

Formation CAO électronique RTS 2020

Le but de cette formation est de fournir les bases de la conception de circuit imprimé aux nouveaux membres de RTS. Nous allons donc voir comment créer une petite carte mère permettant de piloter un servo moteur avec un potentiomètre.

Logiciel nécessaire :

- KiCad téléchargeable gratuitement ici : <https://kicad-pcb.org/>

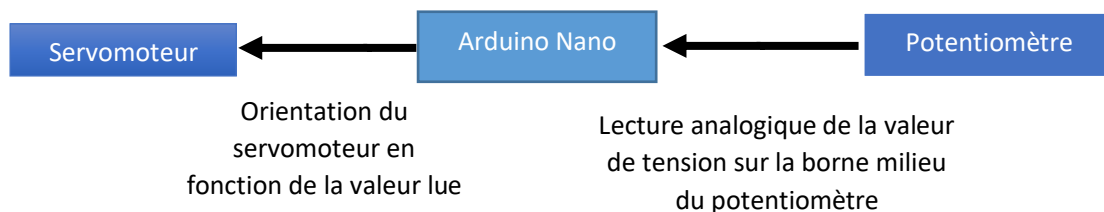
Matériel nécessaire :

- Pour prototyper, puis au moment de la soudure :
 - o 1x Arduino Nano
 - o 1x Potentiomètre
 - o 1x Servomoteur

Cahier des charges

Le cahier des charges est simple : le circuit assemblé doit permettre de piloter un servo-moteur avec un potentiomètre. Au final on veut juste éviter d'avoir à mettre en place une breadboard 😊.

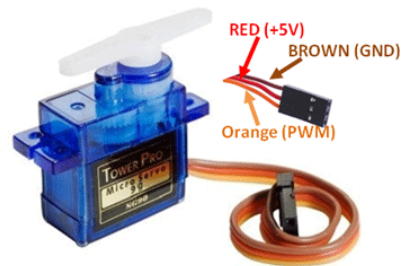
Le principe de fonctionnement est décrit ici :



Servomoteur

On ne peut pas directement souder un servomoteur à notre circuit imprimé, il va donc falloir placer un connecteur pour brancher facilement le moteur.

Comme décrit sur l'image, le connecteur à placer sera un connecteur 01x03 mâle, donc les signaux devront être dans un ordre bien précis.



Source : <https://components101.com/servo-motor-basics-pinout-datasheet>

Potentiomètre

Le principe de fonctionnement d'un potentiomètre est plutôt simple. Pour un potentiomètre d'une valeur donnée (disons par exemple 10kΩ), il s'agit en réalité de deux résistances en série dont la somme des valeurs est toujours égale à 10kΩ.

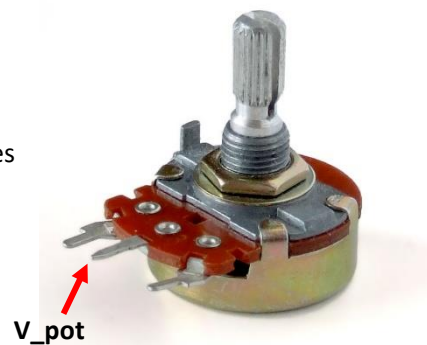
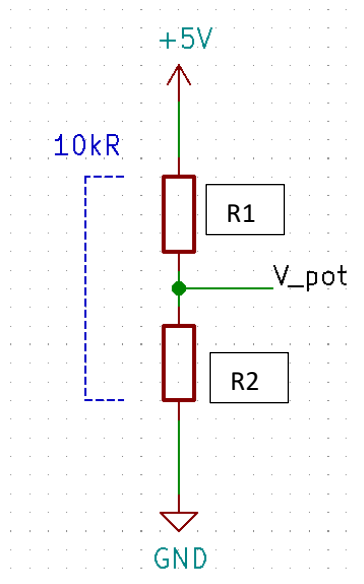


Schéma équivalent :



Un potentiomètre a 3 broches. On impose une tension fixe aux bornes les plus extérieures, et on vient lire la tension entre les deux résistances (V_{pot} sur le schéma).

Par formule du pont diviseur de tension, V_{pot} dépend de la valeur de la résistance $R2$, qui varie quand on tourne le potentiomètre :

$$V_{pot} = \frac{R2}{R1+R2} \cdot 5V = \frac{5 \cdot R2}{10 \cdot 10^3}$$

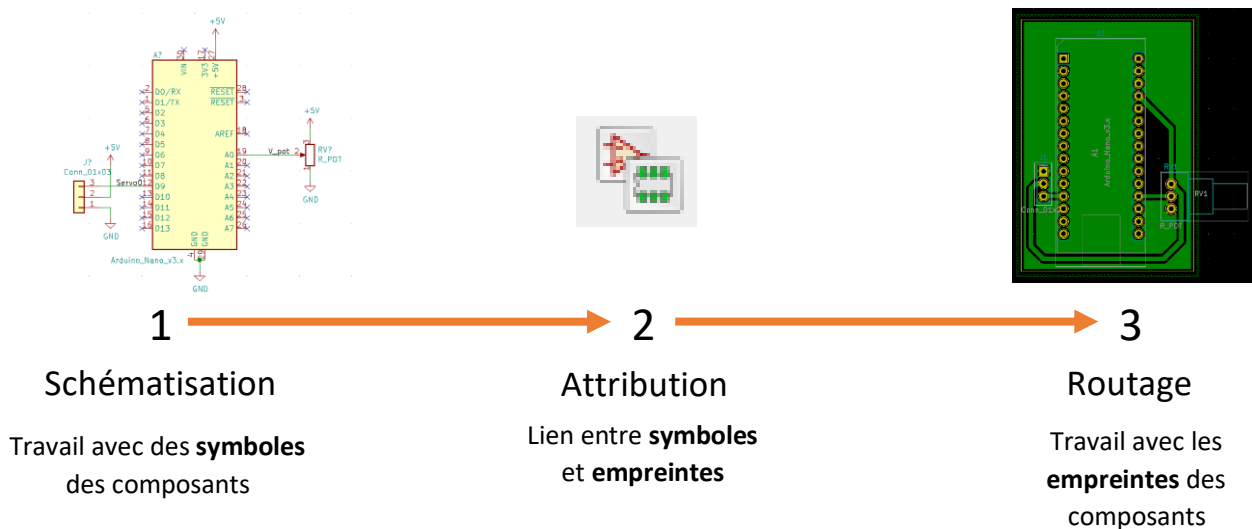
en posant toujours $(R1 + R2) = 10k\Omega$.

La tension V_{pot} sera lue sur une broche analogique de l'Arduino.

KiCad 0.1 : Comment ça fonctionne ?

La création d'un circuit imprimé sur KiCad suit une procédure que l'on rencontre dans la quasi-totalité des logiciels de conception électronique. On va la résumer ici pour vous donner une vision d'ensemble avant d'entrer dans le détail de chacune des étapes.

Un schéma pour résumer ça :



Les deux principales étapes sont la **schématisation**, puis le **routage**.

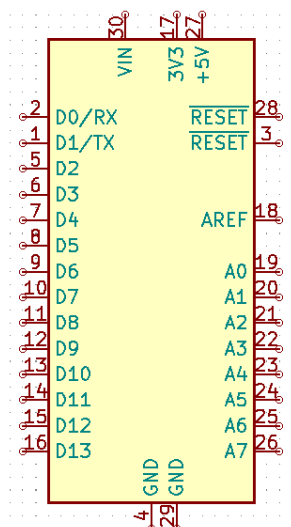
Lors de la schématisation, on travaille avec des **symboles** des composants, et on les relie entre eux pour avoir le circuit que l'on souhaite.

On passe ensuite à l'attribution, où l'on choisit l'**empreinte** attribuée à chaque composant. L'empreinte correspond à la forme qu'aura le composant dans la vie réelle.

