

Présentation générale du projet

Le projet bloc consiste en la réalisation et la conception d'un jeu pédagogique destiné aux classes préparatoires (CP) et donc aux enfants âgés de 5/6 ans.

Son objectif est de développer la créativité et la réflexion de l'enfant sous forme de jeu à un prix relativement accessible.

Ce projet est réalisé en 70 heures en coopération avec un second groupe : l'un s'occupant de l'intelligence embarquée et de l'énergie, tandis que l'autre à en charge la conception.

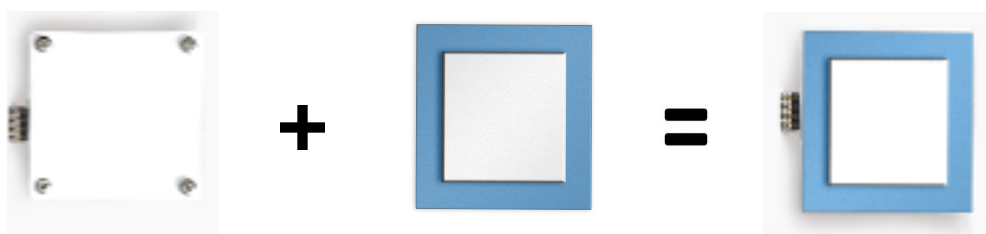
L'objectif final étant de déposer une licence afin de le protéger et potentiellement de le commercialiser.

Fonctionnement :

Le système est divisé en trois parties distinctes : le bloc maître, le bloc programmable et le robot :



Sur les blocs programmables se fixent des couvercles interchangeables, qui ont différentes formes et représentent différentes actions : par exemple, si le couvercle représente un carré, alors une fois fixé sur le bloc, l'action résultante du bloc sera : "faire un carré". Au total, 16 formes différentes ont été imaginées.



Les enfants vont créer une séquence en empilant les blocs les uns à la suite des autres à partir du bloc maître.


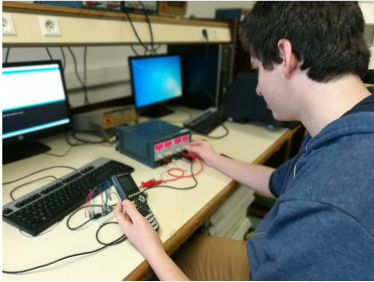
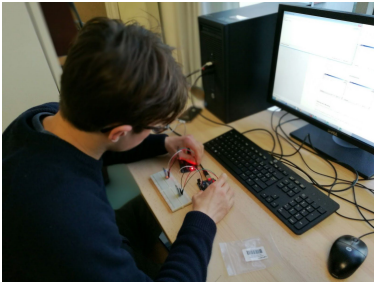



Un fois fait, ils vont appuyer sur le bouton central du bloc maître pour indiquer que la séquence est terminée.

Ce bloc va compiler les informations pour les communiquer au robot qui va reproduire les formes sur une feuille à l'aide d'un crayon accroché au préalable.

Répartition des tâches :

Le travail a été réparti de façon équitable et logique de façon à ce que le projet avance le plus vite possible et que chaque personne ait la même charge de travail.

| Répartition des tâches | | |
|--|--|---|
| Qui ? | Quoi ? | Pourquoi ? |
| Hugo DENIER  | <ul style="list-style-type: none"> -<u>Communication bloc maître - robot</u> -Fonctionnement des moteurs -Choix des constituants | <p>Hugo s'est occupé de la communication entre le bloc maître et le robot en réalisant plusieurs expérimentations pour savoir comment la réaliser.</p> <p>Il s'est également intéressé au fonctionnement des moteurs et donc au choix des drivers.</p> |
| Martin PORÉE  | <ul style="list-style-type: none"> -Calcul des batteries -<u>Détection de la batterie restante</u> -Maquettage d'un robot 3D | <p>Martin s'est occupé du choix de la batterie pour tenir une demi-heure et a réalisé un programme permettant de savoir le niveau de la batterie pour prévenir de la recharger.</p> <p>Il a également réalisé un robot imprimé à l'imprimante 3D afin de pouvoir effectuer des tests.</p> |
| Sullivan LE TOUZIC  | <ul style="list-style-type: none"> -<u>Détection du couvercle</u> -Réalisation des circuits imprimés | <p>Sullivan a réalisé un programme pour la détection du couvercle pour que le bloc programmable puisse connaître sa fonction. Afin de proposer un projet fini il a aussi réalisé des circuits imprimés pour chaque sous système.</p> |
| Pierre NICOLAS  | <ul style="list-style-type: none"> -<u>Communication entre les blocs</u> -Programme coeur : robot / bloc maître / bloc programmable -Réalisation des circuits imprimés -Choix des constituants | <p>En plus de la communication entre les différents blocs, je me suis chargé de réaliser le programme finale de chaque sous système.</p> <p>J'ai aidé au choix des constituants et à la réalisation des circuits imprimés.</p> |

Puisque les blocs sont destinés à être assemblés, une communication physique semble être adaptée.

