

**HUGO DENIER**

FILIÈRE STI2D OPTION SYSTÈME  
D'INFORMATION ET NUMÉRIQUE

# PROJET BLOCS | 2017



LYCÉE EUGÈNE LIVET,  
NANTES

Depuis la rentrée scolaire 2016, certaines écoles primaires françaises enseignent la programmation informatique dès l'âge de 6 ans. Les élèves apprennent à programmer les déplacements d'un robot ou d'un personnage sur un écran.

Durant 70 heures, nous avons dû réaliser un projet pour clôturer notre cycle STI2D de spécialité *Système d'Information et Numérique (SIN)*, débuté en septembre 2015. Ce projet a été réalisé par un groupe de 4 étudiants de ma spécialité ainsi que 3 élèves d'option *Innovation Technologique et Éco-Conception (ITEC)* toujours dans la filière STI2D.

**01**

## PRÉSENTATION GÉNÉRALE DU PROJET

page4

A - Diagramme de contexte.....	5
B - Diagramme d'exigence.....	6
C - Bloc programmable.....	7
D - Le bloc maître.....	8
E - Le robot.....	9
F - Répartition des tâches.....	10

**02**

## UNE COMMUNICATION ADAPTÉE

page11

A - Une liaison bluetooth.....	11
B - Une liaison radio.....	12
C - Algorithme .....	14

**03**

## ÉLABORATION D'UN ROBOT

page15

A - Motorisation du robot.....	15
B - Les fonctions créées.....	17
C - Algorithmes du robot.....	18

**04**

## CONCLUSION

page19

A - Synthèse.....	19
B - Annexes.....	20

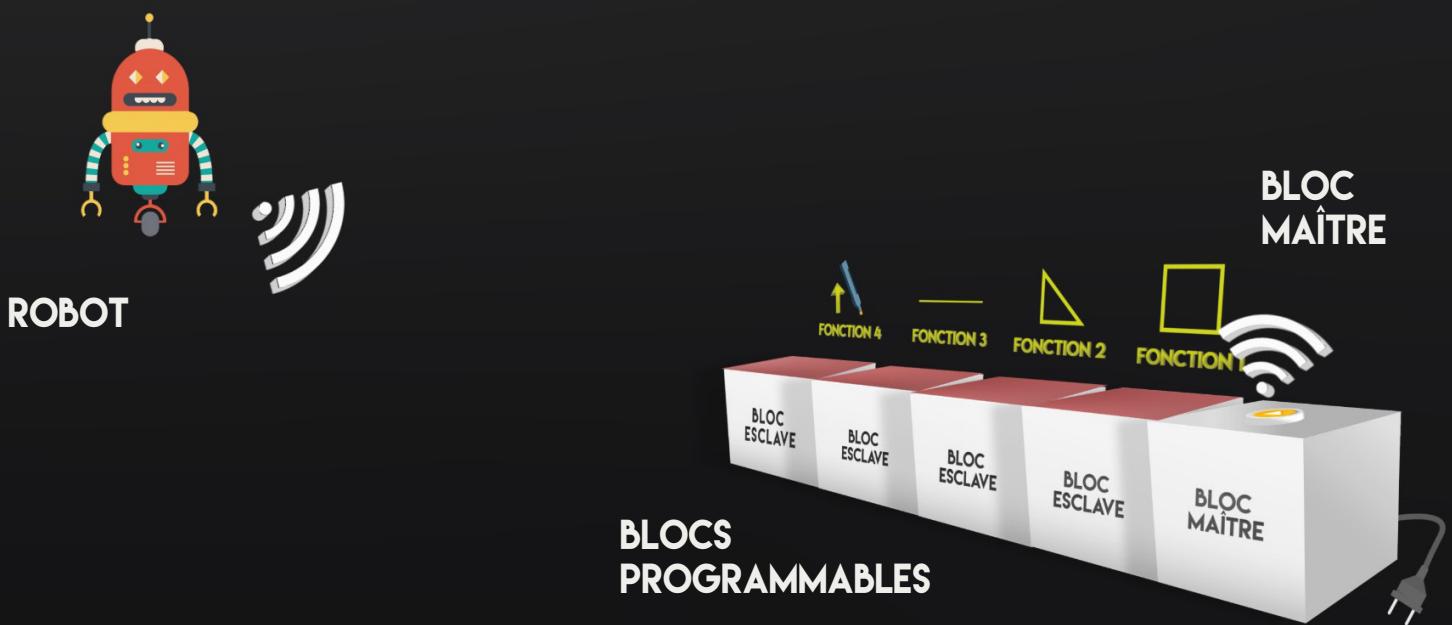
# 1 PRÉSENTATION DU PROJET

Appelé Projet Blocs, il est destiné aux enfants de 5 à 6 ans et a pour objectif d'offrir un esprit logique de programmation dès le plus jeune âge tout en gardant un esprit ludique. Nos exigences étaient de concevoir un robot pouvant réaliser au maximum 16 fonctions. Une fonction peut -être à la fois une forme ou une simple action tel que multiplier la fonction précédente.

## EXIGENCE GLOBALE

<>requirement>>  
**Programmer à l'aide de cube**

text="Le placement de cube de fonction permet de programmer le déplacement du robot ainsi que le pilotage du crayon."  
Id="0"



Notre projet se décompose en 3 parties distinctes : le robot en haut à gauche, les blocs programmables (*appelés aussi blocs esclaves*) et le bloc maître. Sur les blocs programmables, se fixent des couvercles interchangeables ayant pour but de faire réaliser une fonction par le robot. Les enfants vont créer une séquence en empilant les blocs programmables les uns à la suite des autres. Cette séquence est rattachée au bloc maître, qui lui-même, va rassembler les informations pour les transmettre au robot. Le robot va exécuter les fonctions dans l'ordre demandé.

**LES ENFANTS VONT CRÉER UNE SÉQUENCE**  
EN EMPILANT LES BLOCS PROGRAMMABLES  
LES UNS À LA SUITE DES AUTRES

**01**

**02**

**03**

**04**

**LES ENFANTS VONT ENCLENCHER LE PROCESSUS DE DÉMARRAGE**

EN APPUYANT SUR LE BOUTON "START"  
DU BLOC MAITRE

**LE BLOC MAITRE VA RASSEMBLER ET TRANSMETTRE AU ROBOT**

LES FONCTIONS DANS L'ORDRE DEMANDÉ

**LE ROBOT EXÉCUTE LE PROGRAMME REÇU**

ET VA REPRODUIRE LES DIFFÉRENTES FONCTIONS SUR UNE FEUILLE DE PAPIER

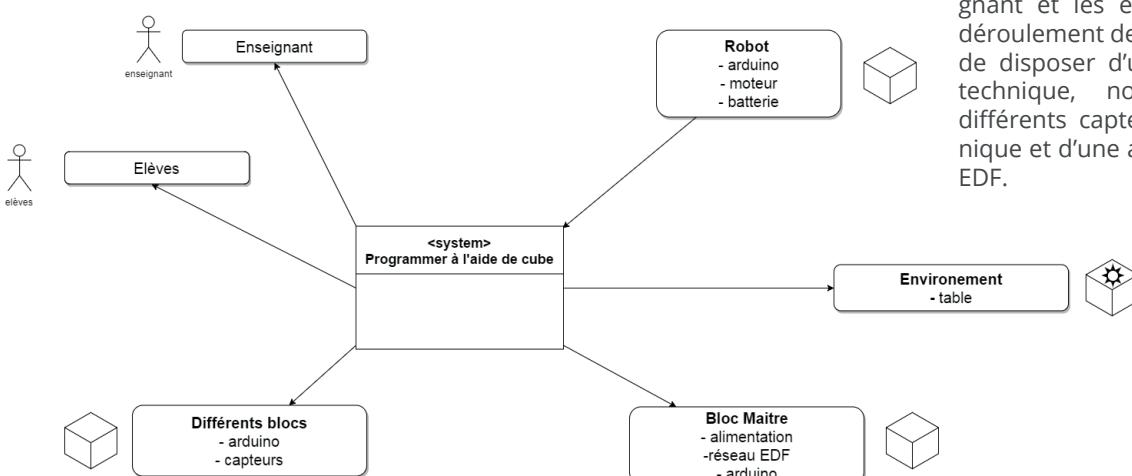
**A**

## > DIAGRAMME DE CONTEXTE

bdd [system] Diagramme de contexte [context etendu]

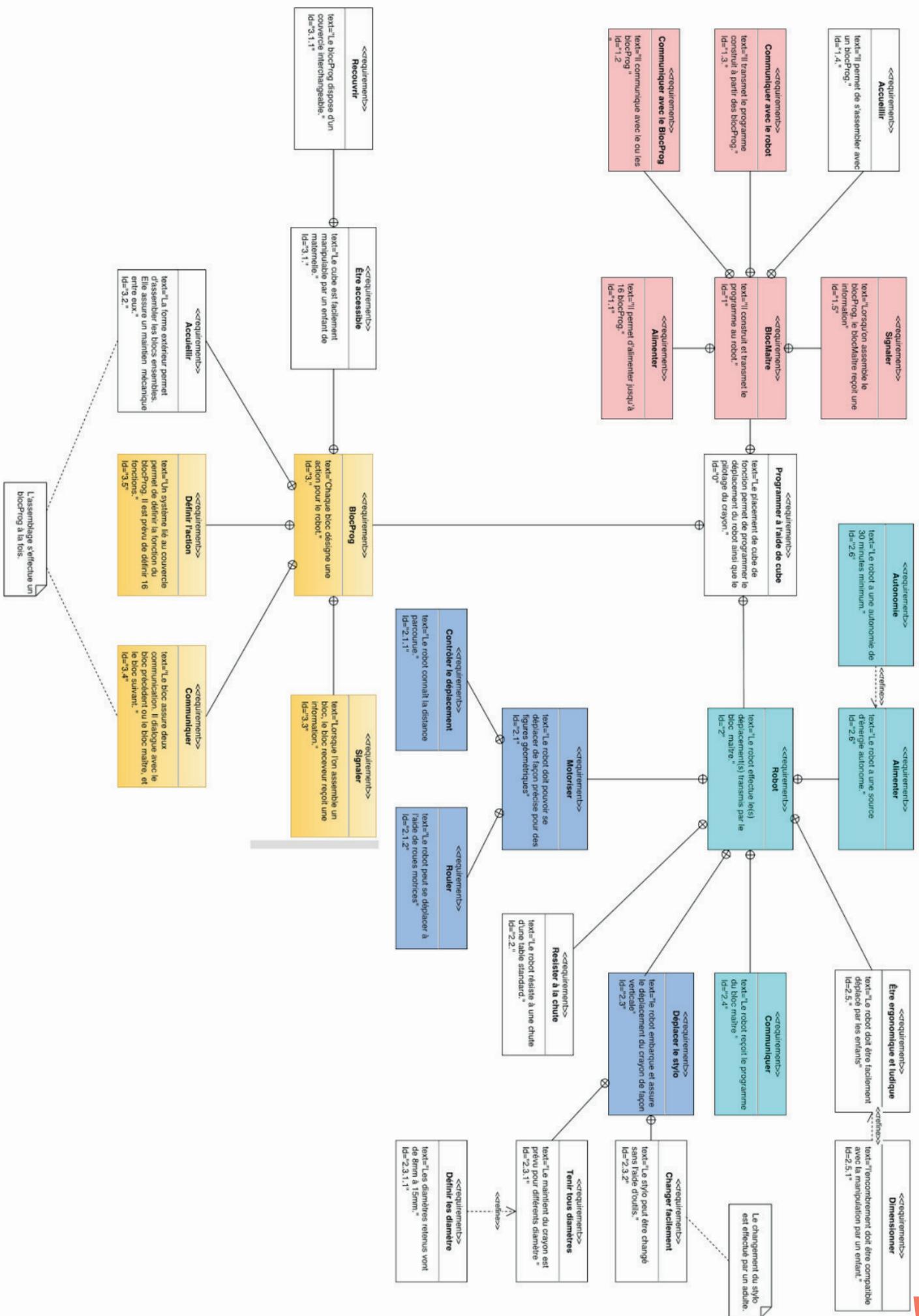
*Le diagramme de contexte va exprimer l'environnement du système. Il fait réfléchir sur les acteurs et éléments environnants du système.*