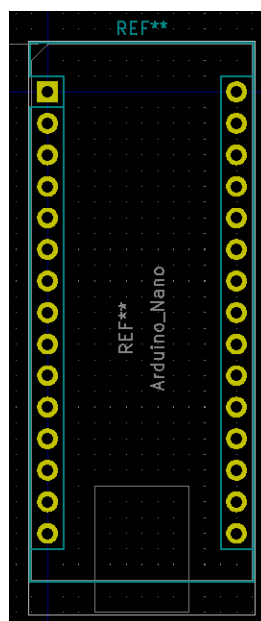
Symbole : Représentation schématique d'un composant

Empreinte : Représentation physique d'un composant

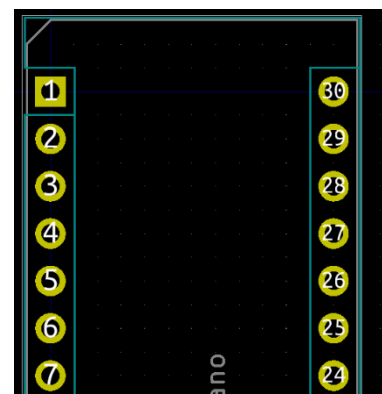
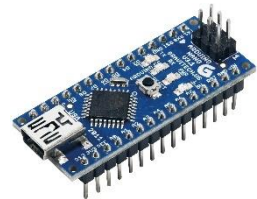
Un exemple avec l'Arduino Nano :



Symbole de l'Arduino Nano



Empreinte de l'Arduino Nano



Zoom sur l'empreinte

Contrairement au symbole de l'Arduino, son empreinte (espacement entre les pattes, forme, etc...) correspond à 100% au modèle sur la photo !

Important pour plus tard : La correspondance entre un symbole et son empreinte est faite par la numérotation des *pins* ! (cf *Zoom sur l'empreinte*)

On reverra ce point au moment de créer un symbole et une empreinte pour nos propres composants.

Commençons à entrer dans le vif du sujet !

Kicad 0.2 : Création du projet

Après avoir ouvert le logiciel, on tombe sur la vue globale du projet. Si c'est la première fois que vous utilisez le logiciel, il faut créer un nouveau projet :

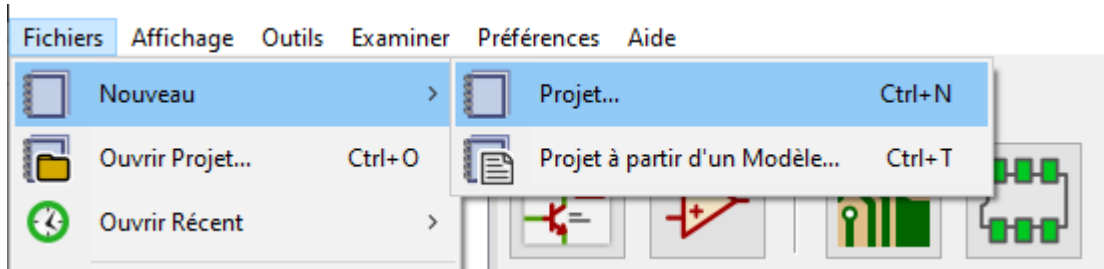


Figure 1 : Création du projet

Apparaît alors à l'écran deux nouveaux fichiers sur le panneau de droite :

Formation.sch : le fichier dans lequel on dessine le schéma du circuit

Formation.kicad_pcb : le fichier dans lequel on fait le routage du circuit

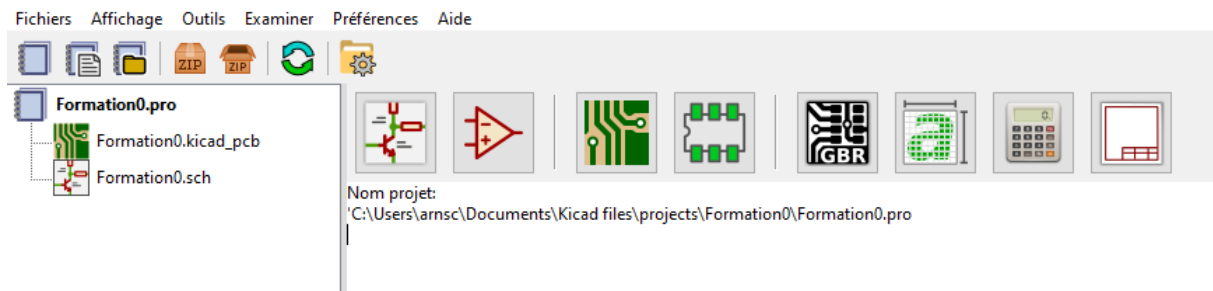


Figure 2 : Fenêtre de gestion de projet

Le panneau central abrite 8 boutons qui prendront leur importance surtout au moment de créer notre propre bibliothèque de composants. Pour le moment, on passe au schéma !

Kicad 0.3 : Schématisation

Pour débiter cette partie, on ouvre le fichier *Formation.sch* en double-cliquant dessus.

S'ouvre alors une grande fenêtre vide avec beaucoup de boutons.

Vous trouverez dans ce tableau tous les raccourcis claviers qui sont utiles pour la schématisation :

Fonction	Raccourci	Description
Connexion/Câblage	W	Permet de dessiner les nœuds de connexions entre les composants
Label*	L	Permet de donner des noms aux nœuds de connexion
Symboles	A	Ouvre le menu permettant de choisir et placer un symbole
Symboles de puissance	P	Ouvre le menu permettant de placer des <i>labels</i> spécifiques aux signaux de puissance (5V, GND, etc...)
Lignes de dessin	I	Permet de tracer des lignes graphiques. Utile pour démarquer des éléments
Déplacer	M	Permet de déplacer un composant. Utilisation : placer son curseur sur le composant, puis appuyer sur M
Attraper	G	Semblable à la fonction Déplacer, mais ne détache pas le composant des nœuds qui lui sont attachés (Testez, c'est difficile à décrire :D)
Marqueur <i>non-connecté</i>	Q	Permet de désigner les pins d'un composant qui ne sont connectés à rien
Copier	C	Permet de copier un composant. Utilisation : placer son curseur sur le composant, puis appuyer sur C

* : deux nœuds de connexions qui ont le même nom sont reliés, même si aucune connexion visible ne relie les deux. C'est très utile lorsque le schéma comme à devenir un sac de nœuds.

Tip : Pour faciliter la lecture et la compréhension du schéma, c'est une bonne idée de donner un nom à un maximum de nœuds.

Placement des composants

Pour commencer, on va placer tous les composants nécessaires :

- Arduino Nano :
 - Appuyer sur A pour ouvrir le menu des symboles
 - Dans le cadre « *filtre* », indiquer « *arduino* »
 - Sélectionner « *Arduino_Nano_v.3.x* » (double-cliquer)
 - Placer ensuite le composant sur le schéma
- Potentiomètre
 - KiCad inclut nativement un grand catalogue de potentiomètres. On va se contenter de placer un symbole générique : chercher « *R_POT* ».
- Servomoteur
 - On ne peut pas souder le servomoteur directement à la carte, on va donc placer un connecteur compatible ([voir illustration servo moteur page 1](#))
 - Ouvrir le menu des symboles
 - Pour les connecteurs génériques, chercher « *conn_0nx0m* » ou n et m sont les nombres de broches en ligne et en colonne. Dans notre cas : « *conn_01x03* ». On a alors le choix ([voir capture ci-dessous](#)) :
 - Un connecteur générique, ni mâle ni femelle
 - Un connecteur mâle
 - Un connecteur femelle

Cela n'a techniquement aucune importance autre que lors du rendu 3D de notre PCB. Néanmoins, il faut toujours être le plus précis possible : on sait qu'on va utiliser un connecteur mâle pour notre servo, on prend donc « Conn_01x03_Male ».

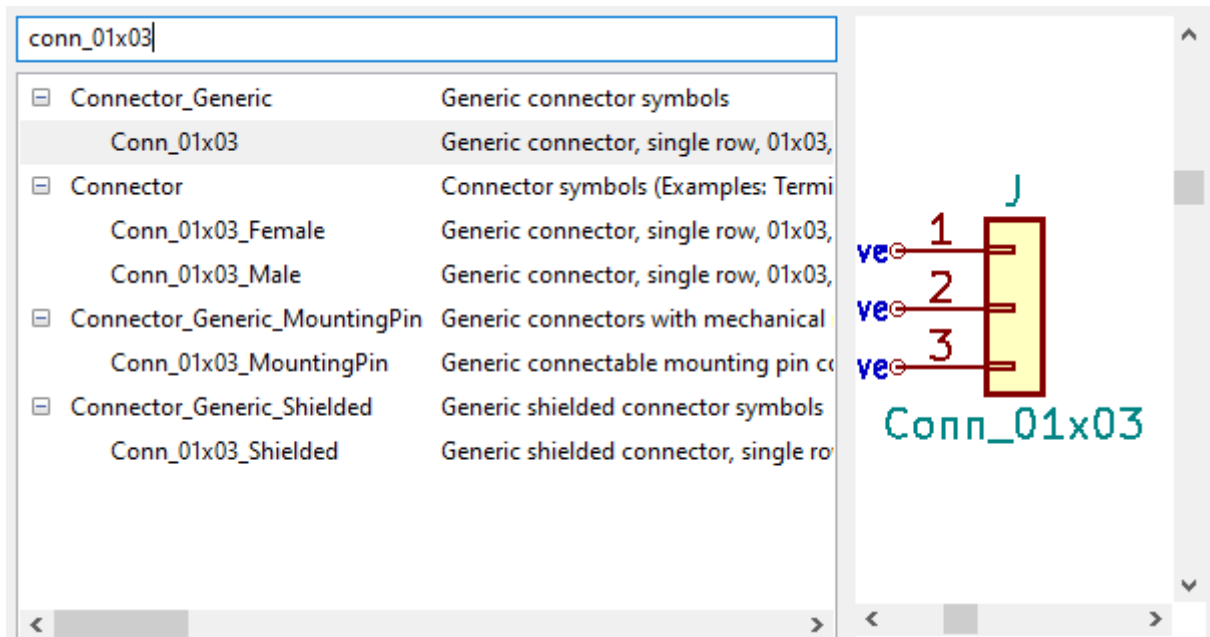
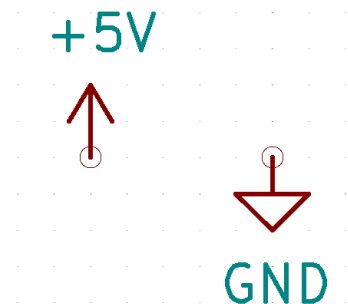


