

Partage de savoirs « Maîtriser KiCad – I »

Les Usines Nouvelles - Bernard Pradines – 14 octobre 2017.

Site officiel KiCad : <http://kicad-pcb.org>

Utilisation de la suite KiCad V.4 :

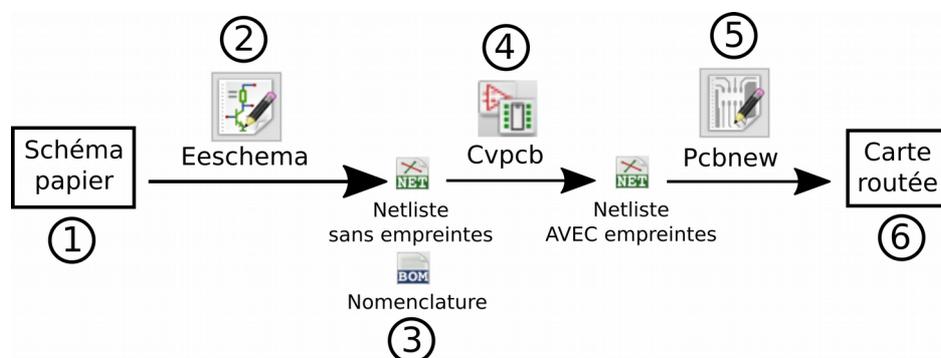
La fenêtre principale du logiciel sert de gestionnaire de « projet », avec les icônes des différents programmes à lancer :

- « Fichiers » permet de choisir de travailler sur un projet existant, ou d'en créer un nouveau ;
- penser avant de quitter à enregistrer son projet en cliquant sur l'icône : «  Sauver le projet courant » ;

Dans la fenêtre il y a plein d'icônes, mais seulement deux servent souvent :

-  « Eeschema » la saisie de schéma. En effet il faut saisir votre schéma papier, pour le reproduire à l'écran, et le préparer avant de le transformer en circuit imprimé ;
-  « Pcbnew » l'éditeur de circuit imprimé dans lequel vous allez placer les empreintes des composants sur votre carte, et router les pistes cuivrées.

Voici une illustration du cheminement dans KiCad :



1. On part d'un schéma dessiné sur papier, étape indispensable pour être plus efficace (et faire moins d'essais / erreurs / corrections lors de la saisie) ;

Le schéma qui va nous servir d'exercice tout au long de cette présentation est très simple comme vous pouvez le voir ci-dessous. Il s'agit d'un petit testeur pour maquette de feu tricolore.

- On exporte la nomenclature du schéma (BOM, ou liste des composants). Cette liste sert à commander les composants immédiatement, et on attend de les recevoir, car pour l'étape suivante il est indispensable de disposer des composants réels entre ses mains ;

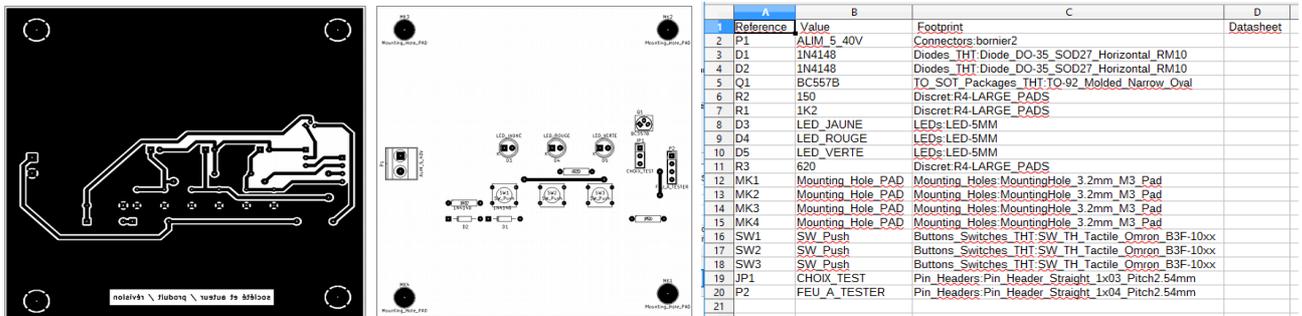
Reference	Quantity	Value	Footprint
D1 D2	2	1N4148	Diodes_THT:D_DO-35_SOD27_P10.16mm_Horizontal
D3	1	LED JAUNE	LEDs:LED_D5.0mm
D4	1	LED ROUGE	LEDs:LED_D5.0mm
D5	1	LED VERTE	LEDs:LED_D5.0mm
JP1	1	CHOIX_TEST	Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
MK1 MK2 MK3 MK4	4	Mounting_Hole_PAD	Mounting_Holes:MountingHole_3.2mm_M3_Pad
P1	1	ALIM_5_40V	Connectors:bornier2
P2	1	FEU_A_TESTER	Pin_Headers:Pin_Header_Straight_1x04_Pitch2.54mm
Q1	1	BC557B	TO_SOT_Packages_THT:TQ-92_Molded_Narrow_Oval
R1	1	1K2	Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R2	1	1150	Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R3	1	620	Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
SW1 SW2 SW3	3	SW_Push	Buttons_Switches_THT:SW_PUSH_6mm_h4.3mm

- On utilise Cypcb (programme qui se lance depuis Eeschema). On cherche dans les bibliothèques d'empreintes de KiCad, pour chaque composant du schéma, l'empreinte de composant correspondant au composant réel qu'on a reçu, empreinte qui sera dessinée sur le circuit imprimé. Si l'empreinte n'existe pas dans les bibliothèques standard, il faut la créer à l'aide de l'éditeur d'empreintes, et l'associer au composant correspondant. Quand on a trouvé une empreinte pour chaque composant du schéma, on enregistre son travail, et on exporte à nouveau la Netliste à partir de Eeschema ;

