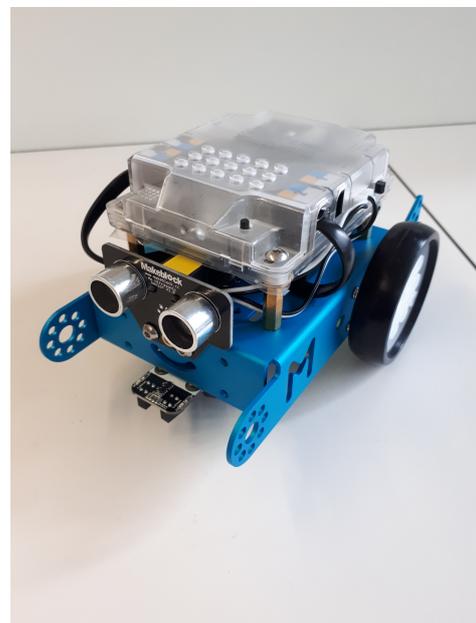
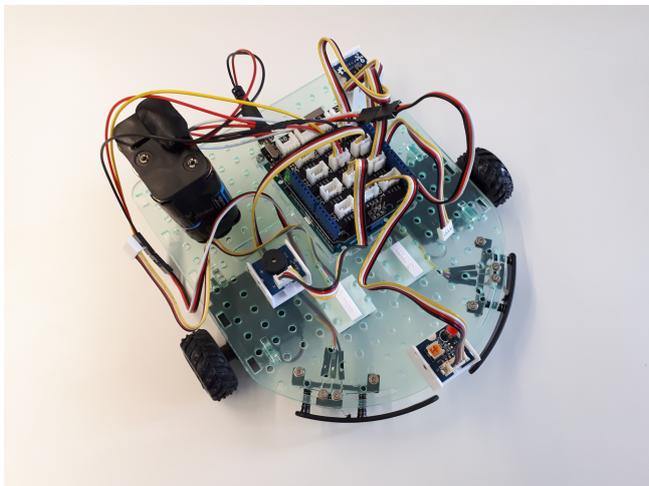


Découverte de la robotique, assemblage et programmation



Étudiants :

Zoé BEIQUE

Marie LANTERNIER

Clara HOOSE

Hortense PHILIPPE

Lise KLEIBER

Mathias VAN AUDENHOVE

Enseignant-responsable du projet :

Faouzi DHAOUADI

Date de remise du rapport : **17/06/2019**

Référence du projet : **STPI/P6/2019 – 23**

Intitulé du projet : **Découverte de la robotique, assemblage et programmation**

Type de projet : **expérimental**

Objectifs du projet :

Ce projet nous permettra d'apprendre à monter un robot ainsi qu'à souder pour le rendre fonctionnel. L'objectif ici est de savoir construire un robot.

Par ailleurs, nous avons pour but d'appliquer nos acquis en termes de programmation afin de modifier les programmes sur la carte mère d'un robot pour adapter son comportement à nos besoins.

Enfin, ce projet nous permettra également de nous accoutumer à une vision plus réaliste : savoir maîtriser un budget en sélectionnant les robots et les composants répondant à nos critères.

Mots-clefs du projet : **robots, soudure, programmation et montage**

TABLE DES MATIÈRES

1. Introduction.....	6
2. Méthodologie / Organisation du travail.....	7
3. Travail réalisé et résultats.....	8
3.1. Robot Arduino.....	8
3.1.1. Présentation et objectifs du Robot 1.....	8
3.1.2. Problèmes rencontrés.....	9
3.1.3. Programmation Carte Arduino Uno.....	10
3.2. Robot mBot Education.....	12
3.2.1. Présentation et objectifs du Robot 2.....	12
3.2.2. Problèmes rencontrés.....	12
3.2.3. Phase de test du robot 2.....	13
3.2.4. Deuxième version du Robot 2.....	14
3.2.5. Troisième version du Robot 2.....	16
4. Conclusions et perspectives.....	18
5. Bibliographie.....	19
6. Annexes.....	20
6.1. Documentation technique.....	20
6.2. Listings des programmes réalisés.....	21
6.3. Notices de montage.....	22
6.4. Parcours réalisés.....	23

REMERCIEMENTS

Nous tenions à remercier M.DHAOUADI de nous avoir aidé à mener à bien ce projet. Nous tenions également à remercier les enseignants responsables de cet EC P6, car grâce à eux nous avons pu acquérir de nouvelles connaissances sur un sujet qui nous intéressait particulièrement.

1. INTRODUCTION

Lors de notre formation STPI, nous sommes amenés à réaliser un projet scientifique afin de nous perfectionner dans un domaine ou dans notre cas, d'en découvrir un nouveau. De plus, ce projet est une nouvelle occasion pour travailler en équipe. Il permet également d'acquérir de la rigueur de par la gestion du planning et le respect des dates butoirs.

Le sujet "Découverte de la robotique, assemblage et programmation" a été le premier choix des six membres du groupe. Nous trouvons effectivement intéressant d'explorer l'aspect manuel de la robotique, ainsi que d'appliquer nos connaissances en programmation de manière ludique. Pour cela, une heure et demie nous ont été allouées, sans compter le travail personnel fourni en plus de ce créneau.

Notre projet consiste tout d'abord à sélectionner les robots à étudier, en respectant certains critères. Notre objectif est d'étudier deux robots. Le premier nous permettra de nous familiariser avec la construction et nous servira à nous entraîner à la programmation. Notre robot final, complètement différent, aura pour objectif de mettre en application nos acquis et de nous attarder sur les options possibles.

2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Comme indiqué précédemment, nous avons une heure et demie par semaine encadrées par M. Dhaouadi. En complément, nous nous sommes réunis en dehors de ces heures et nous avons travaillé en autonomie chacun de notre côté.

La répartition des tâches s'est effectuée naturellement et de façon équitable au sein du groupe. Par exemple, nous ne pouvions pas nous retrouver à six sur un même poste. Malgré la division du groupe pour un souci d'efficacité, nous réservions du temps à la fin de chaque séance pour récapituler l'avancée de chaque groupe, ainsi que pour prévoir les tâches futures.

Ci-dessous, se trouve le tableau récapitulatif de la répartition des tâches sur l'ensemble des séances de projet.