Pour ce projet, il y a deux acteurs : l'enseignant et les élèves. Utile pour le bon déroulement de l'activité, il est nécessaire de disposer d'une table. Pour la partie technique, nous avons besoin de différents capteurs, d'une carte électronique et d'une alimentation sur le réseau EDF.



## > DIAGRAMME D'EXIGENCE

req Programmer à l'aide de bloc



Dans cette partie, vous allez pouvoir mieux comprendre le fonctionnement des trois éléments constituant le système. Vous y trouverez trois diagrammes de cas d'utilisation (permettant de décrire l'interaction entre les acteurs et le système) ainsi que trois chaines d'information et d'énergie (permettant de mettre en évidence son fonctionnement).

C

## BLOC PROGRAMMABLE

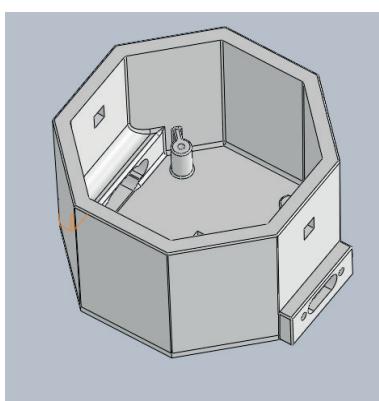
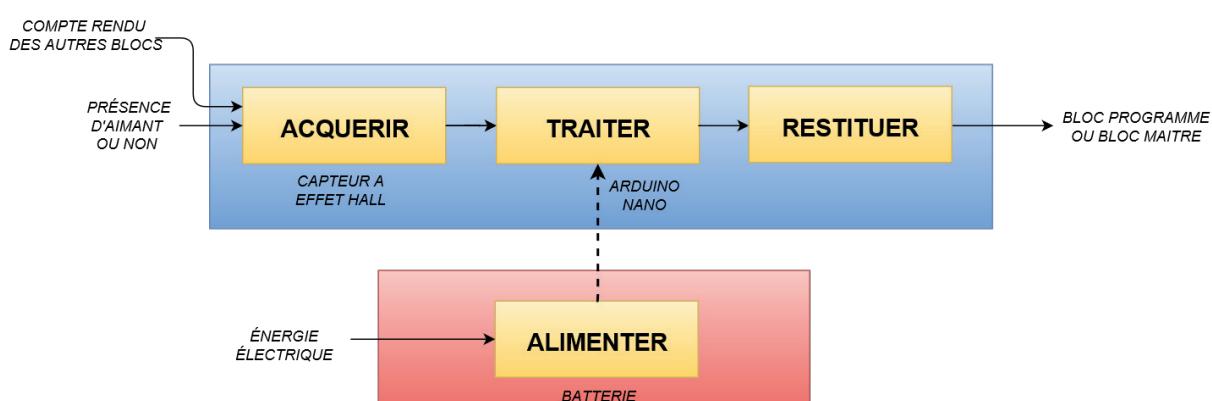
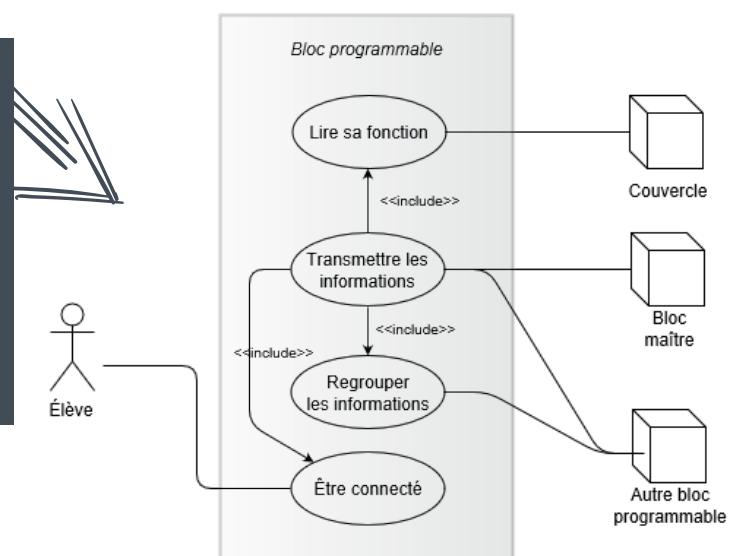
PAGE  
voir schéma  
structurel **21**

### DIAGRAMME CAS D'UTILISATION

*Le but du bloc programmable est de récupérer les informations des autres blocs derrière lui et de les transmettre au bloc maître.*

Afin que les blocs programmables transmettent les informations, l'élève doit le connecter au reste du système pour le fournir en énergie. Le bloc programmable va lire la fonction via le couvercle et va regrouper l'information avec les informations des autres blocs derrière lui.

Le but étant que les informations passent de bloc en bloc pour arriver au bloc maître.

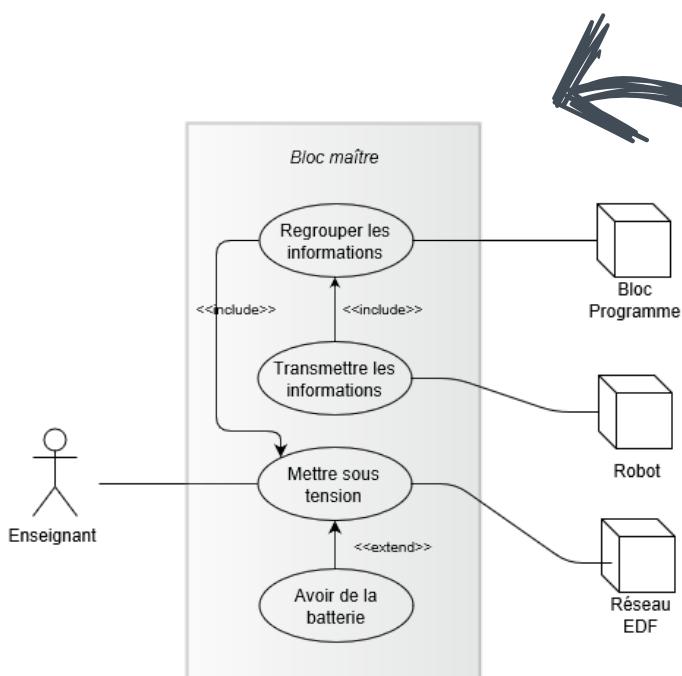


### CHAINE D'INFORMATION ET D'ÉNERGIE

Le capteur à effet hall va capter la présence d'aimant. Si oui, il va informer la carte arduino du nombre d'aimant. Ce qui va permettre de retrouver la fonction correspondante.

La carte arduino va devoir également traiter le compte rendu des blocs précédents pour le transmettre au bloc maître.

Pour ce qui concerne la chaîne d'énergie, la carte arduino va être alimentée par la batterie contenue dans le bloc maître. C'est grâce à deux contacts que l'énergie électrique va pouvoir se propager.

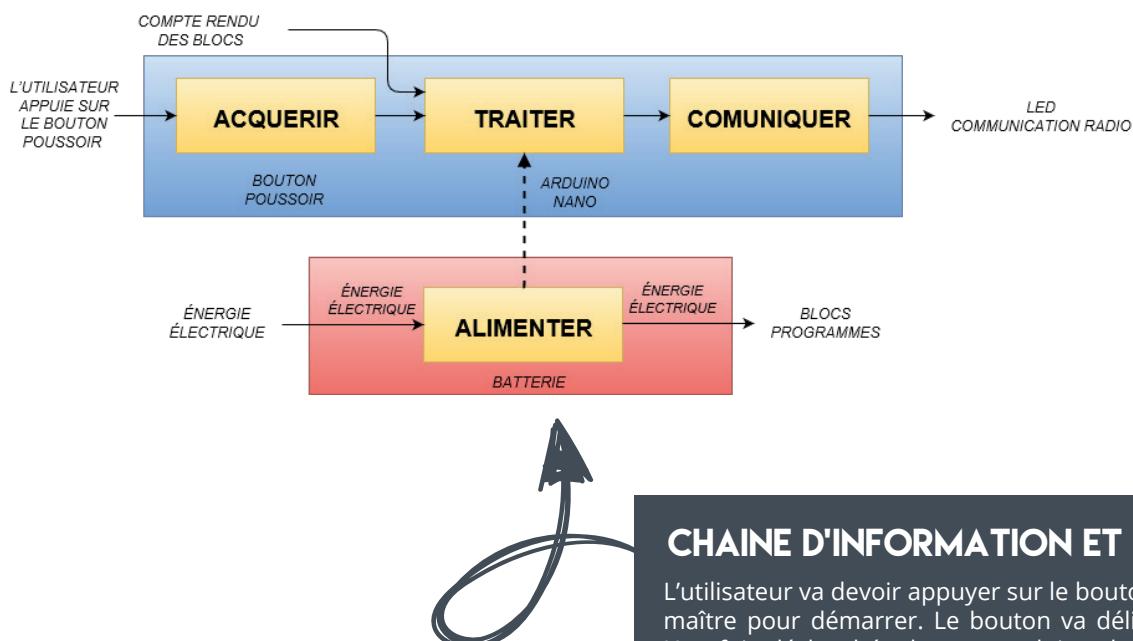
**DIAGRAMME CAS D'UTILISATION**

*Le bloc maître est le système nerveux, il fait la liaison entre les blocs programmables et le robot. Il reçoit et envoie des informations.*

Seul l'enseignant est autorisé à le mettre sous tension et pour cela il est nécessaire d'avoir une batterie chargée.

