

DECOUVERTE DE LA ROBOTIQUE, ASSEMBLAGE ET PROGRAMMATION

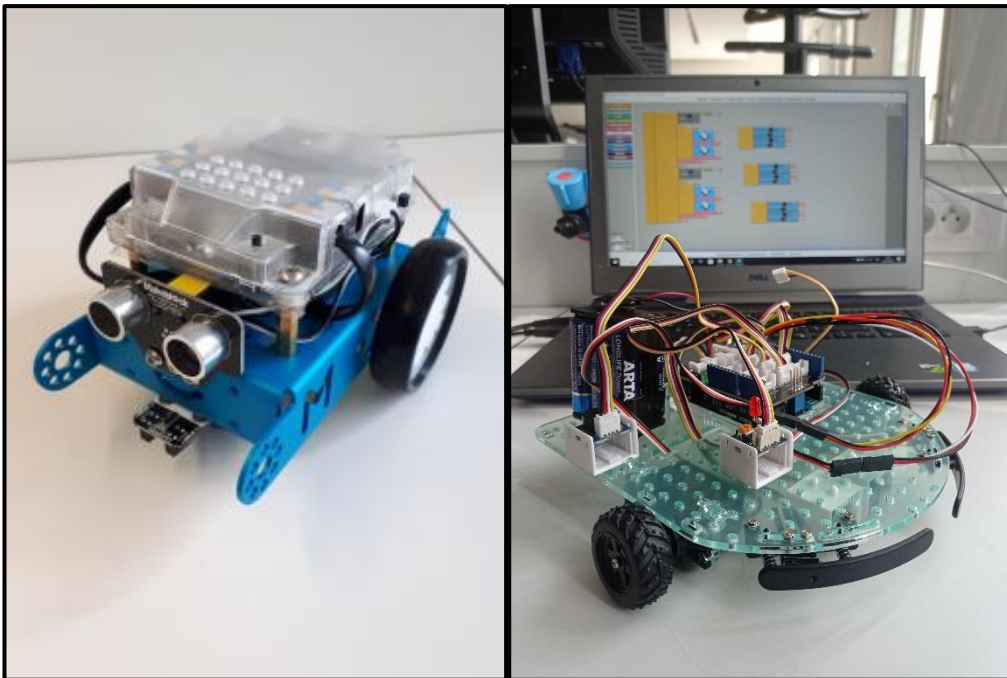


Figure 1 : Photos du robot mBot et du robot Arduino Evolutif

Etudiants :

Léa BREBANT

Colombe MACE

Jules SCHLICK

Jules FORTIN

Georges MAROUN

Raphaël TAORMINA

Enseignant-responsable du projet :

Faouzi DHAOUADI

Date de remise du rapport : **11/06/2021**

Référence du projet : **STPI/P6/2021 – 018**

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

Type de projet : **Expérimental**

Objectifs du projet :

L'objectif de ce projet était de découvrir la robotique en passant par l'élaboration de robots. Pour cela, il nous a fallu acquérir des compétences variées telles que la soudure, l'assemblage des composants ainsi que la programmation.

Nos robots avaient pour objectif de s'orienter dans un labyrinthe et d'être capable de suivre la trajectoire d'une ligne sur le sol.

En parallèle, nous avons dû nous former à l'utilisation de nouveaux outils de programmation. En effet, nous avons été amenés à utiliser les logiciels Ardublock Education et mBlock pour programmer nos deux robots.

Enfin, ce projet nous a permis de nous projeter dans le monde de l'ingénierie à travers un projet d'équipe réaliste.

Mots-clefs du projet :

SOUDURE	ASSEMBLAGE
PROGRAMMATION	ROBOTIQUE

TABLE DES MATIERES

REMERCIEMENTS	5
1. Introduction	6
2. Methodologie / Organisation du travail.....	7
3. Travail réalisé et résultats	8
3.1. Apprentissage de la Soudure	8
3.1.1. Assemblage d'une lampe solaire.....	8
3.1.2. Assemblage de la voiture radiocommandée.....	9
3.1.3. Photos des assemblages terminés	10
3.2. Robotique et programmation.....	11
3.2.1. Robot Arduino Evolutif.....	11
3.2.1.1 Présentation du robot et fonctionnalités attendues	11
3.2.1.2 Présentation des programmes utilisés.....	13
3.2.1.3 Phase d'exécution des différents programmes.....	15
3.2.2. Robot mBot Educational.....	17
3.2.2.1 Présentation du robot et fonctionnalités attendues.....	17
3.2.2.2 Problèmes rencontrés et solutions mises en place.....	17
3.2.2.3 Phase de test du robot mBot.....	18
3.2.2.4 Premier programme du robot mBot	19
3.2.2.5 Deuxième programme du robot mBot.....	20
4. Conclusion.....	22
5. Bibliographie.....	23
6. Annexes	24
6.1. Documentation technique.....	24
6.2. Schéma de montage.....	26

REMERCIEMENTS

En tout premier lieu, nous tenons à remercier M. DHAOUADI Faouzi, notre responsable de projet pour son implication, son soutien et son enseignement tout au long du projet. Le partage de ses connaissances en robotique et en soudure nous a permis de comprendre pleinement notre sujet et de nous épanouir dans le développement de notre projet. De plus, nous le remercions de nous avoir fourni tous les supports nécessaires (voiture, lampes, robots).

Par la suite, nous souhaitons remercier l'équipe responsable de l'atelier dans lequel nous avons soudé et assemblé nos différents systèmes, Mme. RADE Hélène et M. JOLLY Michaël. Leur disposition et leur aide dans l'apprentissage de la soudure nous ont servi à développer nos différents robots.

Enfin, nous aimerions remercier l'INSA, pour la mise en place de ces projets de P6 qui nous ont permis de découvrir un enseignement auquel nous n'aurions pas pu avoir accès lors de la formation STPI. Nous les remercions également d'avoir laissé à disposition le matériel de soudure ainsi que des piles.

1. INTRODUCTION

Lors de notre formation en cycle préparatoire de STPI, nous sommes amenés à effectuer plusieurs projets de groupe. Ces enseignements ont une part importante dans notre formation d'ingénieur. En effet, ils ont pour but de nous faire travailler en équipe autour de sujets variés en complémentarité avec nos enseignements théoriques. Ils permettent également d'acquérir de la rigueur dans leur organisation et leur gestion.

La « Découverte de la robotique, assemblage et programmation » a été notre sujet d'étude pendant ce semestre. Il nous a permis de découvrir le domaine de la robotique qui nous intéressait fortement mais que nous ne connaissions pas. De plus, la démarche d'expérimentation proposée par ce sujet, nous a été utile pour assimiler pleinement les différents enjeux du projet.

L'objectif de notre projet était donc de nous faire découvrir la soudure, la robotique et la programmation, via l'assemblage de deux robots. Nous avons dans un premier temps appris la soudure en manipulant de vieilles pièces et en assemblant des petits objets électroniques pour nous familiariser avec cette technologie. Ensuite, nous sommes passés à l'assemblage d'un des robots, ce qui nous a permis de comprendre pleinement le fonctionnement mécanique de ce dernier. Pour finir, nous avons programmé nos deux robots à l'aide de logiciels spécialisés pour développer différentes fonctionnalités.

