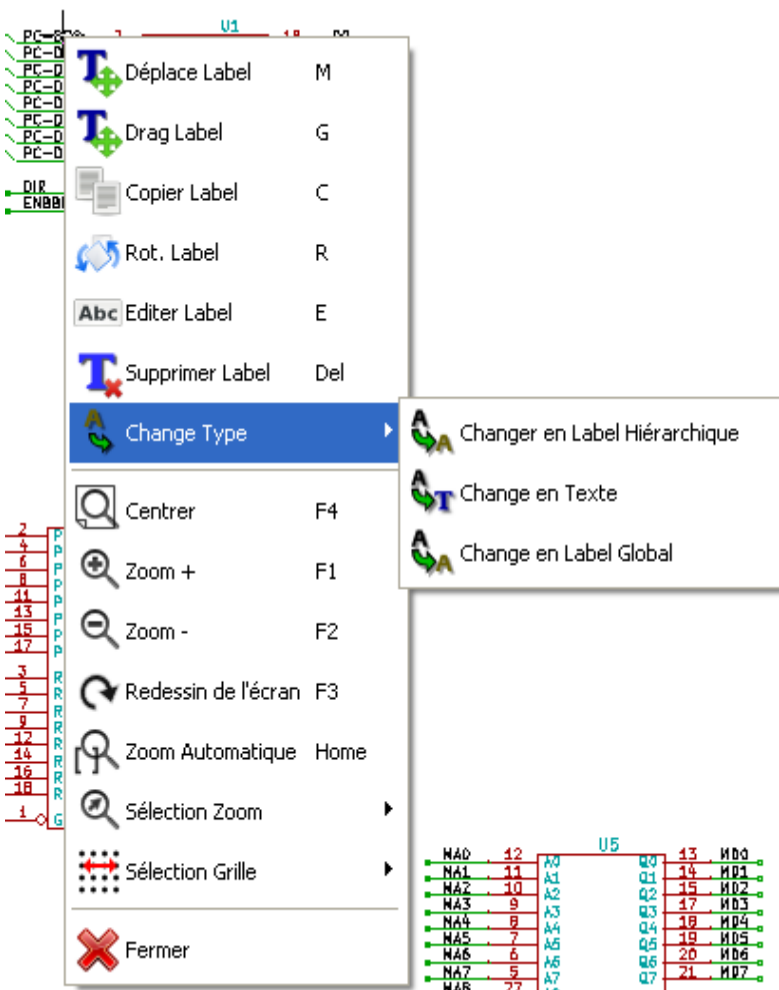
2.11 - Menu « Pop Up » et éditions rapides d'éléments

Un click sur le bouton droit de la souris fait apparaître un menu « Pop Up » dont le contenu dépend de l'élément pointé par le curseur de la souris (s'il y en a un).  
On a ainsi immédiatement accès à:

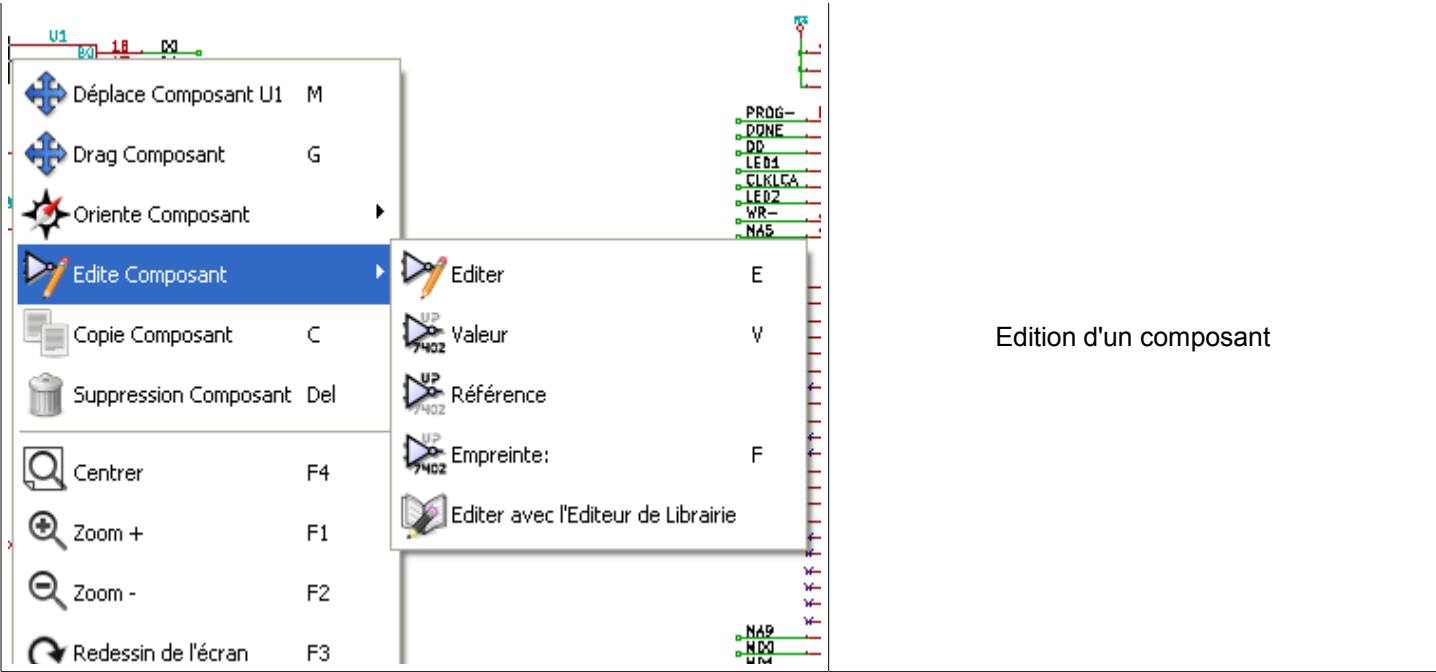
- Au choix du zoom.
- Au réglage de grille.
- Et selon le cas à l'édition des paramètres les plus couramment modifiés de l'élément.



Pop Up sans sélection d'éléments.



Edition d'un label.



Edition d'un composant

3 - Menu principal

Table des matières

3 - Menu principal..... 1

3.1 - Menu Files..... 1

3.2 - Menu Préférences..... 2

3.2.1 - Préférences générales:..... 2

3.2.2 - Menu des Hotkeys..... 3

3.2.3 - Menu Préférences/Libs et Dir..... 3

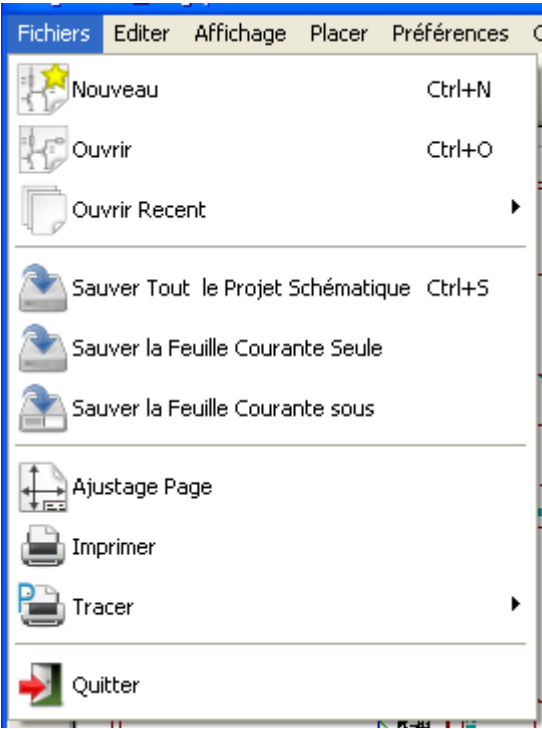
3.2.4 - Menu Préférences/Couleurs..... 4

3.2.5 - Menu Préférences/Options..... 5

3.2.6 - Menu Préférences/Langue..... 6

3.3 - Menu Help..... 6

3.1 - Menu Files



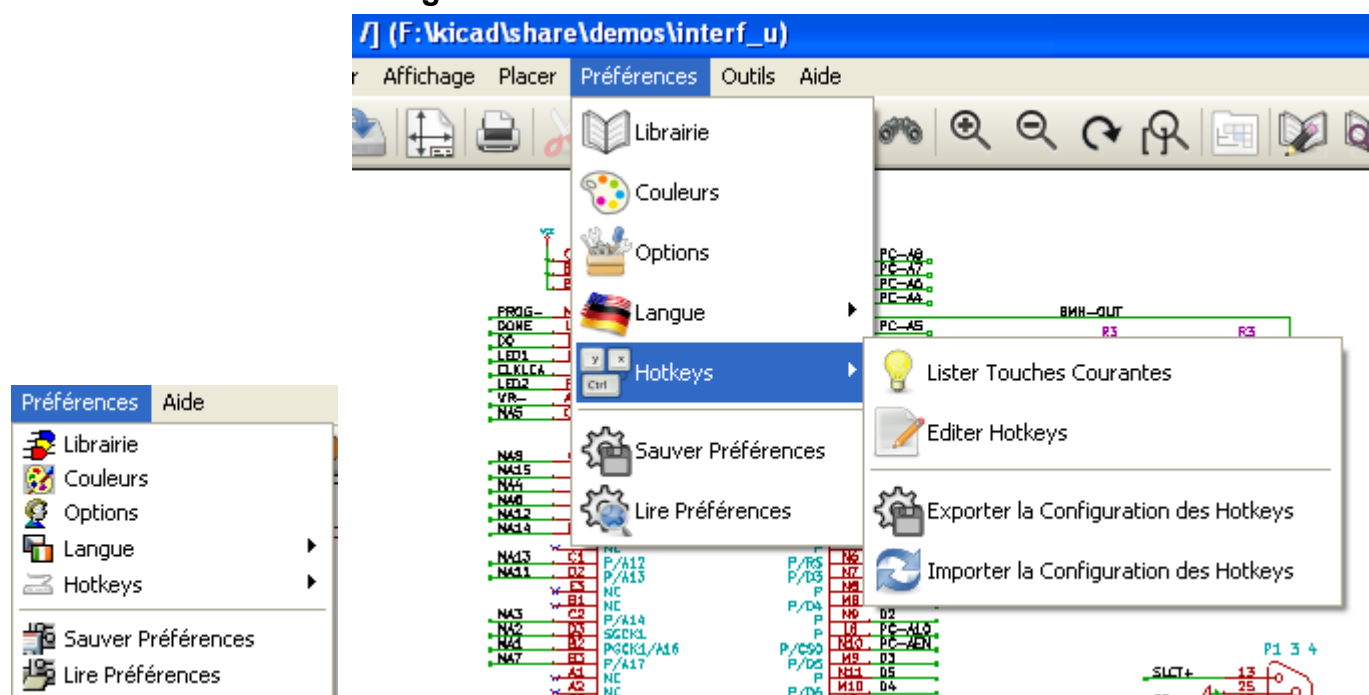
Nouveau	Réinitialiser pour un nouveau projet schématique. l'ancien schéma sera perdu.
Ouvrir	Chargement d'un schéma et de sa hiérarchie
Ouvrir Récent	Accès à la liste des fichiers précédemment chargés.
Sauver Tout le Projet Schématique	Sauvegarde du schéma en cours et toute sa hiérarchie
Sauver la Feuille Courante	Sauvegarde de la feuille de schématique actuellement affichée à l'écran.
Sauver Feuille Courante sous..	Sauvegarde de la feuille de schématique courante sous un nouveau nom.
Imprimer	Accès au menu d'impression des schémas (Voir chapitre « Print et Plot »).

## Eeschema - Menu principal

<b>Tracer</b>	Accès au menu de tracé (format Postscript ou HPGL) (Voir chapitre « Print et Plot »).
<b>Quitter</b>	Sortie de Eeschema (n'assure pas la sauvegarde des données, mais il y aura confirmation de sortie si les schémas ou les librairies ont été modifiée mais non sauvegardées).

## 3.2 - Menu Préférences

### 3.2.1 - Préférences générales:

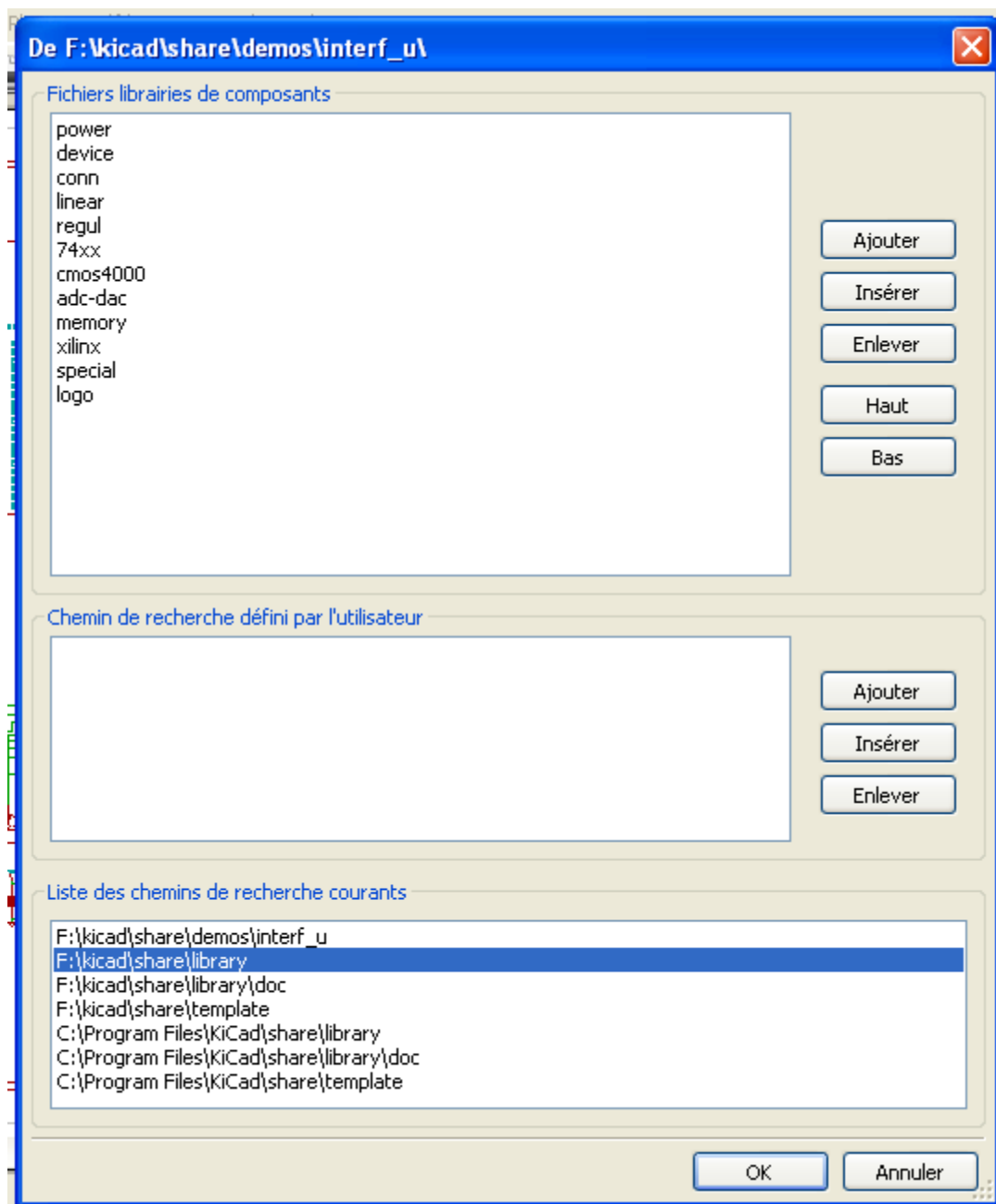


<b>Librairies</b>	Sélection des librairies, de leur chemin par défaut et des extensions des fichiers.
<b>Couleurs</b>	Accès au menu de choix des couleurs.
<b>Options</b>	Sélection de diverses options d'affichage (Unités, Grille..).
<b>Langue</b>	Sélection de la langue pour affichage. L'option défaut est à préférer. Cette option est surtout utile aux traducteurs.
<b>Hotkeys</b>	Accès au menu des hotkeys.
<b>Sauver Préférences</b>	Enregistrement du fichier de configuration
<b>Lire Préférences</b>	Relire un fichier de configuration

### 3.2.2 - Menu des Hotkeys

<b>Lister Touches Courantes</b>	Afficher la liste courante des raccourcis
<b>Editer Hotkeys</b>	Lancement de l'éditeur de hotkeys
<b>Exporter la Configuration des Hotkeys</b>	Exporter un fichier configuration des raccourcis.
<b>Importer la Configuration des Hotkeys</b>	Lire un fichier configuration des raccourcis préalablement exporté par la commande précédente.

### 3.2.3 - Menu Préférences/Libs et Dir



La configuration de Eeschema est essentiellement :

- La définition du **chemin pour les librairies** de composants...
- **La liste des librairies de composants**.

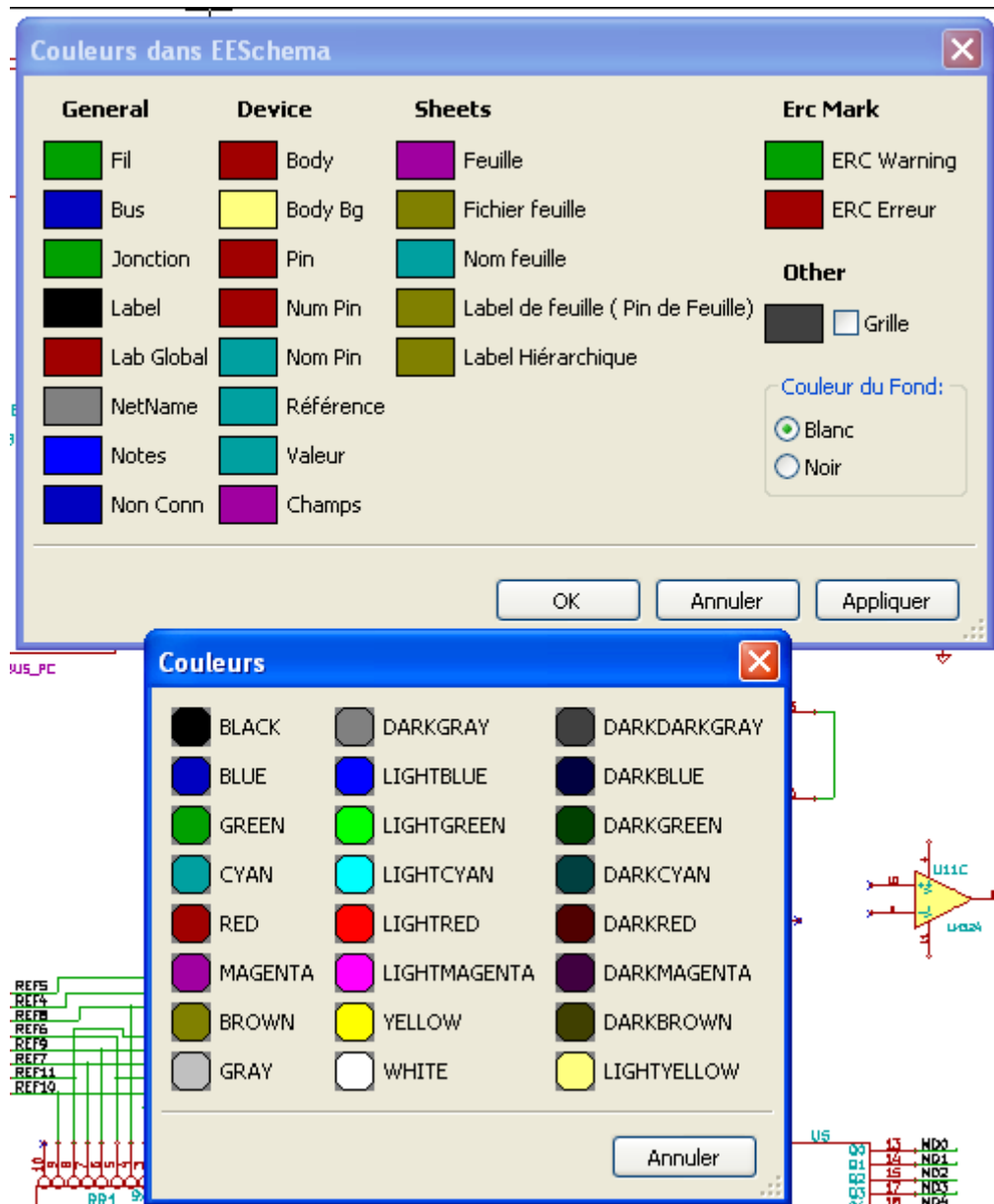
Les paramètres de cette configuration sont mémorisés dans le fichier <projet>.pro).

On peut avoir différents fichiers de configuration, dans différents répertoires de travail.

Eeschema recherche et utilise par ordre de priorité décroissante :

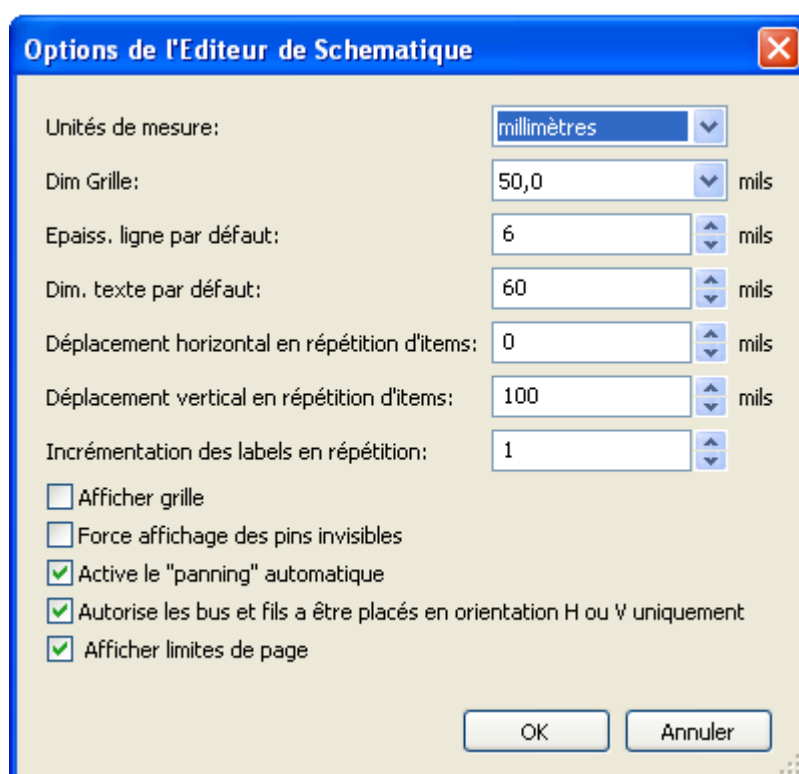
1. Le fichier de configuration **<projet>.pro** dans le répertoire courant.
2. Le fichier de configuration **kicad.pro** dans le répertoire **template** de l'arborescence **kicad**.  
Ce fichier peut donc être une **configuration par défaut**.
3. Des valeurs par défaut si aucun fichier n'est trouvé. Il faudra au moins alors remplir la liste des librairies à charger et sauver la configuration.

3.2.4 - Menu Préférences/Couleurs



Sélection de la couleur d'affichage des divers éléments de dessin, et celle du fond d'écran (blanc ou noir uniquement).

## 3.2.5 - Menu Préférences/Options



<b>Unités</b>	Sélection de l'unité d'affichage des coordonnées curseur (Inches ou Millimètres).
<b>Dim Grille:</b>	Travailler à grille normale (0,050 pouces ou 1,27 mm). Les grilles plus fines sont utiles pour la construction de composants de bibliothèque.
<b>Epaiss.ligne par défaut</b>	épaisseur utilisée pour dessiner des traits des éléments graphiques n'ayant pas une épaisseur spécifiée.
<b>Déplacement horizontal en répétition d'items</b>	valeur du décalage selon l'axe X lors de la duplication des éléments ( valeur usuelle 0 )
<b>Déplacement vertical en répétition d'items</b>	valeur du décalage selon l'axe Y lors de la duplication des éléments ( valeur usuelle 0,100 pouces ou 2,54 mm)
<b>Incrément des labels en répétition</b>	valeur de l'incrément de texte pour duplication de textes terminés par un nombre, tels que membres de Bus ( valeur usuelle 1 ou -1 ).
<b>Afficher Grille</b>	Si actif : Affichage de la grille de travail.
<b>Force affichage des pins invisibles</b>	show alls actif : Affichage des pins normalement invisibles (permet de visualiser les pins d'alimentations).
<b>Active le « panning » automatique</b>	Si actif, recadrage automatique si le curseur sort de la fenêtre, en tracé de fils, ou en déplacement d'éléments.
<b>Autorise les bus et fils a être placés en orientation H ou V uniquement</b>	Actif: On ne peut tracer que des traits horizontaux ou verticaux. inactif : On peut tracer des traits d'inclinaison quelconque
<b>Afficher limites de page</b>	Affiche à l'écran les bords de la page

### 3.2.6 - Menu Préférences/Langue

Permet l'affichage en anglais, français ou autre.

La sélection normale est « Défaut ».

Surtout utile pour les traducteurs.

Il faut relancer Eeschema pour que cela prenne effet.

### 3.3 - Menu Help

Accès à l'aide en ligne (ce document) et au numéro de version de Eeschema.

## 4 - Outils du Toolbar général

## Table des matières

4 - Outils du Toolbar général.....	1
4.1 - Gestion de la feuille de tracé.....	1
4.2 - Options de l'Editeur de schemas :.....	2
4.2.1 - Options générales :.....	2
4.2.2 - Noms des champs modèles :.....	3
4.3 - Outil de recherche.....	3
4.4 - Outil Netliste.....	3
4.5 - Outil Annotate.....	4
4.6 - Outil E.R.C.....	5
4.6.1 - Panneau principal.....	5
4.6.2 - Panneau Options.....	6
4.7 - Outil Liste du Matériel.....	7
4.8 - Outil import pour rétro-annotation.....	9

### 4.1 - Gestion de la feuille de tracé



Permet l'accès au menu:

On peut ainsi sélectionner les dimensions de la feuille de tracé et éditer le contenu du cartouche.



**Ajustage opt Page** [X]

Dim Page:      Number of feuilles: 1      Numéro feuille: 1

☐ Format A4  
☒ Format A3  
☐ Format A2  
☐ Format A1  
☐ Format A0  
☐ Format A  
☐ Format B  
☐ Format C  
☐ Format D  
☐ Format E  
☐ Format libre

Format libre dim X: 17,000

Format libre dim Y: 11,000

**Révision:**  
 2B      ☐ Exporter vers autres feuilles

**Titre:**  
 INTERFACE UNIVERSEL      ☐ Exporter vers autres feuilles

**Société:**  
 KICAD suite GPL      ☐ Exporter vers autres feuilles

**Commentaire1:**  
 Comment 1      ☐ Exporter vers autres feuilles

**Commentaire2:**  
 Comment 2      ☐ Exporter vers autres feuilles

**Commentaire3:**  
 Comment 3      ☐ Exporter vers autres feuilles

**Commentaire4:**  
 Comment 4      ☐ Exporter vers autres feuilles

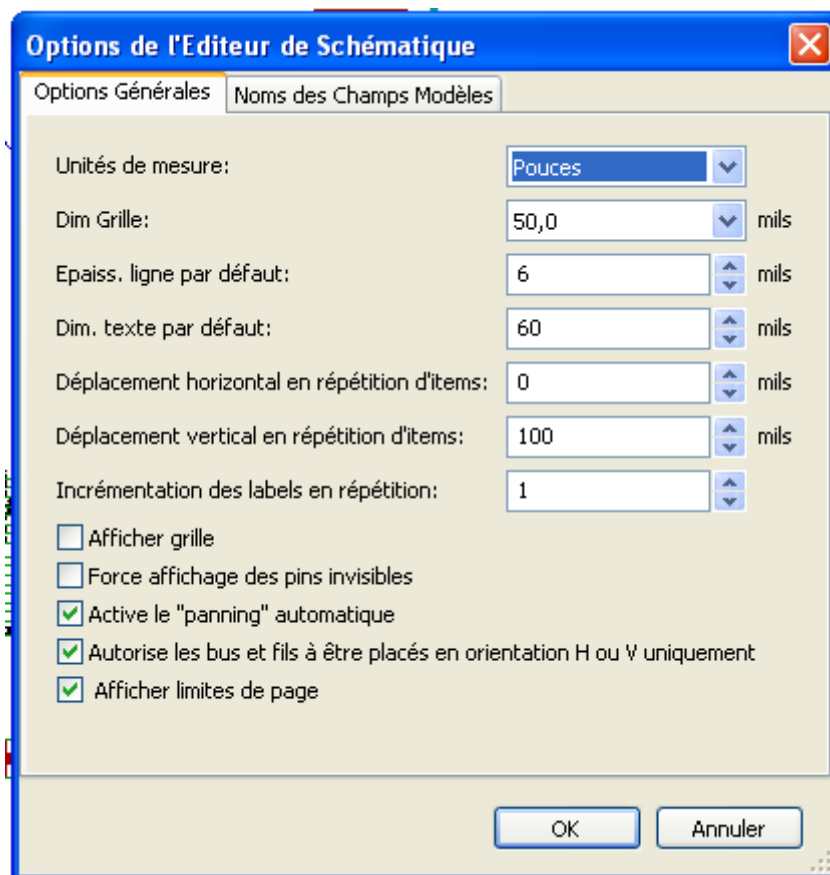
OK Annuler

**Note :**  
 La date affichée dans le cartouche est automatiquement mise à jour.  
 Number of sheets et Sheet number sont automatiquement mis à jour par l'outil de numérotation (annotation) automatique.

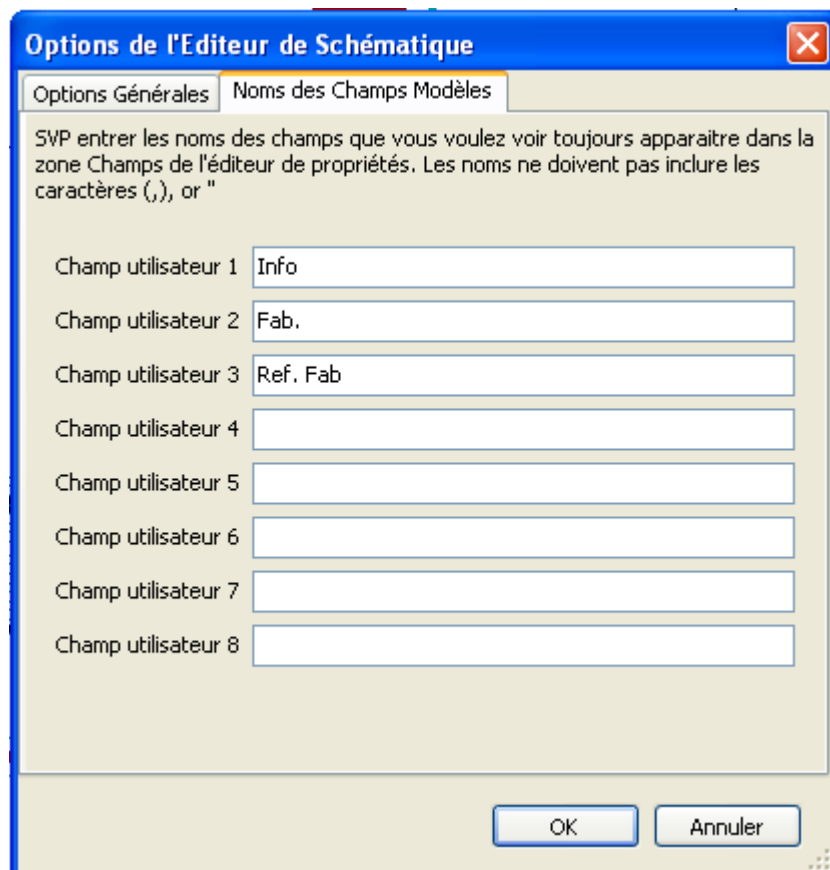
## 4.2 - Options de l'Editeur de schemas :

### 4.2.1 - Options générales :

Permet de définir quelques options générales pour d'affichage et pour la commande de répétition (touche ***Ins***)



#### 4.2.2 - Noms des champs modèles :



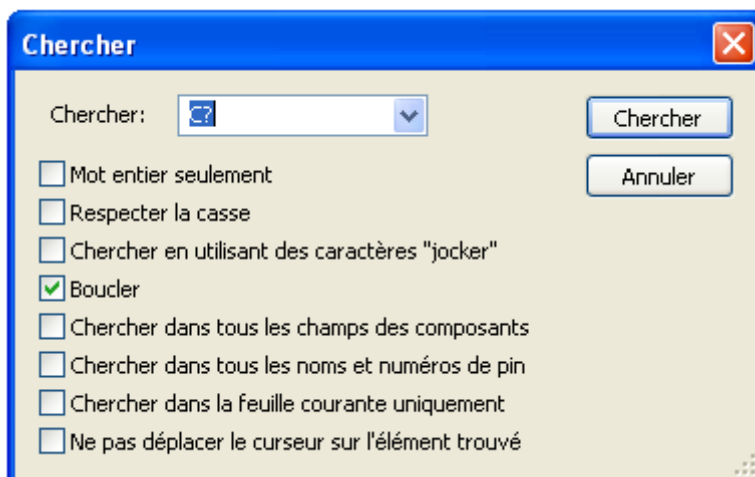
## Eeschema - Outils du Toolbar général

Permet de définir des champs utilisateurs **qui seront toujours présents** pour tous les composants de la schématique, même s'il restent vides pour certains composants.

### 4.3 - Outil de recherche



Permet l'accès à cet outil.



On peut rechercher un composant, une valeur, un texte dans la feuille courante ou dans toute la hiérarchie. Le curseur souris sera positionné (sur option) sur l'élément trouvé, dans la sous feuille concernée.

### 4.4 - Outil Netliste



Permet l'accès à cet outil, qui permet la génération du fichier netliste.

Ce fichier peut être relatif à toute la hiérarchie ( option usuelle ), ou seulement à la feuille courante( la netliste est alors partielle, mais cette option peut être utile pour certains logiciels).

Dans les schémas multifeuilles en hiérarchie, tout label local n'est connu que de la feuille à laquelle il appartient.

Ainsi le label TOTO de la feuille 3 est différent du label TOTO de la feuille 5 ( si l'on n'a pas introduit des connexion hiérarchiques pour les connecter volontairement).

Ceci est dû au fait le numéro de feuille (mis à jour par la commande annotate) est associé au label local. Dans l'exemple précédent, le premier label TOTO est en réalité TOTO\_3, et le deuxième label TOTO est en réalité TOTO\_5.

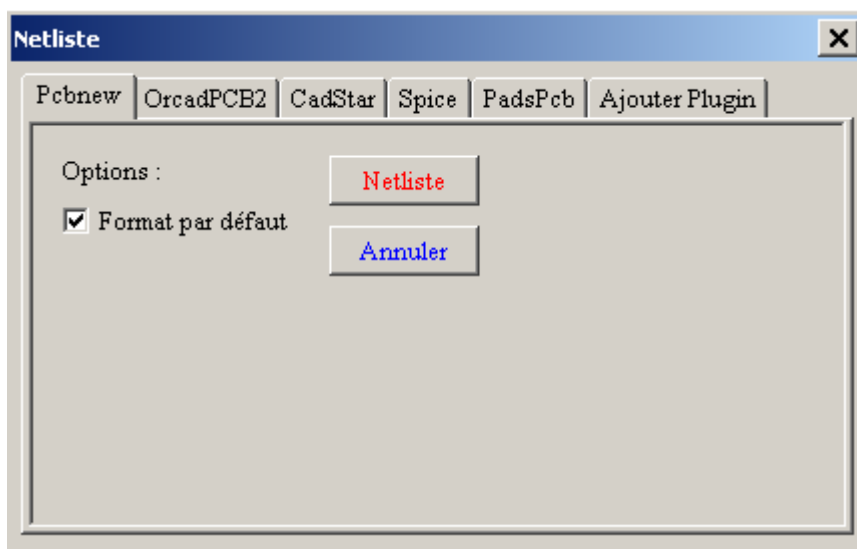
Cette association peut être inhibée si on le désire, mais attention alors à de possibles connexions non souhaitée.

#### **Remarque 1 :**

La longueur des labels n'est pas limitée par EESchema, mais les logiciels exploitant les netlistes générées peuvent être limités sur ce point.

#### **Remarque 2 :**

Éviter les espaces dans les labels, car ils apparaissent alors comme plusieurs mots. Ce n'est pas non plus une limitation de Eeschema, mais des formats de netlistes qui supposent pour la plupart qu'un label est un seul mot.



**Option:**

**Format par défaut:**

Sélection du type de netliste générée (Pcbnew OrcadPcb2, CadStar, Spice) = Pcbnew.

On peut aussi générer une netliste aux formats

- Orcad PCB2
- CadStar
- Spice, pour le simulateur Spice.

## 4.5 - Outil Annotate

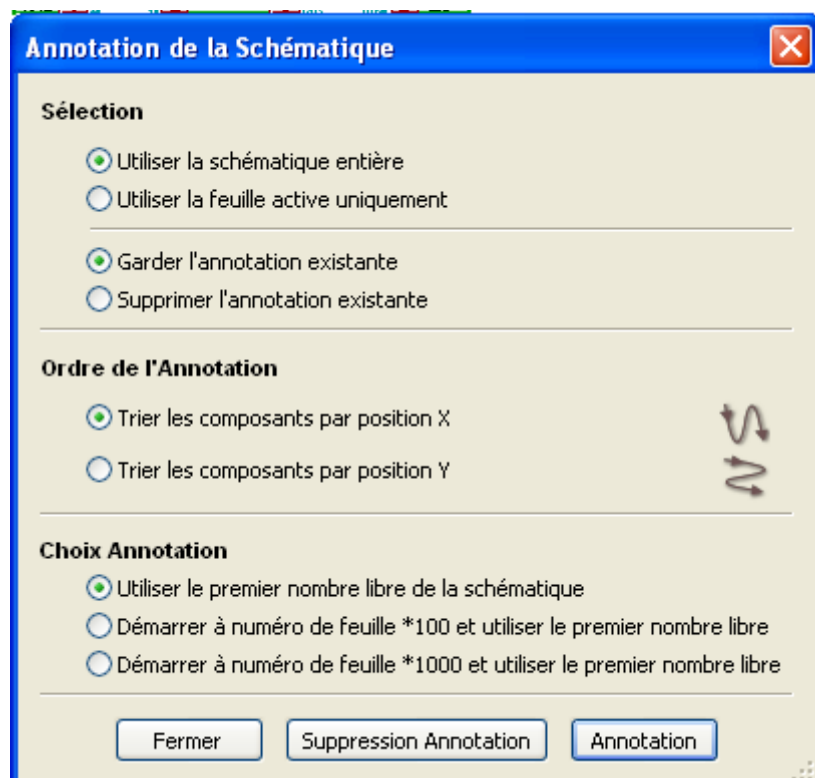


Permet l'accès à cet outil.