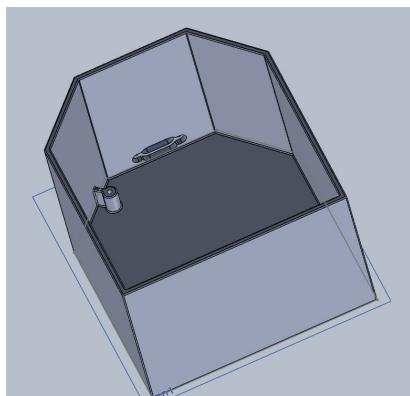
Une fois mis sous tension, le bloc maître peut regrouper les informations provenant des blocs programmables et enfin les transmettre au robot.

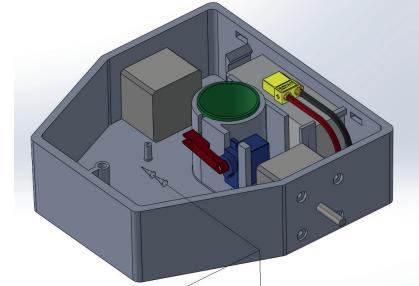
A noter que le bloc maître dispose de deux solutions énergétiques : une autonomie par batterie ou une prise de courant connectée au réseau EDF.

**CHAINNE D'INFORMATION ET D'ÉNERGIE**

L'utilisateur va devoir appuyer sur le bouton situé au dessus du bloc maître pour démarrer. Le bouton va délivrer un signal électrique. Une fois déclenchée, la carte arduino du bloc maître va traiter les informations provenant des blocs programmables et les communiquer au robot. Une néopixel (led) va servir d'indication de mise sous tension ou du niveau de la batterie.

Le bloc maître est alimenté uniquement d'énergie électrique par la batterie ou la prise secteur.



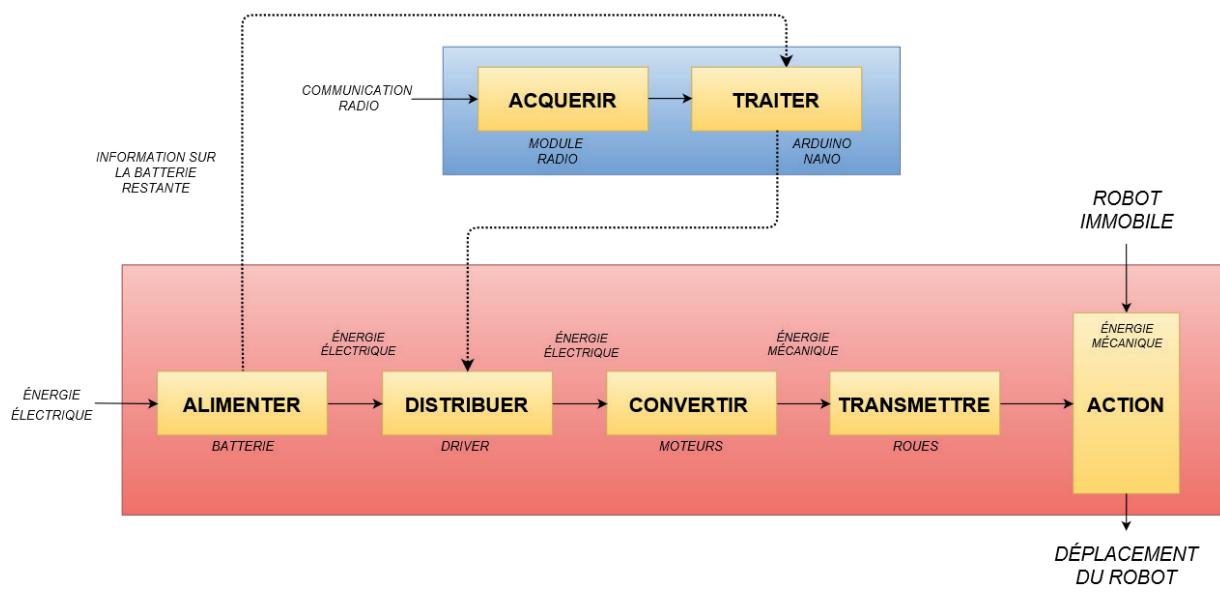
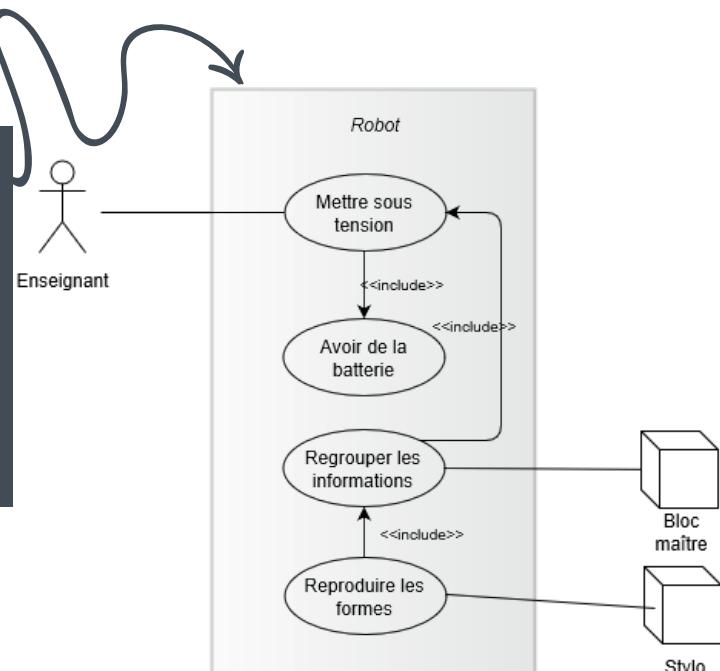


## DIAGRAMME CAS D'UTILISATION

Le robot à pour rôle de reproduire les fonctions transmises par le bloc maître.

Encore une fois, seul l'enseignant est habilité à mettre le robot sous tension.

Une fois mis sous tension, le robot va reproduire les fonctions à l'aide du stylo. Pour cela, il est obligatoire de regrouper les informations provenant du bloc maître.



## CHAINNE D'INFORMATION ET D'ÉNERGIE

Le robot va acquérir le programme provenant du bloc maître grâce au module radio. L'arduino va traiter ce programme et donner les instructions aux drivers des moteurs. La carte arduino va calculer le taux de remplissage de la batterie afin de connaître son autonomie.

Le robot va être alimenté en énergie électrique par sa batterie. Deux drivers vont distribuer cette énergie au moteur qui eux vont convertir cette énergie électrique en énergie mécanique. Ainsi, grâce aux roues, le robot va pouvoir se déplacer.

**F**

## > RÉPARTITION DES TÂCHES

Pour mener à bien ce projet, une bonne distribution des tâches a été essentielle. Je vais vous présenter les missions de Martin, Pierre et Sullivan ainsi que la mienne.



### MARTIN PORÉE

Le rôle de Martin a été de choisir des batteries adaptées à notre système dont l'exigence est de tenir 30 minutes.

Par la suite, il a dû mettre au point un programme pour connaître le niveau de la batterie afin d'alerter l'utilisateur.

Pour finir, il a réalisé un robot à l'aide de l'imprimante 3D pour pouvoir effectuer les premières expérimentations.

### PIERRE NICOLAS

Pierre a réalisé la communication entre les blocs programmables et le bloc maître.

Il a aidé à choisir les constituants et à co-réalisé les circuits imprimés.

### SULLIVAN LE TOUZIC

Sullivan a conçu un programme pour la détection du couvercle afin que le bloc programmable puisse connaître sa fonction.

Il a également co-réalisé avec Pierre les circuits imprimés permettant d'offrir un produit fini.

### HUGO DENIER

Quant à moi, comme vous le verrez dans ce compte rendu, je me suis occupé de créer la communication entre le bloc maître et le robot.

Je me suis également occupé du fonctionnement des drivers et moteurs pour faire fonctionner le robot.

## DÉCEMBRE ANALYSE DU BESOIN

Cette première étape nous a permis de prendre connaissance du projet, de l'étudier et chercher les premières idées. Lors du brainstorming, nous avons réfléchi à diverses méthodes possibles et par conséquent à différents diagrammes.



## MARS-AVRIL RÉALISATION

L'étape la plus concrète du projet étant la réalisation. Après la conception, il a fallu réaliser les différentes tâches et faire face aux nombreux problèmes techniques ou changements d'orientation.



## JANVIER-FÉVRIER CONCEPTION

Ce début d'année nous a permis de concevoir les premiers programmes, de réfléchir à la motorisation ou encore d'évaluer l'autonomie du robot.



## MAI VALIDATION

Étape ultime du projet. Fin du chrono réglementaire, c'est l'heure du bilan. Nous nous sommes regroupés pour réfléchir au système actuel et sur ce que nous pourrions améliorer.



## 2 UNE COMMUNICATION ADAPTÉE

Une des étapes les plus compliquées de ce projet a été de réaliser les différentes communications. Il a fallu trouver une communication entre le bloc maître et le robot. Plusieurs expérimentations ont été effectuées pour répondre correctement aux exigences suivantes :

ROBOT	BLOC MAÎTRE
<>requirement>> <b>Communiquer</b>  text="Le robot reçoit le programme du bloc maître " Id="2.4"	<>requirement>> <b>BlocMaître</b>  text="Il construit et transmet le programme au robot." Id="1"

Afin que le bloc maître communique avec le robot, il a fallu trouver une communication adaptée à la situation. Une solution sans fil qui accepte une distance nécessaire au bon fonctionnement du robot. Ce sont deux critères essentiels. Dans la suite de cette première partie, vous aller pouvoir visualiser nos différentes phases de tests.