Figure 3: Résultat de la recherche de connecteurs

Arrivé à ce stade, tous les composants dont on a besoin sont sur le schéma, plus qu'à les relier !

Connexion des composants

- Arduino :
 - o Ce composant fait office de générateur de tension dans notre cas. Il faut donc relier :
 - Toutes les bornes *GND* à un label « GND »
 - La borne 5V à un label « 5V »
 Ceci permettra de facilement alimenter les autres composants.
 On peut trouver les symboles spéciaux en appuyant sur P*.
 Il faut ensuite chercher « GND » et « 5V ».



Capture des symboles +5V et GND

* : les symboles du menu P (pour Power) permettent de représenter clairement les signaux d'alimentation d'un circuit. *Ils fonctionnent comme des labels* : deux nœuds visuellement séparés mais attachés à un même symbole de puissance sont en fait reliés.

- Potentiomètre :
 - o Comme expliqué précédemment, la borne du milieu est à relier à une entrée analogique de votre choix
 - o Les deux autres sont à relier à la masse (GND) et au 5V, avec les symboles de puissance attribués précédemment.
- Servomoteur :
 - o Il faut respecter l'ordre de branchement indiqué sur la figure page 1, sinon ça ne va pas fonctionner
 Rouge : +5V ; Noir : GND ; Jaune : Signal de commande.

Le signal de commande peut être relié à quasi toutes les entrées/sorties digitales de l'Arduino. On va cependant éviter les bornes D0 et D1 qui sont liées à la liaison série entre l'Arduino et l'USB.

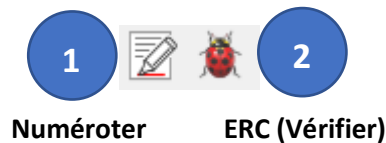
Note : Le mieux est de rester sur les branchements testés pendant votre prototypage. Rien ne vaut de tester un montage avant d'en faire un circuit imprimé, ça réduit le risque de tout recommencer ! 😊

Normalement à ce stade, tous les composants sont branchés !

Fin du schéma

Il reste encore quelques petits détails :

1. Marquer toutes les bornes de l'Arduino qui ne sont pas connectées avec le symbole « Not Connected »
 - **Comment :** On utilise la touche **Q** en pointant là où on veut placer le marqueur
 - **Pourquoi :** Cela permet au logiciel de vérifier s'il n'y a pas eu d'erreurs d'inattention ou d'oubli de la part du concepteur.



- 1) Numérotage le schéma
 - On utilise le bouton « Numérotage » situé en-haut de l'écran (voir illustration ci-dessus)
 - Une fenêtre s'ouvre alors : cliquer sur « Numérotation » en bas à droite de la fenêtre, puis fermer
 - **Pourquoi :** Numérotage le schéma revient à donner un nom de variable à chaque composant, qui permettra au logiciel de les inscrire dans la *netlist* qui fera la transition entre le schéma et le circuit imprimé
- 2) Vérifier le schéma
 - On va effectuer une ERC (Electrical Rule Check) : une vérification automatique du schéma pour voir si des erreurs fondamentales de design ont été commises. Cet outil ne fait pas de miracle, mais peut éviter des erreurs bêtes.
 - On utilise le bouton « ERC » situé également en-haut de l'écran (voir ci-dessus).
 - Sur la fenêtre ouverte : cliquer sur « Exécuter »
 - S'affiche alors au moins une erreur :

Pin connectée à d'autres pins, mais aucune pin pour la piloter
[@ \(142.24 mm, 130.81 mm\)](#): Pin 29 (Power input) du composant A1 non pilotée (Net 14).

Figure 4: Message d'erreur

- Les messages d'erreur sont rarement clairs au premier abord. Ici, le logiciel attend qu'une entrée de puissance (power input) soit reliée à un signal de puissance.

- **Pour régler le problème** : relier un symbole GND à un symbole PWR_FLAG via le menu P, comme dans l'exemple ci-dessous. Cela permet d'indiquer au logiciel que GND est un signal de puissance.

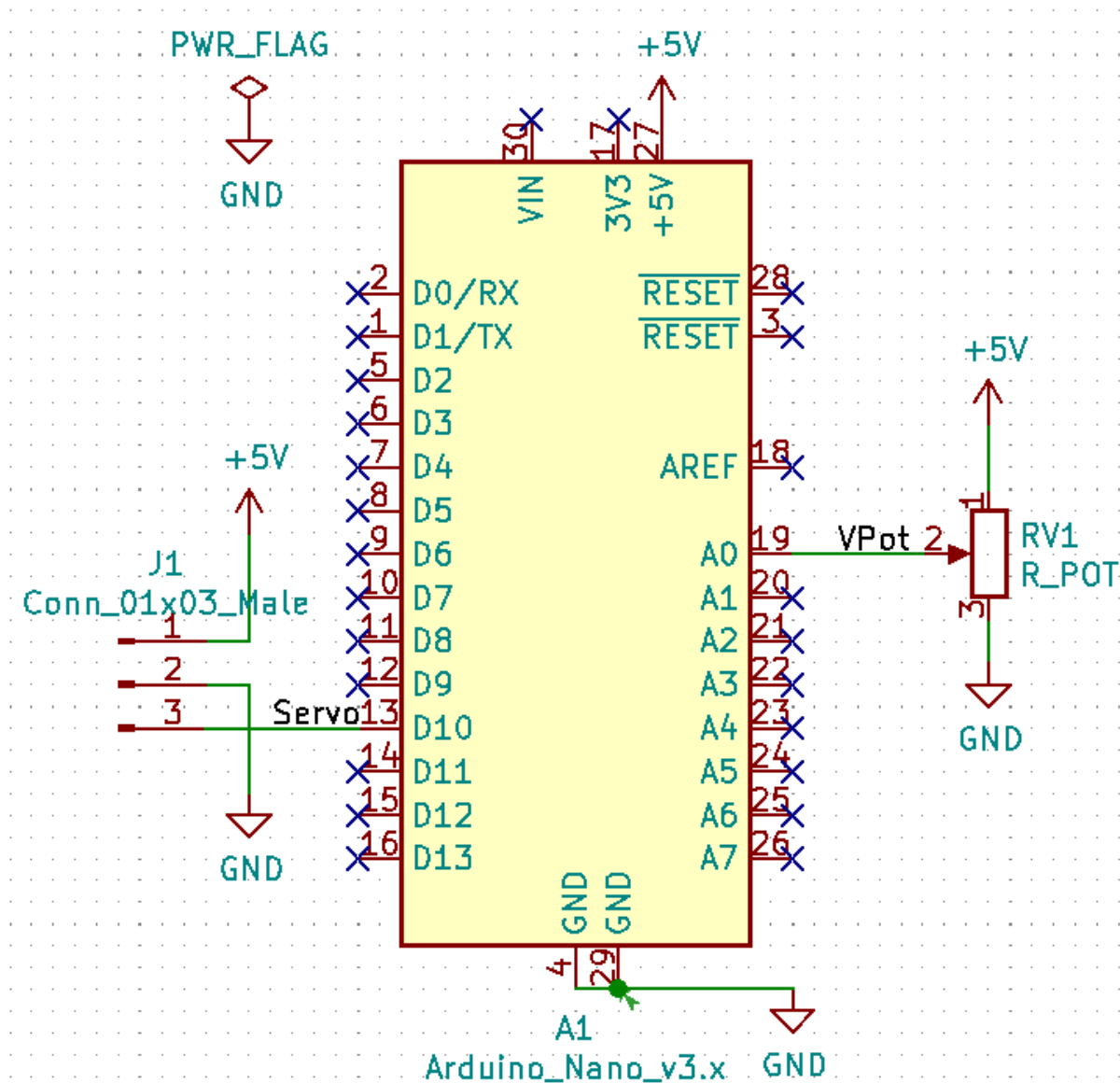


Figure 5: Exemple de schéma

Entamons maintenant la deuxième étape : l'attribution !

