

06 Concevoir son projet

Dans ce chapitre, on va analyser comment concevoir un projet de modélisme ferroviaire et on prendra comme exemple le projet de PN (*passage à niveau*) qui a été imaginé pour le réseau Train' In Box (*TIB*), projet qui a été publié dans Loco-Revue (*janvier et février 2022*) en collaboration avec le site LOCODUINO ([figure 6-1](#)).

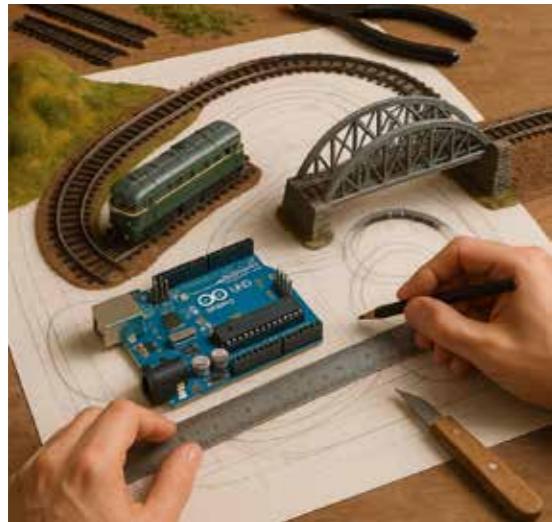
La première question qu'il faut se poser est **que veut-on vraiment faire ?** Il faut définir un **cahier des charges**, puis penser aux **différentes étapes** qu'il faut réussir avant de passer aux étapes suivantes. Car un projet quel qu'il soit doit se construire par étapes, en

commençant toujours par quelque chose de simple qu'on va compliquer progressivement. Vouloir tout faire d'un coup est voué à l'échec.

6.1 / LE BUT À ATTEINDRE : CONCEPTION D'UN CAHIER DES CHARGES

Le but, on le connaît : reproduire exactement un PN automatisé de type SAL 2 (deux barrières). La première chose à faire est de chercher sur YouTube des vidéos de ce type de PN. Cela permet de définir les différentes phases de fonctionnement :

1. Détection d'un train en approche
2. Sonnerie et clignotement des feux routiers
3. Attendre un certain délai (de l'ordre de 8 secondes)
4. Abaissement et fermeture des barrières
5. Arrêt de la sonnerie alors que les feux continuent à clignoter
6. Train qui passe, une fois passé :
7. Arrêt du clignotement et ouverture des barrières simultanément



Concevoir son projet (image réalisée par une IA)

Figure 6-1
Réseau Train'in Box de l'auteur



Le **point 1** peut se faire soit avec des détecteurs d'occupation par consommation de courant, soit avec des capteurs ponctuels comme des I.L.S. Mais le PN doit pouvoir être installé sur un réseau déjà fini ; la solution I.L.S est préférable car simple à ajouter.

Le **point 2** est facile si on emploie un mini-player de fichier MP3 ; de plus, il existe une bibliothèque publiée par LOCODUINO pour le clignotement des feux en simulant l'inertie thermique du filament.

Le **point 3** n'est pas évident sur un petit réseau comme TIB, il faut donc pouvoir régler ce délai à 0 seconde sur TIB, tout en gardant la possibilité de le régler à 8 secondes sur un réseau plus grand. Les paramètres doivent être réglables pour que le PN soit adapté au réseau possédé par le modéliste.

Le **point 4** fait appel à des servomoteurs synchronisés dont le mouvement dépend de la géométrie de la barrière.

Les **points 5, 6 et 7** ne présentent pas de difficultés particulières. Il faut ajouter à cela d'autres contraintes : le PN doit pouvoir être construit sans machines-outils spécialisées (tour, fraiseuse, imprimante 3D) et avoir l'électronique la plus simple possible, son prix de revient doit être inférieur à ce qu'on peut trouver d'équivalent dans le commerce. De plus, le PN doit pouvoir être ajouté à un réseau déjà construit sans avoir à réaliser d'importants travaux comme tronçonner des rails et réalimenter des portions de voies.