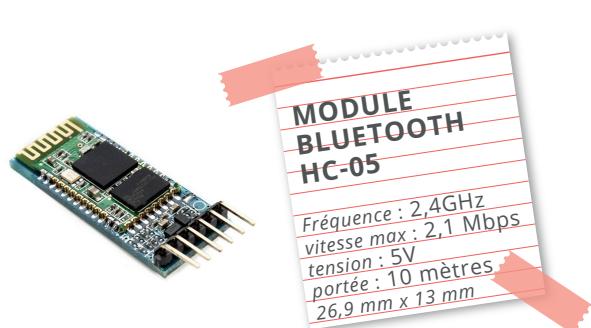
A

### ➤ UNE LIAISON BLUETOOTH

Connaissant bien la liaison bluetooth, nous avons décidé d'en tester une afin de voir si cette dernière pouvait correspondre à notre besoin. Déjà expérimentée depuis le début de notre cycle d'apprentissage, nous utilisons deux modules identiques (*HC-05*) faisant office d'émetteur et de récepteur.

Le branchement d'un tel module se fait sur quatre broches que l'on connecte directement à la carte arduino. L'un servira de transmetteur (*Tx*) et l'autre de récepteur (*Rx*) connectés aux pins digitaux ainsi qu'à la tension d'alimentation (5V) et à la masse (GND).

La liaison bluetooth est une communication RS-232. Les bits d'informations sont transmis tour à tour. Il n'y a pas de bit d'horloge. La liaison bluetooth ne peut relier que deux appareils comme ici l'émetteur et le récepteur. Elle fonctionne de la façon suivante : l'émetteur parle et l'autre écoute, et inversement.



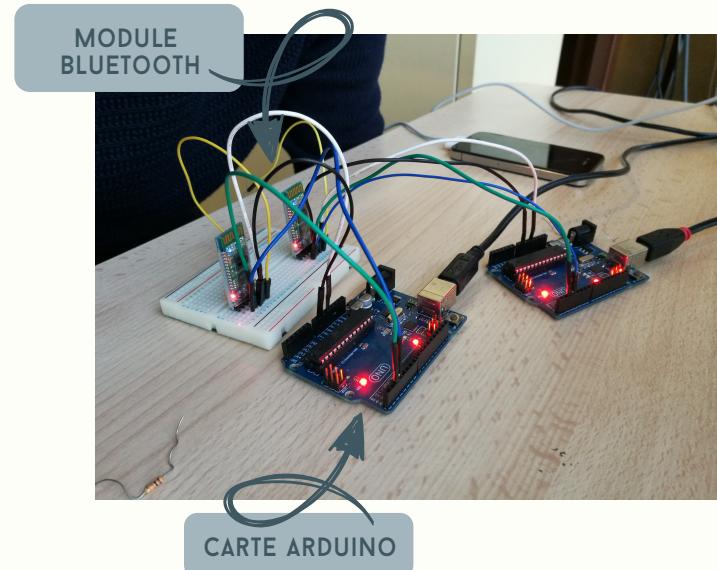
# RÉSULTAT DE L'EXPÉRIMENTATION

Pour effectuer le test, il a suffi de brancher un émetteur et un récepteur à chacune des cartes Arduino afin d'envoyer et de recevoir des données entre elles.

L'envoi de caractères fonctionnait à merveille mais rapidement un problème est survenu. En effet, pour que l'émetteur et le récepteur travaillent ensemble, il fallait obligatoirement les appareiller.

Cette étape a posé beaucoup de problèmes techniques car l'émetteur et le récepteur ne se connectent pas automatiquement ensemble. C'est une étape manuelle qui doit être effectuée à chaque mise en route du système. Cela voudrait dire qu'il faudrait ouvrir les boîtiers à chaque début d'activité.

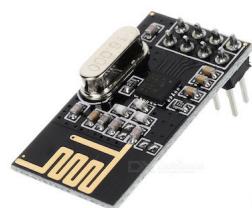
**Sans solution, nous avons été contraints d'abandonner cette idée et de se reporter sur une communication radio.**



B

## > UNE LIAISON RADIO

Dû à l'échec du bluetooth, nous nous sommes reportés sur une liaison radio. Pour un gain de rapidité, nous avons utilisé les mêmes modules étudiés quelques mois plus tôt. Même fonctionnement que le bluetooth, nous disposons d'un émetteur et d'un récepteur identiques qui sont à configurer dans le logiciel Arduino selon le rôle qu'on leur donnera.



**MODULE  
RADIO  
nRF24L01**

Fréquence : 2,4GHz  
vitesse max : 2 Mbps  
tension : 3.3V  
portée : 250 mètres  
29 mm x 15 mm



Pour connecter ce module radio à l'arduino, il faut utiliser les huit broches comme sur le schéma à votre gauche. Il y a la tension appelée *VCC* et la masse *GND*. Les broches *CE* et *CSN* vont être connectées à l'arduino sur les ports digitaux 7 et 8. En haut à gauche, la broche *IRQ* va servir d'interruption déclenchée par la carte arduino pour arrêter la liaison. C'est par l'intermédiaire de ces trois fils notés *MOSI*, *MISO* et *SCK* que va s'effectuer la communication SPI.

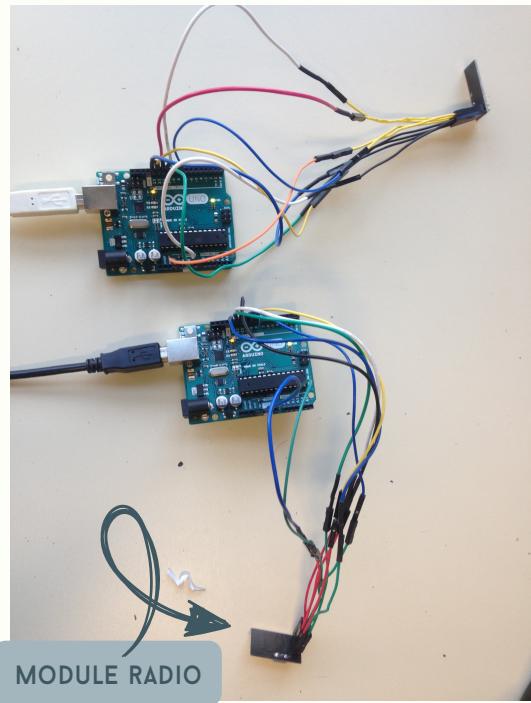
Cette liaison radio est une communication SPI. Elle s'établie entre un maître (émetteur) et un esclave (récepteur) et permet l'échange de données en série. L'échange de données peut s'effectuer dans les deux sens, ce qu'on appelle *fullduplex*. C'est une liaison synchrone c'est-à-dire que le récepteur reçoit de façon continue (même lorsqu'aucun bit n'est transmis) les informations au rythme où l'émetteur les envoie.

Pour information :

**MOSI** : Master Output Slave Input

**MISO** : Master Input Slave Output

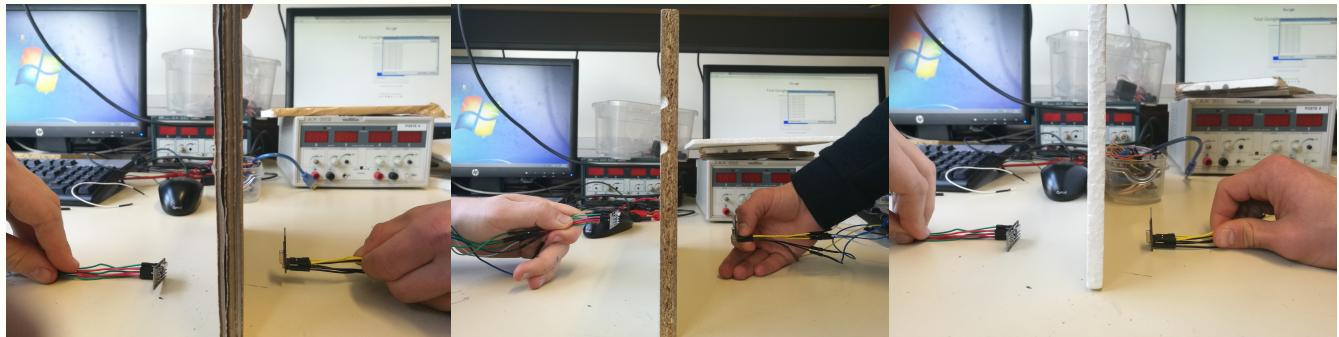
**SCK** : SPI serial Clock



Pour la réalisation du test, comme pour le bluetooth, nous avons simuler l'envoi de données entre le bloc maître et le robot. Les deux cartes arduino ont été branchées sur le même ordinateur. A l'aide de la fenêtre "moniteur série" du logiciel Arduino, nous avons vu que l'envoi et la réception de caractères fonctionnaient correctement.

L'avantage important de cette communication est que lorsque le module est mis sous tension, l'émetteur et le récepteur vont se mettre en relation directement en attendant l'échange d'informations.

*Capture d'écran du récepteur, première phrase échangée lors du test de la communication.*