2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Nous avons travaillé toutes les semaines avec M.Dhaouadi sur des créneaux de 1h30. De manière générale, nous n'avons pas vraiment besoin de travailler en supplément en-dehors de cette heure-là car la majorité de notre projet se déroulait dans l'atelier mis à notre disposition. Seule la rédaction du rapport nous a demandé de nous organiser en dehors des heures de cours. Voilà donc comment nous nous sommes répartis le travail.

	Léa	Jules F	Colombe	Georges	Jules S	Raphaël
Initiation à la soudure	✓	✓	✓	✓	✓	✓
Soudure d'une lampe		✓		✓	✓	✓
Soudure d'une voiture	✓		✓			
Robot 1 Assemblage		✓		✓	✓	✓
Robot 1 Programmation		✓		✓	✓	✓
Robot 2 Programmation	✓		✓			
Rédaction du rapport	✓	✓	✓	✓	✓	✓
Poster		✓				
Préparation soutenance	✓	✓	✓	✓	✓	✓

Pour la répartition du travail, nous n'avons pas eu de problèmes particuliers, tout le monde était volontaire et participait correctement.

Pour la soudure, nous nous sommes tous entraînés ce qui nous a permis de nous répartir les deux projets. Nous avons fait un groupe de 4 pour la lampe, car il y en avait plusieurs à disposition, et un groupe 2 pour l'unique voiture neuve.

Pour les robots, nous avons conservé les mêmes groupes, celui de 4 s'occupait du robot 1 (Arduino) car il avait besoin d'être assemblé, soudé puis programmé et le groupe de 2 s'est, quant à lui, occupé du robot 2 (mBot) car il avait seulement besoin d'être programmé. Enfin, nous avons tous participé au rapport et à la préparation de la soutenance.

3. TRAVAIL REALISE ET RESULTATS

3.1. Apprentissage de la Soudure

Quand nous avons débuté le projet, le premier objectif était d'apprendre à souder des composants électroniques et des fils. En effet, aucun membre du groupe n'avait jamais effectué de soudure auparavant. Notre professeur-responsable nous a alors donné d'anciennes cartes électroniques pour faire quelques essais de soudage et dessoudage de composants, pour s'entraîner et pouvoir ensuite appliquer cette technique sur des assemblages plus complexes. Cette période d'apprentissage s'est déroulée lors des deux premières séances.

Une fois que nous avons bien compris comment manier le fer à souder, ainsi que la pompe à dessouder, M. DHAOUADI nous a mis à disposition 3 petits appareils à monter. Nous nous sommes alors séparés en trois binômes : un binôme travaillant sur l'assemblage d'une voiture radiocommandée et deux binômes travaillant chacun sur une lampe, pour mieux appréhender son assemblage, s'allumant ou non en fonction de l'éclairage ambiant.

3.1.1. Assemblage d'une lampe solaire

Dans un premier temps, nous avons vérifié, à l'aide de la fiche technique de la lampe¹, que nous avons tous les composants nécessaires à sa réalisation.

Ensuite, nous devons faire attention à l'ordre d'implantation des composants. En effet, nous avons commencé par mettre en place les résistances. Nous devons être très attentifs à celles que nous utilisons et au sens dans lesquelles elles étaient disposées dans le circuit. Pour cela, nous avons dû nous renseigner sur la signification du code couleur présent sur ces dernières.

Par la suite, nous nous sommes occupés de les intégrer au circuit imprimé. En effet, nous devons plier les tiges des résistances à l'aide d'un dispositif de pliage pour obtenir un espacement correct et les positionner convenablement. Une fois toutes ces étapes réalisées nous pouvions enfin les souder au circuit. La manipulation du fer à souder devait être très précise car les pièces étaient de petites dimensions.

Nous avons fait de même pour les autres composants de la lampe :

- Transistor
- Condensateur
- Diode
- Inductance
- D.E.L.
- Interrupteur

¹ Voir le document correspondant dans l'annexe des schémas de montage

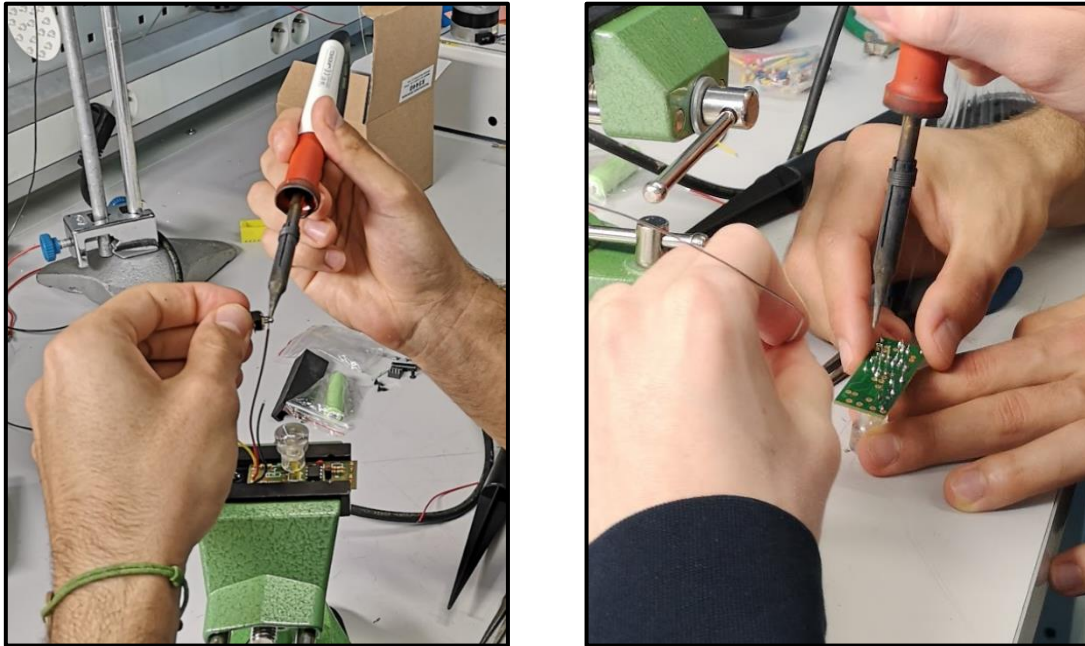


Figure 2 : Photo du soudage des différents éléments au circuit imprimé

Pour finir, nous avons intégré le circuit imprimé et tous ses composants à la structure de la lampe et au panneau photovoltaïque. Nous avons donc testé la lampe en bloquant l'arrivée de la lumière avec notre main sur le panneau. Les deux lampes fonctionnaient sans problèmes, nous en avons donc déduit que notre soudure était efficace.

3.1.2. Assemblage de la voiture radiocommandée

Pour l'assemblage de la voiture radiocommandée, nous avons principalement soudé des fils électriques. Nous avons pu nous aider de plusieurs anciens modèles de voitures. De plus, nous avons dû visser les différentes parties de la voiture entre-elles. Nous avons également appris à utiliser une gaine thermo-rétractable.

En effet, nous en avons disposé autour des résistances et des fils afin de les empêcher d'entrer en contact entre-eux. Nous avons également assemblé la télécommande. Nous avons pu vérifier notre travail en essayant de faire fonctionner la voiture. Celle-ci répondait correctement aux instructions que nous lui avons transmises grâce à la télécommande.

