



Les formations sont conçues  
en partenariat entre



Le projet a bénéficié du Programme  
d'Investissements d'Avenir





# PARCOURS

## “ROBOTIQUE”



**D-Clics**  
**NUMÉRIQUES**  
Découvrir Décrypter Diffuser

# INTRODUCTION DES CHOIX ÉDUCATIFS ET PÉDAGOGIQUES

**Notre société est traversée par les usages numériques : monde de données et d'algorithmes parfois complexes, les usages numériques dématérialisés peuvent apparaître désincarnés et, de ce fait, difficiles à appréhender par les enfants et les adolescents. Parce qu'elles permettent d'expérimenter en situation l'influence de ces données sur le monde physique, les activités robotiques sont dans ce contexte d'excellentes portes d'entrée pour accompagner les enfants et les adolescents dans leur éducation numérique. A travers la construction de robots, ils comprennent comment sont construites les machines qui nous entourent et, à travers leur programmation, comment les algorithmes peuvent influencer notre quotidien. Il y a là un véritable vivier d'activités scientifiques et techniques que ce parcours propose d'explorer.**

**De plus, le contexte est à l'utilisation croissante de robots disposant parfois d'une "semi-autonomie", et pour de multiples usages : appui aux activités humaines, secours aux victimes, surveillance, mais aussi parfois comme appui et/ou complément à des interventions militaires.**

**On comprend que la notion de responsabilité citoyenne liée à l'utilisation de robots évolue et pose de nouvelles questions éthiques. La robotique est donc une porte d'entrée pour travailler d'autres questions non seulement scientifiques mais aussi citoyennes. Qui utilise des robots aujourd'hui ? Pour quels objectifs ? Qu'est-ce que cela nous apprend sur la société humaine ?**

**Que sont les objets connectés et l'homme augmenté et quel est leur rapport au numérique que cela implique ?**

**Ce parcours regroupe 10 séances d'activités pédagogiques destinées à la découverte de la robotique et de la programmation auprès des enfants de 8 à 16 ans. Chaque activité est prévue pour une séance d'environ d'une heure.**

## Pédagogie

La pédagogie utilisée met l'enfant au centre du processus d'apprentissage. L'animation se fait par l'expérimentation et par la méthode d'essais-erreurs. En passant par la robotique nous faisons le lien entre le monde numérique et le monde physique. Il est possible de visualiser directement les résultats de nos actions de programmation ce qui a un effet positif sur l'acquisition structurée des connaissances.

La robotique constitue un moyen ludique pour libérer le potentiel créatif des enfants et des adolescents (en matière graphique, corporelle, logique...). Elle développe l'imaginaire des enfants. Elle les invite à réfléchir sur leur place dans une société qui se transforme rapidement.

Nous avons également voulu favoriser le travail en équipe, le partage et la communication, et permettre une première sensibilisation à la gestion de projets. Via la robotique, les enfants expérimentent la réalisation d'un projet dans le monde réel et appréhendent ainsi son caractère complexe. Le collectif prend alors tout son sens : ensemble, on va plus loin.

La robotique est donc une vraie entrée éducative pour la formation de futurs citoyens créatifs, critiques et responsables de leurs usages numériques.

## Do It Yourself

La culture du "Do It Yourself" renvoie à l'image du "bricoleur" mais ne s'arrête pas là. Plus profondément, il s'agit d'être acteur et non pas consommateur, d'être créatif, de faire partie d'une communauté de partage de connaissances. Cette philosophie met en avant la culture du bien commun (libre, open source) et forme des citoyens responsables dans une société où chacun/e doit peut y trouver sa place.

## Le rôle de l'animateur/trice

Afin de mettre en place ce parcours et réussir ses objectifs, l'animateur/trice doit incarner le rôle de l'accompagnateur. L'animateur/trice, ne maîtrisant pas tous les projets mis en place par les enfants, apparaît comme un médiateur plus qu'un "sachant". Son rôle est donc d'aider les enfants à surmonter les difficultés rencontrées et poursuivre le travail qu'eux mêmes ont mis en place. L'autonomie des enfants est mise en avant.

L'animateur/trice est également libre à adapter le parcours à son groupe d'enfants (âge, niveau de compétences informatiques, temps...). Il/elle peut changer l'ordre de séances et/ou d'activités.

## Choix du robot

Le kit robotique utilisé dans notre parcours est basé sur l'Arduino. Nous l'avons conçu avec l'apport de l'association la Maison du Libre à Brest.

## Arduino :

- est une carte électronique, programmable et open source
- A un prix raisonnable
- dispose d'un logiciel open source et gratuit, développé en Java.
- est compatible avec toutes les plateformes (Windows, Gnu/Linux et Mac OS).
- est accompagnée d'une communauté très développée : des forums d'entre-aide, de présentations de projets, de propositions de programmes et de bibliothèques, ...
- Et d'un site en anglais [arduino.cc](https://www.arduino.cc) et un autre en français [https://www.flossmanualsfr.net/arduino/ch033\\_ressources-en-ligne](https://www.flossmanualsfr.net/arduino/ch033_ressources-en-ligne) où on trouve la référence Arduino, le matériel, des exemples d'utilisations, de l'aide pour débiter, des explications sur le logiciel et le matériel, etc.

## Choix du langage informatique

Des outils d'accompagnement spécifiques sont proposés dans le parcours avec le langage Arduino. Cependant, nous avons souhaité laisser la liberté à chaque animateur/trice voire chaque enfant de choisir le langage informatique qui lui convient.

## Les langages proposés dans ce parcours sont libres et gratuits :

- **Arduino** : langage textuel proposé par les créateurs de la carte Arduino. Langage qui se rapproche du C/C++.
- **Scratch** : langage de programmation visuel<sup>1</sup>. Idéal pour les plus jeunes (8-10 ans).
- **Blockly** : programme web permettant la programmation visuelle pour Arduino. Ce logiciel est un outil de qualité pour démarrer facilement sur Arduino sans connaissances en programmation. L'utilisation d'un environnement de programmation graphique permet des résultats rapides quitte à passer progressivement au codage traditionnel.

## Découvrir, Décrypter, Diffuser

Le présent parcours répond à ces objectifs et à ces ambitions. Partie intégrante du projet "**D-Clics numériques**", il reprend sa dynamique pédagogique, et vise successivement à faire découvrir, décrypter, transmettre.





## DÉCOUVRIR

Nous intégrons dans cet axe l'ensemble des activités d'initiation, de deux types :

Celles qui conduisent à faire connaître aux enfants et aux jeunes la richesse de la robotique, tant des points de vue techniques et artistiques, que des usages.

Celles qui permettent de découvrir les bases de la robotique et de la programmation simple des robots. Elles doivent également donner confiance aux enfants et aux jeunes dans leur capacité à construire et programmer des robots plus complexes.



## DÉCRYPTER

Nous sommes convaincus que pour décrypter et comprendre le monde, il faut autant disposer de connaissances et de clefs de compréhension qu'être mis en situation active par la pratique. Nous intégrons dans cet axe les activités qui permettent progressivement aux enfants et aux jeunes d'expérimenter et de découvrir leur potentiel créatif.



## DIFFUSER

Nous accordons une grande importance aux cultures du libre, des biens communs, et de l'intelligence collective, qui nous semblent devoir être transmises aux enfants et aux jeunes. Ainsi, nous proposons des activités qui leur permettent d'agir ensemble et de partager leurs créations (en vue de les faire découvrir comme de les voir améliorées par d'autres), y compris à l'extérieur de l'espace éducatif dans lequel ils ont bénéficié de l'activité.



<sup>1</sup> Un langage de programmation graphique ou visuel est un langage de programmation dans lequel les programmes sont écrits par assemblage d'éléments graphiques.

# UN PARCOURS QUI INTÈGRE LES GRANDS ENJEUX DE L'ÉCOLE DU SOCLE COMMUN

Le parcours « Robotique et DIY », partie intégrante du projet de D-Clics numériques, répond transversalement aux enjeux de l'école du socle commun, comme définis au décret n° 2015-372 relatif au socle commun de connaissances, de compétences et de culture du Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche.

Notre parcours répond directement aux enjeux correspondant à plusieurs domaines du socle commun :



## “LES LANGAGES POUR PENSER ET COMMUNIQUER”

Il vise entre autres l'apprentissage des langages informatiques et des médias. « Ce domaine permet l'accès [...] à une culture rendant possible l'exercice de l'esprit critique. Il implique la maîtrise de codes ; de règles, de systèmes de signes et de représentations. » (Décret n° 2015-372). Au travers du projet D-Clics numériques, l'élève prend conscience du fait que les langages informatiques sont utilisés pour programmer des outils numériques comme des robots et réalise des traitements automatiques de données. Il apprend les principes de base de l'algorithme et de la conception des programmes informatiques. Il les met en pratique pour programmer un robot en utilisant différents logiciels et hardwares libres et ouverts à tous (ex. Arduino, Scratch...). L'élève s'exprime également en utilisant la langue française à l'oral et à l'écrit en fin de parcours pour la présentation de son projet en groupe.



## “MÉTHODES ET OUTILS POUR APPRENDRE”

Autrement dit, comment apprendre à apprendre. Ce domaine vise entre autres un enseignement explicite des outils numériques. « La maîtrise des méthodes et outils pour apprendre [...] favorise l'implication dans le travail commun, l'entraide et la coopération » (Décret n° 2015-372), valeurs partagés également par le projet D-Clics numériques. Plus précisément, l'élève sera amené à pratiquer la programmation en découvrant différents langages informatiques, en expérimentant, en passant par une démarche d'essais et erreurs. Il sera ensuite amené à co-construire un projet en équipe pour la création et programmation d'un robot. Il apprend ainsi à travailler en équipe, partager des tâches, s'engager dans un dialogue constructif, négocier, chercher un consensus, gérer un projet, planifier les tâches et évaluer l'atteinte de ses objectifs.



## “ SYSTÈMES NATURELS ET DES SYSTÈMES TECHNIQUES ”

Ce domaine vise à développer la curiosité, le sens de l'observation et la capacité à résoudre des problèmes. Un de ses objectifs principaux est de donner à l'élève les fondements de la culture technologique. A travers des discussions autour des robots qui nous entourent dans notre quotidien, les expériences de l'élève et les questions éthiques sur l'utilisation des robots aujourd'hui, on mène la réflexion sur ses vécus et connaissances déjà en place mais en même temps on éveille sa curiosité, son envie de se poser des questions, de chercher des réponses et d'inventer. Le projet final proposé dans le parcours ici présenté, pour la programmation des robots par les élèves, les familiarise avec le monde technique, numérique mais aussi physique. Ils doivent d'abord comprendre la programmation et les algorithmes et ensuite être capables de concevoir et réaliser eux-mêmes des projets en groupe. Ces activités « sont des occasions de prendre conscience que la démarche technologique consiste à rechercher l'efficacité dans un milieu contraint (en particulier par les ressources) pour répondre à des besoins humains, en tenant compte des impacts sociaux et environnementaux. » (Décret n° 2015-372) car pour la personnalisation des robots les enfants utiliseront des matériaux de récupération afin de limiter les déchets, l'impact sur l'environnement et favoriser l'imagination.



## “ REPRÉSENTATIONS DU MONDE ET D'ACTIVITÉ HUMAINE ”

Ce domaine est consacré à la compréhension des sociétés dans le temps et l'espace. Les diverses discussions autour du sujet du numérique, de l'évolution de la robotique, de l'identité numérique, de la programmation visent une meilleure compréhension de la société dite numérique et du monde contemporain dans lequel on vit. « Ce domaine vise également [...] la construction de la citoyenneté en permettant à l'élève d'aborder de façon éclairée de grands débats du monde contemporain » (Décret n° 2015-372).

S É  
A N



**Retrouvez, ci-après, les fiches séances qui vous permettront de pratiquer, avec les enfants, l'ensemble des activités proposées dans le cadre de ce parcours "Robotique".**

## **SOMMAIRE**

### **TRONC COMMUN**

<b>10</b>	<b>SÉANCE 1</b>	ANNEXES 1, 2, 3, 4, 5, 6
<b>12</b>	<b>SÉANCE 2</b>	ANNEXES 7, 8, 9, 10, 11
<b>13</b>	<b>SÉANCE 3</b>	ANNEXES 12, 13, 14, 15, 16
<b>14</b>	<b>SÉANCE 4</b>	ANNEXES 17, 18, 19, 20
<b>16</b>	<b>SÉANCE 5</b>	ANNEXES 11, 12, 21, 22
<b>17</b>	<b>SÉANCE 6</b>	ANNEXES 4, 23, 24, 25, 26
<b>19</b>	<b>SÉANCE 7</b>	ANNEXE 26
<b>20</b>	<b>SÉANCE 8</b>	ANNEXE 22
<b>21</b>	<b>SÉANCE 9-10</b>	ANNEXES 27, 28, 29

# SÉANCE 1 1/2

Durée – 1 heure

## QUESTIONS D'USAGES :

permettre aux enfants de réfléchir à la notion de robot. Qu'est ce qu'un robot ? Quels sont les organes, les fonctions, les éléments qui constituent un robot ? À quoi sert un robot ?

### Objectifs pédagogiques

Repérer, identifier et différencier :

- les principes techniques :
  - Un robot est un système automatisé.
- les types de robots : Les robots mêmhanocentrés (rover), les robots anthropocentrés (humanoïdes), les robots zoocentrés ou biocentrés (ornithoptères)
- les pratiques et usages liés à la robotique :
  - Milieus hostiles à l'homme, médecine, industrie, loisirs, usages domestiques, etc.

### Connaissances requises par les enfants pour aborder la séance

Aucune en particulier

### Éléments d'évaluation des acquis enfants

Les enfants sont capables de citer les robots qu'ils ont déjà vus dans leur périmètre proche et identifier leur type.

### Déroulement

#### 1 Temps de rencontre et présentation de l'atelier 10 min

L'animateur/trice peut demander aux enfants auparavant de réfléchir à l'atelier dans sa globalité.

#### 2 Un robot: Qu'est ce que c'est ? - 50 min (cf. 1,6)

L'animateur/trice prépare les ordinateurs avec l'environnement de développement Arduino et teste avec le code blink en amont (cf. 2). Il/elle monte un robot (cf. 3).

##### a. Démonstration d'un robot qui bouge - 10 min

L'animateur/trice montre aux enfants un robot ROSA déjà réalisé (voir <https://wiki.mdl29.net/doku.php?id=projets:robotarduino> pour le montage).

ROSA réalisera des mouvements simples comme : aller tout droit, reculer, s'arrêter.

Questionner les enfants : de quel type de robot s'agit-il ?

Peut-on interagir avec lui ? Leur demander d'identifier son capteur, contrôleur, actionneur, énergie, pièces d'assemblage.

##### b. Discussion autour de la robotique - 15 min

A partir de cette démonstration une discussion et réflexion autour de la robotique peuvent être lancées. Demander aux enfants de donner leur représentation lorsqu'ils entendent le mot robot.

À quoi sert un robot ? Que peut faire un robot ? Dans quel milieu un robot peut intervenir ? À quoi ressemble un robot ? L'animateur/trice classera les idées des enfants en fonction des types différents des robots qui apparaissent (cf. 4).

Pour les plus jeunes l'animateur/trice peut utiliser une catégorisation plus simple comme la suivante: robots industriels, ménagers, jouets, humanoïdes, sous-marins, volants, roulants...

##### c. Représentation des robots - 25 min

Donner aux enfants du papier et des crayons et les laisser imaginer ce que serait leur robot. Possibilité de dessiner un robot avec les morceaux différents proposés par différents groupes d'enfants. Penser à afficher puis récupérer les dessins effectués par les enfants (ils seront réutilisés dans d'autres séances ultérieures).

*Proposition:* Afin de présenter à la fin du parcours (séances 9 et 10), les projets des enfants il faudra en début de projet prévoir un "reporter" dans le groupe pour mettre en oeuvre les pré-requis suivants:

- prise de photos à chaque étape, tout au long du projet,
- copies d'écran lors de la programmation,
- éventuellement captation vidéo tout au long du projet (essais durant le projet, interviews, etc.).

L'ensemble des productions écrites réalisées lors du projet doivent être réservées par l'animateur/trice : croquis, dessins, schémas. **Concernant les productions papier, il est souhaitable que l'animateur/trice les numérise** en les scannant ou en les prenant en photo.

*Proposition:* On peut également commencer par le temps c, ensuite b pour arriver au temps a. L'objectif étant de ne pas influencer les représentations des enfants.

### Aménagement et matériel spécifique nécessaire

- Ordinateur
- Vidéo projecteur
- Tableau
- Grandes feuilles de papier
- Stylo feutres de couleur
- Papiers
- Crayons
- Post it

### Boîtes à outils: ressources

- (1) Introduction robotique
- (2) Didacticiel sur comment installer Arduino.exe et tester avec Blink (pour Windows, Linux, Mac) p.28
- (3) Didacticiel sur comment monter le kit robotique p.28
- (4) Les différents types de robots
- (5) Vidéos indiquant les différents aspects des robots
- (6) Les robots

# SÉANCE 1 2/2

Durée – 1 heure

## Et si je dispose de plus de temps ?

- Ne pas montrer ROSA dès la première séance pour laisser les enfants imaginer des différents types de robots et avoir leur propre représentation des robots. Cela donnera plus de temps pour continuer des discussions autour de la robotique. Quelques idées : automates contre robots, les lois de la robotique pour aborder de questions autour de citoyenne et robotique, l'histoire de la robotique pour parler du premier robot, la robotique de demain pour parler de l'intelligence artificielle... (cf. 1)
- Déconstruire les représentations.  
En montrant plusieurs petits films (cf. 5), l'animateur/trice montrera qu'un robot peut avoir plusieurs formes et plusieurs fonctions (parfois simples parfois plus complexes). Analyser à partir de plusieurs vidéos, ce que fait la machine.
- Aller voir sur le site de Boston Dynamics, le site de la DARPA, le site de l'INRIA

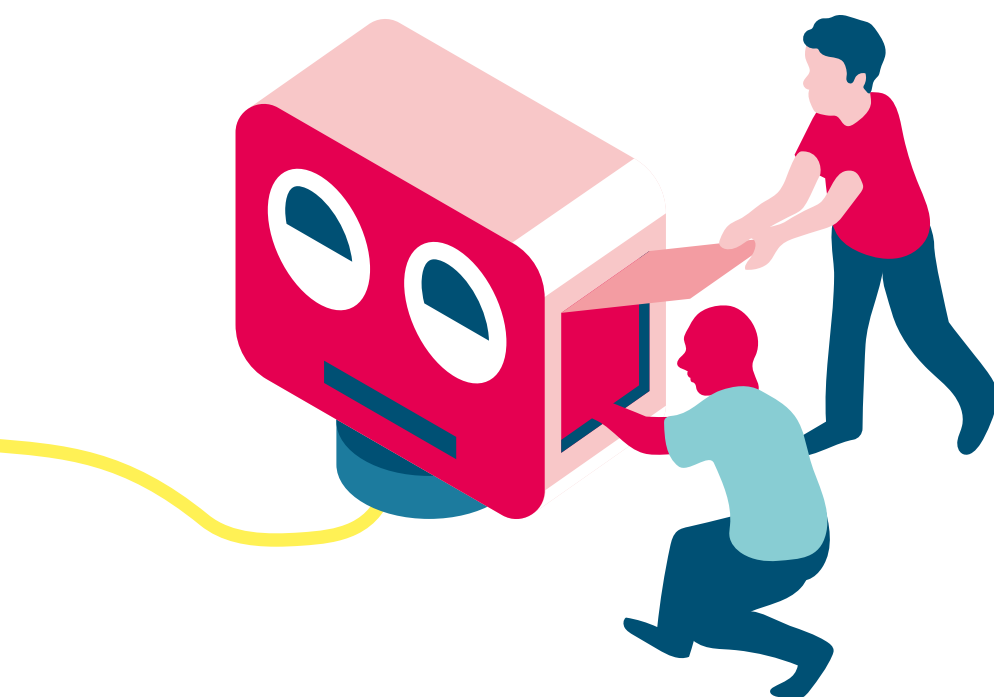
## Capacités de l'animateur

### – Gestion du groupe

L'animateur/trice aura au préalable visiter les sites de robotique présentés dans la fiche. Il conviendra de faire une veille concernant les robots et leurs concepts associés.

## Aller plus loin - autres ressources

- L'animateur/trice qui a besoin de plus d'informations sur les notions de la programmation et robotique peut suivre **le MOOC de Class'Code** :  
<https://pixees.fr/classcode/accueil/#moocs>
- Ainsi que sur le livre "1,2,3...Codez" :  
<http://www.fondation-lamap.org/fr/123codez> dont un extrait autour de la robotique se trouve à l'annexe 6.





# SÉANCE 2

Durée – 1 heure

## MONTAGE DU CHÂSSIS DE ROSA (ROBOT OPEN SOURCE ARDUINO):

L'animateur/trice accompagnera les enfants au montage du châssis du robot.

### Objectifs pédagogiques

- Mettre en exergue la partie bricoleur et Do It Yourself
- Comprendre les mouvements de base (avancer, reculer)
- Comprendre le fonctionnement d'un moteur

### Connaissances requises par les enfants pour aborder la séance

Avoir eu une première réflexion sur ce qu'est un robot.

### Éléments d'évaluation des acquis enfants

Avoir monté le châssis de ROSA.

### Déroulement

#### 1 Montage du châssis - 60 min

L'animateur/trice amène avec lui son robot déjà monté pour que les enfants puissent visualiser le produit fini et se motiver pour le montage de leurs robots.

**Option 1 :** Chaque enfant a son kit robotique.

Une fois que l'enfant a fini le montage, il/elle peut accompagner les autres. L'animateur/trice imprime la documentation pour chaque robot.

**Option 2 :** Pour mettre en avant l'esprit d'équipe et la coopération, l'animateur/trice crée des équipes. Par rapport au nombre d'enfants et de kits robotiques, on peut faire des équipes de 2 à 4 enfants par robot. L'animateur/trice imprime la documentation pour chaque groupe. Les enfants montent le kit avec l'aide de l'animateur/trice (cf. 7)

*Proposition 1 :* L'animateur/trice peut mettre la vidéo du montage du châssis dans un ordinateur par groupe. L'image est parfois plus ludique qu'un document papier. Les enfants sont ainsi plus autonomes.

*Proposition 2 :* L'animateur/trice peut diffuser avec un vidéo projecteur la vidéo du montage comme un outil de repère - référence commune. Cela permettra aux enfants de suivre les séquences et aux animateur/trices de ne pas avoir des grands écarts entre les enfants qui finissent rapidement et d'autres qui ont besoin de plus de temps.

*Proposition 3 :* Si les enfants sont très jeunes pour monter le robot, l'animateur peut monter les robots en avance et faire le câblage en laissant de côté le capteur de distance.

### Aménagement et matériel spécifique nécessaire

- 1 pile 9V ou 4,5V
- Un ordinateur par groupe
- Un vidéo projecteur idéalement
- Documentation imprimée

### Boîtes à outils : ressources

- (7) Didacticiel (vidéo) sur le montage du châssis p.28
- (8) Capteurs et actionnaires
- (9) Explication de mouvements simples
- (10) Déplacement du robot p.28
- (11) Test des moteurs avec une pile p.28

### Aller plus loin – autres ressources

- Si vous souhaitez utiliser un autre robot pour faire le parcours, vous pouvez consulter les ressources faites par Fréquence École et Zoomacom sur **Primo**: <https://frequence-ecoles.org/ressources-2/primopour-decouvrir-les-bases-de-la-programmation-dun-robot/>
- **Thymio**: <https://frequence-ecoles.org/ressources-2/thymio-pour-apprendre-le-code-et-developper-un-esprit-logique/>
- **Cubelet**: <https://frequence-ecoles.org/ressources-2/cubelets-pour-aborder-les-notions-de-logique-et-d-observation-scientifique/>
- **Robot insecte**: <https://wiki.mdl29.net/doku.php?id=les-petitshackers:robotsinsecte>
- **RoboduLAB**: <http://obcprovence.com/2015/05/robodulab-robot-pedagogique-pour-apprendre-a-coder/>

### Et si je dispose de plus de temps ?

- Explication des mouvements du robot aux enfants. Qu'est-ce qu'une rotation, translation. (cf. 9) Quels déplacements peut faire le robot ? (cf. 10)
- Expliquer le sens de rotation d'un moteur grâce à une pile. (cf. 11)

### Capacités de l'animateur

#### - Gestion du groupe

- L'animateur/trice, avant la séance, se prépare en lisant et faisant des tests:
- Pour différentes parties de ROSA ( cf. 8)
  - Pour des questions de mouvements simples d'un moteur (cf. 9,10)
  - Pour tester les moteurs avec une pile (cf. 11)

# SÉANCE 3

Durée – 1 heure

## MONTAGE ÉLECTRONIQUE DE ROSA :

L'animateur/trice accompagnera les enfants lors du montage électronique du robot en leur présentant également des bases de l'électronique.

### Objectifs pédagogiques

- Repérer, identifier et différencier:
- La partie électronique du montage d'un robot
  - La carte électronique Arduino
  - Le lien entre la programmation et sa traduction en mouvements physiques

### Éléments d'évaluation des acquis enfants

Les enfants sont capables de faire le montage électronique de leur robot.

### Connaissances requises par les enfants pour aborder la séance

- Connaissances manuelles de bases (e.x. assembler, couper...)
- Pouvoir manipuler des objets fragiles

### Déroulement

L'animateur/trice regarde en amont la fiche sur la carte Arduino ([www.arduino.cc](http://www.arduino.cc)) (cf. 12,13) pour pouvoir répondre aux questions des enfants si besoin.

#### 1 Montage électronique - 30 min

Les enfants montent la partie électronique de ROSA avec l'aide de l'animateur/trice, de la documentation et des vidéos (cf.14)

Montage de la carte Arduino, du contrôleur de moteurs, du capteur de distance et branchement des câbles.

**Pédagogie:** un modèle d'apprentissage type essai/erreur peut être mis en place, l'animateur/trice peut accompagner le câblage du robot. Par exemple, si je branche le fil rouge; le moteur s'actionne.

*Proposition 1:* L'animateur/trice peut mettre la vidéo du montage du châssis dans un ordinateur par groupe. L'image est parfois plus ludique qu'un document papier. Les enfants sont ainsi plus autonomes. ET/OU

*Proposition 2:* L'animateur/trice peut diffuser avec un vidéo projecteur la vidéo du montage comme un outil de repère-référence commune. Cela permettra aux enfants de suivre les séquences et aux animateur/trices de ne pas avoir des grands écarts entre les enfants qui finissent rapidement et d'autres qui ont besoin de plus de temps.

#### 2 Atelier déconnecté - 30 min

Après avoir finalisé la structure de base de ROSA, on peut commencer à s'initier aux notions de base de la programmation. Pour cela, faisons un jeu (cf. 15). En donnant le rôle du robot à un enfant, il doit réaliser plusieurs actions. Les enfants ou l'animateur/trice donnent les consignes écrites ou orales.

- Se déplacer d'un point A vers un point B (donner une consigne courte et simple en nombre de pas)
- Sortir de la pièce (donner une consigne plus complexe avec plusieurs actions les unes enchaînées avec les autres). Pensez à organiser la classe.
- Se déplacer d'un point A vers un point B en mettant un obstacle entre les deux (reprendre le premier programme et tester lorsque l'enfant entre en contact avec l'obstacle).

**d.** Mettre en place la notion de Si Alors en demandant à l'enfant robot de se servir de ces bras pour récupérer des informations de son environnement. Par exemple, "si tu touches le mur, tourne à gauche". Nous pouvons initier ainsi la notion des capteurs d'un robot (proposer une suite d'actions en fonction de l'environnement dans lequel je me trouve).

**L'objectif est de faire le lien entre la programmation et les mouvements physiques du robot.**

### Aménagements et matériel spécifique nécessaire

- Carte Arduino UNO
- Documentation imprimée

### Et si je dispose de plus de temps ?

Aller plus loin sur l'explication des notions de moteurs, capteurs, contrôleur de moteurs...

### Boîte à outils : ressources

- (12) Présentation Arduino
- (13) Présentation PPT Arduino
- (14) Montage électronique
- (15) Robot-idiot (fiche faite par INRIA) p.28
- (16) Les lois simples de l'électricité p.28

### Aller plus loin – autres ressources

- Débuter avec l'Arduino  
<https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/>
- **Ressources Arduino et composants:** [https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:arduino\\_dossier\\_resource.pdf](https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:arduino_dossier_resource.pdf) (Ressource proposée par la Maison du Libre)
- **Construction d'un robot autonome:** <https://wiki.mdl29.net/doku.php?id=lespetitshackers:robots> (Ressource proposée par la Maison du Libre)
- **FlossManuals.** Manuels libres pour logiciels libres: <http://fr.flossmanuals.net/arduino/historique-du-projet-arduino/>

### Capacités de l'animateur

#### - Gestion du groupe

- Si l'animateur/trice n'est pas "expert" en Arduino, se rassurer ! Il faudra bien préparer la séance en avance en consultant toutes les ressources données dans la boîte à outil et les avoir testées avant la mise en pratique de la séance avec les enfants.
- Faire une lecture du document sur les principes de l'électricité (cf. 16)

# SÉANCE 4 1/2

Durée – 1 heure

## DÉCOUVRONS LES BASES DE LA PROGRAMMATION

L'animateur/trice initie les enfants à la logique de la programmation et aux concepts informatiques comme les algorithmes.