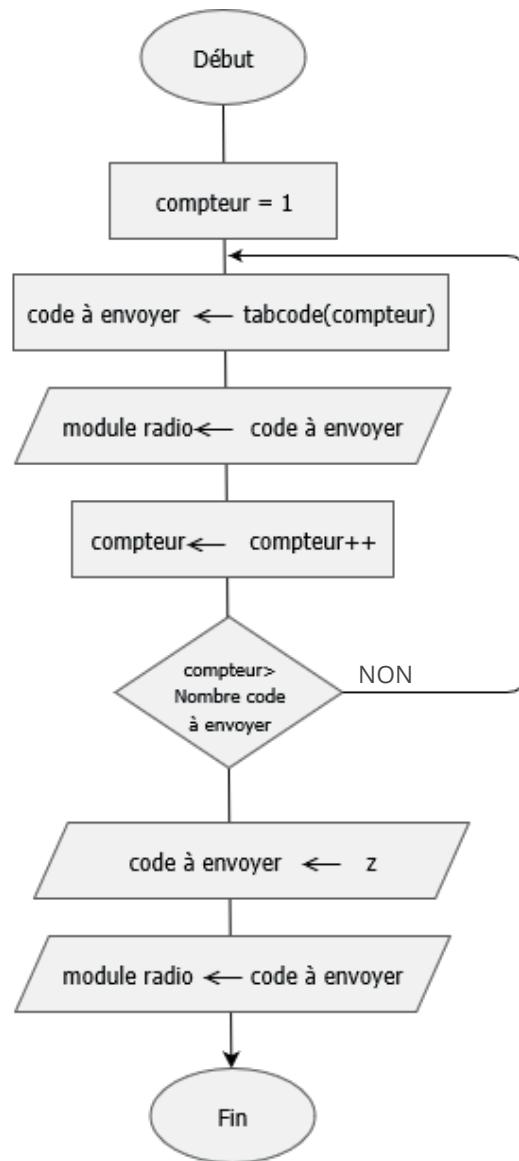
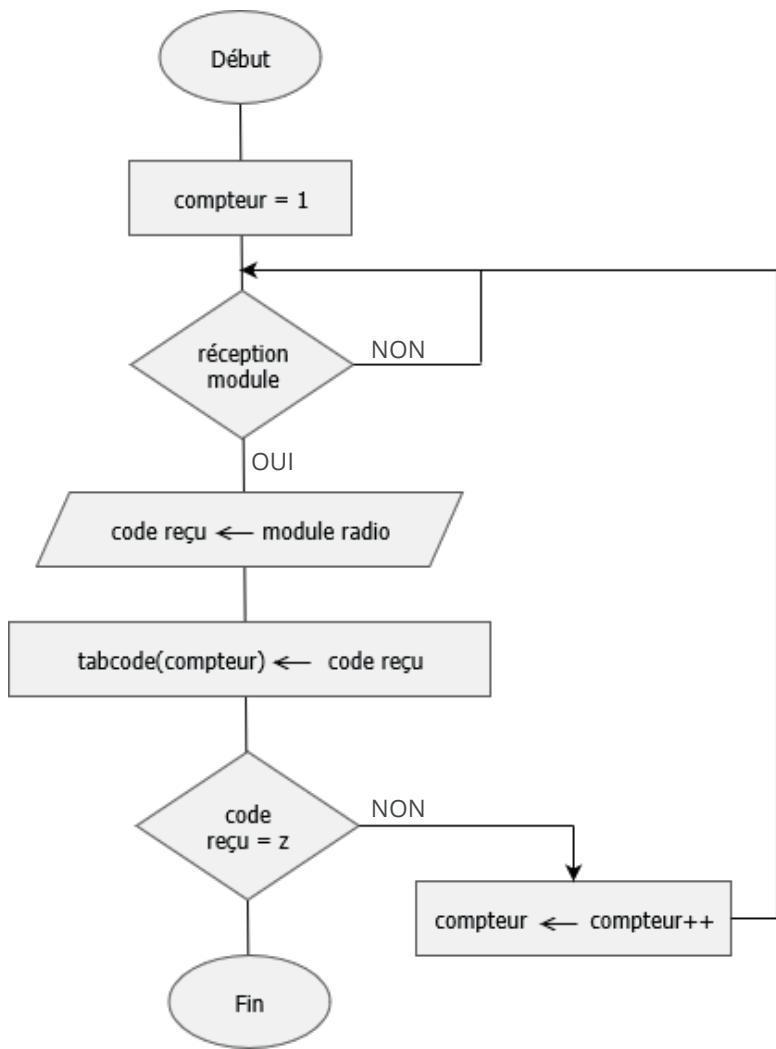
Avant de valider cette hypothèse, nous avons voulu quand même tester la capacité de l'onde à traverser un matériau. Les modules radios sont contenus dans le bloc maître et le robot. Il faut pouvoir être certain que l'émetteur et le récepteur vont pouvoir communiquer à l'intérieur de leurs blocs respectifs. Après le test avec les différents matériaux tels que le bois, le carton, le polystyrène et bien évidemment le plastique, nous avons démontrer que dans tout ces cas de figures, l'onde pourra traverser les matériaux sans perturber le système.

Ces algorithmes représentent la suite d'instructions que la liaison radio va exécuter pour communiquer entre le bloc maître et le robot. L'un représente les instructions du bloc maître en tant qu'émetteur et le second en tant que récepteur dans le robot. Grâce aux algorithmes, nous avons pu créer le code de programmation plus facilement.

## BLOC MAÎTRE

Il faut savoir qu'avant d'envoyer les informations, le bloc maître récupère l'ordre et les fonctions des blocs programmables présents dans la séquence pour ensuite placer les fonctions dans un tableau de programmation. Une fonction est représentée par un caractère.

Chaque caractère va se placer l'un après l'autre dans ce tableau jusqu'à trouver le caractère "Z" qui signale la fin du tableau et donc l'arrêt de la communication.



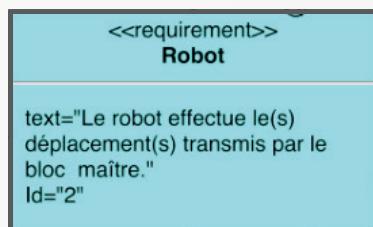
## ROBOT

Le robot est en attente de la réception de la communication radio. S'il y a liaison radio, le robot va stocker le tableau envoyé par le bloc maître. Pour connaître le processus de traitement de ces informations, il vous suffit d'aller page 18.

## 3 ÉLABORATION D'UN ROBOT

Lors de l'élaboration du robot, il a fallu réfléchir avec les ITEC à la forme que nous voulions lui donner, à un choix intelligent de moteur correspondant à nos besoins. Nous avons dû prendre en compte une autonomie de batterie minimale. Pour ce faire, Martin a dû réaliser de nombreux calculs afin que la batterie soit adaptée au système. Dans cette troisième partie, vous allez pouvoir découvrir nos différentes étapes qui nous ont amenées aux choix définitifs des moteurs.

### ROBOT



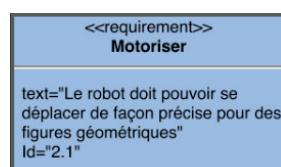
A

### ► MOTORISATION DU ROBOT

Pour motoriser le robot, nous avons répondu à l'exigence située à droite de ce document (*id="2.1"*). Comme elle le souligne, nous avons donc dû choisir des moteurs qui se déplacent de la manière la plus précise possible pour la réalisation des fonctions.

Deux hypothèses de types de moteur se sont posées à nous : soit le moteur pas à pas ou le motoréducteur avec encodeur pour calculer le nombre de tours. Nous avons comparé les caractéristiques de chacun d'entre eux afin de sélectionner celui qui nous convenait le mieux.

### ROBOT



**"NOTRE CHOIX S'EST PORTÉ SUR UN MOTEUR PAS À PAS."**

Le moteur pas à pas a une meilleure précision. Son procédé est basé sur la transformation d'une impulsion électrique en un mouvement angulaire. Contrairement au motoréducteur qui demande une variation entre la vitesse et le temps.



## MOTEUR PAS À PAS, bipolaire

400 pas par tour

Tension : 10 V

Couple : 0,107 N.m

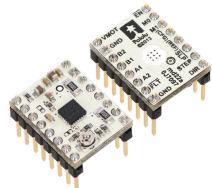
Pas de 0,9°

Poids : 120g

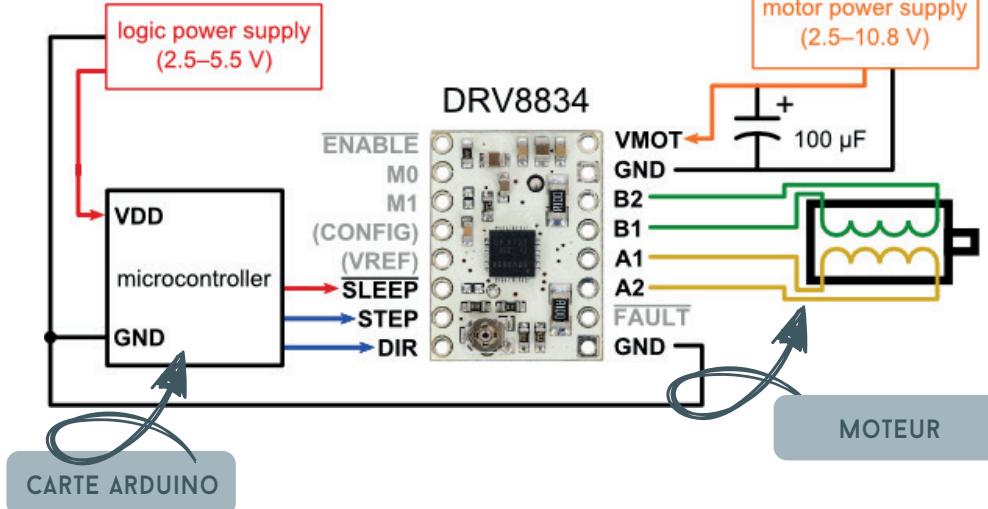
Une fois notre décision prise sur le type de moteur, nous avons étudié les différentes caractéristiques des moteurs pas à pas afin de sélectionner le moteur le plus adapté. Pour notre choix, les ITEC nous ont conseillé de prendre un couple de 0,107 N.m.

Notre choix s'est porté sur un moteur de 400 pas par tour avec un pas de 0,9° c'est-à-dire que  $400 \times 0,9 = 360^\circ = 1 \text{ tour}$ . Pour mieux comprendre, chaque impulsion électrique fera tourner le moteur de 0,9°.

Néammoins, pour faire fonctionner un moteur, il faut passer par l'intermédiaire d'un driver. C'est à lui qu'on va donner les consignes pour le piloter via la carte arduino nano. Il nous en a fallu deux, un pour chaque moteur.



*Voici le schéma de fonctionnement du driver fourni par le constructeur :*



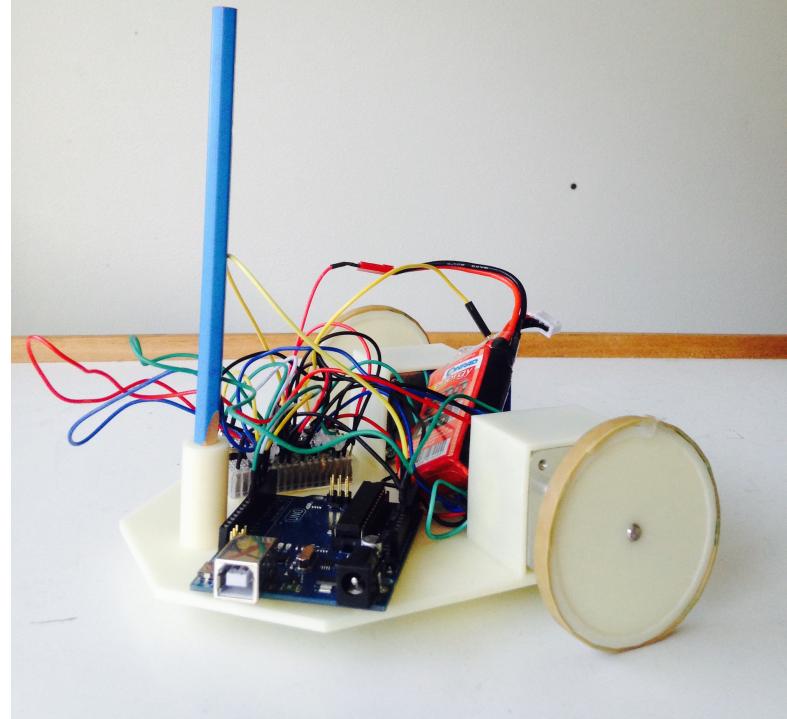
Chacune des trois fonctions a un rôle précis. Il est important de s'assurer de connaître leurs tâches pour écrire le code de programmation.