Cet outil permet la numérotation automatique des composants.

Pour les composants comportant plusieurs parts par boîtiers ( comme le boîtier 7400 qui contient 4 parts ), il y aura de plus répartition automatique des parts ( ainsi le boîtier 7400 noté U3 sera réparti en U3A, U3B, U3C et U3D ).

On peut annoter inconditionnellement tous les composants, ou seulement les nouveaux composants, c'est à dire ceux qui n'ont pas encore été annotés (ceux qui ont une référence qui finit par ? comme par exemple U?).



## Eeschema - Outils du Toolbar général

### **Sélection:**

#### **Utiliser la schématique entière:**

Toutes les feuilles sont re numérotées (Option usuelle).

#### **Utiliser la feuille active uniquement:**

Seule la feuille courante est renumérotée (Option à n'utiliser que dans des cas spéciaux, par exemple pour évaluer le nombre de résistances dans la feuille courante..)

#### **Garder l'annotation existante:**

Annotation conditionnelle, seuls les nouveaux composants seront renumérotés (option usuelle).

#### **Supprimer l'annotation existante:**

Annotation inconditionnelle, tous les composants seront re numérotés.

#### **Ordre:**

Sélection du critère pour attribuer un numéro d'ordre dans l'annotation.

## **4.6 - Outil E.R.C**



Permet l'accès à cet outil.

Cet outil permet le contrôle électrique du schéma ( test type **E**lectrical **R**ule **C**heck).

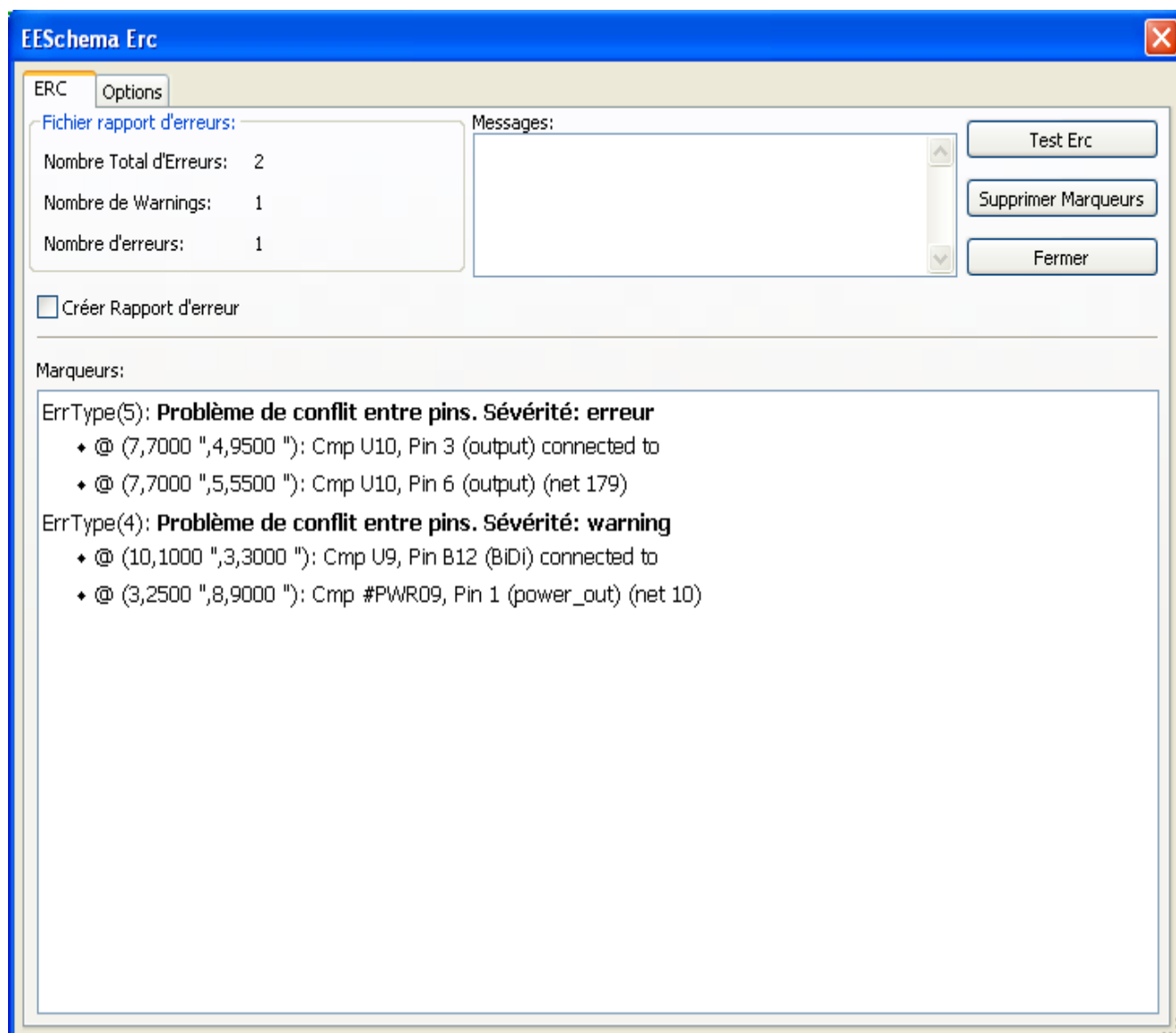
Cette fonction est tout particulièrement utile pour déceler des connexions oubliées, et des incohérences

Eeschema place des marqueurs sur les pins ou les labels pouvant poser problème

Le diagnostic d'erreurs peut être déterminé en cliquant ( bouton gauche de la souris ) sur ces marqueurs.

Un fichier d'erreurs peut aussi être généré.

### **4.6.1 - Panneau principal**



Les erreurs sont totalisées dans la fenêtre **Erc Diags**:

- Total général.
- Nombre d'erreurs type **ERREUR**.
- Nombre d'erreurs type **Warning**.

### Option:

- *Créer rapport d'erreur*: Si coché un fichier de liste des erreurs sera généré à la fin du test ERC.

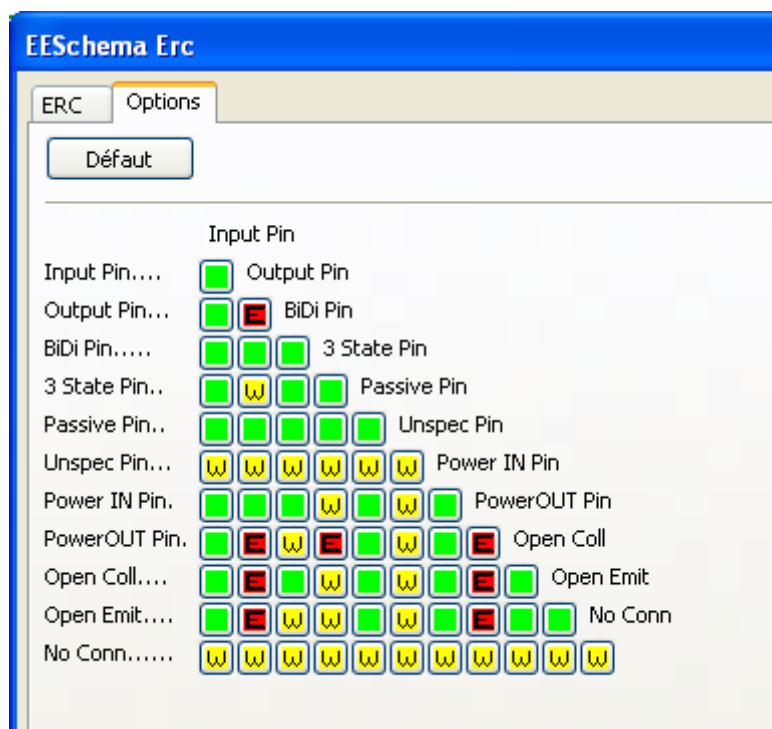
### Commandes:

- *Test ERC*: exécute le test
- *Supprimer Marqueurs*: Effacement des marqueurs E.R.C. L'exécution d'un nouveau test E.R.C efface toujours les anciens marqueurs.
- *Fermer* pour quitter la boîte de dialogue.

### Note:

En cliquant sur un message d'erreur, on saute au marqueur correspondant en schématique.

## 4.6.2 - Panneau Options



Ce panneau permet de régler la matrice des conflits et erreurs détectés.

A chaque type de connexion on peut choisir 1 parmi 3 options:

- Pas d'erreur
- Warning
- Error

Chaque case de la matrice de gestion des conflits peut être modifiée en cliquant sur cette case.

### 4.7 - Outil Liste du Matériel



Ce menu permet la génération rapport, c'est à dire un fichier listing des composants et/ou des connexions hiérarchiques (labels hiérarchiques et globaux).

Les composants peuvent être listés classés par

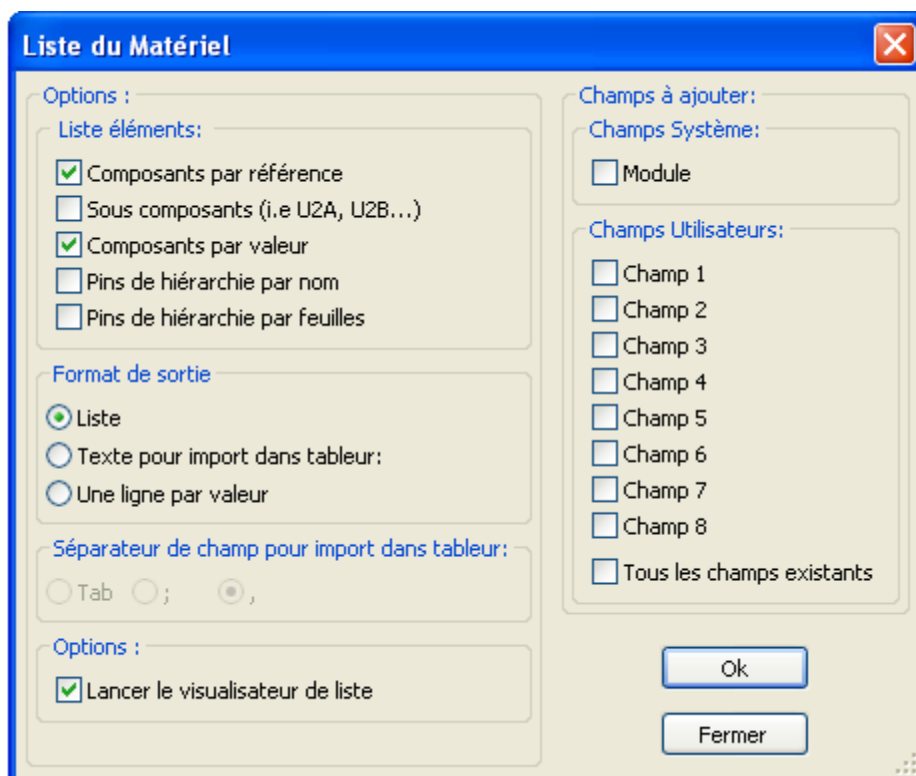
- Référence
- Valeur

et les boîtiers a parts multiples peuvent être listés part à part

Les labels hiérarchiques peuvent être listés classés par

- Classement alphabétique.
- Feuille de hiérarchie.

Les différents classements peuvent être générés simultanément.



Les options sont:

<b>Composants par Référence</b>	Génération de la liste des composant triés par Référence.
<b>Composants par Valeur</b>	Génération de la liste des composant triés par Valeur.
<b>Sous Composants</b>	La liste générée montre des parts par boîtier (ex U2A, U2B ...).
<b>Pins de hiérarchie par nom</b>	Génération de la liste des connexions hiérarchiques triées alphabétiquement.
<b>Pins de hiérarchie par feuilles</b>	Génération de la liste des connexions hiérarchiques triées par numéro de feuilles.
<b>Liste</b>	Génération du fichier sous forme texte imprimable.
<b>Texte pour import dans tableur</b>	Génération du fichier sous forme texte importable facilement dans un tableur, 1 ligne par composant.
<b>Une ligne par valeur</b>	Génération du fichier sous forme texte importable facilement dans un tableur, composants triés et groupés par valeurs, 1 ligne par valeur
<b>Lancer le visualisateur de liste</b>	Lanceur l'éditeur de texte pour visualiser le fichier rapport après création



## Eeschema - Outils du Toolbar général

Un ensemble de renseignements utiles pour utiliser une liste de matériel est:

- Valeur – nom unique pour chaque composant (éviter d'avoir 100 pour 100 Ohms ET 100 pour 100pF)
- Module – Nom de l'empreinte utilisée (entré manuellement ou par rétro-annotation (voir plus loin))
- Field1 – Nom du fabricant
- Field2 – Nom du distributeur ou référence fabricant
- Field3 – Référence commande du distributeur

Par exemple:

Champs	
Référence	X1
Valeur	8MHz
Module	HC-18UH
Documentation	
Info	
Fab.	KONY
Ref. Fab	123-456

L'usage du format « **Une ligne par valeur** » ne requiert ces renseignements que pour un seul composant d'une valeur donnée dans le schéma, et non pour tous les autres ayant cette valeur.

Cependant, s'il existe plusieurs composants ayant la même valeur, mais des empreintes différentes, comme 2 résistances de 33K, une de 1/10 W et une autre de 1/4 W, utiliser 33K pour l'une et 33K/Big et elles seront listées sur 2 lignes différentes.

Le fichier est généré sous forme texte importable facilement dans un tableur, avec lequel les coûts peuvent être calculés.

### 4.8 - Outil import pour rétro-annotation



Importe un fichier **.stf** créé par Cvp pcb pour initialiser le champ modules des composants.

Cette fonction lit le fichier d'association composants/empreintes **.stf** généré par Cvp pcb et copie le nom de l'empreinte dans champ modules du composant correspondant.

Elle ne sert que si on désire initialiser dans Eeschema le champ "modules" après sélection des modules par Cvp pcb, ce qui n'est pas nécessaire pour Pcbnew car Pcbnew utilise pour cela le fichier **.cmp** généré par Cvp pcb, et associé au fichier netliste **.net** correspondant, mais peut être utile pour générer les rapport et listes de matériel.

#### **Note pour Pcbnew:**

Lorsque Pcbnew ne trouve pas de fichier **.cmp**, il utilise l'association composants/empreintes trouvée, si elle existe, dans la netliste.

## **Eeschema - Outils du Toolbar général**

Cependant l'utilisation du fichier intermédiaire .cmp reste très utile, car si le concepteur change une associations directement dans Pcbnew, le fichier .cmp correspondant est automatiquement mis à jour.

## 5 - Création / Edition de Schémas

### Table des matières

5 - Création / Edition de Schémas.....	1
5.1 - Quelques définitions.....	1
5.2 - Considérations générales.....	1
5.3 - La chaîne de développement.....	2
5.4 - Placement et Édition des composants.....	2
5.4.1 - Appel et placement d'un composant.....	2
5.4.2 - Placement d'une alimentation.....	3
5.4.3 - Edition/Modification d'un composant déjà placé.....	3
5.4.3.1 - Modification du composant.....	4
5.4.3.2 - Modification d'un champ du composant.....	4
5.5 - Fils, Bus, Labels, Alimentations.....	5
5.5.1 - Les éléments de base.....	5
5.5.2 - Établissement des connexions (Fils et Labels).....	5
5.5.3 - Établissement des connexions (Bus).....	6
5.5.3.1 - Membres d'un bus.....	6
5.5.3.2 - Connexions entre membres d'un bus.....	6
5.5.3.3 - Connexions globales entre bus.....	7
5.5.4 - Connexion des alimentations.....	7
5.5.5 - Utilisation des symboles « NoConnect ».....	8
5.6 - Compléments.....	8
5.6.1 - Commentaires.....	8
5.6.2 - Cartouche.....	9

#### 5.1 - Quelques définitions

Un schéma peut être représenté sur une seule feuille, mais la plupart du temps nécessitera plusieurs feuilles.

Un schéma représenté sur plusieurs feuilles sera dit alors **hiérarchique**, et l'ensemble de ces feuilles (chacune représentée par un fichier propre) constitue pour Eeschema un **projet**.

Le projet est constitué du schéma principal, appelé schéma « racine » (ou « root »), et des sous schémas constituant la hiérarchie.

De façon à ce que Eeschema puisse, à partir du schéma racine, retrouver tous les autres fichiers du projet, on devra suivre des règles de dessin qui seront développées par la suite.

Par la suite on parlera de projet, aussi bien pour les schémas réduits à une seule feuille, que pour les schémas multi feuilles en hiérarchie.

Par ailleurs un chapitre spécial développe l'utilisation de la hiérarchie et ses particularités.

#### 5.2 - Considérations générales

Un schéma réalisé grâce à Eeschema est plus qu'une simple représentation graphique d'un montage électronique.

C'est normalement le point d'entrée d'une chaîne de développement qui permet :

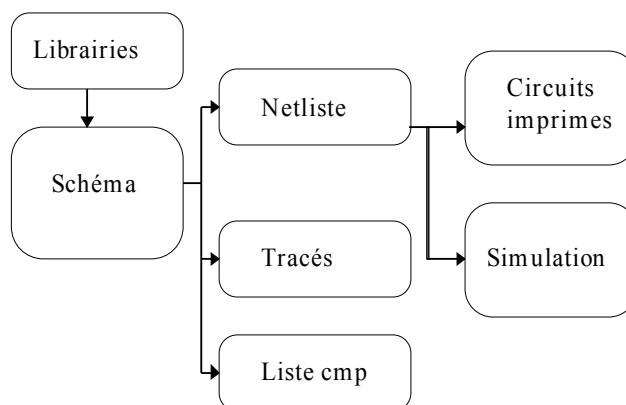
- Un contrôle des règles électrique (contrôle E.RC.) permettant souvent de déceler automatiquement des erreurs ou des oublis sur le schéma.
- La génération automatique de la liste des composants
- La génération de « netlistes » pour la simulation de fonctionnement grâce aux logiciels de simulation comme Pspice.
- La génération de « netlistes » pour la réalisation de circuits imprimés (PCBNEW). Le contrôle de cohérence entre le schéma et le circuit imprimé est alors automatique et instantané

De façon à pouvoir profiter de toutes ces possibilités, on devra respecter certaines contraintes et conventions, pour éviter les mauvaises surprises et les erreurs.

Un schéma est principalement constitué de composants, de fils de connexion ou « wires », de labels, jonctions, bus et alimentations.

Pour plus de clarté dans le schéma, on pourra y placer des éléments purement graphiques comme les entrées de bus, des textes de commentaires, et des traits pointillés pour encadrer des sous ensembles.

#### 5.3 - La chaîne de développement



Le logiciel de schématisation travaille à partir de librairies de composants.

Outre les fichiers de tracé, le fichier netliste est tout particulièrement important car c'est lui que les autres logiciels utilisent.

Un fichier netliste donne la liste des composants et la liste des connexions issues du schéma.

Il existe (malheureusement pour l'utilisateur) un grand nombre de formats de netliste, dont certains sont plus connus. C'est le cas du format Spice par exemple.

## 5.4 - Placement et Édition des composants

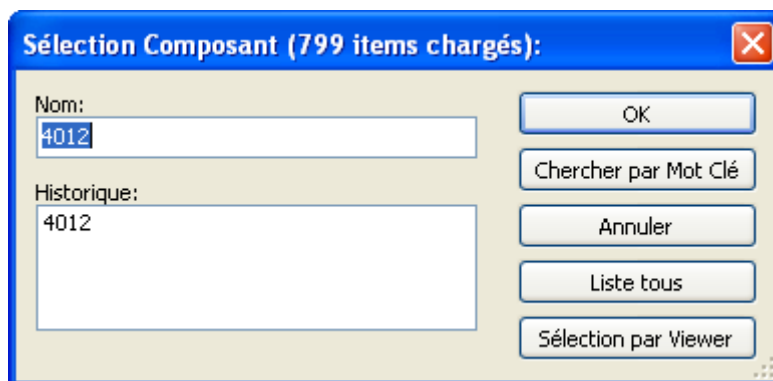
### 5.4.1 - Appel et placement d'un composant



Pour charger un composant, utiliser l'outil

Pour placer un nouveau composant, cliquer à la position voulue pour ce placement.

Une fenêtre d'affichage permet d'entrer le nom du module à charger.



La fenêtre d'historique affiche les derniers éléments chargés et permet de sélectionner immédiatement l'un d'eux.

Si l'on entre uniquement \*, ou si on sélectionne le bouton « **liste tous** », Eeschema affichera alors la liste des librairies, puis des composants disponibles.

On peut aussi lister une sélection: par exemple si on entre **LM2\***, tous les composants dont le nom commence par **LM2** seront listés

On peut aussi sélectionner les composants par mot clé.

Si l'on entre le symbole = suivi de mots clés, Eeschema affichera alors la liste des composants disponibles, restreinte aux modules comportant dans leur liste de mots clés **tous** les mots clés entrés.

Ou rentrer le mot clé et utiliser le bouton « Chercher mot clé »

On peut aussi utiliser le visualisateur de librairies pour sélectionner un composant.

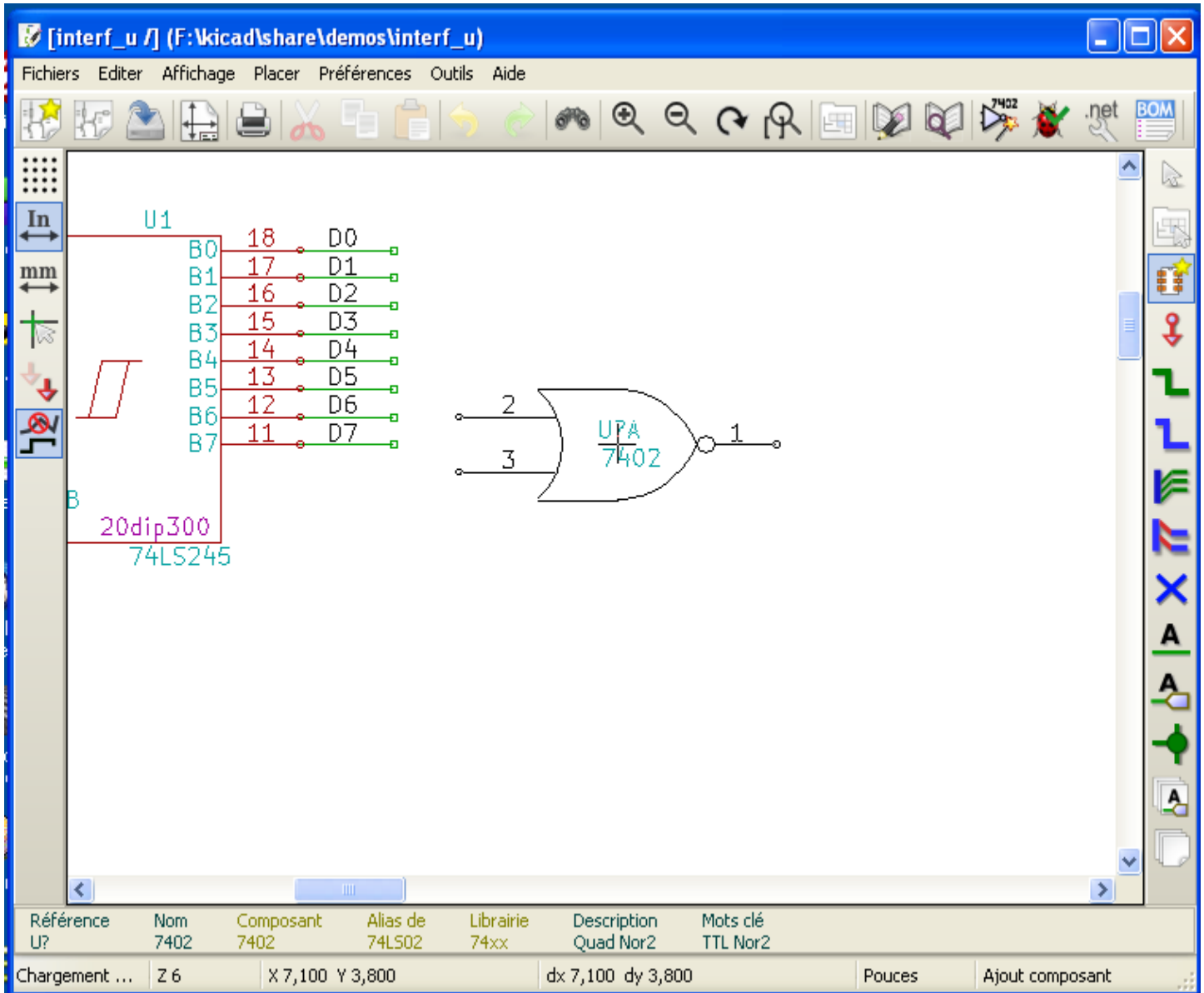
Le composant sélectionné apparaîtra à l'écran, en mode déplacement.

On pourra avant de le placer en position désirée (cliquer sur le bouton gauche de la souris), le faire tourner de 90 degrés (et par rotations successives de 180 et 270 degrés), de le mettre en miroir selon l'axe X ou Y ou sélectionner sa représentation convertie (cliquer sur le bouton droit de la souris pour appeler le menu « Pop up » d'édition rapide).

On pourra évidemment le faire tout aussi facilement après placement.

Si le composant exactement désiré n'existe pas, penser que l'on peut la plupart du temps charger un composant analogue et modifier après coup sa valeur : si l'on veut un 54LS00 on peut fort bien charger un 74LS00 et éditer la valeur 74LS00 en 54LS00.

Voici un composant en cours de placement :



### 5.4.2 - Placement d'une alimentation

Un symbole d'alimentation est un composant (Les symboles sont regroupés dans la librairie « **power** »).

On peut utiliser donc la commande précédente.



Mais comme ces placements sont fréquents; on dispose de l'outil

Cet outil est analogue au précédent, mais la recherche se fait directement dans la librairie **power**, d'où gain de temps.

### 5.4.3 - Edition/Modification d'un composant déjà placé

Il y a deux type d'édition.

- La modification (position, orientation, sélection de la représentation ou de la part) du composant lui même.
- La modification d'un des champs ( Référence, valeur, ou autres ) du composant.

Lorsqu'un composant vient d'être placé, il faudra peut être modifier sa valeur ( particulièrement pour les résistances, condensateurs...), mais il est inutile de vouloir lui attribuer immédiatement un numéro de référence, ou sélectionner la part (pour les composants à parts multiples comme le 7400).

Ceci pourra avantageusement être fait automatiquement par la fonction d'annotation automatique.

### 5.4.3.1 - Modification du composant

Positionner le curseur souris sur le composant (ne pas se positionner sur un champ). On peut alors :

- Cliquer 2 fois par le bouton de gauche de la souris pour ouvrir la fenêtre d'édition complète du composant.
- Cliquer par le bouton de droite de la souris pour ouvrir le menu « Pop Up », et utiliser l'une des commandes affichées (Move, Orient, Edit, Delete).

### 5.4.3.2 - Modification d'un champ du composant

On peut pour chaque champ (il y a au moins valeur et référence), modifier le contenu, la position, l'orientation la taille et la visibilité du champ.

Pour les éditions simples, positionner le curseur souris sur le texte du composant et (au choix):

- Cliquer 2 fois par le bouton de gauche de la souris pour ouvrir la fenêtre de modification du texte.
- Cliquer par le bouton de droite de la souris pour ouvrir le menu « Pop Up », et utiliser l'une des commandes affichées (Move, Rot, Edit, Delete) (si ce n'est pas le champ valeur ou référence).

Pour des éditions plus complètes ou des créations de champ il faut éditer le composant correspondant. En voici le menu:

**Propriétés du Composant**

**Options**

Unité: 1

Orientation (Degrés): 0

Miroir: Normal

Nom en librairie: CRYSTAL

☐ Convert

Remettre aux Valeurs en Lib.

**Champs**

Référence	X1
Valeur	8MHz
Module	HC-18UH
Documentation	
Info	
Fab.	
Ref. Fab	

Ajouter Champ

Supprimer Champ

Vers le haut ^

**Justification du Texte:**

Justification Horiz.: Gauche, Centrer, Droite

Justification Vert.: Dessous, Centrer, Dessus

**Visibilité**

☒ Visible

☐ Rotation

**Style:**

Normal, Italique, Gras, Gras Italique

Nom Champ: Référence

Texte Champ: X1

Taille ("): 0,070

Pos X ("): 0,000

Pos Y ("): 0,200

OK Annuler

On peut alors choisir l'orientation et d'autres options du composant, et éditer, ajouter ou supprimer un champ.

Chaque champ peut être visible ou non et être horizontal ou vertical.

La position affichée (et modifiable) est toujours indiquée pour un composant affiché normalement (pas de rotation ou de miroir) et est relative au point d'ancrage du composant.

L'option « **Remettre aux Valeurs par Défaut en Librairie** » réinitialise l'orientation du composant, et la position et les options des champs aux valeurs par défaut de la librairie.

Cependant les **textes des champs ne sont pas modifiés**, car cela aurait des conséquences graves pour la schématique.

## 5.5 - Fils, Bus, Labels, Alimentations

### 5.5.1 - Les éléments de base

Tous les autres éléments de dessin peuvent être placés grâce également aux outils spécifiques du toolbar vertical droit. Ces éléments de dessin sont :

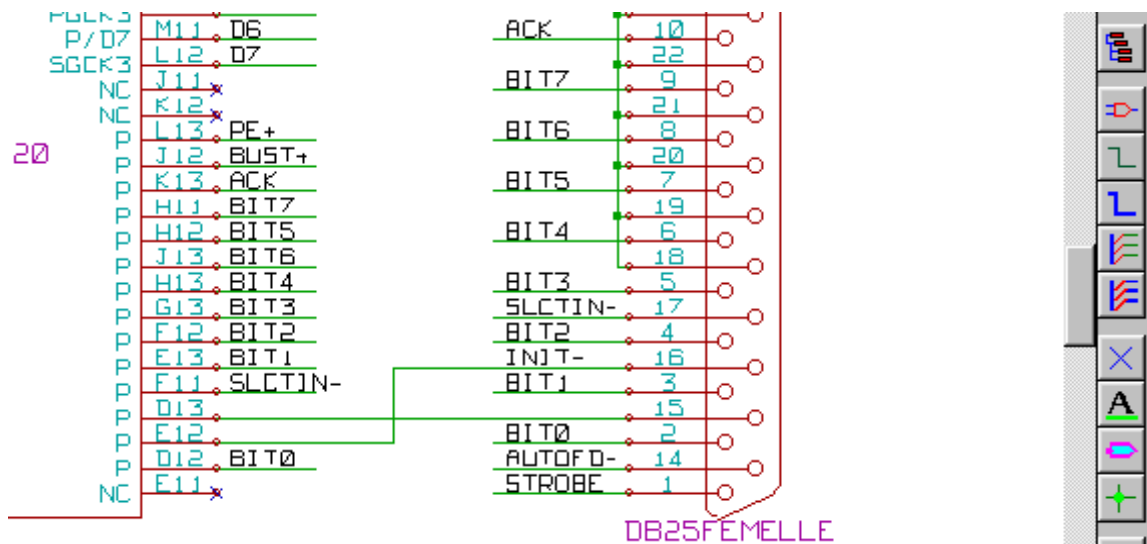
- Les  **fils de connexion**  (Wires) pour les connexions usuelles.
- Les  **bus**  (qui ne servent à connecter que des labels de bus, et à l'esthétique du dessin)
- Les  **traits**  pointillés, pour du graphisme de présentation.
- Les  **jonctions** , pour la connexion forcée de fils ou de bus se croisant.
- Les  **entrées**  de bus, type wire/bus ou bus/bus, pour l'esthétique du dessin.
- Les  **labels**  pour des connexions usuelles.
- Les  **labels globaux** , pour les connexions entre feuilles de hiérarchie.
- Les  **textes**  de commentaire.
- Les «  **NoConnect**  » (symboles de Non Connexion)
- Les feuilles de hiérarchie, et leurs pins de connexions.

### 5.5.2 - Établissement des connexions (Fils et Labels)

Il existe deux moyens pour établir des connexions :

- Tracer des fils ( Wires ) de pin à pin.
- Utiliser des Labels.

La figure suivante montre les deux méthodes :



#### Remarque 1 :

Le point de « contact » (ou d'ancrage) d'un label est le coin bas gauche de la première lettre du label. Ce point doit donc être en contact avec un fil, ou être superposé au point de contact d'une pin pour que ce label soit pris en compte.

#### Remarque 2 :

Pour établir une connexion, un segment de fil doit être connecté par ses extrémités à une extrémité d'un autre segment ou à celle d'une pin.

**S'il y a chevauchement** (si un fil passe sur une pin, mais sans y être connecté par une extrémité), **il n'y a pas de liaison**.

Cependant, un label sera connecté à un fil quelle que soit la position du point d'ancrage du label, sur ce fil.

#### Remarque 3 :

Si un fil doit être connecté à un autre fil, autrement que par leurs extrémités, il faudra placer une jonction (commande **P**lace/**J**onction) au point de croisement.

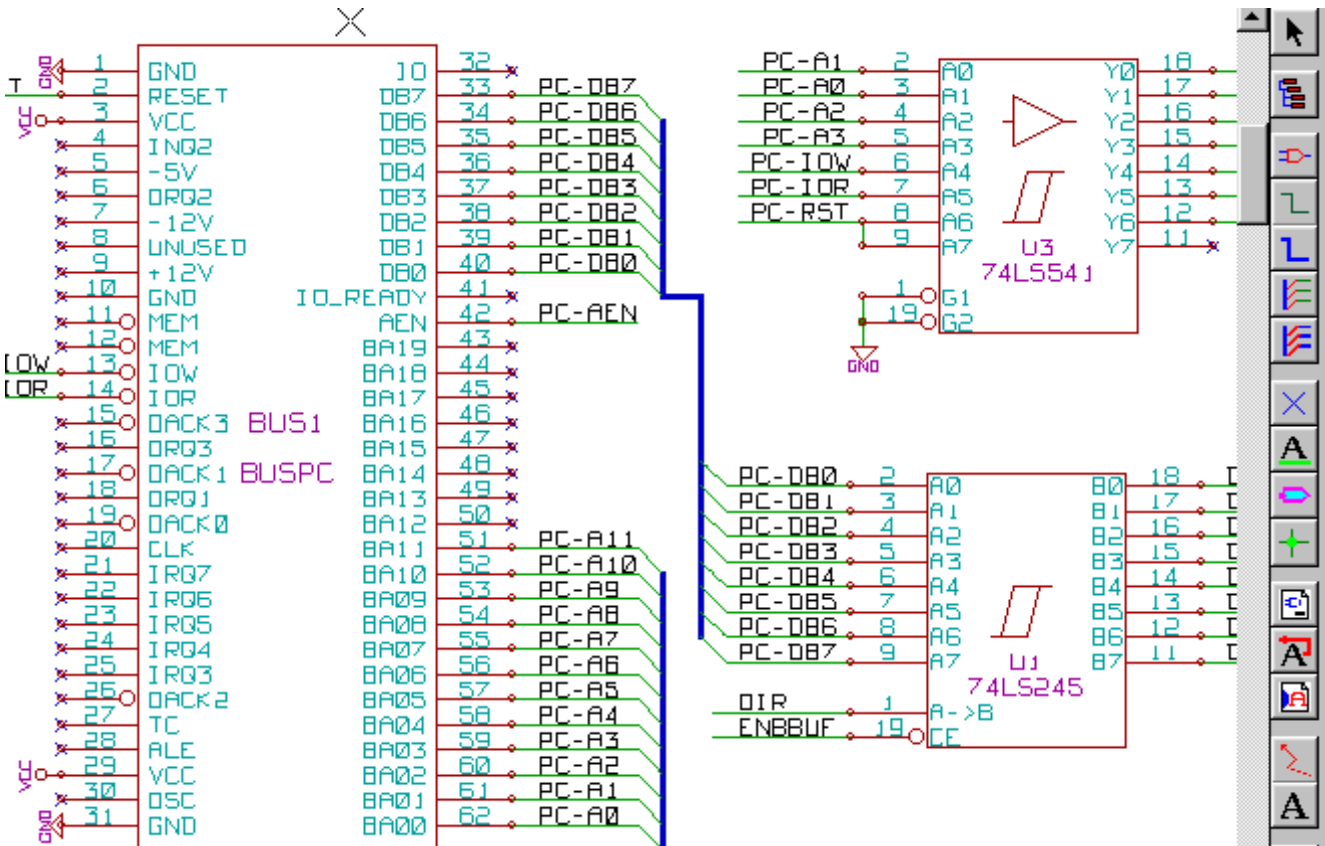
La figure précédente ( sur le fil connecté à U1B pin 6) montre un tel cas de connexion utilisant une jonction.

#### Remarque 4 :

Si deux labels différents sont placés sur le même fil, ils sont connectés entre eux et deviennent équivalents : tous les autres éléments connectés à l'un ou l'autre des deux labels sont alors connectés entre eux.

### 5.5.3 - Établissement des connexions (Bus)

Considérons le schéma suivant :



De nombreuses pins (composants U1 et BUS1 en particulier) sont connectées à des bus.

#### 5.5.3.1 - Membres d'un bus

A strictement parler, du point de vue schématique, un bus est un ensemble de signaux, ayant un nom commençant par préfixe commun, et terminé par un nombre.

Ce n'est pas donc tout à fait la notion de bus telle que celui d'un microprocesseur.

Chacun des signaux est un **membre** du bus.

PCA0, PCA1, PCA2, sont ainsi des membres du bus PCA.

Le bus complet est noté PCA[n..m], où n et m sont le premier et le dernier numéro de membre de ce bus.

Ainsi si PCA a 20 membres de 0 à 19, le bus complet est noté PCA[0..19].

Mais un ensemble de signaux comme PCA0, PCA1, PCA2, WRITE, READ ne peuvent pas être regroupés en bus.

#### 5.5.3.2 - Connexions entre membres d'un bus

Les connexions entre pins reliées à un même membre de bus doivent se faire par **labels**.

En effet, relier directement une pin à un bus n'a aucun sens, puisque un bus est un ensemble de signaux, et cette connexion est du reste ignorée par Eeschema.

Dans l'exemple ci-dessus, les connexions sont faites par les labels placés sur les fils reliés aux pins.

Les connexions par l'intermédiaire d'entrées de bus (segments de fils à 45 degrés) aux fils type bus n'ont qu'une valeur esthétique, et ne sont pas nécessaires sur le plan purement schématique.