### Objectifs pédagogiques

- Comprendre les logiques de la programmation (algorithme, boucles, conditions, variables)
- Comprendre le lien entre les mouvements du robot et l'informatique.

### Connaissances requises par les enfants pour aborder la séance

Avoir suivi les séances précédentes.

### Éléments d'évaluation des acquis enfants

Les enfants sont capables de créer des algorithmes cohérents.

Ils peuvent déduire ou faire des hypothèses sur des lignes de code pour faire bouger un robot et utiliser les éléments de langage liés à l'algorithme.

### Déroulement

#### 1 Comment communiquer avec l'ordinateur - 40 min

L'animateur/trice peut emmener son robot et le mettre en mouvement. A partir de cette démonstration la discussion autour de la programmation commence.

- 15 min Lancer une discussion avec les enfants autour de ce qu'est un ordinateur, de la programmation et le langage informatique. (cf. 17)
- 15 min Continuer la discussion autour de la programmation en regardant plus précisément ce qu'est un algorithme et un organigramme de programmation (cf. 17, 18)

**Activité :** Si on voulait créer un robot qui soit bon perdant, comment nous y prendrions-nous ? (cf. 19)  
Écrire avec les enfants l'organigramme de programmation correspondant ou préparer des losanges et des rectangles en carton pour créer l'algorithme en 3D. À travers de cette activité les enfants apprennent aussi comment jouer ensemble et ils s'interrogent sur le comportement qu'on doit avoir pour vivre ensemble.

*Proposition :* Pour les plus jeunes l'animateur/trice peut utiliser l'organigramme de programmation sur la récréation (cf. 20) et ne pas introduire le concept de la variable.

#### 2 Programmer un robot - 20 min

**a. 10 min** - Sous forme d'échange avec le groupe aborder les sujets suivants :

- Qu'est ce qui fait bouger un robot ?
- Où se trouvent les ordres dans un robot ? (Caché dans le programme)
- Qui les gère ? (l'appareil qui fait tourner le robot)
- Qui les a écrites ? (Le créateur du robot / celui qui l'a programmé)
- Est-ce qu'on les connaît ? (On est au courant de certaines ordres qui sont visibles (ce qui fait avancer, reculer le robot) mais on n'a pas accès à tous les ordres, qui ne sont pas visibles ou pas activés... sauf à avoir accès au "code source" et à savoir le lire !)

**b. 10 min - Activité :** Échanger avec le groupe sur les mouvements que les robots, faits par les enfants, pourraient faire. Création d'un tableau où on liste les mouvements. A la fin de cette séance la notion à transmettre c'est que les ordres auxquels un robot obéit sont son code. Le robot c'est l'exécutant.

### Aménagements et matériel spécifique nécessaire

- Tableau ou paper board
- Papiers crayons
- Impression des fiches annexes

### Boîte à outils : ressources

- (17) Fiches ressources  
"La programmation expliquée aux enfants" (outil proposé par BSF dans le cadre de l'opération "les voyageurs du code")
- (18) Algorithme VS organigramme de programmation
- (19) Organigramme "le bon perdant"
- (20) Organigramme "la récréation"

# SÉANCE 4 2/2

Durée – 1 heure

## Et si je dispose de plus de temps ?

- Laisser les enfants créer leurs propres algorithmes : le bon perdant, le bon élève... Tout comportement prévisible peut faire l'objet d'un tel algorithme.  
*Attention :* éviter tout exemple visant les enfants pour ne pas créer un phénomène de bouc émissaire. Nous essayons de valoriser plutôt des comportements positifs que négatifs.
- Faire le parcours D-Clics numériques de l'application **www.declick.net** (faite par **Colombbus**) pour apprendre la programmation aux enfants. Vous pouvez faire une partie (quelques minutes) ou tous les exercices (4 heures) en fonction du temps disponible et l'âge des enfants.
- Avec les plus grands, on peut aller plus loin sur la réflexion autour des questions d'éthique de la robotique. Certains robots ont la capacité de capter des données personnelles (photos ou vidéos de personnes...), comme les robots de surveillance. Leur déploiement soulève alors des questions liées à la protection de la vie privée et des données personnelles. La robotique est donc une porte d'entrée pour travailler d'autres questions pas seulement scientifiques mais aussi citoyennes. Qui utilise des robots aujourd'hui? Pour quels objectifs? Qu'est-ce que cela signifie de la place de l'humain dans la société?

## Capacités de l'animateur

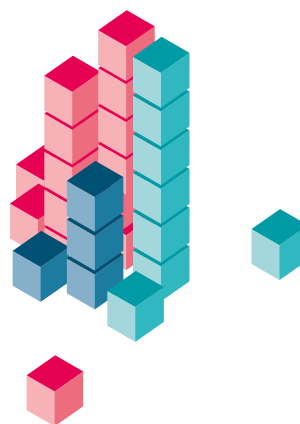
### - Gestion du groupe

- Cette séance d'animation est très théorique, il est donc très important de l'animer pour que ça ne soit pas trop scolaire.
- Faites participer les enfants et tous les enfants !
- Ré-appropriiez-vous les fiches d'explication pour les adapter à votre contexte et votre public.

## Aller plus loin - autres ressources

### Quelques jeux de société:

- **You Robot** - ReposProduction (environ 20€).  
Jeu de mime. Durée moins de 30 minutes. Une équipe de robot, une équipe de concepteur. Donner les bonnes consignes pour que les robots fassent ce qui leur est demandé.
- **Ricochet Robot** - Rio Grande Game (environ 35€).  
Jeu de plateau à partir de 10 ans. L'objectif est clairement de réfléchir chacun dans son coin et plus vite que les adversaires à l'algorithme qui permettra d'amener le robot sur son objectif.
- **Initiation à la robotique** (environ 30€). 3 robots à construire qui avancent en fonction de leur sensibilité: 1 robot sensible à la lumière, 1 robot sensible aux obstacles, 1 robot sensible à la technologie infrarouge d'une télécommande, 3 façade interchangeable pour 3 styles de robots.



# SÉANCE 5

Durée – 1 heure

## INITIATION À LA PROGRAMMATION AVEC ARDUINO

L'animateur/trice présente aux enfants la carte Arduino et les initie à la programmation du robot avec le langage Arduino.

### Objectifs pédagogiques

- Apprendre à programmer avec un langage textuel (Arduino)

### Éléments d'évaluation des acquis enfants

Les enfants sont capables d'installer le code Robot\_dclics sur leurs cartes Arduino.

### Connaissances requises par les enfants pour aborder la séance

Avoir compris les bases et les notions de la programmation (algorithmes, boucles, conditions).

### Déroulement

L'animateur/trice a déjà installé et préparé les ordinateurs. Si chaque enfant a monté son robot, vous pouvez, à ce stade là, faire des groupes de 2 à 3 enfants et donner 1 ordinateur par groupe. Cela incitera le travail en équipe et résoudra le problème d'équipement en ordinateurs.

#### 1 Présentation de la carte Arduino - 10 min

L'animateur/trice a lu auparavant les fiches sur Arduino (cf. 11,12) pour pouvoir la présenter rapidement et répondre aux questions des enfants.

#### 2 Tester le fonctionnement du capteur de distance 20 min

Nous testons tous ensemble le fonctionnement du capteur qui calcule la distance. L'animateur/trice projette son écran et les enfants suivent la procédure pour installer la bibliothèque NewPing.zip et la téléverser à la carte Arduino. (cf. 21)  
L'objectif est de faire l'expérience de l'installation d'une bibliothèque dans une carte Arduino sans passer par la partie théorique et des notions complexes. Dans le cas où chaque enfant a son robot, l'animateur/trice doit prévoir le temps pour que tous les membres de l'équipe puissent tester le capteur de distance de leurs robots.

#### 3 Le code de ROSA - 30 min

Les enfants installent dans leurs ordinateurs le code complet de ROSA (cf. 22) en suivant les instructions de l'animateur/trice qui s'affichent sur l'écran du vidéoprojecteur. Une fois le code téléversé dans la carte Arduino, ROSA avance, détecte un obstacle, s'arrête, recule, change de direction et recommence.

*Attention :* Les deux moteurs n'avancent pas toujours à la même vitesse. Il peut donc être nécessaire de recoder les lignes qui définissent la vitesse des moteurs. Il est important que les enfants aient du temps pour jouer avec ROSA et tester le code.

*Proposition :* L'animateur/trice peut accompagner la découverte progressive du code. On monte dans un premier temps, ROSA sans y ajouter le capteur de distance. On remarque que ROSA avance et recule mais n'est pas capable de détecter les obstacles. On demande aux enfants ce qu'il lui manque. La réponse est « des yeux » soit un capteur de distance. On retourne aux séances 3 et 4 pour monter le capteur de distance. ROSA est maintenant capable de détecter les obstacles !

### Aménagements et matériel spécifique nécessaire

- 1 ordinateur par groupe
- Installation logiciel Arduino
- Robots déjà montés par les enfants dans les séances précédentes
- Vidéo projecteur (facultatif)

### Boîte à outils : ressources

- (11) Présentation Arduino p.28
- (12) Présentation PPT Arduino
- (21) Installation bibliothèque NewPing
- (22) Code Rosa

### Et si je dispose de plus de temps ?

- Laisser encore plus de temps aux enfants pour expérimenter avec ROSA.
- Effectuer des modifications dans le code principal (robot\_dclics). Exemple: faites avancer ROSA en permanence, reculer, faire des tours....

### Capacités de l'animateur

#### – Gestion du groupe

- Avant cette séance, l'animateur/trice doit bien préparer tout le matériel utilisé pendant la séance, le tester et expérimenter plusieurs fois pour qu'il/elle se sente à l'aise.
- Organiser des groupes pour que tous les enfants puissent faire l'expérience de la programmation et jouer avec le robot.



# SÉANCE 6 1/2

Durée – 1 heure

## PROJET COLLECTIF 1/3: RÉPARTITION DES GROUPES ET DÉMARRAGE DU PROJET.

L'animateur/trice accompagne les enfants dans la mise en place d'un projet de robotique.

### Objectifs pédagogiques

- Réutiliser les connaissances sur les différentes catégories des robots pour différents usages : sous-marins, volants, roulants, humanoïdes, utilitaires
- Créer un robot agréable à utiliser par la suite, car fonctionnel et solide
- Travailler en coopération pour que la conception du robot s'enrichisse des expériences des enfants (notion d'intelligence collective)
- Développer la créativité des enfants qui sera ensuite mise en pratique par les travaux manuels

### Connaissances requises par les enfants pour aborder la séance

- Connaissances manuelles de bases (ex. assembler des pièces, couper du carton, dessiner etc)
- Savoir écrire

### Éléments d'évaluation des acquis enfants

Les enfants sont capables de créer la structure de leur robot.

### Avant l'atelier

Les enfants doivent réfléchir auparavant à l'aspect de leur robot : humanoïde, boîte, dinosaure, autre

### Déroulement

#### 1 Répartition de groupes - 35 min

##### a. Explications - 5 min

L'animateur/trice explique la procédure pour la répartition des groupes. Ensuite il/elle fait un rappel sur les différents types de robot qui ont été discutés à la séance 1 et montre des images des robots faits avec un arduino pour leur donner des idées (cf.4).

##### b. Fiche individuelle - 15 min

Dans la discussion en début de séquence, on demande à chaque enfant d'inscrire le nom et le type de robot qu'il veut faire sur une feuille de papier (cf. 23). Cela doit être soit une forme déjà abordée, soit une forme qui n'a pas été vue en spécifiant ce qu'il devra faire.

*Proposition* : Les enfants veulent faire une forme inédite; il est possible de leur indiquer qu'ils peuvent se baser sur une structure existante (ex: boîte)

Nous affichons les groupes sur un tableau/mur et après discussion collective : les groupes seront composés de 5 enfants environ.

#### c. Projet collectif - 15 min

L'animateur/trice distribue à chaque groupe la feuille du projet collectif (cf. 24). Les enfants ont 15 min pour se mettre d'accord sur les modalités de leur projet et le remplir collectivement.

#### 2 Début des travaux - 25 min

##### a. Distribution du matériel - 5 min

L'animateur peut prévoir un espace où se trouvera les matériels (ex. table, boîte...).

Les enfants vont utiliser ou réutiliser des matériaux de récupération afin de limiter les déchets, l'impact sur l'environnement et favoriser l'imagination. C'est un bon moment pour sensibiliser les jeunes aux impacts environnementaux des démarches technologiques.

##### b. Dessiner la structure de base de ROSA - 20 min

L'animateur/trice peut distribuer le modèle boîte de ROSA (cf. 25,26) et laisser les enfants dessiner les leurs.

### Aménagements et matériel spécifique nécessaire

- Fiches sur les différents aspects et utilisation des robots
- Gabarit de la carte utilisée
- Crayons, stylo, colle, ciseaux, cartons, bois de récupération, règles
- 3 Modèles de bases de simples robots (x le nombre de robots)
- Les dessins de robot faits dans les séances précédentes
- Espace pour diffuser le matériel
- Espace ou dispositif pour ranger les robots en fin de séance

### Boîte à outils : ressources

- (4) Fiche: Les différents types de robots
- (23) Fiche individuelle
- (24) Projet robotique
- (25) Gabarit de la carte électronique utilisée
- (26) Modèles de boîte de robot

# SÉANCE 6 2/2

Durée – 1 heure

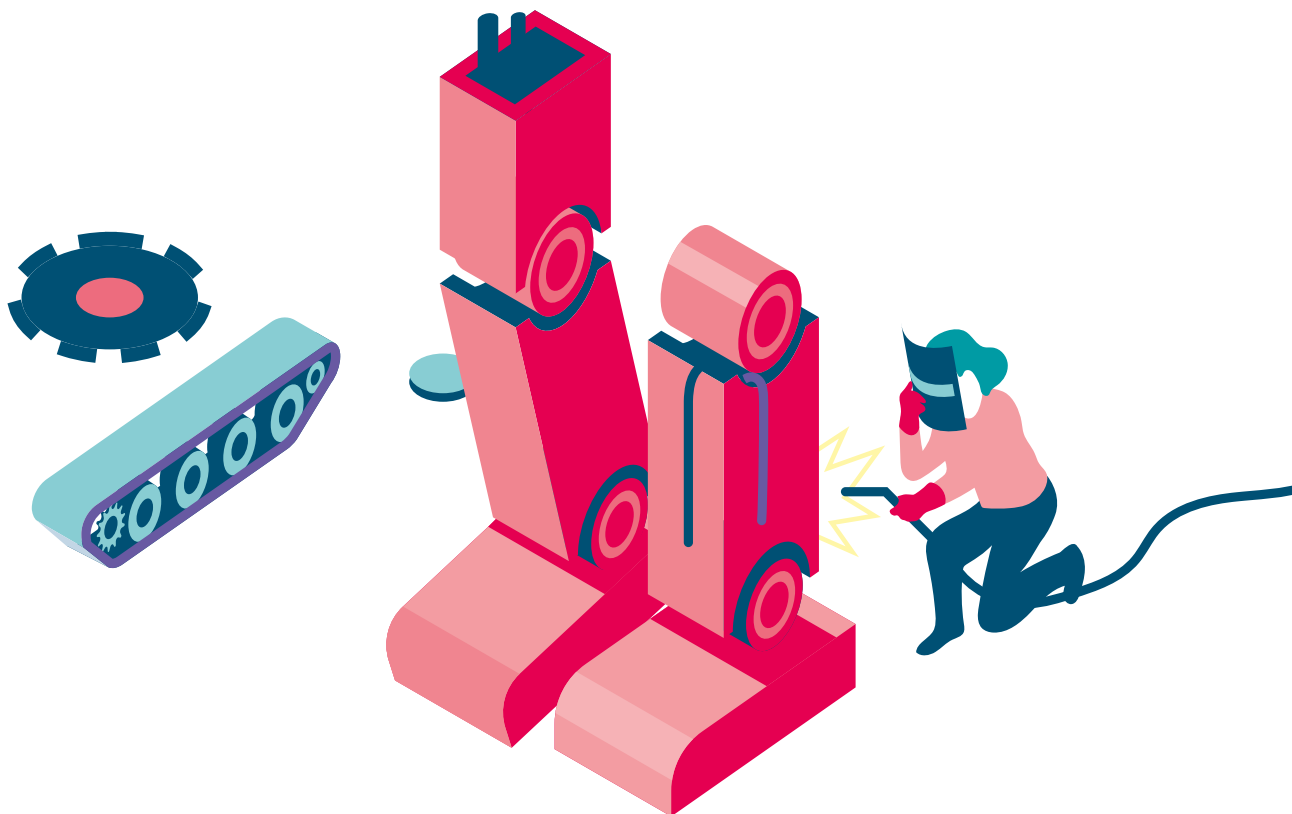
## Et si je dispose de plus de temps ?

- **Enrichir la phase 2b :** Après avoir effectué une première fois des groupes de 5 personnes, on leur donne la feuille du projet collectif (cf. 24), on leur donne du temps pour qu'ils commencent à la remplir et voir la faisabilité de leur projet. Ensuite, on rediscute le premier choix de groupes et donne la possibilité aux enfants de rechanger de groupes par rapport aux projets écrits. On finalise alors les groupes en leur donnant la feuille du projet collectif (cf. 24) et leur laissant du temps pour écrire le nouveau projet.
- Sur la deuxième partie de la création, il sera possible de créer un module à disposer sur la structure de base pour augmenter la complexité et la taille de ROSA

## Capacités de l'animateur

### – Gestion du groupe

- Organiser la prise de décision par les enfants dans les différents groupes et les orienter dans des formes de robot approchantes si les enfants veulent faire des formes inédites
- Organiser la gestion du temps qui risquent de manquer : il faudra se limiter à faire des structures simples qui seront réalisables dans le temps de la séance





# SÉANCE 7

Durée – 1 heure

## PROJET COLLECTIF 2/3: PERSONNALISATION DE ROSA.

L'animateur/trice accompagne les groupes à la construction de la structure de leurs robots.

### Objectifs pédagogiques

Personnaliser les structures de base créées précédemment afin de permettre une appropriation des robots.

- Repérer, identifier et différencier les différents aspects décoratifs des robots
- Apporter des modifications à un objet déjà construit
- Imaginer des formes pour décorer le robot
- Initier à la "mise en vie" de leur construction technique afin de la rendre plus appropriable par les enfants

### Connaissances requises par les enfants pour aborder la séance

- Avoir commencé les structures de bases de robots effectuées en séance 6
- Savoir effectuer des travaux manuels simples : couper du carton/papier avec des ciseaux, coller, encoller, colorier

### Éléments d'évaluation des acquis enfants

Les enfants ont créé leur robot dans lequel viendra se loger la carte électronique qui constituera la partie "commande"

### Déroulement

#### 1 Création des robots - 60 min

##### a. Temps d'échanges - 10 min

Temps d'échange entre les enfants d'un même groupe sur les différents éléments de personnalisation à construire afin de lier leur robot à leur imaginaire.

Ces différents éléments seront: la couleur, la/les formes, les accessoires, les textures utilisées, les dimensions souhaitées en lien avec la capacité technique du robot.

L'animateur/trice pose un cadre de séance en proposant aux enfants un nombre limité de matériaux, de couleur, et de technique à employer. Celui-ci pourra faire une démonstration de techniques (ex: la peinture) au préalable afin d'éviter aux enfants d'être mis en difficulté lors de la réalisation.

Ce temps permettra aux groupes de se fixer des objectifs réalisables en cours de séance de façon démocratique. Et également de travailler de manière coopérative en se distribuant les tâches à effectuer.

#### b. Création d'éléments structurels - 50 min

Création d'éléments structurels faisant appel à leur imaginaire comme des lumières, des bras ou toutes autres parties permettant d'identifier leur réalisation comme un robot selon leurs représentations. Ces éléments n'ont pas vocation à être fonctionnel, le but étant de construire un lien entre leurs propres représentations et leur création afin de faciliter l'appropriation de leur robot et donc des connaissances techniques transmises lors des ateliers précédents.

Si l'animateur/trice craint de manquer de temps, il pourra se reposer sur le modèle de base (cf. 26) ou ajouter une séance complémentaire pour laisser les enfants être créatifs au maximum. Durant la création des robots, l'animateur/trice s'assurera que la carte Arduino rentrera dans le robot final et qu'on pourra brancher le câble pour programmer ensuite le robot. On peut proposer aux enfants de manipuler le gabarit de la carte Arduino en carton (plutôt que la carte électronique elle-même qui est fragile : ce sera l'occasion de commencer à les sensibiliser à cet aspect).

### Aménagements et matériel spécifique nécessaire

- Feutres de couleur
- Papier, ciseaux, colle
- Papier décoratifs (papiers cadeaux)
- Formes rondes (pour créer des roues)
- optionnel: bouchons, objets ronds pouvant servir de roues

### Boîte à outils : ressources

(26) Modèles de boîte de robot

### Aller plus loin - autres ressources

Une autre proposition de forme de châssis, si vous vous disposez d'une imprimante 3D, est celle proposée par Quesnel Jonathan et son robot "ABC": [http://beta.wiki-fab.org/index.php/Construire\\_le\\_robot\\_%22ABC%22](http://beta.wiki-fab.org/index.php/Construire_le_robot_%22ABC%22)

### Et si je dispose de plus de temps ?

Création d'objets issus de leur imaginaire (des cubes à déplacer par exemple) pouvant interagir avec ROSA et ces différentes commandes.

### Capacités de l'animateur

#### – Gestion du groupe

S'assurer que tous les enfants participent au projet et que chacun trouve sa place

# SÉANCE 8

Durée – 1 heure

## PROJET COLLECTIF 3/3: PROGRAMMATION DE ROSA.

L'animateur/trice accompagne les groupes à la programmation de leur robot par rapport aux objectifs posés pendant la phase 1 du projet (cf. séance 6).

### Objectifs pédagogiques

- Faciliter l'appropriation de savoirs et savoirs-faire techniques de manière ludique
- Savoir programmer le robot en utilisant le langage Arduino
- Scénarisation en groupe

### Connaissances requises par les enfants pour aborder la séance

Avoir compris les bases de la programmation Arduino.  
Avoir complété la séance 6.

### Éléments d'évaluation des acquis enfants

Les enfants ont programmé ROSA en utilisant des commandes vues à la séance 5

### Déroulement

#### 1 Programmation de ROSA - 30 min

Les enfants en groupe programment ROSA par rapport aux objectifs de leur projet en utilisant les commandes apprises pendant la séance 5.

L'animateur/trice les accompagne aux modifications du code (cf.22) . Il est important d'expliquer aux enfants qu'il faut utiliser ce qu'ils savent déjà faire. Cela évitera de fixer des objectifs difficiles à atteindre.

#### 2 Scénarisation - 30 min

Les enfants dessinent et construisent un environnement avec lequel ROSA interagira. L'objectif est de mettre en avant l'imagination des enfants et leur permettre de scénariser leur projet. L'animateur/trice peut accompagner les enfants pour que l'environnement construit soit en lien avec la thématique de leurs projets.  
Exemple: Si le robot fait par une équipe ressemble à une araignée, l'environnement peut être la nature et les obstacles des arbres ou autres animaux.

### Aménagements et matériel spécifique nécessaire

- Feutres de couleur
- Papier, ciseaux, colle
- Papier décoratifs (papiers cadeaux)

### Aller plus loin - autres ressources

L'animateur/trice qui souhaite aller plus loin au niveau de l'Arduino peut suivre des cours en ligne sur la plateforme FUN : **MOOC Arduino** pour proposer aux enfants plus de fonctions au niveau de la programmation de ROSA.

### Boîte à outils : ressources

(22) Code ROSA p.28

### Et si je dispose de plus de temps ?

- Faire une scénarisation commune des différents projets en créant une histoire et en environnement adéquat.

### Capacités de l'animateur

#### - Gestion du groupe

- Bonne compréhension de la programmation sous Arduino et des modifications à faire dans le programme principal (ligue\_dclics)



# SÉANCE 9-10

Durée - 2h (dont 20 min de présentation pour l'ensemble des groupes)

## PRÉPARATION DE LA PRÉSENTATION ET PRÉSENTATION DES PROJETS ROBOTIQUES :

L'animateur/trice accompagne les enfants à la mise en place et organisation de la présentation de leurs projets.

### Objectifs pédagogiques

- s'exprimer en utilisant la langue française à l'oral et à l'écrit
- mobiliser les outils numériques pour échanger et communiquer
- valoriser le travail des enfants
- promouvoir le travail collaboratif

### Pré-requis par les enfants pour aborder la séance

En début de projet : prévoir un "reporter" dans le groupe pour mettre en oeuvre les pré-requis suivants :

- prise de photo tout au long du projet, à chaque étape
- copies d'écran lors de la programmation
- éventuellement captation vidéo tout au long du projet (essais durant le projet, interviews, etc.)
- conserver l'ensemble des productions écrites réalisées lors du projet : croquis, dessins, schémas.

**Concernant les productions papier, il est souhaitable que l'animateur/trice les numérise en les scannant.**

### Déroulement

Si l'animateur/trice dispose de plus de temps, il/elle peut consacrer une ou deux séances sur la présentation des projets de différents groupes et les échanges entre les enfants sur leurs projets, les difficultés rencontrées, les points forts et faibles de leurs projets etc.

**1 Définition des parties de la présentation - 20 min**  
**a. phase de recherche (individuelle) - 5 min**  
 chaque enfant s'isole 5 min pour se rappeler les différentes étapes du projet du groupe. Chacun note ses idées sur papier. Les parties retrouvées devraient être : construction de ROSA ; personnalisation ; programmation (en précisant le thème ou le scénario).

**b. phase de mise en commun - 15 min**  
 Mise en commun rapide sous la forme de discussion. Le groupe doit délivrer, au bout de 15 min, le plan de sa présentation (cf. 27)

Ensuite, les enfants se répartissent les parties de la présentation. Idéalement, un groupe de 3 enfants -> 1 enfant pour traiter chaque partie.

### 2 Réalisation des parties de la présentation - 1h15

**a. 15 min - Identifier les médias à intégrer** (photos, vidéo, dessins, etc.) pour les différentes étapes de la partie. Les enfants visionnent les médias réalisés pendant le projet. Si nécessaire, ils font le choix de ce qui leur semble être les bons éléments d'illustration. Pour chaque média sélectionné, ils indiquent l'étape dans laquelle elle va (si besoin).

**b. 15 min - Écriture d'une courte phrase** pour chaque étape, incluant, le cas échéant, le vocabulaire approprié. Description de chaque étape par une ou plusieurs phrases courtes. L'enfant devra utiliser le vocabulaire adapté.

**c. 45 min - Réalisation de la présentation de parties.**

### Aménagements et matériel spécifique nécessaire

- 1. papier crayon
- 2.a. accès aux médias produits pendant le projet, sur support USB ou sur lecteur distant.
- 2.b. écriture au choix, sur papier, bloc note de l'ordinateur ou directement dans l'application de présentation
- 2.c. utilisation d'une application de présentation type Prezi ou Sozi
- Démonstration préalable nécessaire.

### Boîte à outils : ressources

- (27) Présentation du projet
- (28) Prezi
- (29) Didacticiel Sozi

### Capacités de l'animateur

#### - Gestion du groupe

- Maîtrise du temps
- Avoir déjà testé les outils comme Prezi et Sozi pour pouvoir accompagner les enfants

### Et si je dispose de plus de temps ?

- Mise en page de la présentation avec un outil de type Prezi (cf. 28) ou Sozi (cf. 29)
- Création d'un glossaire
- Création d'un blog de présentation
- Présenter le projet auprès des parents, des enfants qui n'ont pas participé à ces ateliers, d'un public plus large.