	Zoé	Clara	Lise	Marie	Hortense	Mathias
Soudure et montage Robot 1	✓	✓	✓	✓	✓	✓
Programmation Robot 1			✓	✓		
Test 1 ^{ère} version Robot 2	✓	✓			✓	✓
Programmation et test 2 ^{ème} version Robot 2		✓		✓		✓
Programmation et test 3 ^{ème} version Robot 2	✓		✓		✓	
Rédaction du rapport	✓	✓			✓	
Élaboration du poster	✓					
Diaporama de la soutenance			✓	✓		✓
Préparation de l'oral	✓	✓	✓	✓	✓	✓

L'attribution des tâches n'était pas figée, malgré des implications plus marquées dans certaines opérations, chacun de nous a pu participer aux différents pôles.

3. TRAVAIL RÉALISÉ ET RÉSULTATS

3.1. Robot Arduino

3.1.1. Présentation et objectifs du Robot 1

Nous avons choisi pour notre premier robot le Robot Arduino Evolutif (réf. UNOEVO) sous les conseils de notre enseignant responsable du projet. Ayant un budget et des caractéristiques à respecter, nous n'avons pas trouvé un modèle aussi intéressant que celui du projet de l'an passé, c'est pourquoi nous avons repris le même.

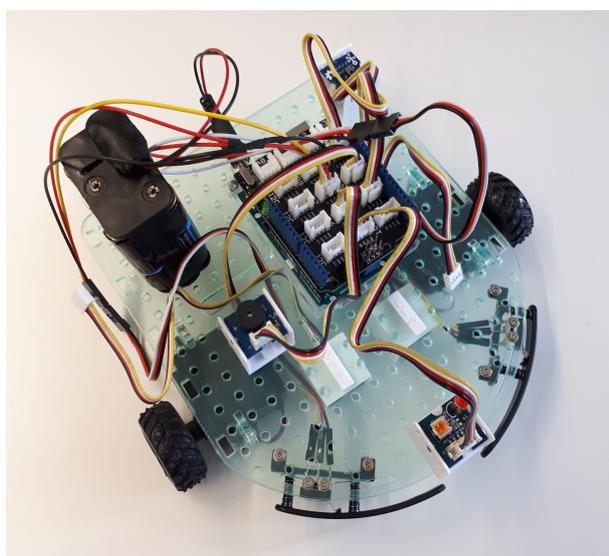


Illustration 1: Vue du dessus du Robot 1

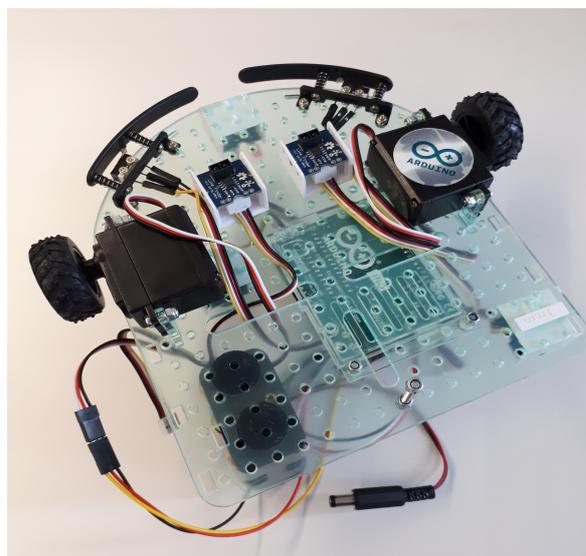


Illustration 2: Vue du dessous du Robot 1

Concernant ce modèle, les objectifs étaient multiples. Le montage de ce premier robot devait nous apprendre comment souder, en effet la réalisation minutieuse de cette étape est essentielle. La technique à adopter demande de l'entraînement. Nous nous sommes donc tout d'abord exercés à souder et à dessouder sur une plaquette de circuit imprimé, afin de maîtriser le geste avant de l'appliquer au robot. Puis, à l'aide de la notice, nous avons dû assembler étape par étape les différentes pièces du robot afin d'obtenir la version présentée sur les figures 1 et 2 ci-dessus.

Le dernier objectif de cette version était de se familiariser avec le logiciel de simulation mBlock dans le but de programmer par la suite la carte Arduino Uno. Cela permettra au robot d'effectuer les actions souhaitées.

3.1.2. *Problèmes rencontrés*

Lors des premières séances, nous avons rencontré plusieurs problèmes. Premièrement, concernant le matériel, les outils qui nous ont été prêtés étaient parfois mal adaptés. En effet, nous n'avions pas à notre disposition de petits tournevis et le matériel de soudure était dans un premier temps de faible efficacité. Autrement dit, l'appareil manquait de puissance et le fil d'étain était trop fin pour nos besoins. Lors de la soudure des composants, nous avons eu quelques difficultés à maîtriser le geste afin d'obtenir un résultat soigné. De plus, malgré nos efforts, une pièce s'est détachée lors du montage. Nous avons donc été contraints de ressouder ce composant et donc de démonter une partie du robot.

Par la suite, nous avons été confrontés à certaines incohérences entre les pièces et la notice de montage. Pour les pièces switch (interrupteur) et suiveurs de ligne, le support blanc était trop étroit pour laisser passer les fils qui alimentent les composants (figure 3 ci-dessous). Le plus simple était de mettre ces deux pièces à l'envers mais cela aurait empêché les suiveurs de lignes de fonctionner correctement et l'interrupteur n'aurait pas été accessible. Pour y remédier, nous avons plié les fils au risque de les abîmer et nous avons fixé les supports différemment sur la base de plexiglas.