En pratique, grâce à la commande de **répétition** (touche **Inser**), les connexions seront très vite mises en place de la façon suivante, si les pins du composant sont alignées dans l'ordre des membres (cas très courant en pratique sur les composants tels que mémoires, microprocesseurs...) :

- Placer le premier label (par exemple PCA0)
- Utiliser la commande de répétition autant de fois qu'il y a de membres à placer. EESchema créera automatiquement les autres labels (PCA1, PCA2...) alignés verticalement, en principe au niveau des autres pins.
- Dessiner le fil sous le premier label. Puis utiliser la commande de répétition pour placer automatiquement les autres fils sous les autres label.



- Si désiré, placer les entrées de bus de la même façon (Placer la première entrée, puis la commande de répétition).

### Remarque :

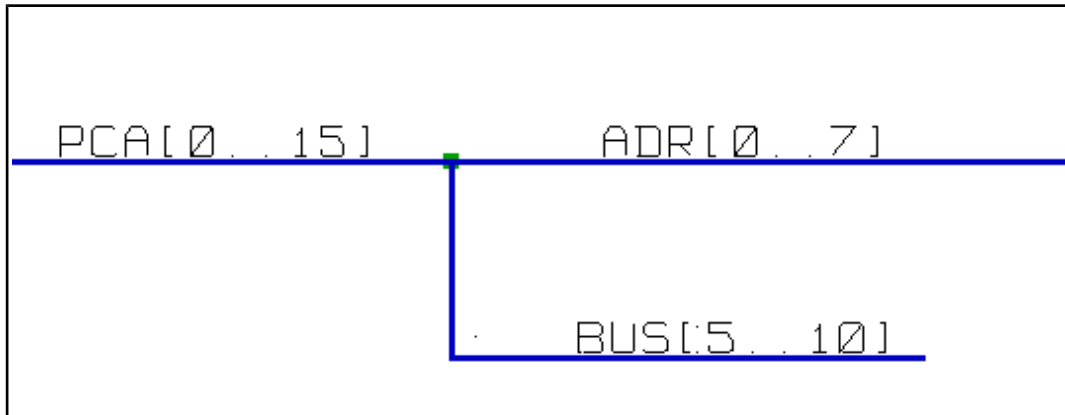
Dans le menu **Préférences/Options** on peut régler les paramètres de répétition :

- Pas vertical
- Pas horizontal
- Incrémentation du label (qui peut donc être incrémenté par 2, 3.. ou être décrémenté).

### 5.5.3.3 - Connexions globales entre bus

On peut vouloir faire des connexions entre bus, soit pour réunir deux bus de nom différents, soit dans le cas de hiérarchie, pour établir les connexions entre feuilles différentes.

On peut faire globalement ces connexions de la façon suivante.



Les bus PCA[0..15], ADRL[0..7] et BUS[5..10] sont connectés ( remarquer ici la jonction puisque le fil de bus vertical arrive au milieu du segment bus horizontal).

Plus précisément, les membres de poids correspondants sont connectés ensemble.

PCA0, ADRL0 sont reliés, ( de même que PCA1 et ADRL1 ... PCA7 et ADRL7).

De plus PCA5, BUS5 et ADRL5 sont reliés ( de même que PCA6, BUS6 et ADRL6 ainsi que PCA7, BUS7 et ADRL7).

PCA8 et BUS8 sont également reliés (de même que PCA9 et BUS9, PCA10 et BUS10)

Par contre on ne peut relier ainsi globalement des membres de poids différents.

Si l'on veut relier des membres de poids différents de deux bus différents, il faudra le faire membre à membre comme deux labels usuels, en les plaçant sur un même fil (type Wire).

### 5.5.4 - Connexion des alimentations

Lorsque les pins d'alimentation des composants sont visibles, elles doivent être reliées entre elles, comme pour n'importe quel autre signal.


La difficulté vient des composants (tels que portes et bascules) pour lesquels les pins d'alimentation sont normalement invisibles (**pins « power invisibles »**).

La difficulté est double car :

- On ne peut y connecter des fils, du fait de leur invisibilité.
- On ne connaît pas non plus leur nom.

Et pas ailleurs, il serait une mauvaise idée de les rendre visibles et de les connecter comme les autres pins, car le schéma deviendrait lourd et non conforme aux conventions usuelles.

### Remarque :

Si l'on veut forcer l'affichage de ces pins « power invisibles », activer dans le dialogue Préférences/Option du menu principal, l'option **Forcer affichage des pins invisibles**, ou l'icône  du toolbar gauche (toolbar d'options)

EESchema utilise donc une technique de connexion automatique des pins d'alimentations invisibles :

**Toutes les pins « power invisible » de même nom sont connectées automatiquement entre elles** même en l'absence de toute autre connexion..

Cependant cette connexion automatique doit être complétée :

- Par les connexions aux autres pins visibles, connectées à ces alimentations.
- Éventuellement par les connexions entre groupe de pins invisibles, mais de nom différents (par exemple, les pins de masse sont usuellement appelées « GND » en TTL et « VSS » en MOS, et elles doivent être reliées ensembles.).

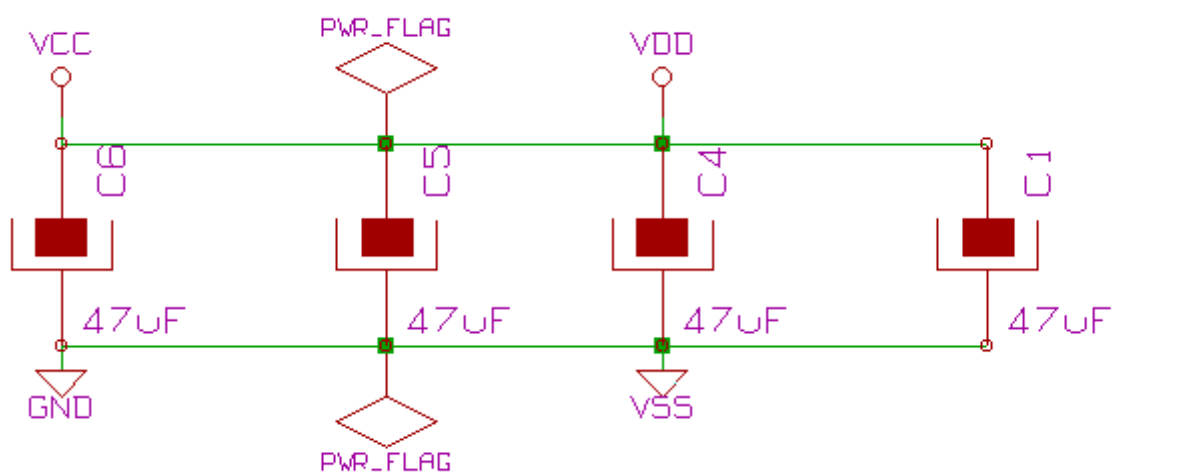
Pour ces connexions, on doit utiliser des symboles d'alimentation (composants spécialement conçus pour cela, que l'on

doit créer et modifier par l'éditeur de librairie).

Ces symboles sont constitués d'une pin « power invisible » associée au graphique voulu.

Ne pas utiliser de labels, qui n'ont qu'un pouvoir de connexion « local », et qui ne connecteraient pas les pins « power invisibles ». (Voir les notions sur la hiérarchie pour plus de détails).

Voici un exemple de connexions d'alimentations.



Dans cet exemple, la masse (GND) est reliée à l'alimentation VSS, et l'alimentation VCC est reliée à VDD.


Par ailleurs la pin 2 de LED1 et la pin 2 de C7 (pins standards visibles) sont reliées à GND, par un symbole de masse. On remarquera les deux symboles PWR\_FLAG. Ils signalent que les deux alimentations VCC et GND sont bien connectées à une source d'alimentation.

Sans ces flags, l'outil ERC Donnerait un diagnostic: *Warning: alimentations non pilotées.*

Tous ces symboles sont des composants faisant partie des librairies schématiques utilisées.

### 5.5.5 - Utilisation des symboles « NoConnect »

Ces symboles sont très utiles pour la fonction de contrôle automatique des règles électriques (Fonction dite E.R.C.). Ce contrôle signale toutes les pins non connectées.

Si des pins doivent donc réellement rester non connectées, il faut placer un symbole de non connexion (outil ) sur ces pins, de façon à ce que la fonction E.R.C. ne génère pas inutilement d'erreurs.

La présence ou l'absence de ces symboles n'a cependant aucune influence sur les netlistes générées.

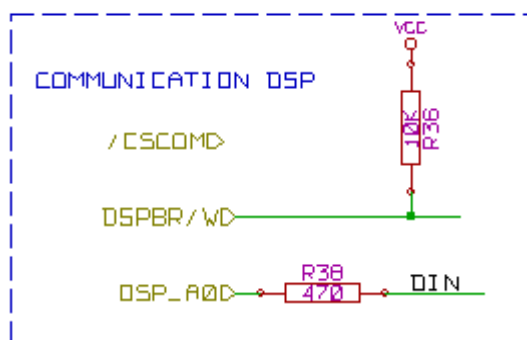
## 5.6 - Compléments

### 5.6.1 - Commentaires

Il peut être bon (pour la bonne compréhension du schéma) de placer des indications telles que légendes, encadrement de sous ensembles.

Pour cela on doit utiliser des **textes** (outil ) et des **lignes pointillées** (outil ) et non des labels et des fils, qui sont des éléments de connexion.

Voici un élément de présentation :



### 5.6.2 - Cartouche

Le cartouche est initialisé grâce à l'outil  :

Ajustage opt Page

Dim Page:

Format A4

☒ Format A3

Format A2

Format A1

Format A0

Format A

Format B

Format C

Format D

Format E

Format libre

Format libre dim X:

17,000

Format libre dim Y:

11,000

Number of feuilles: 1

Numéro feuille: 1

Révision:

2B

☐ Exporter vers autres feuilles

Titre:

INTERFACE UNIVERSEL

☐ Exporter vers autres feuilles

Société:

KICAD suite GPL

☐ Exporter vers autres feuilles

Commentaire1:

Comment 1

☐ Exporter vers autres feuilles

Commentaire2:

Comment 2

☐ Exporter vers autres feuilles

Commentaire3:

Comment 3

☐ Exporter vers autres feuilles

Commentaire4:

Comment 4

☐ Exporter vers autres feuilles

OK

Annuler

## Eeschema - Création / Edition de Schémas

Le cartouche complet est alors :

Comment 4		
Comment 3		
Comment 2		
Comment 1		
<b>KICAD</b>		
File: interf_u.sch +		
Sheet: /		
<b>Title: INTERFACE UNIVERSEL</b>		
Size: A3	Date: 4 jul 2009	<b>Rev: 2B</b>
KiCad E.D.A. eeschema (2009-11-15-unstable)		Id: 1/1

La date et le numéro de feuille (Sheet nn) sont mis à jour automatiquement :

- Pour la date, à chaque modification du schéma.
- Pour le numéro de feuille (utile en hiérarchie), lors des fonctions d'annotation.

## 6 - Schémas en Hiérarchies

### Table des matières

6 - Schémas en Hiérarchies.....	1
6.1 - Présentation.....	1
6.2 - Navigation dans la Hiérarchie.....	2
6.3 - Labels locaux, hiérarchiques et globaux.....	3
6.3.1 - propriétés:.....	3
6.3.2 - Remarques:.....	3
6.4 - Création d'une hiérarchie. Généralités.....	3
6.5 - Placement d'un symbole de hiérarchie.....	3
6.6 - Placement des Connexions : Pins hiérarchiques.....	4
6.7 - Placement des Connexions : Labels Hiérarchiques.....	5
6.8 - Labels, Labels Hiérarchiques, Labels Globaux et Pins Power invisibles.....	6
6.8.1 - Labels simples.....	6
6.8.2 - Labels Hiérarchiques.....	7
6.8.3 - Pins « Power invisibles ».....	7
6.8.4 - Labels Globaux.....	7
6.9 - Hiérarchie Complexe.....	7
6.10 - Hiérarchie "à plat".....	8

#### 6.1 - Présentation

L'organisation d'un schéma un peu important en hiérarchie est généralement une bonne solution.

Si l'on veut réaliser un schéma d'une certaine importance, il faudra :

- Soit le réaliser sur une feuille de grandes dimensions, ce qui posera des problèmes pratiques de tracé, copie et manutention.
- Soit le réaliser sur plusieurs feuilles différentes, ce qui conduit à la structure en hiérarchie.

Le schéma complet est alors constitué du schéma principal, appelé schéma « racine » (ou « root »), et des sous schémas constituant la hiérarchie.

En outre, une découpe habile du schéma général en feuilles séparées permet souvent d'en améliorer la lisibilité.

A partir de la racine, on doit pouvoir retrouver tous les sous schémas complémentaires.

Eeschema permet une gestion très facile d'une hiérarchie, grâce à un « navigateur » de hiérarchie



intégré (bouton du toolbar horizontal ou vertical, détaillé plus loin).

En fait, il existe deux approches de hiérarchie (pouvant exister simultanément) :

La première est celle qui vient d'être évoquée et est d'usage général.

La deuxième consiste à créer en librairie des composants qui apparaissent en schématique comme des composants classiques, mais qui en réalité, correspondent à un schéma qui décrit leur structure interne.

Ce deuxième type est plutôt utilisé lorsque l'on développe des circuits intégrés, car dans ce cas on est amené à utiliser des bibliothèques de fonctions dans le schéma que l'on construit.

Eeschema ne traite pas actuellement ce deuxième cas.

**Une hiérarchie peut être:**

- **simple: une feuille est utilisée une seule fois**
- **complexe: une feuille est utilisée plusieurs fois (multiples instances)**
- **"A plat" qui est une hiérarchie simple particulière, où les connexions entre feuilles ne sont pas représentées.**

**Eeschema traite toutes ces hiérarchies.**

La création d'une hiérarchie est simple, et l'ensemble de la hiérarchie se manipule à partir du schéma racine, comme s'il n'y avait un seul schéma.

Les deux points à connaître sont :

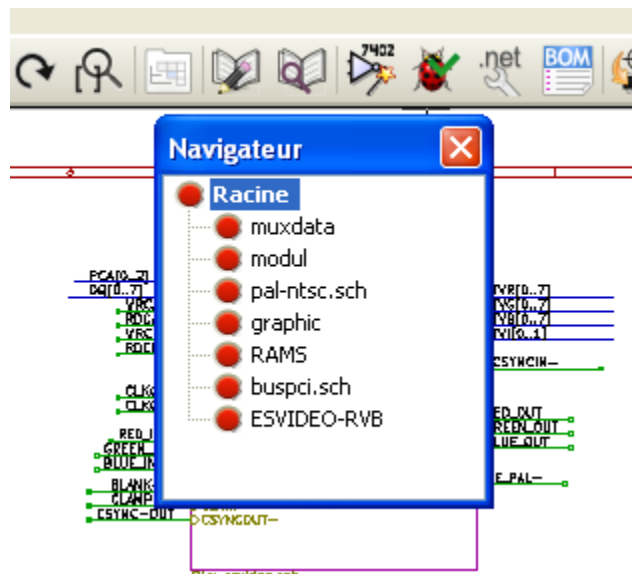
- Comment créer un sous schéma.
- Comment établir les liaisons électriques entre les schémas constituant la hiérarchie.

#### 6.2 - Navigation dans la Hiérarchie



Elle se fait très simplement grâce au navigateur (outil du toolbar **horizontal**) En voici un

exemple :




On peut accéder directement à chaque feuille, en cliquant sur le nom de cette feuille.

### **Navigation rapide:**

Positionner le curseur souris sur une feuille de hiérarchie, puis soit:

- Activer le bouton droit et choisir **enter sheet (Entrer dans feuille)**
- Double – cliquer (bouton gauche).

On peut aussi rapidement accéder à la feuille mère, ou à une feuille fille grâce à l'outil  du toolbar **vertical**.


Après sélection de l'outil:

- Cliquer sur un symbole de feuille pointée par la souris = sélection de cette feuille.
- Cliquer ailleurs: = sélection de la feuille mère.

## 6.3 - Labels locaux, hiérarchiques et globaux.

### 6.3.1 - propriétés:

Les labels locaux (outil ) ne connectent des signaux qu'à l'intérieur d'une feuille

Les labels hiérarchiques (outil ) connectent des signaux à l'intérieur d'une feuille **et** à une pin de hiérarchie apparaissant dans la feuille mère.

Les labels globaux (outil ) connectent des signaux dans **toute** la hiérarchie.

Les pins d'alimentations (donc du type **power in** et **power out**) **invisibles** sont analogues à des labels globaux car elles sont considérées comme connectées entre elles à travers toute la hiérarchie.

### 6.3.2 - Remarques:

Dans une hiérarchie (simple ou complexe) on peut utiliser à la fois des labels hiérarchiques et des labels globaux.

## 6.4 - Création d'une hiérarchie. Généralités

On doit :

- Placer dans la feuille de schéma courante, qui sera la feuille mère (initialement la feuille racine) un symbole de hiérarchie appelé « sheet ».
- Grâce au navigateur de hiérarchie, se placer dans le nouveau schéma (schéma fils) et le dessiner, comme un schéma classique.
- Établir les liens électriques entre les deux schémas en plaçant dans le nouveau schéma des Labels Hiérarchiques, et dans la feuille mère des labels dits Pins de Hiérarchie de même nom. Ces Pins de Hiérarchie

seront connectés dans la feuille mère aux autres éléments du schéma comme des pins d'un composant standard.

### 6.5 - Placement d'un symbole de hiérarchie

Il faut dessiner un rectangle défini par deux points diagonaux symbolisant la feuille dite feuille fille ( ou schéma fils). La taille de ce rectangle doit permettre le placement ultérieur de labels particuliers type pins de hiérarchie, qui correspondent dans la feuille fille aux labels hiérarchiques. Ces labels jouent le même rôle que les pins d'un composant usuel.



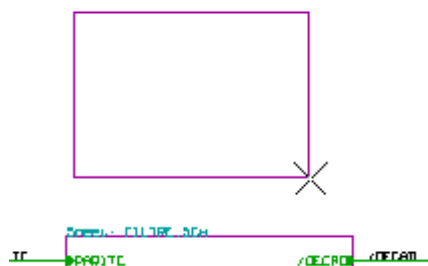
Sélectionner l'outil .

Cliquer à l'emplacement de premier point du rectangle.

La taille du rectangle est réglée à la souris.

Cliquer à l'emplacement du deuxième point pour valider la création de cette feuille.

Voici un exemple :



Lors de la validation, un nom du fichier de sauvegarde du fichier schématique correspondant et un nom de symbole (pour pouvoir accéder au schéma correspondant à l'aide du navigateur de hiérarchie) seront demandés.

Il faut donner au moins un nom de fichier. Si il n'y a pas de nom de symbole, c'est le nom du fichier qui sera pris comme nom de symbole (façon usuelle de faire).

### 6.6 - Placement des Connexions : Pins hiérarchiques.

On va ici créer des points de connexion (pins hiérarchiques) pour le symbole qui vient d'être créé.

Ces points de connexion sont analogues à des pins pour un composant normal, avec toutefois la possibilité de connecter un bus complet avec un seul point de connexion.

#### Il y a deux façons de faire:

- Placer les différentes pins **avant** de réaliser le schéma correspondant de la sous feuille (placement manuel).
- Placer les différentes pins **après** avoir réalisé le schéma correspondant de la sous feuille et dessiné les labels hiérarchiques (placement semi-automatique).

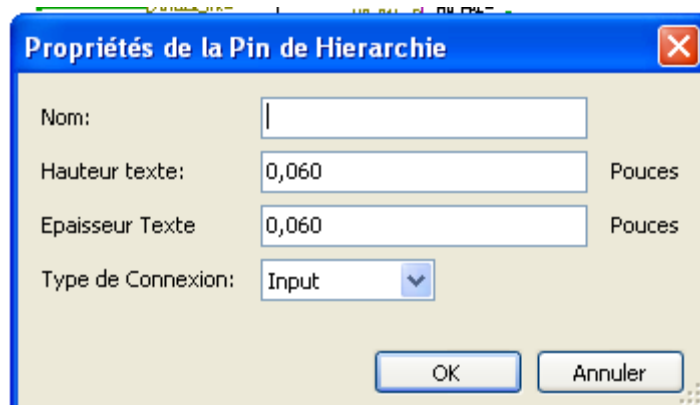
Il est bien préférable de choisir la deuxième solution, chaque fois que ce sera possible.

#### Placement manuel:



- Sélectionner l'outil
- Cliquer sur le symbole de hiérarchie où l'on veut placer cette pin.

Voici comme exemple la création de la pin hiérarchique « CONNEXION ».



On peut définir son graphisme, et sa taille à la création ou après coup, par la commande Edit: (menu PopUp activé en cliquant sur le bouton droit de la souris).

Les différents graphismes du symbole sont :

- Input
- OutPut
- BiDi
- TriState
- NoSpecified

Le choix ne joue que sur l'aspect graphique, et n'a aucune importance par ailleurs.

### **Placement automatique:**



- Sélectionner l'outil
- Cliquer sur le symbole de hiérarchie où l'on veut importer les pins à partir des labels hiérarchiques placés dans le schéma correspondant. Une pin hiérarchique doit apparaître, s'il en existe un label hiérarchique nouveau, c'est à dire ne correspondant pas à une pin déjà placée.
- Cliquer à l'endroit où l'on veut placer cette pin.

Toutes les pins nécessaires peuvent ainsi être placées rapidement et sans erreur. Leur forme correspondront aux label hiérarchique correspondants.

## **6.7 - Placement des Connexions : Labels Hiérarchiques.**

A chaque pin du symbole hiérarchique qui vient d'être créé, doit correspondre un label dit label hiérarchique dans le schéma fils.

Un label hiérarchique a un rôle analogue à un label, mais il assure la connexion entre la feuille fille et feuille mère.

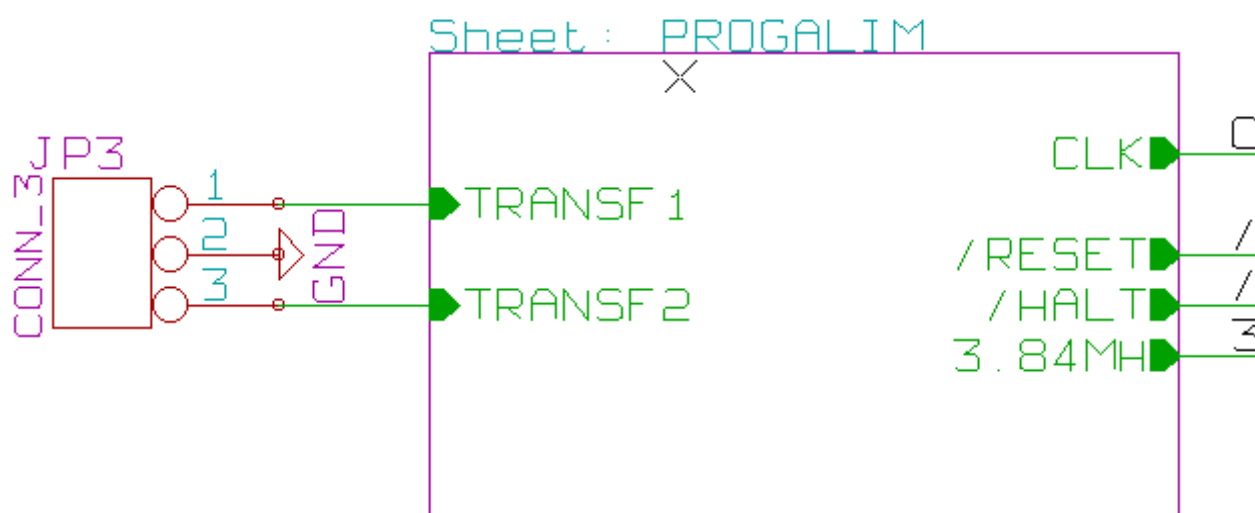
Le graphisme des deux labels complémentaires (pin et label hiérarchique) est analogue.



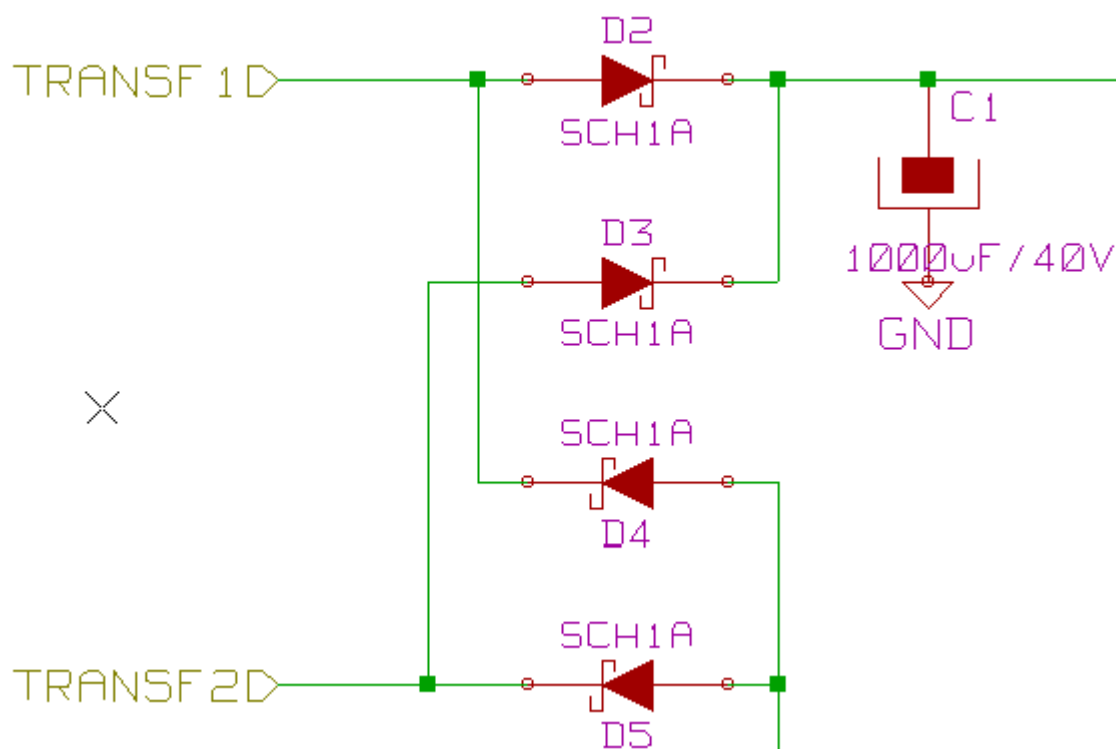
La création dans un schéma d'un label hiérarchique est faite par l'outil

Voici un exemple de feuille mère :





Remarquer les deux pins TRANSF1 et TRANSF2, par ailleurs connectés à un composant.  
Voici la connexion complémentaire dans la feuille fille :



On retrouve dans ce schéma, les deux labels hiérarchiques correspondants, qui assurent donc la connexion entre les deux feuilles hiérarchiques.

**Remarque** : On peut utiliser des labels hiérarchiques et des pins hiérarchiques pour connecter deux bus, selon la syntaxe (BUS[n..m]) décrite précédemment.

## 6.8 - Labels, Labels Hiérarchiques, Labels Globaux et Pins Power invisibles

Voici quelques compléments sur les particularité des différents moyens de connexion, autres que les connexions par fils (wires).

### 6.8.1 - Labels simples

Les labels simples ont un pouvoir de connexion local, c'est à dire limité à la feuille schématique où il est placé.

## Eeschema - Schémas en Hiérarchies

Ceci est du au fait que :

- Chaque feuille a un numéro interne de code (Signature temporelle).
- Ce code est associé au Label.

Ainsi, si l'on place le label « TOTO » dans une sous feuille de signature temporelle 01234567, apparaissant dans la feuille racine, le vrai label est en fait /01234567/TOTO, .

Si l'on place également un label « TOTO » dans la feuille racine on place en fait le label /TOTO, différent de /01234567/TOTO.

Ceci est toujours vrai, même s'il n'y a qu'une feuille.

### 6.8.2 - Labels Hiérarchiques

Ce qui est dit pour les labels simples est aussi vrai pour un label hiérarchique.

Ainsi dans la même feuille un label hiérarchique « TOTO » est considéré comme connecté à un label local « TOTO », mais n'est pas connecté à un label hiérarchique ou un label de même nom dans une autre feuille.

### 6.8.3 - Pins « Power invisibles »

On a vu qu'elles étaient connectées ensembles si elles avaient le même nom.

Ainsi toutes les pins déclarées « Power Invisibles » et de nom VCC sont connectées et forment l'équipotentielle VCC, quelle que soit la feuille de hiérarchie ou elles se trouvent.

Ceci explique que si l'on place dans une feuille de hiérarchie un label VCC, il ne sera pas connecté aux pins VCC, car ce label est en réalité VCC\_n, où n est le numéro de feuille.

Si l'on veut que ce label VCC soit réellement connecté à l'équipotentielle VCC, il faudra le connecter explicitement à une pin power invisible, grâce à un symbole d'alimentation VCC.

### 6.8.4 - Labels Globaux

Les labels globaux sont des labels qui sont connectés à travers toute la hiérarchie.

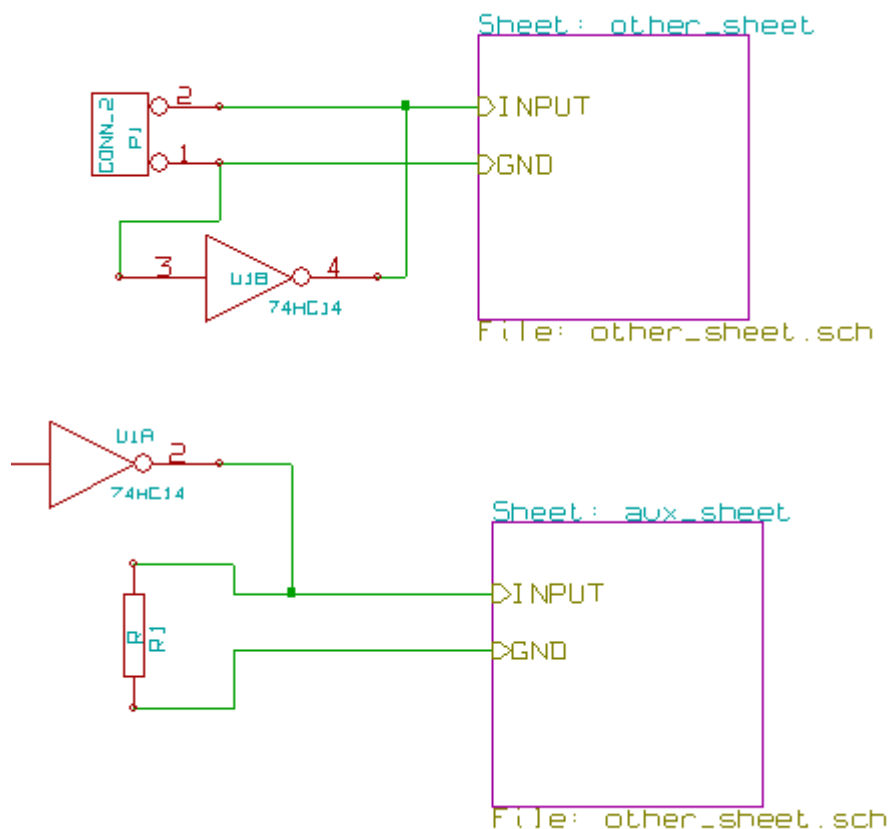
(les alimentations comme VCC ... sont une sorte de labels globaux)

## 6.9 - Hiérarchie Complexe

Voici un exemple où le même schéma est utilisé deux fois.

***Les 2 feuilles de hiérarchie partagent le même schéma du fait que le nom du fichier est identique pour les deux feuilles ("other\_sheet.sch").***

Il faut toutefois que les noms des feuilles soient différents.



## 6.10 - Hiérarchie "à plat"

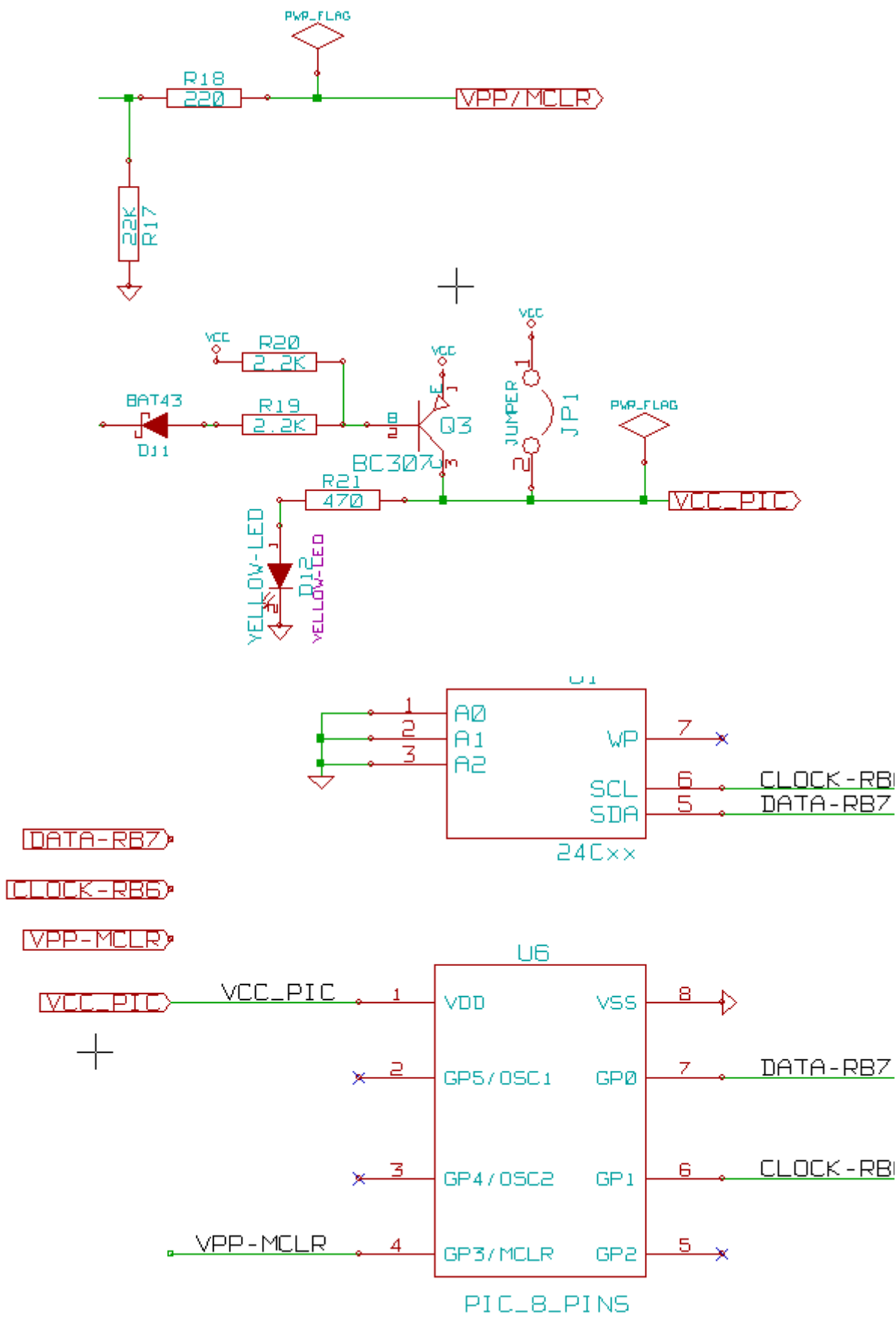
On peut créer un schéma sur plusieurs feuilles, sans avoir besoin de créer les connexions explicitement entre ces feuilles (hiérarchie à plat) en respectant les règles suivantes:

- On devra créer une feuille maitre dans laquelle apparaitront toutes les autres feuilles, comme dans une hiérarchie classique. Il faut la considérer uniquement comme un lien entre les feuilles utiles.
- A la différence d'une hiérarchie classique, on n'aura pas besoin de placer de connexions entre feuilles.
- Tous les connexions entre feuilles utiliseront des Labels Globaux au lieu de labels hiérarchiques.

Voici la feuille maitre:



Voici les 2 feuilles, inter connectées par des labels globaux:



Remarquer les labels globaux	<div>DATA-RB7</div> <div>CLOCK-RB6</div> <div>VPP-MCLR</div>
------------------------------	--


## 7 - Numérotation automatique ( ou Annotation).

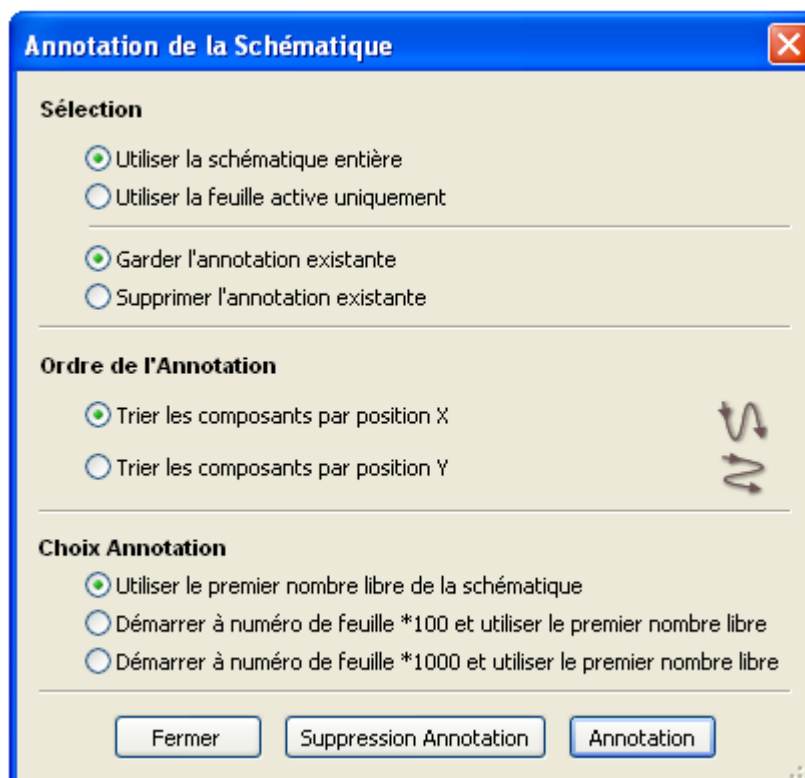
# Table des matières

7 - Numérotation automatique ( ou Annotation).....	1
8 - Rôle.....	1
8.1 - Exemples.....	2
8.1.1 - Sens de Annotation:.....	2
8.1.2 - Choix Annotation:.....	3

## 7.1 - Rôle.



Cette commande (outil ) permet d'attribuer automatiquement un numéro de référence aux composants, et pour les composants à parts multiples, de répartir au mieux ces parts pour minimiser le nombre de ces boîtiers. Le menu est:



Différentes possibilités sont offertes :

- Traiter toute la hiérarchie (option normale).
- Ne traiter que la feuille en cours (n'est utile que pour des tests).
- Annoter tous les composants
- Annoter que les nouveaux composants, c'est à dire ceux dont la référence finit par ? comme IC?. (Garder l'annotation existante).