A N

N E

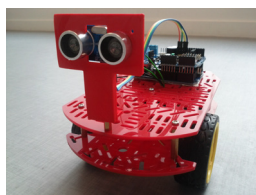


**Retrouvez, ci-après, les fiches annexes qui vous permettront de compléter les séances conçues dans le cadre des parcours “Robotique”.**

## **SOMMAIRE**

### **ANNEXES**

<b>24</b>	<b>ANNEXE 1</b>	SÉANCE 1
<b>33</b>	<b>ANNEXE 2</b>	SÉANCE 1
<b>34</b>	<b>ANNEXE 3</b>	SÉANCE 1
<b>45</b>	<b>ANNEXE 4</b>	SÉANCES 1, 6
<b>47</b>	<b>ANNEXE 5</b>	SÉANCE 1
<b>48</b>	<b>ANNEXE 6</b>	SÉANCE 1
<b>54</b>	<b>ANNEXE 7</b>	SÉANCE 2
<b>55</b>	<b>ANNEXE 8</b>	SÉANCE 2
<b>56</b>	<b>ANNEXE 9</b>	SÉANCE 2
<b>57</b>	<b>ANNEXE 10</b>	SÉANCE 2
<b>58</b>	<b>ANNEXE 11</b>	SÉANCES 2, 5
<b>59</b>	<b>ANNEXE 12</b>	SÉANCES 3, 5
<b>60</b>	<b>ANNEXE 13</b>	SÉANCE 3
<b>68</b>	<b>ANNEXE 14</b>	SÉANCE 3
<b>82</b>	<b>ANNEXE 15</b>	SÉANCE 3
<b>85</b>	<b>ANNEXE 16</b>	SÉANCE 3
<b>86</b>	<b>ANNEXE 17</b>	SÉANCE 4
<b>91</b>	<b>ANNEXE 18</b>	SÉANCE 4
<b>92</b>	<b>ANNEXE 19</b>	SÉANCE 4
<b>93</b>	<b>ANNEXE 20</b>	SÉANCE 4
<b>94</b>	<b>ANNEXE 21</b>	SÉANCE 5
<b>95</b>	<b>ANNEXE 22</b>	SÉANCES 5, 8
<b>101</b>	<b>ANNEXE 23</b>	SÉANCE 6
<b>102</b>	<b>ANNEXE 24</b>	SÉANCE 6
<b>103</b>	<b>ANNEXE 25</b>	SÉANCE 6
<b>104</b>	<b>ANNEXE 26</b>	SÉANCES 6, 7
<b>105</b>	<b>ANNEXE 27</b>	SÉANCES 9, 10
<b>106</b>	<b>ANNEXE 28</b>	SÉANCES 9, 10
<b>107</b>	<b>ANNEXE 29</b>	SÉANCES 9, 10



## INTRODUCTION ET DEFINITION DE LA ROBOTIQUE

### Sommaire

Qu'est ce qu'un robot .....	2
Composition d'un robot .....	3
<b>Les capteurs</b> .....	3
<b>Les circuits électroniques</b> .....	3
<b>Les actionneurs</b> .....	3
<b>Autonomie</b> .....	3
Origine de la robotique .....	4
<b>Les premiers robots</b> .....	4
<b>Usages</b> .....	5
Différence entre automate et robot .....	5
Les lois de la robotique .....	5
La conception d'un robot .....	6
Usages des Robots .....	6
Métiers de la Robotique .....	7
Les activités autour de la robotique en milieu scolaire et périscolaire .....	7
<b>Préambule</b> .....	7
<b>Démarche pédagogique " Apprendre en faisant "</b> .....	7
<b>Démarche expérimentale et apprentissages</b> .....	7
<b>Des ateliers contribuant à la validation de compétences transversales</b> .....	7
<b>Un usage naturel des outils numériques</b> .....	8
<b>La programmation et la robotique</b> .....	8
Conclusion : la robotique pédagogique .....	9
<b>Robotique Pédagogique et pensée critique</b> .....	9
<b>Robotique pédagogique et créativité</b> .....	9
<b>Robotique pédagogique et pensée informatique</b> .....	9
<b>Robotique pédagogique et collaboration</b> .....	9
<b>Robotique pédagogique et résolution de problèmes</b> .....	9

Adrien Payet  
 Directeur du CRREP

Centre de Ressources en Robotique Educative et Professionnelle

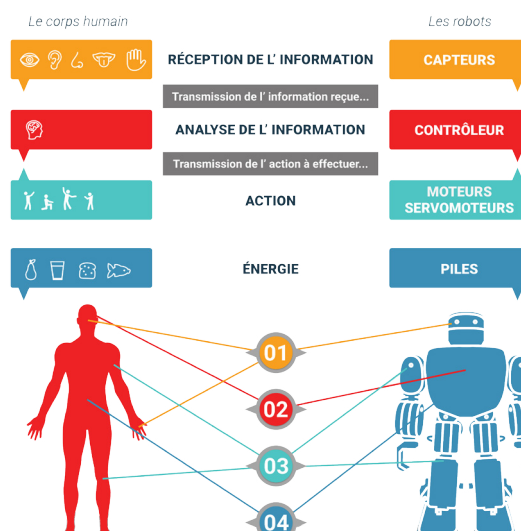
## Qu'est ce qu'un robot

Le terme robot fait partie de notre quotidien depuis les années 1960 où il apparaît dans l'électroménager. Pourtant, aujourd'hui encore définir un robot est moins facile qu'il n'y paraît, la question étant de déterminer les caractéristiques qui font basculer une machine dans le monde de la robotique. Il n'existe pas de définition unique néanmoins il est d'usage de dire qu'un robot est l'association indissociable de trois éléments :

- **Les capteurs** qui servent à appréhender l'environnement, éléments qui s'apparentent aux sens (la vue, le toucher,...)
- **L'unité de programmation** pour "raisonner", élément électronique qui fait office de cerveau.
- **Les actionneurs** pour interagir avec le monde réel, les "membres".

Ces trois éléments donnent la capacité aux robots d'analyser leur environnement, de décider et d'agir sur le monde réel.

- **L'énergie** permet l'autonomie du robot



*Si l'on reprend les différentes définitions de la robotique (le petit Robert, le Larousse, Wikipédia,...) on s'aperçoit qu'elles varient énormément !*

### Le petit Robert :

Mécanisme automatique à commande électronique pouvant se substituer à l'homme pour effectuer certaines opérations et capable d'en modifier de lui-même le cycle en appréhendant son environnement.

### Le Larousse :

Appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe, modifiable ou adaptable.

### Wikipédia :

Un robot est un dispositif mécatronique accomplissant automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains, soit des tâches plus simples mais en les réalisant mieux que ce que ferait un être humain.

*Entre l'usage de l'utilisation du mot, "robot ménager" par exemple, et l'évolution de la technologie, il s'avère compliqué de définir de façon exacte et précise ce qu'est la robotique.*

*C'est pourquoi nous vous proposons la définition qui nous paraît la plus appropriée pour définir un robot de façon simple et claire.*

**Un Robot est un dispositif autonome qui dispose d'au moins un capteur et/ou une télécommande, un contrôleur et un actionneur**

Adrien Payet  
Directeur du CRREP

Centre de Ressources en Robotique Educative et Professionnelle



## Composition d'un robot

Un robot est l'assemblage complexe de pièces mécaniques et de pièces électroniques pouvant être pilotées par une intelligence artificielle afin d'exécuter des actions autonomes.

*Les robots autonomes possèdent une source d'énergie embarquée (batterie, piles,...).*

### Les capteurs

Il en existe une grande variété. Par exemple :

- **Capteurs à Infrarouge**
- **Capteurs tactiles**
- **Les sondeurs** (ou télémètres) à Ultra-son ou LASER. Ces derniers sont à la base des scanners laser permettant à l'unité centrale du robot de prendre "conscience" de son environnement en 3D.
- Les **caméras** sont les yeux des robots. Il en faut au moins deux pour permettre la vision en trois dimensions. Le traitement automatique des images pour y détecter les formes, les objets, voire les visages, demande en général un traitement matériel car les microprocesseurs embarqués ne sont pas assez puissants pour le réaliser.

Dans le cadre d'un robot roulant sur des roues, les **roues codeuses** permettent un déplacement précis en mesurant les angles de rotation. (Information proprioceptive).

### Les circuits électroniques

Les microprocesseurs ou les microcontrôleurs sont des éléments primordiaux d'un robot, car ils permettent l'exécution de logiciels informatiques donnant son autonomie au robot. On trouve souvent dans un robot des modèles à très faible consommation, notamment pour des robots de petite taille, qui ne peuvent pas emporter avec eux une source d'énergie importante.

### Les actionneurs

Les actionneurs les plus usuels sont :

- des moteurs électriques rotatifs, qui sont fréquemment associés à des réducteurs mécaniques à engrenages et les servomoteurs.
- des vérins hydrauliques, reliés par une tuyauterie à des pompes fournissant des pressions élevées.

Généralement, un actionneur peut être considéré comme un constituant d'un système mécanique (exemple : bras, patte, roue motrice...) et correspond à un degré de liberté.

Les interfaces haptiques permettent au robot de saisir des objets. Les moteurs permettent à des éléments mobiles de bouger suivant un ou plusieurs degrés de liberté; elles sont plutôt des constituants appartenant au domaine de la télémanipulation.

### Autonomie

On cherche à réaliser des systèmes capables de réagir seuls à l'environnement, c'est-à-dire à un certain imprévu. C'est ce plus ou moins grand degré d'autonomie (permis par une intelligence artificielle) qui rapproche les robots des systèmes complètement autonomes envisagés par la science-fiction et la recherche de pointe.

Une certaine capacité d'adaptation à un environnement inconnu peut, dans les systèmes semi-autonomes actuels, être assurée pourvu que l'inconnu reste relativement prévisible : l'exemple déjà opérationnel du robot aspirateur en est une bonne illustration : le logiciel qui pilote cet appareil est en mesure de réagir aux obstacles qui peuvent se rencontrer dans une habitation, de les contourner, de les mémoriser. Il sauvegarde le plan de l'appartement et peut le modifier en cas de besoin. Il retourne en fin de programme se connecter à son chargeur. Il doit donc fournir une réponse correcte au plus grand nombre possible de stimulations, qui sont autant de données entrées, non par un opérateur, mais par l'environnement.

L'autonomie suppose que le programme d'instructions prévoit la survenue de certains événements, puis la ou les réactions appropriées à ceux-ci. Lorsque l'aspirateur évite un buffet parce qu'il *sait* que le buffet est là, il exécute un programme intégrant ce buffet, par exemple les coordonnées X-Y de son emplacement. Si

Adrien Payet  
Directeur du CRREP

Centre de Ressources en Robotique Educative et Professionnelle

ce buffet est déplacé ou supprimé, le robot est capable de modifier son plan en conséquence et de traiter une zone du sol qu'il ne prenait pas en compte jusqu'alors.

## Origine de la robotique

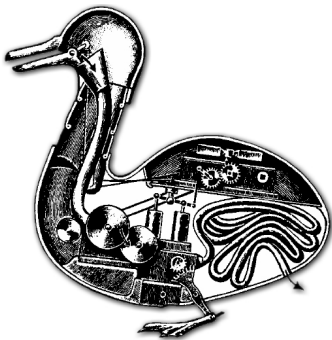
Le terme Robot a été créé en 1920 par Josef Capek, c'est la déclinaison de robota, mot tchèque signifiant " corvée ". Sa première utilisation en a été faite par son frère Karel Capek dans la pièce de science fiction " Rossum's Universal Robots " jouée en 1922 à New York. Les robots de la pièce sont proches de ce qu'on appelle aujourd'hui des androïdes ou des clones, machines biologiques à l'apparence humaine.

Ce mot est vite adopté par le corps scientifique pour nommer un dispositif mécatronique alliant mécanique, électronique et informatique. Les robots ont pour fonction d'accomplir automatiquement des tâches dangereuses, pénibles, répétitives ou impossibles pour les humains.

La vulgarisation de ce terme entraîne aujourd'hui son utilisation pour évoquer la haute technicité d'un dispositif.

### Les premiers robots

Les ancêtres des robots sont les automates. Un automate très évolué fut présenté par Jacques de Vaucanson en 1738 : il représentait un homme jouant d'un instrument de musique à vent. Jacques de Vaucanson créa également un automate représentant un canard mangeant et refoulant sa nourriture après ingestion de cette dernière. Néanmoins les automates sont bien plus anciens et l'on peut citer le lion automate conçu par Léonard de Vinci et présenté en 1515 à François 1<sup>er</sup>.



*Canard de Vaucanson*



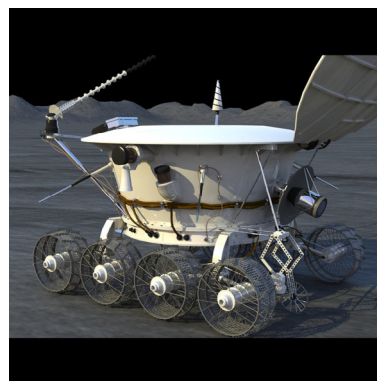
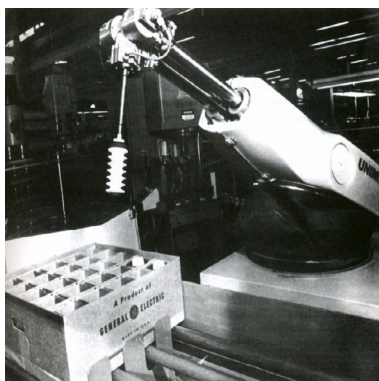
*Joueur de flûte Vaucanson*



*Le lion de Léonard de Vinci  
(reproduction)*

On estime que Unimate est le premier robot abouti, robot industriel intégré aux lignes d'assemblage de Général Motors en 1961.

Le robot lunaire Lunokhod 1 envoyé sur la lune par l'Union Soviétique en 1970, a voyagé sur une distance de 10 km pour récolter et transmettre plus de 20 000 images.



*Adrien Payet  
Directeur du CRREP*

Centre de Ressources en Robotique Educative et Professionnelle

**Usages**

La robotique possède de nombreux domaines d'application. Les robots ont été installés dans les industries, ce qui permet de faire des tâches répétitives avec une précision constante. À la suite de l'évolution des techniques on retrouve des robots dans des secteurs de pointe tels que le spatial, la médecine, chez les militaires. Depuis quelques années la robotique est devenue grand public et on retrouve désormais de nombreux robots domestiques et ludiques. (Cf. *Usage des robots*)

**Différence entre automate et robot**

Un automate du grec *automatos* [qui se meut de soi-même] est une machine qui, par le moyen de dispositifs mécaniques, pneumatiques, hydrauliques, électriques ou électroniques, est capable d'actes imitant ceux des corps animés.

<http://www.larousse.fr/dictionnaires/francais/automate/6746>

Dans la robotique, il n'est pas rare de confondre Automate et Robot il est donc important de savoir faire la différence entre les deux.

Un automate est un dispositif se comportant de manière automatique, sans intervention extérieure. Il renferme divers dispositifs mécaniques et/ou électriques, qui lui permettent d'exécuter une séquence déterminée d'opérations de manière synchronisée. Il est programmé et possède donc une mémoire. Ce comportement est figé, le système fera toujours la même succession d'action.

**Les lois de la robotique**

Les trois lois de la robotique furent le fruit de discussions entre l'écrivain de science-fiction Isaac Asimov et son éditeur John W. Campbell autour du thème des robots. Elles furent formulées explicitement pour la première fois en 1942 dans la nouvelle *Cycle fermé (Runaround)*. Elles permettent d'introduire le "danger" de la robotique et l'intelligence artificielle.

- [1] Un robot ne peut porter atteinte à un être humain, ni, en restant passif, permettre qu'un être humain soit exposé au danger.
- [2] Un robot doit obéir aux ordres qui lui sont donnés par un être humain, sauf si de tels ordres entrent en conflit avec la première loi.
- [3] Un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la première ou la deuxième loi.

**Une nouvelle loi a été ajoutée :**

Un robot ne peut porter atteinte à l'humanité ni, restant passif, laisser l'humanité exposée au danger. Avec la modification consécutive de la première loi :

« Un robot ne peut porter atteinte à un être humain ni, restant passif, laisser cet être humain exposé au danger, sauf quand cela s'oppose à la précédente loi. »

**Désormais une cinquième loi est apparue :**

Un robot ne pourra pas se réparer lui-même.

La robotique est l'enjeu de demain et une réflexion sur les robots "qui remplaceront" l'Homme est un sujet inhérent à la robotique. Les lois de la robotique sont un support intéressant pour mener cette réflexion. Ce sujet a été abordé dans de nombreux films de science fiction, qui peut par des visions optimiste ou pessimiste être un complément à un débat "enjeux de la robotique".

Adrien Payet  
Directeur du CRREP

Centre de Ressources en Robotique Educative et Professionnelle

## La conception d'un robot

Les concepteurs de robots, quelqu'en soient les formes et les fonctions, sont confrontés à de nombreuses problématiques, la robotique est un concentré de technologies complexes et variées faisant appel à différents domaines tel que la programmation, la mécanique, la mécatronique, la cognitive, le design...

Les problématiques techniques rencontrées sont :

- **L'intelligence** du robot qui renvoie au domaine de la programmation et des logiciels.
- **L'architecture** qui peut se définir comme l'art de choisir les meilleures façons de structurer et combiner les composants du robot tel que les robots industriels de soudage qui doivent assembler des tôles très complexes et auront des bras avec 4 à 5 articulations.
- **Les actionneurs et préhenseurs** tels que les moteurs, les vérins, les pinces,... et tous les composants nécessaires à leur commande et au contrôle de leur position.
- **Les capteurs**, organes par lesquels le robot perçoit son environnement, qui doivent à la fois donner aux robots des informations de base sur son environnement (température, proximité d'un obstacle...) et lui permettre d'acquérir des informations plus fines (micro pour prendre en compte des instructions orales, ou caméra pour identifier un objet ou un obstacle).
- **La communication** du robot qui n'est pas limitée à l'échange d'informations entre le robot et l'utilisateur, il est aussi en lien avec d'autres appareils (ordinateur, équipements domotiques...) situés dans son environnement de travail entraînant des questions de protocole de transmission et de reconnaissance mutuelle.
- **L'énergie**, le plus souvent l'électricité par alimentation directe, qui reste également un enjeu important pour optimiser la consommation et augmenter l'autonomie des robots.

## Usages des Robots

- **Robots domestiques** : tâches ménagères
- **Robots mobiles** : capables de se déplacer dans leur environnement, avec manipulateurs ou non.
- **Humanoïdes** : apparence d'un bipède
- **Robot industriel**
- **Robotique de service**
- **Robotique d'assistance à la personne**
- **Robotique liée à la santé**
- ...
- **robot collaboratif** : hommes et robots travaillent ensemble pour faciliter le travail de l'opérateur.  
Précision des mouvements, diminution de la pénibilité

### La Cobotique :

Alors que jusqu'à présent les robots travaillaient de manière autonome, le besoin de réaliser des tâches variées et changeantes implique un travail coopératif et coordonné entre les robots et les humains. Aujourd'hui, les roboticiens se concentrent donc sur la notion de **sécurité** en développant des robots capables de travailler côte à côte avec les humains.

*Selon une étude de l'IFR, 2142 millions de robots ont été fabriqués entre les années 60 et la fin 2010, les analystes estiment qu'aujourd'hui, de 1 à 1,3 million de robots travaillent pour nous dans les usines dans le monde. <http://www.larousse.fr/encyclopedie/divers/robot/88768>*

Adrien Payet  
Directeur du CRREP

Centre de Ressources en Robotique Educative et Professionnelle

## Métiers de la Robotique

### Préambule

Si dès à présent nous pouvons lister les métiers autour de la robotique, ce domaine n'en étant qu'à ses balbutiements les besoins et les métiers vont énormément évoluer dans les quelques années à venir.

*Le nombre de robots est exponentiel et il est encore difficile aujourd'hui d'imaginer l'explosion de la robotique dans les décennies à venir. Ce qui est sûr c'est que la « Robolution » est en marche et qu'il est désormais indispensable de s'y intéresser plutôt que d'en avoir peur.*

## Les activités autour de la robotique en milieu scolaire et périscolaire

### Préambule

Il est fortement conseillé d'accompagner les ateliers robotiques par la transmission des connaissances sur la robotique.

Connaissances autour des notions de base de la robotique : Définition de la robotique, composition d'un robot,... ainsi que les connaissances générales sur l'histoire de la robotique, notre environnement et la robotique, la mécatronique, les métiers de la robotique...

Ces notions acquises complèteront les ateliers et permettront aux enfants d'une part de comprendre le fonctionnement des robots qu'ils construiront et d'autre part de faire un rapprochement entre ce qu'ils réaliseront et ce qu'ils peuvent observer autour d'eux. (*Pourquoi et comment les portes d'un ascenseur ne se ferment pas quand on monte dedans, comment une machine distributrice déclenche l'obtention d'une boisson et rend la monnaie, fonctionnement du corps humain,...*)

### Démarche pédagogique " Apprendre en faisant "

Les activités robotiques font appel à **une pratique coopérative**, les enfants apprennent avec l'animateur mais aussi par et avec les autres. Le travail et la réflexion en groupe permettent aux enfants de confronter leurs idées et de prendre des décisions communes, parfois après des expérimentations. **La pluridisciplinarité** de la robotique facilite également l'implication et l'expression de chacun et sollicite chez les enfants une ouverture vers plusieurs savoirs ce qui développe ainsi leurs curiosités et leurs envies d'apprendre. **L'aspect ludique** de la robotique renforce l'intérêt des enfants et augmente ainsi leurs attentions.

L'ensemble de ces éléments renforce la solidarité entre les enfants et facilite les apprentissages.

### Démarche expérimentale et apprentissages

Le moteur principal est l'expérimentation scientifique avec pour thème la " Conception de systèmes robotisés ". Les situations de démarches expérimentales sont nombreuses tant dans la découverte du fonctionnement des articulations que dans la conception de systèmes robotisés reproduisant les mouvements. Concernant la composante " robotique " en particulier, celle-ci a pour but de faire comprendre à des enfants les enjeux mathématiques (angles, leviers, échelles, proportionnalité, calcul de périmètres de cercles et polygones, parallélogrammes, raisonnement, etc... *en fonction du niveau de l'enfant*) qu'ils abordent en robotique.

### Des ateliers contribuant à la validation de compétences transversales

Les enfants ont le rôle principal tandis que les animateurs sont dans une posture d'accompagnateurs. Les enfants doivent imaginer et réaliser le robot par eux-mêmes. Les animateurs sont là pour accompagner et non pour apporter des réponses " expertes ". Les enfants valident leurs réponses par leurs démarches expérimentales réalisées en groupe dans les ateliers. Tout le travail réalisé peut-être réinvesti ensuite dans

Adrien Payet  
Directeur du CRREP

Centre de Ressources en Robotique Educative et Professionnelle

les enseignements en classe de SVT, de physique, de mathématiques... L'activité robotique est facilitatrice pour l'acquisition des compétences du Socle relatives à " L'autonomie et l'initiative " ou bien touchant aux " principaux éléments de mathématiques " et " de culture scientifique et technologique ". Les aptitudes comme " Savoir observer, questionner, formuler une hypothèse et la valider, argumenter, modéliser de façon élémentaire " ou " Comprendre le lien entre les phénomènes de la nature et le langage mathématique qui s'y applique et aide à les décrire " sont particulièrement sollicitées.

#### **Un usage naturel des outils numériques...**

Cette pratique est également riche d'usages numériques par la recherche d'informations et d'investigation, notamment sur Internet, et par la programmation des robots.

Les compétences telles que "S'approprier un environnement informatique de travail ", " Créer, produire, traiter, exploiter des données ", " S'informer, se documenter " ou encore " Communiquer, échanger " sont mises en œuvre.

#### **Il faut distinguer trois applications pédagogiques distinctes de la robotique**

- l'apprentissage **de** la robotique  
*implique l'utilisation du robot en tant que support pour apprendre la mécanique, l'électronique, l'informatique à travers des activités pratiques collaboratives.*  
La finalité est l'acquisition de connaissances et de compétences inhérentes à la construction et à la programmation de robots.
- l'apprentissage **avec** la robotique  
*repose sur l'interaction entre les enfants et un robot humanoïde ou animoïde qui recouvre le rôle de compagnon pour les enfants ou d'assistant pour l'enseignant.*  
La finalité vise à provoquer des réactions empathiques et à créer des interactions cognitives et sociales.
- l'apprentissage **par** la robotique  
*implique l'usage de kits robotiques de construction et de programmation.*  
La finalité est l'acquisition de connaissances et compétences liées à une matière scolaire précise – mathématiques, sciences, technologie. Mais sa finalité éducative réside aussi dans l'acquisition de compétences transversales (résoudre des problèmes, communiquer, prendre des initiatives, ...) et dans le développement des facultés cognitives, métacognitives et sociales des enfants à travers l'autocorrection, la planification, l'esprit critique, le travail collaboratif, la confiance en soi...

#### **Le potentiel de la robotique pour les apprentissages est lié à la mise en œuvre d'une approche et d'une scénarisation pédagogiques adaptées à l'âge des enfants et aux contextes des ateliers.**

##### **La programmation et la robotique**

Il est important de différencier deux types de programmation :

- La programmation d'« Actions », actions sous conditions.  
*(ex. si le robot rencontre un obstacle, alors on lui demande de tourner) « SI - ALORS - SINON »*
- La programmation de « Motions », mouvements que peut effectuer un robot  
*(ex. avancer, reculer, tourner,... pour un robot humanoïde)*

Le principe de la programmation d'actions est de tester les capteurs présents sur le robot afin de déclencher une action. Si cette action est un mouvement il sera nécessaire de programmer ce mouvement (moteurs, servomoteurs, vérins,...)

La plupart des outils robotiques à visée pédagogique ne nécessitent pas la programmation de mouvements. Les interfaces proposent le plus souvent des programmations sous conditions qui font appels à des fonctions qui réalisent les mouvements désirés.

*Adrien Payet  
Directeur du CRREP*

Centre de Ressources en Robotique Educative et Professionnelle

## Conclusion : la robotique pédagogique Pour le développement des compétences du 21ème siècle

### **Robotique Pédagogique et pensée critique**

Comprendre et être critique face aux technologies existantes. Développer une réflexion sur les défis éthiques des relations personne-robot.

### **Robotique pédagogique et créativité**

Développer la créativité au niveau de la conception de la construction ou de la programmation. Trouver des solutions nouvelles innovantes et pertinentes pour répondre à un défi robotique. Aller au-delà de la consommation passive ou interactive des technologies et développer une approche créative aux technologies.

### **Robotique pédagogique et pensée informatique**

Apprendre à programmer par le biais d'interfaces de programmation visuelles qui facilitent la compréhension des processus et des méthodes informatiques. Développer la capacité d'abstraction, de décomposition et de structuration des données et des processus nécessaires à l'élaboration de la programmation du robot.

### **Robotique pédagogique et collaboration**

Développer la collaboration face à des défis robotiques en équipe qui nécessitent une coordination des différents membres. Mettre en valeur la diversité de compétences et de talents des membres de l'équipe. Développer l'engagement des apprenants par des mécaniques de coopération et de compétition.

### **Robotique pédagogique et résolution de problèmes**

Développer une attitude positive face aux problèmes comme source d'apprentissage et de résilience. Développer une approche interactive et de prototype (design thinking) pour résoudre une situation problème complexe. Développer une analyse afin de décomposer les besoins pour un défi robotique. Développer une capacité à déterminer une solution à construire et à mettre en œuvre.

*Adrien Payet*  
*Directeur du CRREP*

Centre de Ressources en Robotique Educative et Professionnelle



## Installation de la carte Arduino Uno

La carte Arduino est une carte électronique open source et simple d'utilisation. Elle est développée par un italien nommé Massimo Banzi. Cette carte permet de rendre l'électronique accessible au plus grand nombre. Le modèle de base permet de contrôler 14 entrées/sorties digitales dont 6 sorties PWM (Modulation de Largeur d'Impulsion) pour faire varier la tension de sortie, et 6 entrées analogiques grâce à un microcontrôleur ATMEGA-328. Ressources en ligne.

Ressources	PDF	Notes
<b>Présentation de la carte Arduino</b>	<a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:presentation_arduino.pdf">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:presentation_arduino.pdf</a>	
<b>Installation d'Arduino sous Windows</b>	<a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installation_arduino_ardublock_windows.pdf">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installation_arduino_ardublock_windows.pdf</a>	Ardublock facultatif pour le moment
<b>Installation d'Arduino sous Mac</b>	<a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installer_arduino_sur_un_mac.pdf">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installer_arduino_sur_un_mac.pdf</a>	Ardublock facultatif pour le moment
<b>Installation d'Arduino sous Gnu/Linux</b>	<a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installation_arduino_linux.pdf">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installation_arduino_linux.pdf</a>	Installation orientée Raspberry



Retrouvez cette annexe sur :  
<https://frama.link/installationArduino>

## Magician Chassis

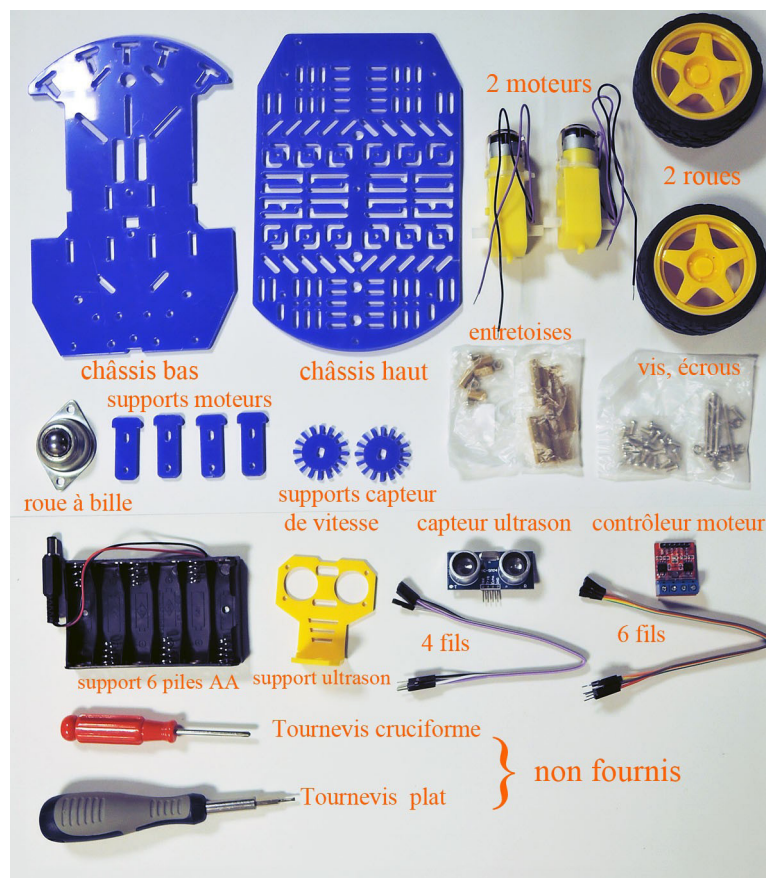
Le fichier en ligne : [https://wiki.mdl29.net/doku.php?id=projets:robotarduino#montage\\_du\\_chassis](https://wiki.mdl29.net/doku.php?id=projets:robotarduino#montage_du_chassis)

### Introduction




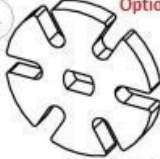
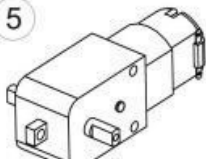
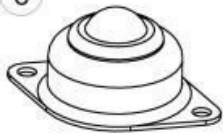

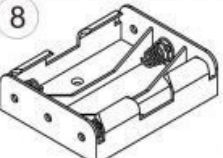
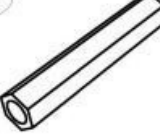
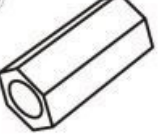

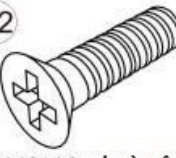

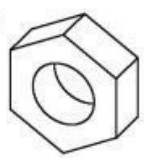
Le Magician chassis est une plateforme robotique facile à assembler soi-même. Il propose un support 6 piles AA pour l'alimentation. Une fois le châssis monté, on peut l'équiper de capteurs ou d'actionneurs pour en faire un véritable robot intelligent et autonome.