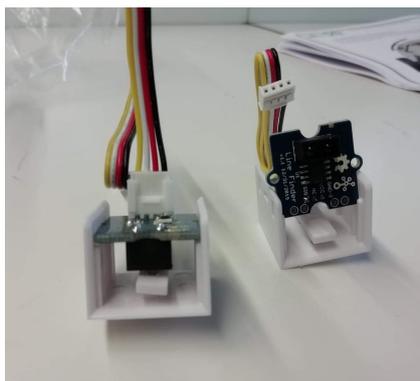
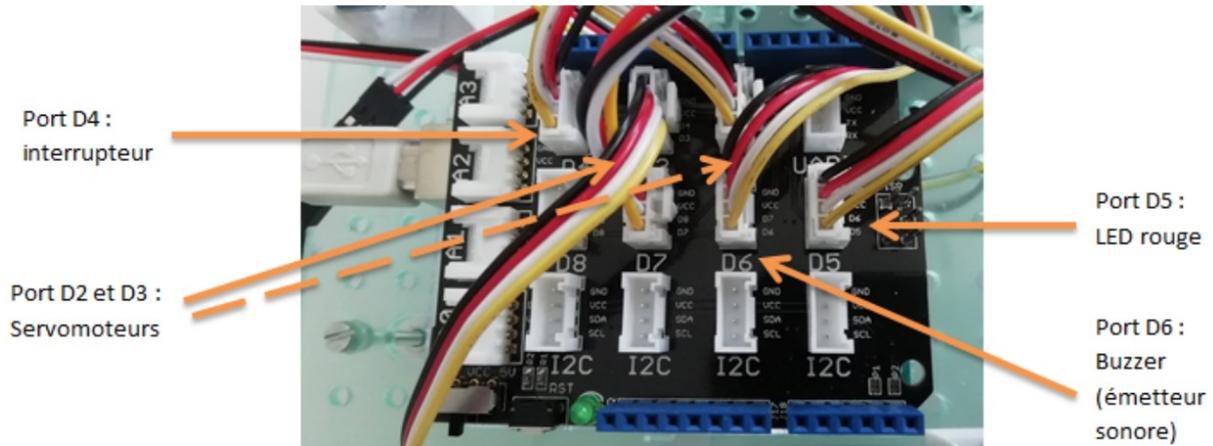


Illustration 3: Interrupteur et suiveurs de ligne du Robot 1

Enfin, le logiciel de simulation mBlock ne fonctionnait pas sur les ordinateurs de l'INSA. N'ayant pas les clés d'accès pour réinstaller ce logiciel sur ces ordinateurs, nous avons dû le télécharger sur nos ordinateurs personnels et les amener à chaque séance de projet. Ce problème nous a empêché d'avancer sur la programmation pendant deux séances. Nous avons donc été contraints de changer de logiciel, nous nous sommes dirigés vers Ardublock.

3.1.3. Programmation Carte Arduino Uno

Détail des ports de la carte



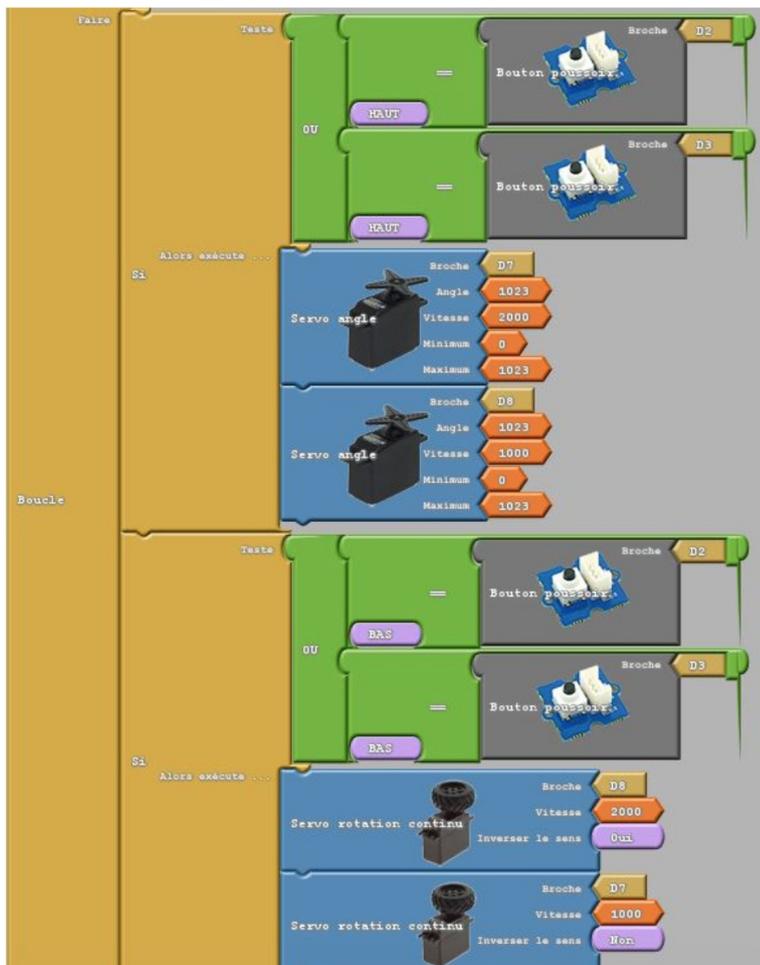
Pour se familiariser avec la programmation, nous avons commencé par coder un programme simple grâce au logiciel de programmation Ardublock : lorsque l'on appuie sur l'une des moustaches (bouton poussoir) la LED rouge s'allume et lorsque l'on appuie sur l'autre, la LED s'éteint. Nous sommes parvenus à téléverser le programme sur la carte Arduino de notre robot mais ce dernier ne réalisait pas les actions attendues, tandis qu'un autre robot (celui de l'année passée qui est du même modèle) les exécutait sans problème. Nous avons donc vérifié que l'alimentation était bien de 9V (vérification à l'ohmmètre). Nous nous sommes également assurés que la carte Arduino fonctionnait correctement en important sur la carte le programme basique "Blink", qui fait clignoter la LED jaune située sous la carte. Nous avons finalement émis l'hypothèse que le problème était lié à la LED rouge et avons décidé de coder un deuxième programme qui faisait appel à d'autres fonctionnalités du robot.



Le premier programme réalisé a été codé sous le logiciel Ardublock Education, qui permet de simplifier le langage C grâce à un système de blocs à assembler. Voir annexe 2 pour version du code en C. Notre objectif était de manipuler la LED et les moustaches (boutons poussoir). Si le premier bouton poussoir (port D2 de la carte Arduino) est enclenché (en position HAUT) alors l'état de la LED rouge (port D5 de la carte) passe à ON. Si le second bouton poussoir (port D3) est enclenché (HAUT) alors l'état de la LED passe à OFF.