L'option **Sens de Annotation** permet de choisir comment l'attribution des numéros de référence est faite, au sein de chaque feuille de hiérarchie.

Sauf cas particuliers, une numérotation automatique porte sur le projet entier (ensemble des schémas de la hiérarchie) et sur les nouveaux composants, si on ne veut pas modifier la numérotation déjà faite.

L'option **Choix Annotation** permet de choisir comment la détermination des numéros de référence est faite:

- **Utiliser le premier numéro libre de la schématique:** les composants sont numérotés à partir de 1 (pour chaque référence racine).  
Si une numérotation existe, les numéros non déjà utilisés seront alors utilisés.
- **Démarrer à numéro de feuille\*100 et utiliser le premier numéro libre:**  
La numérotation commence à 101 pour la feuille 1, à 201 pour la feuille 2...

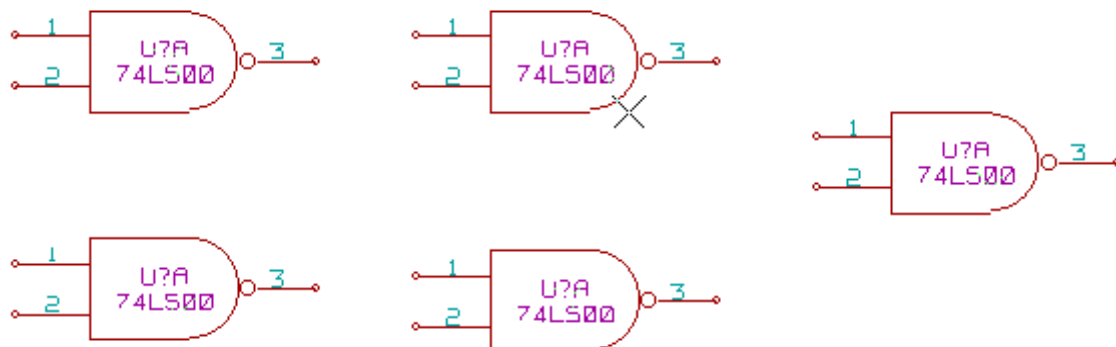
## Eeschema - Numérotation automatique ( ou Annotation).

Dans le cas ou il y a plus de 99 éléments de même référence racine (U, R) dans la feuille 1, la numérotation continue au delà de 200 et la numérotation de la feuille 2 sera décalée.

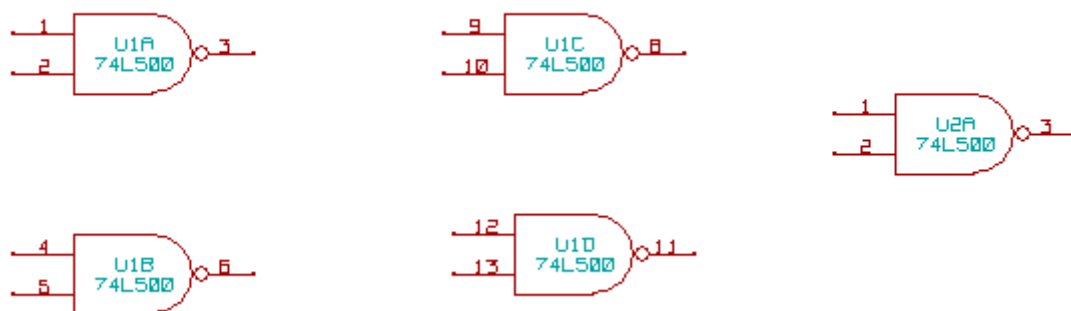
- **Démarrer à numéro de feuille\*1000 et utiliser le premier numéro libre.**  
La numérotation commence à 1001 pour la feuille 1, à 2001 pour la feuille 2...

## 7.2 - Exemples.

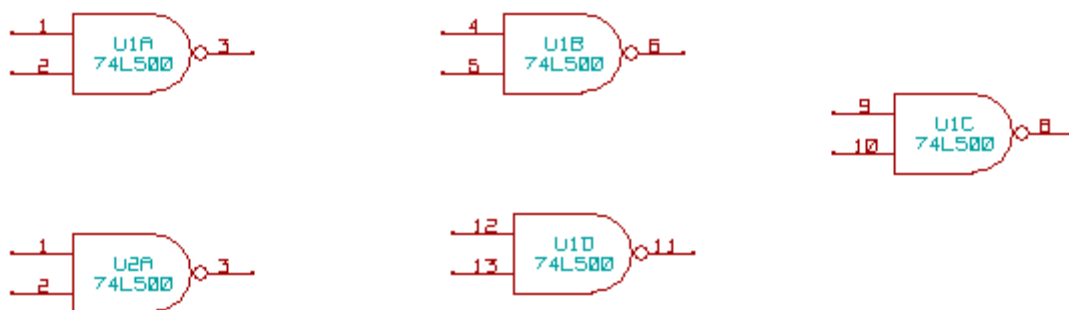
### 7.2.1 - Sens de Annotation:



Cet exemple montre un exemple de 5 éléments placés, mais non annotés.  
Après re-numérotation (annotation) automatique, voici le résultat:



Tri par X position



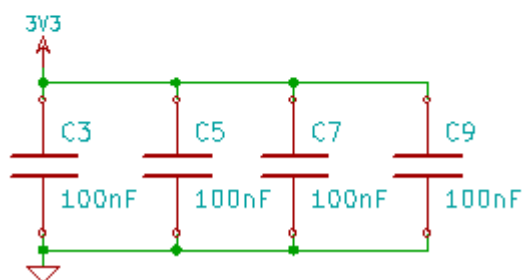
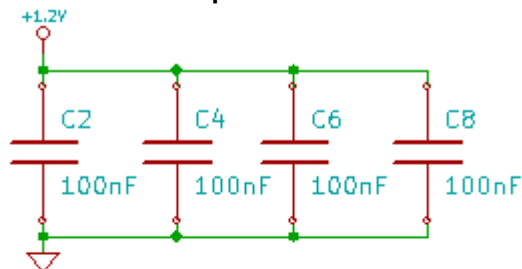
Tri par Y position

On voit que quatre portes 74LS00 ont été réparties dans un boîtier U1, et que la cinquième porte 74LS00 a donc été affectée au boîtier suivant, U2.

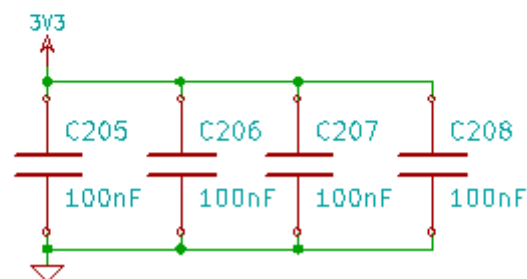
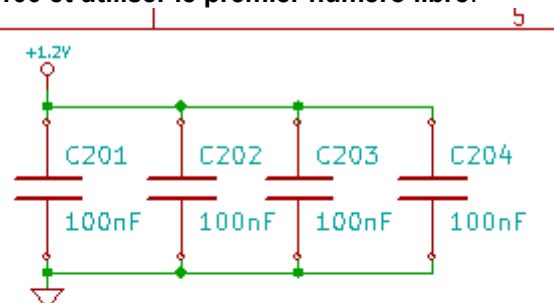
### 7.2.2 - Choix Annotation:

Voici également une numérotation de composants sur la **feuille 2**:

Option **Utiliser le premier numéro libre de la schématique**:



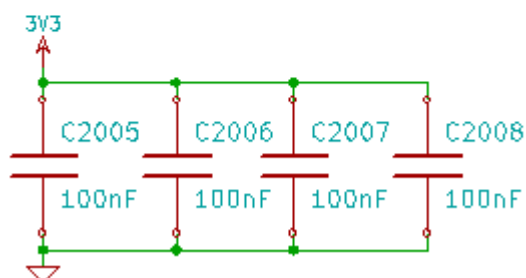
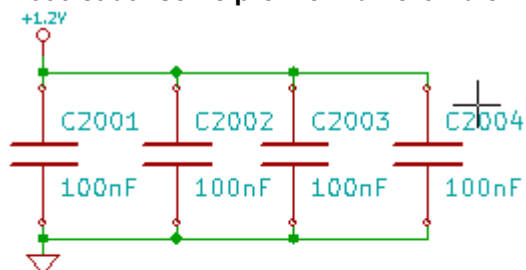
Option **Démarrer à numéro de feuille\*100 et utiliser le premier numéro libre**:





## Eeschema - Numérotation automatique ( ou Annotation).

Option **Démarrer à numéro de feuille\*1000** et utiliser le premier numéro libre:



## 8 - Le contrôle E.R.C.

### Table des matières

8 - Le contrôle E.R.C.....	1
8.1 - Rôle.....	1
8.2 - Utilisation.....	2
8.3 - Exemple de contrôle ERC.....	2
8.4 - Affichage des diagnostics:.....	3
8.5 - Alimentations et Power Flags:.....	4
8.6 - Configuration.....	4
8.7 - Génération du fichier de diagnostic.....	5

#### 8.1 - Rôle.

La fonction « **E**lectrical **R**ules **C**heck » permet un contrôle automatique d'un schéma créé.

Elle signale toute anomalie dans un schéma, telle que pins non connectées, symboles hiérarchiques non connectés, sorties en court-circuit...

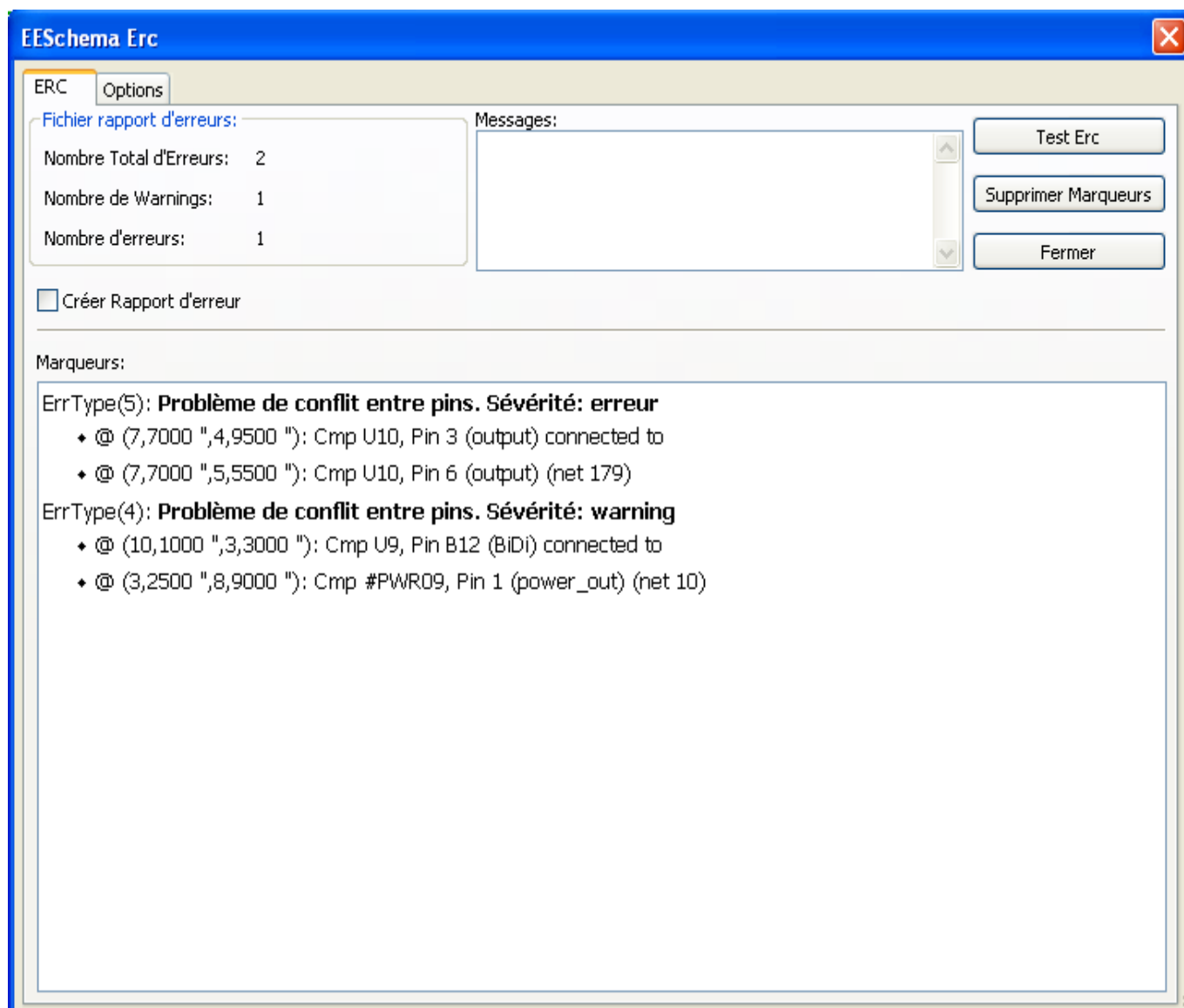
Naturellement, un contrôle automatique n'est pas infallible, et certaines erreurs détectées n'en seront pas forcément, et le logiciel permettant de déceler les erreurs de conception n'a pas encore été écrit.

Néanmoins, un tel contrôle est très précieux, car il permet de trouver beaucoup d'oublis et de petites erreurs.

En fait toutes les erreurs détectées doivent être ou corrigées, ou vérifiées et acceptées comme normales.

La qualité de la détection est aussi directement liée au soin que l'on aura pris lorsque les composants seront créés en librairie, pour la déclaration des propriétés électriques des pins.

Les anomalies détectées peuvent être définies comme « erreurs » ou « warnings ».



## 8.2 - Utilisation.



Le contrôle E.R.C. est lancé par l'icône

Si des erreurs sont décelées, des marqueurs seront placés sur les éléments concernés (pins, ou labels).

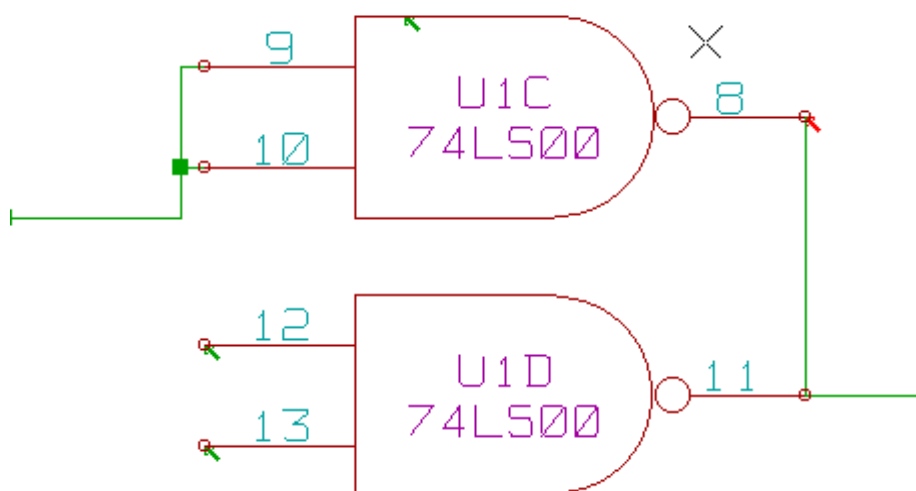
### Notes:

- Dans la fenêtre de ce dialogue, en cliquant sur un message d'erreur, on saute au marqueur correspondant en schématique.
- Dans la schématique, cliquer sur les marqueurs, avec le bouton droit de la souris, pour accéder au diagnostic correspondant.

L'effacement des marqueurs est aussi disponible dans ce menu.

## 8.3 - Exemple de contrôle ERC

Voici un exemple de contrôle :

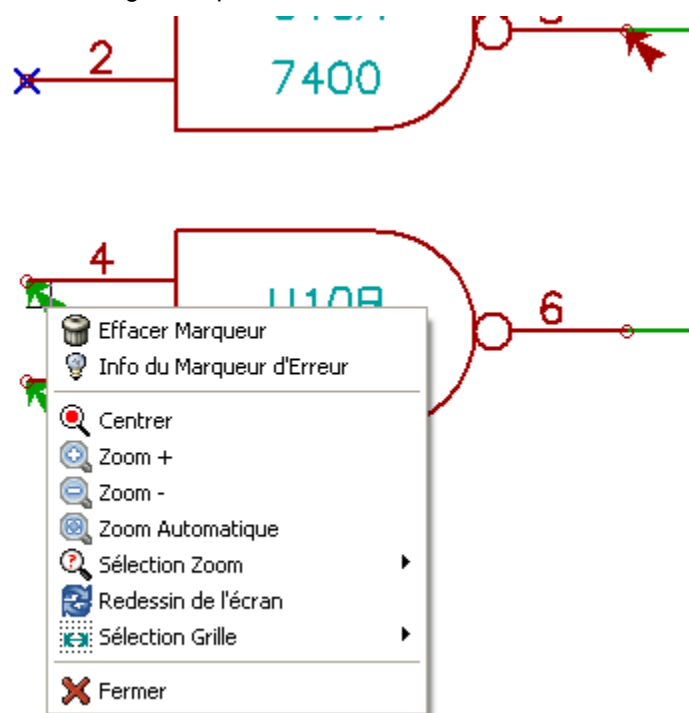


On peut y voir 4 erreurs:

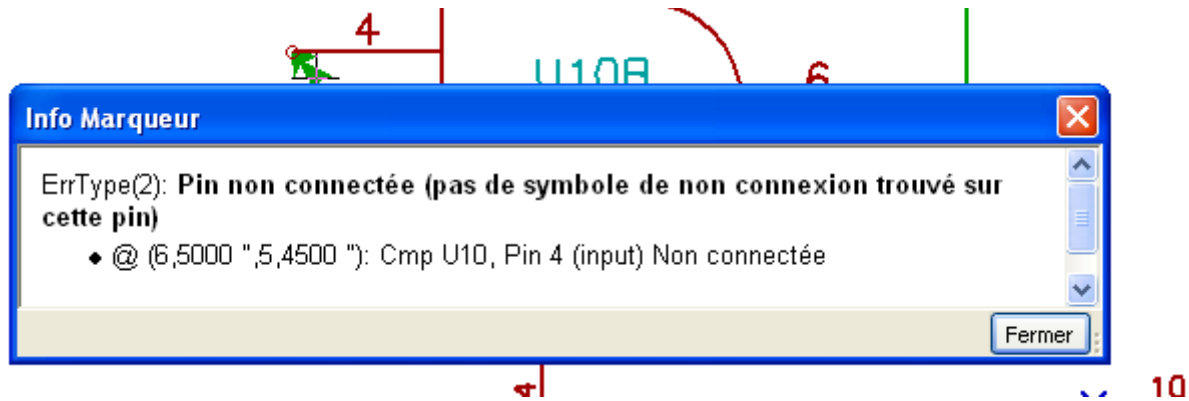
- Deux sorties reliées ensemble.
- Deux entrées « en l'air ».
- Une erreur sur une pin **invisible** d'alimentation (ici on n'a pas mis une sortie d'alimentation pour la piloter).

#### 8.4 - Affichage des diagnostics:

Si on clique (bouton droit) sur un marqueur, le diagnostic pourra être affiché.



et en sélectionnant **Info du Marqueur d'erreurs**:



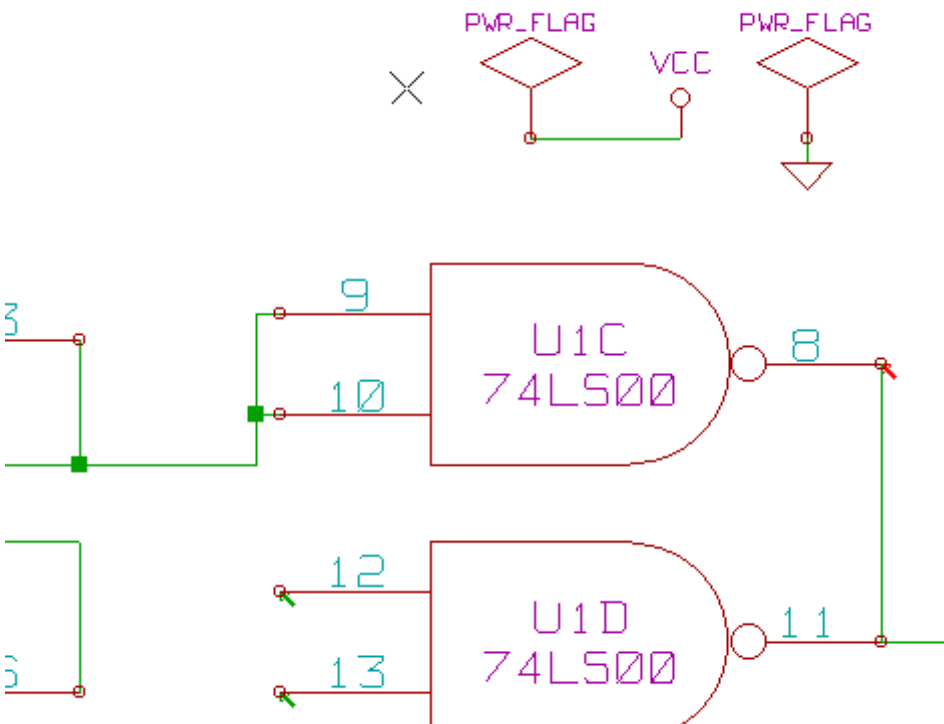
## 8.5 - Alimentations et Power Flags:

Il est courant d'avoir une erreur (type warning) sur les alimentations, alors que tout semble normal (voir exemple ci dessus).

Ceci est dû à ce que dans la plupart des schémas, les alimentations sont apportées par des connecteurs, et non par des **sources d'alimentations** (comme une sortie de régulateur, qui doit être du type électrique **Power out**).

Le contrôle ERC **ne détecte donc** généralement **pas de pins** du type **Power out** pour piloter ces lignes et les signale donc logiquement comme non pilotées (non pilotées par une source d'alimentation).

On conseille donc de placer sur de telles alimentations de placer un symbole d'alimentation « **PWR\_FLAG** » (qui symbolise en fait une sortie d'alimentation).



On voit que l'erreur précédente a disparu.

La plupart du temps, on doit connecter un pwr\_flag à la masse (ou aux masses) car usuellement les régulateurs on la sortie déclarée comme power out, mais jamais la masse (normalement déclarée comme power in). Aussi les masses apparaissent toujours comme non connectées à une source d'alimentation, si on ne connecte pas un pwr\_flag.

## 8.6 - Configuration

Le panneau **Options** permet de configurer la fonction E.R.C. pour adapter la génération des erreurs à chaque cas particulier :

Voici ce menu (matrice de sélection) :



Pour chaque cas de connexion sur la matrice de définition des erreurs, on peut définir une connexion comme « normale », « warning » ou « errreur), en cliquant une ou plusieurs fois sur le bouton correspondant au cas particulier de connexion.

### 8.7 - Génération du fichier de diagnostic.

On peut également générer un fichier liste des erreurs en activant l'option **Write erc report**. Le fichier généré a pour extension standard **.erc**. Voici un exemple :

ERC control (4/1/1997-14:16:4)

\*\*\*\*\* Sheet 1 (INTERFACE UNIVERSEL)

ERC: Warning Pin input Unconnected @ 8.450, 2.350

ERC: Warning Pin passive Unconnected @ 8.450, 1.950

ERC: Warning: Pin BiDi connected to Pin power (net 6) @ 10.100, 3.300

ERC: Warning: Pin power connected to Pin BiDi (net 6) @ 4.950, 1.400

>> Errors ERC: 4

9 - Génération de Netlistes.

Table des matières

9 - Génération de Netlistes..... 1

9.1 - Rôle..... 1

9.2 - Options et exécution (format PCBNEW)..... 1

9.3 - Exemple (-format PSPICE)..... 2

9.4 - Remarques :..... 4

9.4.1 - Précautions générales..... 4

9.4.2 - Cas de la netliste PSPICE..... 4

9.5 - Autres formats, utilisation de « plugins »..... 5

9.5.1 - Préparation de la fenêtre de dialogue:..... 5

9.5.2 - Format de la ligne de commande:..... 6

9.5.3 - Le convertisseur (plugin)..... 6

9.5.4 - Format du fichier netliste intermédiaire:..... 7

9.1 - Rôle.

Cette commande permet de générer le fichier netliste du schéma complet.  
Une netliste est un fichier qui décrit les connexions entre composants.  
On y trouve donc

- La liste des composants
- La liste des connexions, appelées équipotentielles.


La présentation varie selon les formats de netlistes. Quelquefois la liste des composant et celle des équipotentielles sont deux fichiers séparés.  
Cette netliste est fondamentale dans l'utilisation d'un logiciel de schématique, car c'est grâce à cette netliste que l'on fait le lien avec les autres logiciels de CAO électronique, comme :

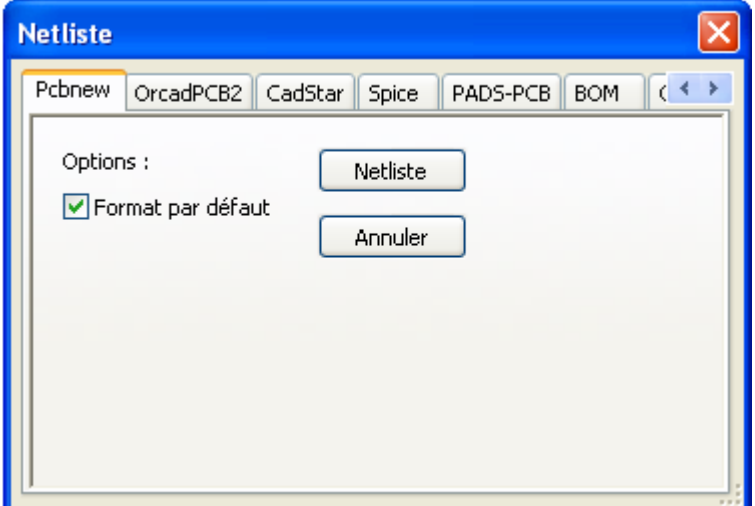
- Les logiciels de circuits imprimés.
- Les simulateurs.
- Les logiciels spécialisé de génération de CPLDs, et autres circuits intégrés programmables.

Eeschema génère plusieurs type de netliste :

- Une netliste au format PCBNEW (circuits imprimés).
- Une netliste au format ORCADPCB2 (circuits imprimés).
- Une netliste au format CADSTAR (circuits imprimés).
- Une netliste au format Spice, pour les simulateurs. (Spice en particulier, mais format utilisé par d'autres simulateurs).

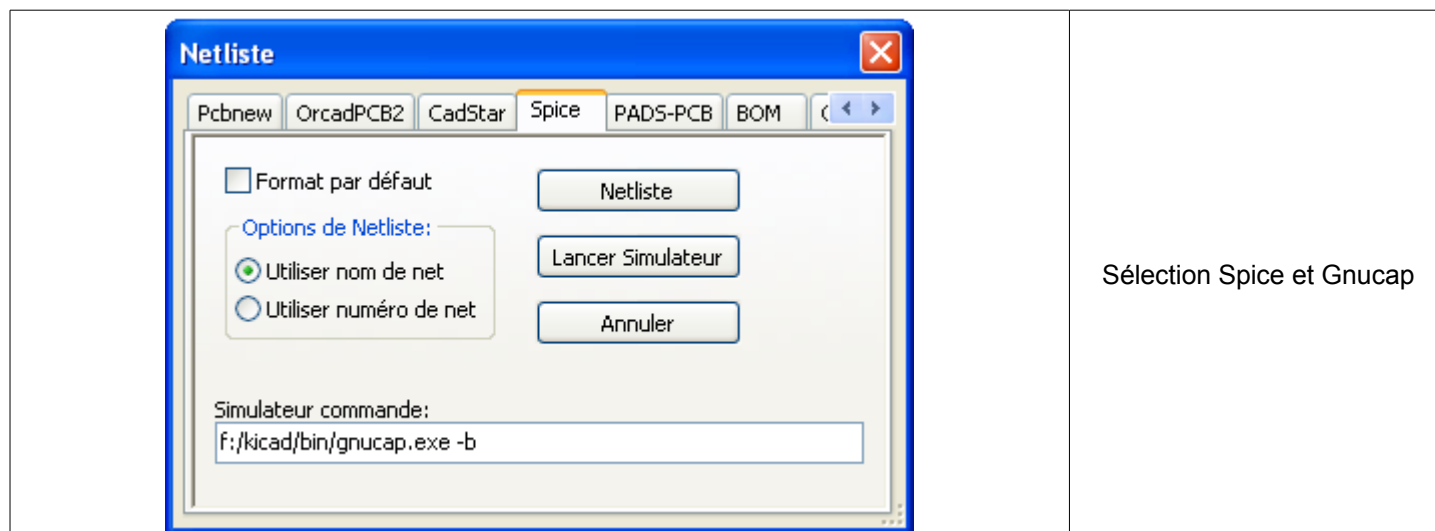
9.2 - Options et exécution (format PCBNEW).

Sélectionner l'outil  pour accéder à la fenêtre de gestion des netlistes:



Sélection Pcbnew





Sélection Spice et Gnucap

Les options sont :

- **Format par défaut (Default Format):** sélection du format PCBNEW comme format par défaut.
- **Utiliser nom / numéro de net.** (Spice) : Génération de netlistes avec soit des noms pour les équipotentielles (c'est plus clair) soit des numéros (les anciens Spices n'acceptent que des numéros)

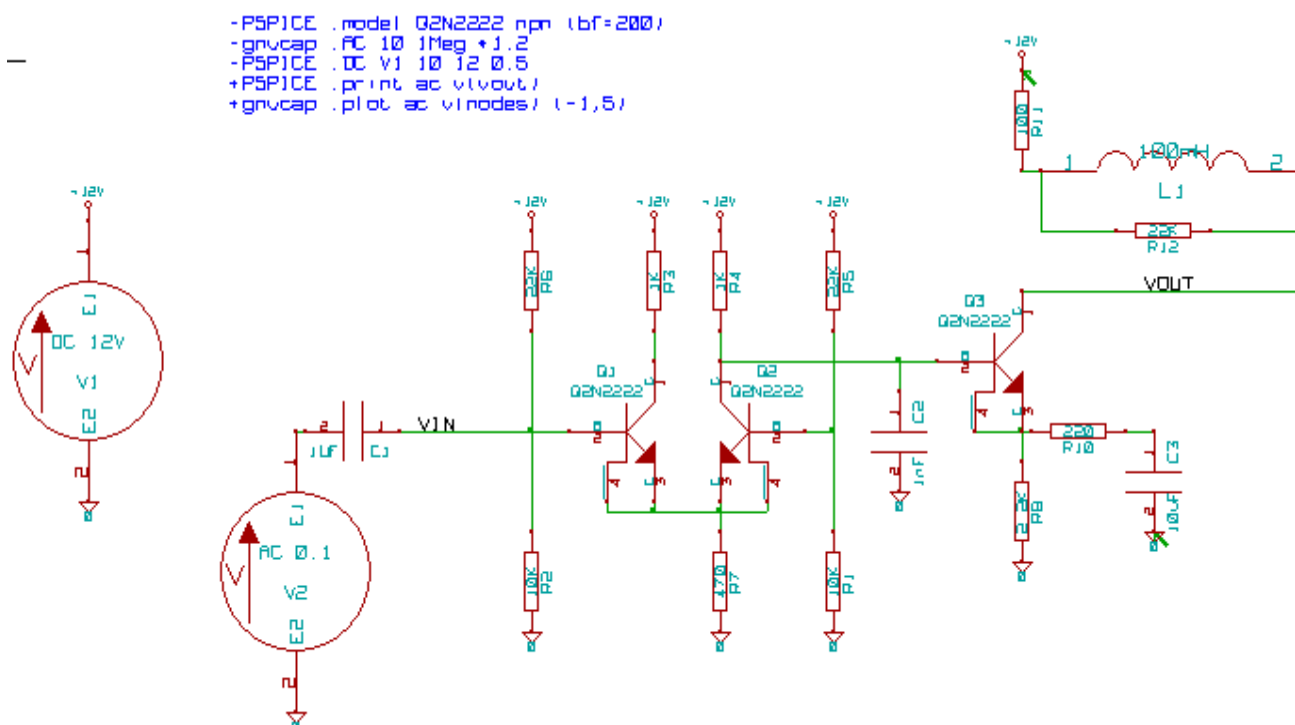
La netliste est générée par la commande **Netlist**, et un nom de fichier sera demandé.

**Remarque :**

Sur de gros projets, un calcul de netliste peut demander quelques minutes.

## 9.3 - Exemple (-format PSPICE)

Voici un schéma construit avec la librairie PSPICE :



En format PCBNEW la netliste générée est de la forme:

```

# EESchema Netlist Version 1.0 generee le 21/1/1997-16:51:15
(
  ( 32E35B76 $noname C2 1NF {Lib=C}
    
```

## EESchema - Génération de Netlistes.

```
(      1 0 )
(      2 VOUT_1 )
)
( 32CFC454 $noname V2 AC_0.1 {Lib=VSOURCE}
(      1 N-000003 )
(      2 0 )
)
( 32CFC413 $noname C1 1UF {Lib=C}
(      1 INPUT_1 )
(      2 N-000003 )
)
( 32CFC337 $noname V1 DC_12V {Lib=VSOURCE}
(      1 +12V )
(      2 0 )
)
( 32CFC293 $noname R2 10K {Lib=R}
(      1 INPUT_1 )
(      2 0 )
)
( 32CFC288 $noname R6 22K {Lib=R}
(      1 +12V )
(      2 INPUT_1 )
)
( 32CFC27F $noname R5 22K {Lib=R}
(      1 +12V )
(      2 N-000008 )
)
( 32CFC277 $noname R1 10K {Lib=R}
(      1 N-000008 )
(      2 0 )
)
( 32CFC25A $noname R7 470 {Lib=R}
(      1 EMET_1 )
(      2 0 )
)
( 32CFC254 $noname R4 1K {Lib=R}
(      1 +12V )
(      2 VOUT_1 )
)
( 32CFC24C $noname R3 1K {Lib=R}
(      1 +12V )
(      2 N-000006 )
)
( 32CFC230 $noname Q2 Q2N2222 {Lib=NPN}
(      1 VOUT_1 )
(      2 N-000008 )
(      3 EMET_1 )
)
( 32CFC227 $noname Q1 Q2N2222 {Lib=NPN}
(      1 N-000006 )
(      2 INPUT_1 )
(      3 EMET_1 )
)
)
#End
```

En format PSPICE, la netliste générée est la suivante :

\* EESchema Netlist Version 1.1 (Spice format) creation date: 18/6/2008-08:38:03

```
.model Q2N2222 npn (bf=200)
.AC 10 1Meg *1.2
.DC V1 10 12 0.5
```

```
R12 /VOUT N-000003 22K
R11 +12V N-000003 100
```

## Eeschema - Génération de Netlistes.

```
L1 N-000003 /VOUT 100mH
R10 N-000005 N-000004 220
C3 N-000005 0 10uF
C2 N-000009 0 1nF
R8 N-000004 0 2.2K
Q3 /VOUT N-000009 N-000004 N-000004 Q2N2222
V2 N-000008 0 AC 0.1
C1 /VIN N-000008 1uF
V1 +12V 0 DC 12V
R2 /VIN 0 10K
R6 +12V /VIN 22K
R5 +12V N-000012 22K
R1 N-000012 0 10K
R7 N-000007 0 470
R4 +12V N-000009 1K
R3 +12V N-000010 1K
Q2 N-000009 N-000012 N-000007 N-000007 Q2N2222
Q1 N-000010 /VIN N-000007 N-000007 Q2N2222
```

```
.print ac v(vout)
.plot ac v(nodes) (-1,5)

.end
```

### 9.4 - Remarques :

#### 9.4.1 - Précautions générales.

Beaucoup de logiciels exploitant les netlistes n'acceptent pas de blanc dans les noms de composants, de pins, d'équipotentielles ou autre.

Éviter systématiquement donc des blancs dans les labels, ou les noms et valeurs des composants ou de leur pins.

De même, certains caractères autres que des lettres et chiffres peuvent poser des problèmes.

Noter que cette limitation n'est pas liée à Eeschema, mais soit au format des netlistes qui peuvent alors devenir non interprétables, soit des logiciels qui exploitent ces netlistes.

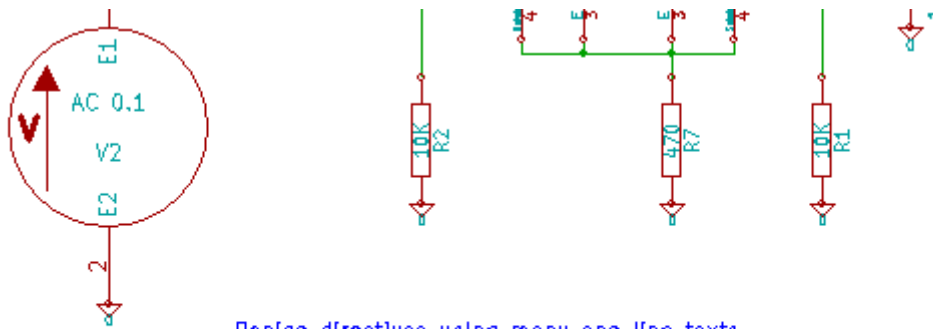
#### 9.4.2 - Cas de la netliste PSPICE.

Pour le simulateur Pspice, il faut inclure dans la netliste proprement dite quelques lignes de commandes (.PROBE, .AC ...).

Toute ligne de texte incluse dans le schéma et qui commence par le mot clé **-pspice** ou **-gnucap** (majuscule ou minuscule) sera insérée (sans le mot clé) dans la netliste, en tête de netliste.

Toute ligne de texte incluse dans le schéma et qui commence par le mot clé **+pspice** ou **-gnucap** sera insérée (sans le mot clé) dans la netliste, en fin de netliste.