**DIR** : Cette première fonction va faire varier le sens de rotation du moteur par état logique 0 ou 1.

**STEP** : Cette fonction correspond au nombre de pas, une période vaut 1 pas.

**SLEEP** : Cette ultime fonction sert à mettre en veille le moteur pour économiser la batterie.

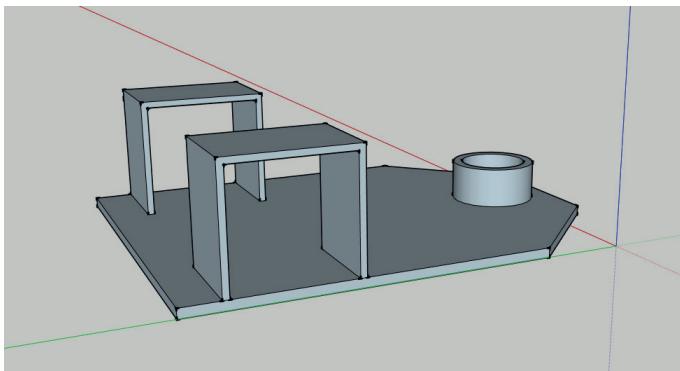
Premier prototype de notre robot. Dans la suite du projet, vous verrez que des circuits imprimés ont été élaboré pour éviter d'avoir de multiples fils sur le robot.



Nous avons adapté le driver à notre configuration. Sur le schéma, on retrouve notre moteur composé de deux bobines. La première sur les broches A1 et A2 et la seconde sur B1 et B2. Cela correspond aux quatre fils sortants de notre moteur.

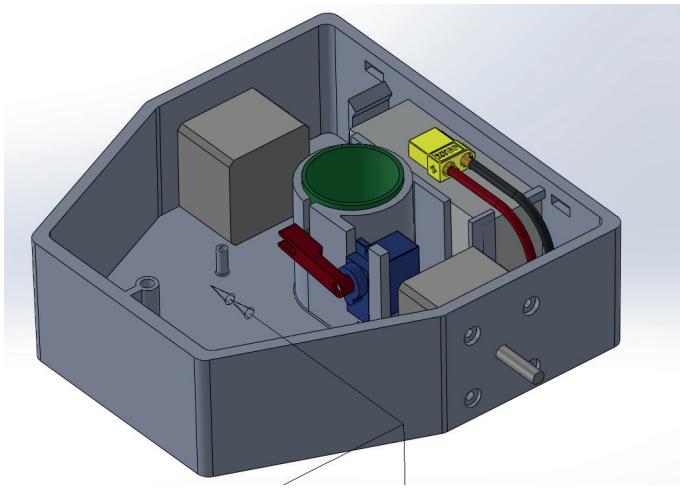
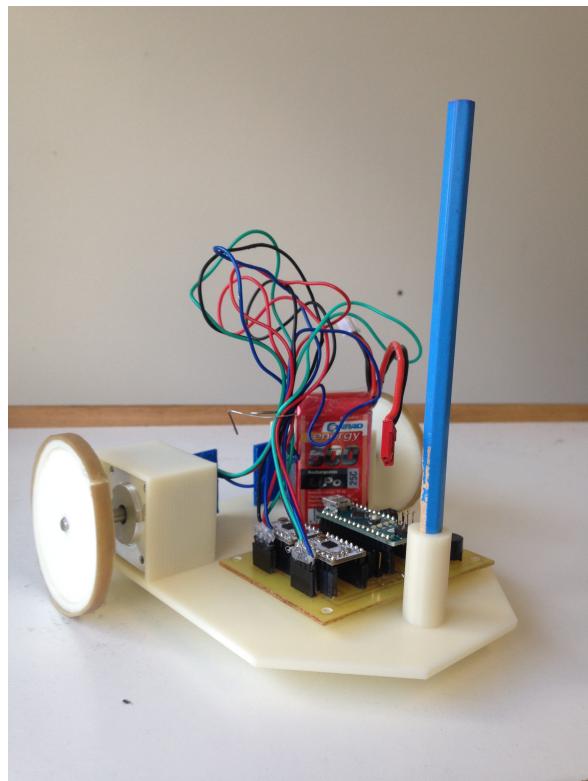
En haut à droite, sur les broches VMOT et GND, on branche notre batterie pour alimenter la carte arduino et le moteur.

Pour finir, en bas à gauche nous retrouvons notre microcontrôleur qui est notre carte arduino Nano. Trois fonctions vont être branchées à l'arduino (SLEEP, STEP et DIR). À préciser que la tension d'alimentation de la carte est reliée directement à la batterie, cela a été possible grâce au circuit imprimé du robot.



Travaillant en collaboration avec l'équipe d'ITEC qui conceptionne le robot et les différents blocs. Un des élèves de notre équipe à dû réaliser un prototype de robot pour commencer à le faire fonctionner en attendant l'arrivée du robot final. Notre prototype ne correspond pas au robot final car le crayon sera placé au centre du robot. Cependant, grâce au prototype, nous avons réussi à réaliser les premiers tests et les premières fonctions.

Comme vous pouvez le voir par rapport à l'image de la page précédente, le circuit imprimé est présent, avec cette fois-ci une arduino nano, les deux drivers, le module radio et la batterie. A signaler qu'un buzzer a été ajouté pour indiquer aux enfants que le robot est en fonctionnement.



A votre gauche se trouve une capture de la réalisation 3d finale du robot, fournie par les ITEC. Il a été conçu de façon à optimiser le moindre espace entre les éléments. Pour répondre aux exigences du robot, un servomoteur y a été installé de façon à changer le stylo facilement et lever le crayon lors de ses déplacements.

## ANNEXE PAGE 20

Retrouver le circuit imprimé du robot ainsi que son schéma structurel.

B

### LES FONCTIONS CRÉÉES

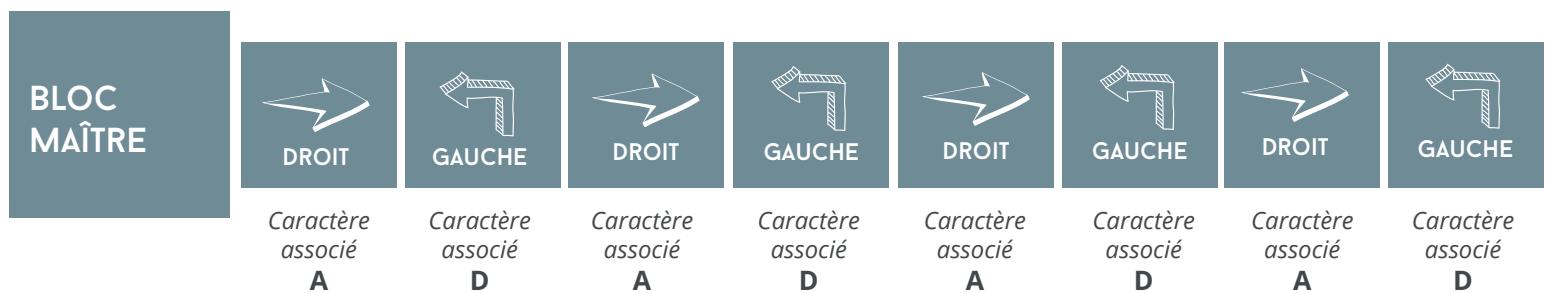
L'exigence id"3.5" nous informe qu'il faut que nous réalisions 16 fonctions. Nous les avons pensées de telle façon à ce qu'elles soient intéressantes pour un élève de 5 - 6 ans et réalisables par le robot.



Chaque fonction de 1 à 16 représente une figure géométrique, un mouvement ou un son. Les cinq premières fonctions dans le tableau à droite sont les plus évidentes. Chaque fonction est associée à un caractère, c'est ce caractère que nous allons transmettre des blocs programmables au robot en passant par le bloc maître.

FORME	LETTRE ASSOCIÉE
FONCTION 1	Avancer tout droit
FONCTION 2	Reculer
FONCTION 3	Tourner à droite
FONCTION 4	Tourner à gauche
FONCTION 5	Lever le crayon

Exemple pour la réalisation d'un carré. Une fonction clé en main a été conçue. Néanmoins, grâce aux fonctions de base (*tourner à droite, tourner à gauche, tout droit, reculer...*) nous pouvons également réaliser un carré.