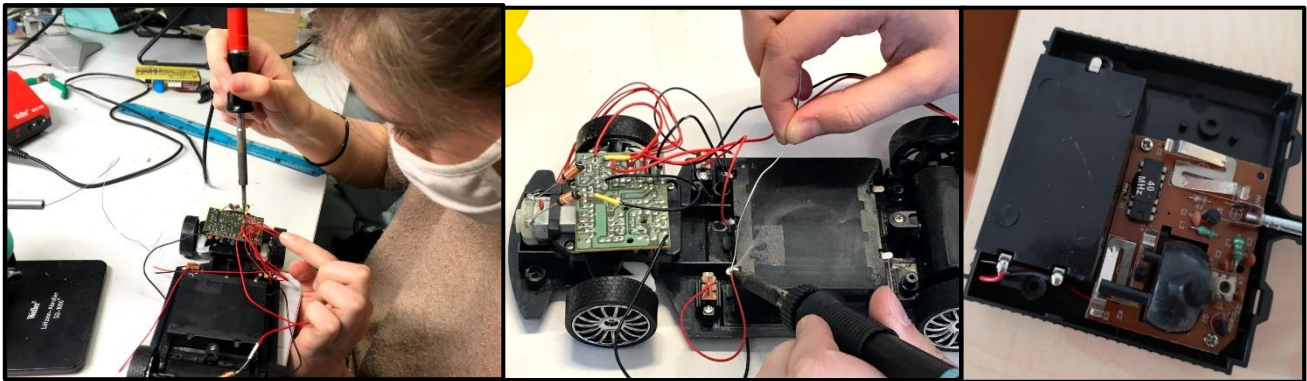


Figure 3 : Photos des différentes étapes de l'assemblage de la voiture radiocommandée

3.1.3. Photos des assemblages terminés

Voici les différentes photos de la lampe solaire et de la voiture radiocommandée une fois assemblées :

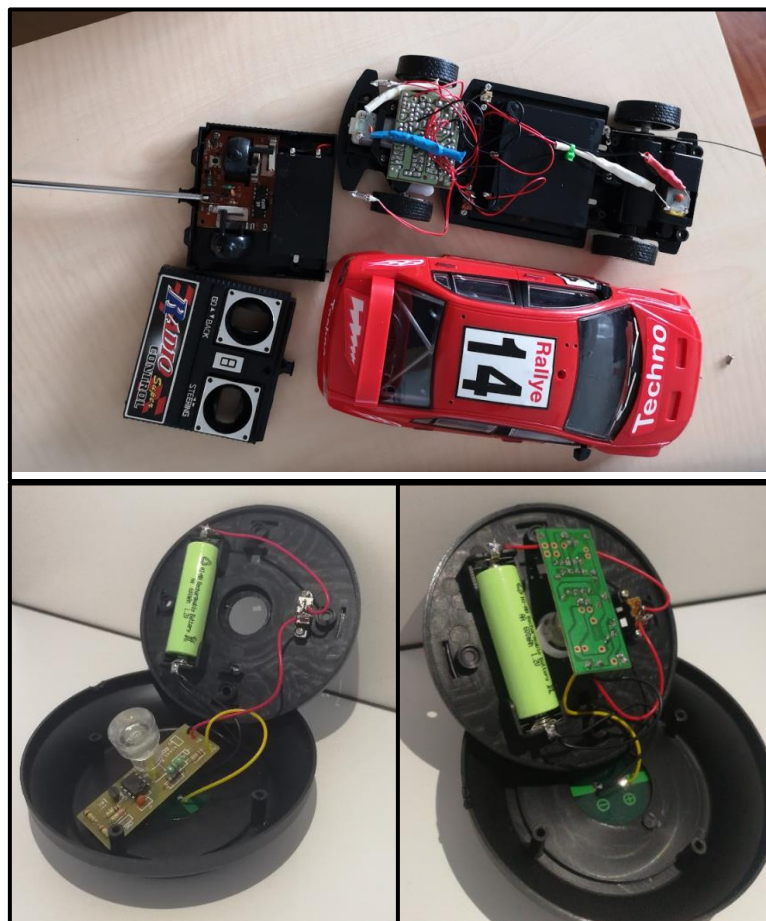


Figure 4 : Photos de la voiture et de la lampe assemblées

3.2. Robotique et programmation

3.2.1. Robot Arduino Evolutif

3.2.1.1 Présentation du robot et fonctionnalités attendues

Notre objectif principal consistait à construire un robot, afin qu'il puisse s'orienter et contourner des obstacles. Pour ce faire, nous avons utilisé le kit robot UnoEvo piloté par Arduino. Il est capable de détecter un obstacle sur sa course, de suivre une ligne et de communiquer des informations visuelles et sonores. Notre robot possède donc deux modes de fonctionnement indépendants.

Bien que le robot nous ait été donné sous forme de kit, il nous a fallu tout d'abord vérifier la présence de la totalité des composants pour ensuite assembler le robot. L'assemblage a nécessité quelques soudures au niveau des capteurs de collisions, de la visserie et du câblage sur la carte Arduino. Nous avons rencontré quelques problèmes inattendus lors de l'assemblage. En effet, les supports des suiveurs de ligne empêchaient le bon branchement des câbles sur le capteur. Il nous a fallu tordre les câbles, au risque de les endommager, ainsi que l'embout du capteur. Quant à la soudure des capteurs moustaches, le métal chauffé ne coagulait pas avec la partie métallique des capteurs. Lors de la fixation des servomoteurs sur la plaque, nous avons rencontré un problème de visserie car ces derniers ne tenaient pas à celle-ci. Les servomoteurs n'étaient pas alignés et il y avait donc un jeu entre les moteurs et la plaque.

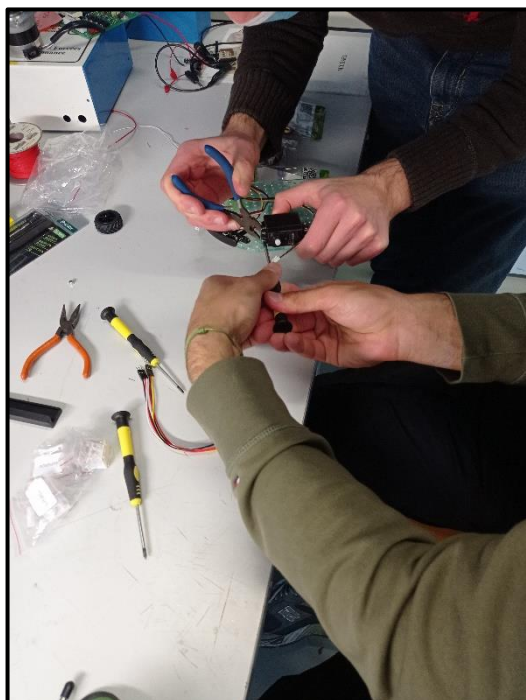


Figure 5 : Assemblage des servomoteurs



Figure 6 : Assemblage du robot Arduino

Notre robot est donc composé de deux capteurs de contacts à l'avant, deux modules suiveurs de ligne au-dessous de la plaque, deux roues reliées à des servomoteurs de chaque côté, d'une carte Arduino et de son shield, d'un support pour piles, d'une LED et d'un buzzer.