La communication permettra de connaître : le nombre de blocs, leur ordre et leurs fonctions.

Ainsi deux types de communications ont été retenues afin d'assurer ce rôle :

La communication I2C (cf. n°1 p9)

La communication RS232 TTL (cf. n°2 p9)

Le choix :

-On remarque que la liaison I2C nécessite un fils de moins que la liaison RS232, cependant cela n'est pas important car l'alimentation électrique empruntera la connexion : il y aura donc un + et un - à rajouter. Or la masse (-) existe déjà sur la liaison RS232. Il n'est donc pas nécessaire d'en rajouter une.

-Cependant, la liaison I2C nécessite quelque chose que l'on ne peut pas avoir. En effet, la liaison I2C fonctionne sur adresse. Chaque bloc à la programmation se verrait définir une adresse différente.

Si je veux parler au bloc 2, je dois appeler le bloc 2. Or comment savoir où est le bloc 2 et s'il a bien été utilisé ? Cela n'est pas possible.

La communication RS232 TTL a donc été retenue, d'une part car elle répond au besoins de l'exigence qui était d'assurer une communication avec n'importe quel bloc mais aussi car il s'agit d'une communication que l'on connaît et que l'on a déjà utilisé en cours et en travaux pratique.

Comment s'y prendre ? :

La liaison RS232 TTL peut se faire de deux manières sur Arduino (carte de programmation cf. ci-dessous) :

-De manière matériel avec une puce intégrée à Arduino (cf. encadré ci-dessous) qui peut gérer la communication en même temps qu'en exécutant le programme sur seulement deux broches.

-De manière logiciel en n'utilisant n'importe quelle broches à l'aide d'une bibliothèque intégrée.

L'avantage de l'un par rapport à l'autre est que la liaison matérielle peut fonctionner en même temps qu'en faisant tourner le programme, cependant elle ne fonctionne que sur deux broches et notre montage en demande quatre. Nous devons donc choisir la manière logicielle

Quoiqu'il en soit la liaison RS232 TTL traduit le message de l'utilisateur en trame électrique binaire sur un octet interprétable par les deux cartes : 0 volt pour 0 (état logique bas) et 5 volts pour 1 (état logique haut) à la place de +12 volts et -12 volts pour la liaison RS232 de base.



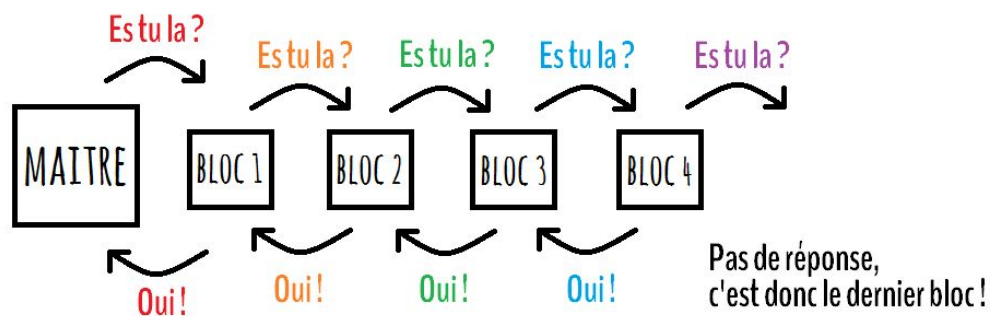
Conception détaillé

DE FÉVRIER À FIN MARS

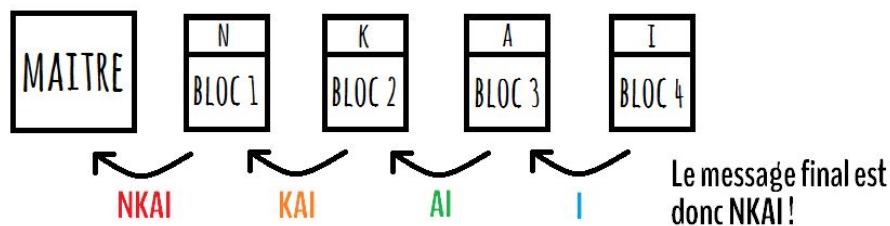
La stratégie :