Kicad 0.4 : Attribution

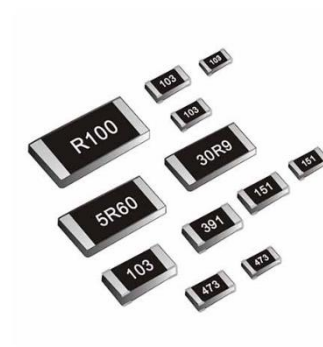
Cette étape permet au concepteur de choisir la forme que va prendre chaque composant, qui sont pour le moment purement abstraits. Car souvent, pour un seul composant, il existe une multitude de formes, ou *package*.

Exemple avec les résistances :

Différentes tailles et formes (ne dépendant pas de la valeur)

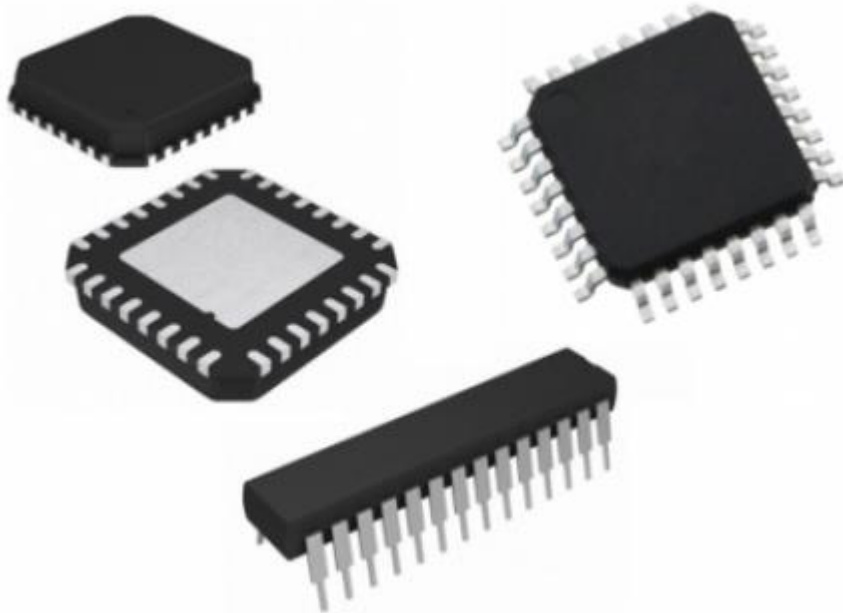


Résistances THT (traversantes)



Résistances SMD (montées en surface)

Avec l'Atmega328p, microcontrôleur de l'Arduino Nano :



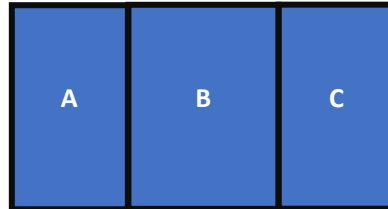
Différents *packages* de l'Atmega328p

Il existe pour chaque package une empreinte qui lui est propre. Notre travail ici va être de choisir les empreintes de nos composants pour le circuit imprimé. Les choses sont ici simplifiées au maximum puisqu'il s'agit d'une simple initiation 😊.

Rendez-vous dans la fenêtre d'attribution des empreintes :



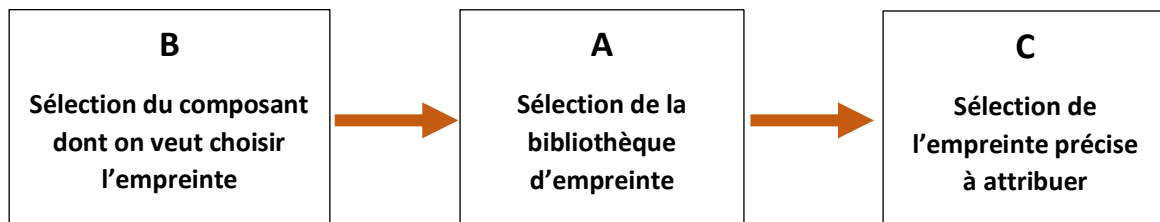
- Le logiciel charge ensuite toutes les empreintes enregistrées, cela peut prendre un petit moment :/
- La fenêtre qui s'ouvre est divisée en 3 parties :



Description des 3 zones :

- A : Bibliothèques d'empreintes. Il s'agit de dossiers contenant une famille d'empreintes. On clique dessus pour en sélectionner une
- B : Zone d'attribution, c'est ici que sont nos composants. On les sélectionne en cliquant dessus
- C : Les empreintes contenues dans la bibliothèque sélectionnée.

L'utilisation de ces zones suit toujours le même schéma :



- L'empreinte de l'Arduino étant déjà attribuée (c'est parfois le cas), on ne va pas y toucher
- Attribution du connecteur
 - Cliquer sur RV1 dans la zone **B**
 - Chercher et sélectionner « Potentiometer_THT » dans la zone **A**
 - Chercher et double-cliquer sur « Potentiometer_THT :Potentiometer_Piher_PC-16_Single_Horizontal » dans la zone **C**
- Attribution du potentiomètre
 - Cliquer sur J1 dans la zone **B**
 - Chercher et sélectionner « Connector_PinHeader_2.54mm » dans la zone **A**
 - Chercher et double-cliquer sur « PinHeader_1x03_P2.54mm_Vertical » dans la zone **C**
- Cliquer sur OK pour fermer la fenêtre

Il ne nous reste plus qu'à générer la **netlist** qui sera utilisée pour débiter le dessin du circuit imprimé :



- Cliquer sur le bouton indiqué ci-dessus pour ouvrir la fenêtre de génération de la netlist
- Appuyer sur le bouton « Génération de la Netliste »
- Enregistrer directement dans l'emplacement proposé

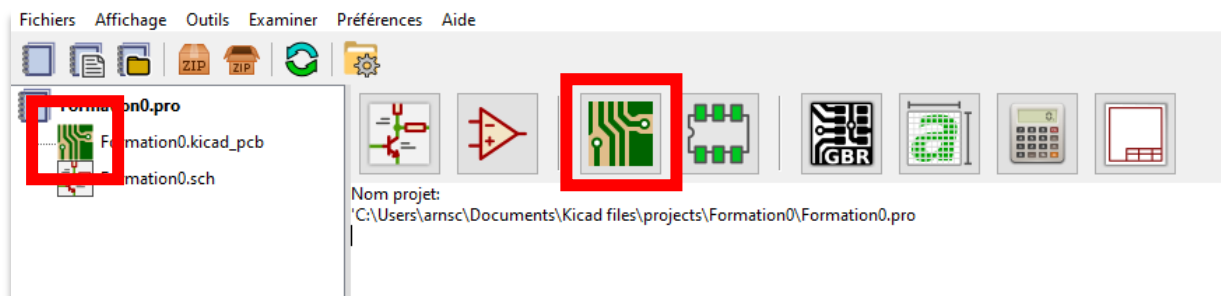
Kicad 0.5 : Dessin du PCB

Maintenant que le schéma et l'attribution sont terminés, et la netlist créée, on peut passer au dessin du PCB.