Tout cela forme donc un cahier des charges à respecter.

6.2 / LES ÉTAPES SUCCESSIVES

Comme je l'ai déjà dit, un projet important ne peut **se construire que par étapes successives**. Pour ce PN, les différentes étapes à réussir sont les suivantes :

1. Déetecter les trains à l'approche du PN
2. Faire clignoter les feux de PN
3. Faire manœuvrer les barrières
4. Ajouter le son

Pour l'**étape 1**, on peut définir une zone autour du PN pouvant être constituée de plusieurs voies avec des I.L.S aux frontières. Comme il ne faut pas louper de déclenchement d'I.L.S, on opte pour une **interruption externe**.

Pour l'**étape 2**, on peut utiliser la **bibliothèque LightDimmer de LOCODUINO** publiée quelques années auparavant. Elle a pour avantage de reproduire l'inertie thermique des lampes à filament.

L'**étape 3** est la plus difficile : la barrière utilisée (Auhagen réf41625) nécessite **un déplacement de la tige de commande en plastique de 2,5 mm** ! Le problème devient donc purement mécanique. Pour respecter cette cote, une came excentrique tournant de 90° est la meilleure solution, mais la tige étant flexible et il est nécessaire de la renforcer (**figure 6-2**).

L'**étape 4** a déjà été réalisée par LOCODUINO pour un projet d'annonces en gare ; cependant, à l'époque de la conception de ce PN, je n'ai pas réussi à intégrer cette fonction dans ce que j'avais déjà produit et je suis parti sur l'idée d'une **deuxième carte Arduino gérant uniquement la partie son**.



Figure 6-2

la came (à gauche) et le support du servomoteur (à droite)

Aucune étape ne doit être entreprise sans que les étapes précédentes aient donné satisfaction. Finalement, l'intégration du PN sur le réseau TIB peut se faire sans trop de problèmes, le circuit étant divisé en deux parts pratiquement

égales, dont une est la zone du PN (les trains doivent être détectés assez loin du PN). Pour TIB, il n'y a pas de délai d'attente entre sonnerie et fermeture des barrières (le réseau étant trop petit) mais ce délai peut être réglé si on dispose d'un réseau plus grand.



Figure 6-3

Avec toutes ces contraintes, des essais d'endurance ont été réalisés et le **train a parcouru plus de 100 fois le tour du réseau TIB** sans qu'il n'y ait le moindre dysfonctionnement (**figure 6-3**).

Des améliorations sont possibles, notamment réaliser un PN pour deux voies indépendantes l'une de l'autre. Sur le réseau 'Train' In Box, un seul train circule à la fois et les deux voies peuvent être traitées comme une seule. Ce qu'on sait faire pour une voie peut être recopié pour une deuxième voie. Il suffit ensuite d'un peu de logique pour déterminer l'état d'occupation de la zone PN. La zone PN est libre si la voie 1 ET la voie 2 sont libres. La zone PN est occupée dès que la voie 1 OU la voie 2 est occupée. Une fois l'état de la zone PN connu, le traitement est le même que pour le réseau TIB.

Le lien ci-dessous vous permet de voir le fonctionnement du PN avec deux voies :
<https://wokwi.com/projects/428656938257258497>

Remarquez que **les voies 1 et 2 peuvent être utilisées dans les deux sens et que le refoulement est pris en compte**. Si on appuie deux fois sur un poussoir en moins de 2 secondes, seul le premier appui est pris en compte car on estime que c'est le même survol d'I.L.S. Le délai entre clignotement et fermeture des barrières est de 4 secondes, mais on peut le modifier en ligne 20 du programme (selon la longueur du réseau). En montage réel, on mettrait deux LED rouges en série avec une résistance de 470 ohms. Il ne reste plus qu'à ajouter la partie sonorisation, ce qui est très facile vu qu'on a fait le choix de la réaliser avec une carte à part.