Une stratégie a été imaginé en deux étapes afin de réaliser la communication :

La première est de compter les blocs. L'informatique n'a pas d'oeil, ou du moins notre système n'en a pas. Pour savoir le nombre de bloc le système va jouer au "téléphone arabe", un message va passer de bloc en bloc à partir du bloc maître jusqu'à arriver au dernier bloc qui n'aura plus personne à qui parler. Pour savoir s'il y a un bloc "2" à côté de lui, le bloc "1" va demander s'il y a quelqu'un à côté de lui. S'il y a un bloc "2" alors la réponse sera "oui" sinon aucune réponse ne sera reçu. Ce bloc sera alors le dernier bloc.



La seconde étape consiste à connaître la fonction de chaque bloc. Le "téléphone arabe" va repartir dans l'autre sens, cette fois ci, du dernier bloc au bloc maître. Chaque bloc va répéter à un autre la séquence qu'on lui aura chuchoté en y ajoutant sa propre fonction (traduit en lettre). De bouche à oreille, l'information finira par arriver au bloc maître qui aura alors toute les informations nécessaire.



Algorigrammes :

Le fonctionnement du bloc maître et celui du bloc programmable est légèrement différent. Les deux étapes de chaque bloc ont été représenté en quatre algorigrammes.

Etape 1 - Bloc maître (cf. p13):

Le schéma structurel du bloc maître se situe en annexe p23.

La communication se déclenche lors de l'appui sur le bouton central du bloc maître par l'enfant. Cela signifie que la séquence est terminée d'être construite. Dès lors le bloc maître envoie la lettre "Z" qui est le code simplifié de "es tu là ?" en attendant la réponse "Y" qui est le code simplifié de "Oui !" (cf. schéma 1 p11). Cela signifie qu'il y a bien un bloc à côté. L'étape 1 est terminée et on peut passer à l'étape 2. Cependant si l'on ne reçoit rien, cela signifie qu'il n'y a pas de bloc à côté et donc que le bloc maître est seul. L'appui sur le bouton est sûrement une erreur, on redémarre la carte.

Etape 1 - Bloc programmable (cf. p13):

Le schéma structurel du bloc programmable se situe en annexe p22.

Le bloc programmable attend qu'on le contacte sur son entrée, c'est à dire par le bloc situé avant lui. Il attend la fameuse question ("Z" -> "Es tu là?") pour y répondre ("Y" -> "Oui !"), avant d'effectuer à son tour la question au bloc situé après lui. S'il reçoit la réponse "Y" cela signifie qu'il y a bien un bloc après lui. L'étape 1 est donc terminée avec la première variante. Cependant s'il ne reçoit rien cela signifie qu'il est le dernier bloc de la séquence. L'étape 1 est donc terminée avec la seconde variante.

Etape 2 - Bloc programmable (cf. p13):

Le programme est adapté aux deux variantes et attend juste de savoir laquelle effectuer. Cette décision est prise lors de l'étape 1.

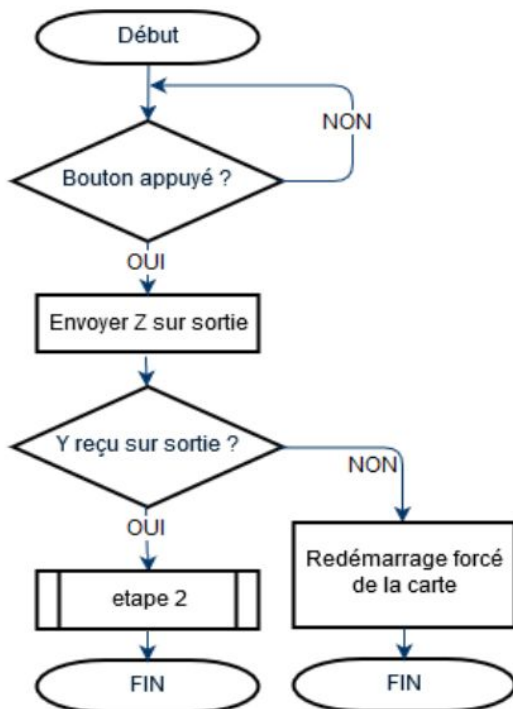
Variante 1 : Lors de la variante 1, cela signifiait que le bloc n'était pas le dernier de la séquence. Il attend donc de recevoir les informations du bloc situé après lui, pour les stocker tant qu'il en reçoit encore puis de les renvoyer au bloc situé avant lui, sans oublier d'ajouter sa propre fonction (cf. schéma 2 p11). Son rôle est à présent terminé et la carte redémarre afin de se réinitialiser.