Ardublock nous a ainsi permis de nous initier au langage de programmation C. Nous en avons compris les principes de base :

- « `_ardublockDigitalRead()` » prend en entrée le numéro du port. « `pinMode()` » configure le port pour qu'il se comporte comme une entrée (input). « `digitalRead()` » lit la valeur HIGH ou LOW.
- « `_ardublockDigitalWrite()` » prend en entrée le numéro du port ainsi qu'un booléen (l'état HIGH ou LOW du bouton poussoir). « `pinMode()` » configure le port pour qu'il se comporte comme une sortie (output). « `digitalWrite()` » écrit la valeur HIGH ou LOW.
- « `void setup () ;` » n'est appelé qu'une seule fois, au moment de la mise sous tension de la carte. Il s'agit de la partie initialisation.
- « `void loop () ;` » est appelé en boucle tant que la carte est alimentée.



Le second programme repose sur le principe suivant : Si le robot rencontre un obstacle (c'est-à-dire si les moustaches sont enclenchées), il tourne de 180° pour éviter le barrage. Ce programme implique les servomoteurs positionnés sur les ports D2 et D3. Il est donc nécessaire d'inclure des bibliothèques concernant les servomoteurs dans le code en C (voir annexe 3). La fonction « `servo_pin_X.attach(X)` » fixe au servomoteur un numéro de port et la fonction « `servo_pin.X.write()` » le contrôle. Elle prend en entrée une vitesse, un angle ainsi que l'intervalle de variation de l'angle.

3.2. Robot mBot Education

3.2.1. Présentation et objectifs du Robot 2

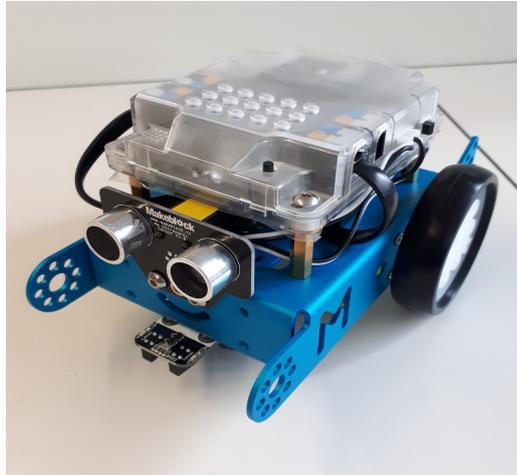


Illustration 4: Robot 2

M. Dhaouadi nous a dans un deuxième temps proposé de travailler sur un robot plus complexe : mBot Educational Robot Kit de la marque Makeblock. Nous n'avons pas eu à monter ce robot, car le but était de se concentrer sur la programmation et la compréhension du fonctionnement du robot. Nous avons donc étudié son comportement en mode automatique.

Les compétences de programmation acquises grâce au Robot 1 nous ont permis de configurer le Robot 2 plus aisément. Nous avons souhaité ajouter une option à notre robot : une pince articulée permettant d'agripper puis de déplacer des objets. Malheureusement, pour des raisons budgétaires, nous n'avons pas pu ajouter cette option. Nous nous sommes donc contentés de récupérer les kits optionnels commandés l'année dernière. De plus, nous avons pu ajouter des programmes en plus de ceux déjà présents sur la carte mère pour exploiter ces options.

3.2.2. Problèmes rencontrés

Avec ce deuxième robot, nous avons rencontré moins de problèmes que précédemment. Le robot, déjà monté, ne pouvait en effet que fonctionner. Nous avons alors étudié son comportement et nous avons été confrontés à un seul souci : lorsque le Robot 2 est bloqué dans une position pendant plus de 5 secondes (il ne réagit pas et ne s'arrête pas, il ne détecte donc pas l'obstacle), le mode erreur s'enclenchait.

3.2.3. Phase de test du robot 2

Dans un premier temps, nous avons testé le Robot 2 dans sa première version, c'est-à-dire avant que l'on effectue les modifications. Il s'agissait ici d'observer le comportement du robot alors qu'un programme automatique était enclenché. Le Robot 2 possède trois modes automatiques :

- il détecte les obstacles au loin et change de direction dès qu'il en croise un dans son champ
- il suit une ligne noire tracée au sol (circuit en forme d'infini fourni dans le kit)
- il reste statique tout en étant allumé si aucun programme n'est ajouté

Lorsque nous avons testé la fonction « détection d'obstacles » nous nous sommes aperçus que le Robot 2 a une vision limitée : si un objet est proche et est moins haut que les capteurs, le robot ne détecte pas l'obstacle. Ces capteurs sont des détecteurs ultrasons, on peut les assimiler aux « yeux » du robot. Lorsque le robot avance en ligne droite, il peut détecter un obstacle éloigné de 50 cm. Par contre, il ne détecte pas les objets fins comme les feuilles de papier.

La fonction « suiveur de ligne » est différente. Les capteurs se situent en dessous, à l'avant du robot. Le but du robot est de suivre une ligne tracée au sol. Cela ne fonctionne que pour une ligne très foncée : noire ou grise. En effet, avec le circuit noir du kit, cela fonctionne, alors qu'avec une ligne violette dessinée par nos soins sur une grande feuille blanche, cela ne fonctionne pas. Enfin, nous avons pu remarquer que le robot a tendance à se diriger vers la gauche s'il ne sait pas où aller. Par ailleurs, le robot n'étant doté que d'un capteur central, il n'était pas apte à détecter des intersections.

Par la suite, nous avons essayé de prendre en compte les spécificités du robot pour créer deux autres versions.