Pour ouvrir la fenêtre de PCB depuis la fenêtre de schéma, on peut utiliser le bouton ci-dessous :



On peut également l'ouvrir depuis la fenêtre de gestion du projet :



La fenêtre s'ouvre alors :

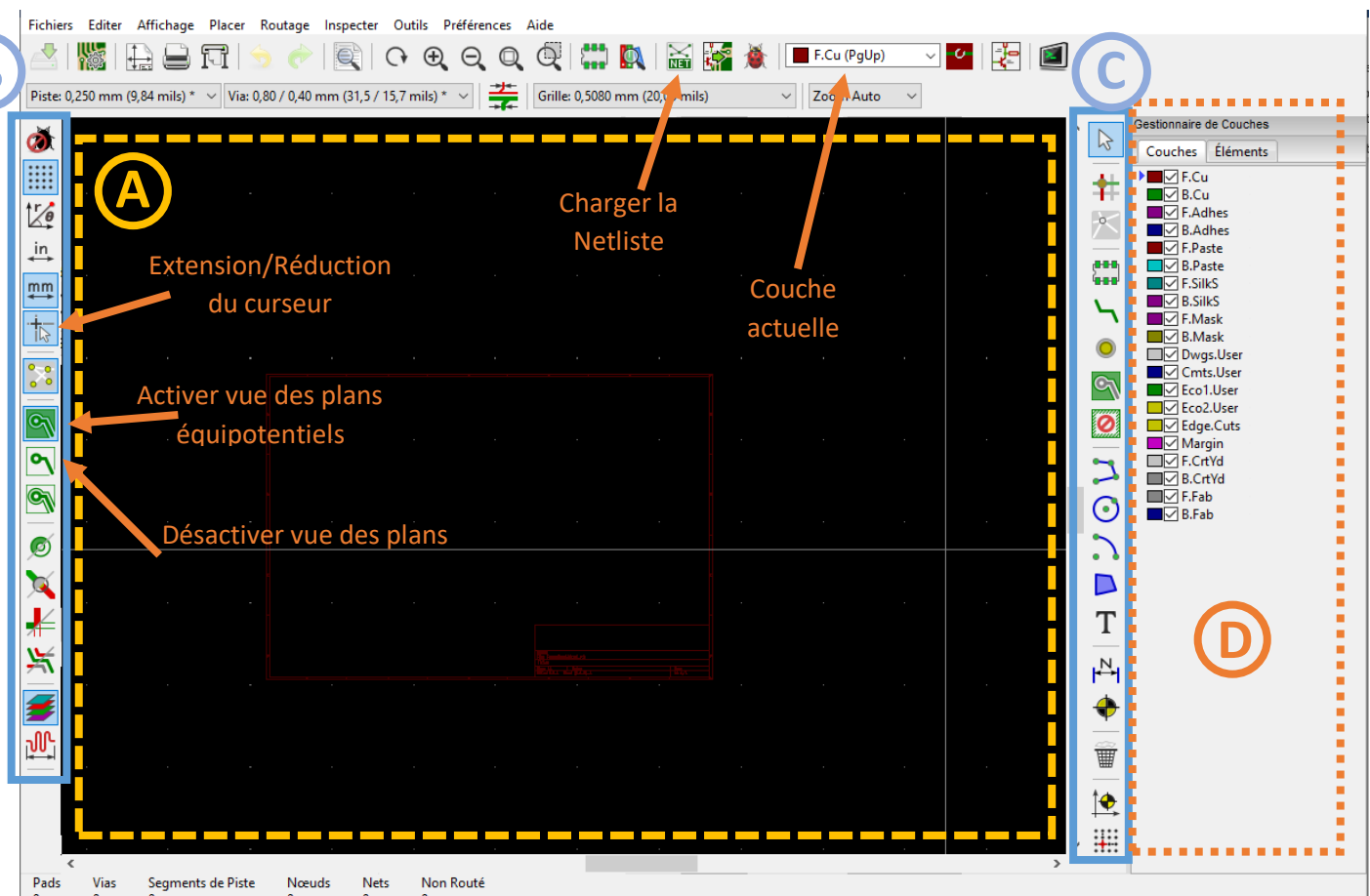


Figure 6: Vue de la fenêtre PCB

Zone	Description
A	Plan de travail. C'est ici qu'on manipule les composants et dessine le PCB
B	Outils visuels. Tous les boutons ici permettent de modifier l'aspect du PCB pour aider le concepteur
C	Outils de dessin. On y trouve tous ce qui est nécessaire pour dessiner notre circuit
D	Gestionnaire de couches. On peut sélectionner les couches visibles sur le plan de travail

Notions élémentaires de PCB :

Couches ou layer : Un PCB est composé de différentes couches.

Une couche de cuivre permet de dessiner des pistes de cuivre pour connecter des composants. Certains PCB sont pourvus d'une seule couche de cuivre, mais la plupart de deux couches de cuivre : le dessus et le dessous (top- et bottom -layer).

On distingue également des couches non cuivrées :

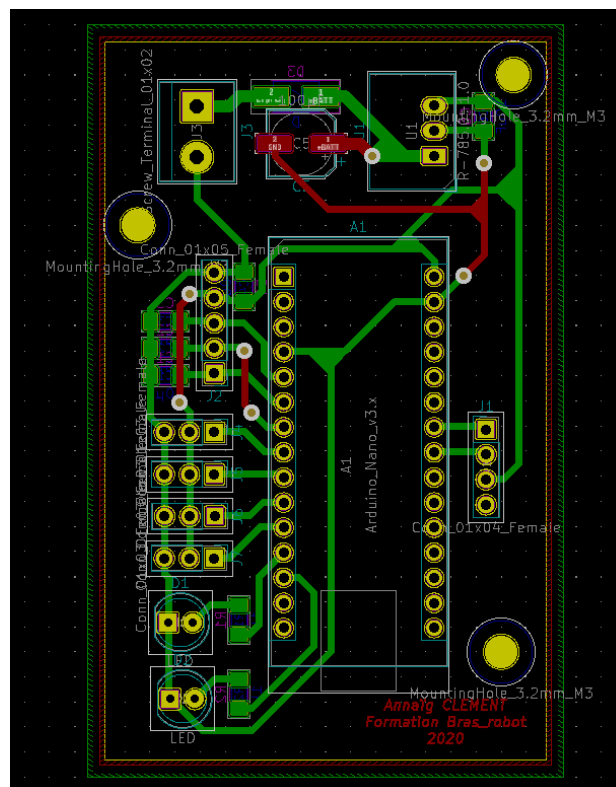
Silkscreen layer ou couche sérigraphie : permet de dessiner (en blanc, la plupart du temps) des inscriptions sur le circuit

Edge Cuts layer ou couche de découpe : permet de délimiter la forme finale du PCB

Désignation des couches sur Kicad :

Nom	Désignation Kicad	Couleur
Bottom Layer	B.Cu	Vert
Top Layer	F.Cu	Rouge
Couche découpe	Edge.Cuts	Jaune foncé
Bottom Silkscreen	B.Silk	Mauve
Top Silkscreen	F.Silk	Cyan

Exemple de PCB avec couches visibles :



Traces ou pistes : lignes de cuivre dessinées par le concepteur, pour relier deux points du circuit

Vias : trous dans le PCB permettant de passer une trace sur une autre couche du PCB, typiquement de la top-layer à la bottom-layer ou inversement