Variante 2 : Lors de la variante 2, cela signifiait que le bloc était le dernier de la séquence. C'est donc à lui de commencer à envoyer sa fonction au bloc situé avant lui. Il ne recevra pas d'information car il n'y a pas de blocs situés après lui. Son rôle est à présent terminé et la carte redémarre afin de se réinitialiser.

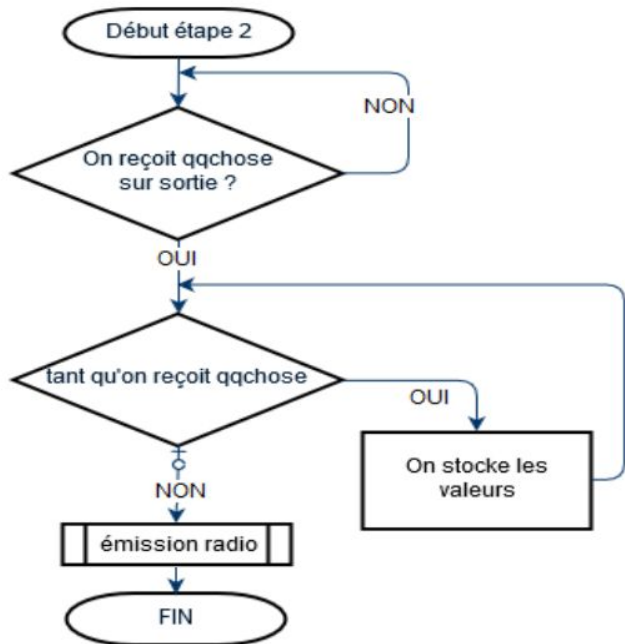
Etape 2 - Bloc maître (cf. p13):

Le bloc maître est dans l'attente de recevoir l'information finale, celle combinant les fonctions de tous les blocs réunis. Il va stocker les valeurs tant qu'il en reçoit puis une fois fait pourra commencer l'émission radio afin de lui transmettre les informations..

ETAPE 1 - BLOC MAITRE

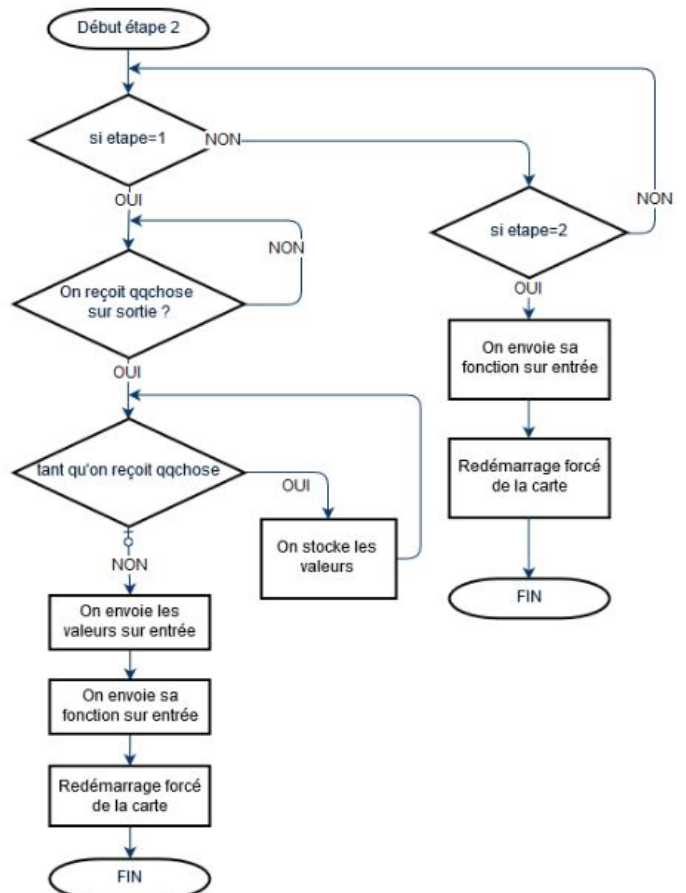
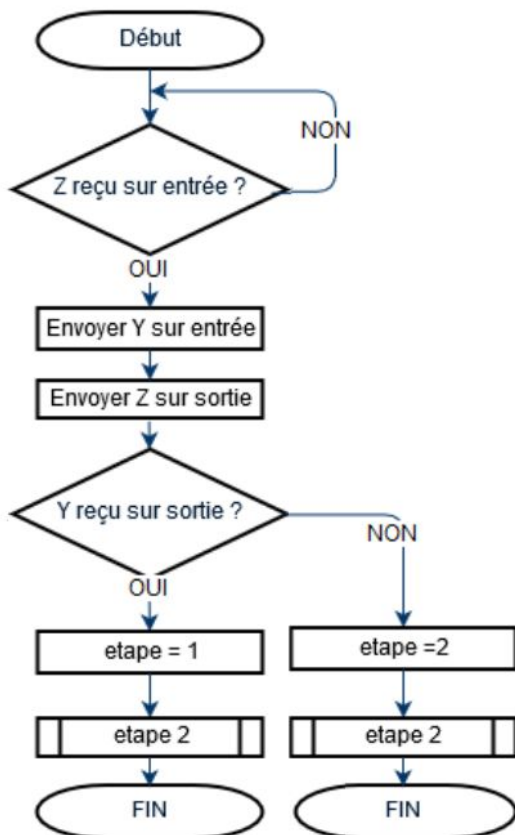