3.2.4. Deuxième version du Robot 2

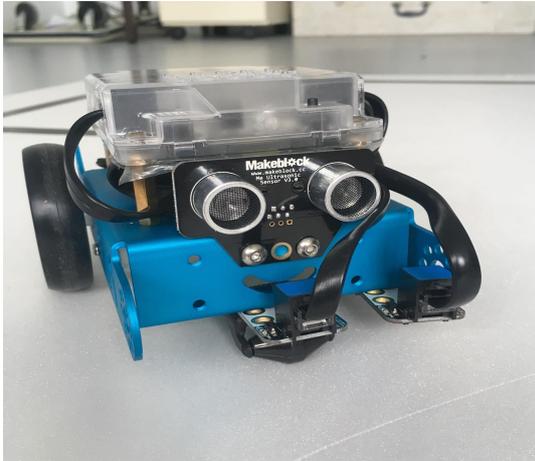


Illustration 5: Deuxième version du Robot 2

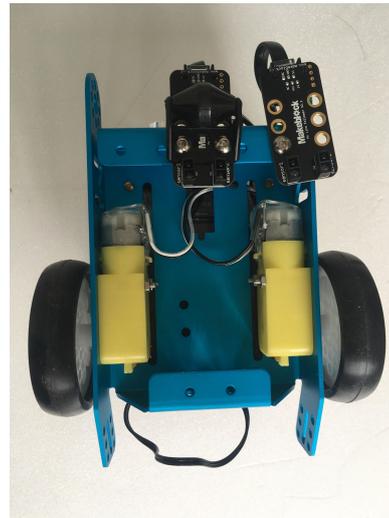


Illustration 6: Dessous de la 2^{ème} version du Robot 2

Pour cette deuxième version, nous avons décidé d'ajouter un suiveur de ligne sur la gauche du robot. Notre objectif était de permettre au robot de mieux repérer les lignes noires au sol, ainsi que de pouvoir détecter les intersections. Nous avons réalisé un parcours et programmé le robot dans le but qu'il suive les lignes de scotch noir (cf. annexe 6).

Le capteur que nous avons ajouté avait pour rôle de détecter les croisements à gauche. En effet, si nous avions voulu détecter également les croisements à droite, nous aurions eu besoin d'un troisième capteur.

L'objectif du programme était que le robot aille à gauche à chaque intersection, qu'il fasse demi-tour lorsqu'il se retrouve dans un cul-de-sac, et enfin qu'il s'illumine de toutes les couleurs pour signifier la fin du parcours.

```

Lorsque le robot(mcore) démarre
définir NbDemiTours à 0

répéter jusqu' à le capteur de suiveur de ligne port2 détecte tout blanc ? et le capteur de suiveur de ligne port1 détecte tout blanc ? et NbDemiTours = 1
répéter jusqu' à le capteur de suiveur de ligne port2 détecte tout blanc ? ou le capteur de suiveur de ligne port1 détecte côté droit noir ?
avancer à 25 % de puissance
si le capteur de suiveur de ligne port2 détecte côté droit blanc ? et le capteur de suiveur de ligne port2 détecte côté gauche noir ? alors
répéter jusqu' à le capteur de suiveur de ligne port2 détecte tout noir ? ou le capteur de suiveur de ligne port2 détecte tout blanc ?
roue gauche tourne à 40 % de puissance, roue droite à 10 % de puissance
si le capteur de suiveur de ligne port2 détecte côté gauche blanc ? et le capteur de suiveur de ligne port2 détecte côté droit noir ? alors
répéter jusqu' à le capteur de suiveur de ligne port2 détecte tout noir ? ou le capteur de suiveur de ligne port2 détecte tout blanc ?
roue gauche tourne à 10 % de puissance, roue droite à 40 % de puissance
si le capteur de suiveur de ligne port2 détecte tout blanc ? et le capteur de suiveur de ligne port1 détecte tout blanc ? et NbDemiTours = 0 alors
répéter jusqu' à le capteur de suiveur de ligne port2 détecte tout noir ?
définir NbDemiTours à 1
roue gauche tourne à 50 % de puissance, roue droite à 50 % de puissance
si le capteur de suiveur de ligne port1 détecte tout noir ? alors
répéter jusqu' à le capteur de suiveur de ligne port2 détecte tout noir ? et le capteur de suiveur de ligne port1 détecte tout blanc ?
roue gauche tourne à 0 % de puissance, roue droite à 50 % de puissance
roue gauche tourne à 50 % de puissance, roue droite à 50 % de puissance
LED tout affiche la couleur pendant 1 secondes
LED tout affiche la couleur pendant 1 secondes
LED tout affiche la couleur pendant 1 secondes
LED tout affiche la couleur pendant 1 secondes
LED tout affiche la couleur pendant 1 secondes
stopper le mouvement
  
```

Explications de ce que fait le robot lors de l'exécution du programme :

Tant que le robot est toujours situé sur la ligne noire et qu'il n'a pas rencontré de virages (pas de route à sa gauche), il continue à avancer en suivant la ligne. Si jamais le robot dévie de sa trajectoire, il se réoriente jusqu'à ce que le capteur central retrouve la ligne noire.

Dans le cas d'une impasse, le robot dépasse la fin de la ligne et tous les capteurs détectent alors du blanc. Le robot fait alors demi-tour.

Lorsque le robot rencontre un virage à gauche, c'est-à-dire quand les deux détecteurs captent du noir, il tourne sur la gauche jusqu'à ce qu'il n'y ait plus que le capteur du milieu qui perçoive du noir.

Afin de marquer la fin du parcours, nous avons rajouté un compteur d'impasses : lorsque l'impasse rencontrée correspond à la fin du parcours, le robot effectue une « danse de la victoire » en s'éclairant et tournant.

Lors de la programmation, nous avons rencontré plusieurs difficultés :

- tout d'abord, l'exécution du programme par le robot était faussée par le faible niveau des piles.
- afin de pouvoir rajouter un second capteur sur le robot, par manque de place, nous avons dû les monter à l'envers (l'avant vers l'arrière). De ce fait, nous avons dû inverser les instructions dans le programme (droite devenant gauche et inversement).
- dans la première version du programme, le robot exécutait l'instruction permettant de suivre la ligne au lieu de l'instruction demi tour. Ceci était dû au fait que lorsqu'une impasse est rencontrée, les deux côtés du capteur central ne captent pas le blanc en même temps. En effet, le robot était programmé de sorte que chaque côté du suiveur de ligne central capte le scotch noir en décalé. Ainsi, le robot se déplaçait jusqu'à ce qu'un des capteurs perçoive une ligne noire. Nous avons donc dû rajouter des conditions plus précises et faire attention à la différenciation des capteurs. Les boucles qui permettent le suivi de la ligne s'arrêtent maintenant dès que le robot ne détecte que du blanc, ce qui enclenche par la suite l'instruction « demi tour ».