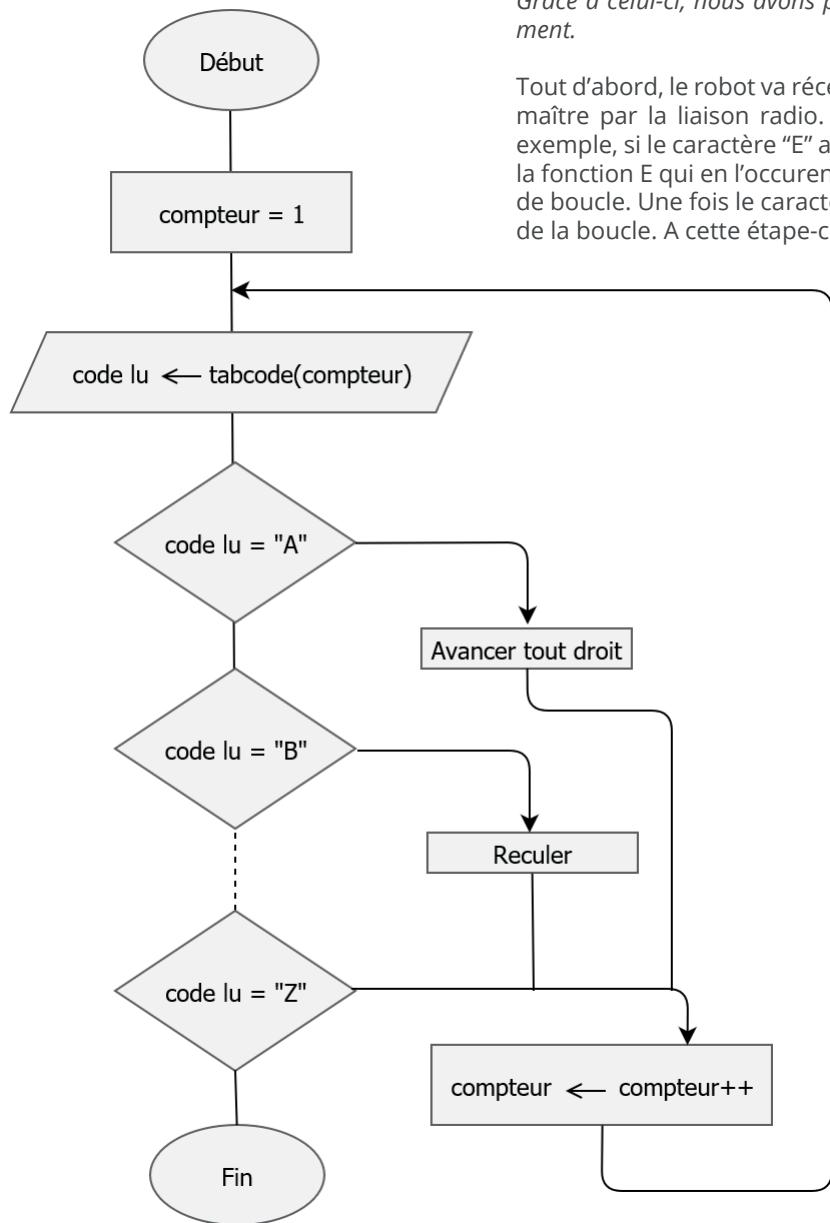


Le code va donc être "ADADADADZ". La lettre Z signifie la fin de la séquence. Pour rappel, les blocs communiquent grâce à une liaison RS-232 réalisée par Pierre.

C

## ALGORITHME DU ROBOT

Cet algorithme représente la suite d'instruction que le robot va exécuter. Grâce à celui-ci, nous avons pu créer le code de programmation plus facilement.

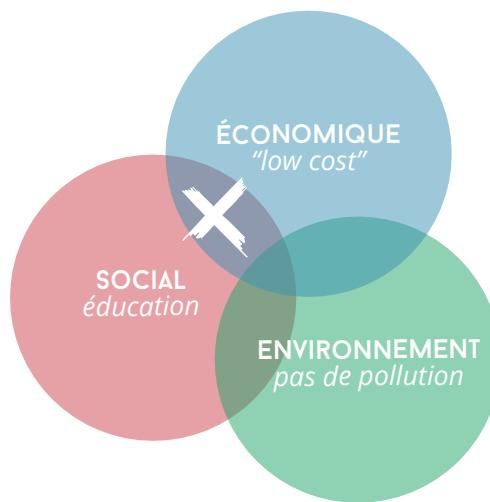


Tout d'abord, le robot va réceptionner les informations provenant du bloc maître par la liaison radio. Il va passer en revue chaque fonction. Par exemple, si le caractère "E" a été envoyé par le bloc maître, il va accomplir la fonction E qui en l'occurrence est le carré. Il va ensuite repartir en début de boucle. Une fois le caractère "Z" rencontré, cela signifie que c'est la fin de la boucle. A cette étape-ci, l'algorithme se termine.

# SYNTHÈSE

<<requirement>> Communiquer	<<requirement>> Robot
text="Le robot reçoit le programme du bloc maître." Id="2.4"	text="Le robot effectue le(s) déplacement(s) transmis par le bloc maître." Id="2"

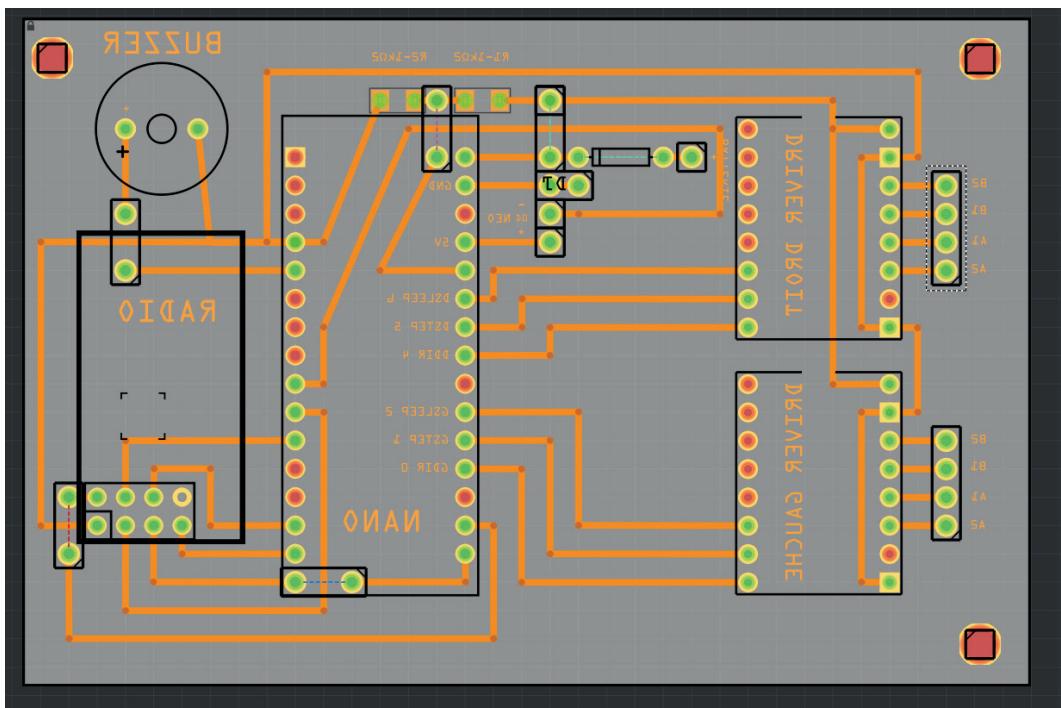
Dans ce projet, ma mission à été de répondre aux deux exigences ci-dessus. A la fin de ces 70 heures de projet, les exigences ont été remplies. Beaucoup de problèmes techniques sont survenus mais nous avons réussi à changer d'orientation rapidement pour éviter de perdre du temps. Notre projet pourra bientôt être mis en service auprès d'élèves.



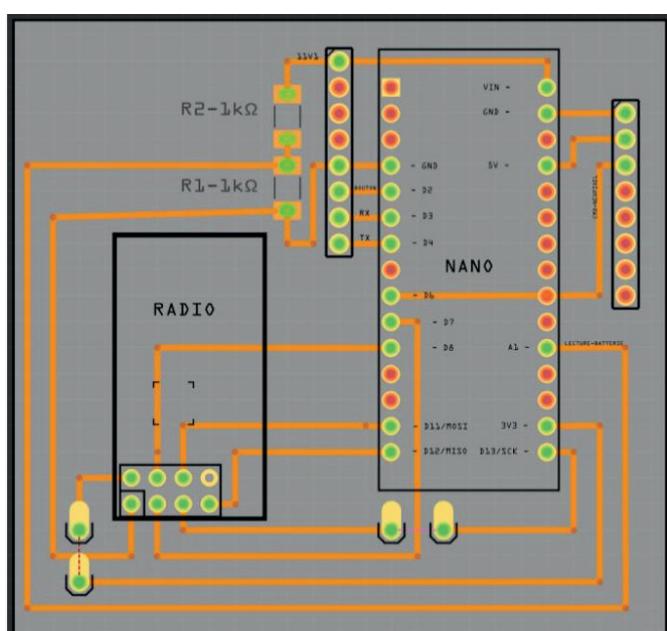
Pour nous, notre projet se classe dans l'axe équitable, entre le pilier économique et social. Effectivement, le pilier social est l'éducation étant donné que notre projet s'adresse à des élèves d'écoles primaires. Pour ce qui est de la question économique, nous voulions un système "low cost", c'est-à-dire de bonne qualité mais à un prix raisonnable de part le choix des matériaux. Nous avons choisi des matériaux et composants peu chers afin de faciliter l'accessibilité du système si un jour il doit être commercialisé. Pour terminer, le projet n'utilise pas d'énergie dite polluante mais une énergie électrique rechargeable. Le projet répond donc à une exigence de développement durable, même si, en effet les piliers sociaux et économiques sont bien plus importants.

Il existe diverses voies d'amélioration comme par exemple remplacer la carte arduino (qui est de base prévue pour le prototypage) par un microprocesseur plus petit et moins cher (par exemple la atmel128) ou encore gérer toutes les communications directement à l'aide d'un système wi-fi intégré.