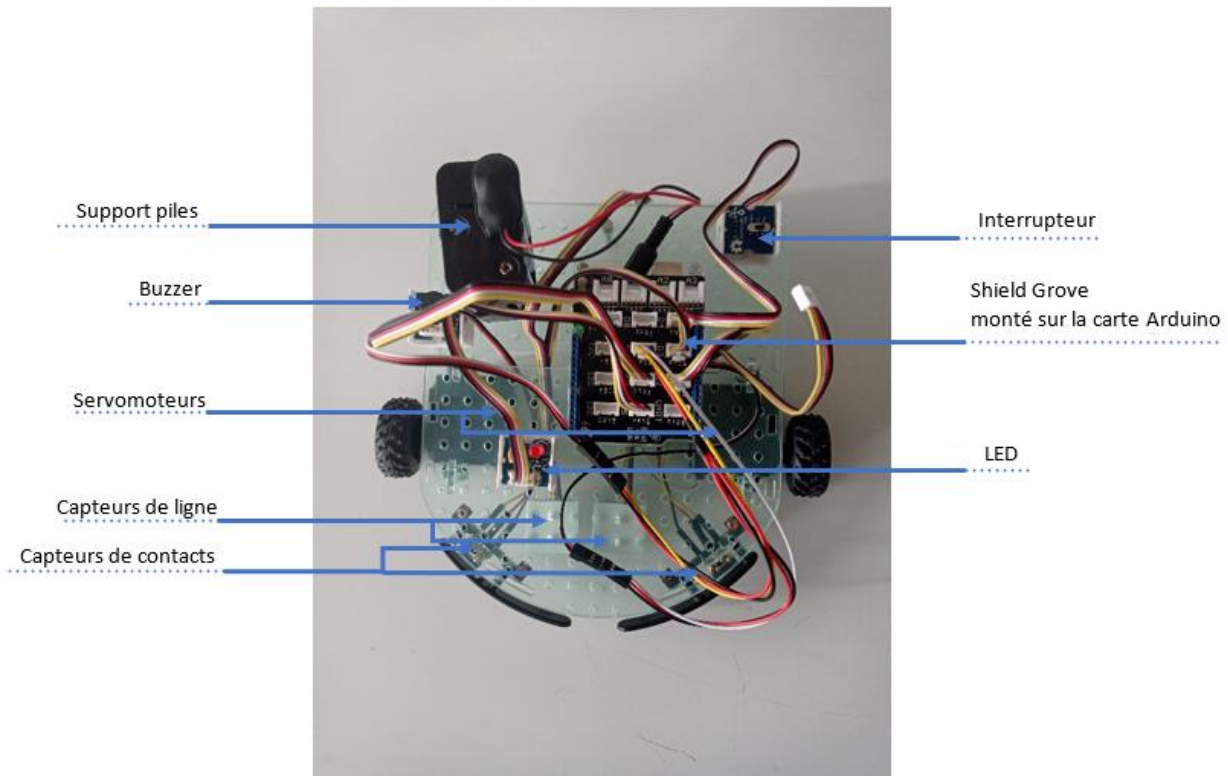


Figure 7 : Robot assemblé avec indications des éléments

La carte Arduino Uno et son microcontrôleur nous ont permis de piloter le robot. Pour faire le lien entre les composants, un shield est placé sur la carte. Il possède plusieurs ports analogiques et numériques mais nous avons seulement utilisé les ports numériques. Cela nous a permis de brancher nos câbles rapidement et facilement sans avoir besoin de passer par la soudure. La carte Arduino possède un port USB qui nous permet de téléverser notre programme sur le microcontrôleur.

Mode 1 : Détecteur d'obstacle :

Le premier mode du robot UnoEvo se base sur la détection d'obstacle. Nous avons utilisé deux capteurs moustaches fixés sur deux microinterrupteurs qui, lorsqu'ils sont enclenchés, permettent de détecter de quel côté du robot se situe l'obstacle. Le fonctionnement du robot dans ce premier mode est très simple, il avance en ligne droite à vitesse constante jusqu'à ce que l'une des deux languettes placées à l'avant soit déclenchée. S'il s'agit de la languette de droite le robot tourne à gauche et inversement.

Mode 2 : Suiveur de ligne :

Pour ce second mode de fonctionnement, le robot doit être capable de suivre une ligne. Nous avons utilisé deux modules infrarouges constitués chacun d'une LED infrarouge et d'un phototransistor. La LED envoie une lumière infrarouge qui est réfléchi sur une surface blanche ou bien absorbée par une surface noire. Le phototransistor quant à lui détecte la présence d'un rayon infrarouge ou non. Il est ainsi possible de déterminer si le capteur se situe au-dessus de la ligne car ce dernier envoie un signal haut lorsqu'il détecte une surface noire.

Nous avons utilisé deux capteurs pour savoir de quel côté la ligne était franchie lors de changements de direction. En temps normal, la ligne se situait entre les deux capteurs.

Ici, même principe que le premier mode, le robot avance en ligne droite et lorsqu'il détecte une surface noire, il corrige sa trajectoire en fonction du capteur qui a été déclenché.

Comme il n'est pas possible d'utiliser les deux modes simultanément, à chaque changement, il faut remplacer les branchements des capteurs de contacts avec ceux des détecteurs de ligne. Nous voyons ici l'avantage que nous offre le shield Grove, cela permet de faire des modifications rapidement.

3.2.1.2 Présentation des programmes utilisés

Avant de commencer la programmation, il faut fixer les ports des différents composants. Ainsi, la LED sera branchée sur la broche D5 et le buzzer sur la broche D6. Les ports D2 et D3 correspondent respectivement aux suiveurs de ligne droit et gauche ou aux interrupteurs de fin de course suivant l'utilisation que l'on souhaite faire du robot. Puis, les servomoteurs sont reliés aux ports D7 et D8. Enfin, on rajoute un interrupteur sur le port D4.

Nous avons codé grâce au logiciel ArduBlock Education. Cet outil permet de simplifier le langage C/C++ de par un système de blocs à assembler. Une fois compilé, le programme était disponible en C². Cela nous a permis de nous familiariser avec la syntaxe et les fonctions principales de ce langage de programmation :

- « void setup » appelée une seule fois lorsque la carte est alimentée, et permet d'initialiser les ports et le programme.
- « pinMode() » définit si cette broche est une entrée (input) ou une sortie (output). Ici, la led 5mm, le buzzer et les moteurs sont des sorties, tandis que les suiveurs de ligne, les interrupteurs fins de course et l'interrupteur sont des entrées.
- « void loop () » qui répète son contenu en boucle.
- « digitalWrite() » donne la valeur HIGH ou LOW à une broche fixée comme sortie.
- « digitalRead » renvoie la valeur HIGH ou LOW d'une broche fixée en entrée.

² Voir annexe document technique

Le programme est composé de plusieurs parties : le programme principal qui s'exécute en boucle et cinq fonctions appelées par le programme principal.



Figure 8 : Fonctions de déplacement

Les quatre premières fonctions sont liées aux différents modes de déplacement du robot : « Gauche », « Droite », « Avancer » et « Reculer ». Pour chaque fonction on adapte la vitesse et le sens de rotation de chaque roue du robot pour qu'il réponde à notre besoin. Par exemple, pour « Avancer » les deux roues avancent à une vitesse de 1000 vers l'avant. En outre, pour tourner à « Gauche », la roue gauche ralentit à une vitesse de 300 tandis que la roue droite conserve sa vitesse de 1000.