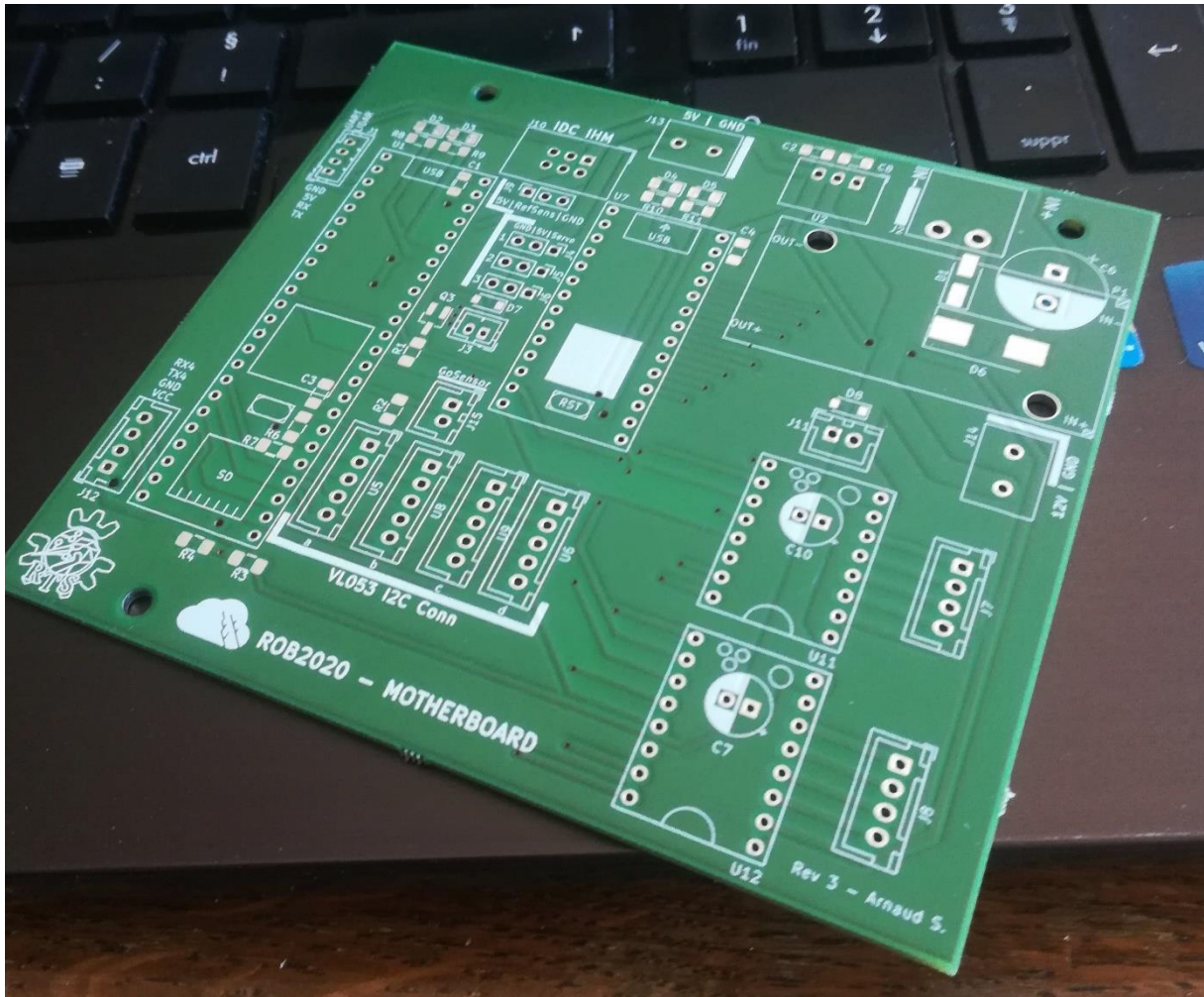


Figure 7: Carte-mère du robot CDR2020

On retrouve tous les éléments sur cette photo :

- En blanc la sérigraphie
- Les traces de cuivre qui relient les composants
- Les vias qui relient les deux côtés du PCB, sous forme de petits trous

Mise en place et paramétrage

Avant de commencer à travailler sur le PCB, il est nécessaire de suivre les étapes suivantes.

1) Importation du schéma

Le plan de travail est vide, il faut importer les données du schéma sur le PCB :

- Appuyer sur le bouton « Charger la netliste » (voir ci-contre et **Figure 6**)
- La fenêtre s'ouvre
- Devant la ligne « Fichier Netliste », cliquer sur l'icône dossier
- Sélectionner le fichier .net dans le dossier
- Cliquer sur « Mise à jour PCB »
- Fermer la fenêtre



Les composants sont maintenant attachés à la souris. Les placer vers le centre du plan de travail, puis cliquer-gauche, et Echap pour les désélectionner.

Voici ce qui devrait être affiché à l'écran :

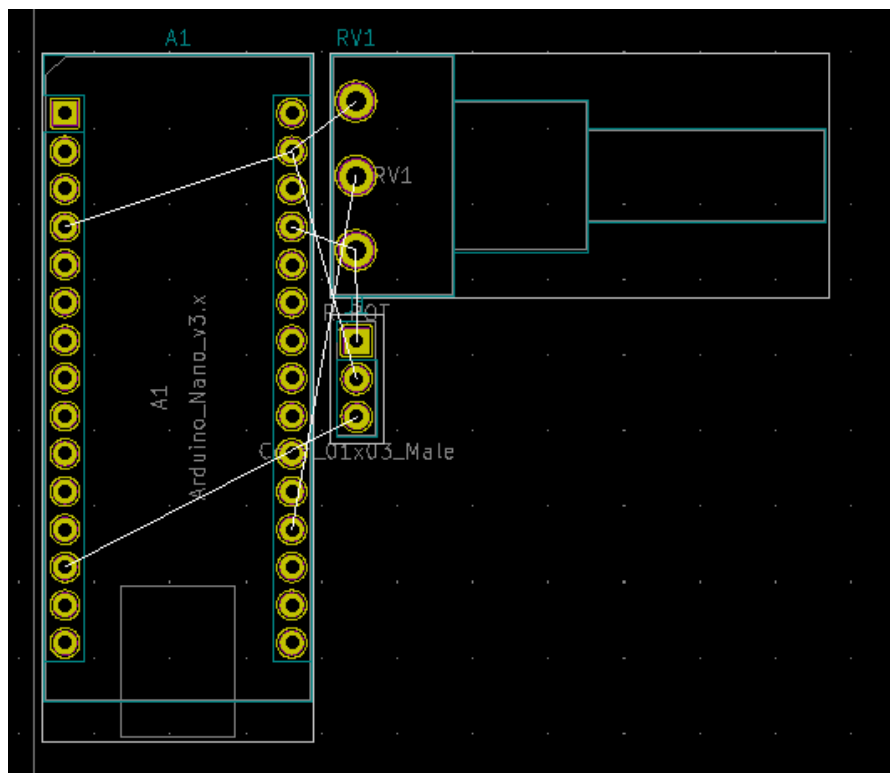


Figure 8: Composants placés

Remarque : On voit bien les empreintes sélectionnées pendant l'attribution !

Les différentes *pins* sont reliées entre-elles par un trait fin, *le chevelu*, permettant d'indiquer au concepteur ce qu'il faut relier ensemble.

Important : Le logiciel vous empêche de relier sur le PCB deux pins qui ne sont pas connectées sur le schéma. C'est toute l'utilité de schématiser le circuit avant de dessiner le PCB : ***fixer les règles de connexion***

Le concepteur peut alors se concentrer sur des aspects propres au routage.

Cela signifie aussi qu'il faut ne pas se planter sur le schéma ! :D

2) Configuration des largeurs de traces et diamètres de via

La largeur des traces de cuivre que nous allons placer sur le PCB est très importante pour les raisons suivantes :

- Risques d'échauffement
 - Un courant à travers un conducteur entraîne l'échauffement de ce dernier. Si le courant est trop important, la trace de cuivre va chauffer, se rompre et/ou endommager le PCB.

- Capacité du processus de fabrication
 - o Chaque processus de fabrication de PCB a ses limites en termes de précision et de fiabilité. Avec la micro-graveuse de TPS par exemple, il est risqué de graver des traces de moins de 0.5mm d'épaisseur, alors que certaines entreprises de PCB permettent d'avoir des traces de 0.2mm.

Ceci s'applique également pour la taille des vias.

Pour configurer tout ceci, cliquer sur le menu « Editer tailles pré-définies » comme indiqué ci-contre :

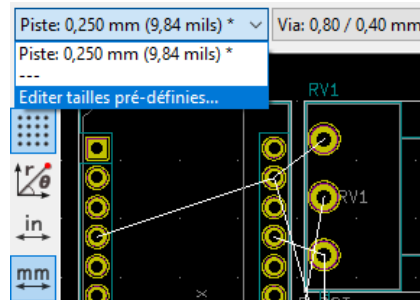


Figure 9 : Accès à la configuration

Une fenêtre s'ouvre alors :

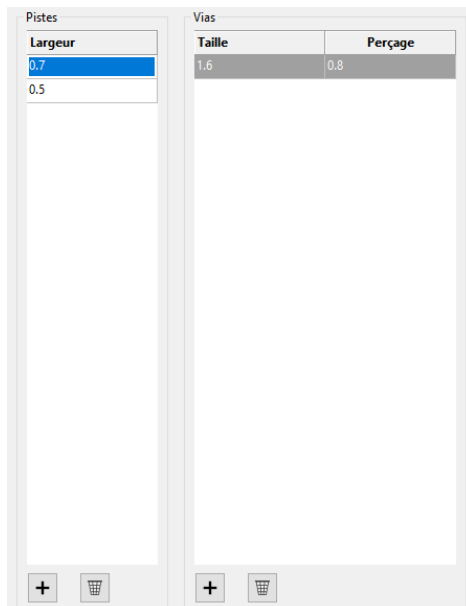


Figure 10 : Configuration de la largeur de trace et de la taille de via

On clique sur le bouton « + » pour ajouter une ligne, qu'il faut ensuite modifier avec la valeur souhaitée.

Les vias ont deux grandeurs à indiquer : la taille et le perçage (le trou).