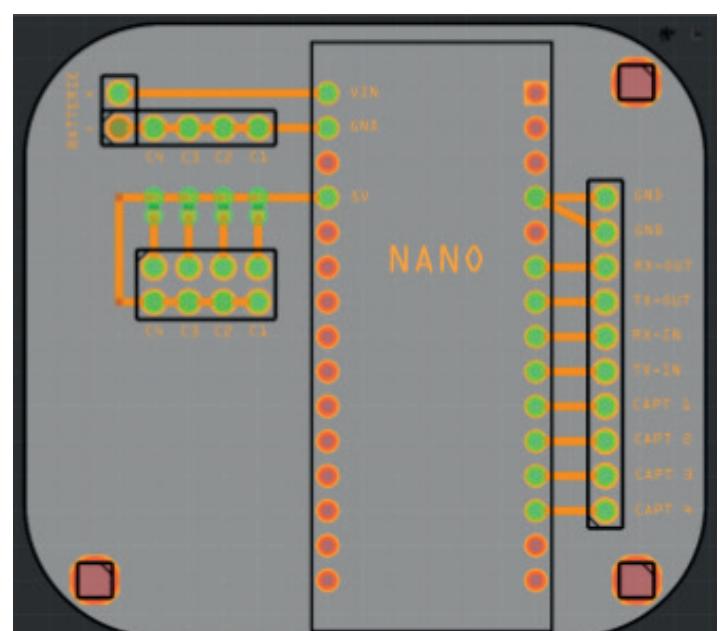
# ANNEXES



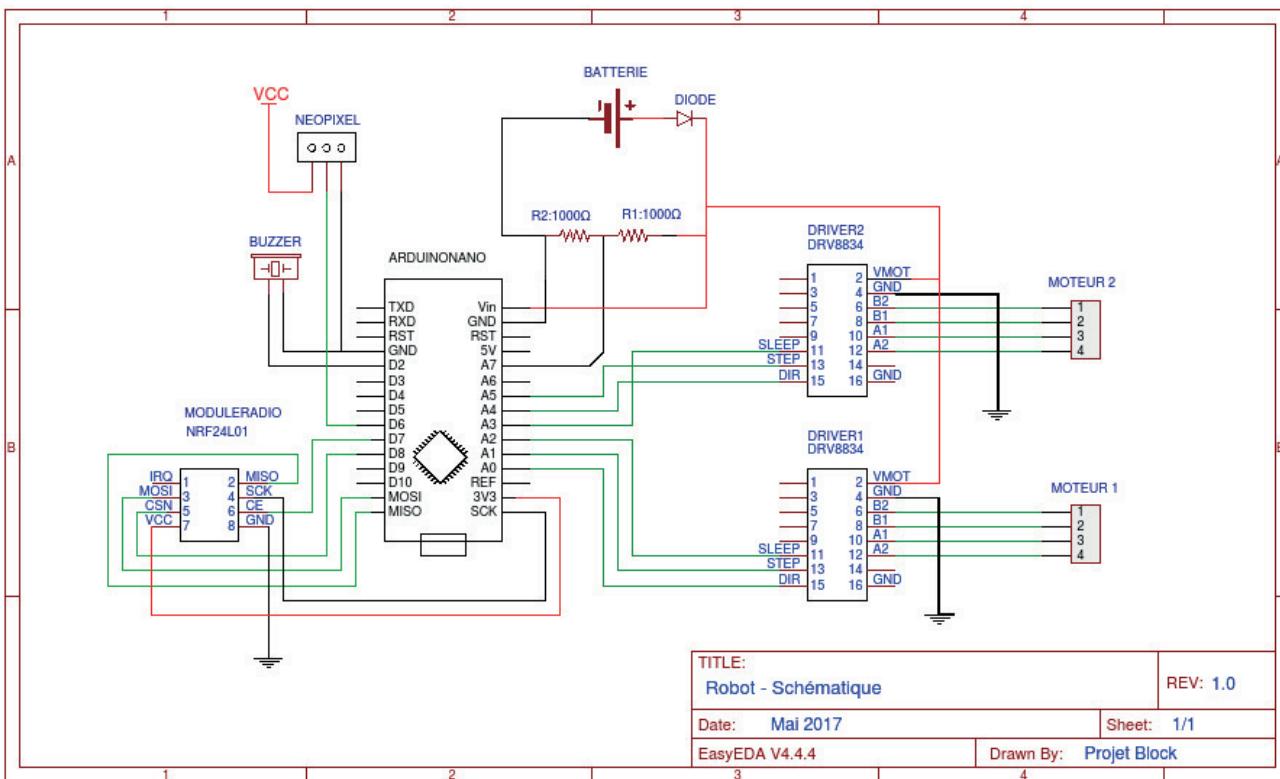
## ANNEXE 1



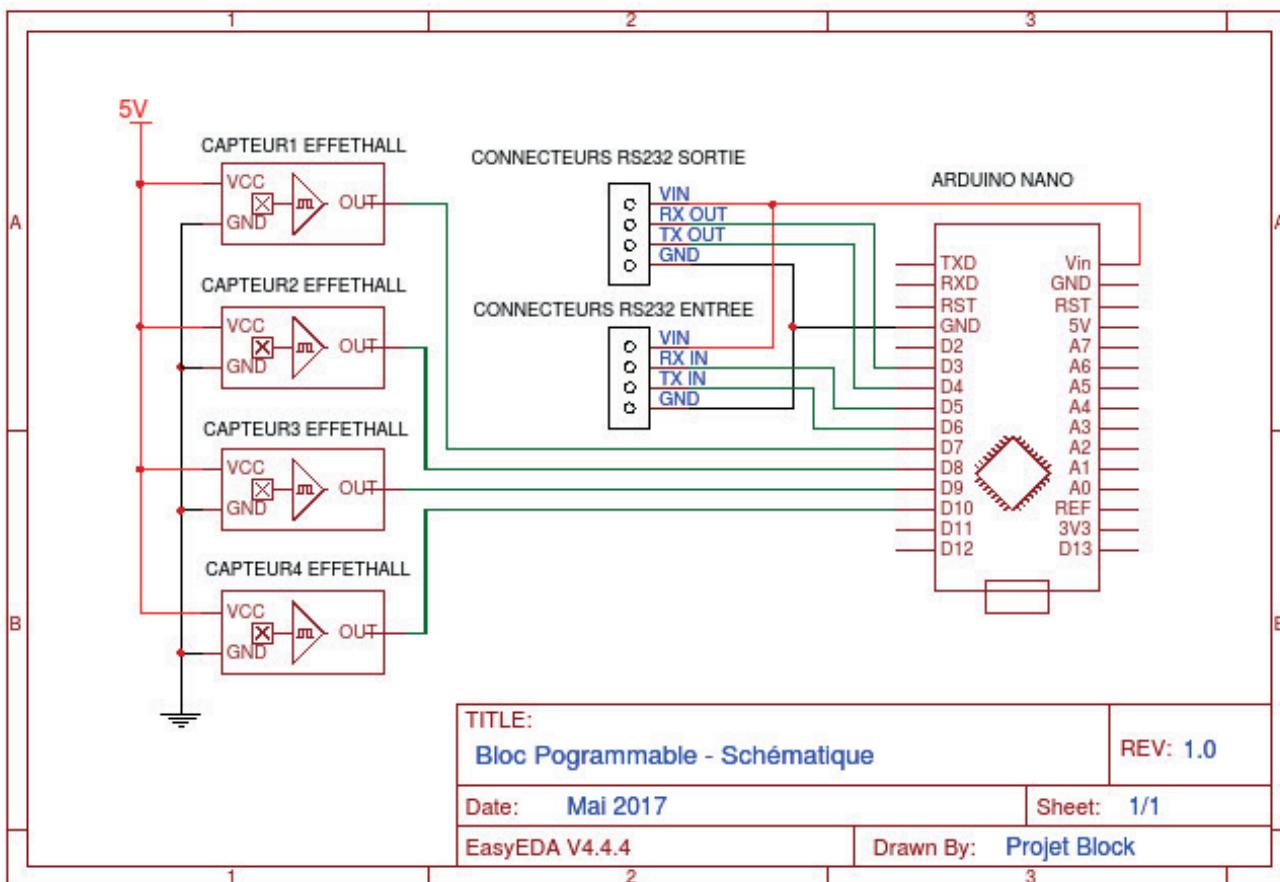
# ANNEXE 2



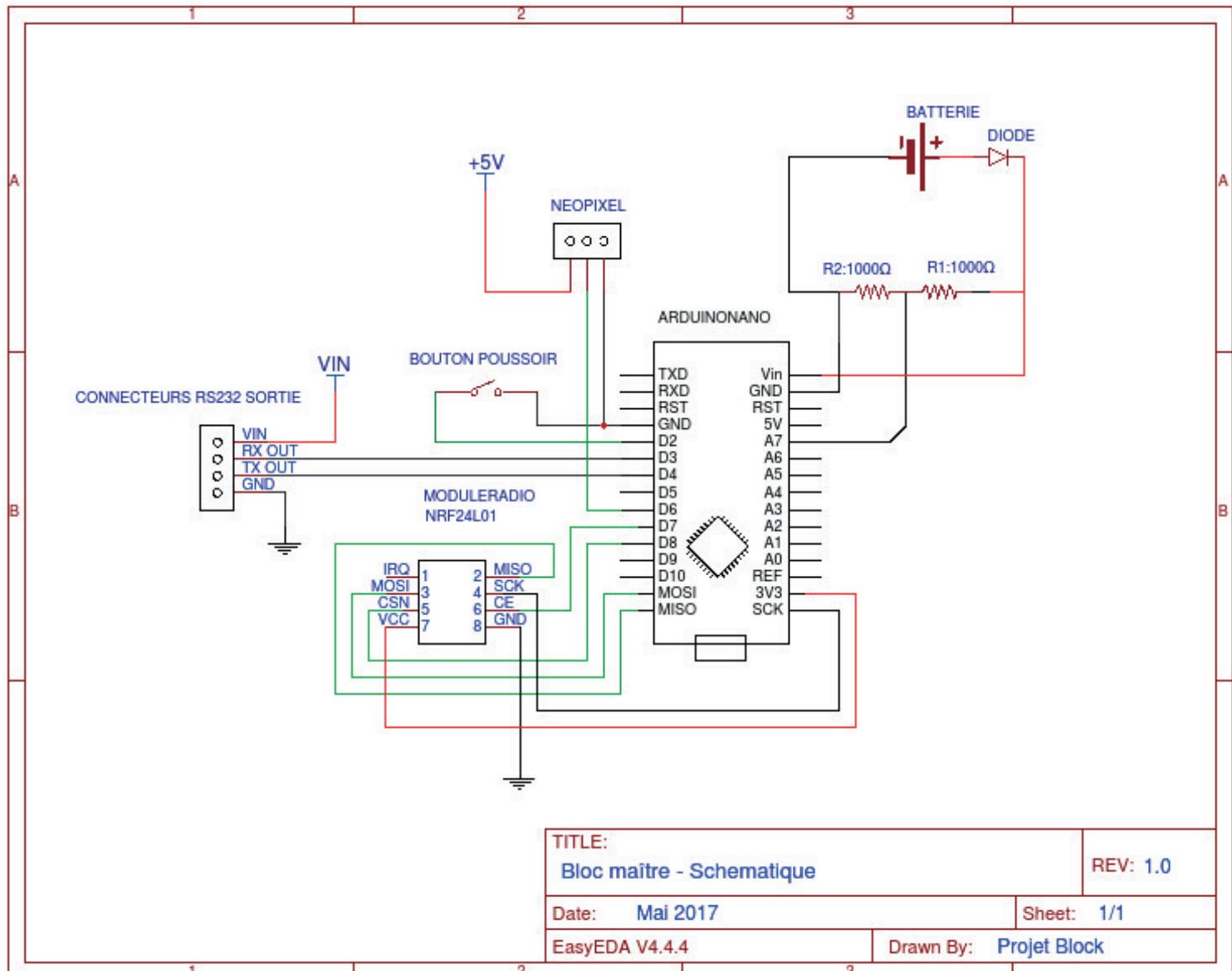
## **ANNEXE 3**



**ANNEXE 4**  
Schéma structurel du robot



**ANNEXE 5**  
Schéma structurel des blocs programmables



## **ANEXE**

Schéma structurel du bloc maître