6.3 / UTILISATION DES SIMULATEURS D'ARDUINO

Les simulateurs d'Arduino comme [Tinkercad](#) ou [Wokwi](#) permettent de gagner du temps pour la mise au point de programme : on économise le temps de téléchargement de la carte et on peut commencer la mise au point avant même d'avoir construit le prototype. J'ai utilisé Wokwi pour **tester les programmes délivrés par l'IA ChatGPT dans mon projet de réseau EX_MACHINA**. Je voyais en temps réel si le programme fonctionnait ou bien devait être corrigé. La plupart des fonctions ont marché du premier coup (voir le chapitre 3), **preuve que l'IA peut rendre bien des services**. Mais même si on développe sans elle, l'utilisation des simulateurs, malgré leurs limites, est fortement recommandée. Le projet de pont tournant du

chapitre 8 a été mis au point en partie (mais pas totalement) sur le simulateur **Wokwi** pour la partie commande du pont et contrôle de la position sur un écran I2C. Même les relais ont été simulés par de simples LED pour tester leur mise en route. On comprend alors **qu'il ne faut pas s'arrêter au fait que tous les composants ne sont pas disponibles** : on peut les remplacer par d'autres ou les simuler par des LED. Les simulateurs **permettent aussi de partager son travail avec d'autres personnes** et de travailler en équipe (voir chapitre 1). Après la mise au point sur simulateur, on peut entreprendre la construction du prototype, faire les essais in-situ et faire également des **tests d'endurance**, comme cela a été fait pour le PN.

6.4 / UTILISATION DE L'INTELLIGENCE ARTIFICIELLE

L'intelligence artificielle permet aussi de gagner du temps dans le développement d'un projet, mais à condition qu'elle soit bien utilisée. L'utilisation de l'IA a été évoquée dans le chapitre 3. Il faut la considérer comme un outil et à ce titre, elle peut écrire du code à notre place à condition de bien lui expliquer toutes les contraintes à respecter et le but exacte à atteindre. Mais l'IA ne peut pas tout faire ou tout décider et votre rôle est de rester chef d'orchestre, seul et unique décideur du projet que

vous voulez réaliser. Certains projets peuvent avoir recours à plusieurs langages (C, C++, Processing, HTML) : imaginez le temps que vous perdriez à les apprendre alors que l'IA les connaît. Il faut tout de même se méfier car l'IA peut aussi ne pas connaître certaines bibliothèques qui sont peu connues (par exemple, celle que j'ai écrite et qui n'est pas suffisamment utilisée dans le monde pour faire partie des connaissances de l'IA).

6.5 / TRAVAILLER EN ÉQUIPE

Travailler en équipe permet d'être plus créatif et, une fois de plus, de gagner du temps. Par exemple, vous pouvez travailler sur une centrale DCC parce que vous en avez déjà fabriqué une, pendant qu'un ami travaillera sur l'intégration d'un bus CAN sur cette centrale. Un autre, plus férus d'électronique, peut se charger de la conception du circuit imprimé et de choisir les composants.

L'utilisation de la plateforme GitHub (voir chapitre 1) permet justement le partage des résultats et devient l'outil idéal si vous ne résidez pas dans la même région. Comme je l'ai dit plus haut, la rédaction d'un cahier des charges, et d'un plan de progression si possible associé à un calendrier, permet de planifier la réalisation du projet, notamment si le résultat doit être montré en

exposition dont la date est déjà fixée. Il convient de **définir les rôles de chaque membre de l'équipe en fonction des compétences et de la disponibilité**. Il est prudent de compter large pour le calendrier car certains problèmes peuvent retarder les choses. Enfin, une équipe de travail

dans un loisir n'a rien à voir avec une équipe dans une entreprise ; les gens sont là pour se faire plaisir et personne ne peut les forcer à agir. Il est donc indispensable de bien choisir ses partenaires et de bien connaître les motivations de chacun.