### Contenu

- 2 plaques en acrylique de 3mm découpées au laser
- 2 moteurs DC 6V double axe
- 2 roues complètes
- 1 roue à bille
- 1 support de 6 piles AA
- 1 support ultrason
- 1 ensemble de quincaillerie, vis et écrous
- 1 ensemble de fil de câblage



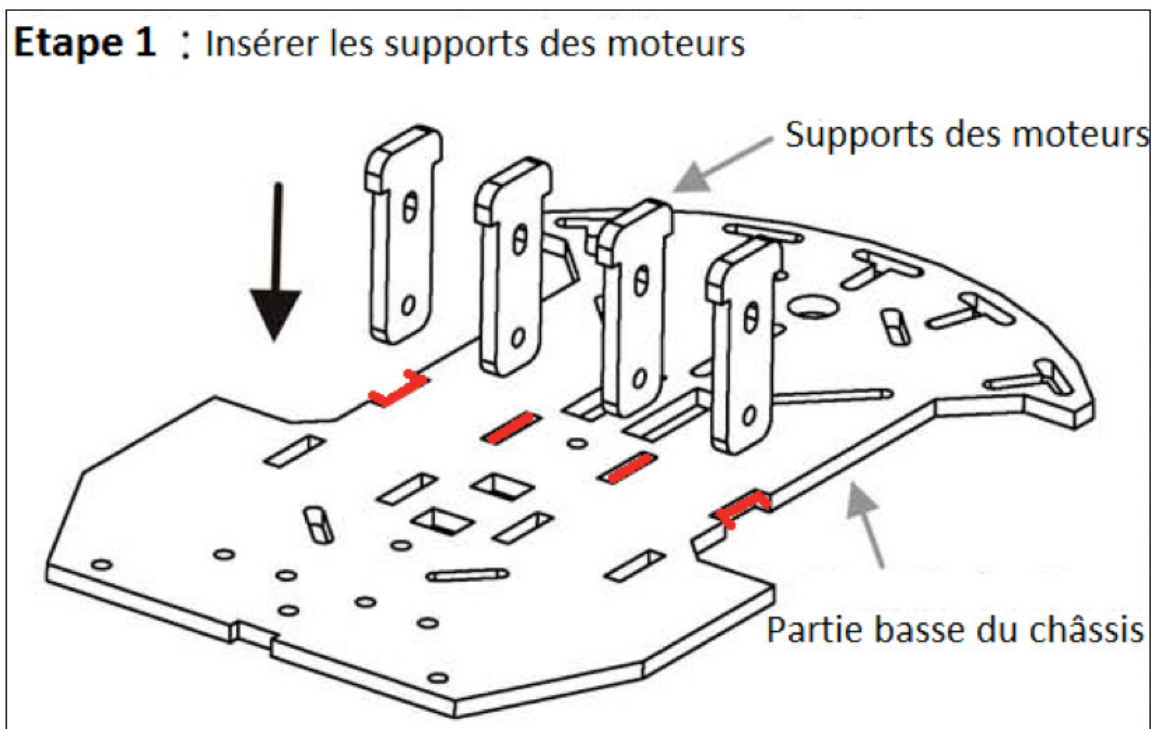
Liste des composants pour la première partie du robot :

 <p>1 Chassis-up 1PC</p>	 <p>2 Chassis-bottom 1PC</p>	 <p>3 Support de moteurs 4PCS</p>	 <p>4 <i>Optionnel</i> Speed board holder 2PCS</p>	 <p>5 Moteur 2PCS</p>
 <p>6 Roue à bille 1PC</p>	 <p>7 Roue 2PCS</p>	 <p>8 Support de piles 1PC</p>	 <p>9 Entretoise 8PCS</p>	 <p>10 L10 Entretoise 4PCS</p>
 <p>11 M3*30 vis 4PCS</p>	 <p>12 M3*10 vis à tête plat 2PCS</p>	 <p>13 M3*6 vis 22PCS</p>	 <p>14 M3 écrou 2PCS</p>	

### 1 1<sup>ère</sup> étape

Positionnez les quatre supports des moteurs à leur emplacement respectif. Ils seront ensuite fixés sur le châssis une fois les moteurs positionnés et vissés.

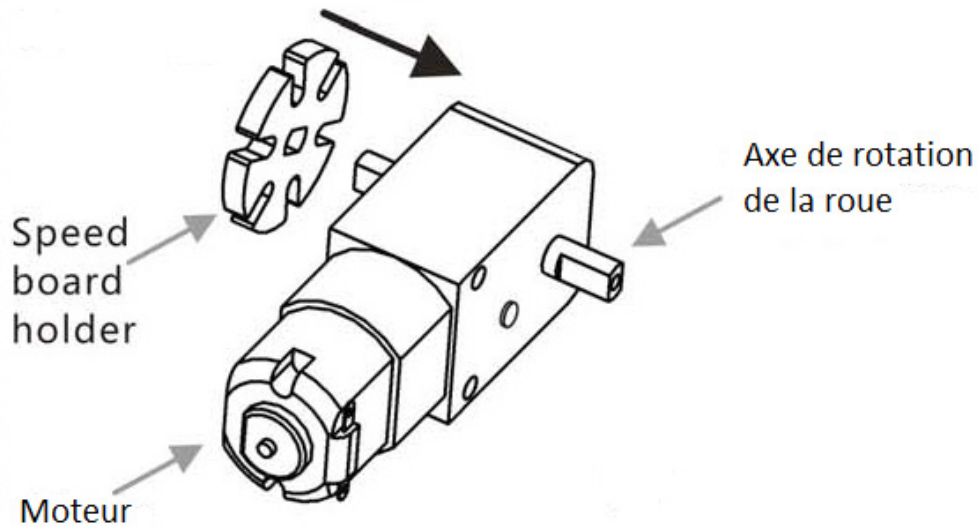
#### Etape 1 : Insérer les supports des moteurs



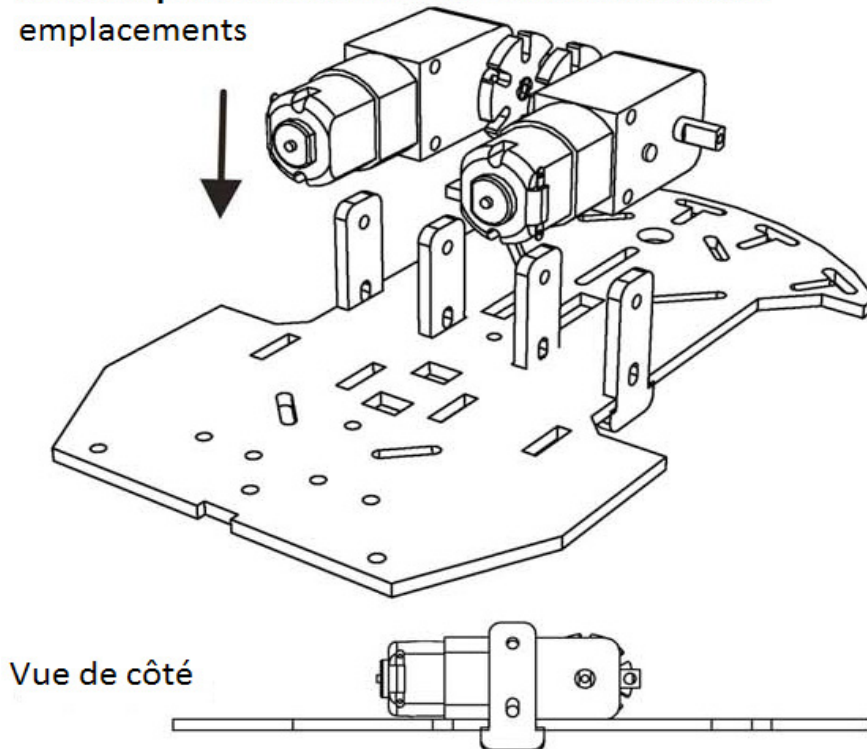
## 2 2<sup>ème</sup> étape (optionnel)

Retourner le châssis. Assemblez le « speed board holder » sur l'axe de rotation de la roue de chaque moteur. Il ne faut pas les enfoncer entièrement pour pouvoir ensuite les positionner correctement et éviter qu'elles frottent contre le châssis. Cette pièce vous permettra à l'aide d'un capteur additionnel de mesurer la vitesse de rotation du moteur.

### Etape 2: Assembler les moteurs



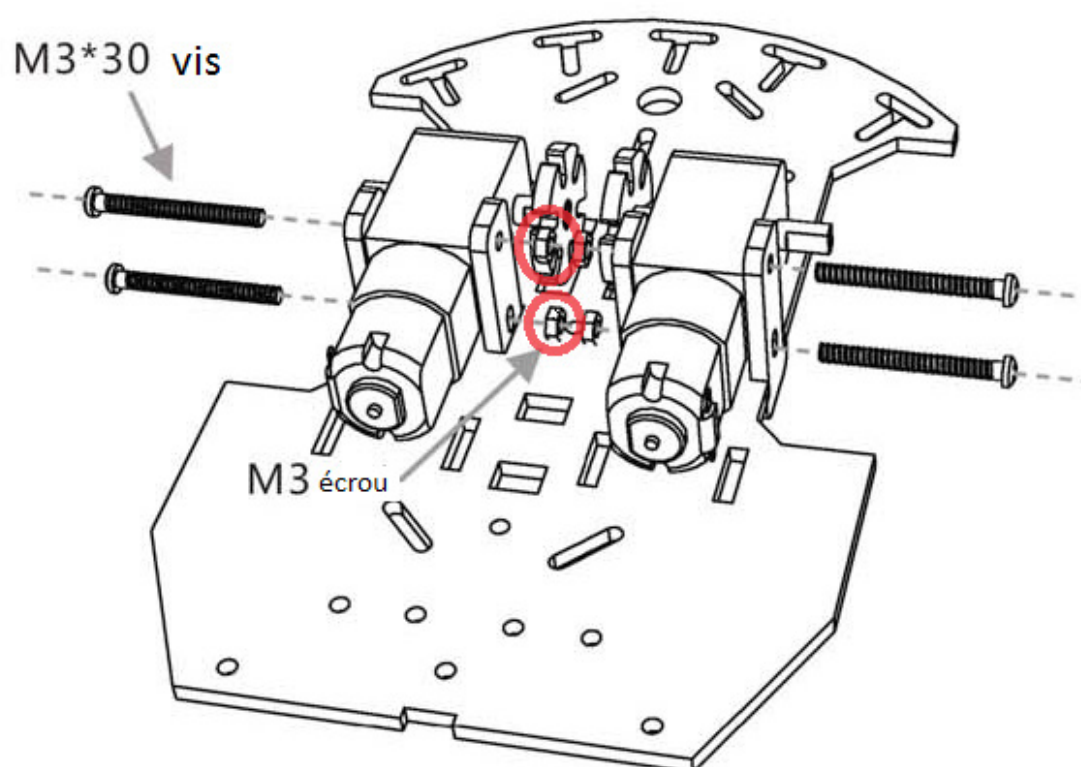
### Suite étape 2: Positionner les moteurs dans leurs emplacements





**3 3<sup>ème</sup> étape**

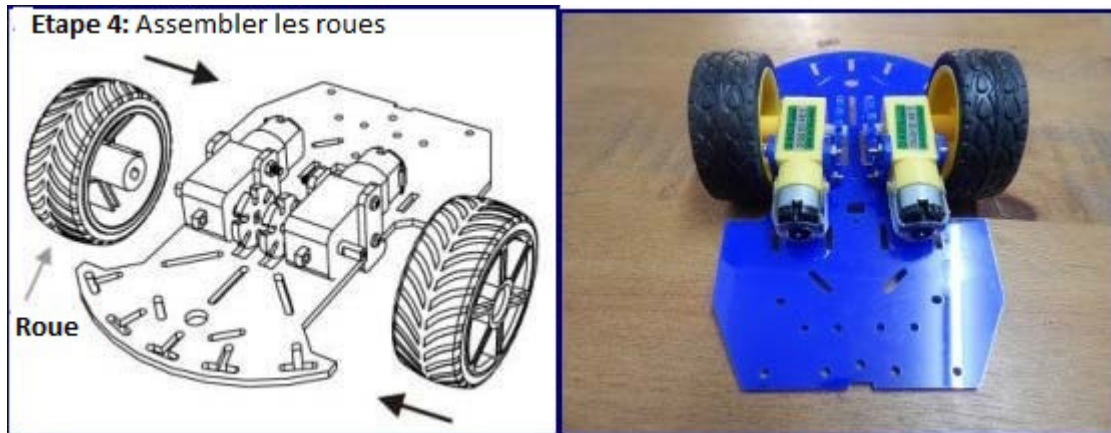
Une fois le moteur positionné sur le châssis entre les deux supports, on peut visser l'ensemble avec les vis M3\*30, deux vis par support. Attention, les vis peuvent être un peu difficiles à insérer. Les animateurs peuvent donc aider les enfants pour visser les moteurs. Les deux vis de bas sont optionnelles.

**Etape 3: Visser les moteurs**

Vue de l'autre côté

#### 4 4<sup>ème</sup> étape

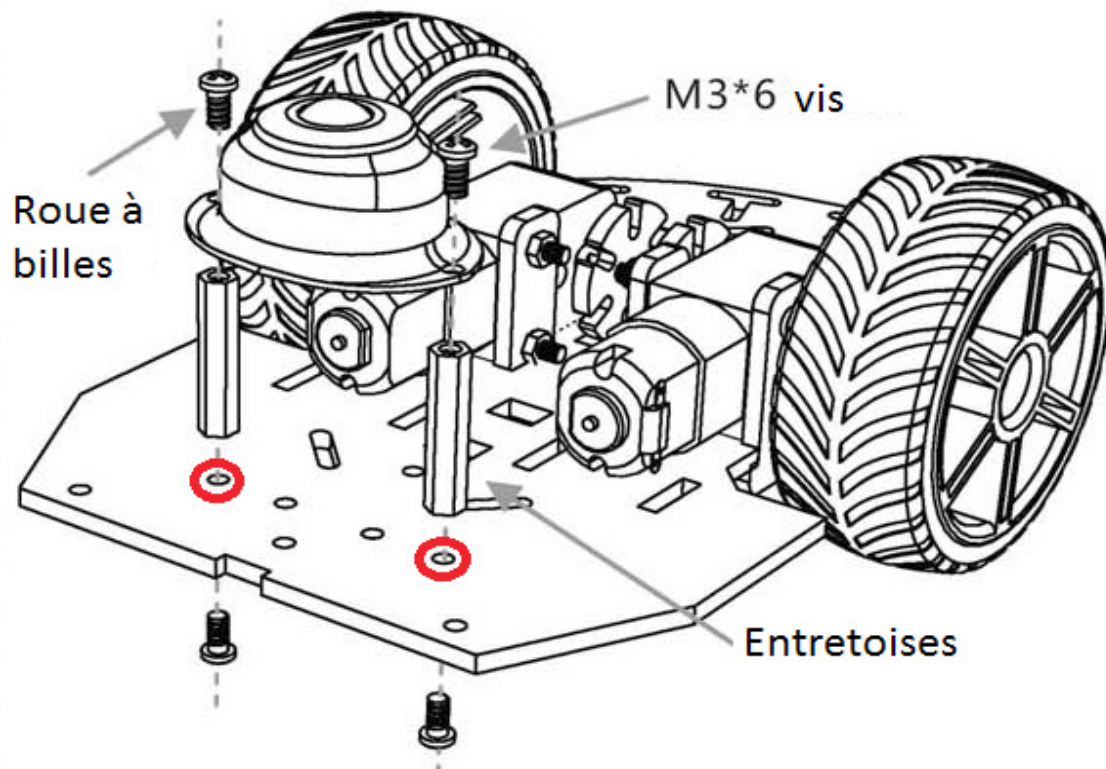
Ajoutez les deux roues de chaque côté des moteurs, sur les axes de rotation. De même, évitez de les emboîter entièrement pour ne pas qu'elles frottent contre le châssis.

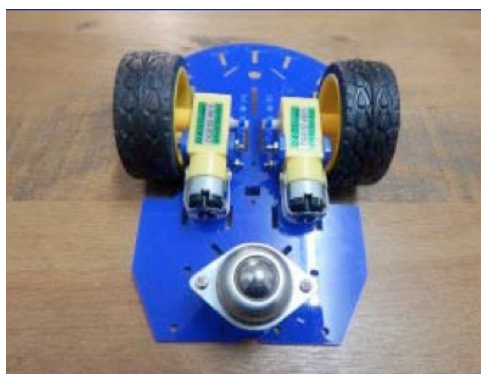


#### 5 5<sup>ème</sup> étape

On vient à présent fixer la roue à bille sur le châssis. Comme indiqué sur la notice du montage, on vient placer la roue sur deux entretoises (L25 spacers) et on visse l'ensemble avec quatre petites vis (M3\*6 screws).

#### Etape 5: Assembler la roue à bille



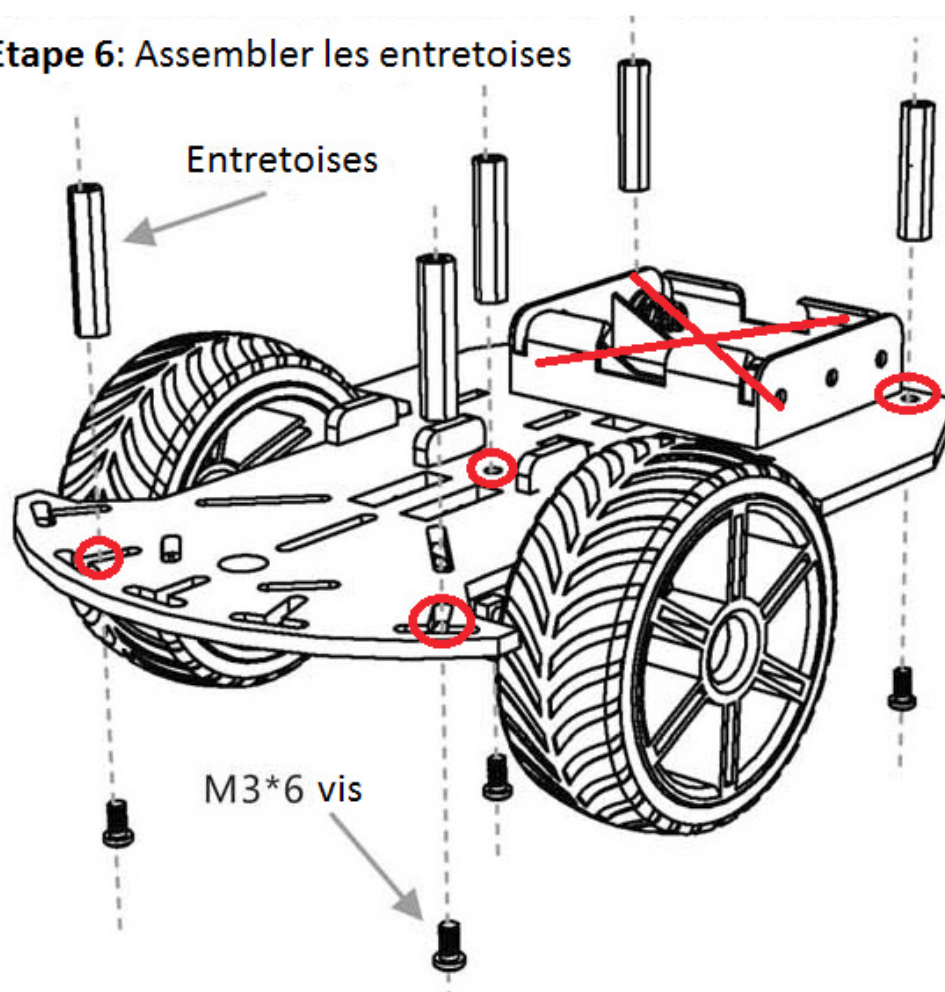


Vue de l'autre côté

### 6 6<sup>ème</sup> étape

Pour pouvoir ensuite fixer la partie supérieure du châssis, on ajoute 5 entretoises (L25 spacers) sur notre montage. On vient visser ses entretoises avec les vis M3\*6 aux emplacements indiqués par des points rouges sur la photo ci-dessus. Pour les deux du haut, il faut bien les visser sur la « base » du T pour pouvoir positionner correctement la plaque supérieure du châssis dans l'étape suivante.

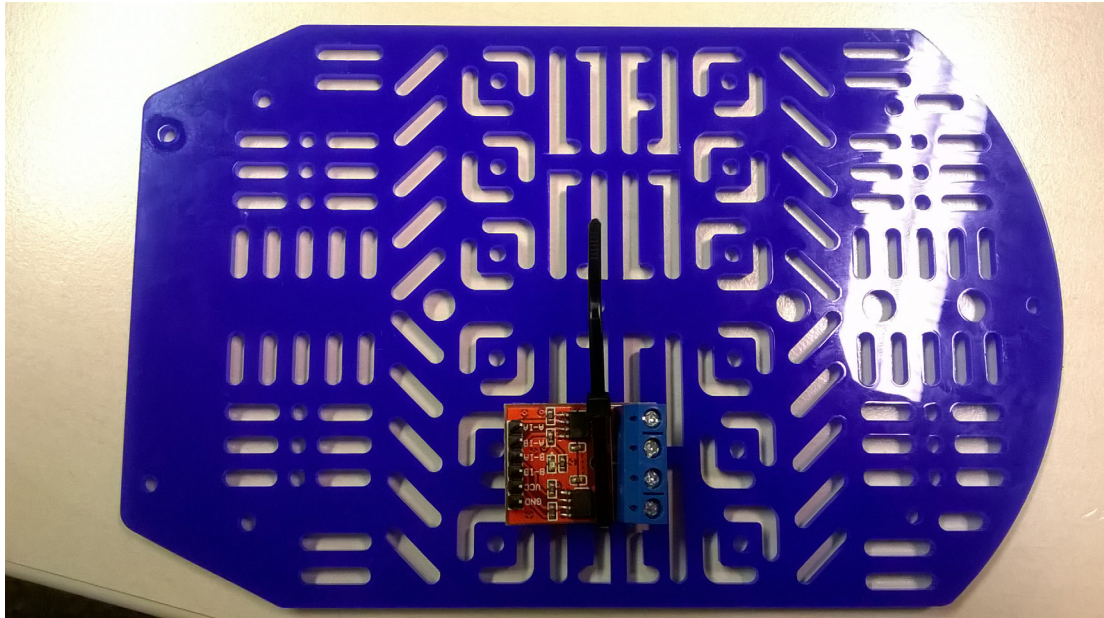
#### Etape 6: Assembler les entretoises





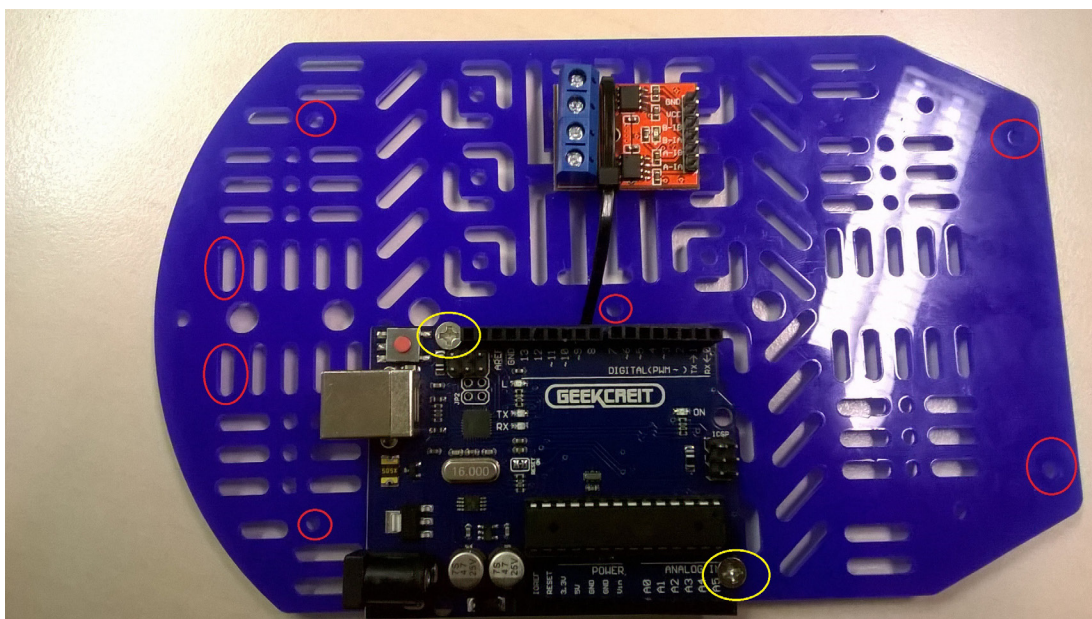
### 7 7<sup>ème</sup> étape

Fixer le contrôleur de moteur sur la partie haute du châssis en utilisant le support noir.



### 8 8<sup>ème</sup> étape

Visser la carte Arduino (montrer en jaune sur la photo). Vous aurez besoin de deux vis à tête plate. Attention à ne pas cacher les trous pour les 7 vis (montrer en rouge sur la photo) dont on aura besoin plus tard.

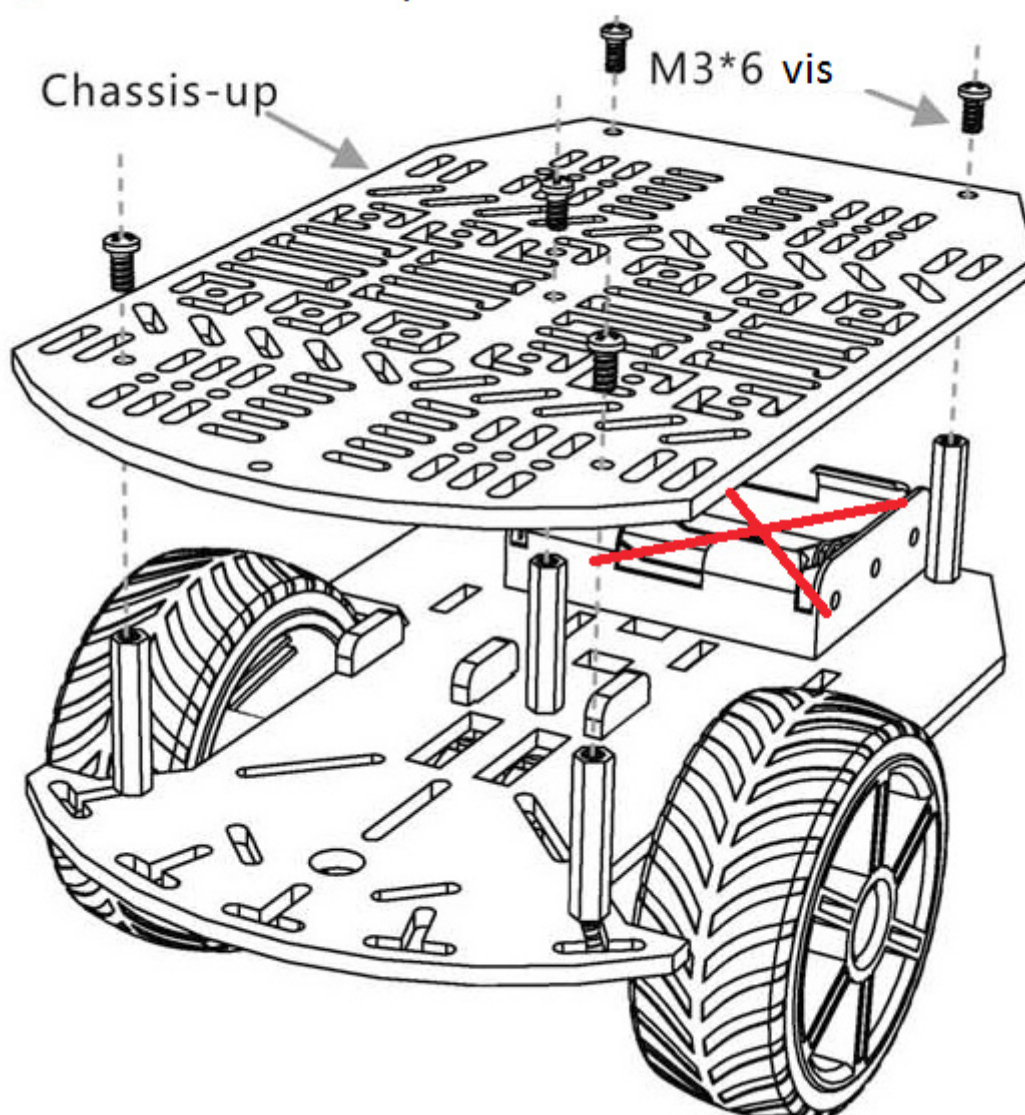


Une option plus pratique serait de fixer la carte avec du scotch double face.