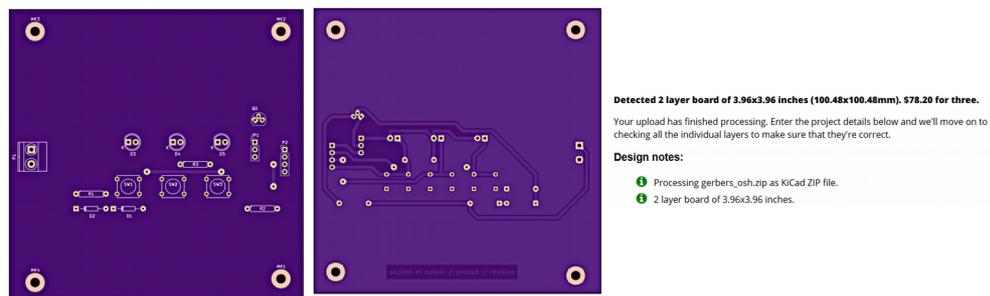
Reference	Value	Footprint
D1	1N4148	Diodes_THT:Diode_DO-35_SOD27_Horizontal_RM10
D2	1N4148	Diodes_THT:Diode_DO-35_SOD27_Horizontal_RM10
D3	LED_JAUNE	LEDs:LED-5MM
D4	LED_ROUGE	LEDs:LED-5MM
D5	LED_VERTE	LEDs:LED-5MM
JP1	CHOIX_TEST	Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
MK1	Mounting_Hole_PAD	Mounting_Holes:MountingHole_3.2mm_M3_Pad
MK2	Mounting_Hole_PAD	Mounting_Holes:MountingHole_3.2mm_M3_Pad
MK3	Mounting_Hole_PAD	Mounting_Holes:MountingHole_3.2mm_M3_Pad
MK4	Mounting_Hole_PAD	Mounting_Holes:MountingHole_3.2mm_M3_Pad
P1	ALIM_5_40V	Connectors:bornier2
P2	FEU_A_TESTER	Pin_Headers:Pin_Header_Straight_1x04_Pitch2.54mm
Q1	BC557B	TO_SOT_Packages_THT:TQ-92_Molded_Narrow_Oval
R1	1K2	Discret:R4-LARGE_PADS
R2	150	Discret:R4-LARGE_PADS
R3	620	Discret:R4-LARGE_PADS
SW1	SW_Push	Buttons_Switches_THT:SW_TH_Tactile_Omron_B3F-10xx
SW2	SW_Push	Buttons_Switches_THT:SW_TH_Tactile_Omron_B3F-10xx
SW3	SW_Push	Buttons_Switches_THT:SW_TH_Tactile_Omron_B3F-10xx

- Lorsque la carte est complètement routée (plus aucune erreur), on imprime un dossier de fabrication papier pour garder une trace de son travail à long terme. Puis on exporte les fichiers de fabrication (fichiers Gerber), pour les envoyer à un service de fabrication de C.I., ou bien on imprime soi-même les typons nécessaires à la réalisation manuelle de la carte.

Eléments du dossier de fabrication :



Visualisation des fichiers de fabrication (ici par OSH Park) :



Nous allons faire ensemble pas-à-pas chacune des étapes qui mènent à ce résultat.

Démarche pour la saisie de schéma :

- créer un dossier pour le projet, un nouveau projet dans ce dossier, et lancer Eeschema, remplir le cartouche, et sauvegarder la feuille du schéma, ce qui l'ajoute au projet ;
- modifier la liste des bibliothèques de composants utilisées dans le schéma ;
- saisir le schéma : privilégier l'approche par les entrées / sorties, commencer par les connecteurs ;
- éventuellement créer une hiérarchie de feuilles de schémas imbriquées dans la feuille principale, puis créer des symboles hiérarchiques pour relier des signaux entre les différentes feuilles ;
- ajouter des composants, saisir leurs valeurs (1K2, 150, ...), et relier les composants par des fils, des labels ou des bus ;
- rajouter les symboles d'alimentation (Vcc, GND, ...) et le POWER_FLAG ;

- ajouter les trous de fixation (dans mechanical.lib), et les relier à GND si nécessaire ;
- penser au symbole de non-connexion pour les pins inutilisées ;
- numéroter automatiquement les composants (R1, R2, ...) ;
- faire le test ERC (test des règles électriques) pour détecter les erreurs ;
- lancer «  CvPcb » pour associer une empreinte (boîtier) à chaque composant du schéma, y-compris les trous de fixation :
 - choisir d'utiliser les librairies en ligne (Github), ou celles installées sur le PC avec KiCad ;
 - si les empreintes des librairies ne suffisent pas, créer sa propre empreinte (bien chercher si pas déjà faite avant!), et éventuellement la soumettre à l'équipe du projet des librairies d'empreintes sur Github, en respectant ses règles ;
 - contrôler que chaque composant a bien un modèle 3D associé (préférer les empreintes qui en ont un).
- enregistrer l'association composant ↔ empreinte depuis CvPcb, et enregistrer tout de suite après le schéma dans Eeschema ;
- faire la liste des composants : utiliser l'icône «  BOM » et son plugin bom2csv pour exporter un fichier .csv vers votre tableur ;
- dans Eeschema, exporter la Netliste avec l'icône «  NET », et lancer «  Pcbnew » dans la foulée : la saisie du schéma est terminée, on peut maintenant router le circuit imprimé.