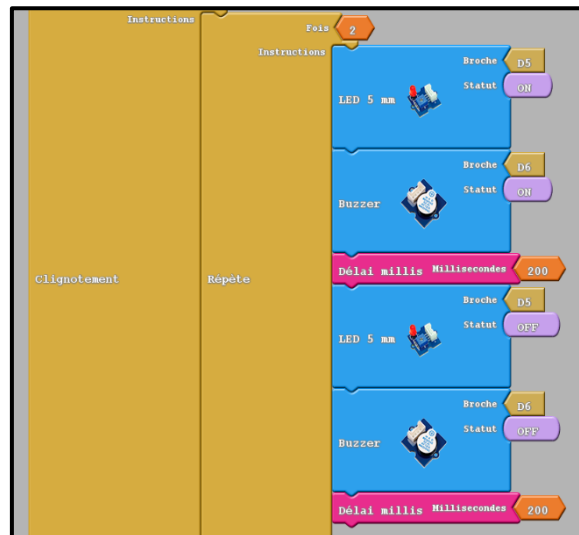


Figure 9 : Fonction clignotement

La dernière fonction appelée « Clignotement » permet grâce aux délais en millisecondes et à une boucle Repeat de faire clignoter la LED et le buzzer qui sont successivement allumés puis éteints. Cependant la LED étant défectueuse et le buzzer (très) bruyant, nous n'avons pas conservé cette fonction optionnelle.

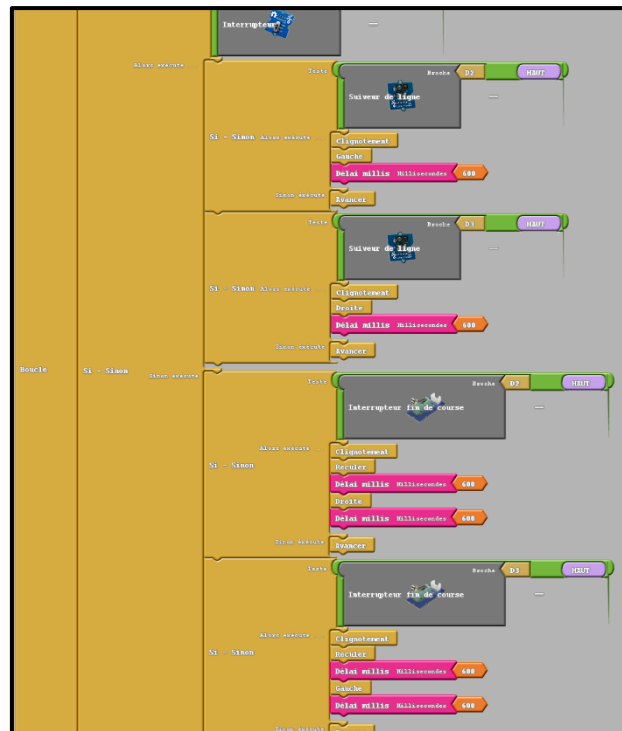


Figure 10.: Programme principal

Le programme principal est en réalité en deux parties, suivant si l'interrupteur est sur ON ou OFF, une partie différente du programme est exécutée. Cela nous permet ainsi de passer facilement du mode « suiveur de ligne » à « détecteur d'obstacle » sans avoir à rebrancher le robot pour téléverser un deuxième programme.

La première partie, celle dite du « suiveur de ligne » repose sur deux boucles « Si-Sinon » et un principe simple : si la ligne est détectée côté gauche, alors le robot se recentre en exécutant « Droit » et inversement si la ligne est détectée par le suiveur de ligne droit. Si rien n'est détecté, le robot avance tout droit.

La deuxième partie, « détecteur d'obstacle » fonctionne de la même manière : lorsqu'un obstacle est détecté d'un côté, le robot « Recule » pour s'éloigner de l'obstacle et tourne vers le côté opposé. Si rien n'est détecté, le robot avance tout droit.

3.2.1.3 Phase d'exécution des différents programmes

Une fois le robot assemblé et le programme téléversé sur la carte Arduino, la phase de tests a pu débuter. Elle s'est déroulée en plusieurs étapes, en commençant par des simples vérifications face à un obstacle au sol, jusqu'à la création d'un court chemin comportant plusieurs obstacles à éviter.

Lors des premiers tests, le robot ne changeait pas de direction lorsqu'il détectait un obstacle. Quelques modifications dans le code au niveau des numéros de PIN de sortie nous ont permis de résoudre ce problème. Ensuite, le robot n'avancait pas en ligne droite parfaite et était dévié à cause de la mauvaise fixation des servomoteurs.

Au vu du défaut des composants, la seule chose que nous avons pu faire a été de renforcer la fixation avec un morceau de ruban adhésif.

Mode 1 : Détecteur d'obstacle :

Pour le mode fonctionnant à l'aide des capteurs de collisions, à ses débuts, le robot ne tournait pas suffisamment et devait s'y reprendre plusieurs fois pour contourner un obstacle. Il nous a fallu corriger le programme et jauger le nombre de tours que devait avoir chaque roue pour tourner du bon angle. Dans l'idéal, le robot devait tourner d'un angle légèrement inférieur à 90° pour éviter qu'il fasse marche arrière. Seulement, avec l'alignement des roues qui n'est pas parfait, le robot ne tournait pas toujours du même angle et avait parfois du mal à s'orienter.

C'est après plusieurs tentatives que le robot a réussi sans problèmes, à s'orienter et traverser un circuit avec quelques obstacles.



Figure 11 : Phase de test du robot avec capteurs de contact

Mode 2 : Suiveur de ligne :

Le second mode de fonctionnement nous a causé plus de difficultés. En effet, les capteurs de ligne n'étant pas assez proches du sol et donc de la ligne, ils ne pouvaient pas détecter correctement le changement de motif sur le sol. Le programme ayant la même structure que celui du premier mode, le robot n'avait pas de problème pour tourner, ce n'était donc pas ça le souci. Seulement, nous n'avons pas pu le faire fonctionner convenablement à cause de ses capteurs trop hauts par rapport au sol.

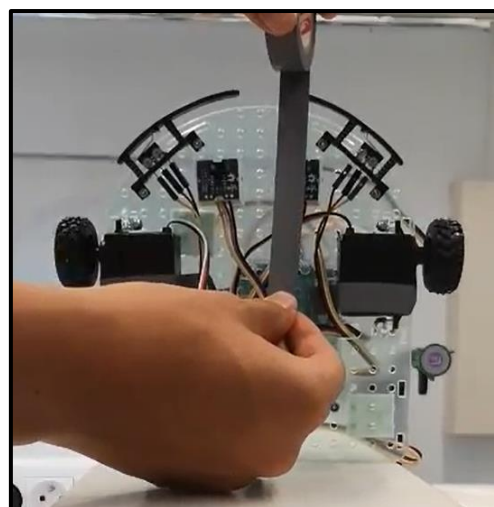


Figure 12 : Détection d'une ligne par les capteurs de ligne

3.2.2. Robot mBot Educational

3.2.2.1 Présentation du robot et fonctionnalités attendues

Dans cette partie, nous avons dû travailler sur la programmation d'un robot mBot Educational. Pour cela, nous avons utilisé le logiciel mBlock. Ce robot possède différents capteurs dont deux auxquels nous nous sommes intéressés : un capteur ultrasonique (constitué d'un émetteur et d'un récepteur) situé sur l'avant du robot, lui permettant de détecter la distance à laquelle se trouvent les obstacles qui lui font face, ainsi qu'un capteur infrarouge, situé en dessous du robot, qui permet de distinguer une couleur sombre d'une couleur claire.