Entrez les valeurs 0,7mm et 0,5mm pour la largeur des traces

1,6mm/0,8mm pour les vias

Pour terminer, appuyer sur **OK**

Note : Pour faire les vias au fablab, le diamètre du trou doit être de 0.8mm, et le diamètre total de 1.6mm

Il suffit ensuite de sélectionner la largeur et la taille de vias qu'on souhaite utiliser pour les prochains tracés (attention, c'est pas automatique !):

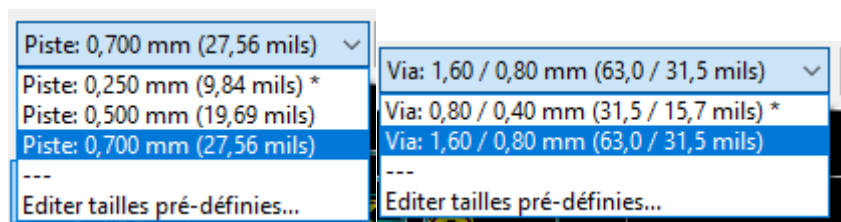


Figure 11 : Sélection de la largeur de trace et du diamètre de via

Conseil : Si vous comptez utiliser la micrograveuse de l'école, il faut faire attention à avoir des traces suffisamment épaisses. J'utilise quasi-toujours du 0.7mm pour des traces qui ne sont pas traversées par un courant important. 1mm, voir plus pour les pistes d'alimentation, qui sont traversées par plus de courant.

Pour choisir la bonne largeur, Kicad est équipé d'un outil de calcul (logo de calculatrice sur la fenêtre de gestion de projet)

Les raccourcis importants :

Fonction	Raccourci	Description
Lâcher outil	Echap	Permet de désélectionner un outil
Outil de tracer	X	Outil de base pour tracer des pistes de cuivre
Placement via	V	Lorsque qu'on trace une piste, appuyer sur V pour changer de couche et placer un via
Retourner empreinte	F	Permet de basculer l'empreinte pointée sur l'autre couche de cuivre
Changement de couche	V	Lorsqu'on n'utilise pas l'outil de tracer, permet de basculer de la F.Cu à la B.Cu
Déplacer empreinte	M	Lorsqu'aucun outil ou objet n'est sélectionné, pointer l'empreinte et appuyer sur M pour la déplacer
Régénération des plans	B	Permet de retracer automatiquement les plans équipotentiels (plan de masse notamment). On verra ça à la fin du tracé 😊

Routage du PCB

On peut maintenant commencer !

1) Démêler le chevelu

Déplacer les composants pour obtenir une disposition plus simple à tracer. On s'aide du chevelu anticiper les chemins qu'on va tracer.

Utiliser la touche M sur les empreintes pour les déplacer

Voir ci-contre un exemple de placement :

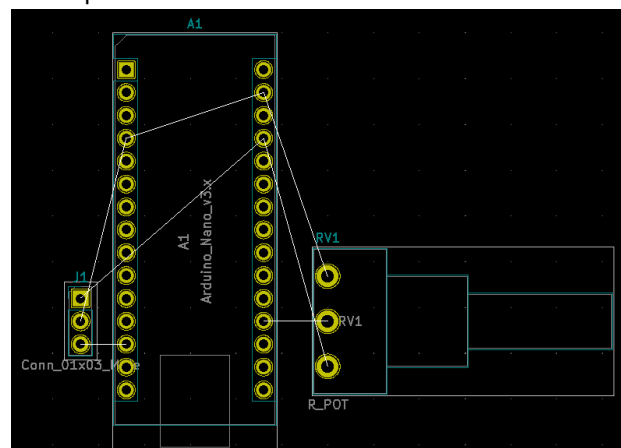


Figure 12 : Exemple de placement

2) Tracer

- **Une règle importante** : surtout pas de traces à 90° ! Surtout important en signaux haute-fréquence, mais les lignes à 45° sont aussi beaucoup plus esthétiques 😊
- Sélectionner la bottom layer (couche inférieure), en utilisant la touche V pour basculer de la top layer à la bottom layer, et inversement. Vérifier que la couche B.Cu est bien sélectionnée (voir figure 6).
- Appuyer sur X pour sélectionner l'outil de tracer.
- Cliquer sur un des pad accroché à un chevelu. La trace de cuivre suit la souris (Normalement verte si vous avez sélectionné la bonne couche).
- Tracer le chemin vers le ou les pads indiqués par le chevelu, en cliquant régulièrement pour créer des points de passage

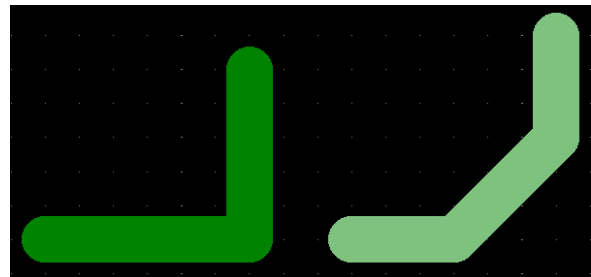


Figure 13 : 90° vs 45°

Votre objectif est de relier tout ce qui est indiqué par le chevelu sur la **bottom layer**.

Pour vous entraîner, vous pouvez aussi utiliser la top layer et/ou des vias.

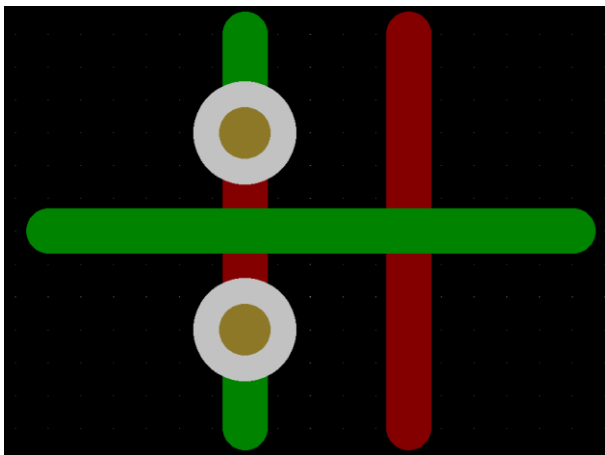


Figure 14 : Exemples de chevauchement

Puisque nous avons ici uniquement des composants traversants, l'utilisation des vias et de la top layer permet principalement d'effectuer des chevauchements, comme illustré ci-contre.

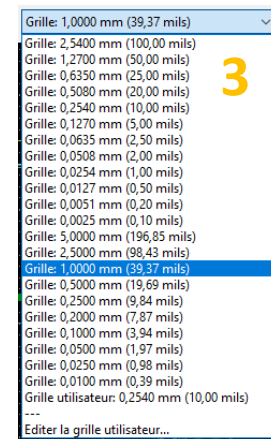
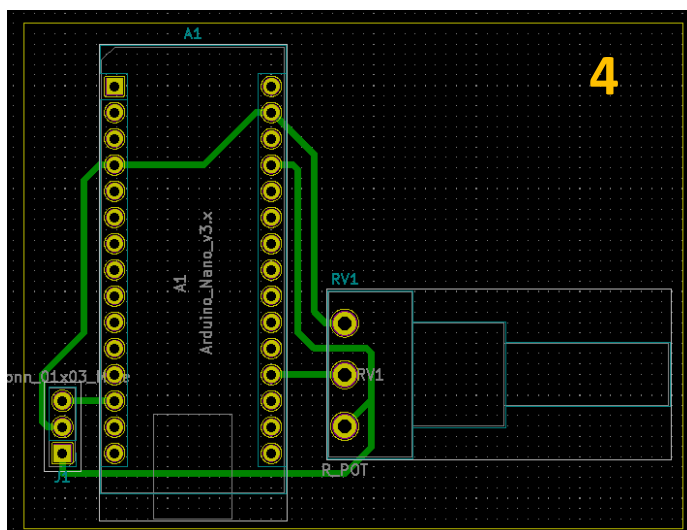
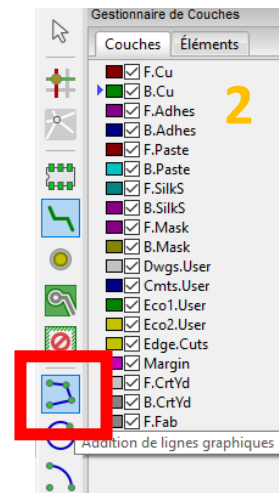
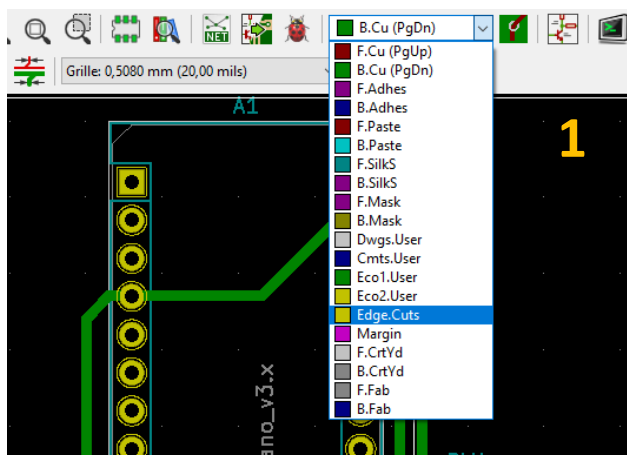
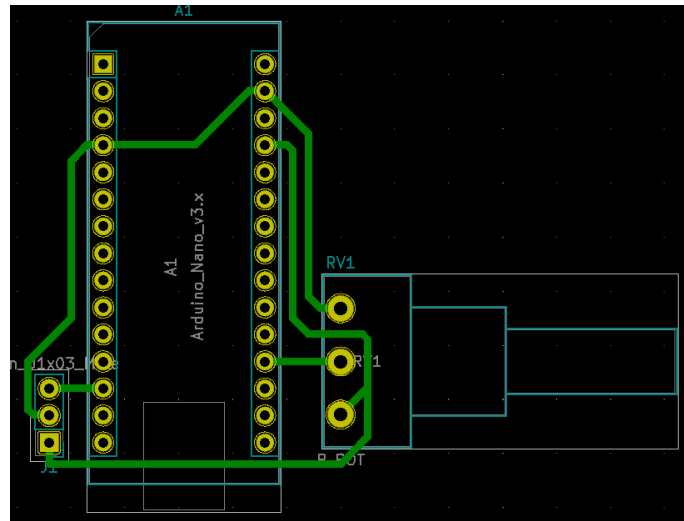
Ci-contre un exemple de réalisation avec tous les tracés sur la bottom-layer