Librairies de composants schématiques (V 4.0.7) :

Nom de la librairie	Contenu	Notes perso
74xgxx.lib	Fonctions logiques, souvent CMS, avec une seule ou 2 portes logiques pour gagner de la place sur le C.I.	
74xx.lib	TTL LS, et quelques circuits CMOS HC / HCT	
ttl_ieee.lib	Séries TTL 7400 et 74LS avec symboles à la norme européenne	
cmos_ieee.lib	Série CMOS 4000 avec symboles à la norme européenne	
cmos4000.lib	Boîtiers les plus courants des séries CMOS 4000 et 4500	
ac-dc.lib	Circuits de commande de Mosfet	
adc-dac.lib	Très grand choix de convertisseurs A/N, et N/A (DAC)	
analog_switches.lib	Interrupteurs analogiques à commande logique, multiplexeurs, ...	
audio.lib	Amplificateurs audio intégrés entre autres	
battery_management.lib	Chargeurs de batteries / accus / ...	
bbd.lib	Lignes à retard analogiques	
conn.lib	Tous types de connecteurs, contacts, points de test, ...	

contrib.lib	1 composant : FX614 modem FSK	
dc-dc.lib Power_Management.lib	Convertisseurs DC-DC, C.I. d'alim. à découpage, ...	
regul.lib	Régulateurs de tension intégrés (7805, low-drop, ...)	
device.lib	Tous les composants courants : condos, quartz, diodes, bobinages, fusibles, jumpers, inductances, LED, microphone, potars, polyfuse, tous types de transistors, résistances, encodeurs rotatifs, cellules solaires, Haut-parleur, thermistance, transfo, ...	
digital-audio.lib	Circuits spécialisés dans le domaine	
diode.lib	Quelques références précises de diodes	
display.lib	Afficheurs 7 segments, LCD, ...	
dsp.lib	Deux processeurs de traitement du signal	
elec-unifil.lib	Electricité : Disjoncteurs, fusibles, interrupteurs, ...	
ESD_Protection.lib	Composants de protection à diodes, pour USB, ...	
graphic_symbols.lib	Logo 'Open-Hardware' Flèches rouges Symboles d'alerte et de sécurité (ESD, THT, ...)	
interface.lib	ex. : MAX232, interface CAN, ...	
leds	Modules spécifiques d'éclairage à base de LEDs	
Worldsemi.lib	WS2812 LED RGB à contrôleur intégré, WS2822	
linear.lib	Ampli-ops, comparateurs, ...	
logic_programmable actel.lib altera.lib Lattice.lib xilinx.lib	Logique programmable : PAL, GAL, FPGA	
mechanical.lib	Radiateurs, trous de perçage, ...	
memory.lib	EEPROM I ² C et SPI, SIMM, SRAM, ...	
microcontrollers.lib atmel.lib (atmega, attiny, avr-isp) hc11.lib microchip_dspic33dsc.lib microchip_pic10mcu.lib microchip_pic12mcu.lib microchip_pic16mcu.lib microchip_pic18mcu.lib microchip_pic24mcu.lib microchip_pic32mcu.lib msp430.lib nxp_armmcu.lib stm32.lib stm8.lib Zilog.lib (Z80)	Microcontrôleurs de différentes marques	
modules.lib	Cartes Arduino, NUCLEO, Pololu	
motor_drivers.lib	ex. : L298	
motors.lib	Ventilateurs, moteurs AC et DC, servomoteurs,	

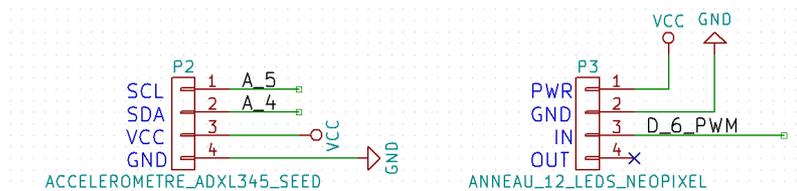
	moteurs pas-à-pas	
opto.lib	Optocoupleurs, très nombreuses variantes.	
Oscillators.lib	ex. : Si570	
power.lib	Alimentations et masses (utiliser l'icône dédiée dans Eeschema)	
pspice.lib	Composants de base pour la simulation analogique	
references.lib	Références de tension	
relays.lib	Relais Finder, ...	
rfcom.lib	Modules Bluetooth, WiFi, RF (nRF24L01), ...	
sensors.lib	Divers capteurs dont DHT11, KTY10, LM35, ...	
switches.lib	dip switches, interrupteurs, commutateurs, boutons poussoirs, relais reed, commutateurs rotatifs	
transf.lib	Transformateurs	
transistors.lib	Transistors des types les plus communs, par référence	
triac_thyristor	Quelques triacs et thyristors	
valves.lib	Tubes (les plus connus)	
video.lib	Circuits intégrés de traitement vidéo analogique : détection de synchro, amplis rapides, ...	
allegro.lib analog_devices.lib bosch.lib brooktre.lib cypress.lib ftdi.lib gennum.lib infineon.lib intel.lib intersil.lib ir.lib LEM.lib maxim.lib microchip.lib motorola.lib nordicsemi.lib nxp.lib onsemi.lib philips.lib powerint.lib RFSolutions.lib silabs.lib siliconi.lib supertex.lib texas.lib wiznet.lib Xicor.lib zetex.lib	Composants spécifiques au fabricant (librairies non-listées plus haut)	

Trouver un composant dans les librairies :