ETAPE 2 - BLOC MAITRE



ETAPE 2 - BLOC PROGRAMME

ETAPE 1 - BLOC PROGRAMME



Prototypage et maquettage

D'AVRIL A LA FIN

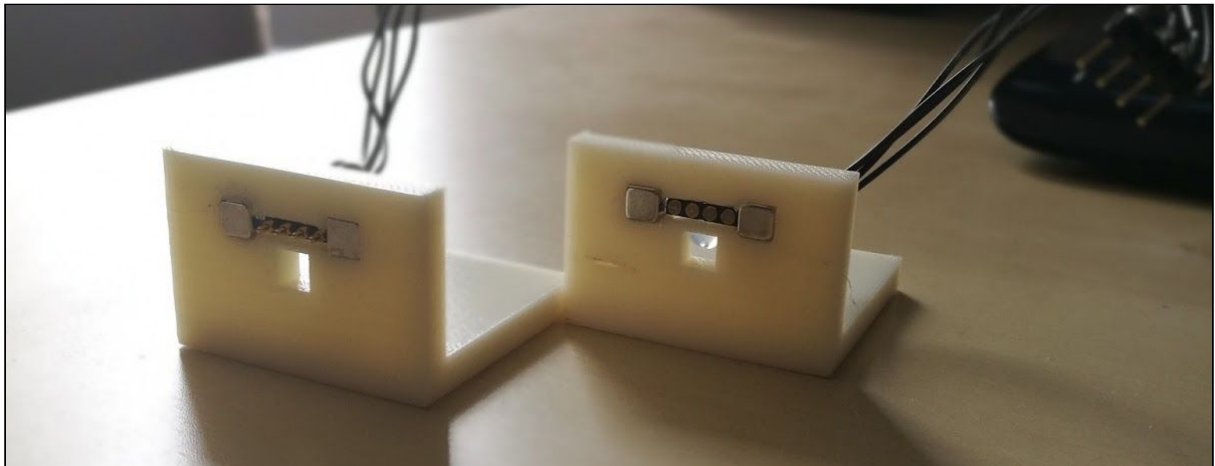
Le prototypage :



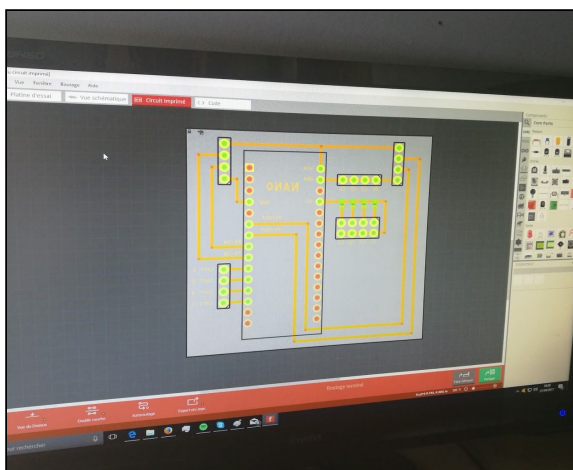
Chaque liaison s'effectuera par un contact physique entre les deux blocs. Il y a donc deux faces possibles : Une face dite mâle(cf. ci-dessous à gauche) et une face dite femelle(cf. ci-dessous à droite).