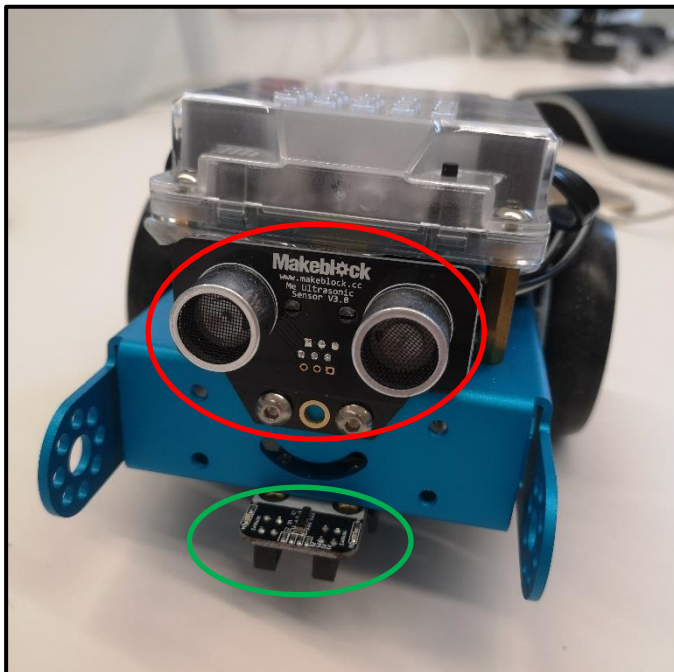


Figure 13 : Photo du robot mBot et de ses capteurs

(Rouge = ultrasonique et Vert = infrarouge)

L'assemblage du robot avait déjà été effectué au préalable afin de pouvoir nous concentrer uniquement sur la programmation. Tout d'abord, nous avons installé le logiciel sur nos PC et nous avons essayé de comprendre les programmes par défaut déjà présents sur le robot.

Nous avons ensuite décidé de réaliser deux programmes différents : un permettant au robot de suivre une ligne disposée sur une surface claire et un autre lui permettant de sortir d'un parcours en repérant les obstacles situés sur son chemin.

3.2.2.2 Problèmes rencontrés et solutions mises en place

Lors de notre travail, nous avons rencontré plusieurs problèmes. Après avoir codé un premier programme sur la version 4.0 de mBlock, nous ne sommes pas parvenus à téléverser celui-ci sur le robot. Malgré plusieurs essais, nous nous sommes rendus compte qu'il était impossible de le téléverser vers notre robot. Pour résoudre ce problème, nous avons finalement essayé d'installer la version v5 3.0, que nous avons trouvée sur le site de makeblock.

Nous avons réécrit dans un premier temps un programme facilement exécutable pour vérifier que le téléversement était possible. Suite à cela écrit un programme test dans cette version et nous avons finalement réussi à le téléverser. Après quelques modifications dans le code, le robot fonctionnait sans difficultés. Nous avons également rencontré des problèmes lors de l'exécution que nous expliquerons dans la partie suivante.

3.2.2.3 Phase de test du robot mBot

La première phase de test a, tout compte fait, été la découverte du robot. En effet, nous avons testé les trois programmes par défaut, qui étaient préinstallés sur le mBot, pour observer son comportement. Voici leurs fonctionnalités :

- Le premier permet de rester statique et allume les LED positionnées sur le dessus
- Le deuxième détecte les obstacles et tourne immédiatement pour les éviter et continuer d'avancer
- Le dernier est un suiveur de ligne qui peut parcourir n'importe quel trajet marqué au sol

Lors de cette phase, nous avons constaté plusieurs paramètres importants qui nous serviront pour la suite de la programmation.

Au niveau de la détection d'obstacles, nous avons vite compris que tous les objets n'étaient pas adaptés à faire obstruction. En effet, les capteurs ultrasons positionnés à l'avant du robot, tels des yeux, ne sont pas capables de détecter les objets trop bas, du fait de la hauteur des yeux, ou même les objets trop fins comme une simple feuille. A partir de ce moment-là, nous n'avons utilisé que des cartons (ou parfois des sacs) pour faire barrière.

Après, au niveau du suiveur de ligne, nous avons rapidement compris le fonctionnement des deux capteurs situés à l'avant, en-dessous du robot. Nous avons aussi constaté qu'il fallait que la ligne soit assez large pour englober les deux capteurs afin que le robot puisse avancer. Enfin, nous nous sommes rendus compte sur le trajet d'il y a deux ans, que le robot ne détectait pas les croisements. En effet, seuls deux capteurs centraux sont montés, ce qui empêche les potentielles détections d'intersections sur le côté.

Grâce à cette première analyse, nous avons pu commencer notre programmation avec de bonnes bases.

3.2.2.4 Premier programme du robot mBot

Pour s’initier à la programmation Mblock, nous avons décidé de refaire le programme du suiveur de ligne mais, en l’améliorant à notre façon.

Nous n’arrivions pas à ouvrir totalement le programme par défaut de ce mode, nous avons donc cherché une vidéo explicative sur Youtube pour avoir un modèle complet. Grâce à ces deux supports, nous avons pu produire le code suivant :

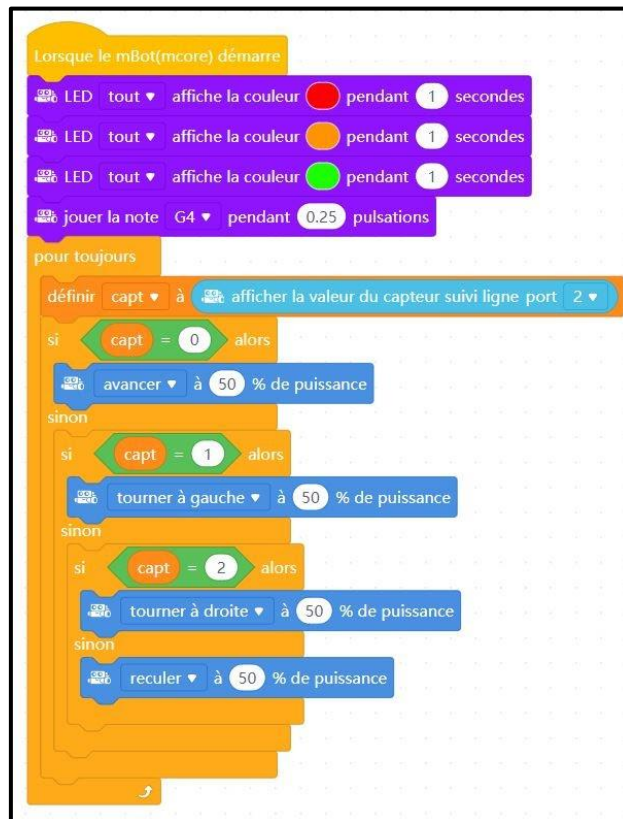


Figure 14 : Code du suiveur de lignes

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

Nous positionnons le mBot sur le parcours et on l’allume. Là, il va s’allumer en rouge, puis orange, puis vert, pour simuler le “A vos marques ! Prêts ? Partez !”. Une petite note va retentir et le robot va démarrer. Tant qu’il trouve la ligne, il ne s’arrête pas.