- chercher d'abord dans device.lib les composants génériques les plus courants (résistance, condensateur, diode, transistor, ...)
- si vous avez une référence précise de diode ou de transistor, chercher dans les librairies spécialisées : diode.lib, transistors.lib ;
- pour les circuits intégrés logiques, si votre référence manque dans une famille logique donnée, cherchez la même dans une autre famille logique : si le brochage est exactement le même (vérifier les datasheet), changez simplement la référence sur le schéma pour mettre la vôtre !
- pour un composant dont vous connaissez le fabricant, si votre référence ne figure pas dans la librairie dédiée au domaine d'application, pensez à chercher aussi dans la librairie portant le nom du fabricant (ex. : le capteur de température DS18B20 n'est pas dans sensors.lib, mais dans maxim.lib) ;
- un domaine d'application peut être couvert par plusieurs librairies : dans le tableau ci-dessus, j'ai essayé de regrouper ces librairies. Par exemple, il y a des circuits intégrés d'alimentation à découpage dans dc-dc.lib, et dans Power_Management.lib, mais aussi des régulateurs de tension dans regul.lib et des chargeurs de batteries ou accus dans battery_management.lib ;
- éléments fréquemment recherchés : les trous de fixation sont dans mechanical.lib, et les boutons poussoirs sont dans switches.lib, mais ces deux librairies ne sont pas incluses dans le projet par défaut !
- ne pas hésiter à parcourir fréquemment toutes les librairies : c'est à force de les parcourir qu'on repère où se trouvent les composants, c'est une question d'habitude ;
- on peut trouver des composants pour KiCad en cherchant sur Internet : si vous téléchargez une librairie, mettez-la d'abord dans le dossier de votre projet pour la tester ; Vous pouvez également extraire un composant d'un schéma trouvé sur Internet et l'enregistrer dans une librairie créée par vos soins pour l'occasion, dans le dossier de votre projet ;
- on crée rarement un composant (symbole) schématique, car les librairies existantes sont déjà très fournies et les composants très nouveaux sont rares ;
- la solution n'est pas toujours de trouver un composant tout fait, mais plutôt des idées dans les schémas des autres : par exemple, dans le dossier « C:\Program Files\KiCad\share\kicad\template » il y a des exemples de schémas autour d'Arduino, Raspberry Pi, ... De nombreux bloggeurs donnent l'adresse de leur Github dans lequel vous trouvez souvent un schéma Kicad avec des composants originaux.

Création de composants pour votre schéma :

Créer un composant personnalisé avec l'éditeur de librairie d'Eeschema, c'est une étape qu'on peut soigneusement éviter, ou contourner, si on veut passer le moins de temps possible à saisir son schéma. En effet, même si un composant n'est pas dans les bibliothèques de symboles de KiCad, vous pouvez aisément le représenter sur votre schéma sous la forme d'un simple connecteur (à une ou deux rangées). C'est surtout vrai pour les composants simples, ou les modules type « breakout board » avec peu de broches.

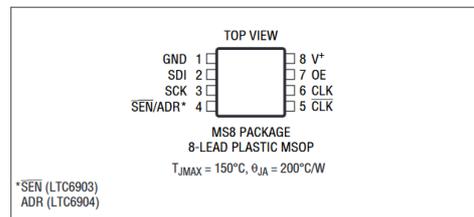


Avec un peu d'expérience, faire son propre composant à partir d'un autre qui lui ressemble (fortement) par simple modification, ne prend pas tellement plus de temps que de placer un connecteur sur son schéma et saisir les noms des signaux ... Si le composant qui vous manque est suffisamment fréquemment employé pour vous (re-)servir dans plusieurs schémas, il ne faut pas hésiter : autant apprendre à le faire ... maintenant.

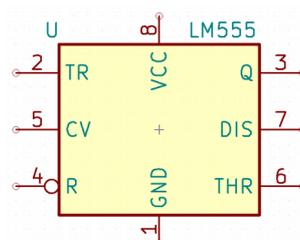
1) Etape 1 / 2 : Copier un composant schématique existant déjà, dans une nouvelle librairie (.lib).

Nous allons créer le LTC6904 de Linear Technology, un oscillateur en boîtier MSOP-8 (CMS avec un corps carré en plastique de 3 x 3 mm et des broches au pas de 0,65 mm donc minuscule !!!) dont la fréquence (du signal carré produit) est programmable par le bus I2C entre 1 kHz et 68 MHz.

PIN CONFIGURATION



Il s'agit en fait d'un simple circuit intégré à 8 broches. On va donc copier un autre composant intégré à 8 broches : le LM555. On cherche dans quelle librairie se trouve notre modèle, à l'aide du «  Visualisateur de librairies » à lancer depuis Eeschema (barre d'icônes du haut de la fenêtre). Après une courte recherche on le trouve dans la librairie « linear » :

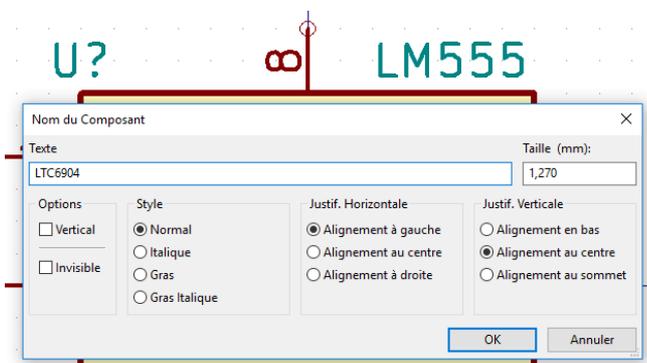


C'est un circuit intégré qui se représente sous forme d'un rectangle : c'est le plus simple qu'on puisse faire. KiCad peut gérer des types de composants plus compliqués. Il existe des composants avec multiples unités : ce sont les circuits logiques qui comportent plusieurs « portes » identiques (plusieurs unités) pouvant elles-

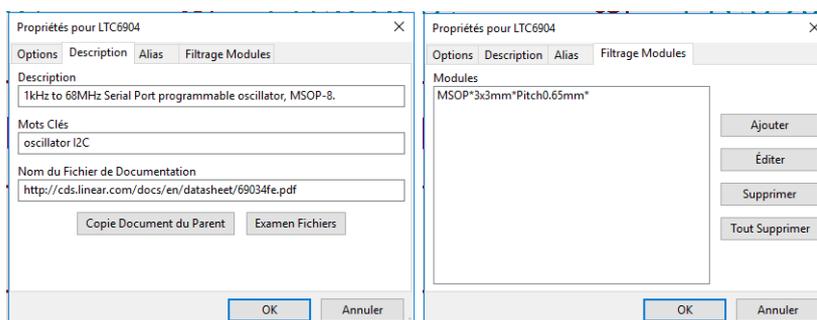
mêmes avoir plusieurs représentations, ou encore les AOP à 2 ou 4 unités dans un même boîtier ... Les composants peuvent avoir des alias : ce sont des références différentes (chez plusieurs fabricants : NE555, LM555, ...) d'un même symbole. Evidemment, il ne faut pas prendre un alias comme base ! Ici on va aller au plus simple : copier un circuit intégré à 8 broches.