Les aimants permettront de maintenir la connexion en place et donnera un côté attrayant au jouet.

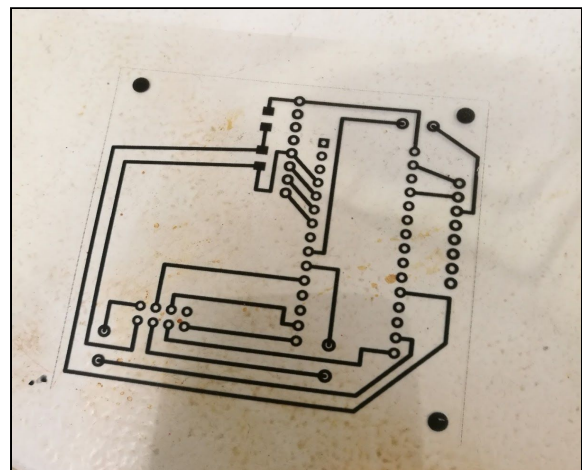
Le bloc maître sera muni d'une face mâle et les blocs programmables d'une face mâle et femelle (voir ci-dessus).



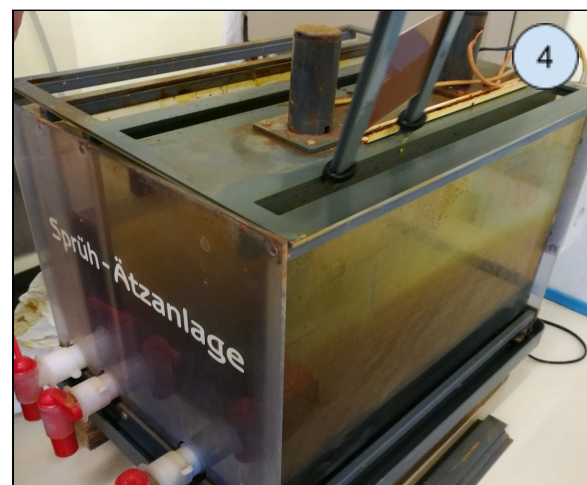
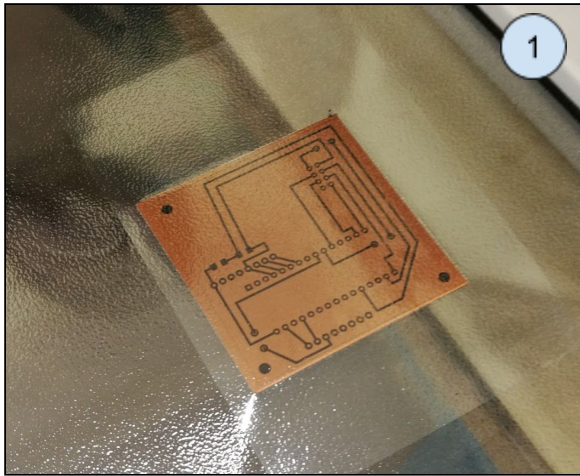
Les connecteurs seront connecté à la carte. Un circuit imprimé à alors été envisagé afin de concrétiser le projet, de faciliter le câblage final et d'optimiser la robustesse du bloc (voir schéma en annexe p20).