Voici un exemple utilisant quelques textes sur une ligne et un texte multi-ligne:



Pspice directives using many one line texts

```
-PSPICE .model Q2N2222 npn (bf=200)
-gnucap .AC dec 10 1Meg *1.2
-PSPICE .DC V1 10 12 0.5
+PSPICE .print ac v(vout)
+gnucap .plot ac v(nodes) (-1.5)
```

Pspice directives using one multiline text:

```
+PSPICE .model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

Exemple : si l'on place dans le schéma le texte (et non pas un label !) :

**-PSPICE .PROBE**

la ligne .PROBE sera insérée dans la netliste.

Dans l'exemple ci dessus, trois lignes ont été insérées au début et deux à la fin avec cette technique.

Si l'on utilise un texte multi-ligne, **+pspice** or **+gnucap** doit être mis une seule fois:

```
+PSPICE .model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

crée les quatre lignes:

```
.model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

Par ailleurs, pour le simulateur Pspice, l'équipotentielle de masse doit avoir comme nom 0 (zéro).

## 9.5 - Autres formats, utilisation de « plugins »

Pour d'autres format de netlistes on peut ajouter des convertisseurs de netlistes.

Ces convertisseurs sont lancés automatiquement.

**Le chapitre 14 donne quelques exemples de convertisseurs.**

Principe:

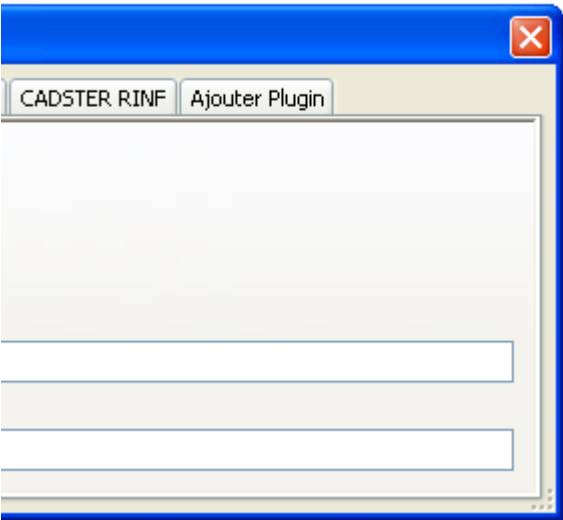
Le convertisseur est un fichier texte format xsl (mais d'autres langages sont possibles, comme PYTHON ou autres).

Lorsque l'on utilise le format xsl, un utilitaire ( xsltproc.exe ou xsltproc ) analyse un fichier intermédiaire généré par Eeschema, et le fichier convertisseur pour générer le fichier de sortie.

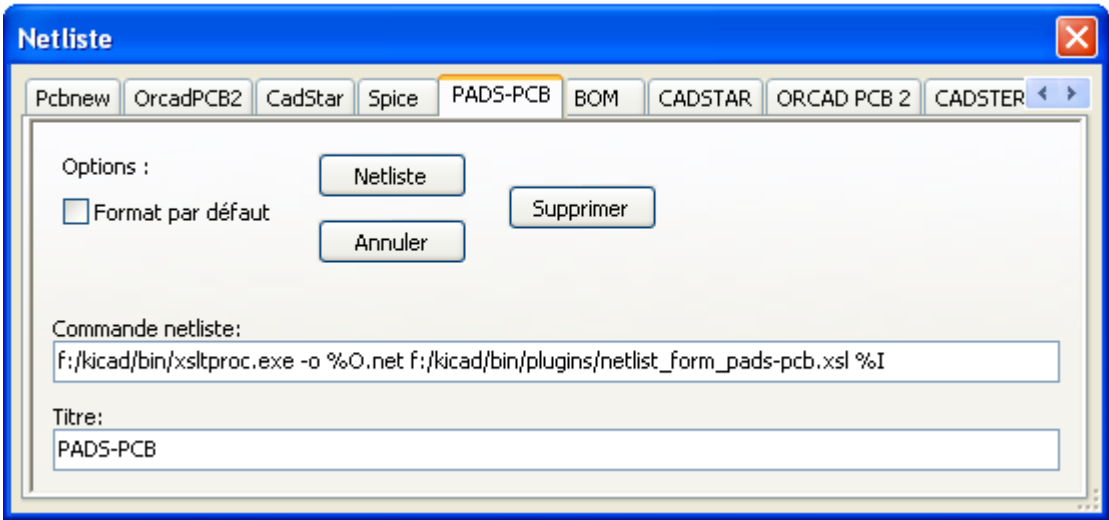
Dans ce cas le fichier convertisseur (une feuille de style) est très court et très simple a créer.

9.5.1 - Préparation de la fenêtre de dialogue:

On peut ajouter une nouvelle option par « Ajouter Plugin »



Voici par exemple l'initialisation du plugin PadsPcb



On doit spécifier:

- Un titre (nom du format netliste)
- Le plugin à lancer

En cliquant sur le bouton netliste :

1. Eeschema crée une netliste intermédiaire \*.tmp, par exemple test.tmp
2. Eeschema lance le plugin, qui lit test.tmp et crée test.net

9.5.2 - Format de la ligne de commande:

En voici un exemple, utilisant « xsltproc.exe » comme interpréteur de fichiers .xml et un fichier « /netlist\_form\_pads-pcb.xml » comme convertisseur:

**f:/kicad/bin/xsltproc.exe -o %O.net f:/kicad/bin/plugins/netlist\_form\_pads-pcb.xml %I**

Avec:

f:/kicad/bin/xsltproc.exe	utilitaire a lancer
-o %O.net	Fichier de sortie: %O sera remplacé part le nom de la netliste (nom du schéma « racine »)
f:/kicad/bin/plugins/netlist_form_pads-pcb.xml	Nom du fichier convertisseur en format xml.
%I	sera remplacé par le nom du fichier temporaire généré par Eeschema (*.tmp).

Pour un schéma appelé test.sch, la commande générée sera:

f:/kicad/bin/xsltproc.exe -o test.net f:/kicad/bin/plugins/netlist\_form\_pads-pcb.xsl test.tmp.

### **9.5.3 - Le convertisseur (plugin)**

C'est un programme aisé à écrire, car il ne fait que remettre en forme un fichier texte format xls (netliste intermédiaire). En outre, à partir de la netliste intermédiaire, on peut aussi générer des fichiers de liste de matériel. Dans le cas de l'utilisation de xsltproc comme convertisseur, seule une « feuille de style » est à écrire.

### **9.5.4 - Format du fichier netliste intermédiaire:**

**Le chapitre 14 décrit le format intermédiaire, l'utilisation de xsltproc, et donne quelques exemples de convertisseurs.**

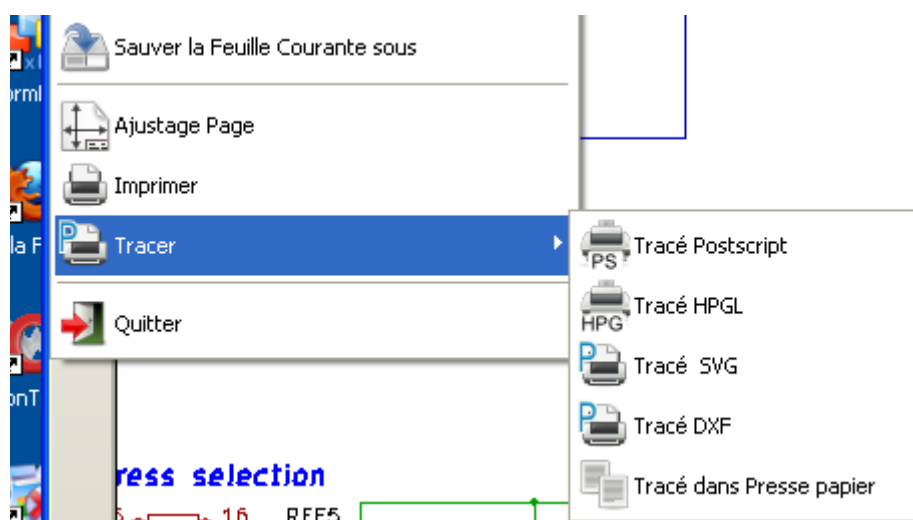
## 10 - Traçage et Impression

### Table des matières

10 - Traçage et Impression.....	1
10.1 - Généralités.....	1
10.2 - Commandes communes.....	1
10.3 - Tracer / Tracé HPGL).....	1
10.3.1 - Sélection de la dimension de la feuille de tracé.....	2
10.3.2 - Réglage des offsets de tracé.....	2
10.4 - Tracer / Tracé Postscript.....	3
10.5 - Tracer / Tracé SVG.....	4
10.6 - Tracer / Tracé DXF.....	4
10.7 - Imprimer.....	5

#### 10.1 - Généralités

L'accès se fait comme suit:



On peut générer des fichiers de tracé au format POSTSCRIPT, HPGL, SVG et DXF.  
L'impression directe sur imprimante est dans le menu Print.

#### 10.2 - Commandes communes:

La commande **Tout tracer** permet de tracer toute la hiérarchie ( un fichier est généré pour chaque schéma).  
La commande **Imprimer courant** ne génère que le fichier de tracé de la feuille courante.

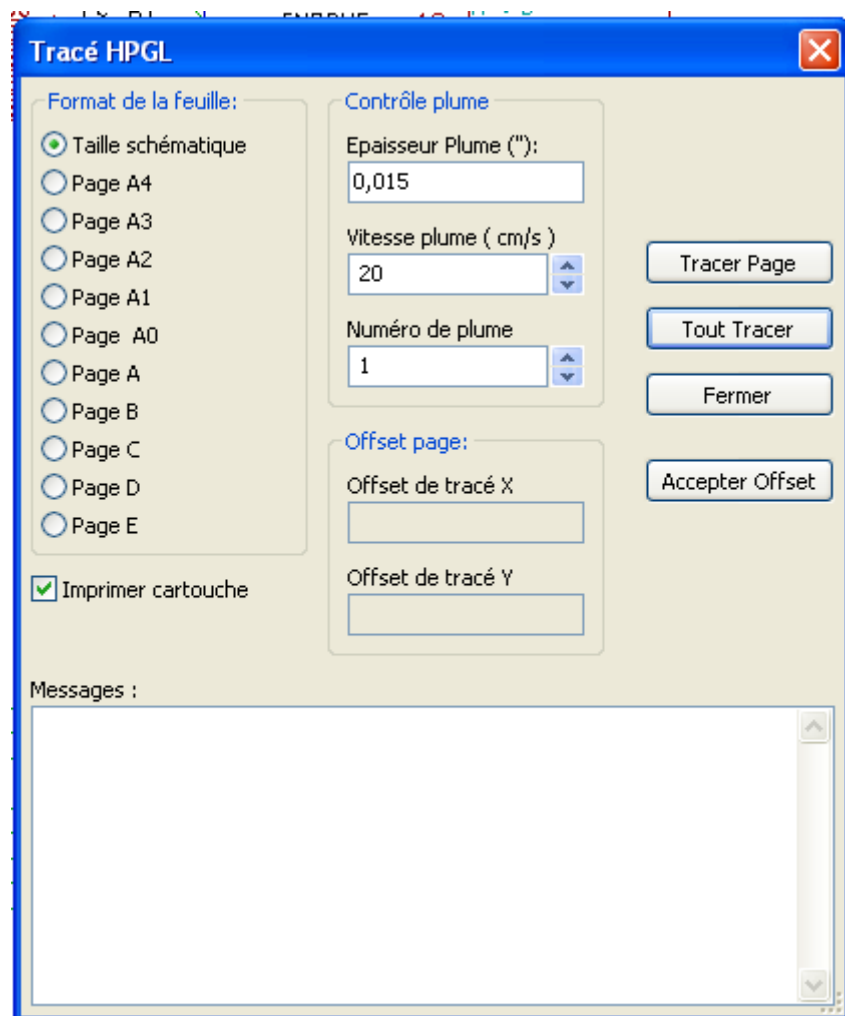
#### 10.3 - Tracer / Tracé HPGL)

Cette commande permet de générer les fichiers de tracé au format HPGL.

Dans ce format on peut définir :

- Le numéro de plume
- Son diamètre (en 0,001 pouce).
- La vitesse de tracé (en cm/s).
- La taille de la feuille de papier.
- Les offsets de tracé.

Voici la fenêtre de gestion :



Chaque fichier généré a pour nom le nom du fichier schématique correspondant et pour extension .plt.

### 10.3.1 - Sélection de la dimension de la feuille de tracé

L'option « **Taille schématique** » est normalement activée.

Ce sont alors les dimensions de la feuille de travail schématique en cours qui sont utilisées.

Le tracé est alors à l'échelle 1.

Si l'on active une des autres options(A4 à A0, ou A à E), ce seront alors ces dimensions qui seront utilisées, et l'échelle de tracé sera alors ajustée automatiquement en conséquence.

### 10.3.2 - Réglage des offsets de tracé

Pour toutes les dimensions normalisées, on peut ajuster des « offsets » de tracé, pour cadrer au mieux le dessin sur la feuille de tracé.

Du fait que les tables traçantes ont une origine des tracés soit au centre de la feuille soit au point bas gauche de la feuille, il est nécessaire de pouvoir introduire des offsets de tracé pour cadrer le tracé du dessin comme on le veut :

**D'une façon générale :**

- Pour les tables traçantes à origine des tracés au centre de la feuille, les offsets doivent négatifs et être réglés à environ la demi dimension correspondante de la feuille
- Pour les tables traçantes à origine des tracés au point bas gauche de la feuille, les offsets doivent proches de 0.

**Pour régler un offset :**

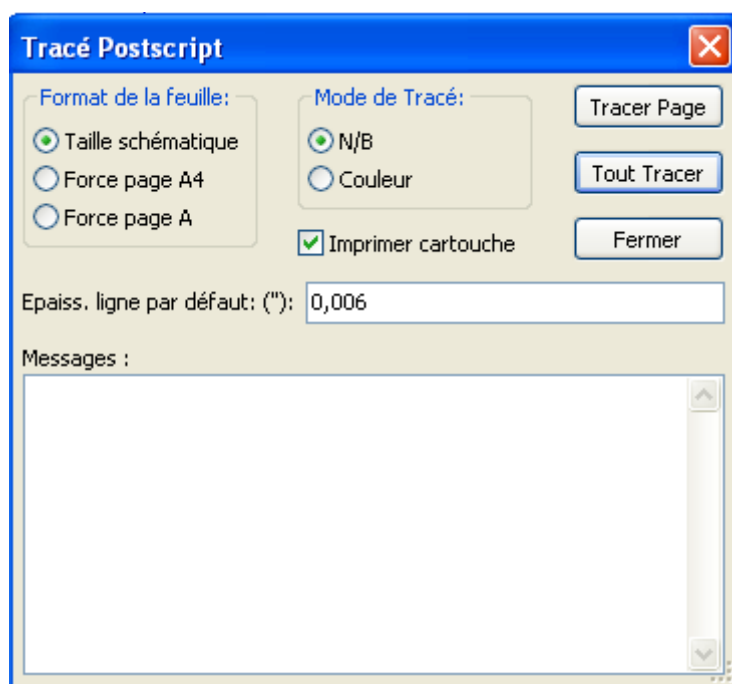
- Sélectionner le format de la feuille.
- Entrer les deux valeurs OffsetX et OffsetY.
- Valider la nouvelle valeur par le bouton **Accepter Offset**.

## 10.4 - Tracer / Tracé Postscript



## Eeschema - Traçage et Impression

Cette commande permet de générer des fichiers de tracé au format **Postscript**. Voici le menu :



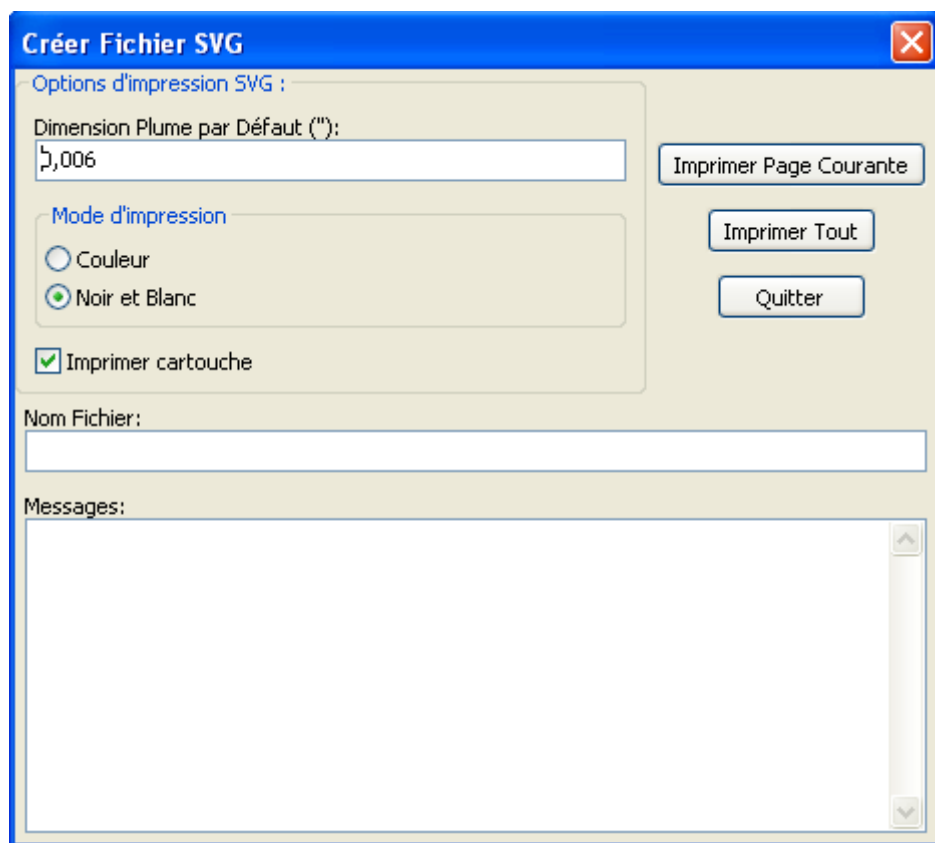
Chaque fichier a pour nom le nom du fichier schématique correspondant et pour extension .ps.

L'option Imprimer cartouche peut être désactivée.

Utile si l'on désire par exemple générer un fichier Postscript pour encapsulation (format .eps) pour insérer un schéma dans un traitement de texte.

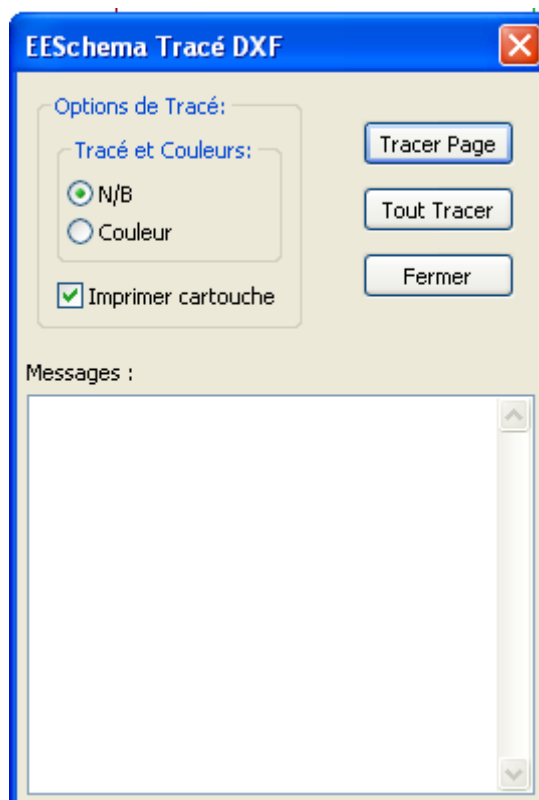
La fenêtre des message affiche le nom des fichiers créés.

### 10.5 - Tracer / Tracé SVG:



Permet de générer des fichiers au format vectoriel SVG.  
Chaque fichier a pour nom le nom du fichier schématique correspondant et pour extension .svg.

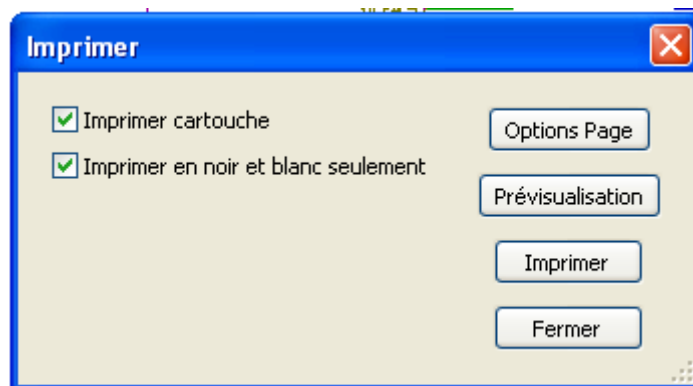
## 10.6 - Tracer / Tracé DXF



Permet de générer des fichiers au format d'échange DXF.  
Chaque fichier a pour nom le nom du fichier schématique correspondant et pour extension .dxf.

## 10.7 - Imprimer

Cette commande est aussi accessible par l'icône . Voici le dialogue:



L'option **Imprimer cartouche** active ou désactive l'impression du cartouche et des références de feuille.

L'option **Imprimer en noir et blanc seulement** permet de tracer en noir et blanc au lieu de l'impression en couleur.

Cette option est généralement nécessaire si on utilise une imprimante laser noir et blanc, car les éléments en couleur sont traduits en demi-teintes peu lisibles.

## Rubriques

11 - Gestion des Composants - Utilisation de LibEdit.....	1
11.1 - Généralités sur les librairies.....	2
11.1.1 - Librairies :.....	2
11.1.2 - Les menus de gestion.....	2
11.2 - Généralités sur les composants.....	2
11.3 - Accès aux composants pour édition.....	3
11.3.1 - Toolbar principal.....	4
11.3.2 - Sélection et maintenance d'une librairie.....	5
11.3.3 - Sélection et sauvegarde d'un composant.....	5
11.3.3.1 - Sélection.....	6
11.3.3.2 - Sauvegarde.....	6
11.3.3.3 - Transfert dans une autre librairie.....	6
11.3.3.4 - Annulation d'une édition de composant.....	7
11.4 - Création d'un composant.....	7
11.4.1 - Création d'un nouveau composant.....	7
11.4.2 - Création à partir d'un autre composant.....	8
11.4.3 - Édition des caractéristiques générales.....	9
11.4.4 - Sélection de l'unité et de la représentation.....	10
11.5 - Édition du graphisme.....	10
11.5.1 - Options d'appartenance des éléments graphiques.....	11
11.5.2 - Les éléments graphiques géométriques.....	12
11.5.3 - Les éléments graphiques type textes.....	12
11.6 - Création et édition des pins.....	12
11.6.1 - Notions générales sur les pins.....	13
11.6.2 - Boîtiers à parts multiples et double représentation.....	13
11.6.3 - Pins : option de base.....	14
11.6.4 - Pins : Définition des caractéristiques.....	14
11.6.5 - Formes des pins.....	16
11.6.6 - Type électrique des pins.....	16
11.6.7 - Modifications globales des pins.....	16
11.6.8 - Pins pour parts multiples et doubles représentations.....	16
11.7 - Édition des champs.....	17
11.8 - Création des symboles d'alimentation.....	19

### 11 - Gestion des Composants - Utilisation de LibEdit.

## Table des matières

11 - Gestion des Composants - Utilisation de LibEdit.....	1
11.1 - Généralités sur les librairies.....	2
11.1.1 - Librairies :.....	2
11.1.2 - Les menus de gestion.....	2
11.2 - Généralités sur les composants.....	2
11.3 - Accès aux composants pour édition.....	3
11.3.1 - Toolbar principal.....	4
11.3.2 - Sélection et maintenance d'une librairie.....	5
11.3.3 - Sélection et sauvegarde d'un composant.....	5
11.3.3.1 - Sélection.....	6
11.3.3.2 - Sauvegarde.....	6
11.3.3.3 - Transfert dans une autre librairie.....	6
11.3.3.4 - Annulation d'une édition de composant.....	7
11.4 - Création d'un composant.....	7
11.4.1 - Création d'un nouveau composant.....	7
11.4.2 - Création à partir d'un autre composant.....	8
11.4.3 - Édition des caractéristiques générales.....	9
11.4.4 - Sélection de l'unité et de la représentation.....	10
11.5 - Édition du graphisme.....	10

11.5.1 - Options d'appartenance des éléments graphiques.....	11
11.5.2 - Les éléments graphiques géométriques.....	12
11.5.3 - Les éléments graphiques type textes.....	12
11.6 - Création et édition des pins.....	12
11.6.1 - Notions générales sur les pins.....	13
11.6.2 - Boîtiers à parts multiples et double représentation.....	13
11.6.3 - Pins : option de base.....	14
11.6.4 - Pins : Définition des caractéristiques.....	14
11.6.5 - Formes des pins.....	16
11.6.6 - Type électrique des pins.....	16
11.6.7 - Modifications globales des pins.....	16
11.6.8 - Pins pour parts multiples et doubles représentations.....	16
11.7 - Édition des champs.....	17
11.8 - Création des symboles d'alimentation.....	19

## 11.1 - Généralités sur les librairies

### 11.1.1 - Librairies :

Tous les composants utilisables dans une schématique sont décrits dans les bibliothèques ou librairies de composants. De façon à pouvoir avoir une gestion raisonnablement simple des ces composants, on utilise plusieurs librairies, chacune regroupant des composants par thèmes (par fonctions, ou constructeurs ..).


Le Menu gestionnaire de librairie permet la maintenance des librairies : création de librairies nouvelles, ajout et suppression de composants dans les librairies, et transfert (avec ou sans édition) d'un composant d'une librairie à une autre.

Il permet aussi naturellement de visualiser rapidement les composants d'une librairie.


### 11.1.2 - Les menus de gestion

Il y a deux menus de gestion des librairies :

- **ViewLib** qui permet uniquement de visualiser les composants, mais avec des facilités pour accéder rapidement aux composants successif d'une librairie

Accès à ce menu par l'icône .

- **LibEdit** qui est le vrai menu de gestion des librairies et des composants.

Accès à ce menu par l'icône .

## 11.2 - Généralités sur les composants

Un **composant** en librairie est constitué :

- De sa représentation graphique (lignes, cercles, textes).
- De pins qui, outre leur graphisme qui doit respecter les standards usuels (pin simple, ou représentant une entrée d'horloge, ou complémentée, ou active par niveau bas...) décrivent des propriétés électriques, utilisées par la fonction E.R.C.
- De champs (textes) tels que référence, valeur, nom du module correspondant pour des implantations de circuits imprimés...

Il peut aussi avoir des alias, c'est à dire plusieurs noms (ainsi un 7400 peut aussi avoir plusieurs alias comme 74LS00, 74HC00, 7437, car tous ces composants sont identiques au point de vue schéma et implantation).

Utiliser des alias est une méthode très intéressante pour avoir des librairies complètes mais compactes et (relativement) vite construites.

### Construire un composant c'est:


- Définir des propriétés générales :
  - a t'il des parts multiples, et combien.
  - a t'il une double représentation (dite de Morgan, et dans Eeschema appelée représentation normale et convertie).
- Tracer son graphisme (hormis les pins) à l'aide de lignes, rectangles, cercles, polygones et textes.
- Ajouter les pins, en ayant soin de bien définir non seulement le graphisme, nom et le numéro de pin, mais aussi ses propriétés électriques (entrée, sortie, 3 états, alimentations...).
- Ajouter des alias si d'autres composants lui sont identiques en représentation graphique et en brochage (ou en supprimer s'il y a création par copie d'un autre composant).

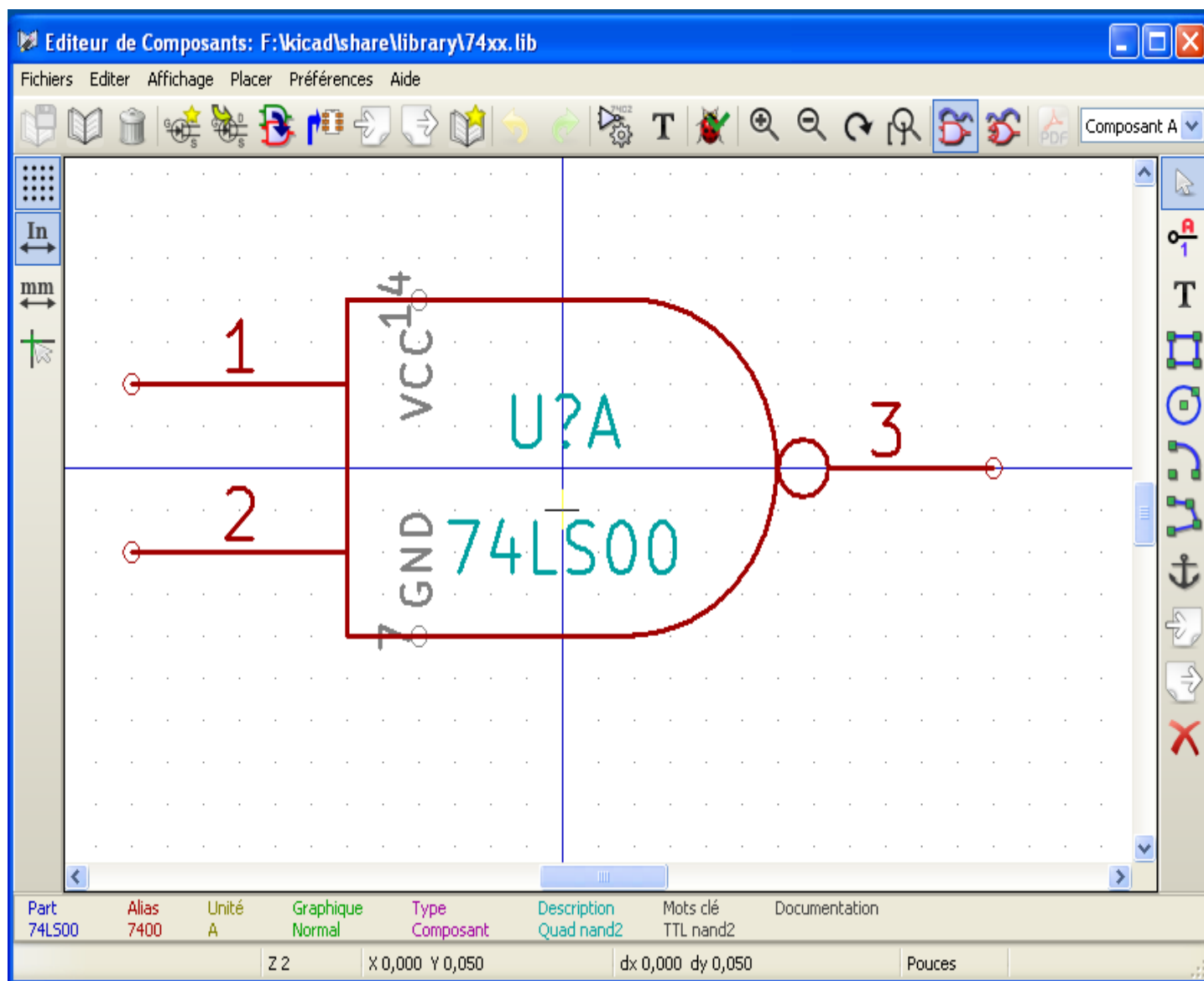
## Eeschema - Gestion des Composants - Utilisation de LibEdit.

- Ajouter éventuellement des champs (essentiellement, mais c'est facultatif, le nom du module pour le logiciel de circuits imprimés) et / ou définir leur visibilité.
- Le documenter.
- Le sauver dans la librairie de son choix.

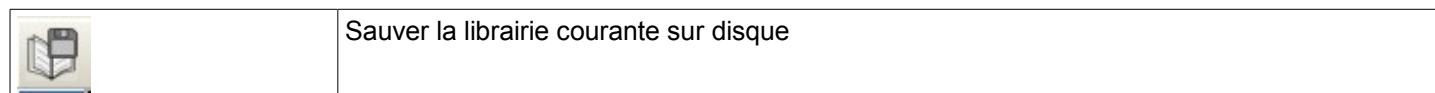
### 11.3 - Accès aux composants pour édition



Cliquer sur l'outil  pour ouvrir **Libedit**, la fenêtre d'édition de composants et gestion des librairies. Libedit se présente comme suit:




#### 11.3.1 - Toolbar principal



	Sélection de la librairie courante
	Effacer un composant dans la librairie courante
	Créer un nouveau composant
	Charger un composant de la librairie courante pour édition
	Créer un nouveau composant à partir du composant courant
	Sauver le composant édité dans la librairie courante en mémoire. Le fichier librairie sur disque n'est pas modifié.
	Importer un seul composant.
	Exporter le composant édité.
	Créer une nouvelle librairie et y sauver le composant.
	Commandes Défaire / Refaire
	Éditer les propriétés du composant.
	Éditer les champs du composant: Référence, valeur/nom en librairie et champs optionnels
	Test pour pins dupliquées ou superposées.
	Afficher représentation normale ou convertie (De Morgan)
	Afficher fichier de documentation associé (s'il existe)
	Sélection part (dans les composants multi parts)
	Sélection alias (pour les composants avec alias)
	Édition des pins: édition indépendante pour les pins des parts et représentation De Morgan

### 11.3.2 - Sélection et maintenance d'une librairie

La sélection de la librairie courante est possible grâce à l'icône , qui affiche pour sélection la liste des librairies disponibles.

## Eeschema - Gestion des Composants - Utilisation de LibEdit.

Lorsque un composant sera chargé ou sauvegardé, il le sera dans cette librairie.  
Le nom de ce composant en librairie est le champ « Valeur ».

### Remarque:

Pour qu'une librairie soit disponible, il faut qu'elle ait été chargée par Eeschema auparavant.



La librairie courante peut être sauvegardée sur disque après modification, par



Un composant peut être supprimé de cette librairie par

### 11.3.3 - Sélection et sauvegarde d'un composant

Lorsque on édite un composant, on ne travaille jamais sur le composant réellement en librairie, mais sur sa copie, en mémoire de travail.

On peut ainsi annuler facilement une édition.

Un composant peut en fait provenir d'une librairie, ou provenir d'une sauvegarde d'un ancien composant.

Une fois chargé, il sera affiché à l'écran.

#### 11.3.3.1 - Sélection



L'icône affiche la liste des composants disponibles, pour sélection et chargement du composant voulu.

#### Remarque 1 :

Si l'on sélectionne un **alias** d'un composant, c'est le **composant principal** qui sera chargé (Eeschema affiche toujours le nom du composant réellement chargé).

- La liste des alias d'un composant est toujours chargée avec ce composant, et elle peut donc être éditée.
- Lorsque l'on veut éditer plus particulièrement un **alias**, cet alias doit être sélectionné dans

la fenêtre du toolbar:



- **Le premier élément de la liste est le composant racine.**

#### Remarque 2 :



Alternativement, la commande **Import** ( ) permet de charger un composant qui a été



préalablement sauvé par la commande **Export** ( ).

#### 11.3.3.2 - Sauvegarde

Après modification, un composant peut être sauvegardé dans la librairie courante, ou dans une nouvelle librairie, ou encore exporté dans un fichier de sauvegarde.



Pour la sauvegarde en librairie courante, exécuter la commande **Update** ( ).

Cependant la mise à jour ne se fait qu'en **mémoire** de l'ordinateur (On peut ainsi dans la schématique, juger le composant).



Si l'on veut sauvegarder complètement le composant, il faudra ensuite activer l'outil qui modifiera le fichier librairie sur disque.

Si l'on veut créer une nouvelle librairie pour y placer ce composant, utiliser la commande **NewLib**




Le nom de la nouvelle librairie à créer sera demandé.

### Remarque:

Si l'on veut pouvoir y accéder, ne pas oublier de l'ajouter à la liste des librairies à charger par Eeschema (voir configuration de Eeschema).