Pour faire la copie, on ferme le visualiseur de bibliothèques et on ouvre, toujours depuis Eeschema, l' « Éditeur de bibliothèque ». On change la bibliothèque de travail en cliquant sur « Sélection de la bibliothèque de travail » et on choisit « linear » dans la boîte de dialogue. On clique sur « Charger un composant à éditer » et dans la liste de la boîte de dialogue on choisit le LM555.

Il faut renommer le composant : dans l'éditeur, faire un clic droit de la souris sur l'étiquette verte « LM555 » au-dessus du carré rouge à fond jaune du composant et choisir « Edition du champ », puis saisir LTC6904 et valider. Le logiciel vous propose d'effacer tous les alias en conflit, ce qui va vider l'onglet « Alias » de la fenêtre des propriétés (voir plus loin) : acceptez !



On modifie les informations du composant. Cliquer sur « Éditer propriétés du composant » :



- l'onglet « Options » reste inchangé ;
- l'onglet « Description » et l'onglet « Filtrage Modules » sont complétés selon les illustrations ci-dessus (le circuit existe uniquement en boîtier MSOP-8) ;
- l'onglet « Alias » a déjà été vidé lors du renommage.

Le champ « Mots clés » de l'onglet « Description » permet de retrouver le composant facilement dans le filtre de recherche de la boîte de dialogue d'ajout de composant sur le schéma (outil puissant). Ainsi, le LTC6904 figurera dans la liste des oscillateurs, et dans la liste des composants I2C. Un choix judicieux des mots clés vous simplifiera la vie lors de la saisie de vos futurs schémas : testez vos mots clés pour voir s'ils sont efficaces.

Notez la syntaxe des filtres d'empreintes dans l'onglet « Filtrage Modules » : le caractère « * » remplace un ou plusieurs caractères, et l'ordre des autres caractères est important. Par exemple, le filtre « MSOP*3x3mm*Pitch0.65mm* » permet de retrouver les empreintes dont le nom commence par « MSOP » (on ne met pas le nombre de broches car KiCad possède ce filtrage optionnel), puis contient plus loin « 3x3mm » et encore plus loin (dans cet ordre précis) la mention « Pitch0.65mm », et enfin une éventuelle spécialisation d'où le « * » final. Si une variante plus large du composant existait (par exemple « 4x4mm » au lieu de « 3x3mm »), il faudrait ajouter une autre ligne de filtrage (bouton « Ajouter ») du type « MSOP*4x4mm*Pitch0.65mm* ». Le but ici est de lister toutes les variantes de boîtiers existant chez le constructeur pour ce composant (ici c'est simple il n'y a qu'un seul boîtier – donc il faudra aussi indiquer directement le bon boîtier dans le champ « Empreinte » ...).

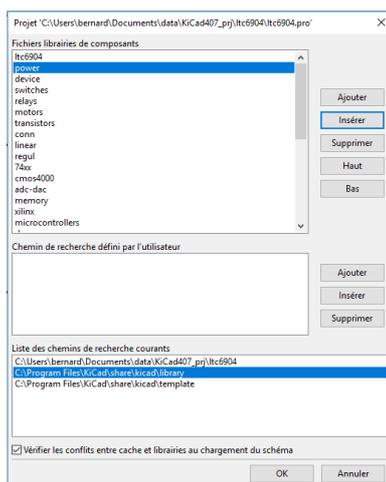
Pour enregistrer la copie, on crée une nouvelle librairie, dans le dossier de votre projet en cours, avec juste ce composant dedans : cliquez sur l'icône «  Sauver le composant courant dans une nouvelle librairie ». Sélectionnez le dossier de votre projet KiCad, et nommez la librairie : « ltc6904.lib ».

2) Etape 2 / 2 : Modifier le composant copié.

a) Charger le composant à modifier.

Vous venez de copier et renommer un composant sans rien modifier d'autre : il reste du travail à faire. Avant de continuer, il faut ajouter la nouvelle librairie (.lib) créée à votre projet. Dans le menu de l'éditeur de composants, choisissez « Préférences → Librairies de composants », et dans la boîte de dialogue :

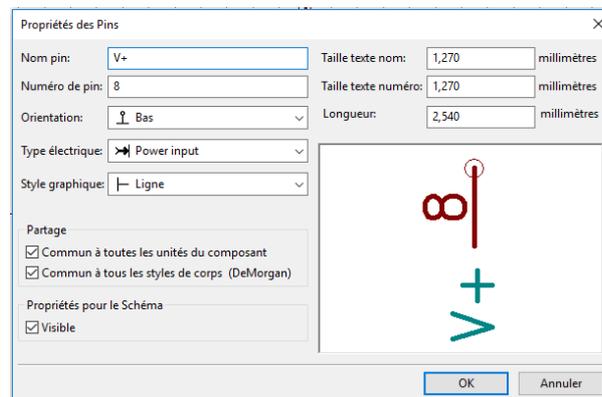
- cliquez sur la première ligne de la liste (ici « power ») ;
- cliquez sur le bouton « Insérer » pour ajouter votre librairie en tête de liste. Allez dans le dossier de votre projet et sélectionnez le fichier « ltc6904.lib » ;
- la librairie apparaît dans la liste de votre projet. Validez la boîte de dialogue qui se referme.



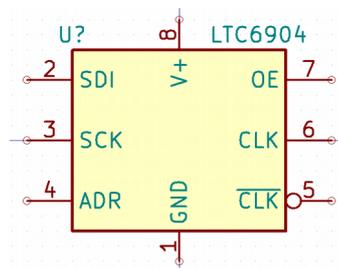
Changez maintenant de librairie courante : cliquez sur « Sélection de la librairie de travail » et choisissez « ltc6904 » puis validez. Chargez le composant à modifier : cliquez sur « Charger un composant à éditer » et cliquez sur le LTC6904. Il doit apparaître dans l'éditeur.

b) Editer les broches du composant.