3) Délimiter le PCB

Pour la suite, il est conseillé d'étendre votre curseur avec le bouton indiqué en figure 6, ce qui facilitera les délimitations.

Maintenant que le routage est terminé, il faut dessiner la forme finale du PCB. Voici la marche à suivre :

- Sélectionner la couche Edge.Cuts
- Sélectionner l'outil de lignes graphiques
- Optionnel : Modifier la taille de la grille pour faciliter le tracer
- Tracer la forme finale du PCB



4) Placer le plan de masse

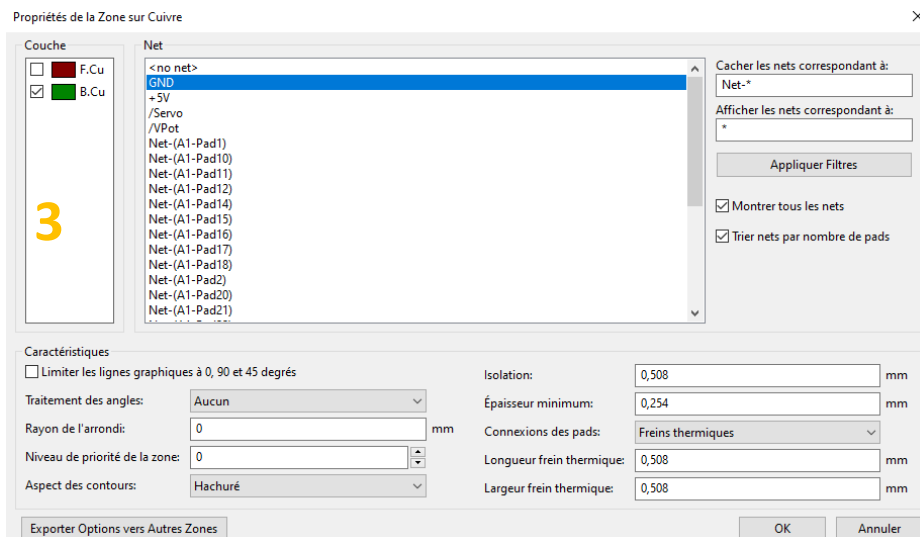
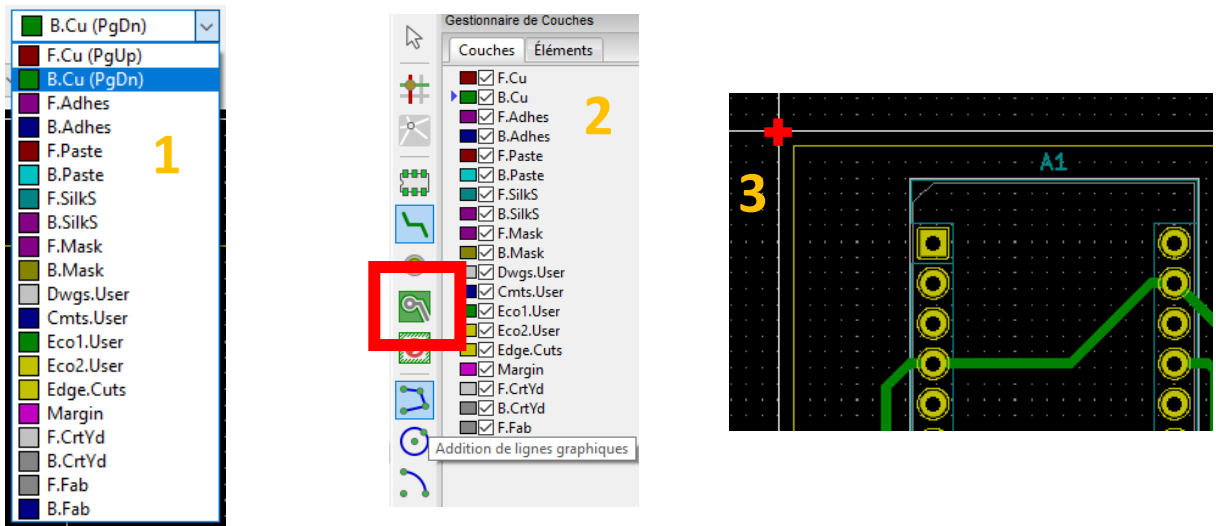
Plan d'équipotentiel : au lieu d'une trace de cuivre, c'est tout une zone qui est relié à un signal particulier.

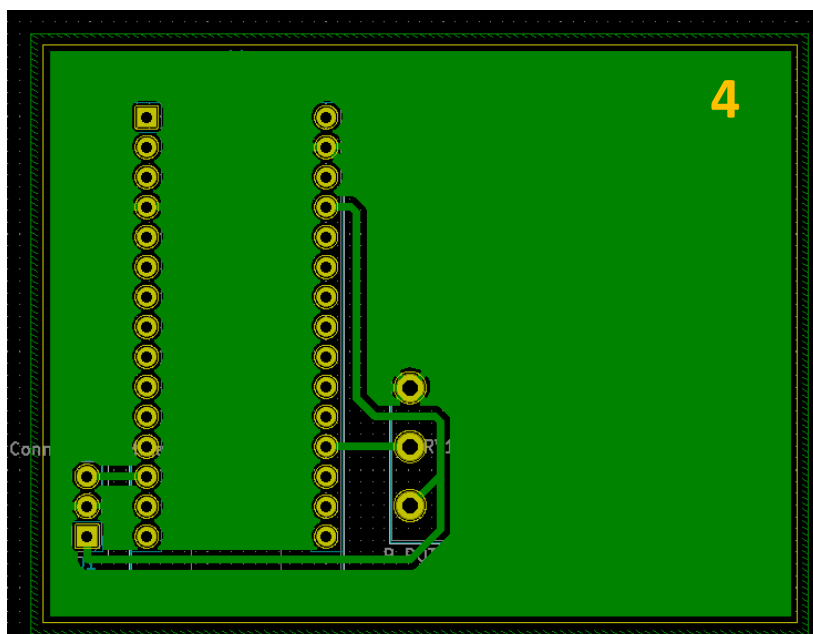
Plan de masse : plan d'équipotentiel relié à la masse du circuit. Utilisé pour diminuer l'impédance de retour et rendre le circuit plus résistant au bruit électromagnétique. Il faut *toujours* en mettre un.

Marche à suivre :

- Sélectionner la bottom-layer
- Sélectionner l'outil de dessin de zone
- Cliquer en dehors du PCB, près d'un coin
- Configurer le plan de masse
- Fermer un polygone autour du PCB

Note : Le plan de masse a tendance à rendre illisible le PCB. Vous pouvez Activer/désactiver la vue des plans d'équipotentiels (voir figure 6)





Historique des versions

Version	Date	Auteur	Description
0.1	17/07/2020	A.Schlumberger	Création du document, première structure, rédaction jusqu'à Kicad 0.1
0.5	25/07/2020	A.Schlumberger	Finalisation Kicad0.1, rédaction Kicad0.2 et 0.3
0.9	22/08/2020	A.Schlumberger	Finalisation de la partie routage
0.95	22/08/2020		Correction après relecture