Enfin on peut utiliser la commande **Export** () pour créer un fichier qui ne contiendra que ce composant (ce fichier est un fichier librairie standard, mais qui ne contient qu'un seul composant). En fait **NewLib** et **Export** sont deux commandes identiques, la première propose par défaut la création dans le répertoire des librairies, et la deuxième dans le répertoire de travail de l'utilisateur.

#### 11.3.3.3 - Transfert dans une autre librairie

On peut très facilement copier un composant d'une librairie source à une librairie destination grâce aux commandes suivantes :

- Sélectionner la librairie source comme librairie courante .

- Charger le composant à transférer .

- Sélectionner la librairie destination comme librairie courante. .

- Sauver le composant en mémoire .

- Sauver la librairie ainsi modifiée .

#### 11.3.3.4 - Annulation d'une édition de composant

Le composant que l'on édite n'est que la copie de travail du composant réellement en librairie.

Tant que l'on ne l'a pas sauvegardé en mémoire, il suffit de le recharger (ou d'en recharger un autre) pour annuler les éditions faites sur ce composant.

Si on l'a déjà sauvegardé en mémoire, et si on n'a pas encore sauvé le fichier librairie sur disque, on peut quitter et redémarrer Eeschema, pour relire la librairie.

### 11.4 - Création d'un composant

#### 11.4.1 - Création d'un nouveau composant



Un nouveau composant peut être créé par la commande **NewPart** ()

Il sera demandé le nom sous lequel ce composant sera placé en librairie (ce nom est aussi le champ valeur pour Libedt et est utilisé comme valeur par défaut pour le champ valeur en schématique), la référence (U, IC, R...), le nombre de parts par boîtier (par exemple un composant type 7400 a 4 parts par boîtiers) et s'il y a une représentation convertie, type « de Morgan ».

Si le champ référence est laissé vide, la référence sera supposée être « U ».

Toutes ces données peuvent être changées par la suite, mais il est préférable de les fixer au début de la construction du composant.

**Propriétés du Composant**

**Options Générales**

Nom du Composant:

Référence par défaut

Nombre de parts par boîtier

☐ Créer un composant avec une forme alternative (DeMorgan)

☐ Créer composant comme symbole d'alimentation

☐ Les parts du boîtier sont verrouillées (ne peuvent pas être interchangeables)

**Parametres Globaux des Pins**

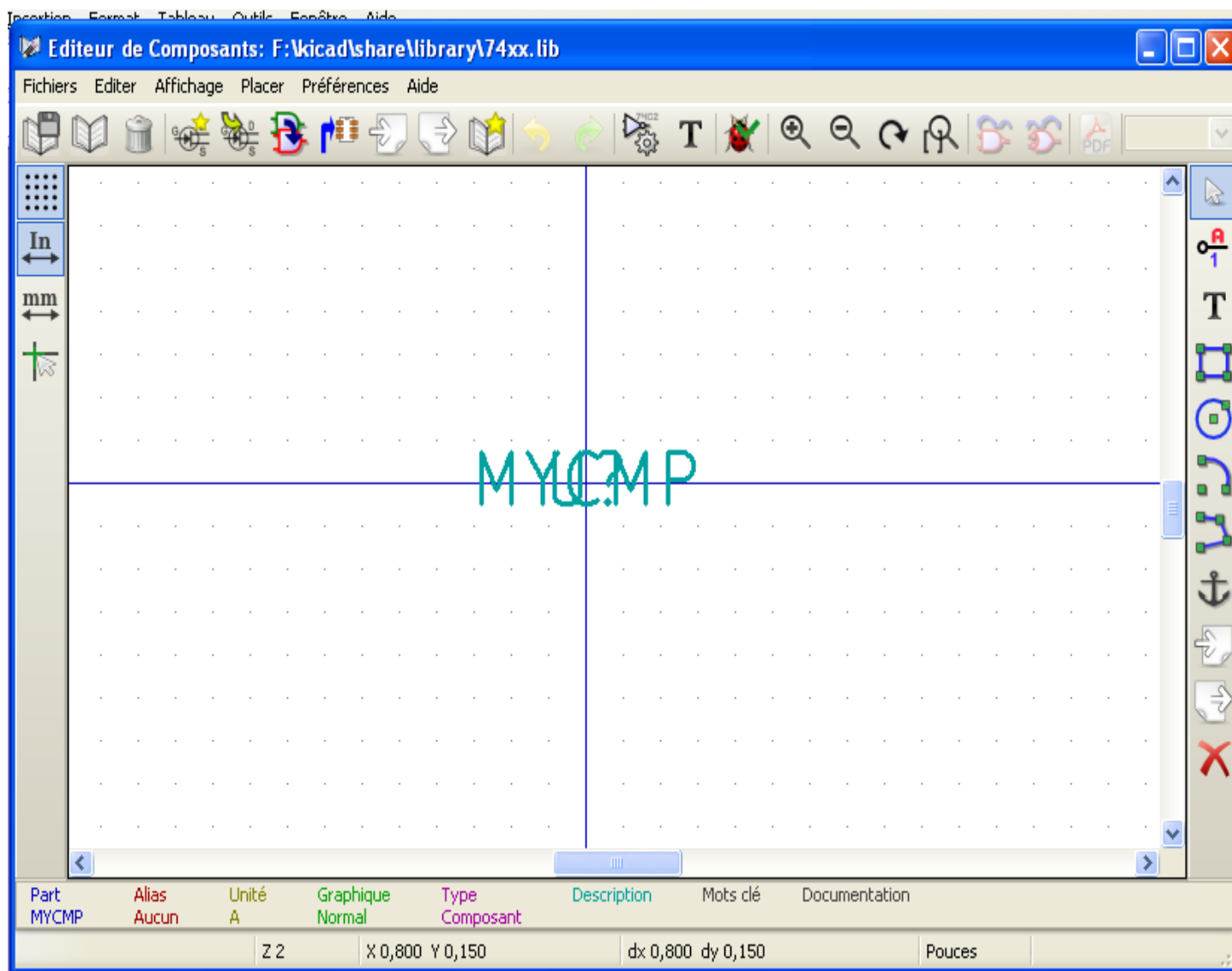
Offset de position du texte de la pin    mils

☒ Afficher le numéro de pin

☒ Afficher le Nom de Pin

☒ Nom de pin à l'intérieur

L'amorce d'un composant est celle ci :



### 11.4.2 - Création à partir d'un autre composant

Lorsque un composant ressemble fortement à un autre composant, il est souvent rentable de charger cet autre composant, et de le modifier.

**Réaliser les étapes suivantes :**

- Charger le composant qui va servir de modèle



- Cliquer sur

On peut aussi modifier son nom en cliquant sur le nom (double click ou souris, bouton de droite) sélectionner Édition du Champ et éditer le texte « Valeur »).

Un nouveau nom sera demandé.

- Si le modèle a des alias, il sera demandé si les alias dupliqués doivent être supprimés de la liste du nouveau composant. Si la réponse est « non » le nouveau composant ne sera pas créé.
- Éditer le nouveau composant.

- Sauver le nouveau composant ainsi créé dans la librairie active par



ou sauver dans une nouvelle librairie par



ou (if si vous voulez sauver ce composant dans une autre librairie déjà existante) sélectionner l'autre librairie et sauver le nouveau composant.

- Sauver la librairie sur le disque par Sauver Librairie:



### 11.4.3 - Édition des caractéristiques générales

Les caractéristiques générales sont :

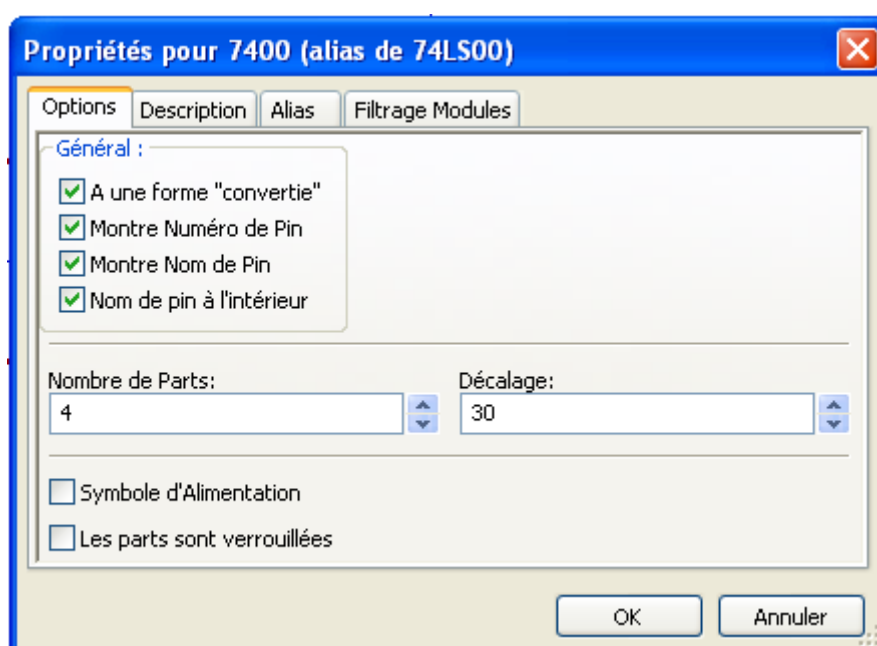
- Le nombre de parts par boîtier
- La présence ou non de la représentation convertie.
- La documentation associée.
- La mise a jour des différents champs.

Ces caractéristiques devraient être correctes, car demandées à la création du composant, ou provenir du composant modèle.



Si cependant elles doivent être modifiées, il faut appeler la commande d'édition

La fenêtre d'édition est la suivante :



Les **options importantes** définissant les propriétés générales sont :

**Nombre de parts** pour définir le nombre de parts par boîtier

**A une forme convertie** : si actif, le composant a une double représentation.

**Il est important que ces deux paramètres soient correctement définis, car lorsque l'on créera ou éditera les pins, les pins correspondantes de toutes les parts seront éditées ou créées ensemble.**

Si l'on augmente le nombre de parts après création/édition des pins, il y aura un travail supplémentaire introduit par cette augmentation.

Néanmoins, il est possible de modifier à tout moment ces paramètres.

Les options graphiques:

- **Montre numéro de Pin** et
- **Montre Nom de Pin**

définissent la visibilité des deux textes numéro de pin et nom de pin (ces textes seront visibles si les options correspondantes sont actives).

L'option :

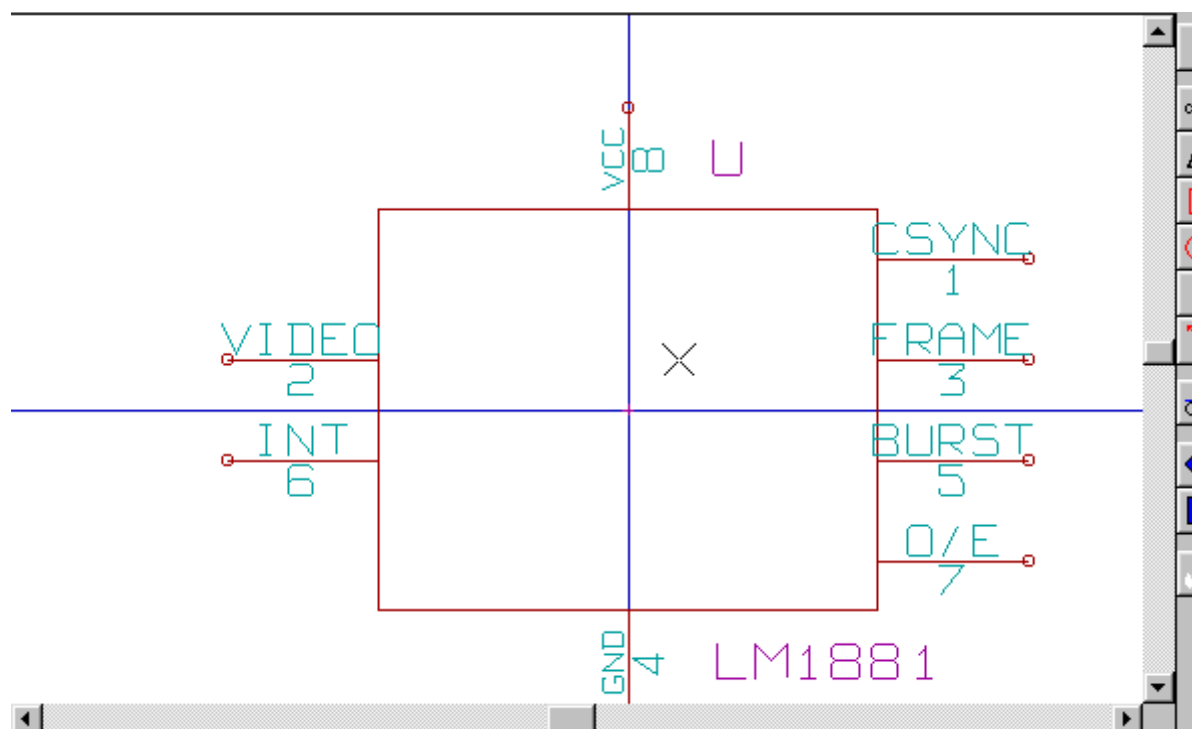
- **Nom de Pin à l'intérieur**

définit la position du texte nom de pin : ce texte sera affiché à l'intérieur du composant si l'option est active).

Dans ce cas le paramètre **Décalage (Skew)** définit le « recul » du texte vers l'intérieur.

Une valeur de 30 à 40 (en 1/1000 pouce) est raisonnable.

L'exemple ci dessous montre le même composant, avec l'option Pin Inside inactive (remarquer la position des noms et des numéros de pin) :



#### Options:

##### Symbole d'alimentation:

Doit être activé pour les composants symboles d'alimentation (VCC, GND ...)

Cela empêche la modification dans la schématique de la valeur du composant (ce qui pourrait provoquer des erreurs, car c'est le nom de la pin du composant qui compte).

Un tel composant n'apparaît pas non plus en liste des composants.

##### Les parts sont verrouillées.



Concerne les parts multiples par boîtier.

Si active, Eeschema ne cherche pas à changer la sélection des parts à l'intérieur d'un boîtier.

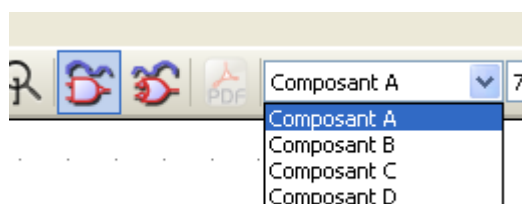
Doit être activée quand les parts ne sont pas interchangeables.

### 11.4.4 - Sélection de l'unité et de la représentation

Lors de l'édition des différents éléments du composant, et si ce composant est à parts ou à représentation multiples, on devra sélectionner les différentes parts ou représentations de ce composant.

Pour la sélection de la représentation: cliquer  ou .

Pour la sélection de la part



### 11.5 - Édition du graphisme

Le toolbar vertical permet de placer tous les éléments d'un composant:



Pour dessiner un composant, on dispose des éléments graphiques suivants :

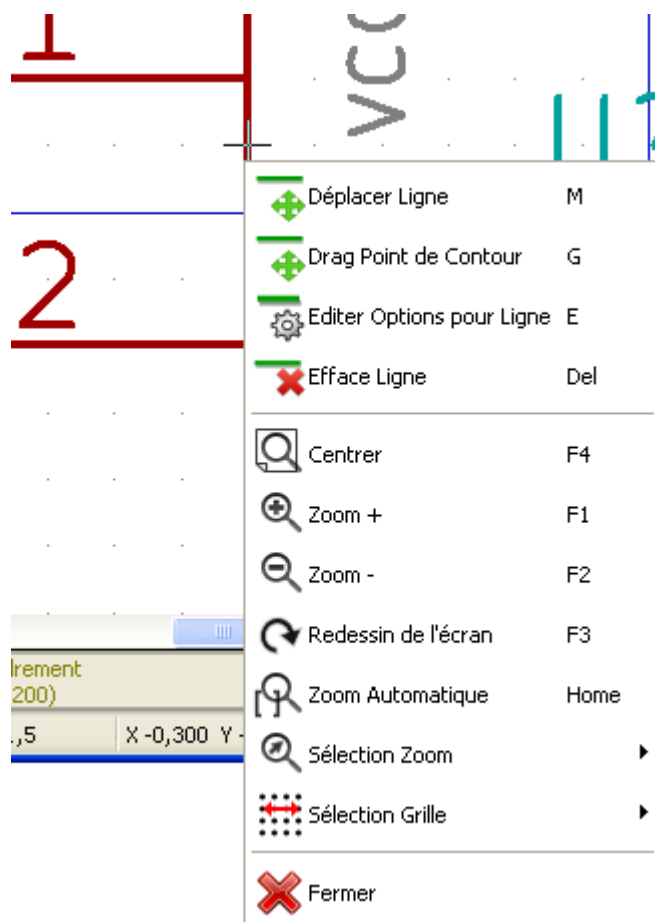
- Lignes (et polygones simples ou remplis).
- Rectangles
- Cercles
- Arcs de cercle.
- Textes (autres que champs et textes de pins).

Les pins et les champs (valeur, référence) sont traités différemment, car ce ne sont pas des éléments de pur graphisme.

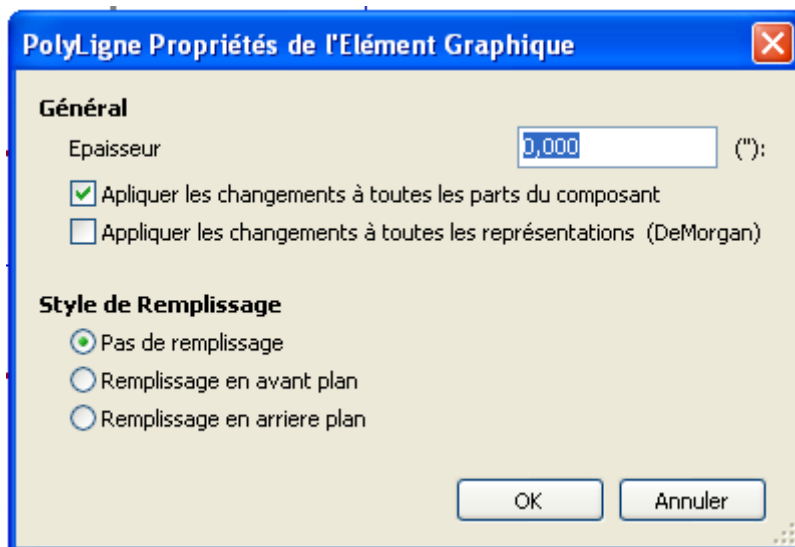
### 11.5.1 - Options d'appartenance des éléments graphiques

Chaque élément graphique peut être défini comme commun ou spécifique soit à un type de représentation (Normale ou convertie) soit aux parts différentes d'un composant.

Le menu d'options est accessible en cliquant avec le bouton droit sur l'élément concerné (ici une ligne):



ou encore en double-cliquant sur cet élément:



Les options normales sont, pour l'élément graphique:

- **Appliquer les changements à toutes les parts du composant activé**, car généralement les différentes parts d'un composant ont même graphique, et il suffit donc de dessiner une seule fois une part.
- **Appliquer les changements à toutes les re^présentations (DeMorgan) désactivé**, car on introduit une double représentation pour avoir un graphisme différent à chaque représentation.

Il faudra alors dessiner le graphisme pour chaque représentation.

Pour les éléments du type « polygone » (lignes tracées successivement) l'option **Plein** permet de générer un polygone rempli opaque, et Fond Plein un polygone rempli non opaque (les textes mis dans ce polygone sont visibles).

Cependant, on peut donc ainsi traiter le cas (heureusement rare) de composants multi parts, ayant des parts de graphisme différent, en activant l'option « Spécifique aux parts ».

Il faudra donc alors dessiner chaque part, et si l'option « Spécifique aux représentations » est activée, pour chaque part il faudra dessiner les deux représentations.

Enfin il peut être intéressant d'activer l'option « Commun aux représentations » pour les composants dessinés à la norme IEEE moderne, puisque l'essentiel du graphisme est identique dans les représentations normales et converties.

### 11.5.2 - Les éléments graphiques géométriques

Leur tracé est possible grâce aux outils:

- Tracé de lignes et polygones simples ou remplis selon que l'option **Filled** est inactive ou active.
- Tracé de rectangles définis par une diagonale.
- Tracé de cercles définis par le centre et un point de la circonférence.
- Tracé d'arcs définis par le point de départ et d'arrivée de l'arc et son centre. L'arc va de 0 à 180 degrés.

### 11.5.3 - Les éléments graphiques type textes



Permet la création de textes graphiques (textes libre)

Les textes sont toujours lisibles, même si le composant est mis en miroir

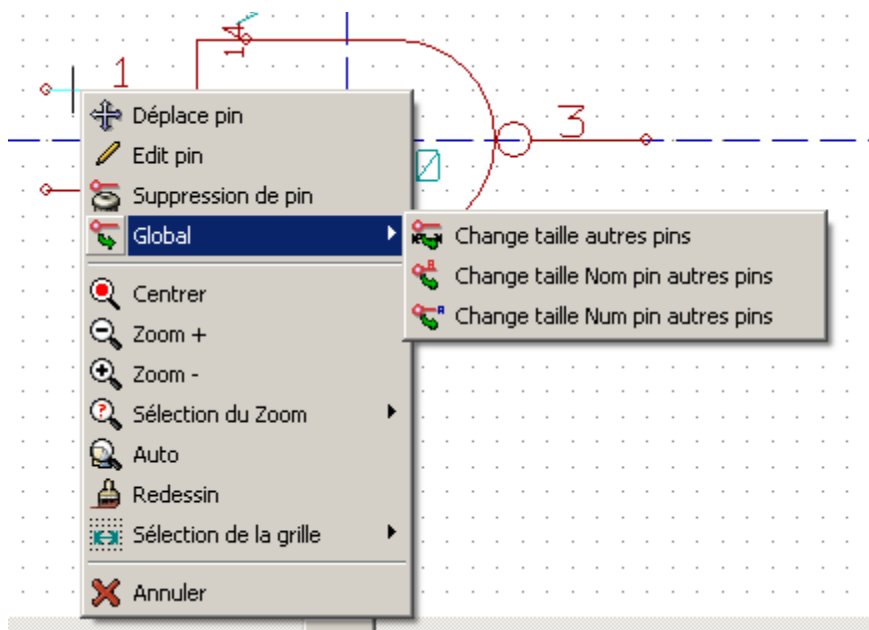
## 11.6 - Création et édition des pins



La création se fait par l'outil

L'édition se fait en double-cliquant sur la pin à éditer

Si on clique par le bouton droit, apparaît le menu rapide d'édition:



La création des pins doit être soigneusement faite, et est critique, car toute erreur aura des conséquences pour la réalisation de circuits imprimés, ou rendra inopérante la fonction E.R.C. Toute pin déjà placée peut être rééditée effacée ou déplacée.

### 11.6.1 - Notions générales sur les pins

Une pin est définie par sa forme ( longueur, forme du graphisme ), son nom et son « numéro » qui n'est pas toujours un nombre (cas des supports PGA dont la pin est définie par une lettre et un nombre, comme A12 ou AB45) Dans Eeschema, le « numéro de pin » est un mot de 4 lettres ou chiffres

Pour les contrôles électriques, le type « électrique » ( entrée, sortie, sortie 3 états... ) doit également être défini, pour la fonction E.R.C .

Si ce type est mal défini, le contrôle E.R.C n'aura aucune valeur.

#### Remarques:

- Éviter les espaces dans les noms et numéros de pins. Cela ne pose pas de problèmes pour Eeschema, mais beaucoup de formats de netlistes n'acceptent pas les espaces.
- Si le nom de la pin commence par le symbole « ~ », ce nom sera surligné.
- Si le nom se réduit à ce seul symbole, la pin est considérée comme sans nom.
- Le nom ne doit pas commencer par le symbole « # », car il a une signification particulière et est utilisé pour créer des symboles d'alimentation.
- Le numéro de pin est constitué de 1 à 4 lettres ou chiffres.  
Un numéro valide est donc 1,2 ..9999 mais aussi A1, B3 ... (notation type PGA) ou encore Anod, Gnd, Vin ...

### 11.6.2 - Boîtiers à parts multiples et double représentation

Rappelons que, en particulier pour des portes logiques, un symbole peut avoir deux représentations (représentation dite « De Morgan », et un boîtier peut comporter plusieurs parts (plusieurs portes NOR par ex.)

Pour certains boîtiers , on peut vouloir avoir plusieurs éléments de graphisme et pins différents.

Par exemple un relai peut être représenté avec des parts différentes :

- Bobine
- contact 1
- contact 2

La gestion des boîtiers à parts multiples, et des composants à double représentation est souple.

En effet, une pin peut être :

- Commun aux diverses parts ou spécifique à chaque part.
- Commun aux deux représentation ou spécifique à chaque représentation.

Par défaut, les pins, spécifiques à chaque représentation à chaque part, car leur numéro diffère pour chaque part, et leur forme diffère pour chaque représentation.

Lorsque une pin est commune il suffit de la dessiner une seule fois (c'est le cas de pins d'alimentation par ex.)

C'est aussi le cas du graphisme qui est presque toujours identique entre parts (mais diffère de la représentation normale à la représentation convertie).



### 11.6.3 - Pins : option de base

Les composants à parts et/ou représentation multiples posent un problème particulier pour la création et l'édition des pins. Dans la mesure où la plupart des pins sont particulières à chaque part (car leur numéro de pin est spécifique de chaque part) et à chaque représentation (car leur forme est spécifique à chaque représentation), la création et l'édition des pins risqueraient donc d'être longue et fastidieuses.

En fait, Eeschema permet la manipulation simultanée des pins :

Par défaut, pour les boîtiers à part multiples et /ou à double représentation, lorsque l'on crée, édite (sauf forme, et numéro) efface ou déplace une pin, ces modifications sont faites pour toutes les pins correspondantes des parts et des représentations (c'est à dire **pour toutes les pins placées à la même coordonnée**).



- Pour la forme les modifications sont faites pour la représentation courante, pour toutes les parts.
- Les numéros sont modifiés pour la part courante, pour les 2 représentations.
- Les noms sont modifiés de façon indépendante.

Cette dépendance a été établie pour permettre des modifications rapides dans une situation correspondant à la très grande majorité des cas.

Cette dépendance dans les modifications peut être désactivée dans le menu Options.

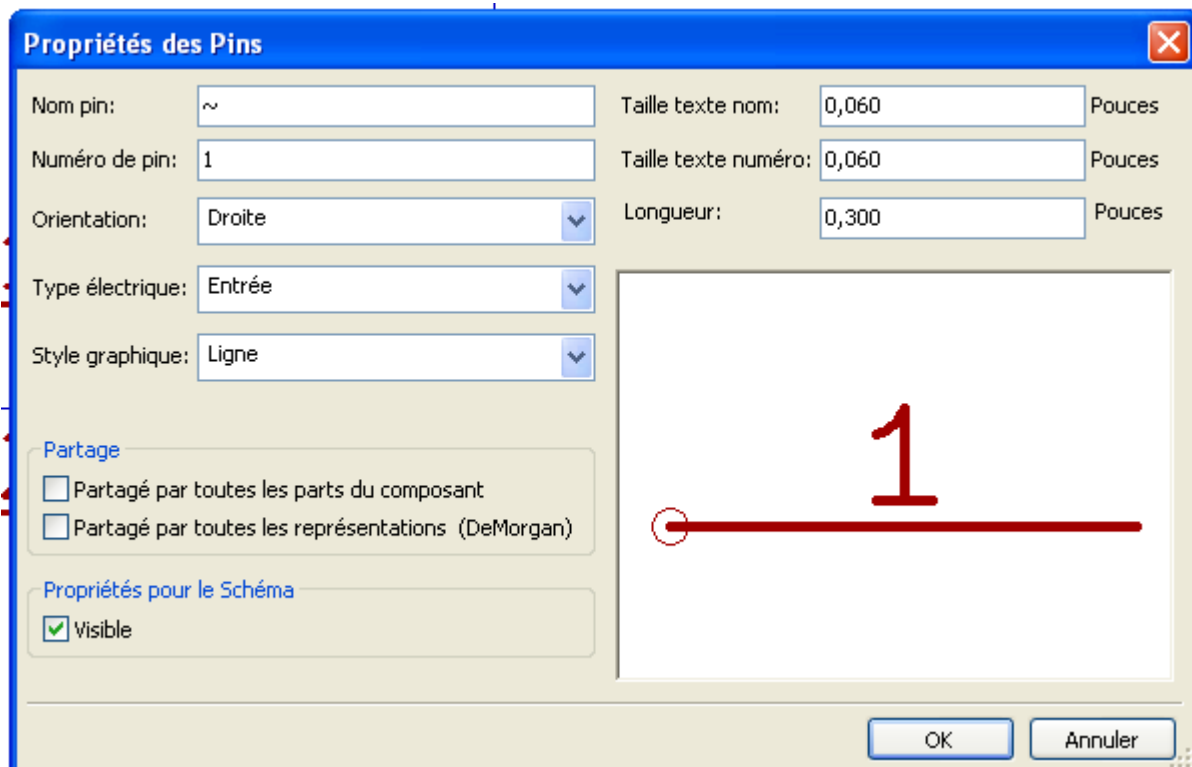
Elle permet alors de créer des composants à parts et représentations de caractéristiques complètement indépendantes.

Cette option de dépendance est gérée par l'outil

- Si  **est activé** : les éditions ne se feront que sur la part et la représentation courantes (c'est à dire sur ce que l'on voit à l'écran). *Cette option est d'utilisation rare.*
- Si  **est désactivé** : les éditions se feront sur **toutes** les parts et les représentations. *C'est l'option normale de travail.*

### 11.6.4 - Pins : Définition des caractéristiques

La fenêtre d'édition permet de définir toutes les caractéristiques d'une pin.



**Propriétés des Pins**

Nom pin: ~ Taille texte nom: 0,060 Pouces

Numéro de pin: 1 Taille texte numéro: 0,060 Pouces

Orientation: Droite Longueur: 0,300 Pouces

Type électrique: Entrée

Style graphique: Ligne

**Partage**

☐ Partagé par toutes les parts du composant

☐ Partagé par toutes les représentations (DeMorgan)

**Propriétés pour le Schéma**

☒ Visible

OK Annuler

Ce menu est appelé automatiquement à la création d'une pin, ou en double cliquant sur une pin existante.

Il permet de définir ou de modifier :

- Le nom et la taille du nom d'une pin.
- Le numéro et la taille du numéro d'une pin.
- la longueur de la pin.
- le type électrique et la forme de la pin.
- son appartenance

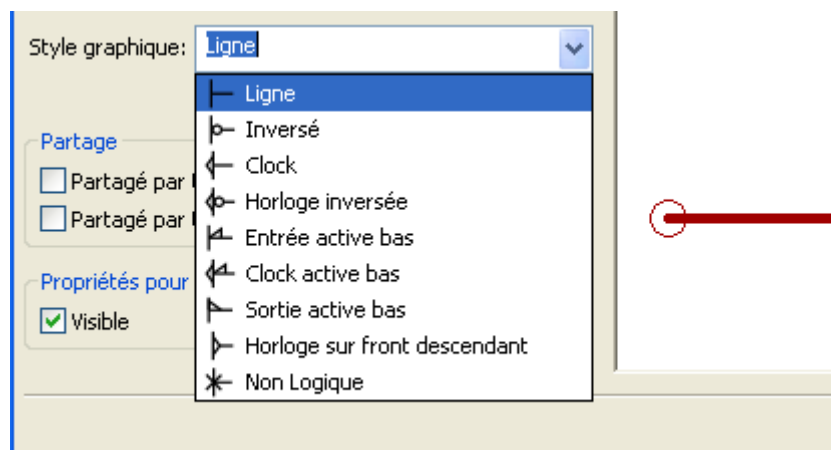
- si elle est invisible (option utile pour les pins d'alimentation).

### Rappel:

- Si le nom de la pin commence par le symbole « ~ », ce nom sera **surligné**.
- Pour un surlignage partiel, les lettres entre 2 symbole « ~ » seront surlignées
- Si le nom se réduit à ce seul symbole, la pin est considérée comme sans nom.
- Le numéro de pin est constitué de 1 à 4 lettres ou chiffres.  
Un numéro valide est donc 1,2 ..9999 mais aussi A1, B3 ... (notation type PGA) ou encore *Anod*, *Gnd*, *V.in* ...

### 11.6.5 - Formes des pins

Voici les diverses formes possibles pour une pin :



Le choix de la forme a une influence purement graphique, et n'a aucune influence sur les fonctions de contrôle ou de netliste.

### 11.6.6 - Type électrique des pins

Le choix du type est important, car il déterminera l'efficacité de la fonction E.R.C.

Le choix est trivial pour des pins de circuits intégrés type **entrée** ou **sortie**.

- Le type **BiDi** désigne des pins commutables en entrée ou en sortie (pins de bornier d'entrée ou de sortie de microprocesseurs par exemple).
- Le type **3 States** est la sortie 3 états usuelle.
- Le type **Passif** est utilisé pour les pins de composants passifs, pour les transistors, connecteurs....
- Le type **Unspec** (non spécifié) peut être utilisé dans les cas où le type électrique paraît sans importance.
- Le type **Power In** est à réserver aux pins d'alimentation des composants.  
En particulier **si la pin est du type Power In, et a l'attribut Invisible, elle ne sera pas affichée en schématique, et elle sera automatiquement connectée aux autres pins du même type et de même nom (Pin Power In Invisible)**.
- Le type **Power Out** est à réserver aux pins de sortie de régulateurs.
- Les type Open Emitter et Open Collector sont à réserver aux pins de sortie de circuits intégrés du type collecteur ou émetteur ouvert.

### 11.6.7 - Modifications globales des pins

On peut modifier la longueur de toutes les pins, ou la taille des textes Nom et Numéro de pin globalement, par la commande **Global** du menu Pop Up.

Un menu permettant d'ajuster l'un de ces trois paramètres sera afficher.

Cliquer sur le paramètre à modifier et ajuster la nouvelle valeur, qui sera alors appliquées à toutes les pins du composant, pour la représentation courante.

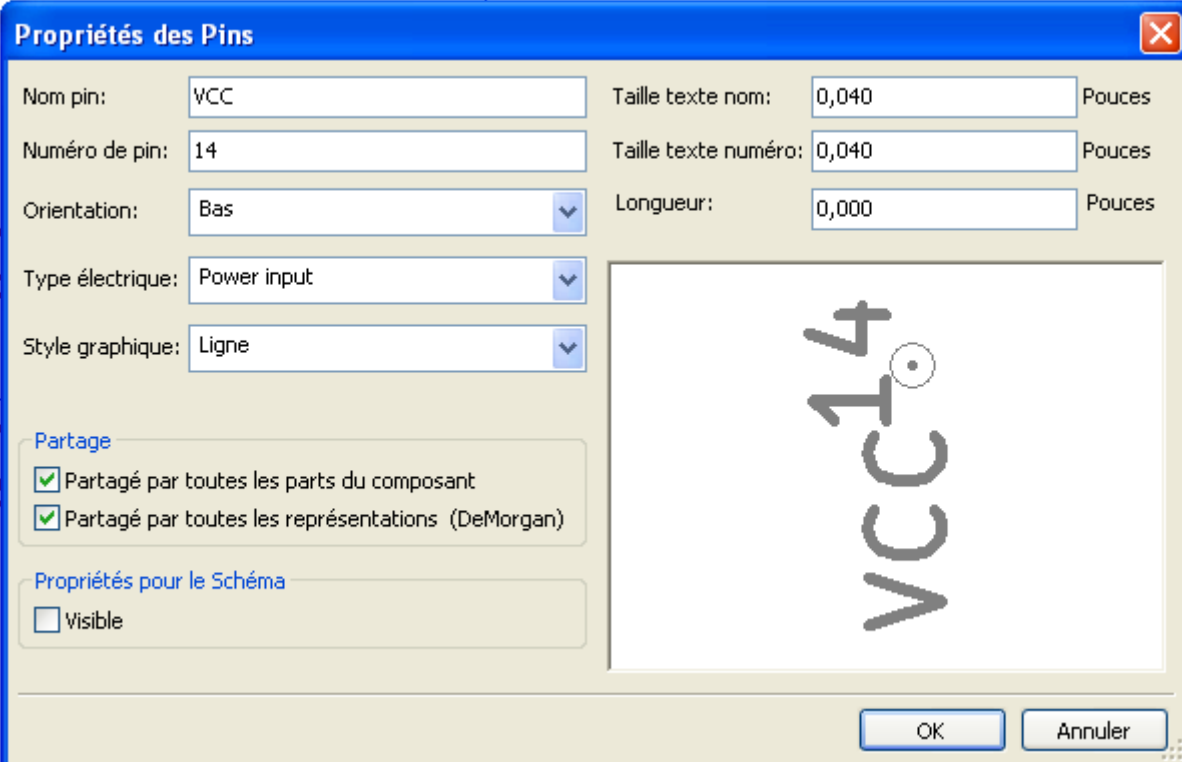
### 11.6.8 - Pins pour parts multiples et doubles représentations

Les différentes parts ou représentation (telles qu'on les rencontre dans un composant du type 7400, 7402 ...) peuvent demander un complément d'édition.

Ce travail complémentaire sera limité si l'on prend les précautions suivantes :

- L'option générale **Editer pins unité par unité**  doit rester **non activée**.
- Les pins d'alimentation seront créées avec l'attribut **Partagé par toutes les parts du boîtier** et **Partagé par toutes les formes alternatives (DeMorgan)** actifs (Elles seront peut être aussi invisibles).

Voici un réglage type:



**Propriétés des Pins**

Nom pin:  Taille texte nom:  Pouces

Numéro de pin:  Taille texte numéro:  Pouces

Orientation:  Longueur:  Pouces

Type électrique:

Style graphique:

**Partage**

☒ Partagé par toutes les parts du composant

☒ Partagé par toutes les représentations (DeMorgan)

**Propriétés pour le Schéma**

☐ Visible