Faites un clic du bouton droit de la souris sur chaque broche du composant, cliquez sur « Editer Pin », renommez et renumérotez chaque broche. Pensez aussi à changer le type électrique (power input ou output, entrée, sortie, passive, ...) et le style graphique (le plus souvent : Ligne, sauf pour la broche 5 : Inversé).



Par exemple, pour la broche 8 nommée « VCC », renommez simplement en « V+ ». Les broches SDI, SCK, ADR et OE sont des entrées ; CLK et ~CLK sont des sorties (saisir le ~ tilde – puis appuyer sur « espace » – en tête du nom de la broche 5 car le signal est inversé). Seule la broche 1 reste inchangée. Vous devez arriver à ceci :



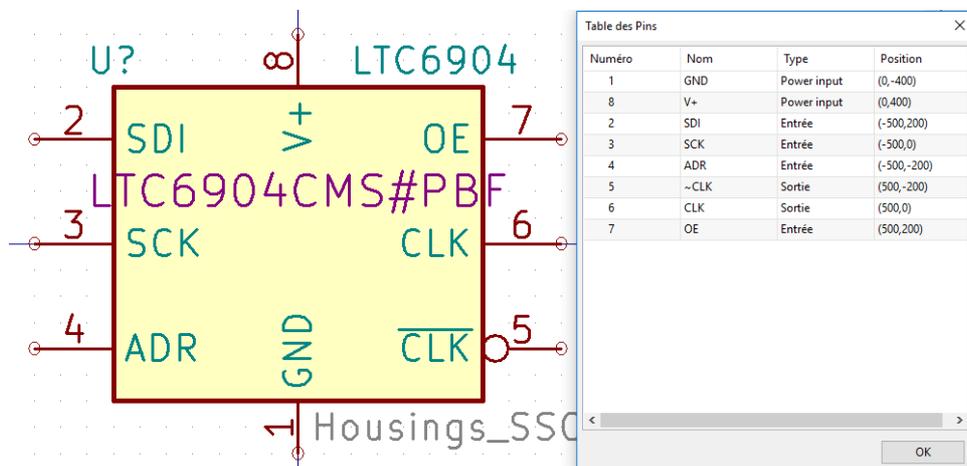
c) Types électriques des broches.

L'approche de mettre un simple connecteur à la place d'un vrai composant a un gros inconvénient : elle empêche les vérifications électriques de se faire correctement ! Et on ne s'en rend compte qu'à la fin, car le « Test ERC » se fait en dernier. Donc cette astuce est à utiliser à la marge, en cas d'échéance imminente.

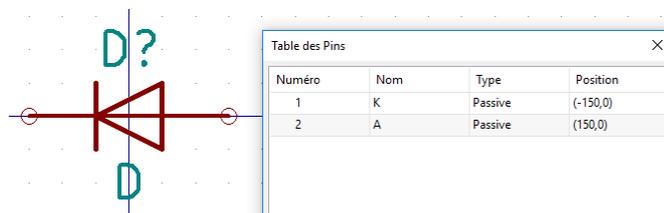
En effet, sur un connecteur, toutes les broches sont de type « Passive », c'est à dire non-typées en fait. « Passive » est le type passe-partout, utilisé quand on ne peut pas savoir à priori quel type de signal arrive sur la broche. Il ne faut pas le confondre avec « Bidirectionnel » qui est un type spécifique à certains circuits logiques (broches pouvant être configurées en entrée ou en sortie par exemple).

Le problème c'est que si les broches du connecteur ne sont pas typées, la présence des alimentations sur certaines broches n'est pas signalée à KiCad, et les vérifications du genre « connexion Vcc à GND », ou « connexion de deux sorties actives entre elles » ne peut pas être détectée ! Vous travaillez alors sans filet, et personne ne fait jamais d'erreur ... On retrouve ce problème dans les bibliothèques de composants issues de la conversion automatisée (par un script ULP) d'Eagle vers KiCad.

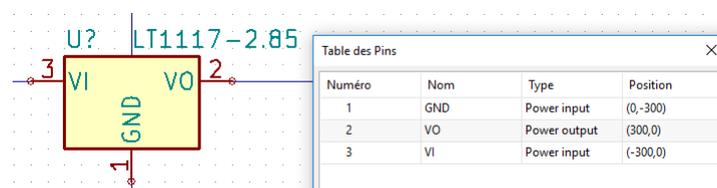
Voici les types utilisés pour les broches de notre composant : vous pouvez les afficher en cliquant sur «  Montrer la table des pins ».



Pour un composant simple : diode, bouton-poussoir, connecteur, ... on aura :



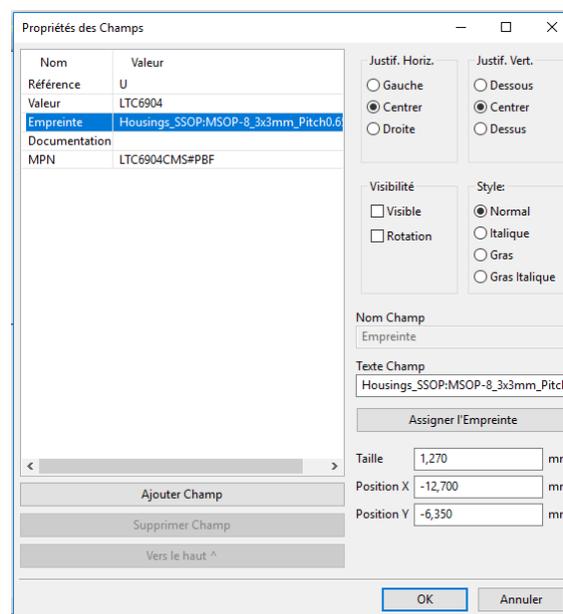
Par contre, pour un régulateur de tension, les types sont très importants :



Donc à minima, pensez à signaler les broches d'alimentation, à identifier correctement les entrées et les sorties, et mettez le reste en « Passive ». D'autres types électriques existent, à utiliser au cas par cas à l'aide de la liste déroulante de la boîte de dialogue « Editer Pin ».

d) Les champs du composant.