### 9 9<sup>ème</sup> étape

On peut maintenant venir fixer la plaque supérieure du châssis sur les entretoises à l'aide de cinq vis M3\*6.

#### Etape 9 : Assembler la partie haute du châssis

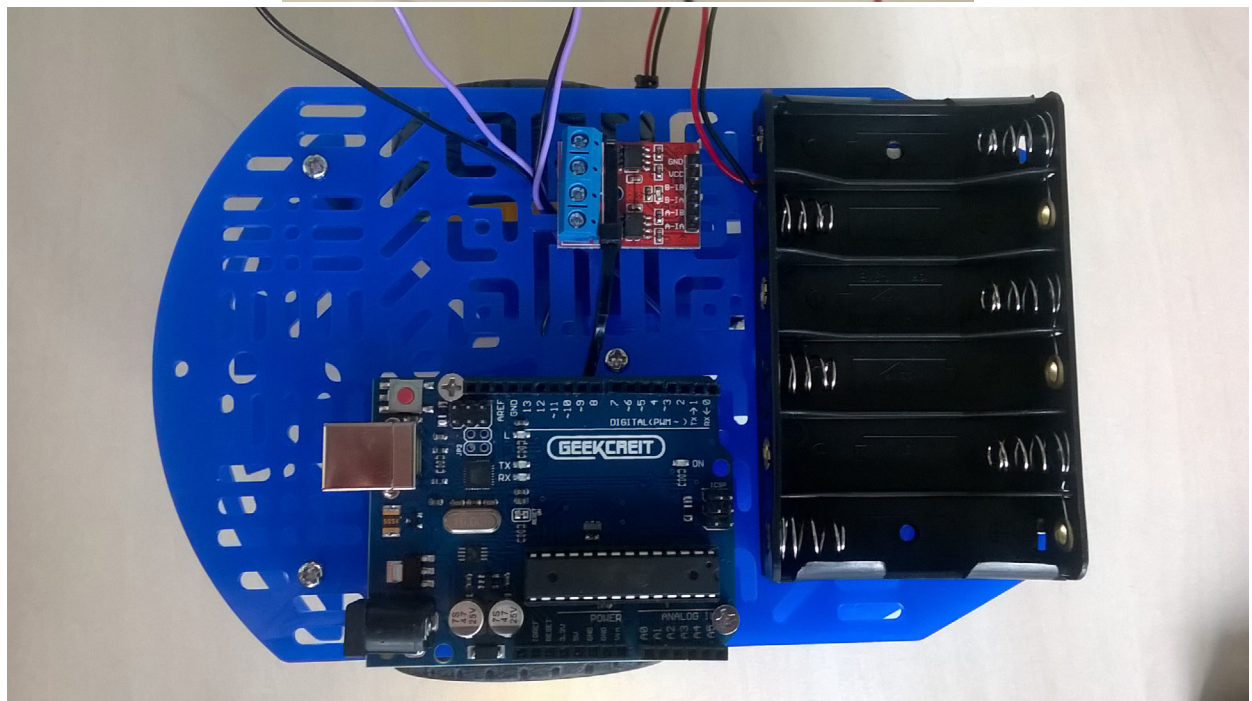
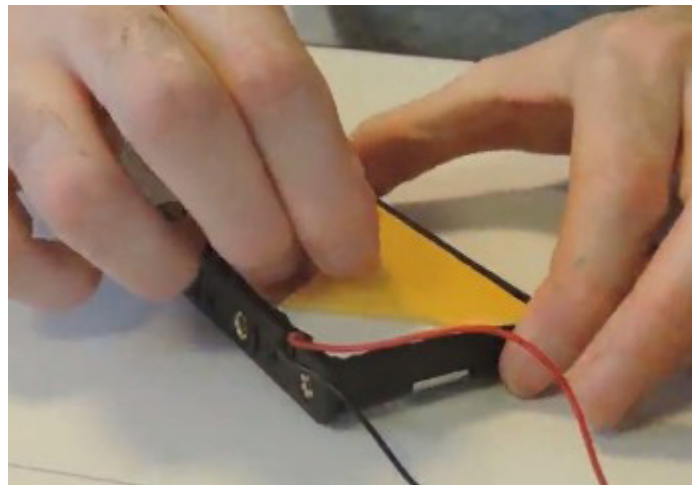


### 10 10<sup>ème</sup> étape

Enfin, vous pouvez coller ou visser le support de piles à la partie basse ou haute du châssis. Nous vous conseillons la partie haute car c'est plus pratique pour changer les piles.

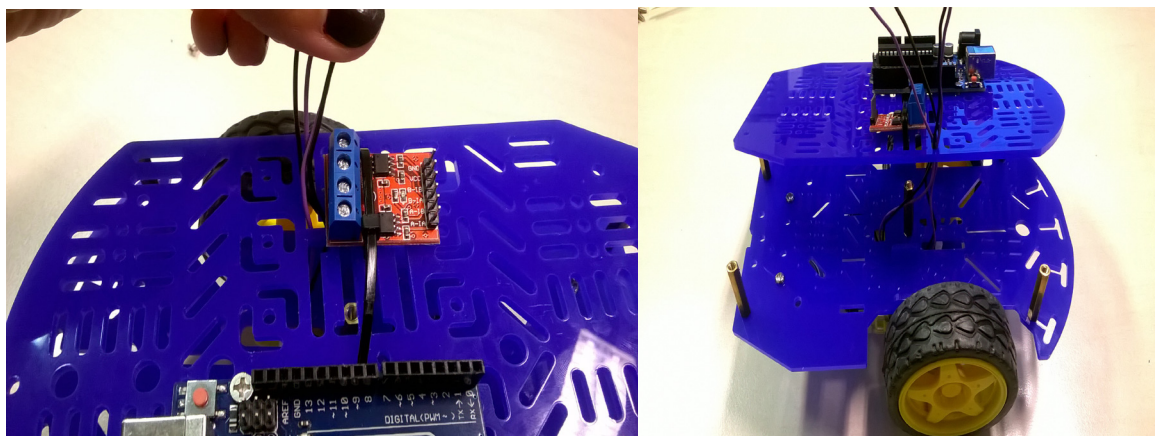
Il peut être maintenu par deux vis à tête plate (M3\*10). Il faut forcer un peu pour faire rentrer les vis dans les trous déjà réalisés dans le support de piles. Sinon, l'utilisation du scotch double face est bien plus pratique pour retirer le support lors du changement des piles.





**11 11<sup>ème</sup> étape**

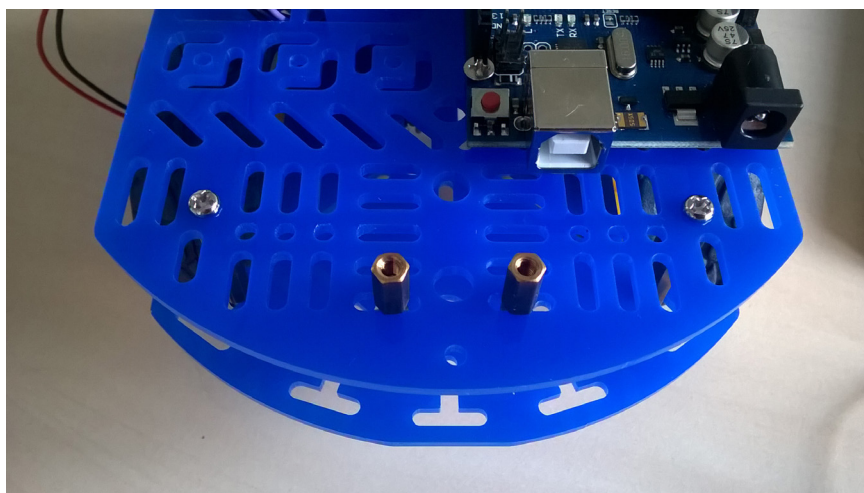
Passez les fils (2 violets et 2 noirs) des moteurs via les deux parties du châssis. Il faut qu'ils soient près du contrôleur de moteurs.

**12 12<sup>ème</sup> étape**

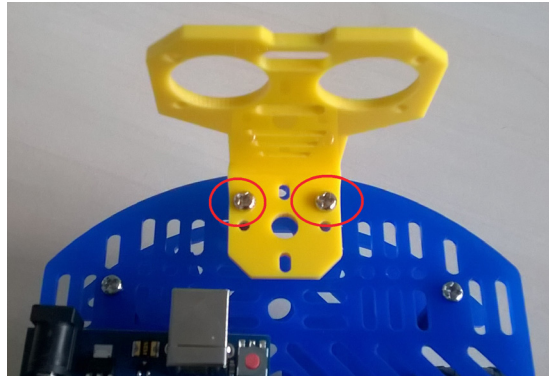
**Deuxième partie : montage de la tête du robot.**

Pour cet étape vous avez besoin de 4 vis M3\*6 et 2 entretoises.

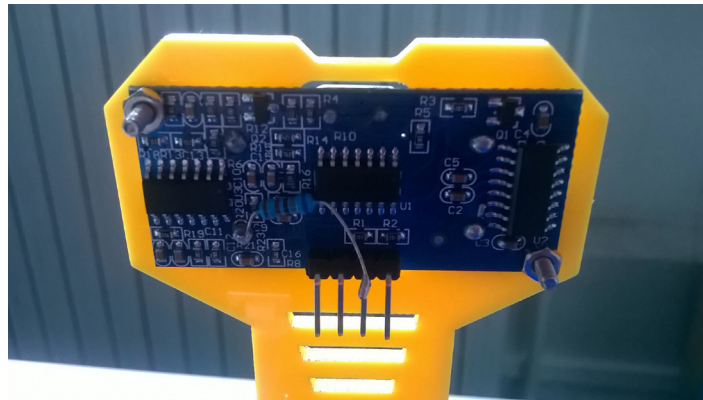
Fixez d'abord les deux vis avec les deux entretoises sur l'avant du châssis. Conseil : ne vissez pas trop fort pour que vous puissiez bouger les entretoises afin de fixer le support de la tête.



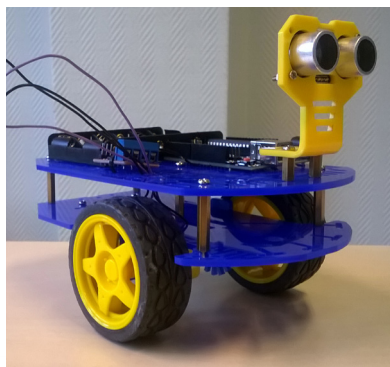
Ajoutez ensuite, le support de la tête est fixé-le en utilisant les deux vis restantes.



Pour finir, vissez le capteur de distance, en utilisant deux vis M3\*30 et deux écrous. Conseil : vous pouvez faire cet étape après la séance 3 « câblage » de ROSA.



**Et voilà, vous avez réussi !  
ROSA est prête !**

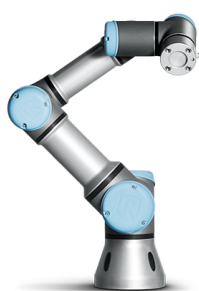


Retrouvez cette annexe sur :  
<https://frama.link/montagechassis>

## Les différents types de robots

### Robot Mêkhano centré :

Robots industriels pour manipulation de pièces ou travail spécifique sur une chaîne de fabrication.



Robot Tripod

### Robot Anthropocentré - Humanoïde :

Un **robot humanoïde** ou androïde est un robot dont l'apparence générale rappelle celle d'un corps humain. Généralement, les robots humanoïdes ont un torse avec une tête, deux bras et deux jambes, bien que certains modèles ne représentent qu'une partie du corps, par exemple à partir de la taille. Certains robots humanoïdes peuvent avoir un « visage », avec des « yeux » et une « bouche ».



Robot NAO



Robot Poppy



Robot Atlas de Boston Dynamics

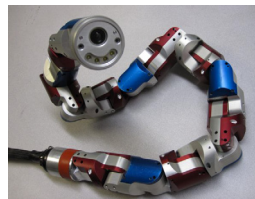


### Robot zoo ou bio centré :

Robot qui ont des formes des animaux.



Robot de Boston Dynamics



Robot Snake

### Quelques robots faits avec une carte Arduino :



Image récupérée sur

<http://duino4projects.com/>

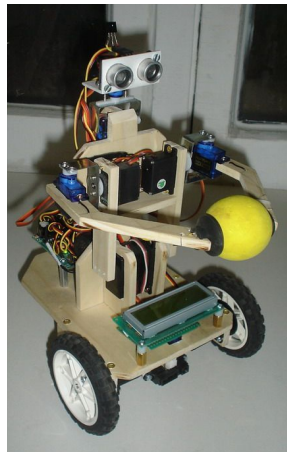


Image récupérée sur

<https://seriousrobotics.files.wordpress.com>



Image récupérée sur

<http://cdn.instructables.com/>



**Nous vous proposons une liste de vidéos qui montrent les différents robots déjà existant et/ou de différentes situations où l'existence d'un robot peut apporter des choses supplémentaires à l'activité humaine.**

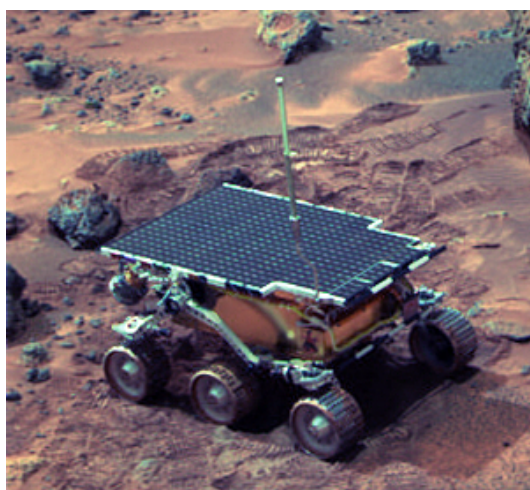
Ces vidéos peuvent être utilisées pour emmener une réflexion sur:

- la place que les robots occupent dans notre société
- la place que les robots occuperont dans le futur proche
- l'utilité des robots dans les différents domaines etc.

Titre	Description	Lien
<b>DEEPSEA CHALLENGE 3D Trailer</b>	La conception du "DEEPSEA CHALLENGER", une machine faite pour aller au point le plus profond dans l'océan.	<a href="https://youtu.be/-8r_-79SjpA">https://youtu.be/-8r_-79SjpA</a> <a href="http://www.deepseachallenge.com/">http://www.deepseachallenge.com/</a>
<b>Atlas, The Next Generation</b>	Boston Dynamics présente Atlas, une nouvelle génération de robot humanoïde. Atlas est alimenté par une batterie. Il utilise plusieurs capteurs dans son corps et ses jambes pour rester en équilibre. Il a des capteurs stéréo dans sa tête pour éviter les obstacles et l'aider à manipuler les objets. Cette version d'Atlas fait 1m75 et pèse 82 kg.	<a href="https://www.youtube.com/watch?v=rVlhMGQgDKY">https://www.youtube.com/watch?v=rVlhMGQgDKY</a>
<b>Introducing Spot</b>	La société Boston Dynamics présente son nouveau robot chien qui s'appelle Spot. Spot pèse 73 kilos, il est alimenté de manière électrique et peut courir librement grâce à un moteur embarqué. Ce robot possède des capteurs au niveau de la tête, lui permettant de se déplacer aisément dans un terrain difficile d'accès.	<a href="https://www.youtube.com/watch?v=M8YjvHYbZ9w">https://www.youtube.com/watch?v=M8YjvHYbZ9w</a> <a href="http://www.clubic.com/mag/trendy/actualite-753657-spot-robot-chien-boston-dynamics.html">http://www.clubic.com/mag/trendy/actualite-753657-spot-robot-chien-boston-dynamics.html</a>
<b>Festo Tripod EXPT - the parallel kinematic robots</b>	Tripod est un manipulateur robotisé haute vitesse pour une liberté de mouvement dans l'espace. Précision et réponse dynamique élevée jusqu'à 150 prélèvements/min.	<a href="https://www.youtube.com/watch?v=aH_t_1-">https://www.youtube.com/watch?v=aH_t_1-</a>
<b>Presentation of UR3: The world's most flexible, light-weight table-top robot</b>	Les bras robotiques d'Universal Robots sont faits pour être intégrés dans les environnements de production existants. Avec six points d'articulation, et une grande amplitude de mouvement, ils sont conçus pour imiter la diversité de mouvements du bras humain.	<a href="https://www.youtube.com/watch?v=NGITRErHkPstI40">https://www.youtube.com/watch?v=NGITRErHkPstI40</a> <a href="http://www.universal-robots.com/fr/">http://www.universal-robots.com/fr/</a>

## Les robots

Parmi les objets qui embarquent des ordinateurs, les « robots » ont une importance scientifique, sociétale et économique grandissante. Les robots sont partout : dans les usines et dans les champs, au fond des mers et dans l'espace, dans les jardins et les salons. En outre, ils ont pénétré notre culture et certains d'entre eux participent au renouvellement de la vision que nous avons de nous-mêmes.



*Quelques exemples de robots. De gauche à droite et de bas en haut: bras mécaniques utilisés dans l'industrie automobile ; robot humanoïde dédié à la reconnaissance et à la reproduction des émotions; robot explorateur sur Mars ; robot militaire destiné à transporter des charges lourdes sur des terrains accidentés.*

D'un point de vue technique, un robot est une machine dotée de capteurs (de contact, de distance, de couleur, de force, ...) qui lui permettent de percevoir son environnement, de

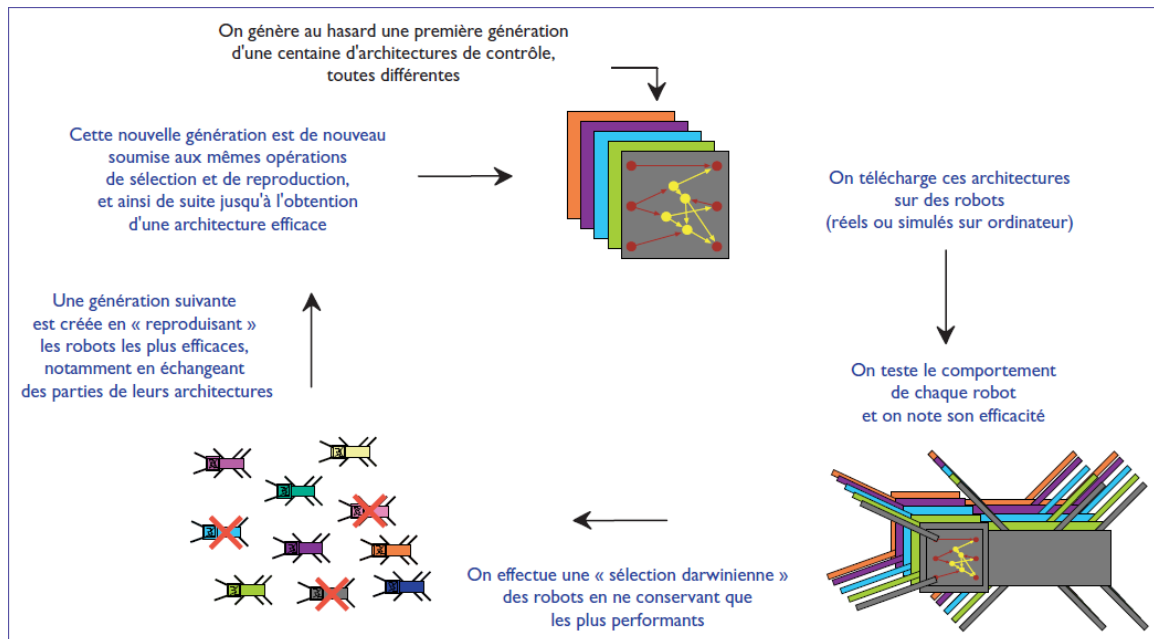
moteurs l'autorisant à bouger et à agir sur cet environnement, et d'un système qui contrôle ce qu'effectue le robot en fonction de ce qu'il perçoit. Une caractéristique fondamentale des robots, qui les distingue des automates, est cette rétroaction entre perception et action. Les automates (tels que ceux de Jacques de Vaucanson ou de Pierre et Henri-Louis Jaquet-Droz au XVIIIe siècle) ne sont pas des robots car leurs mouvements ne dépendent pas de ce qui se passe autour d'eux : ils n'ont pas de capteurs et leurs enchainements sont entièrement prédéterminés par le programme.

En pratique, cette définition d'un robot recouvre une très vaste diversité de machines : les bras articulés programmables des usines automobiles, les voitures (en mode d'assistance à la conduite) et les avions qui sont aujourd'hui largement automatisés, les aspirateurs qui font le ménage tout seuls, certains objets électroniques des magasins de jouets, ou les robots biomimétiques en forme d'animaux (singes, poissons...) que l'on rencontre parfois dans les laboratoires de recherche.

Cette diversité n'est pas seulement une diversité de formes, c'est aussi une diversité d'usages, qui sous-tendent une diversité de logiques de fonctionnement. Ainsi, les robots se spécialisent selon deux axes majeurs : l'autonomie et les capacités d'adaptation et/ou d'apprentissage.

**L'autonomie** : il existe des robots qui agissent sans qu'un humain les guide, et d'autres dont le comportement est, soit influencé par un humain, soit presque totalement contrôlé par un humain. Par exemple, dans une usine, les robots qui travaillent à la chaîne et répètent toujours le même geste le font souvent de manière autonome. En revanche, les robots utilisés dans les centrales nucléaires (pour opérer dans les zones à forte radiation) sont typiquement téléguidés par un humain qui leur dit où aller et quoi faire après chaque action.

**L'adaptation et l'apprentissage** : le comportement de certains robots est figé au départ et une fois pour toutes par le programmeur, alors que d'autres robots sont capables d'acquérir de nouveaux comportements et de nouvelles connaissances par leurs expériences : leur comportement évolue en fonction de l'histoire de leurs interactions avec l'environnement. Ainsi, certains robots sont capables d'apprendre à reconnaître des objets dans des images ou encore d'apprendre à marcher en expérimentant et évaluant eux-mêmes différentes stratégies. Ces mécanismes d'adaptation sont permis par des « algorithmes d'apprentissage », qui reposent sur la détection automatique de régularités dans des flux de données captées par le robot, et sur des méthodes d'« optimisation » permettant de raffiner de manière progressive et itérative les paramètres des solutions à un problème.



Exemple de processus itératif permettant à des « générations » de robot d'apprendre à réaliser une tâche par eux-mêmes (par exemple, se déplacer).

Grâce à ces algorithmes d'apprentissage, certains robots sont capables d'inventer des solutions et des comportements non prédits par leur concepteur, et même de sélectionner pour eux-mêmes des objectifs qui ne sont pas préprogrammés. Il est par exemple possible de programmer un robot en lui donnant des instructions lui demandant de rechercher des situations nouvelles afin d'augmenter ses connaissances sur le monde qui l'entoure. Ainsi, certains algorithmes permettent de doter ces machines de formes d'apprentissage et de créativité. Néanmoins, les capacités et les performances de ces algorithmes sont aujourd'hui, et encore pour très longtemps probablement, très faibles en comparaison des capacités d'adaptation et de raisonnement de nombreux animaux, et en particulier de l'humain.

Comme nous l'avons vu, ces différentes logiques de fonctionnement répondent à différents besoins : il existe en effet une grande variété de raisons, et donc de fonctions, pour lesquelles les robots sont construits et utilisés. On peut en particulier considérer trois familles de fonctions : *travailler et explorer*, *assister l'humain*, *modéliser les mécanismes cognitifs et comportementaux du vivant*.

### Travailler et explorer.

La plupart des robots en service dans le monde sont industriels (on en compte aujourd'hui environ 9 millions). Très tôt, les entreprises se sont intéressées à ces machines pour deux raisons :

- D'abord, les robots peuvent être utilisés pour remplacer les travailleurs humains dans les tâches répétitives, pénibles et nécessitant de faibles compétences, comme les montages, la peinture ou les soudures de pièces ;

- En outre, ces machines sont capables de réaliser des travaux à la chaîne bien plus rapidement et efficacement que les humains.

Le premier robot industriel, *Unimate*, est apparu en 1961 : installé dans une usine automobile de General Motors, ce bras articulé manipulait de lourdes pièces de fonderie. Dans les années 1970, l'usage des robots dans l'industrie a pris son envol. Aujourd'hui, les robots ont pénétré toutes les branches de l'industrie et ne sont plus restreints au seul secteur de l'automobile. Par exemple, dans l'agriculture et l'agro-alimentaire, les robots vont dans les champs cueillir fruits et légumes, certains coupent, pressent et mettent en bouteille ; d'autres trient et mettent en cartons ; d'autres encore groupent en palettes. Dans certains aéroports, des flottes de robots s'occupent de transporter les bagages et de les charger dans les soutes.

Les robots ne sont pas seulement utiles dans l'industrie pour les tâches simples et répétitives, ils sont aussi utilisés pour travailler dans des environnements dangereux pour l'homme. L'industrie nucléaire est un exemple typique. Qu'ils soient autonomes ou partiellement téléguidés, les robots des centrales nucléaires peuvent déambuler dans les enceintes confinées et radioactives, ils peuvent manipuler les substances dangereuses et s'occuper de la maintenance des autres machines. Un autre exemple est l'industrie pétrolière : les robots sous-marins sont par exemple utilisés pour contrôler l'état de la coque des navires afin de prévenir les accidents et d'identifier les « navires oubliés ». Les robots sont enfin cruciaux pour l'exploration des endroits où l'homme ne peut pas aller, au premier rang desquels l'espace et les corps du Système solaire. C'est en 1966 que le premier robot mobile arrive sur la Lune, embarqué dans la sonde *Surveyor*. Suivront le soviétique *Lunokhod*, puis toute la série des américains *Mariner*. En 1997, un robot atterrit sur la planète Mars : *Sojourner*, qui est propulsé par l'énergie qu'il capte grâce à ses panneaux solaires ; il envoie à la Terre des milliers de clichés et provoque un engouement du grand public. *Sojourner* navigue en partie de manière autonome car, étant donné la distance avec la Terre, il est très difficile de le téléguider en temps réel. En 2004, une nouvelle mission robotisée concentre l'attention du monde entier : *Spirit* et *Opportunity*, équipés de spectromètres et d'un bras qui leur permet de creuser la surface, apportent la preuve que de l'eau a coulé sur Mars.



### Assister l'humain dans son quotidien.

Si le XXe siècle a vu l'avènement des robots travailleurs et explorateurs, à l'aube du XXIe siècle, une autre grande famille de robots prend son essor : les robots d'assistance à la personne. Dans nos maisons, les robots d'assistance aux travaux ménagers se démocratisent. On retrouve aussi des assistants robotisés dans les magasins ou sur nos lieux de travail. L'activité médicale est particulièrement transformée par la robotique : si les robots d'assistance chirurgicale sont quotidiennement utilisés depuis une quinzaine d'années, on voit émerger aujourd'hui de nouveaux usages. Ainsi, des robots accompagnent les personnes qui ont des difficultés physiques ou cognitives, par exemple pour les aider à se lever et s'asseoir, pour les stimuler cognitivement quand ils ont des problèmes de mémoire, ou pour jouer le rôle de facilitateur des contacts avec la famille ou l'entourage médical. Demain, des chercheurs introduiront d'autres robots, beaucoup plus petits, dans les blocs opératoires : ce sont des capsules endoscopiques miniaturisées capables d'explorer conduits intestinaux, artères ou veines pour aider le chirurgien dans son diagnostic. Apparaissent aussi depuis quelques années des prothèses robotisées de mains ou de bras entières destinées aux amputés.



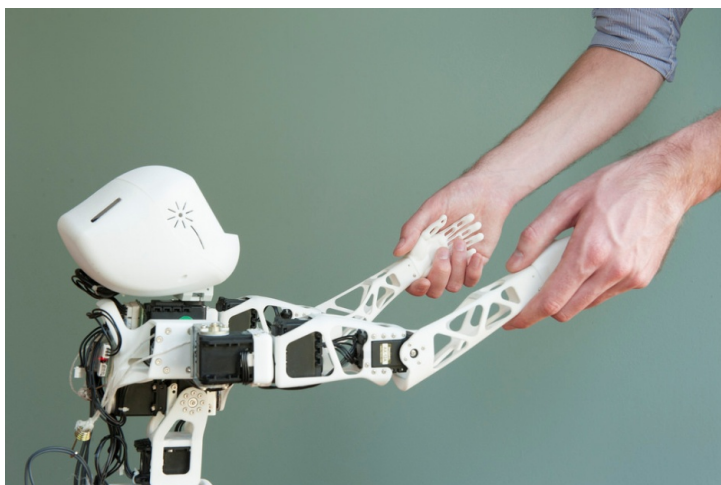
Prothèse robotisée. ©BeBionic

### Modéliser le vivant et la cognition.

Les robots sont devenus essentiels comme outils pour penser et modéliser les systèmes complexes dans le domaine du vivant. En particulier, les robots sont aujourd'hui utilisés dans des laboratoires de recherche pour comprendre comment les êtres vivants s'adaptent à leur environnement naturel, en termes de comportement et de déploiement de capacités cognitives. En effet, ce comportement est le résultat de la dynamique des interactions entre un cerveau, un corps physique et un environnement, dynamique qui se redéfinit en permanence car le cerveau est modifié à chaque nouvelle interaction. Les robots embarquent à la fois un « cerveau » (des programmes qui permettent de traiter

l'information acquise par le robot, selon des règles précises), des « organes sensoriels » (les senseurs) et des « systèmes moteurs » (les actuateurs). De cette manière, les robots peuvent, comme les êtres vivants, modifier et être modifiés par l'environnement physique dans lequel ils agissent. Leur « cerveau » aussi en est modifié, car le robot acquiert de nouvelles connaissances qu'il peut réutiliser dans ses interactions successives.