6.6 / RÉALISATION DES CARTES ÉLECTRONIQUES ET DES INTERFACES

Les cartes à microcontrôleur ont **besoin d'un peu d'électronique pour bien fonctionner**, par exemple pour assurer l'isolation galvanique de tensions très différentes, pour adapter les signaux aux tensions des périphériques, pour

SOLUTION TOUTE FAITE EXISTANT SUR LE MARCHÉ

Il suffit de **chercher sur les sites de vente par correspondance** pour trouver ce qu'on veut et à des prix souvent modiques, et on aurait tort de se priver. Inutile donc de vouloir développer sa carte relais puisqu'on en trouve à des prix défiant toute concurrence, inutile également de développer sa carte moteur car il en existe quasiment dans toutes les marques. On trouve aussi des cartes permettant d'adapter les signaux aux tensions requises (3,3 ou 5 V) et des cartes

amplifier des signaux de sortie, pour filtrer des signaux inutiles, etc. Il y a deux solutions pour produire cette électronique : chercher ce qui existe déjà ou bien concevoir sa propre carte.

qui amplifient les signaux de sorties des cartes Arduino. Profitons de la pléthore des offres, la difficulté étant de trouver vraiment ce qu'on cherche (dans certains cas, ce n'est pas si facile et il faudra vous renseigner auprès d'autres amateurs qui ont peut-être eu le même problème avant vous). La **figure 6-4** montre deux cartes du commerce, celle de gauche est une carte à optocoupleurs de 4 canaux, celle de droite est une carte bus CAN.