Il faut également vérifier que les « Champs » de notre composant sont correctement remplis : cliquer sur l'icône « T » dans la barre du haut de l'éditeur. Seul le champ invisible « Empreinte » doit être complété en cliquant sur le bouton « Assigner l'Empreinte » car le composant a un seul boîtier possible. Si ce n'était pas le cas, il ne faudrait rien mettre. Dans tous les cas les filtres d'empreintes (voir étape 1 / 2) doivent être spécifiés pour pouvoir chercher (et retrouver) l'empreinte dans Cypcb.



e) Enregistrer composant et librairie.

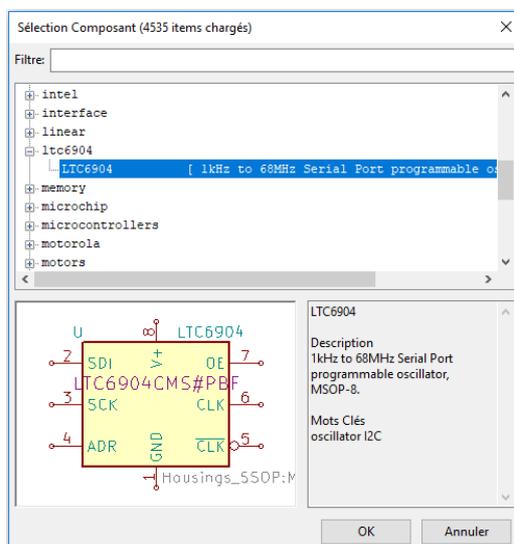
Enregistrez le composant en mémoire dans la librairie active : cliquez sur «  Mettre à jour le composant courant en librairie de travail » et confirmez. Puis enregistrez la librairie ainsi modifiée sur votre disque dur : cliquez sur «  Sauver la librairie courante sur disque ». Vous pouvez refermer l'éditeur de composants. Pensez à enregistrer votre schéma dans Eeschema.

A plusieurs reprises j'ai constaté que si le composant était déjà sur votre schéma avant sa modification en librairie, il reste en mémoire dans son état précédent ! Je conseille donc de :

- retirer du schéma le composant en version précédente,
- enregistrer le schéma, le projet, et quitter KiCad,
- relancer KiCad, rouvrir votre schéma, et ajouter le composant depuis la librairie modifiée.

f) Utiliser et partager le composant.

Maintenant, ajoutez votre nouveau composant à votre schéma pour le tester : dans Eeschema, cliquez sur l'icône «  Placer un composant », sélectionnez la librairie ltc6904, puis le composant LTC6904.



Cliquez sur OK, puis sur la feuille de votre schéma pour le placer : vous avez créé votre propre composant. En lançant Cypcb, vous pourrez vérifier que le filtrage par mots clés fonctionne : les boîtiers MSOP-8 sont proposés, il suffit de filtrer aussi par le nombre de broches « # » pour qu'un seul boîtier soit proposé.

Si vous êtes fier de votre composant, et souhaitez le partager avec la communauté, c'est possible. Des règles de conception sont à respecter pour que les librairies du projet KiCad restent homogènes. Un script Python vous permet de vérifier que vous les avez respectées, mais en copiant un composant existant vous mettez déjà toutes les chances de votre côté. Vous trouverez les règles ici : <https://github.com/KiCad/kicad-library/wiki/Kicad-Library-Convention> .

Alternative : formulaire **Quicklib** de création de composants schématiques

<http://kicad.rohrbacher.net/quicklib.php>

Composants : autres ressources :

- **Sites de librairies** de composants :
<https://www.snapeda.com> (intègre OPL et CPL)
- **Moteurs de recherche** de composants, et BOM :
<https://octopart.com>
- **Scripts Python de BOM** pour KiCad :
 - BOMs Away ! : <https://github.com/Jeff-Ciesielski/Boms-Away>
 - KiCost : <https://kicost.readthedocs.io/en/latest/>
 - Service spécialisé : <https://1clickbom.com/>
- **Librairies « Standards »** de composants :
 - **CPL** d'Octopart - intègre l'OPL. Existe en versions Prototypage THT et Production CMS :
<https://octopart.com/common-parts-library>
 - **OPL** de SeeedStudio : <https://www.seeedstudio.com/opl.html>

Gestion et création de bibliothèques d'empreintes de composants :

Dans le menu « Préférences » des applications Cypcb et Pcbnew, une option « Gestionnaire des bibliothèques d'empreintes » permet de voir quelles bibliothèques ont été configurées et sont disponibles. Les bibliothèques d'empreintes peuvent se trouver à deux endroits :

- sur votre disque dur (fonctionnalité à paramétrer dans le projet) :
 - « Bibliothèques spécifiques au projet » dans le dossier de votre projet en cours,
 - « Bibliothèques globales » dans le dossier des empreintes de KiCad (pour tous les projets) ;
- sur le compte Github de la communauté KiCad (fonctionnalité par défaut).