Les chercheurs peuvent étudier la complexité de l'interaction cerveau-corps-environnement, grâce à des expérimentations possibles sur des robots et impossibles sur des êtres vivants – comme par exemple « éteindre » une partie de leur cerveau artificiel pour voir comment le comportement est modifié ou altérer des parties du corps. Certains laboratoires étudient ainsi le contrôle moteur, la perception visuelle, le repérage dans l'espace, ou encore les mécanismes de l'apprentissage et de l'évolution de la parole et du langage chez l'humain. Dans le cadre de ces projets, les interactions avec les neurosciences, la biologie, la psychologie ou encore l'éthologie jouent un rôle central.





*Le robot humanoïde Poppy est « open source ». Il est développé par l'Inria à la fois pour des recherches sur la cognition et pour des projets éducatifs. Plus de détails sur: <https://www.poppy-project.org>*



## Montage du châssis

Le Magician châssis est une plateforme robotique facile à assembler soi-même. Il fonctionne grâce à deux moto-réducteurs à courant continu. Il propose un support 4 piles AA pour l'alimentation mais il est conseillé d'utiliser un support 6 piles. Une fois le châssis monté, on peut l'équiper de capteurs ou d'actionneurs pour en faire un véritable robot intelligent et autonome.

*Les 2 roues à disque permettent d'y connecter des encodeurs. Montage facultatif.*

Notices	Fichiers
<b>Notice d'assemblage du châssis en français</b>	<a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:montage_chassis_v2.pdf">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:montage_chassis_v2.pdf</a>
<b>Notice de montage du châssis en anglais</b>	<a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:magicianchassisinst.pdf">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:magicianchassisinst.pdf</a>
<b>Montage complet du robot</b>	<p> Lecture de la vidéo <a href="http://rosa.lph.bzh/montage.html">http://rosa.lph.bzh/montage.html</a></p> <p> Télécharger la vidéo <a href="http://rosa.lph.bzh/videos/montage_robot.zip">http://rosa.lph.bzh/videos/montage_robot.zip</a></p>

*Si vous avez des problèmes pour fixer la carte Arduino sur le châssis, je vous conseille de le faire avec du scotch double face.*



Retrouvez cette annexe sur :  
<https://frama.link/montageChassis>

## Quelques informations sur la partie mécanique/électronique de ROSA

### Les capteurs

Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée (comme la température, **la distance**, la luminosité, le débit, le niveau, la pression, le son) en une mesure utilisable. Il permet donc d'acquérir des données, des nombres, comme une amplitude de courant ou de tension, une hauteur de mercure pour un thermomètre, ou encore une distance de déviation d'une bulle pour un niveau. Tout capteur doit être associé à un contrôleur pour pouvoir donner une information manipulable : par exemple dans le cas d'un capteur de distance à infrarouges, le capteur envoie un faisceau infrarouge, et détecte la réflexion de ce faisceau par un objet. Il faut alors qu'un petit contrôleur calcule le temps qu'a mis le faisceau à être réfléchi pour pouvoir déterminer la distance à laquelle se trouve l'objet.

Le capteur que nous utilisons pour ROSA c'est un capteur de distance, autrement dit un capteur Ultrason.



### Les actionneurs

Un actuateur, ou actionneur, est un dispositif d'une machine (**moteur**, vérin, speaker, lampe, etc.) qui permet à une machine de transformer l'énergie qui lui est fournie en un phénomène physique utilisable. Un actionneur exécute les ordres qui lui sont envoyés par la partie commande de la machine.

ROSA utilise deux moteurs, deux roues à vitesse variable qui lui permettent d'avancer, de reculer et de tourner.



*Il s'agit ici de faire comprendre aux enfants que chez les robots rien n'est naturel, et qu'ils ont besoin d'un petit outil pour sentir chacune de ces choses. Par exemple s'ils n'ont pas de capteurs de distance, ils ne peuvent pas voir !*

### Qui sert à ROSA de cerveau ?

Comment fait-elle pour réfléchir, pour réagir ? Tout cela est géré par son programme. Vous pouvez ici introduire la notion de programme sans trop la détailler, car les enfants la comprendront en manipulant lors des activités suivantes.

## La transformation des mouvements – Robotique

### Translation

Une translation est une transformation géométrique qui correspond à l'idée intuitive de « glissement » d'un objet, sans rotation, retournement ni déformation de cet objet.

### Rotation

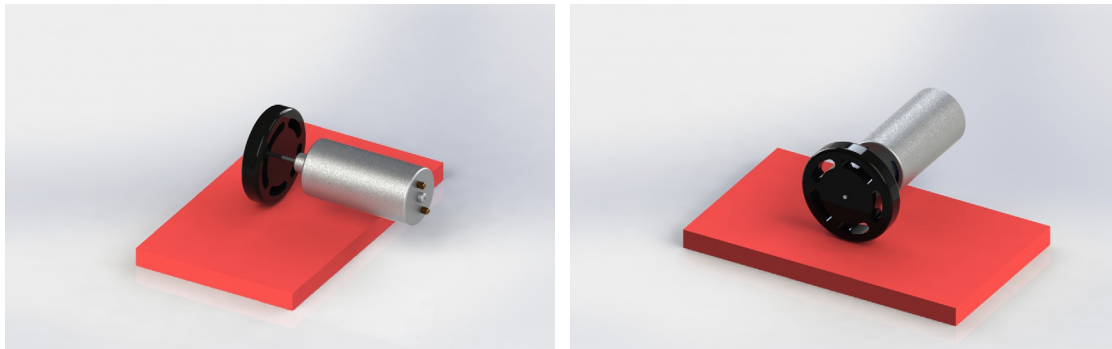
La rotation (du latin rotare : « tourner ») est le mouvement d'un corps autour d'un point ou d'un axe. La rotation d'un moteur caractérise de mouvements circulaires.

Les transformations de mouvements qui nous intéressent en robotique, sont des systèmes qui utilisent la rotation d'un moteur électrique (l'axe du moteur tourne) pour réaliser un mouvement de translation (gauche à droite, haut en bas, avant en arrière).

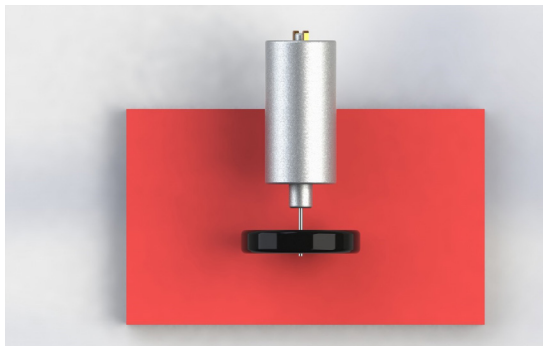
### La roue

Le système le plus connu est la roue. Si on colle une roue à l'axe du moteur, la roue tourne et fait avancer ou reculer un véhicule. Le moteur doit être collé au châssis du véhicule.

Voici une roue vue de côté. La roue déplace le châssis sur un sol fixe :

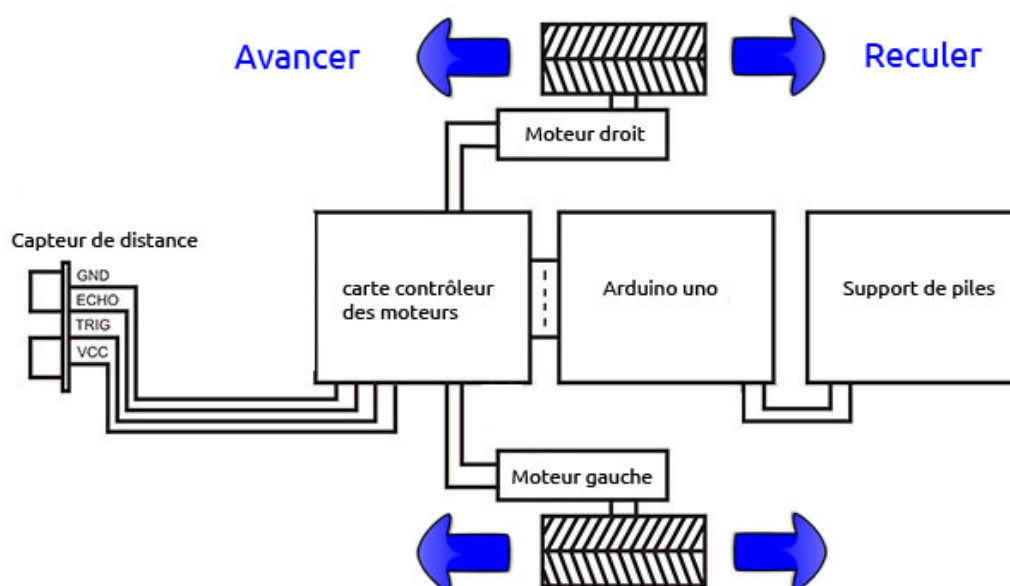


Vu de haut, la roue est collée à l'axe du moteur.



Ce système est utilisé dans les voitures et les trains. Les roues tournent et font avancer le véhicule sur la route.

## Déplacement du robot



Le robot peut:

- Avancer
- Reculer
- Tourner à droite
- Tourner à gauche
- Tourner sur place

Le tout plus ou moins rapidement, en changeant la vitesse des moteurs dans le code.



Retrouvez cette annexe sur:  
<https://frama.link/deplacementRosa>

## Test des moteurs avec une pile

Pour tester le sens des moteurs, il suffit d'utiliser pile AA ou 9v. L'inversion du sens de rotation d'un moteur s'effectue en inversant la polarité de la pile.

 Lecture de la vidéo  
[http://rosa.lph.bzh/test\\_moteurs.html](http://rosa.lph.bzh/test_moteurs.html)

 Télécharger la vidéo  
[http://rosa.lph.bzh/videos/test\\_moteurs.zip](http://rosa.lph.bzh/videos/test_moteurs.zip)



Retrouvez cette annexe sur :  
<https://frama.link/testMoteurs>

## Présentation Arduino

### Introduction

Si on considère qu'un robot est un dispositif **mécatronique** (mécanique, électronique & informatique) pouvant réaliser des tâches programmées de manière autonome, alors il y a longtemps qu'ils sont parmi nous: lave-vaisselle, lave-linge, magnétoscope, sans même parler des robots tondeuses ou aspirateurs! Ce qui les distingue des simples appareils électromécaniques (comme une ancienne machine à laver), c'est leur gestion des tâches à l'aide d'un **microcontrôleur**. Il s'agit d'une sorte de petit ordinateur minimaliste et de faible puissance, mais aussi de faible consommation. C'est lui qui commande les tâches en fonction de son environnement.

Dans le cas d'un **lave-vaisselle moderne**, il s'agit de commander de nombreuses actions (ouverture de vannes, pompes, chauffage de l'eau, ouverture du réservoir de poudre...) selon divers programmes, et en fonction de différents capteurs (porte fermée, arrivée d'eau ouverte, température de l'eau, débitmètre, choix du programme...) et de réagir en fonction de capteurs de pannes (pompe défectueuse, arrivée d'eau fermée, capteur antidébordement...). L'ensemble du dispositif informatique et électronique commandant le lave-vaisselle est appelé **système embarqué**.

Arduino est une famille de cartes contenant un microcontrôleur et de nombreux connecteurs (entrées/sorties) qui peuvent recevoir des informations de capteurs et produire des signaux à même d'interagir avec des moteurs, des **relais** ou d'autres circuits électroniques.

Sa particularité est d'être entièrement Open Source, d'un faible prix et aisément programmable. Dès lors, Arduino est devenu une référence dans le monde des bricoleurs et bidouilleurs de toutes sortes. Il suffit de voir **la diversité des réalisations présentées ici** pour s'en rendre compte!

Arduino permet de faire une introduction à l'automatisme, à l'électronique, à la mécanique et à la programmation à moindre coût. Tout ceci nous amène à penser qu'il a parfaitement sa place sur Edurobot et à l'école!

### Présentation

Le projet Arduino est intéressant à plus d'un titre; à commencer par son origine: l'Italie du Nord. Le magazine en ligne OWN! a publié un bon article sur l'origine et l'histoire d'Arduino; la lecture de cet article est vraiment recommandée, car elle permet d'aborder Arduino en comprenant la philosophie qui a mené au projet actuel.

<http://owni.fr/2011/12/16/arduino-naissance-mythe-bidouille/>

Arduino est donc un projet **Open Hardware**, c'est-à-dire basé sur le principe Open Source. La partie matérielle est sous licence **Creative Commons Attribution Share-Alike** (c'est la raison pour laquelle la section et les documents Arduino d'Edurobot possèdent la même licence). La partie logicielle est sous licences **GPL** et **LGPL**.

Cela signifie que chacun peut construire son propre Arduino, mais que des entreprises peuvent aussi créer et vendre leurs propres modules Arduino. Néanmoins seuls les modules « officiels » portent le nom d'Arduino. Dès lors, toute une série de modules compatibles Arduino est commercialisée sous divers noms: Freeduino, Sanguino, Seeduino, Uduino, Diduino...

Arduino se programme à l'aide d'une application java open source. Le langage de programmation est du **C++**, soit l'un des langages les plus populaires.

### Pourquoi Arduino à l'école?

Première constatation: en général, la programmation et l'électronique ne font pas partie de nos plans d'études. Mais pris ensemble et pas séparément, la partie logicielle et matérielle sont non seulement des outils d'apprentissage, mais aussi de création. L'Arduino prend alors part à un projet créatif bien plus ambitieux et parfaitement compatible avec les plans d'études.

Or, apprendre aux élèves à programmer sur Arduino, c'est leur donner non seulement la clé d'accès aux ordinateurs, mais aussi aux objets qui les entourent.

Les élèves vont acquérir de multiples compétences, par exemple:

- la programmation
- les circuits électriques
- les bases de l'électronique
- l'automatisme

Travailler avec un Arduino en périscolaires et extrascolaires est donc une activité riche, en particulier pour transformer les jeunes en citoyens responsables, créatifs et acteurs du numérique.

Arduino est surtout une plateforme d'expérimentation et de recherche. En cela, elle est idéale pour découvrir le monde et le fonctionnement de bien des appareils.

edurobot.ch



#### Licence d'utilisation

Ce site et son contenu sont à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale 2.5 Suisse](#)

## Présentation Arduino



### Découverte de la carte Arduino



## Sommaire



- Pour quoi faire?
- Pour qui ?
- Qu'est ce que c'est ?
- Comment ça marche ?
- Faut-il des connaissances en électronique ?

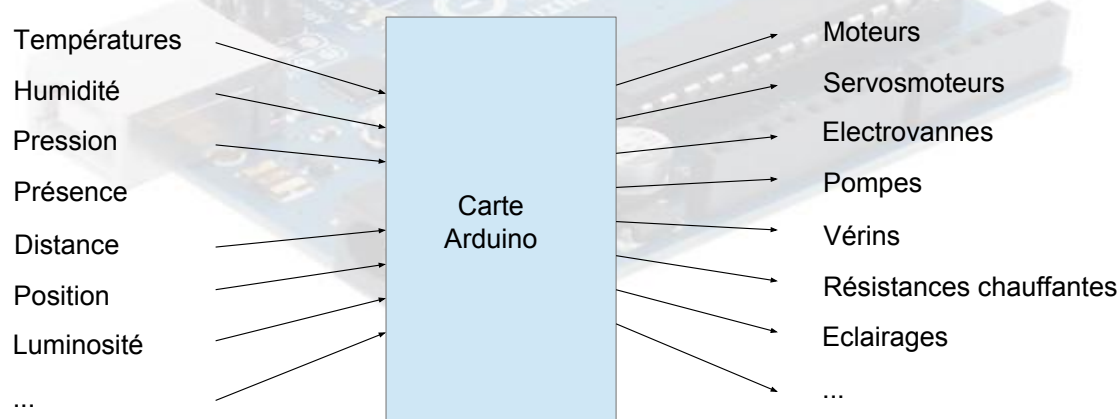


## Pour quoi faire ?



Dans le parcours « robotique » nous utilisons la carte Arduino pour faire interagir le robot avec l'environnement.

Arduino peut interagir avec le monde réel:



## Pour qui?



- Le projet « Arduino » a été initié par un groupe d'enseignants et d'étudiants d'une école de design italienne en 2004 – 2005.
- 
- Les utilisateurs d'Arduino sont :
  - des « bidouilleurs » dont beaucoup ont des connaissances très limitées en électronique ;
  - des artistes qui ont besoin d'animer leurs œuvres ou de créer des interactions avec elles ;
  - des étudiants et des élèves ;
  - et bien sûr, des animateurs, des enfants et des jeunes, dans les différents temps éducatifs et de loisirs

# C'est quoi?



Une plate-forme de développement et de prototypage Open Source.

Le rôle de la carte Arduino est de stocker un programme et de le faire fonctionner.

- **Shields** (cartes d'extension) qui permettent d'ajouter des fonctions qui s'enfichent sur la carte Arduino :
  - Commande pour faire tourner les moteurs, lecteur carte SD...
  - Ethernet, WIFI, GSM (téléphone portable), GPS...
  - Afficheurs LCD...
- **IDE** (Environnement de Développement Intégré) multi OS (système d'exploitation) :
  - édition du programme
  - compilation du programme
  - transfert du programme dans la carte via le port USB

## Diverses Cartes Arduino



C'est celle qu'on utilise dans le parcours



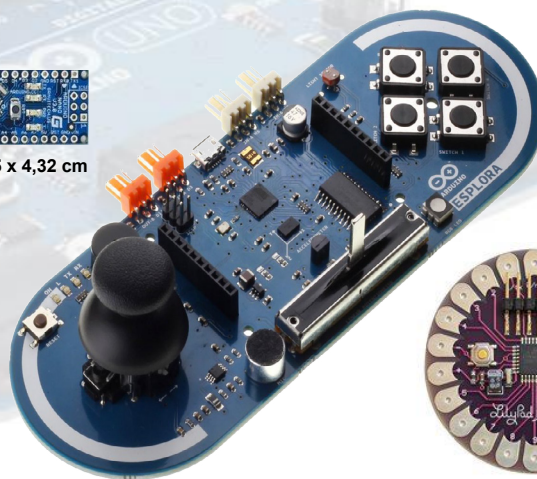
Uno : 5,33 x 6,86 cm



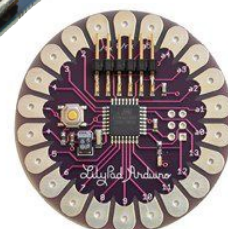
Nano : 1,85 x 4,32 cm



Mega : 5,33 x 10,16 cm

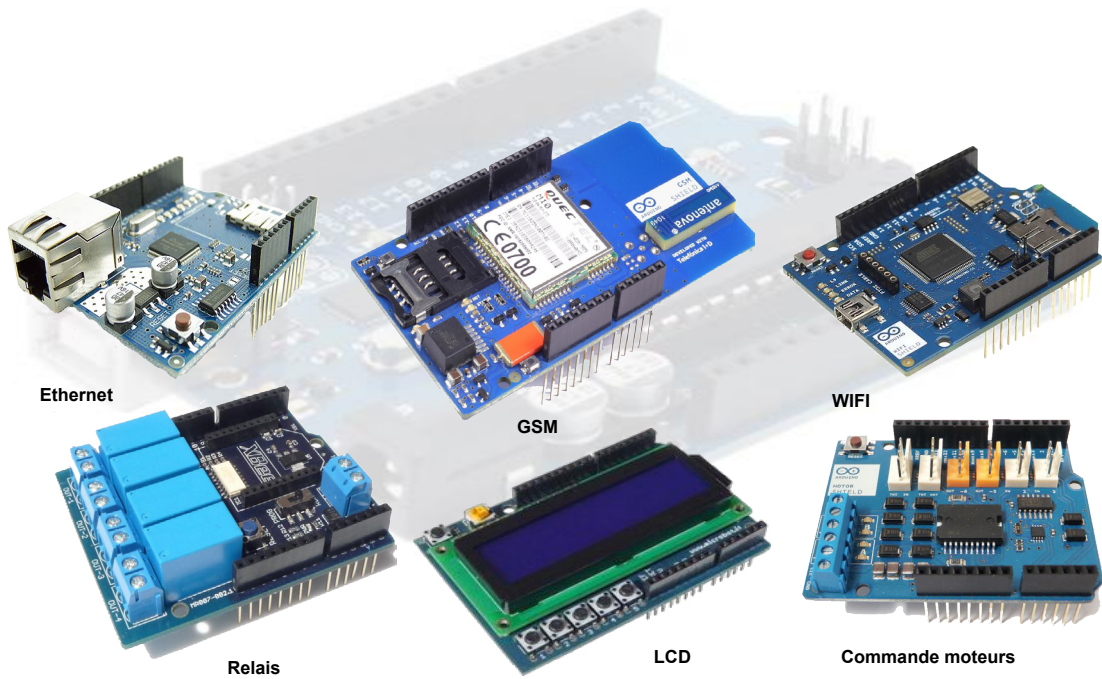


Esplora : 6,1 x 16,51 cm



Lilypad : Ø 5 cm

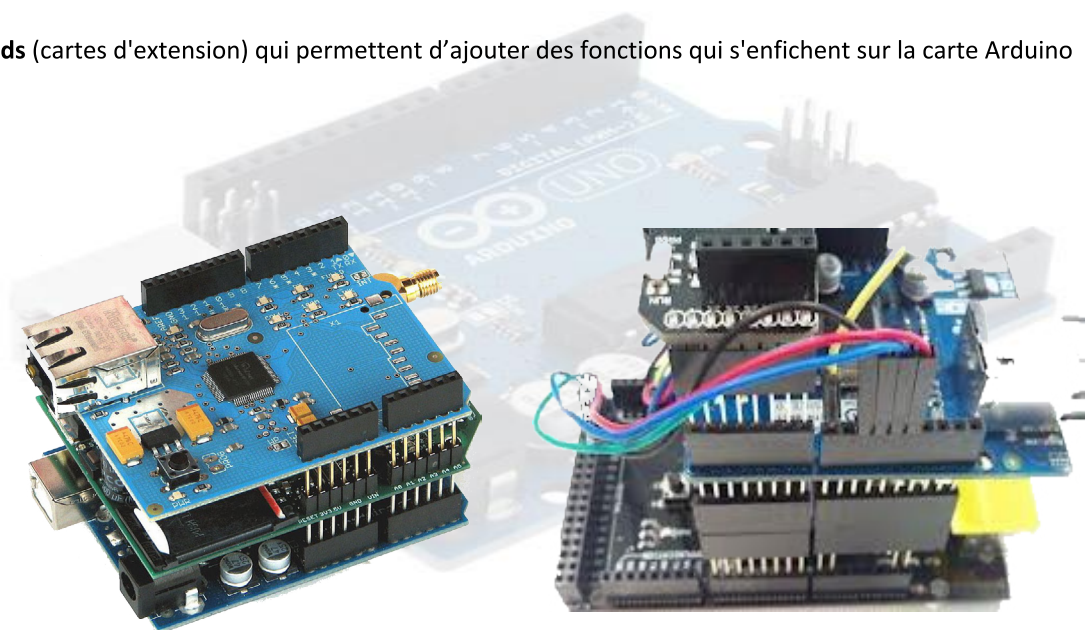
## Divers Shields Arduino



## Carte Arduino + Shields



**Shields** (cartes d'extension) qui permettent d'ajouter des fonctions qui s'enfichent sur la carte Arduino





## C'est quoi?



Un environnement de développement intégré fonctionnant sur divers systèmes d'exploitation (*Windows, Mac OS, Gnu/Linux*) qui permet d'éditer le programme sur un ordinateur et de le transférer via le port USB.

```

Blink | Arduino 1:1.0.5+dfsg2-2
Fichier  Édition  Croquis  Outils  Aide

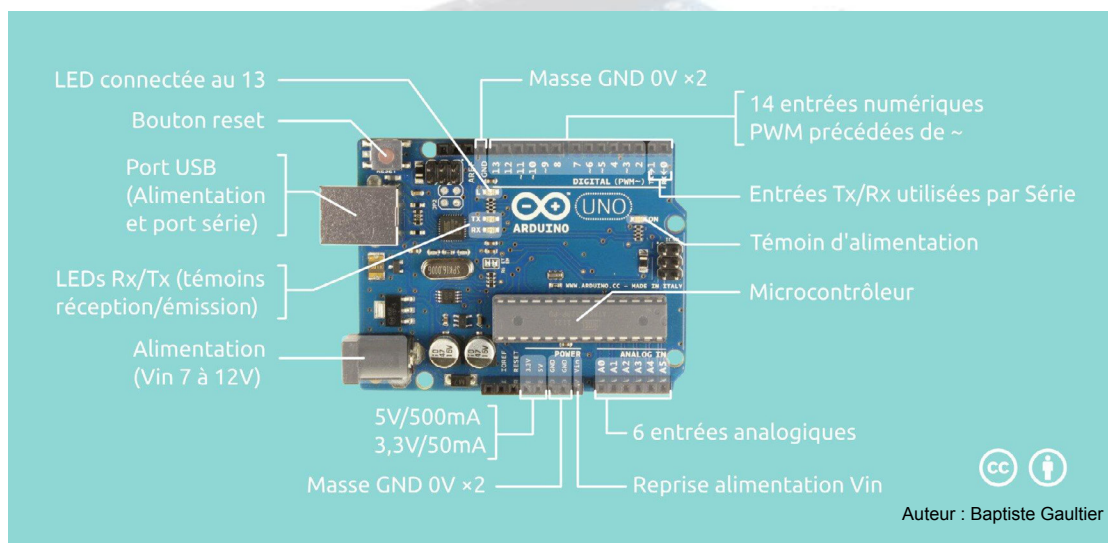
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeat
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the volta
  delay(1000);             // wait for a second
}
  
```

## En quoi cela consiste?



## Quel langage de programmation ?



- 
- Langage proche du C
- 
- Programme structuré :
  - une section « setup » 1 seule exécution après RàZ (remise à zéro);
  - une section « loop » exécutée indéfiniment en boucle.

Setup

Loop

- De très nombreuses bibliothèques logicielles disponibles.

## Programmation avec Arduino



```

Blink | Arduino 1.0.1
File Edit Sketch Tools Help
Blink
int LED = 12 ;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, LOW);
  delay(500);
  digitalWrite(LED, HIGH);
  delay(500);
}

Done Saving.
1 Arduino Duemilanove w/ ATmega328 on COM5
  
```

PC

USB



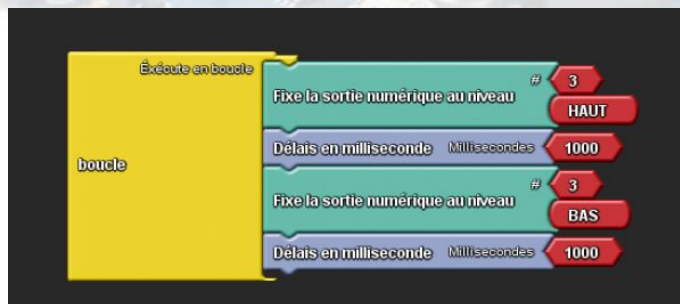
## D'autres outils de programmation



### ➤ Ardublock (programmation en mode graphique)

➤

➤ C'est un outil qui se greffe au logiciel Arduino. Il suffit de créer des blocs et de les paramétrer. Ce logiciel est vraiment un outil de qualité pour démarrer facilement sur Arduino, sans connaissances en programmation.



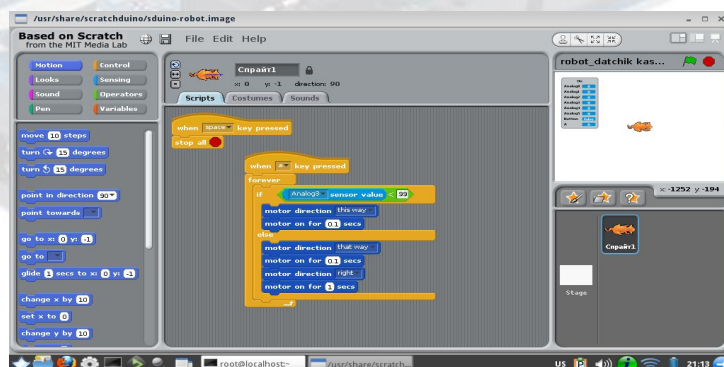
## D'autres outils de programmation



### ➤ Scratch pour Arduino (programmation en mode graphique)

➤

➤ Permet de piloter un Arduino à partir du code SCRATCH et de ce fait rend accessible à tout public la programmation d'un robot à partir d'un environnement aussi ludique, visuel et intuitif que celui de SCRATCH.