Nous l’avons testé sur le circuit en forme de 8, présent dans la boîte originelle du mBot. Le robot ne s’arrête donc pas puisque la ligne est infinie. Cependant ce petit schéma nous a permis de vérifier que notre programme fonctionnait bien : pendant les lignes droites centrales, nous pouvions bien apercevoir que les deux capteurs étaient allumés. A l’inverse, quand le robot était dans un virage, un seul des deux capteurs marchait (selon si c’était virage droite ou gauche), ce qui lui permettait de bien rester sur la ligne.

Lors de l'écriture du programme, nous avons rencontré une petite difficulté. Ce n'est pas grand-chose, mais au tout début nous avons mis la vitesse à 100%. Malheureusement, les capteurs n'ont pas le temps de détecter les lignes dans ce cas-là, et le programme ne fonctionne pas correctement. Nous avons donc réduit la vitesse petit à petit, mais le robot se mettait à "brouter" à chaque détection. Il a fallu descendre jusqu'à minimum 50% pour avoir un trajet assez continu.

3.2.2.5 Deuxième programme du robot mBot

Pour ce programme, nous avons voulu recréer le même style de fonctions que la troisième version du robot 2 de 2019. Cependant, le challenge était de réussir à faire sortir le mBot d'un parcours alors qu'il n'a pas de tête rotative pour voir s'il y a un mur sur la droite ou la gauche. Après quelques réflexions, nous avons réussi à faire fonctionner le code suivant :

```

Lorsque le mBot(mcore) démarre
  répéter 10
    définir dist à ultrasonic sensor port 3 distance(cm)
    avancer à 100 % de puissance
    répéter jusqu'à dist < 8
      avancer à 50 % de puissance
      définir dist à ultrasonic sensor port 3 distance(cm)
    stopper le mouvement
    LED tout affiche la couleur rouge pendant 1.5 secondes
    tourner à gauche à 100 % de puissance pendant 0.25 secondes
    définir dist à ultrasonic sensor port 3 distance(cm)
    si dist < 10 alors
      tourner à droite à 100 % de puissance pendant 0.55 secondes
      définir dist à ultrasonic sensor port 3 distance(cm)
    LED tout affiche la couleur verte pendant 1.5 secondes
  
```

Figure 15 : Code de détection d'obstacles

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

Une fois positionné au début du parcours, on allume le robot et il commence par avancer. Lorsqu'il rencontre un mur (distance < 8 cm), il tourne d'un quart de tour vers la gauche systématiquement. Si à ce moment-là, les capteurs ultrasons mesurent une distance inférieure à 10 cm, alors cela veut dire qu'il fallait aller à droite.

Dans ce cas, le robot va tourner d'un demi-tour vers la droite et recommencer à avancer. Grâce à ce système, le robot peut réussir à sortir du parcours.

Malheureusement, ce programme est moins performant que celui du robot avec la tête qui tourne puisqu'en premier lieu il ira systématiquement sur la gauche. Mais cela suffit à sortir d'un parcours simple.

Lors de l'écriture de ce programme, nous avons rencontré une difficulté majeure. En effet, on ne peut pas demander simplement au robot de tourner d'un certain angle. Il faut forcément choisir un temps et une vitesse d'exécution. Ceci nous a compliqué la tâche car nous devons trouver les bons paramètres pour qu'il puisse tourner à 90° correctement.

Cette dernière exigence est essentielle car si le robot n'arrive pas directement face au mur, il ne peut pas détecter correctement la distance à l'obstacle. Après quelques essais nous avons réussi à trouver des options qui marchent plus ou moins bien. Mais là nous avons compris que le mBot ne va, en fait, pas vraiment droit quand il est censé le faire. Il dévie sur la gauche. Ceci a rendu les choses encore plus difficiles, car même si nous avons les bons angles, nous n'étions jamais sûrs à 100% de l'angle/position de départ du robot.

En essayant de le faire partir un peu vers la droite dès le départ, nous avons réussi à le faire sortir du parcours correctement.

4. CONCLUSION

Ce projet nous a beaucoup apporté, sur différents plans.

Tout d'abord, il nous a permis de nous familiariser avec la soudure, lors de l'assemblage des robots. Nous avons également appris à programmer ceux-ci. De plus, nous avons réussi à réaliser les objectifs que nous nous étions fixés.

Ensuite, ce projet nous a permis de renforcer notre capacité à travailler en groupe. En effet, nous avons dû à plusieurs reprises nous répartir les différentes tâches et nous avons tiré profit des qualités des différents membres du groupe afin de mener à bien notre travail.

Pour finir, ce projet nous a tous beaucoup plu. Aucun de nous n'avait de connaissances en robotique à la base, et la découverte de celle-ci nous a beaucoup intéressés. Il nous a donc permis d'avoir un petit aperçu d'un domaine que nous pourrions éventuellement approfondir plus tard.

5. BIBLIOGRAPHIE

- Vidéo sur les capteurs du robot mBot :

<https://www.youtube.com/watch?v=INie493d7oE> (valide à la date du 09/06/2021)

- Vidéo sur le suiveur de ligne mBot :

<https://www.youtube.com/watch?v=YeaHSdYtMHc> (valide à la date du 09/06/2021)

- Informations complémentaires pour le robot Arduino Evolutif :

<https://www.technologieservices.fr/robot-evolution-unoevo-sans-interface-de-programmation-arduino-275850.html> (valide à la date du 09/06/2021)