3.2.5. *Troisième version du Robot 2*

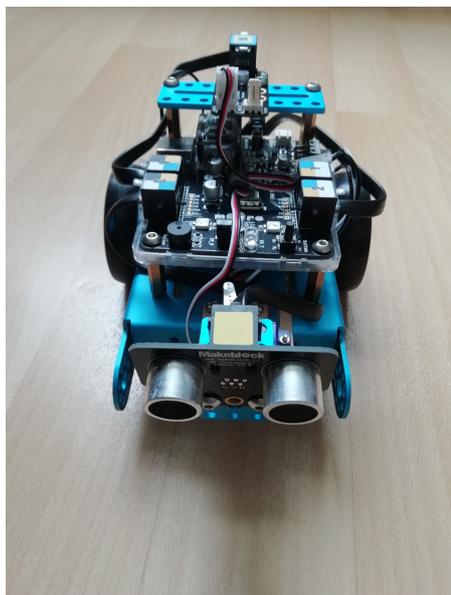


Illustration 7: Troisième version du Robot 2

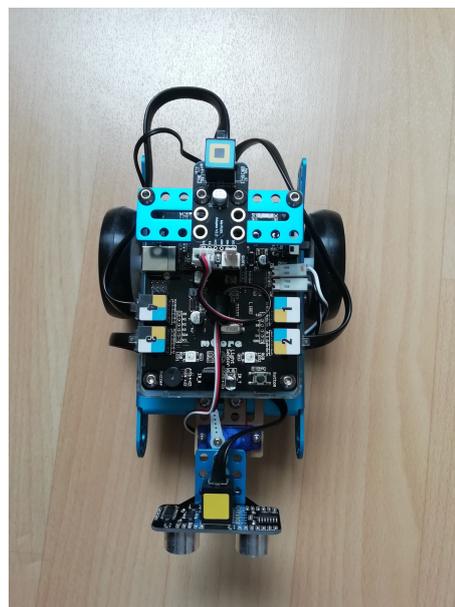
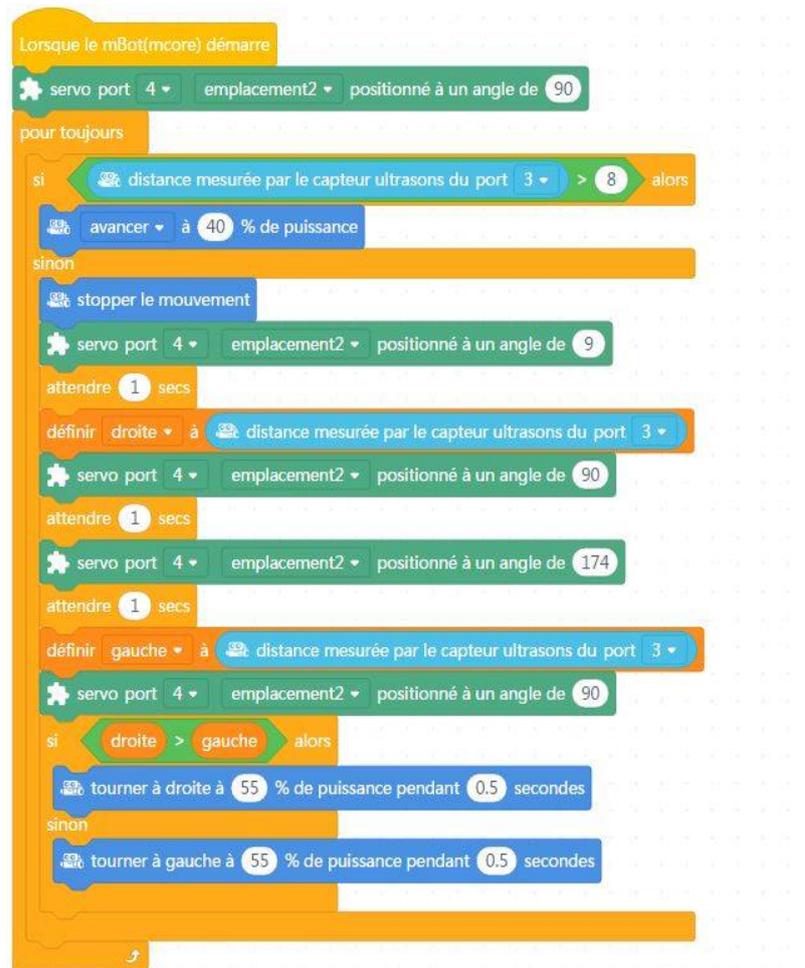


Illustration 8: Vue de dessus de la 3^{ème} version du Robot 2

Pour cette troisième version, nous avons décidé d'ajouter un moteur permettant au robot de tourner la tête, à partir du kit mBot Servo Pack, modèle Head-Shaking Cat. Cela permettait au robot d'évaluer les distances, grâce à sa tête pouvant effectuer une rotation de 180°. Nous avons programmé le robot dans le but qu'il sorte d'un parcours réalisé avec des cartons (cf. annexe 7).



L'objectif de ce programme était que le robot, lorsqu'il se trouve face à un obstacle, tourne la tête de chaque côté et détermine dans quelle direction il doit aller.

Premièrement, le robot place sa tête à 90°, c'est-à-dire en face de lui. Lorsqu'il détecte un obstacle grâce aux capteurs ultrasons il s'arrête. Puis, il regarde à droite (angle de 9°) et mémorise la distance qu'il y a jusqu'au prochain obstacle. Il fait de même à gauche (angle de 174°). Ensuite, il compare les deux distances obtenues, puis tourne dans la direction où la distance est plus grande, autrement dit là où le chemin est le plus dégagé.

Malheureusement nous avons rencontré quelques difficultés lors de l'élaboration du programme. En effet, ce que nous croyions bon à l'écran ne s'exécutait pas de la même façon sur le robot. Nous avons donc dû adapter notre programme, cohérent dans la théorie, à la pratique. Nous avons pu observer notamment quelques actions inadéquates du robot : à la fin, ses capteurs ultrasons semblaient défectueux puisqu'il fonçait dans les murs (ou cartons du parcours).

Après quelques modifications, notamment des paramètres tels que les angles et les distances, nous sommes parvenus à faire fonctionner le robot comme nous le voulions.

4. CONCLUSIONS ET PERSPECTIVES

Conclusions sur le travail réalisé

Nous avons acquis des bases en robotique autant sur l'aspect montage avec la soudure que sur l'aspect programmation. Nous sommes parvenus à réaliser les objectifs que l'on s'était fixés en réalisant deux versions d'un programme permettant de sortir d'un parcours défini.