## Arduino : faut-il des connaissances en électronique ?



- Pas ou peu si on utilise des cartes et des modules tout faits.
- 
- La communauté francophone est très active sur le forum => entraide, tutoriels, exemples de réalisations...
- 
- Il faut des connaissances en électronique si on veut optimiser ou faire du sur-mesure.



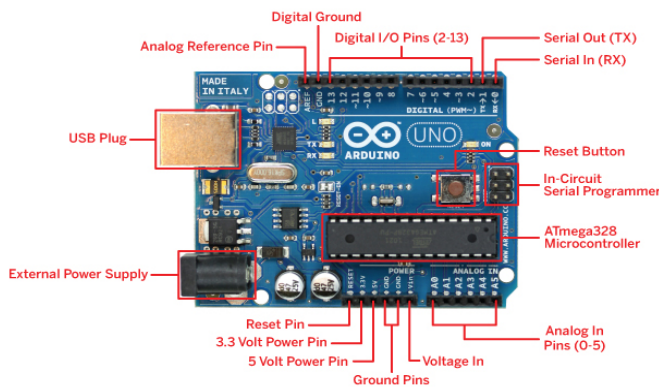
## Vous avez dit Arduino ! UNO

### 1 - La carte Arduino

« Arduino » est une famille de cartes (circuits imprimés) en « open hardware » sur lequel on retrouve un microcontrôleur qui peut être programmé pour réaliser différentes opérations ainsi que de nombreux connecteurs (entrées/sorties) qui permettent d'ajouter des composants à la carte.

Cette carte électronique « Arduino » programmable et son logiciel multiplateforme ont été imaginés et développés par Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti en 2006 afin d'offrir au plus grand nombre la possibilité de créer facilement des systèmes électroniques.

#### Carte Arduino de type UNO



**USB Plug** : Port USB pour discuter avec l'ordinateur.

**External Power Supply** : Entrée pour une alimentation autre que USB (une batterie par exemple)

**Atmega328** : Microcontrôleur de la carte, c'est le cerveau de la carte qui stock le programme et l'exécute.

**Les Pins** : Deux types de pins, analogique et numérique (\*) permettant de brancher tous les éléments externes en entrées/sorties.

**Reset Button** : Permet de réinitialiser (relancer) le programme.

**GND** : Masse (partie conductrice d'un matériel électrique) OU « - »

(\*) Les entrées/sorties sur la carte numérotées de 0 à 13 sont Numériques (Digital en anglais) et celles numérotées de A0 à A5 sont Analogiques.

**Connecteurs numériques** : destinés aux composants ayant 2 positions. Un interrupteur par ex. sera ouvert ou fermé. La carte lira la valeur 0 ou 1. Ou encore une LED sera allumée ou éteinte, il n'y a pas de position milieu.

**Connecteurs analogiques** : destinés à des capteurs comme un variateur. Il sera possible de faire varier le signal de (0) à (1023). Entre les deux toutes les valeurs peuvent être reçues par le microcontrôleur.

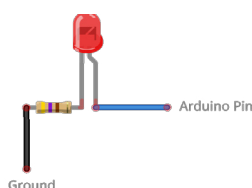
### 2 – Ajout de composants sur la carte Arduino

Chaque composant est prévu pour fonctionner dans des conditions précises, il est donc important de connaître les caractéristiques des composants avant de les ajouter à la carte Arduino.

#### Exemple d'une LED :

Les connecteurs de la carte Arduino délivrent 5V (cf. annexe 16) en sortie, or généralement une LED a besoin d'environ 3V pour un courant de 5mA. Si on branche directement une LED sur la carte, elle ne va pas aimer les 5V qui lui sont fournis et va chauffer jusqu'à cramer si elle reste allumée trop longtemps.

C'est pour cela qu'il faut ajouter une résistance en série pour que la LED ait ses 3V / 5mA, il faut une résistance qui encaisse les 2V de trop.  $R = U / I = 2V / 5mA = 400 \text{ Ohm}$ .



Le montage peut se réaliser sur une Breadboard, plaque additionnelle prévue à cet effet.

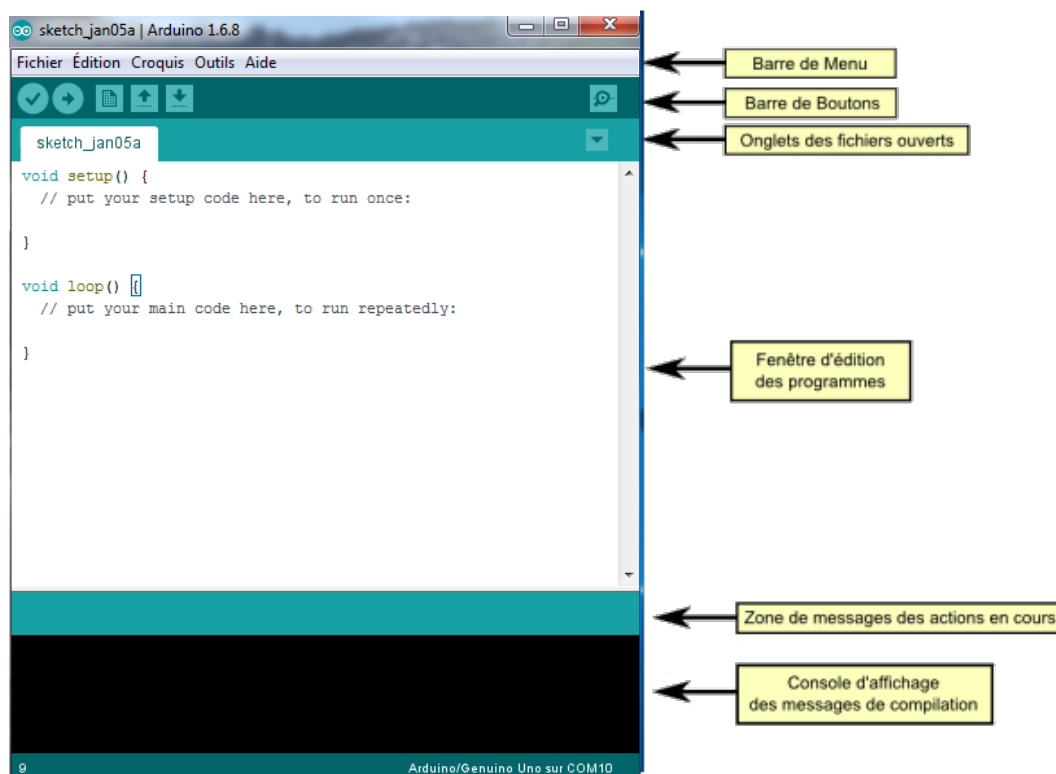
**TEST de connexion d'une LED sans ajouter de résistance** : Connecter la cathode (patte la plus courte) sur un connecteur de masse GND et l'anode (patte la plus longue) sur le connecteur 3.3 Volt Power Pin, puis alimenter la carte.

Puis tester en connectant l'anode sur le connecteur 5 Volt Power Pin  $\Rightarrow$  l'intensité de la LED est plus forte et elle chauffe (la débrancher avant qu'elle ne crame).

**Pour les curieux :** La carte Arduino a un régulateur à faible chute de tension qui assure la conversion entre la tension externe (piles, batterie) et les 5V nécessaires pour la carte. La tension d'alimentation externe recommandée se situe entre 7 volts et 12 volts.

### 3 – L'interface du logiciel Arduino

Après l'installation du logiciel ([lien de téléchargement](#)), le lancer pour rentrer sur l'interface de programmation



Cette interface ressemble à un éditeur de texte simple, il permet d'écrire des programmes en langage C/C++ simplifié, de les tester et de les transférer dans le microcontrôleur de la carte Arduino.

#### Les Boutons du menu du logiciel Arduino



**Ce bouton permet de vérifier le programme.** Le logiciel Arduino va chercher si ce qui est écrit est conforme à ce qui est attendu, il teste et compile le programme. *La compilation est la transformation du langage (C/C++) de programmation en langage machine utilisé par la carte.*



**Ce bouton permet de téléverser le programme.** Le logiciel va transférer le programme compilé dans la mémoire du microcontrôleur de l'Arduino qui le gardera en mémoire. Il l'exécutera tant qu'il sera alimenté en électricité. Il sera donc autonome et ne dépendra plus de l'ordinateur.



**Ce bouton permet d'ouvrir une nouvelle fenêtre.** L'interface de programmation Arduino permet de travailler sur plusieurs programmes (fenêtres) en simultané.



**Ce bouton permet d'ouvrir un programme enregistré.** De nombreux exemples de programmes sont proposés par Arduino et permettent une prise en main rapide ainsi que de retrouver les syntaxes correctes des mots clés les plus courants du langage de programmation.



**Ce bouton permet d'enregistrer votre programme.** Les fichiers sont stockés dans des dossiers créés automatiquement, aussi appelés "Carnet de croquis".

## 4 – La programmation

### 4.1 - La structure du programme

```

1 // (déclarations variables...)
2
3 void setup()
4 // Fonction setup d'initialisation
5 {
6 ...
7 }
8 void loop()
9 /* Fonction loop infinie
10 programme principal */
11 {
12 ...
13 }
14 mafonction()
15 /* Création d'une fonction qui
16 pourra être appelée dans le
17 programme principal */
18 {
19 ...
20 }
21
22

```

Cf. Chapitre 4.2 : Les constantes et les variables

La fonction setup() est appelée au démarrage du programme. Elle est utilisée pour initialiser les variables, le sens des broches, les bibliothèques utilisées... Elle n'est exécutée qu'une seule fois, après chaque mise sous tension ou réinitialisation de la carte Arduino.

La fonction loop() est le programme principal.

Les instructions contenues dans cette fonction sont exécutées en boucle tant que la carte n'est ni éteinte ni réinitialisée.

**Ce sera donc le programme du robot !**

Il est possible de créer ses propres fonctions(), celles-ci pourront être appelées depuis le programme principal. La création de fonctions à l'extérieur du programme principal permet :

- d'alléger la fonction loop qui sera ainsi plus lisible ;
- d'appeler plusieurs fois une fonction sans avoir à la réécrire.

**Remarques :**

- Une fonction est **une succession d'instructions** qui débutent TOUJOURS par « { » et se terminent TOUJOURS par « } »
- **Attention la casse** des lettres est prise en compte en C/C++

Pour commenter son programme il suffit de saisir le code // tout ce qui suivra sur la ligne sera lisible mais transparent pour le programme. Pour un bloc de plusieurs lignes à la suite on utilise : début /\* et fin \*/

### 4.2 - Les constantes et les variables

On déclare les constantes et les variables avant le setup (), on peut les apparenter à des « boîtes » dans lesquelles on range des informations : textes, nombres,... que l'on pourra utiliser à tout moment dans le programme et sur lesquelles on pourra effectuer des opérations. Ces espaces mémoire sont caractérisés par des noms qui permettent d'y accéder facilement.

Comme son nom l'indique la valeur d'une **constante** restera identique au cours du programme tandis que celle d'une **variable** pourra changer au cours du programme.

**Remarque :** Chaque variable et/ou constante que l'on utilise va prendre de la place dans la mémoire de l'Arduino. Il est donc important de bien choisir son type selon son utilisation. Les principaux types sont :

**boolean** maVariable = true ;

« **boolean** » peut prendre deux états TRUE (vrai) (1) ou FALSE (faux) (0) ⇔ Espace utilisé 1 bit

**int** maVariable = 15 ;

« **int** » est un nombre entier de - 32 768 à 32 767 ⇔ Espace utilisé 16 bits (soit 2 octets)

**float** maVariable = 1.23 ;

« **float** » peut être un nombre à virgule de -3.4028235E+38 à 3.4028235E+38 ⇔ Espace utilisé 32 bits

**string** maVariable = « Bonjour » ;

« **string** » peut contenir du texte de 0 à 2 milliards de car. ⇔ Espace utilisé 10 octets + longueur de chaîne

**Exemples de déclarations :**


**const int** pinLed = 13; // On déclare ici une **constante** de type **int** qui aura pour nom **pinLed** et pour valeur **13**.

Je ferais mon montage en mettant ma LED sur le connecteur 13, ainsi à chaque fois que j'aurais besoin du numéro de la pin de la LED je pourrais aller le chercher dans la « boîte » pinLed et récupérer le nombre 13. C'est plus long d'écrire pinLed que 13 cependant, lorsque mon montage changera, par exemple si je branche ma LED sur la pin 10, je n'aurai qu'à changer le nombre à l'intérieur de la boîte pinLed sans changer mon programme.

**float** maVariable = 3.14159265358979323846264338327950288419; // On déclare ici une **variable** de type **float** qui aura pour nom **maVariable** et pour valeur **Pi**.

#### 4.3 – Exemple : « programmation pour faire clignoter une LED »

Une des forces d'apprentissage d'Arduino repose dans ces nombreux exemples intégrés au logiciel qui permettent de se familiariser avec le code et peuvent s'utiliser comme des briques entre-elles. Tous ces exemples se retrouvent dans fichier/exemples.

Lancer le logiciel Arduino puis ouvrir un fichier  répertoire [01.Basics] et sélectionner le croquis mis à disposition en exemple : **Blink**

(On peut également ouvrir le fichier **Blink** en passant par le menu **Fichier** ⇒ **Exemples** ⇒ **01.Basics** ⇒ **Blink**)

##### Analyse du code :

« **void setup ()** » Initialisation de l'Arduino, tout ce qui se situe entre les { } ne sera effectué qu'au démarrage.

« **void loop ()** » Boucle infinie, tout ce qui se trouve entre les { } va être répété indéfiniment.

##### Décomposition de setup() :

**pinMode** (13, **OUTPUT**) ; // C'est l'information nécessaire à l'Arduino pour initialiser la pin 13. On lui indique que le « **mode** » de la « **pin** » numéro 13 est une **sortie (OUTPUT)**.

Les pins peuvent être des entrées ou des sorties, on doit donc OBLIGATOIREMENT définir dans le setup le type de tous les connecteurs utilisés dans le montage. Si le connecteur était une entrée : **pinMode** (13, **INPUT**) ;

**On connecte donc la LED sur le PIN 13 : La cathode sur un connecteur GND et l'anode sur le connecteur 13.**

##### Décomposition de loop() :

```
{ // DEBUT
```

```
digitalWrite (13, HIGH) ; // 1ère instruction (action) à effectuer. On écrit sur la pin 13 une valeur haute ⇒ on envoie du courant sur la sortie 13 ⇒ la LED s'allume.
```

```
delay(1000) ; // 2ème instruction (action) à effectuer. On attend 1000 millisecondes soit 1 sec.
```

```
digitalWrite (13, LOW) ; // 3ème instruction (action) à effectuer. On écrit sur la pin 13 une valeur basse ⇒ on envoie plus de courant sur la sortie 13 ⇒ la LED s'éteint.
```

```
delay(1000) ; // 4ème instruction (action) à effectuer. On attend 1000 millisecondes soit 1 sec.
```

```
} // FIN – La fonction loop() est exécutée en boucle on retourne donc au début
```

##### Remarques :

La PIN 13 est alimentée en 5V lorsqu'on fait un **digitalWrite HIGH** et à 0V lorsqu'on fait un **digitalWrite LOW**.

Noter le caractère « ; » à la fin des lignes. C'est une information pour indiquer la fin d'une action. Ce caractère est INDISPENSABLE à chaque fin d'action pour que le programme fonctionne correctement.

**PAR SIMPLIFICATION DE MONTAGE NOUS N'UTILISONS PAS DE RESISTANCE EN SERIE AVEC LA LED MAIS ATTENTION CE N'EST BON NI POUR LA LED NI POUR LES COMPOSANTS DE LA CARTE ! ET UN ELECTRONICIEEN SAUTERAIT AU PLAFOND EN VOYANT CE MONTAGE. A N'UTILISER DONC QUE DANS LE CAS DE CE TUTO ET SANS EN ABUSER.**

##### Tester le programme en le téléversant dans le microcontrôleur de la carte Arduino :

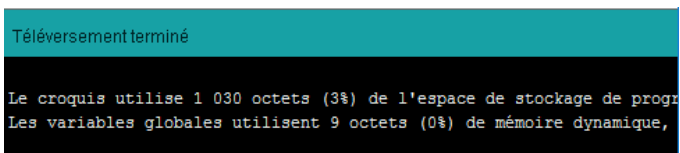
- Connecter la carte à l'ordinateur via un câble USB, choisir le modèle de la carte dans le menu Outils et vérifier la configuration du port COM de la prise USB. (Normalement le port se sélectionne tout seul).

Les paramètres du Port COM (gestionnaire de périphérique de votre PC) doit-être à 115 200 Bits par seconde

- Cliquer sur la petite flèche en haut à gauche « Téléverser ».

Le logiciel va compiler le croquis puis le téléverser vers la carte.

Si tout se passe bien, la LED doit clignoter à intervalle régulier d'une seconde,



**Exercice :** Modifier le programme en y ajoutant une constante pour la Pin sur laquelle est connectée la LED, modifier « **void setup ()** » et « **void loop ()** » en conséquence. Changer les paramètres des instructions afin que la LED s'allume 0.5 seconde puis s'éteigne 2 secondes... (en boucle).

## 5 – La robotique avec une carte Arduino

### 5.1 – Rappel

Les robots sont constitués de capteurs, d'un contrôleur et d'actionneurs. Le contrôleur reçoit des informations des capteurs et envoie des instructions aux actionneurs. **Capteurs ⇌ Contrôleur ⇌ Actionneurs**

- Les capteurs se brancheront sur les Pins de la carte, ils seront configurés en entrées **INPUT**
- Le microcontrôleur de la carte Arduino sera le contrôleur du robot.
- Les actionneurs se brancheront sur les Pins de la carte, ils seront configurés en sorties **OUTPUT**

Pour créer un robot avec une carte Arduino il suffit donc de choisir les capteurs et les actionneurs adaptés au robot souhaité et d'écrire le programme déterminant les actions que réaliseront les actionneurs du robot en fonction des informations reçus par ses capteurs.

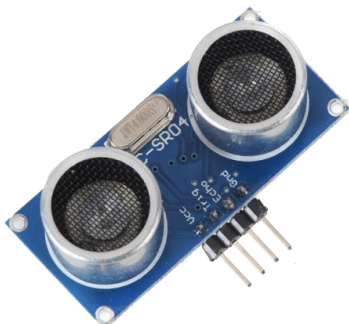
Si l'on regarde l'exemple de la LED **pinMode** (13, **OUTPUT**); elle est connectée en sortie car elle reçoit des instructions du contrôleur **digitalWrite** (13, **HIGH**); **digitalWrite** (13, **LOW**); **C'est un actionneur**

### 5.2 – Les Capteurs

La library (ou bibliothèque de fonctions) associée à un capteur est un ensemble de fonctions qui ont été développées spécifiquement pour ce capteur. Elles sont le plus souvent disponibles en Open Source sur Internet. Pour utiliser une bibliothèque il faut la télécharger et l'installer dans le programme Arduino.

Il sera nécessaire d'inclure cette bibliothèque à votre programme afin que ces fonctions soient utilisables dans le bloc **void loop()** de celui-ci (de la même manière que les fonctions C++ de base). **Pour les curieux** : Il suffira alors de connaître les déclarations nécessaires à l'utilisation du capteur et les paramètres attendus par les fonctions développées dans la bibliothèque pour utiliser facilement un capteur sur votre robot.

#### Capteur de distance à ultrason



Connecteurs :

**Vcc** : Alimentation 5V

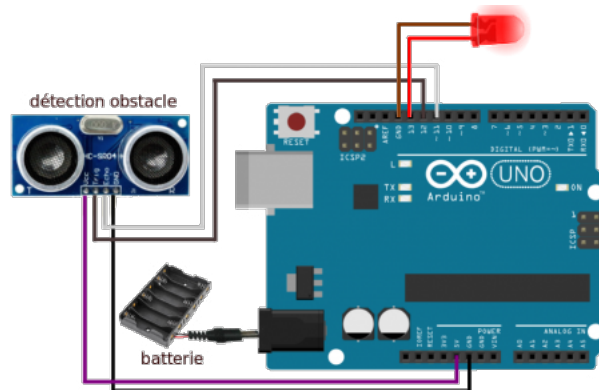
**Trig** : (input) connecté sur la Pin 12

**Echo** : (output) connecté sur le Pin 11

**Gnd** : Masse

Caractéristiques techniques du capteur de distance :

- Alimentation : 5v – Consommation en utilisation : 15 mA.
- Gamme de distance : 2 cm à 5 m et résolution : 0.3 cm.



### 5.3 - Fonctionnement du capteur de distance

Il envoie une impulsion niveau haut (5v) pendant au moins 10  $\mu$ s sur la broche '**Trig Input**' qui déclenche la mesure. En sortie '**Echo Output**' fournit une impulsion de 5v dont la durée est proportionnelle à la distance de l'objet. La distance en cm est calculée en utilisant la formule : **distance = (durée de l'impulsion en  $\mu$ s) / 58**.

**Téléchargement** la bibliothèque newping pour le capteur de distance à ultrason : [Lien de téléchargement](#)

**Installation** : Menu Croquis ⇌ Inclure une bibliothèque ⇌ Ajouter la bibliothèque ZIP ⇌ Retrouver le répertoire de téléchargement de la bibliothèque newping.zip sur votre ordinateur et l'ouvrir.

#### **Pour les curieux** : Structure d'une bibliothèque

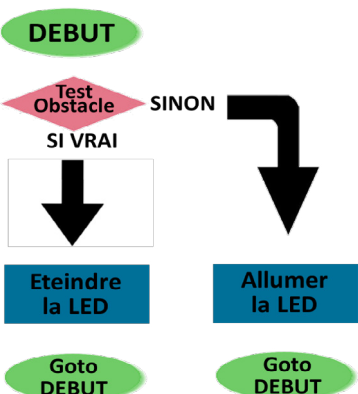
Les bibliothèques Arduino sont composées d'au moins deux fichiers, un fichier d'en tête finissant par .h et un fichier source finissant par .cpp. Le fichier d'en tête contient les définitions des fonctions disponibles et le fichier source contient l'implémentation du code des fonctions qui ont été définies dans le fichier d'en tête.

### 5.4 – Programmation : OBJECTIF « Allumer la LED quand un obstacle se trouve devant le capteur »

#### Déclaration et Initialisation setup()

<pre> 1 // définitions et déclarations 2 #include &lt;NewPing.h&gt; 3 #define trigPin 12 4 #define echoPin 11 5 NewPing distanceCM (trigPin, 6 echoPin); 7 int maximumDistance = 50; 8 int minimumDistance = 0; 9 boolean PasObstacle = false; 10 const int distanceObstacle = 15; 11 int attenteCapteur = 100; 12 // the setup function 13 Serial.begin(115200); 14 pinMode(13, OUTPUT); 15 digitalWrite(13, LOW); 16 delay(300); </pre>	<p><b>Déclarations nécessaires pour l'utilisation du capteur</b></p> <p>Déclaration de la library NewPing.h du capteur la broche (pin) Trigger est branchée sur la broche 12 de l'Arduino la broche (pin) Echo est branchée sur la broche 11 de l'Arduino on initialise la fonction distanceCM</p> <p>distance maximale acceptée (en cm) distance minimale acceptée (en cm) valeur à false (ou 0) si pas d'obstacle détecté sinon true (ou 1) on définit la distance de détection d'un obstacle, ici 15 cm valeur en ms de l'attente de détection du capteur</p> <p><b>Initialisations nécessaires pour le LED</b></p> <p>On initialise la vitesse du moniteur série à 115 200 Bauds On initialise la Pin 13 (pour la LED) en Sortie On éteint la LED On attend 0.3 sec avant de démarrer le programme</p>
---	--

#### Programme principal loop()

Représentation Graphique « organigramme »	Programme en C++	Description
	<pre> { int distance = mesureDistance(); delay(attenteCapteur); if (PasObstacle == false) { digitalWrite(13, LOW); } else { digitalWrite(13, HIGH); Serial.print(distanceCM.ping_cm()); delay(500); } } </pre>	<p>Début du programme Distance de l'objet, on met sa valeur dans variable distance. TEST : Présence d'un obstacle ? <i>début</i> Si pas d'obstacle on éteint la LED <i>fin</i> Sinon <i>début</i> On allume la LED Affichage : distance de l'obstacle (1) On attend 0.5 sec <i>fin</i> Fin du programme ⇌ début</p>

#### Fonction spécifique : Création de la fonction qui va renseigner la variable PasObstacle

<pre> 1 unsigned int mesureDistance() 2 { 3 int cm = distanceCM.ping_cm(); 4 if (cm &gt; distanceObstacle    cm 5 &lt;= minimumDistance) 6 { 7 PasObstacle = vrai; 8 } 9 else 10 { 11 PasObstacle = true; 12 } 13 return cm; 14 } </pre>	<p>Déclaration de la variable mesureDistance DEBUT On déclare et on renseigne la variable cm. (1) Si la distance à l'obstacle est sup. à la valeur renseignée dans distanceObstacle ou inf. à la valeur renseignée dans minimumDistance <i>début</i> SI VRAI on met vrai dans la variable PasObstacle <i>fin</i> SINON <i>début</i> On met true dans la variable PasObstacle <i>fin</i> La fonction retourne la distance de l'obstacle FIN</p>
--	--

(1) **Serial.print** permet à tout moment d'afficher des informations dans le moniteur série du logiciel Arduino.

(2) `distanceCM.ping_cm()` est une fonction de la bibliothèque `newping.h` qui calcule la distance de l'obstacle.



**Code (Arduino) « Pour allumer une LED quand un obstacle se trouve devant le capteur entre 10 et 20cm »**

```

/* Programme permettant d'allumer une LED (branchée sur la Pin 13 de la carte Arduino) lorsque le capteur de
distance détecte un objet à une distance comprise entre 10 et 20 cm. */
#include <NewPing.h>
// définitions et déclarations des variables
#define trigPin 12
#define echoPin 11
NewPing distanceCM (trigPin, echoPin);
int maximumDistance = 50;
int minimumDistance = 10;
boolean PasObstacle = false;
const int distanceObstacle = 20;
int attenteCapteur = 100;

// the setup function
void setup()
{
  Serial.begin(115200);           // On initialise la vitesse du moniteur série à 115 200 bauds
  pinMode(13, OUTPUT);          // On initialise la pin 13 sortie (OUTPUT)
  digitalWrite(13, LOW);        // On éteint la LED
  delay(300);                    // On attend 0.3 sec
}

// the loop function
void loop()
{
  int distance = mesureDistance(); // on stock dans la variable distance la valeur du capteur en cm
  delay(attenteCapteur);           // attente en ms entre chaque mesure du capteur
  if (PasObstacle == vrai)        // S'il n'y a pas d'obstacle
  {
    digitalWrite(13, LOW);        // On éteint la LED
  }
  else                             // Sinon, un obstacle est détecté
  {
    digitalWrite(13, HIGH);       // On allume la LED
    Serial.print("Distance: ");
    Serial.print(distanceCM.ping_cm()); // On affiche la distance de l'obstacle
    Serial.println("cm");
    delay(500);                   // pendant 500 ms
  }
}

// the personal fonction
unsigned int mesureDistance()     // Déclaration de la variable mesureDistance
{
  int cm = distanceCM.ping_cm(); // déclaration de la variable locale (cm) à qui on assigne la distance
  if (cm > distanceObstacle || cm <= minimumDistance) // on définit la plage de détection du capteur
  {
    PasObstacle = true;          // on renvoie true si pas d'obstacle dans la plage définie dans le test
  }
  else
  {
    PasObstacle = false;        // sinon on renvoie false (si il y a un obstacle dans la plage définie)
  }
  return cm;                     // on retourne la distance de l'obstacle à la fonction
}

```

## 6 – ROSA (Robot Open Source Arduino)

### 6.1 – Introduction

Conçu à l'occasion du projet [D-Clics numériques](#), ce robot se veut une machine simple à fabriquer et évolutive. Il s'agit d'un robot éviteur d'obstacles utilisant deux moteurs et un capteur de distance, pour découvrir les bases de la robotique notamment à travers :

- l'assemblage du châssis
- l'électronique (à travers le câblage des différents éléments)
- la programmation de la carte Arduino Uno

L'objectif du cahier des charges est de construire un robot à bas coût et évolutif.

- reproductibilité : toutes les pièces utilisées doivent être faciles à trouver
- low-cost : le coût doit être acceptable pour une utilisation en nombre, de l'ordre de 45€
- polyvalent : la carte Arduino peut être utilisée pour réaliser d'autres projets
- programmation libre et open-source : code open-source et GPL
- simplicité de réalisation
- Il doit être facilement réalisable sans avoir de compétences particulières en mécanique ou bricolage

Ce robot utilise un capteur ultrason qui mesure une distance. L'information est traitée par la carte Arduino et permet au robot lorsqu'il rencontre un obstacle frontal de s'arrêter quelques instants puis reculer. Ensuite il pivote à droite ou à gauche avant de redémarrer jusqu'à l'obstacle suivant.

Il est bien entendu possible de changer la programmation afin de lui attribuer d'autres comportements.