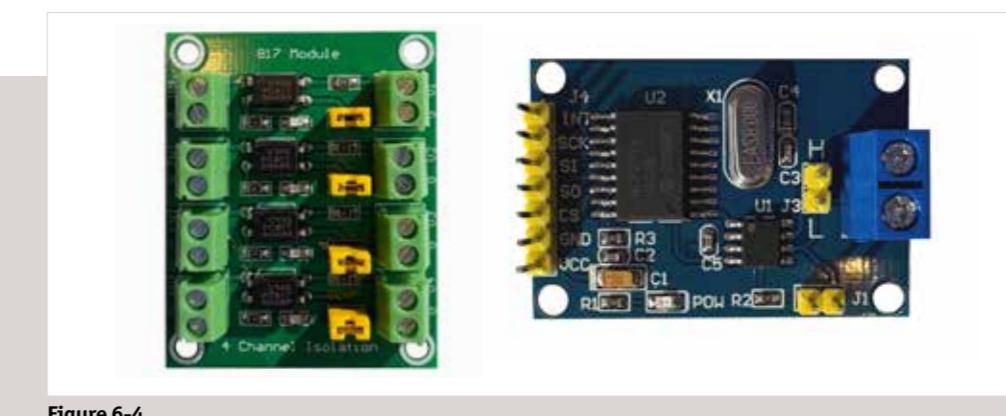


Figure 6-4

CONCEPTION AB-NIHILO

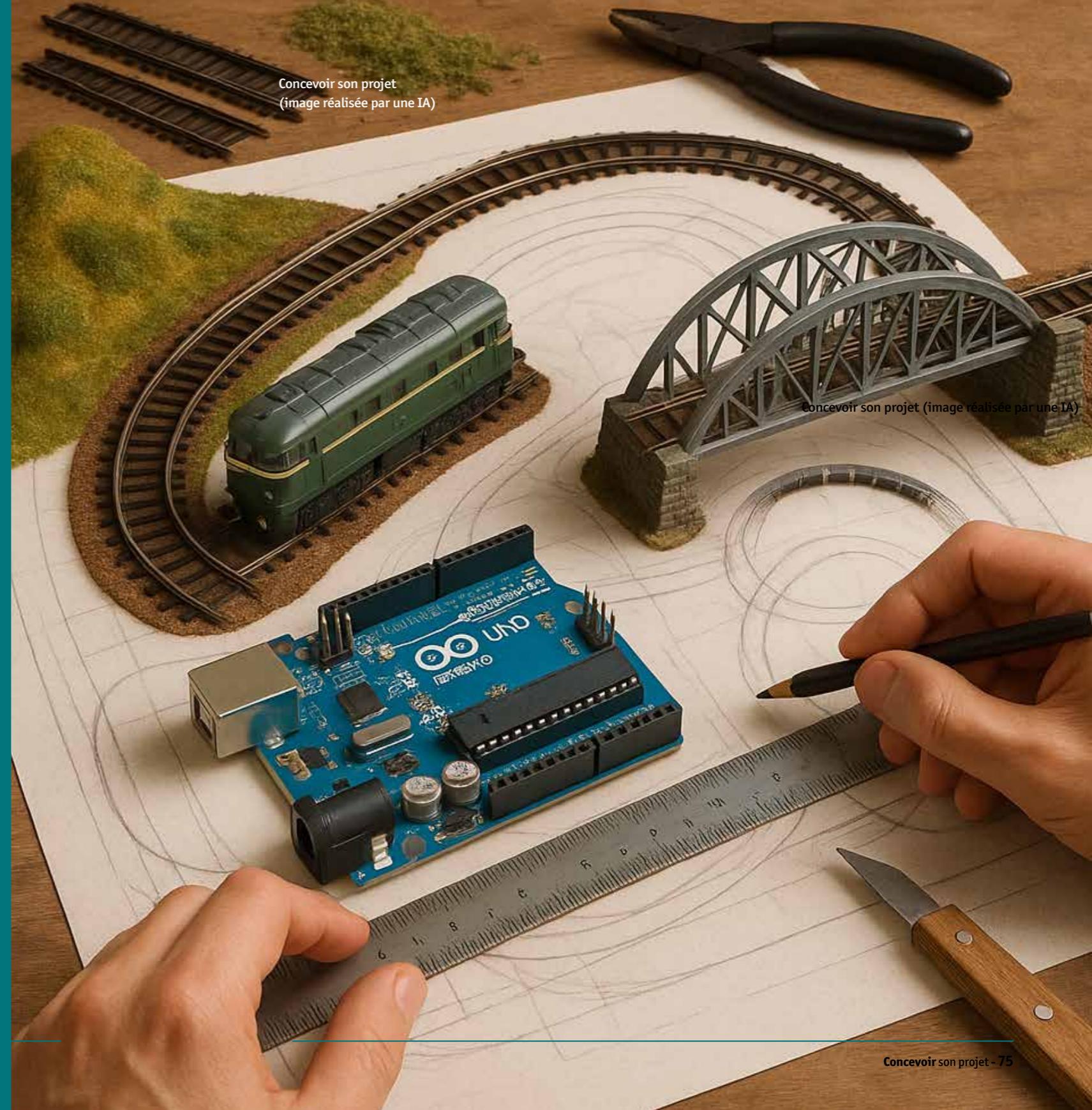
Si la carte désirée ne se trouve pas, il reste la solution de la **développer vous-même et en faire profiter ensuite la communauté**. Pour avoir une carte de qualité professionnelle, il faut concevoir un circuit imprimé et le faire réaliser par une entreprise spécialisée (voir le chapitre 1). Souvent la carte Arduino vient s'enficher sur la carte électronique (solution pour la série Nano) mais on peut aussi développer la carte électronique au format des cartes shield et elle vient alors s'enficher sur la carte Arduino. Dans certains cas, il peut être préférable de développer une carte à part qui sera connectée à la carte Arduino par des **câbles en nappes**, comme c'était le cas pour les ordinateurs avec leurs disques durs avant l'arrivée de la norme SATA (transmission série). La **figure 6-5** montre une carte double face développée pour des animations lumineuses à base d'ATTiny45.



Figure 6-5

RÉSUMÉ DU CHAPITRE 6

Ce chapitre est le complément du chapitre précédent. Un projet d'automatisation est bien souvent une tâche difficile qu'il faut savoir diviser en un ensemble de tâches élémentaires plus simples. En effet, il faut toujours partir de quelque chose de simple et complexifier au fur et à mesure. Une tâche doit être parfaitement opérationnelle pour passer à la tâche suivante, comme je l'ai expliqué au sujet du développement d'un PN pour le réseau TIB. L'utilisation de l'Intelligence Artificielle permet de gagner du temps et l'utilisation de simulateurs pour vérifier ce qu'elle délivre, permet souvent de corriger certains problèmes avant même de fabriquer le premier prototype.



Concevoir son projet
(image réalisée par une IA)

Concevoir son projet (image réalisée par une IA)