OK Annuler

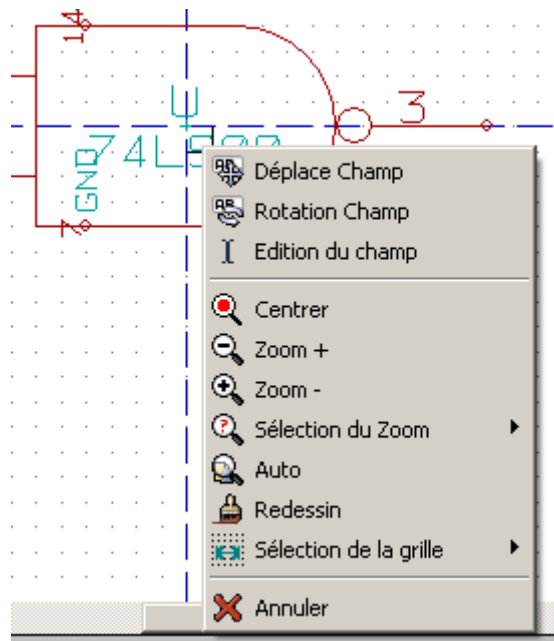
Lorsque l'on aura créé les autres pins, on les aura donc créées pour chaque part et chaque représentation.

Par exemple la pin de sortie de la part A du 7400 aura été créée par Eeschema en 8 exemplaires : 2 par part (il y a 4 parts A,B,C,D et pour chaque part la représentation normale, et la représentation convertie dite de Morgan.). Cependant on aura, pour commencer, probablement créé correctement la part A dans sa représentation normale. Il faudra donc pour chaque part :

- Sélectionner la représentation convertie, et éditer la forme et la longueur de chacune des pins.
- Pour les autres parts que la part A, éditer les numéros de pins.

### 11.7 - Édition des champs

Pour les champs déjà existants, on peut utiliser les commandes rapides d'édition en cliquant par le bouton de droite de la souris sur le champ à modifier:



Pour des éditions plus complètes ou des champs vides, il faut appeler la fenêtre d'édition des champs **T**:

**Propriétés des Champs**

Champs

Référence	U
Valeur	74LS00
Module	
Documentation	
Champ1	
Champ2	
Champ3	
Champ4	
Champ5	
Champ6	
Champ7	
Champ8	

Ajouter Champ

Supprimer Champ

Vers le haut ^

Options

Justification Horiz.

☐ Alignement à gauche

☒ Alignement au centre

☐ Alignement à droite

Justifié Vert.

☐ Alignement en bas

☒ Alignement au centre

☐ Alignement au sommet

Visibilité

☒ Visible

☐ Rotation

Style:

☒ Normal

☐ Italique

☐ Gras

☐ Gras Italique

Nom Champ

Référence

Texte Champ

U

Taille (""):

0,060

Pos X (""):

0,000

Pos Y (""):

-0,050

OK Annuler

On a ici sélectionné le champ (field) *Référence*.

Les champs sont des textes associés au composant, à ne pas confondre avec les textes faisant partie du symbole graphique représentant ce composant.

### On dispose des champs:

- Valeur
- Référence
- Nom du module associé ( pour le PCB )
- Documentation: un lien vers une documentation particulière (surtout prévu pour la schématique).
- Champs modèles (définis dans l'éditeur de schématique)
- On peut ajouter d'autres champs si désiré.

Les champs valeur et référence ont leur contenu défini à la création du composant, et peuvent être modifiés ici.

Il peut être éventuellement utile d'éditer le champ Nom du module associé pour générer directement des netlistes (pour le logiciel de Circuits Imprimés) dans lesquelles le module sera prédéclaré systématiquement.

Le champ Documentation peut être utilisé comme lien vers un fichier auxiliaire.

Les Champs 1 à 8 (et plus) n'ont pas vraiment de raison d'être utilisés en librairie, car ils sont plutôt conçus pour être utilisés dans la partie schématique.

On peut redéfinir leur nom.

Pour la partie librairie, l'édition des champs Valeur et Référence permet essentiellement de définir leur taille et position.

### Remarques importantes:

- Lorsque l'on **modifie** le **texte** du champ **valeur**, cela revient à **créer un nouveau composant**, à partir de la définition d'un ancien composant, utilisé alors comme modèle, car lors des sauvegarde en librairie, le composant prend le nom donné par le champ valeur.
- Pour éditer un champ non visible (c'est à dire vide, car même si le champ a l'attribut Invisible, il est affiché dans LibEdit) on devra nécessairement utiliser la fenêtre d'édition générale ci-dessus.

## 11.8 - Création des symboles d'alimentation

Les symboles d'alimentation sont créés comme des composants usuels.

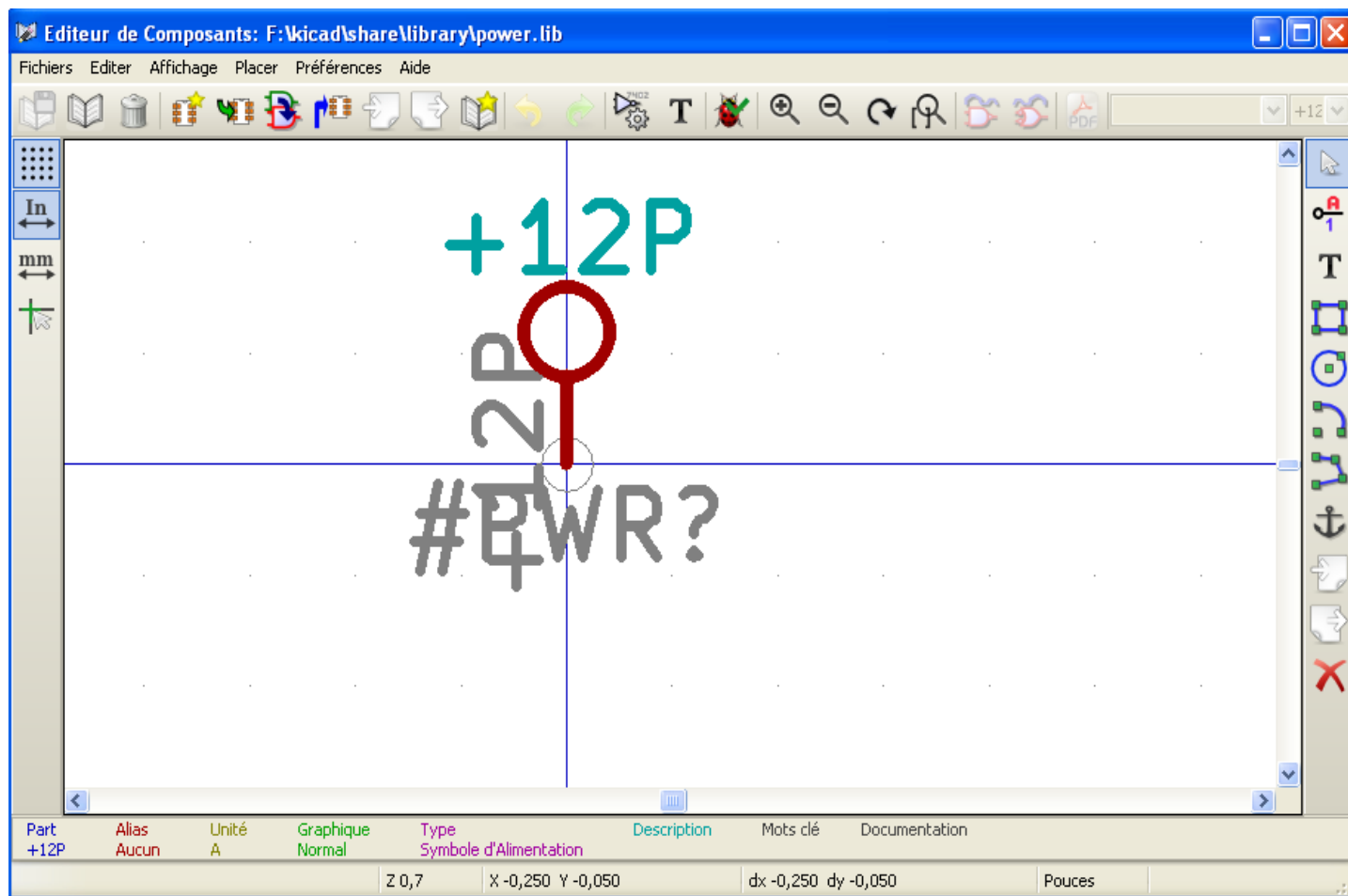
On a intérêt à les regrouper dans une librairie unique telle que **Power.lib**.

Ils sont constitués du graphisme (la forme désirée) et d'une **pin** du **type** « **Power Invisible** ».

Ils seront donc dans la schématique manipulés comme tous les autres composants.

Cependant, quelques précautions s'imposent.

Voici un symbole (alimentation **-5V**) :



Le symbole est réalisé de la façon suivante :

- Une pin « **Power Invisible** » de nom **-5V** (important puis c'est ce nom qui va établir la connexion aux autres pins -5V), de numéro de pin 1 (numéro sans importance) et de longueur nulle.
- La forme est du type « Line », et évidemment le type est « Power » et l'attribut est « Invisible ».
- Un graphisme : ici un petit cercle et un segment allant de la pin au cercle.
- L'ancre du symbole est sur la pin
- La valeur est **-5V** comme le nom de la pin, pour affichage à l'écran de la valeur de ce symbole (la pin étant obligatoirement invisible, son nom n'apparaît donc pas à l'écran).
- La référence est **#-5V** (donc affichée comme **#-5V?**) comme le nom de la pin. Le texte référence est sans grande importance sauf le premier caractère qui est « # ». Par convention, tout composant dont la référence commence par ce symbole n'apparaîtra ni dans la liste des composants, ni dans les netlistes. De plus, en Option de symbole, la référence est déclarée invisible.
- L'option "**Symbole d'alimentation**" est activée.

La création d'un nouveau symbole d'alimentation est facile et rapide si l'on utilise un autre symbole comme modèle. On doit alors :

- Charger le modèle.
- Éditer le **nom** de la **pin** (qui prend alors le nom de nouvelle valeur d'alimentation).
- Éditer le champ **Valeur** (même nom que la pin, si l'on veut afficher la valeur de cette alimentation...).
- Sauver le nouveau composant.

## 12 - LibEdit : Compléments

### Table des matières

12 - LibEdit : Compléments.....	1
12.1 - Généralités.....	1
12.2 - Positionnement de l'ancre.....	2
12.3 - Alias.....	3
12.4 - Champs.....	3
12.5 - Documentation des composants.....	4
12.5.1 - Mots clés (Keywords).....	5
12.5.2 - Documentation du composant (Doc).....	5
12.5.3 - Fichier associé de documentation (DocFileName).....	5
12.5.4 - Filtrage des Modules pour CvPcb.....	5
12.6 - « Librairie » de symboles.....	7
12.6.1 - Exportation/Création de symboles.....	7
12.6.2 - Importation d'un symbole.....	7

#### 12.1 - Généralités

On rappelle qu'un composant comporte plusieurs éléments:

- Son graphisme ( formes géométriques, textes ).
- Les Pins.
- Les champs, ou textes associés, utilisés par les post processeurs : netliste, liste des composants...

**Les 2 champs devant obligatoirement être initialisés sont la référence et la valeur.**

On peut aussi donner le nom du module associé, pour le circuit imprimé.

Le nom du schéma associé au composant n'est pas utilisé, il est prévu pour de futurs développements.

Les autres champs sont les champs libres, Les autres peuvent généralement rester vides, et pourront être complétés en schématique.

Cependant, gérer une documentation associée à ce composant en facilite beaucoup la recherche, l'utilisation et la maintenance des librairies.

Elle est constituée de

- Une ligne de commentaire.
- Une ligne de mots clés ' tels que TTL CMOS NAND2..., séparés par des espaces
- Un nom de fichier attaché (par exemple un schéma d'application, une notice, un fichier pdf...). Par défaut, le fichier attaché est recherché dans le répertoire

**kicad/share/library/doc** puis

**kicad/library/doc**

et sous linux également dans

**/usr/local/kicad/share/library/doc**

**/usr/share/kicad/library/doc**

**/usr/local/share/kicad/library/doc**

Les mots clés permettront de rechercher de façon sélective un composant en fonction d'un critère de sélection

La ligne de commentaire et de mots clés est affichée dans différents menus, et en particulier lorsque l'on sélectionne un composant à partir de la liste affichée des composants d'une librairie.

Le composant possède également une ancre ou point d'ancrage. C'est par rapport à ce point qu'une rotation ou un miroir sont faits, et lors d'un placement, c'est ce point qui sert de référence de position.

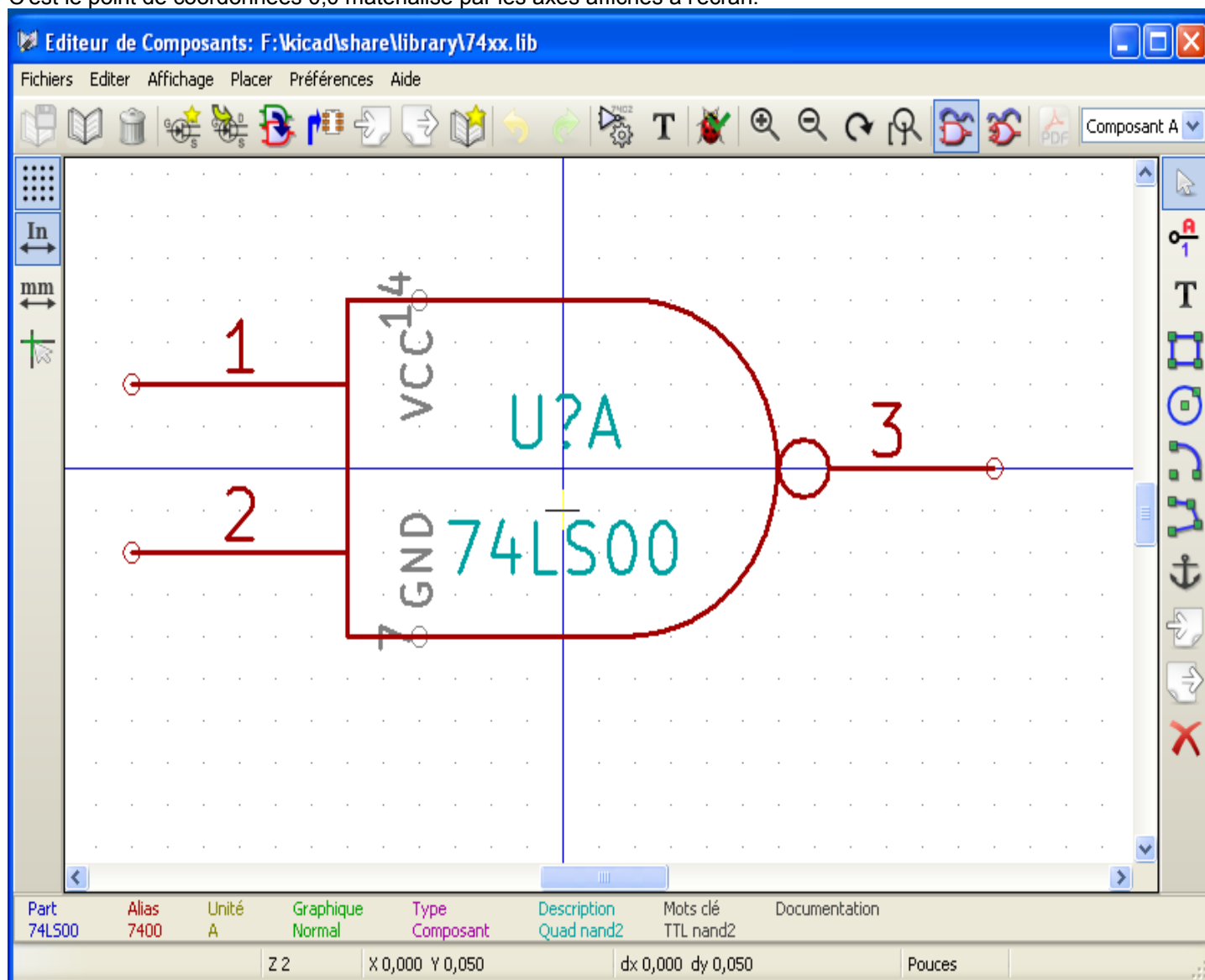
Il est donc bon de positionner intelligemment cette ancre.

Un composant peut aussi avoir des alias, c'est à dire une série de noms équivalents. Cela permet de réduire considérablement de nombre de composants à créer réellement (par exemple, un 74LS00 peut avoir comme alias 74000, 74HC00, 74HCT00 ...).

Enfin les composants sont répartis dans des librairies (classement par thèmes, ou par fabricant..), de façon à faciliter leur gestion.

#### 12.2 - Positionnement de l'ancre

C'est le point de coordonnées 0,0 matérialisé par les axes affichés à l'écran:



On peut repositionner l'ancre de la façon suivante:



Sélectionner l'outil

Puis cliquer sur la nouvelle position désirée de l'ancre. Le dessin sera automatiquement recadré

### 12.3 - Alias


Un alias est un autre nom correspondant au même composant en librairie.

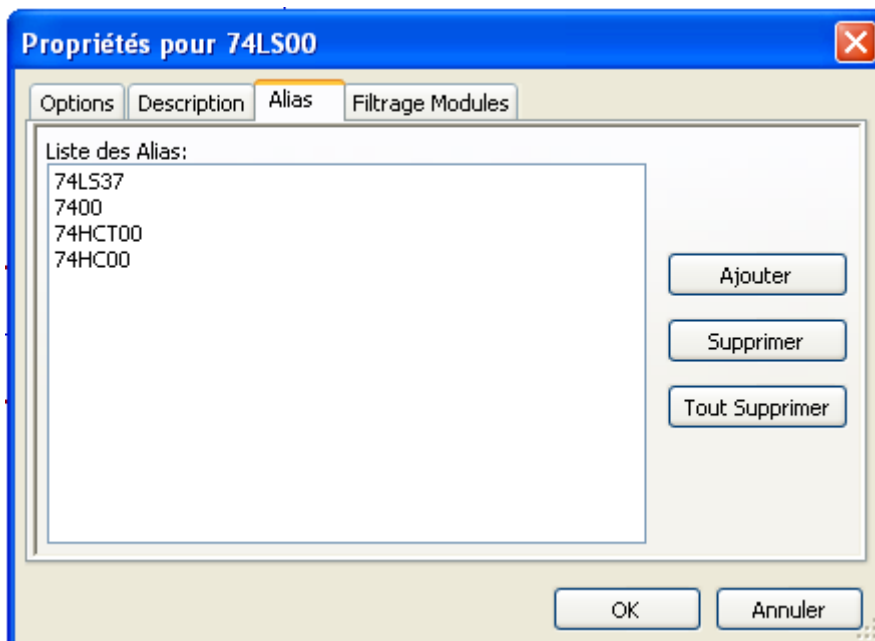
Des composants similaires en brochage et représentation peuvent alors être représentés par un seul composant, auquel on ajoutera plusieurs alias ( ex: 7400 avec alias 74LS00, 74HC00, 74LS37 ... ).

L'utilisation des alias permet de construire beaucoup plus rapidement des librairies complètes.

Par ailleurs ces librairies, beaucoup plus compactes que si chaque variante était construite séparément sont chargées plus rapidement.



Pour modifier la liste des alias, il faut appeler la fenêtre d'édition générale du composant (outil ) et sélectionner **Alias**:




On peut ainsi ajouter ou supprimer tout alias désiré.

L'alias courant ne peut évidemment pas être supprimé puisque il est en cours d'édition.

Pour supprimer tous les alias, il faut d'abord veiller à sélectionner le composant racine ( premier composant de la liste des alias dans la fenêtre de sélection du toolbar général).

### 12.4 - Champs:

Il y a 4 champs (textes attachés au composant) réservés, et 8 champs libres (champs utilisateurs)

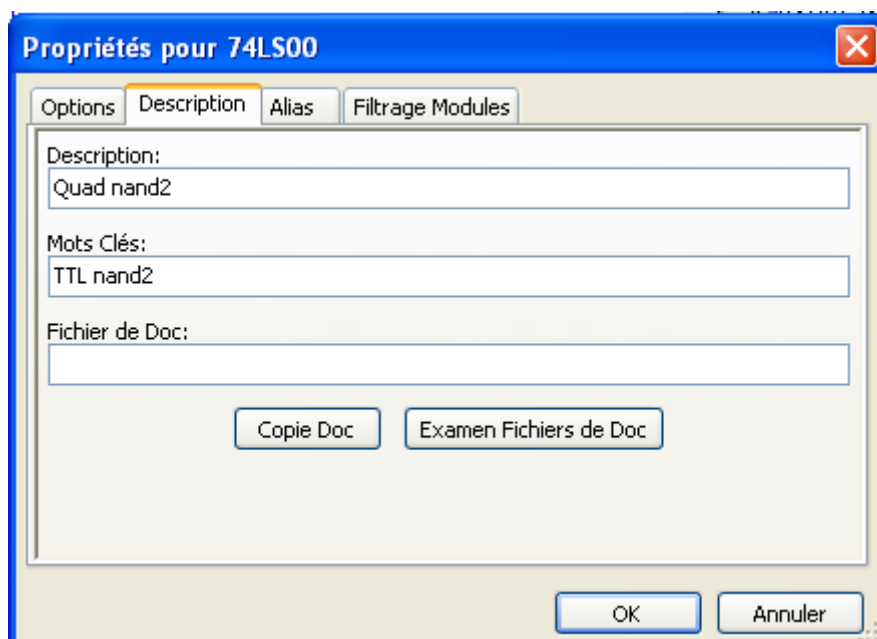
L'éditeur de champs est accessible par l'outil  :

#### Champs réservés:

- Référencer
- Valeur: C'est le nom du composant en librairie, et la valeur par défaut en schématique.
- Module Nom du module utilisé pour le circuit imprimé. Peu utile lorsque on utilise CvPcb pour gérer les modules, mais obligatoire si CvPcb n'est pas utilisé.
- Documentation: réservé (non encore utilisé).

## 12.5 - Documentation des composants)

Pour éditer la documentation, il faut appeler la fenêtre d'édition générale du composant (outil ) et sélectionner **Doc**:



**Attention:**

Bien veiller à la bonne sélection de l'alias, ou du composant racine, car cette documentation est la seule caractéristique qui diffère entre alias.

Le bouton Copy Doc permet de copier les informations de documentation du composant racine vers l'alias en cours d'édition.

### 12.5.1 - Mots clés (Keywords)

Les mots clés permettront de rechercher de façon sélective un composant en fonction d'un critère de sélection (fonction, famille technologique..).

Lors de la recherche, Eeschema ne fait pas de différence entre minuscules et majuscules.

Les mots clés les plus courants utilisés dans les bibliothèques sont

CMOS TTL pour les familles logiques

AND2 NOR3 XOR2 INV... pour les portes ( AND2 = porte AND 2 entrées, NOR3 = porte NOR à 3 entrées).

JKFF DFF ... pour les bascules JK, D (FF = Flip Flop).

CAD CDA MUX...

OpenCol pour les portes à collecteur ouvert.

Ainsi si dans la schématique, on demande le composant par mots clés **NAND2 OpenCol**, Eeschema affichera la liste des seuls composants possédant ces 2 mots clés.

### 12.5.2 - Documentation du composant (Doc)

La ligne de commentaire (et de mots clés) est affichée dans différents menus, et en particulier lorsque l'on sélectionne un composant à partir de la liste affichée des composants d'une bibliothèque et dans le menu **ViewLib**.

Si cette doc existe, elle est aussi accessible dans la schématique dans le menu Popup, en cliquant avec le bouton droit sur un composant.

### 12.5.3 - Fichier associé de documentation (DocFileName)

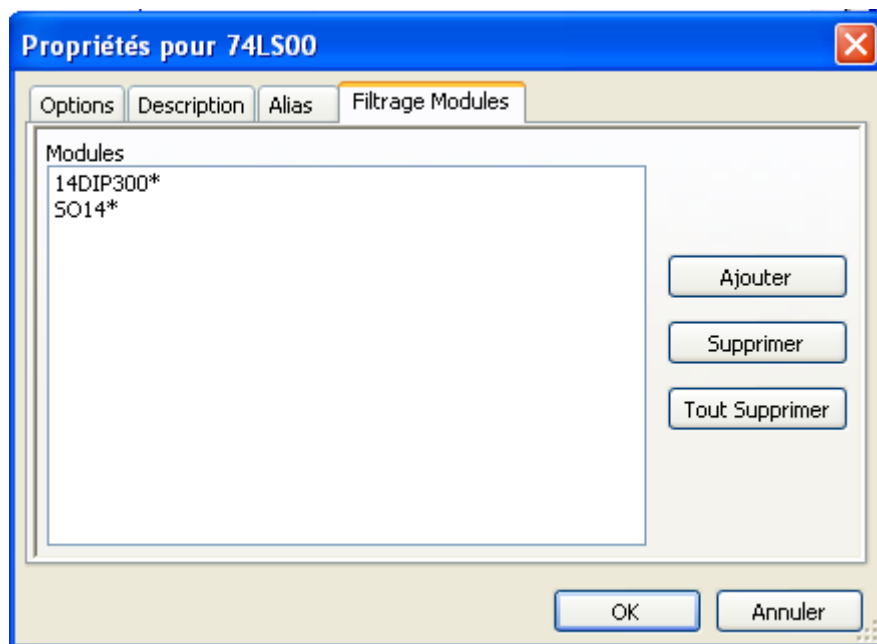
Désigne un fichier attaché (fichier pdf de documentation) qui pourra être visualisé.

### 12.5.4 - Filtrage des Modules pour CvPcb

On peut donner la liste des modules autorisés pour le composant.

Cette liste est un filtre et sera utilisée par CvPcb pour n'afficher que ces modules autorisés.

Une liste vide ne filtre rien.



On peut utiliser des caractères "jocker" dans les nom:

**S014\*** permet d'afficher dans CvPcb tous les modules dont les noms commencent par **S014**

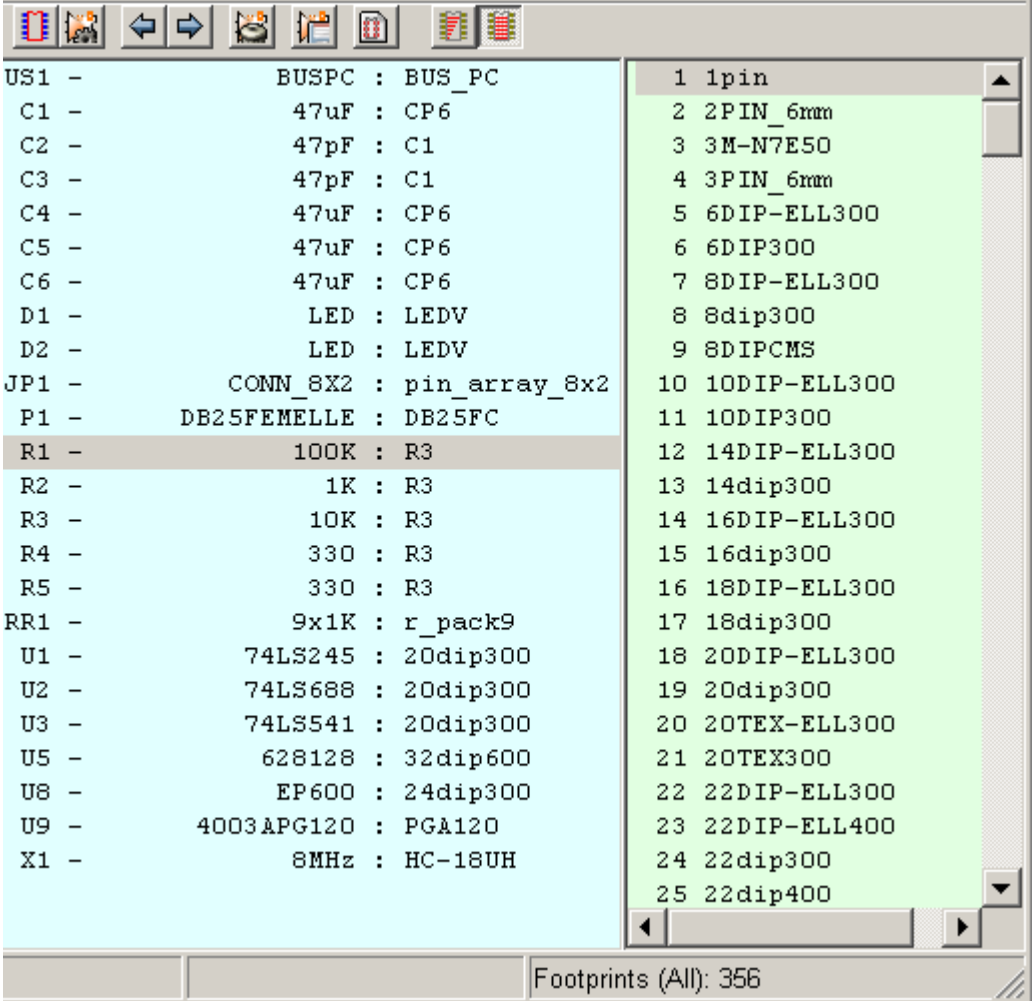
Pour une résistance, R? permet d'afficher tous les modules dont le nom commence par R et est composé de 2 lettres.

Exemple d'affichage avec et sans filtrage pour le même composant:

BUS1 -	BUSPC : BUS_PC	1 R1
C1 -	47uF : CP6	2 R3
C2 -	47pF : C1	3 R4
C3 -	47pF : C1	4 R5
C4 -	47uF : CP6	5 R6
C5 -	47uF : CP6	6 R7
C6 -	47uF : CP6	7 SMO603
D1 -	LED : LEDV	8 SMO805
D2 -	LED : LEDV	
JP1 -	CONN_8X2 : pin_array_8x2	
P1 -	DB25FEMELLE : DB25FC	
R1 -	100K : R3	
R2 -	1K : R3	
R3 -	10K : R3	
R4 -	330 : R3	
R5 -	330 : R3	
RR1 -	9x1K : r_pack9	
U1 -	74LS245 : 20dip300	
U2 -	74LS688 : 20dip300	
U3 -	74LS541 : 20dip300	
U5 -	628128 : 32dip600	
U8 -	EP600 : 24dip300	
U9 -	4003APG120 : PGA120	
X1 -	8MHz : HC-18UH	

Footprints (filtered): 8

Avec filtage



US1 -	BUSPC : BUS_PC	1 1pin
C1 -	47uF : CP6	2 2PIN_6mm
C2 -	47pF : C1	3 3M-N7E50
C3 -	47pF : C1	4 3PIN_6mm
C4 -	47uF : CP6	5 6DIP-ELL300
C5 -	47uF : CP6	6 6DIP300
C6 -	47uF : CP6	7 8DIP-ELL300
D1 -	LED : LEDV	8 8dip300
D2 -	LED : LEDV	9 8DIPCMS
JP1 -	CONN_8X2 : pin_array_8x2	10 10DIP-ELL300
P1 -	DB25FEMELLE : DB25FC	11 10DIP300
R1 -	100K : R3	12 14DIP-ELL300
R2 -	1K : R3	13 14dip300
R3 -	10K : R3	14 16DIP-ELL300
R4 -	330 : R3	15 16dip300
R5 -	330 : R3	16 18DIP-ELL300
RR1 -	9x1K : r_pack9	17 18dip300
U1 -	74LS245 : 20dip300	18 20DIP-ELL300
U2 -	74LS688 : 20dip300	19 20dip300
U3 -	74LS541 : 20dip300	20 20TEX-ELL300
U5 -	628128 : 32dip600	21 20TEX300
U8 -	EP600 : 24dip300	22 22DIP-ELL300
U9 -	4003APG120 : PGA120	23 22DIP-ELL400
X1 -	8MHz : HC-18UH	24 22dip300
		25 22dip400

Footprints (All): 356

Sans filtrage

## 12.6 - « Librairie » de symboles

On peut se constituer facilement des fichiers de symboles graphiques fréquemment utilisés dans la création de composants (triangles, forme de portes ET, OU, OU Exclusif...), les sauver et les réutiliser.

Ces fichiers sont par défaut dans le répertoire des bibliothèques et ont pour extension **.sym**.

Les symboles ne sont pas regroupés en bibliothèques comme les composants car ils sont en général très peu nombreux.

### 12.6.1 - Exportation/Création de symboles



Un composant peut être exporté comme symbole par l'outil

En général on ne crée qu'un graphisme, aussi il sera une bonne idée d'effacer les pins en place, s'il y en a.

### 12.6.2 - Importation d'un symbole

Cela permet d'ajouter des graphismes à un composant en cours d'édition.



Un symbole est importé par l'outil

Le graphisme importé vient s'ajouter tel qu'il a été créé au graphisme existant.

13 - Viewlib

Table des matières

13 - Viewlib.....1

13.1 - Rôle.....1

13.2 - Écran général.....1

13.3 - Toolbar de Viewlib.....2

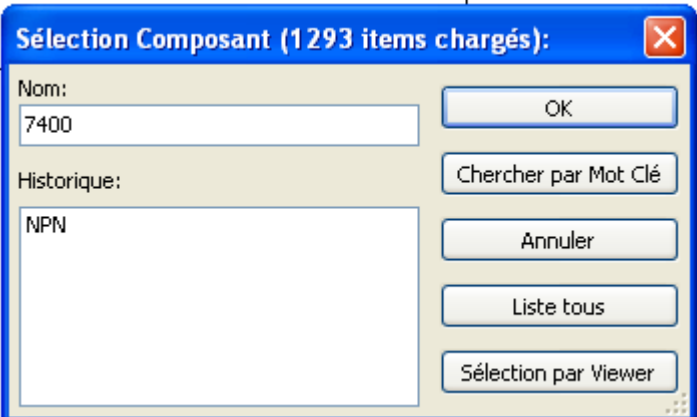
13.1 - Rôle

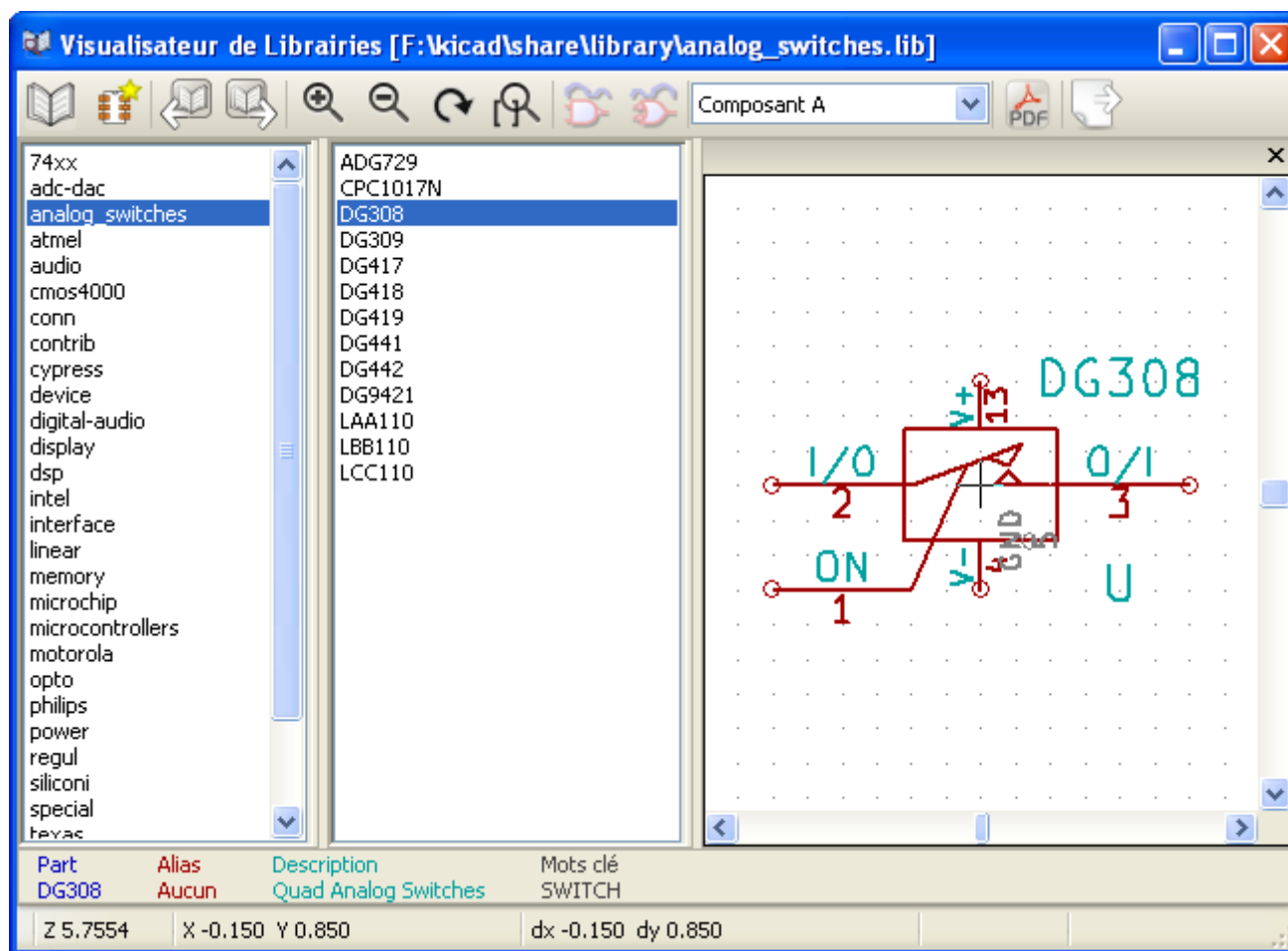
Viewlib permet d'examiner rapidement le contenu des librairies.

Viewlib est appelé par l'outil  ,

ou par le menu de placement de composant:


13.2 - Écran général





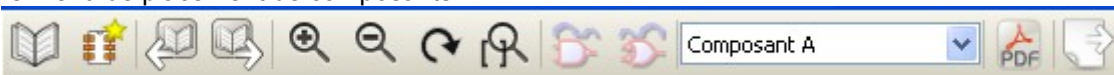
Pour examiner une bibliothèque il faut la sélectionner dans la liste affichée sur la gauche. Son contenu apparaît alors dans la deuxième liste qui permet de sélectionner le composant à examiner.

### 13.3 - Toolbar de Viewlib

Toolbar (appel direct par )







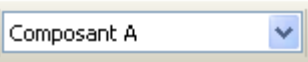




ou appel par le menu de placement de composants



Les commandes sont.

## Eeschema - Viewlib

	Sélection de la librairie courante (qui peut être aussi sélectionnée dans la liste affichée).
	Sélection du composant (qui peut être aussi sélectionnée dans la liste affichée).
	Afficher le composant précédent.
	Afficher le composant suivant.
	Gestion du zoom.
	Sélection de la représentation (normale ou convertie) s'il y en a deux.
	Sélection de la part (s'il y en a plusieurs).
	Affichage du document associé (s'il existe).
	Existe seulement si appel par le menu de placement de composants : Fermer Viewlib et placer le composant sélectionné dans le schéma.



## 14 - Customize Netlists and BOM

### Headers:

14 - Customize Netlists and BOM.....	1
14.1 - The Intermediate Netlist.....	1
14.1.1 - Schematic sample.....	2
14.1.2 - The Intermediate Netlist file sample.....	2
14.2 - Conversion to a new format.....	4
14.3 - XSLT approach.....	4
14.3.1 - Sample to create a Pads-Pcb netlist file.....	5
14.3.2 - Sample to create a Cadstar netlist file.....	6
14.3.2.1 - The Style-Sheet file.....	6
14.3.2.2 - The output file.....	8
14.3.3 - Sample to create a OrcadPCB2 netlist file.....	8
14.3.3.1 - The Style-Sheet file.....	8
14.3.3.2 - The output file.....	12
14.3.4 - Using Eeschema plugins interface.....	12
14.3.4.1 - Init the Dialog window.....	12
14.3.4.2 - Parameters.....	13
14.3.4.3 - The command line.....	14
14.3.4.4 - Command line format.....	14
14.4 - The Intermediate Netlist structure.....	14
14.4.1 - General structure.....	15
14.4.2 - The header section.....	15
14.4.3 - The components section.....	15
14.4.3.1 - Note about time stamps for components.....	16
14.4.4 - The libparts section.....	16
14.4.5 - The libraries section.....	17
14.4.6 - The nets section.....	17
14.5 - More info about xsltproc.....	18
14.5.1 - Introduction.....	18
14.5.2 - Synopsis.....	18
14.5.3 - Command Line Options.....	18
14.5.4 - Return values.....	19
14.5.5 - More Information.....	19

### 14.1 - The Intermediate Netlist:

BOM and netlists can be converted from an Intermediate Netlist file created by Eeschema.

This file uses an XML syntax.

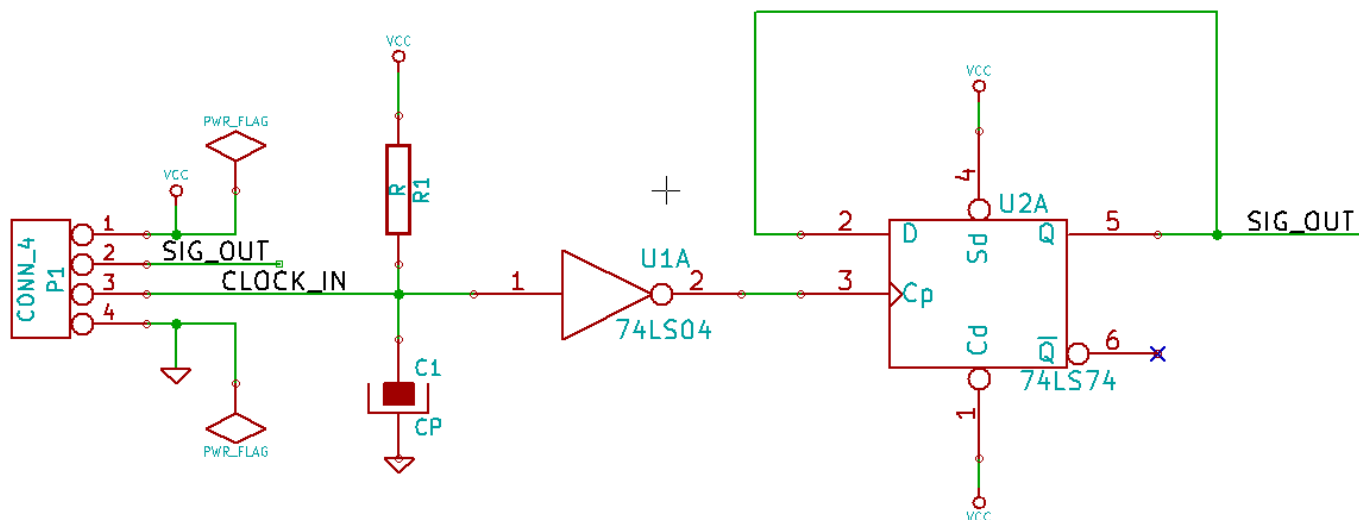
This file is called Intermediate Netlist.

But because it includes a lot of data about components, it can also be used to create BOM or others reports, not just netlists.

Depending on the output (BOM, new netlist), only some of sections in this Intermediate Netlist will be used.

Here is a sample:

## 14.1.1 - Schematic sample:



## 14.1.2 - The Intermediate Netlist file sample:

The corresponding intermediate netlist (using XML syntax) is:

```
<?xml version="1.0" encoding="utf-8"?>
<export version="D">
  <design>
    <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
    <date>29/08/2010 20:35:21</date>
    <tool>eeschema (2010-08-28 BZR 2458)-unstable</tool>
  </design>
  <components>
    <comp ref="P1">
      <value>CONN 4</value>
      <libsource lib="conn" part="CONN 4"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20BA</tstamp>
    </comp>
    <comp ref="U1">
      <value>74LS04</value>
      <libsource lib="74xx" part="74LS04"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20A6</tstamp>
    </comp>
    <comp ref="C1">
      <value>CP</value>
      <libsource lib="device" part="CP"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2094</tstamp>
    </comp>
    <comp ref="R1">
      <value>R</value>
      <libsource lib="device" part="R"/>
```

```

    <sheetpath names="/" tstamp="/" />
    <tstamp>4C6E208A</tstamp>
  </comp>
</components>
<libparts>
  <libpart lib="device" part="C">
    <description>Condensateur non polarise</description>
    <footprints>
      <fp>SM*</fp>
      <fp>C?</fp>
      <fp>C1-1</fp>
    </footprints>
    <fields>
      <field name="Reference">C</field>
      <field name="Value">C</field>
    </fields>
    <pins>
      <pin num="1" name="~" type="passive"/>
      <pin num="2" name="~" type="passive"/>
    </pins>
  </libpart>
  <libpart lib="device" part="R">
    <description>Resistance</description>
    <footprints>
      <fp>R?</fp>
      <fp>SM0603</fp>
      <fp>SM0805</fp>
      <fp>R?-*</fp>
      <fp>SM1206</fp>
    </footprints>
    <fields>
      <field name="Reference">R</field>
      <field name="Value">R</field>
    </fields>
    <pins>
      <pin num="1" name="~" type="passive"/>
      <pin num="2" name="~" type="passive"/>
    </pins>
  </libpart>
  <libpart lib="conn" part="CONN_4">
    <description>Symbole general de connecteur</description>
    <fields>
      <field name="Reference">P</field>
      <field name="Value">CONN_4</field>
    </fields>
    <pins>
      <pin num="1" name="P1" type="passive"/>
      <pin num="2" name="P2" type="passive"/>
      <pin num="3" name="P3" type="passive"/>
      <pin num="4" name="P4" type="passive"/>
    </pins>
  </libpart>
  <libpart lib="74xx" part="74LS04">
    <description>Hex Inverseur</description>
    <fields>
      <field name="Reference">U</field>
      <field name="Value">74LS04</field>
    </fields>
    <pins>
      <pin num="1" name="~" type="input"/>
      <pin num="2" name="~" type="output"/>
      <pin num="3" name="~" type="input"/>
      <pin num="4" name="~" type="output"/>
      <pin num="5" name="~" type="input"/>
      <pin num="6" name="~" type="output"/>
      <pin num="7" name="GND" type="power_in"/>
      <pin num="8" name="~" type="output"/>
      <pin num="9" name="~" type="input"/>
      <pin num="10" name="~" type="output"/>
      <pin num="11" name="~" type="input"/>
      <pin num="12" name="~" type="output"/>
      <pin num="13" name="~" type="input"/>
    </pins>
  </libpart>

```

```

    <pin num="14" name="VCC" type="power_in"/>
  </pins>
</libpart>
<libpart lib="74xx" part="74LS74">
  <description>Dual D FlipFlop, Set & Reset</description>
  <docs>74xx/74hc_hct74.pdf</docs>
  <fields>
    <field name="Reference">U</field>
    <field name="Value">74LS74</field>
  </fields>
  <pins>
    <pin num="1" name="Cd" type="input"/>
    <pin num="2" name="D" type="input"/>
    <pin num="3" name="Cp" type="input"/>
    <pin num="4" name="Sd" type="input"/>
    <pin num="5" name="Q" type="output"/>
    <pin num="6" name="~Q" type="output"/>
    <pin num="7" name="GND" type="power_in"/>
    <pin num="8" name="~Q" type="output"/>
    <pin num="9" name="Q" type="output"/>
    <pin num="10" name="Sd" type="input"/>
    <pin num="11" name="Cp" type="input"/>
    <pin num="12" name="D" type="input"/>
    <pin num="13" name="Cd" type="input"/>
    <pin num="14" name="VCC" type="power_in"/>
  </pins>
</libpart>
</libparts>
<libraries>
  <library logical="device">
    <uri>F:\kicad\share\library\device.lib</uri>
  </library>
  <library logical="conn">
    <uri>F:\kicad\share\library\conn.lib</uri>
  </library>
  <library logical="74xx">
    <uri>F:\kicad\share\library\74xx.lib</uri>
  </library>
</libraries>
<nets>
  <net code="1" name="GND">
    <node ref="U1" pin="7"/>
    <node ref="C1" pin="2"/>
    <node ref="U2" pin="7"/>
    <node ref="P1" pin="4"/>
  </net>
  <net code="2" name="VCC">
    <node ref="R1" pin="1"/>
    <node ref="U1" pin="14"/>
    <node ref="U2" pin="4"/>
    <node ref="U2" pin="1"/>
    <node ref="U2" pin="14"/>
    <node ref="P1" pin="1"/>
  </net>
  <net code="3" name="">
    <node ref="U2" pin="6"/>
  </net>
  <net code="4" name="">
    <node ref="U1" pin="2"/>
    <node ref="U2" pin="3"/>
  </net>
  <net code="5" name="/SIG_OUT">
    <node ref="P1" pin="2"/>
    <node ref="U2" pin="5"/>
    <node ref="U2" pin="2"/>
  </net>
  <net code="6" name="/CLOCK_IN">
    <node ref="R1" pin="2"/>
    <node ref="C1" pin="1"/>
    <node ref="U1" pin="1"/>
    <node ref="P1" pin="3"/>
  </net>

```

```
</nets>
</export>
```

### 14.2 - Conversion to a new format:

By applying a filter to this Intermediate Netlist file, one can generate netlists in a lot of others format, and B.O.M. Because this is a text to text transform, this filter can be easily written using PYTHON or XSLT.

XSLT itself is a an XML language to define transformation for XML.

When using XSLT, *Xsltproc* program can be used to read an XML file, applies a style-sheet (the "filter" and save the resulting output.

So the used has to create a style-sheet file using XSLT conventions.

The full conversion process is handle by Eeschema, and is transparent.

### 14.3 - XSLT approach:

The XSL Transformations (XSLT) documentation is here:

<http://www.w3.org/TR/xslt>

#### 14.3.1 - Sample to create a Pads-Pcb netlist file

This format has two sections:

- A footprint list
- A Nets list (grouping pads references by nets)

Here is a style-sheet sample to convert the Intermediate Netlist file to a Pads-Pcb netlist format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to EESchema Generic Netlist Format to PADS netlist format
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

How to use:
https://lists.launchpad.net/kicad-developers/msg05157.html
-->

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<xsl:template match="/export">
  <xsl:text>*PADS-PCB*&nl;*PART*&nl;</xsl:text>
  <xsl:apply-templates select="components/comp"/>
  <xsl:text>&nl;*NET*&nl;</xsl:text>
  <xsl:apply-templates select="nets/net"/>
  <xsl:text>*END*&nl;</xsl:text>
</xsl:template>

<!-- for each component -->
<xsl:template match="comp">
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "footprint != ' ' ">
      <xsl:apply-templates select="footprint"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>unknown</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

<!-- for each net -->
<xsl:template match="net">
  <!-- nets are output only if there is more than one pin in net -->
  <xsl:if test="count(node)>1">
```

## Eeschema - Customize Netlists and BOM

```
<xsl:text>*SIGNAL* </xsl:text>
<xsl:choose>
  <xsl:when test = "@name != '' ">
    <xsl:value-of select="@name"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>N-</xsl:text>
    <xsl:value-of select="@code"/>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>&nl;</xsl:text>
<xsl:apply-templates select="node"/>
</xsl:if>
</xsl:template>

<!-- for each node -->
<xsl:template match="node">
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text>.</xsl:text>
  <xsl:value-of select="@pin"/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>
```

**</xsl:stylesheet>**

Here is the final output, after applying *xsltproc*:

```
*PADS-PCB*
*PART*
P1 unknown
U2 unknown
U1 unknown
C1 unknown
R1 unknown

*NET*
*SIGNAL* GND
U1.7
C1.2
U2.7
P1.4
*SIGNAL* VCC
R1.1
U1.14
U2.4
U2.1
U2.14
P1.1
*SIGNAL* N-4
U1.2
U2.3
*SIGNAL* /SIG_OUT
P1.2
U2.5
U2.2
*SIGNAL* /CLOCK_IN
R1.2
C1.1
U1.1
P1.3
*END*
```

And the command line to convert the netlist files is:

```
f:/kicad/bin/xsltproc.exe -o test.net f:/kicad/bin/plugins/netlist_form_pads-pcb.xsl test.tmp
```

### 14.3.2 - Sample to create a Cadstar netlist file

This format has two sections:

- A footprint list
- A Nets list (grouping pads references by nets)

#### 14.3.2.1 - The Style-Sheet file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to EESCHEMA Generic Netlist Format to CADSTAR netlist format
Copyright (C) 2010, Jean-Pierre Charras.
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<!-- Netlist header -->
<xsl:template match="/export">
  <xsl:text>.HEA&nl;</xsl:text>
  <xsl:apply-templates select="design/date"/> <!-- Generate line .TIM <time> -->
  <xsl:apply-templates select="design/tool"/> <!-- Generate line .APP <eeschema
version> -->
  <xsl:apply-templates select="components/comp"/> <!-- Generate list of components
-->
  <xsl:text>&nl;&nl;</xsl:text>
  <xsl:apply-templates select="nets/net"/> <!-- Generate list of nets and
connections -->
  <xsl:text>&nl;.END&nl;</xsl:text>
</xsl:template>

  <!-- Generate line .TIM 20/08/2010 10:45:33 -->
<xsl:template match="tool">
  <xsl:text>.APP "</xsl:text>
  <xsl:apply-templates/>
  <xsl:text>"&nl;</xsl:text>
</xsl:template>

  <!-- Generate line .APP "eeschema (2010-08-17 BZR 2450)-unstable" -->
<xsl:template match="date">
  <xsl:text>.TIM </xsl:text>
  <xsl:apply-templates/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

  <!-- for each component -->
<xsl:template match="comp">
  <xsl:text>.ADD_COM </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "value != '' ">
      <xsl:text>"</xsl:text> <xsl:apply-templates select="value"/>
<xsl:text>"</xsl:text>
    </xsl:when>
```

```

        <xsl:otherwise>
            <xsl:text>"</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>&nl;</xsl:text>
</xsl:template>

<!-- for each net -->
<xsl:template match="net">
    <!-- nets are output only if there is more than one pin in net -->
    <xsl:if test="count(node)>1">
        <xsl:variable name="netname">
            <xsl:text>"</xsl:text>
            <xsl:choose>
                <xsl:when test = "@name != ' ' ">
                    <xsl:value-of select="@name"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:text>N-</xsl:text>
                    <xsl:value-of select="@code"/>
                </xsl:otherwise>
            </xsl:choose>
            <xsl:text>"&nl;</xsl:text>
        </xsl:variable>
        <xsl:apply-templates select="node" mode="first"/>
        <xsl:value-of select="$netname"/>
        <xsl:apply-templates select="node" mode="others"/>
    </xsl:if>
</xsl:template>

<!-- for each node -->
<xsl:template match="node" mode="first">
    <xsl:if test="position()=1">
        <xsl:text>.ADD_TER </xsl:text>
        <xsl:value-of select="@ref"/>
        <xsl:text>.</xsl:text>
        <xsl:value-of select="@pin"/>
        <xsl:text> </xsl:text>
    </xsl:if>
</xsl:template>

<xsl:template match="node" mode="others">
    <xsl:choose>
        <xsl:when test='position()=1'>
            </xsl:when>
        <xsl:when test='position()=2'>
            <xsl:text>.TER </xsl:text>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text> </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="position()>1">
        <xsl:value-of select="@ref"/>
        <xsl:text>.</xsl:text>
        <xsl:value-of select="@pin"/>
        <xsl:text>&nl;</xsl:text>
    </xsl:if>
</xsl:template>

</xsl:stylesheet>

```



### 14.3.2.2 - The output file:

```
.HEA
.TIM 21/08/2010 08:12:08
.APP "eeschema (2010-08-09 BZR 2439)-unstable"
.ADD_COM P1 "CONN 4"
.ADD_COM U2 "74LS74"
.ADD_COM U1 "74LS04"
.ADD_COM C1 "CP"
.ADD_COM R1 "R"

.ADD_TER U1.7 "GND"
.TER      C1.2
          U2.7
          P1.4
.ADD_TER R1.1 "VCC"
.TER      U1.14
          U2.4
          U2.1
          U2.14
          P1.1
.ADD_TER U1.2 "N-4"
.TER      U2.3
.ADD_TER P1.2 "/SIG_OUT"
.TER      U2.5
          U2.2
.ADD_TER R1.2 "/CLOCK_IN"
.TER      C1.1
          U1.1
          P1.3

.END
```

### 14.3.3 - Sample to create a OrcadPCB2 netlist file

This format has only one section: the footprint list:

Each footprint includes its list of pads with reference to a net.

#### 14.3.3.1 - The Style-Sheet file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to EESCHEMA Generic Netlist Format to CADSTAR netlist format
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

How to use:
https://lists.launchpad.net/kicad-developers/msg05157.html
-->

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<!--
Netlist header
Creates the entire netlist
(can be seen as equivalent to main function in C
-->
<xsl:template match="/export">
```

```

<xsl:text>({ EESchema Netlist Version 1.1 </xsl:text>
<!-- Generate line .TIM <time> -->
<xsl:apply-templates select="design/date"/>
<!-- Generate line eeschema version ... -->
<xsl:apply-templates select="design/tool"/>
<xsl:text>}&nl;</xsl:text>

<!-- Generate the list of components -->
<xsl:apply-templates select="components/comp"/> <!-- Generate list of components -->

<!-- end of file -->
<xsl:text>)&nl;*&nl;</xsl:text>
</xsl:template>

<!--
Generate id in header like "eeschema (2010-08-17 BZR 2450)-unstable"
-->
<xsl:template match="tool">
  <xsl:apply-templates/>
</xsl:template>

<!--
Generate date in header like "20/08/2010 10:45:33"
-->
<xsl:template match="date">
  <xsl:apply-templates/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

<!--
This template read each component
(path = /export/components/comp)
creates lines:
( 3EBF7DBD $noname U1 74LS125
... pin list ...
)
and calls "create_pin_list" template to build the pin list
-->
<xsl:template match="comp">
  <xsl:text> (</xsl:text>
  <xsl:choose>
    <xsl:when test = "tstamp != " ">
      <xsl:apply-templates select="tstamp"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>00000000</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "footprint != " ">
      <xsl:apply-templates select="footprint"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>$noname</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>

```

```

<xsl:when test = "value != " ">
  <xsl:apply-templates select="value"/>
</xsl:when>
<xsl:otherwise>
  <xsl:text>"~"</xsl:text>
</xsl:otherwise>
</xsl:choose>
<xsl:text>&nl;</xsl:text>
<xsl:call-template name="Search_pin_list" >
  <xsl:with-param name="cmplib_id" select="libsourc/@part"/>
  <xsl:with-param name="cmp_ref" select="@ref"/>
</xsl:call-template>
<xsl:text>)&nl;</xsl:text>
</xsl:template>

<!--
  This template search for a given lib component description in list
  lib component descriptions are in /export/libparts,
  and each description start at ./libpart
  We search here for the list of pins of the given component
  This template has 2 parameters:
    "cmplib_id" (reference in libparts)
    "cmp_ref" (schematic reference of the given component)
-->
<xsl:template name="Search_pin_list" >
  <xsl:param name="cmplib_id" select="0" />
  <xsl:param name="cmp_ref" select="0" />
  <xsl:for-each select="/export/libparts/libpart">
    <xsl:if test = "@part = $cmplib_id">
      <xsl:apply-templates name="build_pin_list" select="pins/pin">
        <xsl:with-param name="cmp_ref" select="$cmp_ref"/>
      </xsl:apply-templates>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

<!--
  This template writes the pin list of a component
  from the pin list of the library description
  The pin list from library description is something like
    <pins>
      <pin num="1" type="passive"/>
      <pin num="2" type="passive"/>
    </pins>
  Output pin list is ( <pin num> <net name> )
  something like
    ( 1 VCC )
    ( 2 GND )
-->
<xsl:template name="build_pin_list" match="pin">
  <xsl:param name="cmp_ref" select="0" />

  <!-- write pin numner and separator -->
  <xsl:text> (</xsl:text>
  <xsl:value-of select="@num"/>
  <xsl:text> </xsl:text>

  <!-- search net name in nets section and write it: -->
  <xsl:variable name="pinNum" select="@num" />
  <xsl:for-each select="/export/nets/net">

```

```

<!-- net name is output only if there is more than one pin in net
      else use "?" as net name, so count items in this net
-->
<xsl:variable name="pinCnt" select="count(node)" />
<xsl:apply-templates name="Search_pin_netname" select="node">
  <xsl:with-param name="cmp_ref" select="$cmp_ref"/>
  <xsl:with-param name="pin_cnt_in_net" select="$pinCnt"/>
  <xsl:with-param name="pin_num"> <xsl:value-of select="$pinNum"/>
</xsl:with-param>
</xsl:apply-templates>
</xsl:for-each>

<!-- close line -->
<xsl:text>)&nl;</xsl:text>
</xsl:template>

<!--
This template writes the pin netname of a given pin of a given component
from the nets list
The nets list description is something like
<nets>
  <net code="1" name="GND">
    <node ref="J1" pin="20"/>
    <node ref="C2" pin="2"/>
  </net>
  <net code="2" name="">
    <node ref="U2" pin="11"/>
  </net>
</nets>
This template has 2 parameters:
  "cmp_ref" (schematic reference of the given component)
  "pin_num" (pin number)
-->

<xsl:template name="Search_pin_netname" match="node">
  <xsl:param name="cmp_ref" select="0" />
  <xsl:param name="pin_num" select="0" />
  <xsl:param name="pin_cnt_in_net" select="0" />

  <xsl:if test="@ref = $cmp_ref">
    <xsl:if test="@pin = $pin_num">
      <!-- net name is output only if there is more than one pin in net
            else use "?" as net name
      -->
      <xsl:if test="$pin_cnt_in_net > 1">
        <xsl:choose>
          <!-- if a net has a name, use it,
                else build a name from its net code
          -->
          <xsl:when test="../@name != """>
            <xsl:value-of select="../@name"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:text>$N-0</xsl:text><xsl:value-of select="../@code"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:if>
      <xsl:if test="$pin_cnt_in_net < 2">
        <xsl:text>?</xsl:text>
      </xsl:if>
    </xsl:if>
  </xsl:if>

```

```
</xsl:if>

</xsl:template>

</xsl:stylesheet>
```

### 14.3.3.2 - The output file:

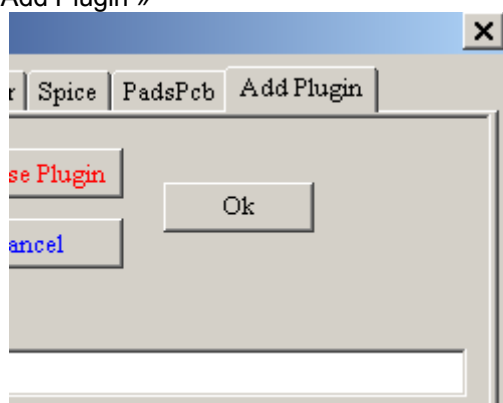
```
( { EESchema Netlist Version 1.1 29/08/2010 21:07:51
eeschema (2010-08-28 BZR 2458)-unstable}
( 4C6E2141 $noname P1 CONN_4
( 1 VCC )
( 2 /SIG_OUT )
( 3 /CLOCK_IN )
( 4 GND )
)
( 4C6E20BA $noname U2 74LS74
( 1 VCC )
( 2 /SIG_OUT )
( 3 N-04 )
( 4 VCC )
( 5 /SIG_OUT )
( 6 ? )
( 7 GND )
( 14 VCC )
)
( 4C6E20A6 $noname U1 74LS04
( 1 /CLOCK_IN )
( 2 N-04 )
( 7 GND )
( 14 VCC )
)
( 4C6E2094 $noname C1 CP
( 1 /CLOCK_IN )
( 2 GND )
)
( 4C6E208A $noname R1 R
( 1 VCC )
( 2 /CLOCK_IN )
)
)
*
```

### 14.3.4 - Using Eeschema plugins interface:

Intermediate Netlist converters can be automatically launched by Eeschema

#### 14.3.4.1 - Init the Dialog window:

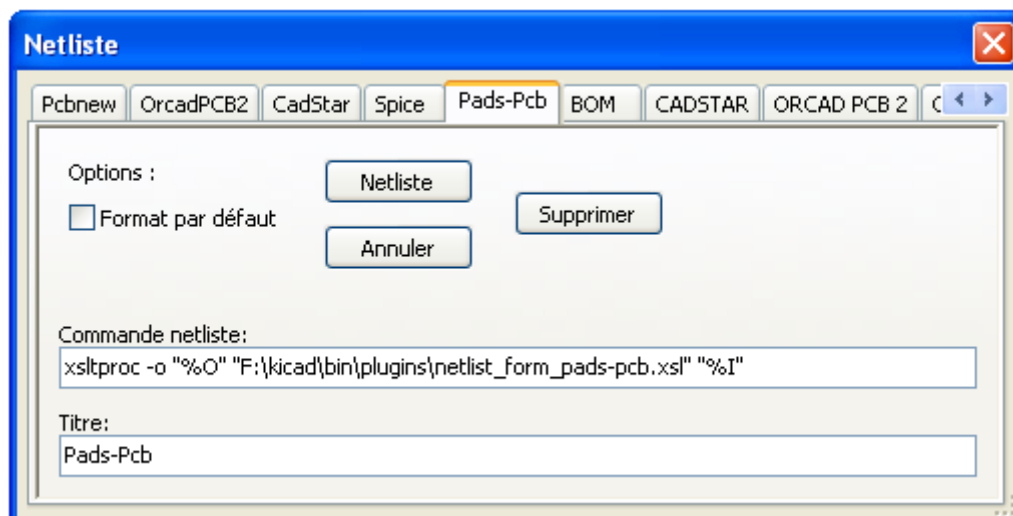
One can add a new netlist plug-in by « Add Plugin »



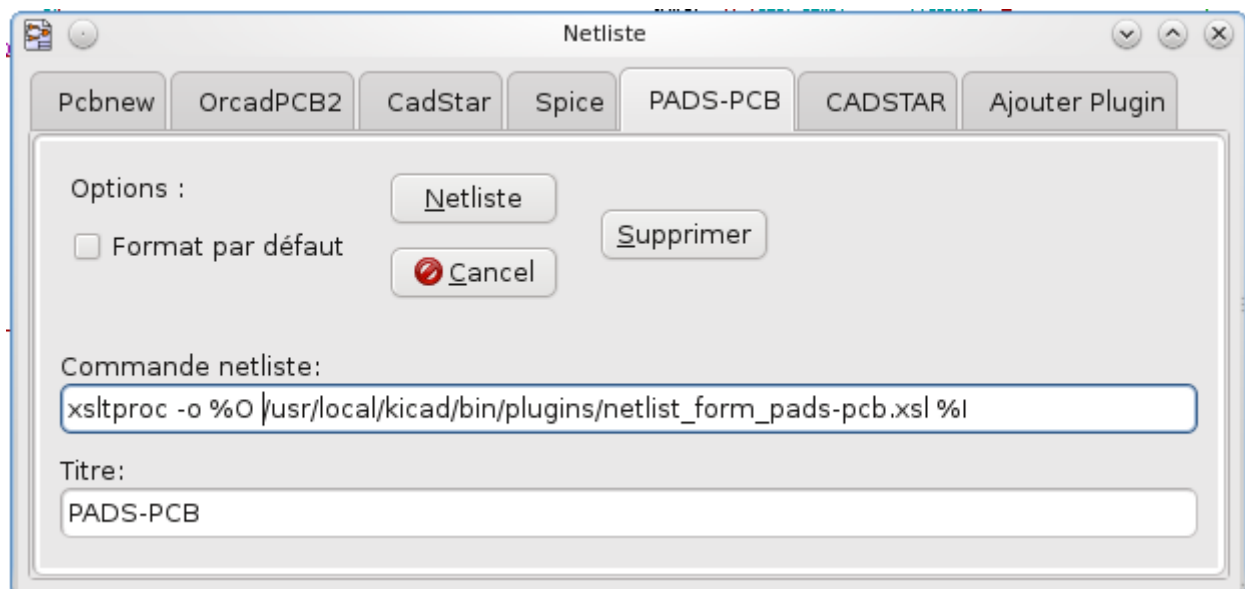
## Eeschema - Customize Netlists and BOM

Here is the plug-in « PadsPcb » setup:

**Windows:**

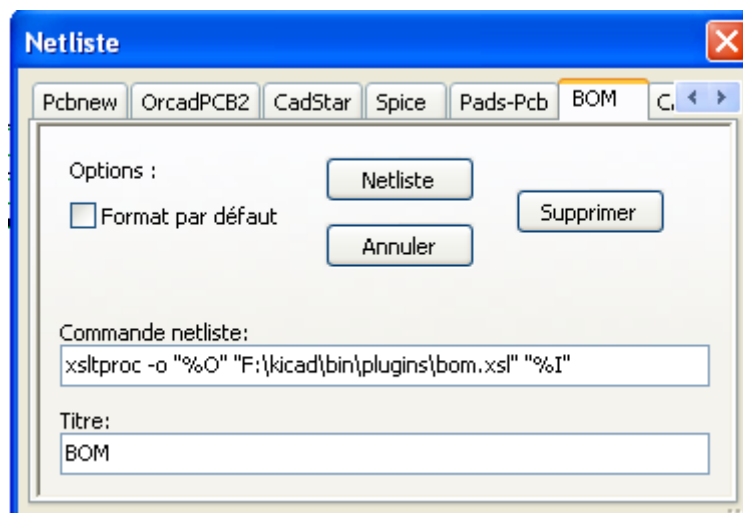


**Linux:**



Because this intermediate file contains most of info about components, a BOM can be extracted from it this file. Here is the plug-in setup to create a customized Bill Of Material (BOM):

**Windows:**



### 14.3.4.2 - Parameters:

The setup needs:

- A title (for instance: the name of the netlist format)
- The command line to launch the converter.

On activate the Netlist button:

1. Eeschema creates an intermediate netlist file \*.xml, for instance *test.xml*
2. Eeschema runs the plug-in, what read test.xml and creates test.net

### 14.3.4.3 - The command line:

Assuming we are using *xsltproc.exe* to apply the sheet style to the intermediate file:

*xsltproc.exe* needs a command like:

*xsltproc.exe* -o <output filename> < style-sheet filename> <input XML file to convert>

So, in Kicad the command line is (using Windows):

```
"f:/kicad/bin/xsltproc.exe" -o "%O" "f:/kicad/bin/plugins/netlist_form_pads-pcb.xml" "%I"
```

or (using Linux)

```
xsltproc -o "%O" "/usr/local/kicad/bin/plugins/netlist_form_pads-pcb.xml" "%I"
```

*netlist\_form\_pads-pcb.xml* is the style-sheet.

**Do not forget to quote file names if they have spaces in name.**

### 14.3.4.4 - Command line format:

<path of *xsltproc* >*xsltproc* < *xsltproc* parameters>

For parameters, supported formatting sequences are:

- %B => base filename and path of selected output file, minus path and extension.
- %I => complete filename and path of the temporary input file.
- %O => complete filename and path of the user chosen output file.

%I will be replaced by the actual intermediate file name

%O will be replaced by the actual output file name (the final netlist file)

And a command line could be

- under Windows:  
***f:/kicad/bin/xsltproc.exe -o %O f:/kicad/bin/plugins/netlist\_form\_pads-pcb.xml %I***
- under Linux)  
***xsltproc -o %O /usr/local/kicad/bin/plugins/netlist\_form\_pads-pcb.xml %I***

Assuming ***xsltproc*** is installed on your PC b(under Windows, all files in *kicad/bin*).

## 14.4 - The Intermediate Netlist structure:

This sample gives an idea of the file.

```
<?xml version="1.0" encoding="utf-8"?>
<export version="D">
  <design>
    <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
    <date>29/08/2010 21:07:51</date>
    <tool>eeschema (2010-08-28 BZR 2458)-unstable</tool>
  </design>
  <components>
    <comp ref="P1">
      <value>CONN_4</value>
      <libsource lib="conn" part="CONN_4"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20BA</tstamp>
    </comp>
    <comp ref="U1">
      <value>74LS04</value>
      <libsource lib="74xx" part="74LS04"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20A6</tstamp>
    </comp>
  </components>
</export>
```

```
</comp>
<comp ref="C1">
  <value>CP</value>
  <libsource lib="device" part="CP"/>
  <sheetpath names="/" tstamps="/">
  <tstamp>4C6E2094</tstamp>
</comp>
<comp ref="R1">
  <value>R</value>
  <libsource lib="device" part="R"/>
  <sheetpath names="/" tstamps="/">
  <tstamp>4C6E208A</tstamp>
</comp>
</components>
<libparts/>
<libraries/>
<nets>
  <net code="1" name="GND">
    <node ref="U1" pin="7"/>
    <node ref="C1" pin="2"/>
    <node ref="U2" pin="7"/>
    <node ref="P1" pin="4"/>
  </net>
  <net code="2" name="VCC">
    <node ref="R1" pin="1"/>
    <node ref="U1" pin="14"/>
    <node ref="U2" pin="4"/>
    <node ref="U2" pin="1"/>
    <node ref="U2" pin="14"/>
    <node ref="P1" pin="1"/>
  </net>
  <net code="3" name="">
    <node ref="U2" pin="6"/>
  </net>
  <net code="4" name="">
    <node ref="U1" pin="2"/>
    <node ref="U2" pin="3"/>
  </net>
  <net code="5" name="/SIG_OUT">
    <node ref="P1" pin="2"/>
    <node ref="U2" pin="5"/>
    <node ref="U2" pin="2"/>
  </net>
  <net code="6" name="/CLOCK_IN">
    <node ref="R1" pin="2"/>
    <node ref="C1" pin="1"/>
    <node ref="U1" pin="1"/>
    <node ref="P1" pin="3"/>
  </net>
</nets>
</export>
```

### 14.4.1 - General structure:

The intermediate Netlist has 5 sections:

- the header section.
- The component section.
- The lib parts section.
- The libraries section.
- The nets section.

The whole file has the delimiter <export>

```
<export version="D">
...
</export>
```

### 14.4.2 - The header section:

The header has the delimiter <design>



```
<design>
  <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
  <date>21/08/2010 08:12:08</date>
  <tool>eeschema (2010-08-09 BZR 2439)-unstable</tool>
</design>
```

This section can be seen as comments.

### 14.4.3 - The components section:

The component section has the delimiter <components>

```
<components>
  <comp ref="P1">
    <value>CONN_4</value>
    <libsource lib="conn" part="CONN_4"/>
    <sheetpath names="/" tstamps="/" />
    <tstamp>4C6E2141</tstamp>
  </comp>
</components>
```

This is the list of components in schematic.  
Each component is described like this:

```
<comp ref="P1">
  <value>CONN_4</value>
  <libsource lib="conn" part="CONN_4"/>
  <sheetpath names="/" tstamps="/" />
  <tstamp>4C6E2141</tstamp>
</comp>
```

<b>libsource</b>	name of the lib where this component was found.
<b>part</b>	component name inside this library.
<b>sheetpath</b>	« path » of the sheet inside the hierarchy: identify the sheet within the full schematic hierarchy.
<b>tstamps (time stamps)</b>	time stamp of the schematic file.
<b>tstamp (time stamp)</b>	time stamp of the component.

#### 14.4.3.1 - Note about time stamps for components:

To identify a component in a netlist (and therefore on a board), the reference is used, and is unique for each component. However Kicad provides an auxiliary way to identify a component, and its corresponding footprint on the board. This allows the re-annotation of components in a schematic project and do not loose the link between the component and its footprint.

A time stamp is an unique identifier for each component or sheet in a schematic project.

However, in complex hierarchies, the same sheet is used more than once, so this sheet contains components having the same time stamp.

A given sheet (inside a complex hierarchy) has an unique identifier: its sheetpath.

A given component (inside a complex hierarchy) has an unique identifier: the sheetpath+its tstamp

### 14.4.4 - The libparts section:

The libparts section has the delimiter <libparts>, and the data in this section is defined in the schematic libraries  
Its contains for each component:

- The allowed footprints names (names use jokers) delimiter <fp>
- The fields defined in the library delimiter <fields>.
- The list of pins delimiter <pins>

```
<libparts>
  <libpart lib="device" part="CP">
    <description>Condensateur polarise</description>
    <footprints>
```

```
<fp>CP*</fp>
<fp>SM*</fp>
</footprints>
<fields>
  <field name="Reference">C</field>
  <field name="Valeur">CP</field>
</fields>
<pins>
  <pin num="1" name="1" type="passive"/>
  <pin num="2" name="2" type="passive"/>
</pins>
</libpart>
</libparts>
```

Lines like `<pin num="1" type="passive"/>` give also the electrical pin type.  
Electrical pin types are:

Input	Usual input pin
Output	Usual output
Bidirectional	Input or Output
Tri-state	Bus input/output
Passive	Usual ends of passive components
Unspecified	Unknown electrical type
Power input	Power input of a component
Power output	Power output like a regulator output
Open collector	Open collector often found in analog comparators
Open emitter	Open collector sometimes found in logic.
Not connected	Must be left open in schematic

14.4.5 - The libraries section:

The libraries section has the delimiter `<libraries>`  
Gives the list of schematic libraries used in the schematic project.

```
<libraries>
  <library logical="device">
    <uri>F:\kicad\share\library\device.lib</uri>
  </library>
  <library logical="conn">
    <uri>F:\kicad\share\library\conn.lib</uri>
  </library>
</libraries>
```

14.4.6 - The nets section:

The nets section has the delimiter `<nets>`.  
This is the connectivity of the schematic.

```
<nets>
  <net code="1" name="GND">
    <node ref="U1" pin="7"/>
    <node ref="C1" pin="2"/>
    <node ref="U2" pin="7"/>
    <node ref="P1" pin="4"/>
  </net>
  <net code="2" name="VCC">
    <node ref="R1" pin="1"/>
    <node ref="U1" pin="14"/>
  </net>
```

```
<node ref="U2" pin="4"/>
<node ref="U2" pin="1"/>
<node ref="U2" pin="14"/>
<node ref="P1" pin="1"/>
</net>
</nets>
```

This sections lists all nets in the schematic.

A given net is :

```
<net code="1" name="GND">
  <node ref="U1" pin="7"/>
  <node ref="C1" pin="2"/>
  <node ref="U2" pin="7"/>
  <node ref="P1" pin="4"/>
</net>
```

net code	is an internal identifier for this net
name	is a name for this net
node	give a pin reference connected to this net

### 14.5 - More info about xsltproc:

See <http://xmlsoft.org/XSLT/xsltproc.html>

#### 14.5.1 - Introduction

xsltproc is a command line tool for applying XSLT stylesheets to XML documents.

It is part of libxslt, the XSLT C library for GNOME.

While it was developed as part of the GNOME project, it can operate independently of the GNOME desktop.

xsltproc is invoked from the command line with the name of the stylesheet to be used followed by the name of the file or files to which the stylesheet is to be applied. It will use the standard input if a filename provided is - .

If a stylesheet is included in an XML document with a Stylesheet Processing Instruction, no stylesheet need be named at the command line. xsltproc will automatically detect the included stylesheet and use it.

By default, output is to stdout. You can specify a file for output using the -o option.

#### 14.5.2 - Synopsis

```
xsltproc [[-V] | [-v] | [-o file] | [--timing] | [--repeat] | [--debug] | [--novalid] | [--noout] | [--maxdepth val] | [--html] | [--param name value] | [--stringparam name value] | [--nonet] | [--path paths] | [--load-trace] | [--catalogs] | [--xinclude] | [--profile] | [--dumpextensions] | [--nowrite] | [--nomkdir] | [--writesubtree] | [--nodtdattr] | [stylesheet] [file1] [file2] [...]]
```

#### 14.5.3 - Command Line Options

-V or --version

Show the version of libxml and libxslt used.

-v or --verbose

Output each step taken by xsltproc in processing the stylesheet and the document.

-o or --output *file*

Direct output to the file named *file*. For multiple outputs, also known as "chunking", -o *directory/* directs the output files to a specified directory. The directory must already exist.

--timing

Display the time used for parsing the stylesheet, parsing the document and applying the stylesheet and saving the result. Displayed in milliseconds.

--repeat

Run the transformation 20 times. Used for timing tests.

--debug

Output an XML tree of the transformed document for debugging purposes.

--novalid

Skip loading the document's DTD.

--noout

## Eeschema - Customize Netlists and BOM

- Do not output the result.
- maxdepth *value*  
Adjust the maximum depth of the template stack before libxslt concludes it is in an infinite loop. The default is 500.
- html  
The input document is an HTML file.
- param *name value*  
Pass a parameter of name *name* and value *value* to the stylesheet. You may pass multiple name/value pairs up to a maximum of 32. If the value being passed is a string rather than a node identifier, use --stringparam instead.
- stringparam *name value*  
Pass a parameter of name *name* and value *value* where *value* is a string rather than a node identifier. (Note: The string must be utf-8.)
- nonet  
Do not use the Internet to fetch DTD's, entities or documents.
- path *paths*  
Use the list (separated by space or column) of filesystem paths specified by *paths* to load DTDs, entities or documents.
- load-trace  
Display to stderr all the documents loaded during the processing.
- catalogs  
Use the SGML catalog specified in SGML\_CATALOG\_FILES to resolve the location of external entities. By default, xsltproc looks for the catalog specified in XML\_CATALOG\_FILES. If that is not specified, it uses /etc/xml/catalog.
- xinclude  
Process the input document using the Xinclude specification. More details on this can be found in the Xinclude specification: <http://www.w3.org/TR/xinclude/>
- profile or --norman  
Output profiling information detailing the amount of time spent in each part of the stylesheet. This is useful in optimizing stylesheet performance.
- dumpextensions  
Dumps the list of all registered extensions on stdout.
- nowrite  
Refuses to write to any file or resource.
- nomkdir  
Refuses to create directories.
- writesubtree *path*  
Allow file write only within the *path* subtree.
- nodtdattr  
Do not apply default attributes from the document's DTD.

### 14.5.4 - Return values

xsltproc's return codes provide information that can be used when calling it from scripts.

- 0: normal
- 1: no argument
- 2: too many parameters
- 3: unknown option
- 4: failed to parse the stylesheet
- 5: error in the stylesheet
- 6: error in one of the documents
- 7: unsupported xsl:output method
- 8: string parameter contains both quote and double-quotes
- 9: internal processing error
- 10: processing was stopped by a terminating message
- 11: could not write the result to the output file

### 14.5.5 - More Information

libxml web page: <http://www.xmlsoft.org/>  
W3C XSLT page: <http://www.w3.org/TR/xslt>