Conclusions sur l'apport personnel de cet E.C. projet

Ce projet nous a beaucoup enrichi personnellement par les aspects travail de groupe et respect des contraintes. De plus, la plupart des membres du groupe est intéressé par la filière ASI, ce projet était donc pour nous un moyen d'expérimenter une application de l'informatique que nous ne connaissions pas.

Perspectives pour la poursuite de ce projet

La robotique est un domaine intéressant qui mérite d'être approfondi. Nous aurions aimé poursuivre ce projet, si nous en avons eu la possibilité. Dans ce cas, nous nous serions sans doute attardés sur le robot mBot Educational, étant le robot le plus avancé des deux modèles que nous avons étudiés. Nous aurions essayé d'ajouter d'autres options afin d'exploiter au mieux les performances de cet outil robotique.

5. BIBLIOGRAPHIE

[1] lien internet : <https://www.technologieservices.fr> (valide à la date du 05/06/2019)

[2] lien internet : <https://www.makeblock.com/steam-kits/mbot> (valide à la date du 05/06/2019)

6. ANNEXES

6.1. Documentation technique

Comment réaliser un objet technique ?

Séquence 2. Fabrication de l'objet technique.

Capacité. Réaliser un assemblage ou tout ou partie d'un objet technique en suivant une procédure formalisée (2).

L'élèveur un acte technique en respectant les consignes (2).



**LE BRASAGE
D'UN COMPOSANT**

Sitetechno.fr - Niveau 6^e

Le ___/___/___ 2014

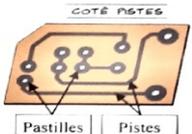
NOM : _____

Prénom : _____

Classe : _____

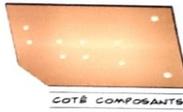
Comment est constitué un circuit électronique ? Quel type d'assemblage met-il en œuvre ?

1 - Le circuit imprimé :
Un circuit imprimé est un support permettant de relier électriquement des composants électroniques entre eux.
Les pistes relient électriquement les pastilles.
Une fois perforées, les pastilles établissent une liaison électrique entre les composants électroniques.



COTÉ PISTES

Pastilles Pistes



COTÉ COMPOSANTS

2 - Le poste de brasage :
Avant de commencer le brasage, il faut vérifier que le matériel est bien disponible :

Fer à souder

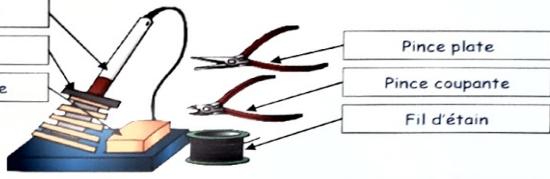
Repose fer

Eponge humide

Pince plate

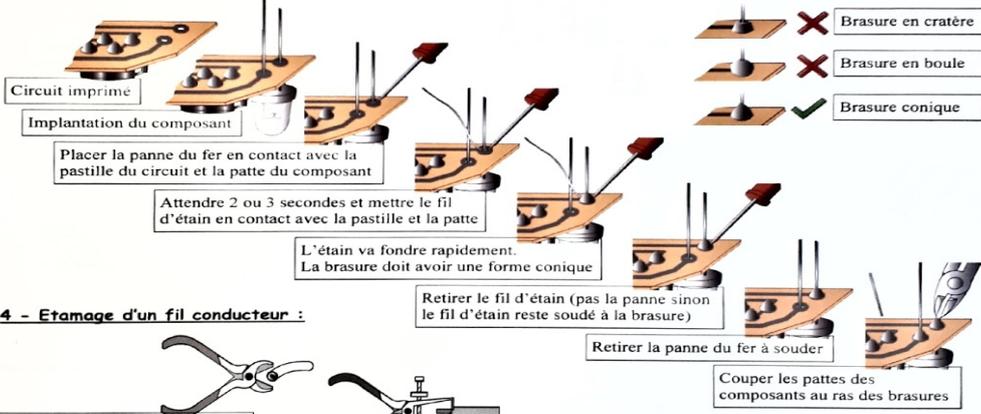
Pince coupante

Fil d'étain



Dessins. Animations Techno-Flash.com

3 - Implantation et fixation d'un composant :



Circuit imprimé

Implantation du composant

Placer la panne du fer en contact avec la pastille du circuit et la patte du composant

Attendre 2 ou 3 secondes et mettre le fil d'étain en contact avec la pastille et la patte

L'étain va fondre rapidement.
La brasure doit avoir une forme conique

Retirer le fil d'étain (pas la panne sinon le fil d'étain reste soudé à la brasure)

Retirer la panne du fer à souder

Couper les pattes des composants au ras des brasures

Brasure en cratère

Brasure en boule

Brasure conique

4 - Etamage d'un fil conducteur :

Couper le fil à la bonne longueur

Dénuder l'extrémité du fil sur 5 mm environ

Torsader les brins à la main

Incliner la partie à étamer vers le bas.
Chauffer les brins pendant 2 ou 3 secondes.
Déposer l'étain sur les brins.
Laisser pénétrer l'étain.

Dès la formation d'une goutte, retirer l'étain puis le fer à souder

Couper la goutte à l'aide d'une pince coupante

Une fois le fil étamé, il reste à l'implanter et à le brasier comme un composant électronique classique.