Personnaliser une empreinte existante :

Il faut travailler en copiant le composant le plus proche dans les bibliothèques KiCad, et le modifier ensuite pour le personnaliser en fonction de ses besoins :

- Lancer l'éditeur d'empreintes depuis la saisie de schéma avec l'icône  ;
- Utiliser la fonction «  Visualiseur d'empreintes » pour trouver le composant le plus proche. Dans le visualiseur, noter le nom de la bibliothèque dans laquelle se trouve votre composant de base ;
- Changer la bibliothèque en cours dans l'éditeur d'empreintes. Pour cela, cliquer sur l'icône de sélection de la bibliothèque active :  . Sélectionnez la bibliothèque dont vous avez noté le nom ;
- Chargez ensuite le composant de base que vous avez repéré auparavant : icône  puis sur le bouton « Sélection par Viewer ». Retrouvez le composant dans la liste, et double-cliquez sur son nom et validez la boîte de dialogue : le composant se charge dans la fenêtre d'édition ;
- Changez tout de suite le nom du composant. Pour cela, cliquez sur l'icône des propriétés de l'empreinte :  . Dans la boîte de dialogue, il faut tout de suite modifier les champs suivants :
 - champ « Valeur »,
 - champ « Nom du module en bibliothèque ».
- profitez-en pour éditer les autres champs, ça évite d'oublier de le faire après : champ « Doc », champ « Mots clés » : très important pour le trouver dans Cypcb. Champ « Référence » : n'y touchez pas.
- Validez la boîte de dialogue par « OK » pour la refermer.
- Enregistrez tout de suite le composant dans une nouvelle bibliothèque avec l'icône « Créer une nouvelle bibliothèque et y sauver l'empreinte » :  . Comme chemin, il faut choisir le dossier de votre projet Kicad, et coller à nouveau le nom du composant : vous obtiendrez une bibliothèque du même nom que le composant, dans le dossier de votre projet.
- Ajoutez tout de suite la nouvelle bibliothèque dans votre projet : dans le menu « Préférences → Assistant d'ajout des bibliothèques d'empreintes », cochez l'option « Fichiers sur mon ordinateur », sélectionnez ensuite le dossier de votre projet KiCad, puis dans le dossier, le sous-dossier de votre nouvelle bibliothèque, et cliquez sur « Next » ; Sélectionnez l'option « A la configuration de bibliothèque du projet

actuel » et validez avec le bouton « Finish » : le chemin de votre librairie est ajouté à la configuration du projet.

- Enregistrez tout de suite votre travail avec les icônes « Sauver le module en librairie de travail » :  , et dans l'éditeur de schéma, avec l'icône « Sauver le projet schématique » :  . En effet la liste des librairies se trouve dans le fichier de projet.
- Vous devez retourner ensuite dans l'éditeur d'empreinte pour modifier votre composant de base.

Edition du composant copié :

- L'icône en forme d'encre sert à définir un point d'origine pour votre composant ;
- Les pads correspondant aux broches du composant sont numérotés et correspondent aux broches de même numéro sur le symbole schématique. On peut avoir plusieurs broches avec le même numéro si elles sont reliées au même signal ;
- Les pads peuvent être modifiés (clic droit sur le pad). On peut changer la forme (cercle, carré, ovale, ...), le diamètre de perçage, et le diamètre du pad lui-même (padX) ;
- Taille des pads : n'hésitez pas à les agrandir, c'est un bon exercice de débutant que de créer des versions « pads larges » dans votre projet pour vous faciliter la soudure manuelle ;
- Tableau de diamètres de pads pour les composants traversants usuels (soudure manuelle) :

Composant	Diamètre de perçage (mm)			Diamètre des pads (padX – en mm)			
	0,8	1,0	1,2	1,778	2	2,54	3,81
Résistance 1/4 W	X				X		
Petits condensateurs	X				X		
Transistor TO92	X			X			
Connecteurs HE10 / 14		X			X		
Résistance 1 W		X				X	
Potars, poussoirs, inters		X				X	
Selfs pour alims, de choc		X				X	
Borniers, grosses diodes			X			X	
Résistance 5 W			X				X

- Repères pour la fixation du composant :
 - broches de fixation (parfois reliées à la masse) : on mettra un connecteur dans l’empreinte ;
 - vis : on mettra un pad pour repérer l’endroit où percer. Le pad ne sera pas numéroté pour ne pas déclencher d’erreur, car il ne correspond pas à une connexion électrique sur le composant (sauf mise à la masse) ;
- On figurera la taille exacte du composant par son contour (rectangle au minimum), pour éviter le chevauchement involontaire de composants ;
- Mettre l’abréviation de la fonction des broches à côté de leur pad, sous forme de texte (icône « T »). Pensez à faire figurer la polarité (signes + / -) des composants polarisés (condensateurs chimiques, connecteurs d’alimentation, LEDs, ...). Faire figurer également les repères nécessaires à l’orientation du composant (détrompage, sigle, flèches), à son alignement (traits pour les axes, croix pour le centrage), à son soudage (zone à encoller pour fixer un CMS), son refroidissement (figurer le volume d’un radiateur pour les boîtiers TO220), ...
- On peut aussi prévoir une double implantation : par exemple ajouter des pads à l’empreinte d’une résistance ou d’une inductance pour implantation verticale ou horizontale. Parfois la forme du composant peut légèrement varier selon les approvisionnements : il faut pouvoir facilement substituer un composant avec un pas différent (exemple : condensateur au pas de 2,54 au lieu de 5,08 mm, ou diamètre différent) ;
- Edition des contours : le composant a plusieurs contours de couleurs différentes, mais une seule (la couleur bleue) se retrouvera au final sur la couche F.Silks (sérigraphie des composants, imprimée en blanc sur le vernis de couleur de la carte). On y met le dessin classique : forme ou contour (+symbole à l’intérieur), référence, et éventuellement valeur ;
- Vérifications : Pensez à associer l’empreinte créée au composant du schéma dans Cypcb, enregistrer le schéma / le projet pour ne pas perdre les modifications. Je conseille fortement d’imprimer l’empreinte que vous venez de dessiner, et d’y superposer le composant réel pour voir si ça colle. C’est en voyant l’empreinte imprimée que les oublis sautent aux yeux :
 - tailles de pads suffisantes pour souder ;
 - comment repérer l’orientation / la broche 1 ;
 - est-ce que la référence apparaît et est-elle lisible (étiquette REF**, son emplacement, son orientation) ;
 - où dois-je percer pour fixer (vis).