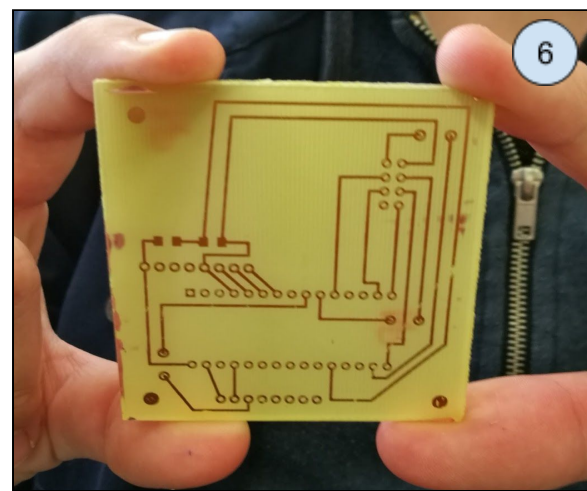
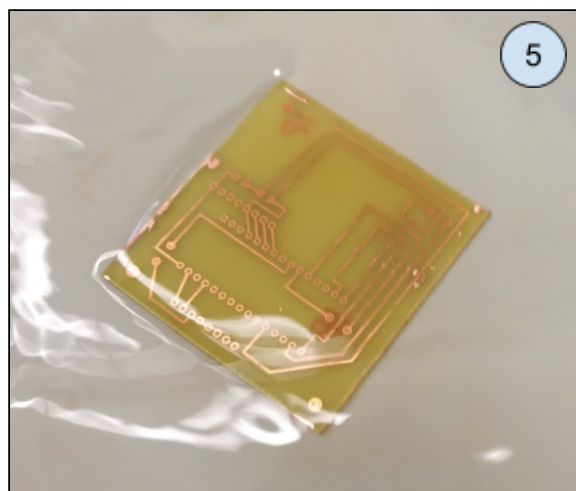
Conception sur ordinateur avec Fritzing

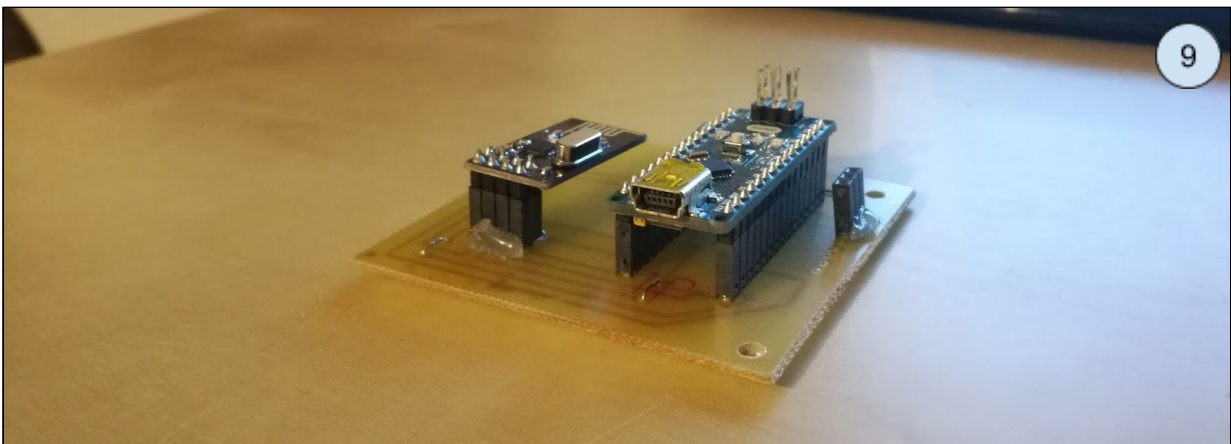


Impression à échelle 1:1 sur transparent



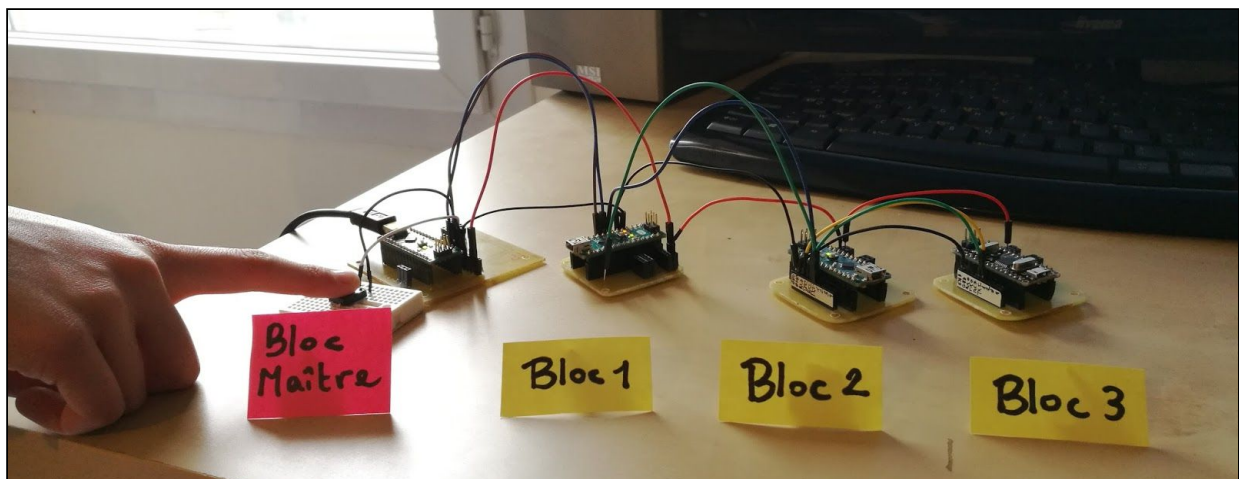
L'impression des pistes se fait sur une plaque d'époxy qui est un matériau composé d'une couche de cuivre recouvert d'un revêtement résineux très sensible au UV. La préparation est simple, il suffit de placer le transparent sur la plaque du côté que l'on souhaite faire apparaître(1) puis de l'isoler dans une machine générant des UV pendant une durée d'environ 3 minutes(2). Il faut maintenant "révéler" la plaque dans une base forte, c'est à dire faire apparaître les pistes(3) avant de la plonger dans du perchloreure de fer qui va ronger le cuivre(4). Un dernier bain dans l'acétone permettra d'enlever les restes de résine(5) afin de faire apparaître le résultat final(6).