6. ANNEXES

6.1. Documentation technique

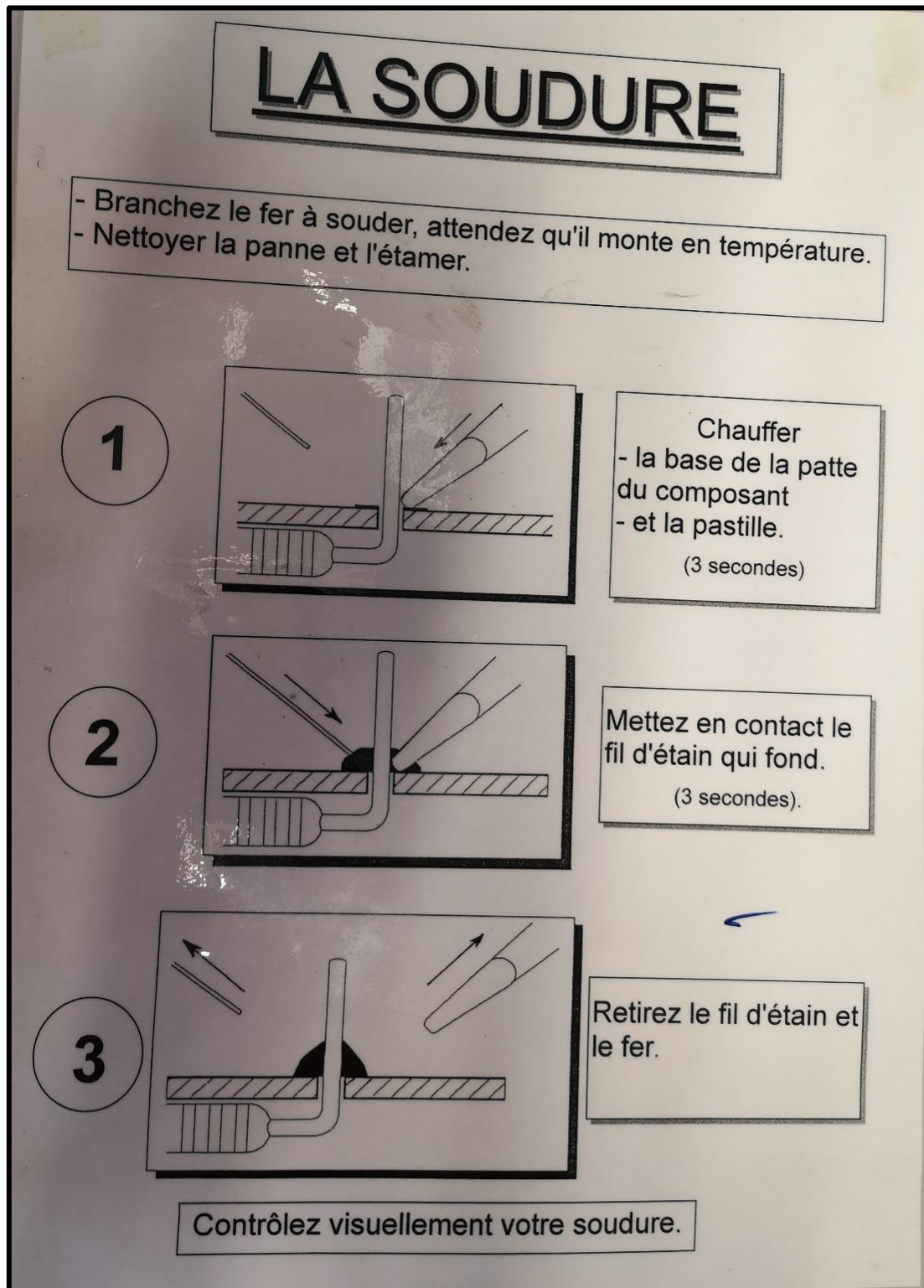


Figure 16 : Document présentant les différentes étapes de la soudure


```

#include <TSServo.h>

TSServo servo_pin_7;
TSServo servo_pin_8;
int _ABVAR_1_a;
void __ardublockDigitalWrite(int pinNumber, boolean status)
{
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, status);
}

void Gauche();
void Avancer();
void Reculer();
void Clignotement();
void Droite();

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

void loop()
{
}

void Droite()
{
    servo_pin_7.write(300, true);
    servo_pin_8.write(1000, true);
}

void Gauche()
{
    servo_pin_7.write(1000, false);
    servo_pin_8.write(300, false);
}

void Clignotement()
{
    for (_ABVAR_1_a=1; _ABVAR_1_a<= ( 2 ); ++_ABVAR_1_a )
    {
        __ardublockDigitalWrite(5, HIGH);
        __ardublockDigitalWrite(6, HIGH);
        delay( 200 );
        __ardublockDigitalWrite(5, LOW);
        __ardublockDigitalWrite(6, LOW);
        delay( 200 );
    }
}

void Avancer()
{
    servo_pin_7.write(1000, false);
    servo_pin_8.write(1000, true);
}

void Reculer()
{
    servo_pin_7.write(1000, true);
    servo_pin_8.write(1000, false);
}

```

Figure 17 : Code du robot Arduino en C/C++

6.2. Schéma de montage

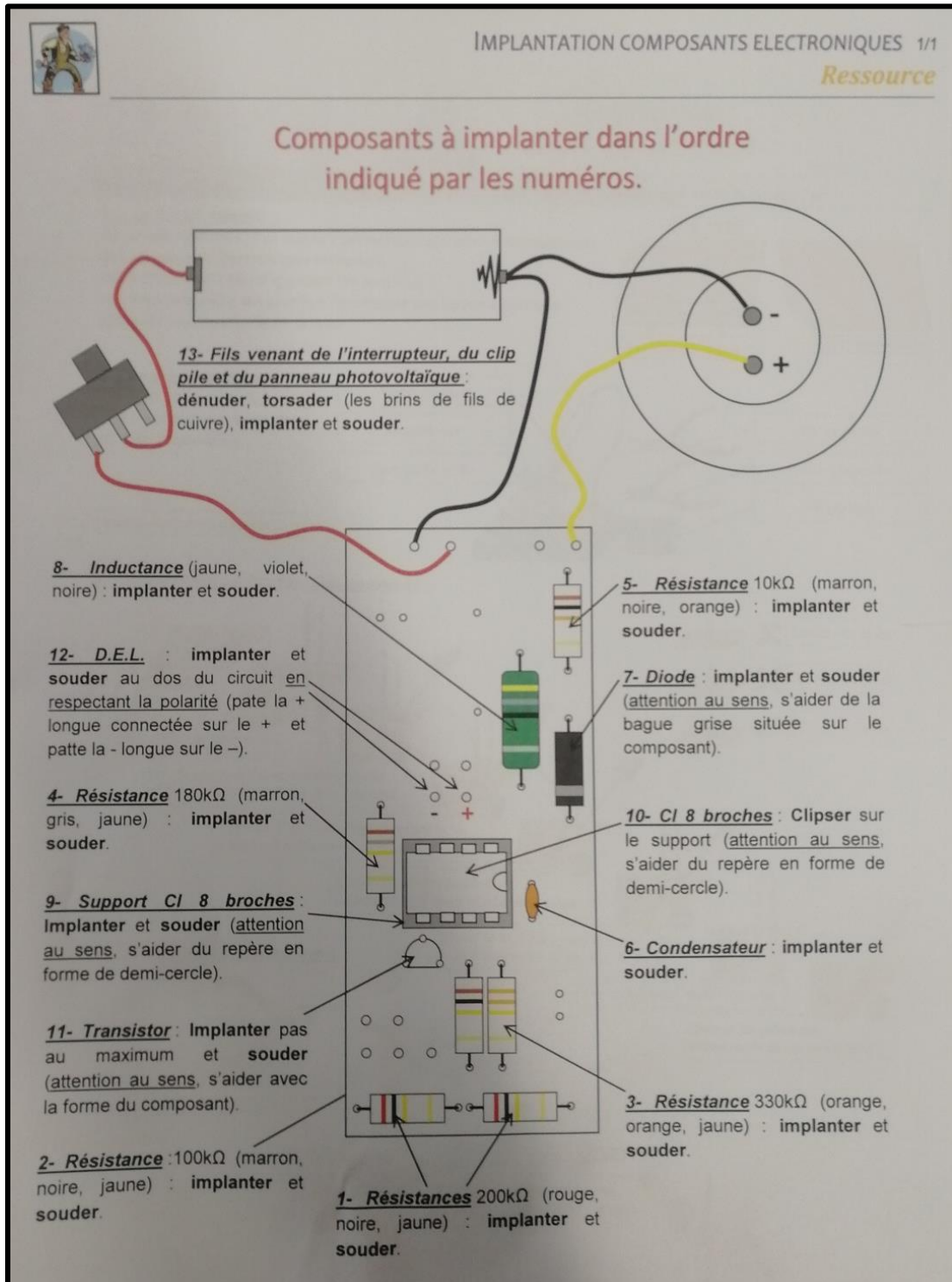


Figure 18 : Schéma de montage sur le circuit imprimé de la lampe solaire