Annexe 1 : Fiche pour apprendre à souder

6.2. Listings des programmes réalisés

Traduction du code en C

```

boolean __ardublockDigitalRead(int pinNumber)
{
    pinMode(pinNumber, INPUT);
    return digitalRead(pinNumber);
}

void __ardublockDigitalWrite(int pinNumber, boolean status)
{
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, status);
}

void setup()
{
}

void loop()
{
    if (( ( HIGH ) == ( __ardublockDigitalRead(2) ) ))
    {
        __ardublockDigitalWrite(5, HIGH);
    }
    if (( ( HIGH ) == ( __ardublockDigitalRead(3) ) ))
    {
        __ardublockDigitalWrite(5, LOW);
    }
}

```

Annexe 2 : code en C du programme p.10

```

#include <TSServo.h>

boolean __ardublockDigitalRead(int pinNumber)
{
    pinMode(pinNumber, INPUT);
    return digitalRead(pinNumber);
}

TSServo servo_pin_7;
TSServo servo_pin_8;

void setup()
{
    servo_pin_7.attach(7);
    servo_pin_8.attach(8);
}

void loop()
{
    if (( ( ( HIGH ) == ( __ardublockDigitalRead(2) ) ) || ( ( HIGH ) == ( __ardublockDigitalRead(3) ) ) ))
    {
        servo_pin_7.write( 1023, 2000, 0, 1023 );
        servo_pin_8.write( 1023, 1000, 0, 1023 );
    }
    if (( ( ( LOW ) == ( __ardublockDigitalRead(2) ) ) || ( ( LOW ) == ( __ardublockDigitalRead(3) ) ) ))
    {
        servo_pin_8.write(2000, true);
        servo_pin_7.write(1000, false);
    }
}

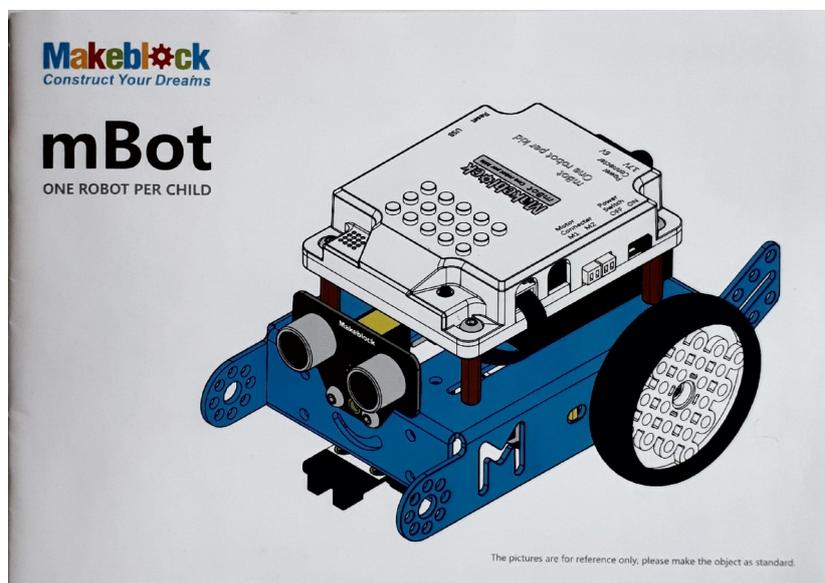
```

Annexe 3 : Code en C du programme p.11

6.3. Notices de montage

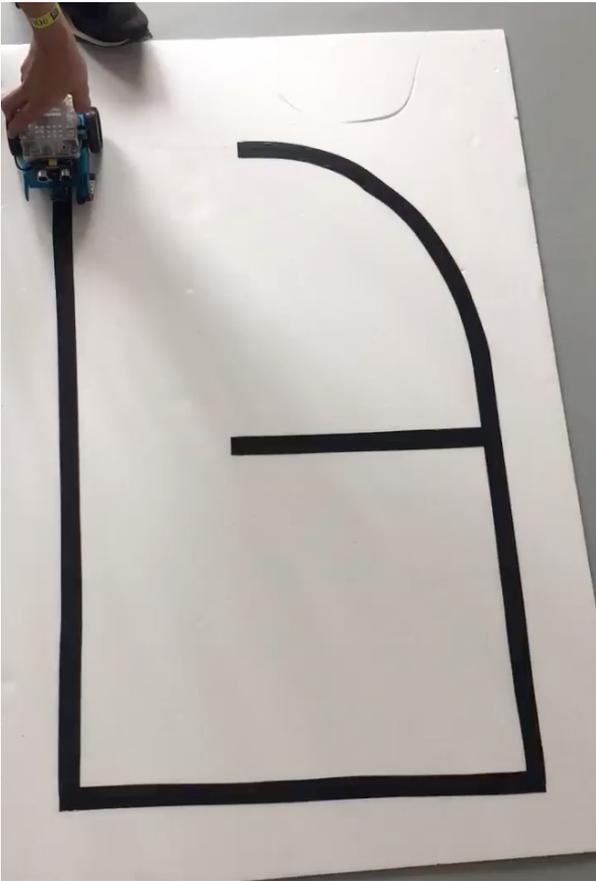


Annexe 4

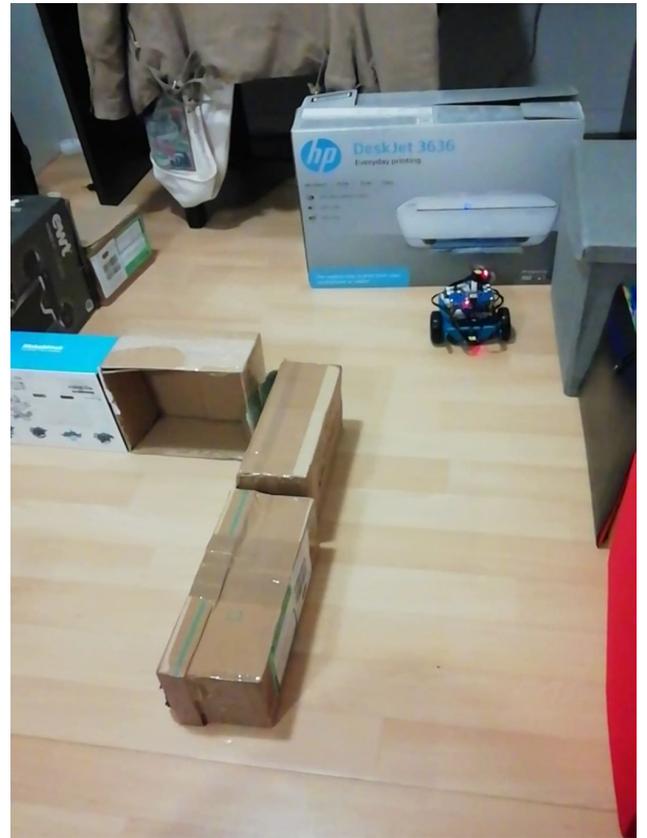


Annexe 5

6.4. Parcours réalisés



Annexe 6 : Parcours réalisé pour la deuxième version du Robot 2



Annexe 7 : Parcours réalisé pour la troisième version du Robot 2