La plaque est ensuite percée minutieusement à la perceuse avec un foret de 0.8mm(7) afin de pouvoir y souder à l'étain(8) tous les composants nécessaire au bon fonctionnement du système(9).

Une fois tous les composants fixés, les circuits imprimés sont terminés. Il peuvent à présent être fixés dans les blocs prévus par le second groupe. Cependant, pour les tests, les blocs ne sont pas nécessaires



Les tests :

Le logiciel arduino embarque un moniteur série qui permet d'interagir avec son programme, notamment afin de surveiller son déroulement. Des lignes de codes ont donc été insérés dans le programme afin de faire apparaître à chaque étape importante, du texte sur le moniteur série permettant ainsi de se situer dans l'exécution du programme.

Ordre de communication (voir p18):

Etape 1 :

B.M : *"Attente de l'appuie sur le bouton"*: Le bloc maître attend que l'utilisateur appuie sur le bouton.
B.M : *"Envoie de la lettre Z"*: le bouton a été appuyé, la lettre Z a été envoyée au bloc d' à côté (bloc 1).
B.1 : *"On ne reçoit rien"*: Le Z n'a pas encore été reçu.
B.1 : *"Lecture de la lettre Z"*: La lettre Z a été reçue du bloc précédent pour savoir si un bloc est là.
B.1 : *"Réponse par la lettre Y"*: Le bloc est là, on répond par Y pour signaler la présence.
B.M : *"Lecture de la lettre Y"*: Un bloc est à côté, en attente.
B.1 : *"Envoie de la lettre Z"*: Le bouton a été appuyé, la lettre Z a été envoyée au bloc d' à côté (bloc 2).
// même opération jusqu' à arriver au bloc 3
B.3 : *"Envoie de la lettre Z"*: Le bouton a été appuyé, la lettre Z a été envoyée au bloc d' à côté (bloc 4).
B.3 : *"Pas de réponse"*: Cela signifie qu'aucun bloc n'est situé à côté.

Etape 2 :

B.3 : *"Etat capteur"*: On regarde sa fonction (voir tâche de Sullivan).
B.3 : *"Envoie de ma fonction"*: On envoie sa fonction traduite par une lettre comprise par tous les blocs.
B.3 : *"Redémarrage du bloc"*: Le bloc a fini son rôle il peut redémarrer afin de se réinitialiser
// voir bloc 1
B.1 : *"On reçoit quelque chose"*: On reçoit les fonctions du bloc précédent.
B.1 : *"Etat capteur"*: On regarde sa fonction (voir tâche de Sullivan).
B.1 : *"Envoie du caractère du bloc"*: On envoie la fonction des autres blocs.
B.1 : *"Envoie de ma fonction"*: On envoie sa fonction traduite par une lettre comprise par tous les blocs.
B.1 : *"Redémarrage du bloc"*: Le bloc a fini son rôle il peut redémarrer afin de se réinitialiser
// même opération jusqu' à arriver au bloc maître qui relaie l'information au robot

Légende : B.M = Bloc maître ; B.1 = Bloc 1 ; B.2 = Bloc 2 ; B.3 = Bloc 3