### 6.2 – Éléments du robot



- Le châssis  
2 plaques en acrylique de 3mm découpées au laser
- 2 roues complètes + 1 roue à bille
- Pièces de jonction  
+ 1 ensemble de quincaillerie, vis et écrous
- 1 capteur ultrason de distance avec son support
- 1 contrôleur (carte Arduino)
- 2 moteurs DC 6V double axe
- 1 ensemble de fil de câblage  
+ 1 support de 6 piles AA

Le Robot est donc constitué d'un **capteur** (capteur de distance), d'un **contrôleur** (carte Arduino) et de deux **actionneurs** (moteurs).

Un **moteur électrique** permet de convertir l'énergie électrique (par ses composants qui transforment la puissance d'un champ magnétique) en énergie mécanique.

**Caractéristiques** : - Plus le voltage est élevé et plus le moteur tourne vite

- Le champ magnétique a 2 pôles, si on inverse ces pôles le moteur tournera en sens inverse

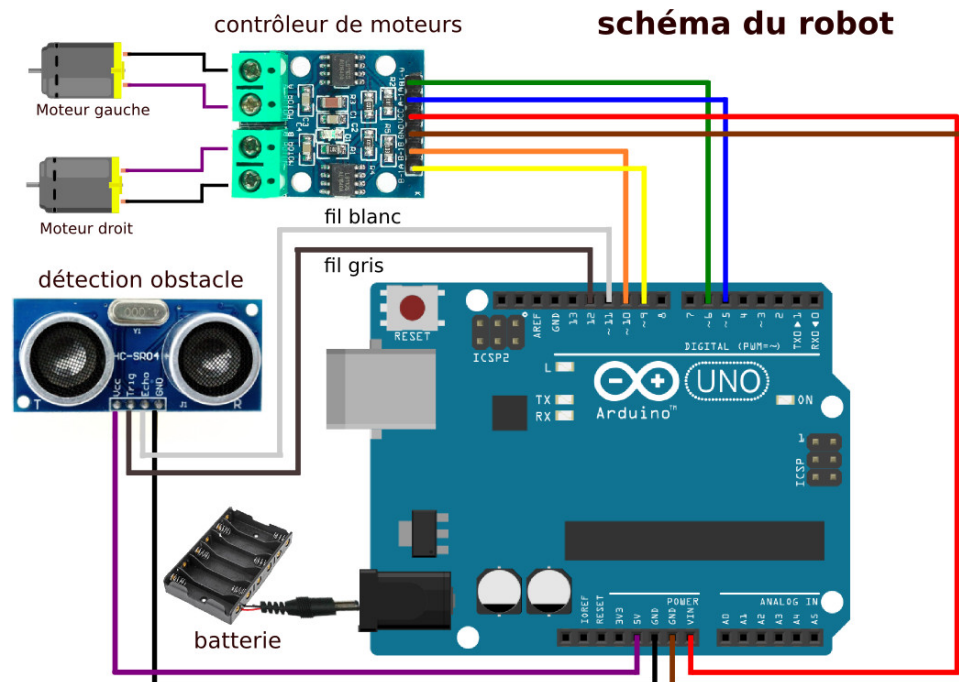
*Pour tester les moteurs il suffit de relier les fils à chaque extrémité d'une pile, en inversant les fils le moteur tournera en sens inverse.*

⇒ Si on envoie le courant sur un des fils le moteur tourne dans un sens et si on l'envoie sur l'autre fil il tourne en sens inverse.

### 6.3 – Montage du châssis

Cf. fiche de montage

### 6.4 – Câblages sur la carte Arduino



Connexions	Contrôleur moteur	Carte Arduino	Remarques
Câble marron	GND = Masse	Pin GND	Correspond à la masse du contrôleur moteur
Câble rouge	VCC = Alimentation 5V	Pin VIN	Alimentation 5V du contrôleur moteur
Câble orange	Moteur droit BIB	Pin 10	Pour faire tourner le moteur droit ↻ en avant
Câble jaune	Moteur droit BIA	Pin 9	↻ en arrière
Câble vert	Moteur gauche AIB	Pin 6	Pour faire tourner le moteur gauche ↻ en avant
Câble bleu	Moteur gauche AIA	Pin 5	↻ en arrière

### 6.5 – Programmation

La programmation de « ROSA » s'apparente à la programmation précédemment effectuée avec le capteur de distance et la LED (La LED est remplacée par des moteurs !)

En fonction de l'information reçue par le capteur on enverra des instructions aux moteurs.

#### Déclarations

```

1 // définitions et déclarations
2 #include <NewPing.h>
3   Reprendre les déclarations
4   utilisées pour le capteur
5
6 #define vitesse_MG 150
7 #define vitesse_MD 150
8
9 int AIA = 5;
10 int AIB = 6;
11
12 int BIA = 9;
13 int BIB = 10;
14
```

#### Déclarations

Déclaration de la library NewPing.h du capteur

On reprend l'ensemble des déclarations nécessaires au capteur

#### Déclarations liées aux moteurs

vitesse du moteur gauche (0 à 255)

vitesse du moteur droit (0 à 255)

#### Moteur A (moteur Gauche)

Broche A connectée à la Pin 5

Broche B connectée à la Pin 6

#### Moteur B (moteur Droit)

Broche A connectée à la Pin 9

Broche B connectée à la Pin 10

**Initialisations Void setup()**

<pre> 15 // the setup function 16 pinMode(AIA, OUTPUT); 17 pinMode(AIB, OUTPUT); 18 pinMode(BIA, OUTPUT); 19 pinMode(BIB, OUTPUT); 20 stopRobot(); 21 delay(300); </pre>	<p><b>Initialisations nécessaires pour les moteurs</b></p> <p>On initialise la Pin 5 en Sortie</p> <p>On initialise la Pin 6 en Sortie</p> <p>On initialise la Pin 9 en Sortie</p> <p>On initialise la Pin 10 en Sortie</p> <p>On stop le robot avec la fonction stopRobot() (Cf. Rosa Fonctions)</p> <p>On attend 300 millisecondes avant de démarrer le loop().</p>
--	---

**Programme principal loop()**

Cahier des charges du fonctionnement du Robot	Programme en C++	Description
<p>Le Robot ROSA avance tant qu'il ne rencontre pas d'obstacle.</p> <p>Si le capteur de distance détecte un obstacle à X cm (1) :</p> <ul style="list-style-type: none"> <li>- Le robot stoppe</li> <li>- Il fait une pause de 0.3 sec</li> <li>- Le robot recule pendant 0.2 sec</li> <li>- Le robot tourne à droite pendant 0.2 sec.</li> </ul> <p>Puis il redémarre...</p> <p>(1) X est défini dans les déclarations const int distanceObstacle = X;</p>	<pre> { int distance = mesureDistance(); delay(attenteCapteur); if (PasObstacle == true) { avanceRobot(); } else { stopRobot(); delay(300); reculeRobot(); delay(200); tourneDroite(); delay(200); } } </pre>	<p>Début du programme</p> <p>Distance de l'objet, on met sa valeur dans variable distance.</p> <p>TEST : Présence d'un obstacle ?</p> <p><i>début</i></p> <p>Si pas d'obstacle on avance</p> <p><i>fin</i></p> <p>Sinon</p> <p><i>début</i></p> <p>On arrête le robot</p> <p>On attend 0.3 sec</p> <p>Le robot recule</p> <p>On attend 0.2 sec (Pendant 0.2 sec)</p> <p>Le robot tourne à droite</p> <p>On attend 0.2 sec (Pendant 0.2 sec)</p> <p><i>fin</i></p> <p>Fin du programme ⇐ début</p>

**Fonction spécifique : Création des fonctions qui agissent sur les moteurs**

<pre> 1 void avanceRobot() 2 { 3   analogWrite(AIA, vitesse_MG); 4   analogWrite(AIB, LOW); 5   analogWrite(BIA, vitesse_MD); 6   analogWrite(BIB, LOW); 7 } 8 void stopRobot() 9 { 10  analogWrite(AIA, LOW); 11  analogWrite(AIB, LOW); 12  analogWrite(BIA, LOW); 13  analogWrite(BIB, LOW); 14 } 15 void tourneDroite() 16 { 17   analogWrite(AIA, vitesse_MG); 18   analogWrite(AIB, LOW); 19   analogWrite(BIA, LOW); 20   analogWrite(BIB, LOW); 21 } 22 ... </pre>	<p>Pour créer les fonctions spécifiques du robot il suffit de déterminer pour chaque action à quelle broche on doit envoyer du courant.</p> <p><b>void avanceRobot()</b></p> <ul style="list-style-type: none"> <li>- on envoie du courant sur les broches A des deux moteurs</li> </ul> <p><b>void reculeRobot()</b></p> <ul style="list-style-type: none"> <li>- on envoie du courant sur les broches B des deux moteurs</li> </ul> <p><b>void stopRobot()</b></p> <ul style="list-style-type: none"> <li>- on n'envoie plus de courant</li> </ul> <p><b>void tourneDroite()</b></p> <ul style="list-style-type: none"> <li>- on envoie du courant sur les broches A du moteur Gauche</li> </ul> <p><b>void tourneGauche()</b></p> <ul style="list-style-type: none"> <li>- on envoie du courant sur les broches A du moteur Droit</li> </ul> <p><b>void robotSurPlaceGauche()</b></p> <ul style="list-style-type: none"> <li>- on envoie du courant sur les broches B du moteur Gauche</li> <li>- on envoie du courant sur les broches A du moteur Droit</li> </ul> <p><b>void robotSurPlaceDroite()</b></p> <ul style="list-style-type: none"> <li>- on envoie du courant sur les broches A du moteur Gauche</li> <li>- on envoie du courant sur les broches B du moteur Droit</li> </ul>
--	---

## Code (Arduino) « du Robot ROSA »

```

/* Programmation : Si le robot rencontre un obstacle il s'arrête, recule puis tourne à droite et repart en avant */
#include <NewPing.h>

// définitions et déclarations des variables
// déclarations pour le capteur
#define trigPin 12
#define echoPin 11
NewPing distanceCM (trigPin, echoPin);
int maximumDistance = 50;
int minimumDistance = 10;
boolean PasObstacle = true;
const int distanceObstacle = 20;
int attenteCapteur = 100;
// déclaration pour les moteurs
#define vitesse_MG 150 // vitesse du moteur gauche
#define vitesse_MD 150 // vitesse du moteur droit
// Les moteurs ne tournant pas exactement à la même vitesse, on peut jouer sur les valeurs (Maximum 255 = 5V)
int AIA = 5; // Broche A du moteur gauche connecté à la Pin 5 de l'Arduino
int AIB = 6; // Broche B du moteur gauche connecté à la Pin 6 de l'Arduino
int BIA = 9; // Broche A du moteur droit connecté à la Pin 9 de l'Arduino
int BIB = 10; // Broche B du moteur droit connecté à la Pin 10 de l'Arduino

// the setup function
void setup()
{
  pinMode(AIA, OUTPUT); // On initialise la pin 5 en sortie (OUTPUT)
  pinMode(AIB, OUTPUT); // On initialise la pin 6 en sortie (OUTPUT)
  pinMode(BIA, OUTPUT); // On initialise la pin 9 en sortie (OUTPUT)
  pinMode(BIB, OUTPUT); // On initialise la pin 10 en sortie (OUTPUT)
  stopRobot(); // On lance la fonction stopRobot() qui stoppe le robot
  delay(300); // On attend 0.3 sec
}

// the loop function
void loop()
{
  // DEBUT
  int distance = mesureDistance(); // on stock dans la variable distance la valeur du capteur en cm
  delay(attenteCapteur); // attente en ms entre chaque mesure du capteur
  if (PasObstacle == true) // S'il n'y a pas d'obstacle
  {
    avanceRobot(); // On lance la fonction avanceRobot()
  }
  else // Sinon, un obstacle est détecté
  {
    stopRobot(); // On lance la fonction stopRobot()
    delay(300); // On attend 0.3 sec
    reculeRobot(); // On lance la fonction reculeRobot()
    delay(200); // On attend 0.2 sec
    tourneDroite(); // On lance la fonction tourneDroite()
    delay(200); // On attend 0.2 sec
  }
} // FIN on retourne au DEBUT

/* Le Loop peut être modifié facilement avec les enfants en utilisant toutes les fonctions prédéfinies : avanceRobot()
reculeRobot() tourneDroite() tourneGauche() robotSurPlaceGauche() robotSurPlaceDroite() stopRobot() */

```

```

// the personal fonction // Fonctions de mouvements des moteurs
void avanceRobot() { // Fonction : Le robot avance
  analogWrite(AIA, vitesse_MG);
  analogWrite(AIB, LOW);
  analogWrite(BIA, vitesse_MD);
  analogWrite(BIB, LOW);
}
void reculeRobot() { // Fonction : Le robot recule
  analogWrite(AIA, LOW);
  analogWrite(AIB, vitesse_MG);
  analogWrite(BIA, LOW);
  analogWrite(BIB, vitesse_MD);
}
void tourneDroite() { // Fonction : Le robot tourne à droite
  analogWrite(AIA, vitesse_MG);
  analogWrite(AIB, LOW);
  analogWrite(BIA, LOW);
  analogWrite(BIB, LOW);
}
void tourneGauche() { // Fonction : Le robot tourne à gauche
  analogWrite(AIA, LOW);
  analogWrite(AIB, LOW);
  analogWrite(BIA, vitesse_MD);
  analogWrite(BIB, LOW);
}
void robotSurPlaceGauche() { // Fonction : Le robot fait la toupie à gauche
  analogWrite(AIA, LOW);
  analogWrite(AIB, vitesse_MG);
  analogWrite(BIA, vitesse_MD);
  analogWrite(BIB, LOW);
}
void robotSurPlaceDroite() { // Fonction : Le robot fait la toupie à droite
  analogWrite(AIA, vitesse_MG);
  analogWrite(AIB, LOW);
  analogWrite(BIA, LOW);
  analogWrite(BIB, vitesse_MD);
}
void stopRobot() { // Fonction : Le robot s'arrête
  digitalWrite(AIA, LOW);
  digitalWrite(AIB, LOW);
  digitalWrite(BIA, LOW);
  digitalWrite(BIB, LOW);
}
// Fonction du capteur
unsigned int mesureDistance() { // Déclaration de la variable mesureDistance
int cm = distanceCM.ping_cm(); // déclaration de la variable locale (cm) à qui on assigne la distance
  if (cm > distanceObstacle || cm <= minimumDistance) // on définit la plage de détection du capteur
  {
    PasObstacle = true; // on renvoie true si pas d'obstacle dans la plage définie dans le test
  }
  else
  {
    PasObstacle = false; // sinon on renvoie false (si il y a un obstacle dans la plage définie)
  }
return cm; // on retourne la distance de l'obstacle à la fonction
}

```



## 7 - Récapitulatif

1. **Déclaration d'une variable** : on vient avec cette ligne stocker la valeur à droite du signe égal (=) dans la **variable** à gauche du signe égal.

```
int maximumDistance = 50;
```

Dans notre cas, cela signifie que la **variable** appelée *maximumDistance* viendra prendre la valeur 50. Le mot clé *int* en début de phrase signifie que la variable sera un nombre entier.

2. **Les blocs d'instructions** : **setup** regroupe toutes les instructions qui seront exécutées au démarrage du programme. La fonction **setup** n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino.

**loop** (boucle en anglais) contient les instructions que l'on souhaite voir exécutées encore et encore tant que l'Arduino est branché.

```
void setup() {}
void loop() {}
```

3. **Les fonctions** : sont des instructions qui permettent d'exécuter une ou plusieurs actions. Les fonctions sont définies avec :

**Un nom** : ce qu'on devra taper pour appeler la fonction.

**Une ou des entrées** : ce sont des variables passées à la fonction appelées paramètres ou arguments. Ces arguments sont placés entre parenthèses.

**Une sortie** : le résultat de la fonction qui peut être stocké dans une variable.

Prenons l'exemple de la fonction suivante :

```
analogWrite(A1B, LOW);
```

Dans ce cas, le nom de la fonction est **analogWrite**. Nous passons deux paramètres à la fonction : *A1B* et *LOW*. La fonction **analogWrite** n'a pas de sortie. Avec cette fonction, nous éteignons la broche située sur la broche passée avec le premier paramètre (qui peut être un nombre ou une variable). Lorsque le second argument est placé à *LOW*, on vient d'arrêter le moteur. Tandis qu'on mettra en marche le moteur si le second argument utilise un nombre entier supérieur à *0*. Ce nombre est compris entre *0* et *255*. Faisons l'analogie avec l'électricité :

*0* correspond à *0 volt*.

*255* correspond à *5 volts*.

Plus la tension est élevée plus le moteur tournera vite.

4. **Autres fonctions**

**pinMode** configure la broche spécifiée dans le premier paramètre pour qu'elle se comporte soit en entrée (INPUT), soit en sortie (OUTPUT) passée avec le second paramètre :

```
pinMode(A1A, OUTPUT);
```

**delay** correspond au temps d'exécution d'une fonction. La durée est mesurée en millisecondes:

```
avanceRobot()
delay(200);      Le robot avancera pendant 200ms
```

Pour aller plus loin : [Compléments sur la programmation Arduino](#) par Eskimon

## 8 – Glossaire

### Actionneurs

Les actionneurs sont des composants matériels, qui une fois correctement connectés à la carte Arduino, permettent d'agir sur le monde extérieur. Ils convertissent une valeur électrique en action physique.

### Arduino

Arduino est une plate-forme libre de création d'objets électroniques composée d'un appareil d'entrée-sortie configurable (*microcontrôleur*) et d'un environnement de programmation.

### Bibliothèque

Une bibliothèque est un ensemble de fonctions regroupées et mises à disposition des utilisateurs d'Arduino.

### Capteurs

Les capteurs sont des composants matériels, qui une fois correctement connecté à la carte Arduino, peuvent fournir des informations sur le monde extérieur.

### Circuit imprimé

Un circuit imprimé est un support, en général une plaque, permettant de relier électriquement un ensemble de composants électroniques entre eux, dans le but de réaliser un circuit électronique.

### Fonction

Une fonction effectue un traitement en fonction des infos qu'on lui donne (les arguments). L'instruction « return » sert à renvoyer le résultat des traitements effectués par la fonction à ce qui a appelé la dite fonction

### IDE (*Integrated Development Environment*)

Environnement de développement intégré (*EDI*) est un programme regroupant un ensemble d'outils pour le développement de logiciels. Un EDI regroupe un éditeur de texte, un compilateur, des outils automatiques de fabrication, et souvent un débogueur.

### Interface de programmation

Une interface de programmation applicative, souvent désignée par le terme *API (Application Programming Interface)* est une « façade », surcouche clairement délimitée afin de simplifier la programmation en cachant les détails de la mise en œuvre. Plusieurs interfaces existent pour programmer une carte Arduino : Ardublock, Scratch, ou encore Blockly.

### Langage de programmation

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. Le langage des cartes Arduino est le C/C++.

### LED

Une LED ou DEL (*diode électroluminescente en français*), ou est un composant capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. Les LEDs laissent passer le courant que dans un seul sens, on parle de polarité des deux pattes, une positive (+, anode) et une négative (-, cathode).

### Moniteur sériel

Le moniteur sériel est un élément de l'environnement de programmation Arduino. Il permet de recevoir et d'envoyer des messages en communication sérielle à partir de l'ordinateur.

### Pin

Il s'agit des ports de l'Arduino. Ce sont les broches qui permettent de connecter des fils à la carte. Elles seront déclarées en entrée (INPUT) d'information ou en sortie (OUTPUT). (Cf. Tx - Rx)

### Platine d'essai

Une platine d'essai est un support, le plus souvent en plastique, qui comporte des petits trous dans lesquels on peut positionner des composants ainsi que des fils qui permettent de réaliser un circuit électrique.

### Platine de prototypage

Une platine de prototypage est une plaque en époxyde ou en bakélite (plastique synthétique) qui comporte des trous et des lignes de cuivres permettant d'y souder des composants pour réaliser un circuit électronique.

### Servomoteur

Moteur qui peut recevoir des informations de positionnement et les atteindre de manière autonome.

### Shield

Carte comprenant un circuit complexe qui se connecte directement à l'Arduino et qui assure une fonction spécifique (communication internet, lecture de mp3, etc...).

### Variable

Une variable est un espace mémoire caractérisé par un nom dans lequel on range des informations : textes, nombres,... et que l'on pourra utiliser à tout moment dans le programme.



## Jouer à "Robot-idiot" pour s'initier aux algorithmes

Qu'est ce qu'un algorithme ? Et pourquoi ne pas répondre à cette question en jouant ?

### Jeu

« robot-idiot »

### Acteurs

Adulte et enfants

### Résumé

Le « robot-idiot » doit sortir d'un petit labyrinthe que l'on aura construit dans le séjour en déplaçant quelques tables ou chaises, ou en dessinant à la craie sur le sol de la cour. On se met dans la « peau » d'un robot pour voir ce qu'il peut ou pas faire.

 Comprendre l'informatique en jouant à faire le robot  
<https://youtu.be/9AtmJ9mTaB0>

### La fiche d'activité

Disponible ici.  
 <https://pixees.fr/wp-content/uploads/2015/09/Activit%C3%A9-d%C3%A9branch%C3%A9e-Le-jeu-du-robot-idiot.doc>

L'activité dans sa première partie est adaptée au plus petit (dès le début du primaire), de même que la version proposée en vidéo, la seconde partie « La rupture et si on connaissait pas la sortie ?! ? » est prévue pour aller plus loin quand l'enfant est en cycle 3 (à partir du cours moyen).

### Références

Un document complet pour le parent ou l'animateur (<http://images.math.cnrs.fr/Dis-maman-ou-papa-c-est-quoi-un.html>)  
 une vidéo récréative sur le sujet (<https://files.inria.fr/mecsci/grains-videos3.0/videos/18-algorithmes.mp4>)  
 et un tutoriel vidéo « Comprendre l'informatique en jouant à faire le robot » par Marie Duflot.

### Objectif

Initier les enfants à la notion d'algorithme en les faisant agir et en se dirigeant selon un « programme » préparé au préalable. Donner un sens à la notion « d'algorithme » à travers un savoir-faire concret. Mais aussi proposer un savoir-être pertinent par rapport à l'intelligence mécanique (la machine calcule très vite et de manière exacte ... mais est bête comme nos pieds !).

## Notions scientifiques

Algorithme (<https://pixees.fr/?p=435>),

Variable ([http://fr.wikipedia.org/wiki/Variable\\_%28math%C3%A9matiques%29](http://fr.wikipedia.org/wiki/Variable_%28math%C3%A9matiques%29)),

Bug



Comprendre la différence entre mon intelligence et celle de la machine en jouant à « robot-idiot »

<https://youtu.be/inXtI0KgeZI>

## Initiation au jeu

Préparer la pièce, au préalable faire un parcours simple sans trop d'obstacle puis selon la compréhension des enfants, élever le niveau.

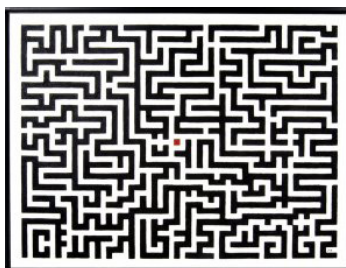
Celui qui incarne le rôle du robot n'a pas le droit de comprendre le langage humain, mais juste un langage très limité pour effectuer une action décomposée en étapes élémentaires.

## Actions

L'enfant qui imitera le robot ne pourra que :

- Avancer d'un pas
- Tourner à gauche d'un quart de tour
- Tourner à droite d'un quart de tour

Comment sortir d'un labyrinthe ?



On fabriquera des petites cartes à jouer en découpant un vieux carton en petits carrés avec les mots « avancer », « gauche », « droite ». Et on lui donnera une séquence de ces cartes qui sera son « algorithme ».

Il devra exécuter cet algorithme sans « réfléchir » (gare au mur – et à la rigolade – s'il y a un bug !). Ensuite on aura sûrement envie de ne pas répéter « avance d'un pas, avance d'un pas, avance d'un pas » mais « avance de trois pas ». Donc l'instruction aura une valeur variable qui permettra d'avancer plus efficacement. Avec un crayon à papier et une gomme, on pourra mémoriser la valeur et l'effacer ensuite.

Le jeu pourra se compliquer s'il y a une porte (concrétisée par un objet quelconque) qui peut-être fermée ou ouverte, sans qu'on le sache à l'avance. Il faudra alors introduire une condition dans notre algorithme : « si la porte est fermée alors [fais le tour] »... mais bien entendu il faudra expliquer en détail ce que veut dire « fais le tour » ! Pour le robot, il y aura alors deux paquets de cartes à choisir selon la condition.



Ce qui arrivera  
en cas de bug !

### Aller plus loin

Tester deux langages différents.







Un langage formel se distingue d'une langue naturelle par sa spécialisation, son caractère artificiel, le caractère limité de son lexique et la simplicité des règles qui régissent sa grammaire. Un exemple simple est le langage formé de quatre mots : « nord », « sud », « est » et « ouest » et d'une construction, la séquence, qui permet de former des suites de tels mots. Ce langage permet d'indiquer un chemin à suivre sur une grille carrée.

Par exemple l'expression « nord, nord, nord, est, est, est, sud, sud, sud, ouest, ouest, ouest » indique, par exemple, de se déplacer de trois carreaux vers le nord, puis de trois carreaux vers l'est, puis de trois carreaux vers le sud et enfin de trois carreaux vers l'ouest, dessinant ainsi un carré sur le sol.

Ce même mouvement peut être exprimé dans le langage : « avancer, avancer, avancer, tourner-à-droite, avancer, avancer, avancer, tourner-à-droite, avancer, avancer, avancer, tourner-à-droite, avancer, avancer, avancer, tourner-à-droite » qui ne comprend que trois mots : « avancer », « tourner-à-droite » et « tourner à gauche », composés par une opération de séquence.

On pourra alors étudier ces deux langages et les comparer.

- Apprendre à traduire un itinéraire d'un langage dans un autre.
- Étudier les avantages et inconvénients de chaque langage. On peut, par exemple, introduire une petite erreur exprès dans la séquence d'instructions, comme un « tourner à droite d'un quart de tour » à la place d'un « tout droit ». Avec le langage « boussole » on voit que l'erreur aura pour conséquence un décalage dans le déplacement et que la boussole nous remettra dans le bon sens rapidement (sans pour autant réparer l'erreur). Avec le langage « avance/tourne » on voit que cette erreur va nous conduire vraiment très loin (toute une partie du chemin va tourner et nous conduire très loin de l'objectif sur la grille). Mais attention : gardons à l'esprit que dans les deux cas, il s'agit bien d'un bug... loin ou pas, on ne va pas atteindre l'objectif ! Si il s'agit de sortir du labyrinthe mais que juste à côté de la porte il y a un piège... alors on ne veut vraiment pas se tromper. Même pas un tout petit peu !
- Rechercher des propriétés: par exemple une expression formée dans le premier des langages présentés ci-avant dans le second –, la mise en évidence de la redondance d'un langage – par exemple, « tourner-à-gauche » pourrait être remplacé par une séquence de trois « tourner-à-droite ».

 <b>ACTUALITÉ</b> > NEWS > TWITTER > PROJETS	 <b>RESSOURCES</b> > THÉMATIQUE > PUBLIC > FORMES > RECHERCHE	 <b>INTERVENTIONS</b> > CONFÉRENCES > ATELIERS > FORMATIONS > SÉMINAires > EXEMPLES	 <b>CONTACTS</b> > EN LIGNE > MESSAGE > LES THÉÂTRE	 <b>DIVERS</b> > PARTENAIRES > PRÉSENTATION > MENTIONS > LIENS > CREDITS > BACK OFFICE	<b>PIXEES</b> RESSOURCES POUR LES SCIENCES DU NUMÉRIQUE 
---	---	--	--	--	---



Retrouvez cette annexe sur :

<https://pixees.fr/dis-maman-ou-papa-cest-quoi-un-algorithme-dans-ce-monde-numerique-%E2%80%A8/>

Arnaud Reugnoat  
La Maison du Libre  
<https://wiki.mdl29.net/doku.php?id=projets:robotarduino>

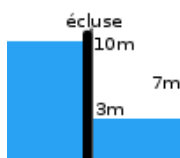


## Lois simples d'électricité

Les deux notions fondamentales sont le **courant** et la **tension**, elles peuvent être facilement comprises en prenant l'analogie d'un circuit électronique avec un cours d'eau.

### La tension

La tension est une différence de potentiel entre deux points du circuit, ce qui n'est pas très parlant. En prenant l'exemple d'une écluse sur un cours d'eau, une tension peut-être comparée à la différence de niveau entre l'amont de l'écluse et l'aval de l'écluse.



Ici la différence de niveau est de 7 mètres. Dans un circuit électronique, l'unité utilisée est le **Volt (V)**, et sa valeur peut être positive ou négative. La tension est donc la pression de l'eau.

### Le courant

Le courant est la quantité d'électrons qui parcourt le circuit électronique. Il est en général comparé au débit d'un cours d'eau, lui mesuré en m<sup>3</sup>/s. Le courant est le **diamètre du tuyau d'eau** qui permet ce débit. Le courant d'un circuit utilise l'unité **Ampère (A)**. L'intensité du courant se note avec la lettre **I**.

### Résistance

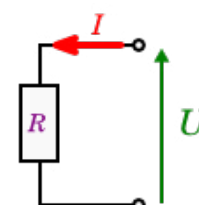
La résistance est ce qui s'oppose au débit de l'eau ou qui permet de le réguler, c'est le robinet ou les gorges d'une rivière, les rochers. La résistance est le composant électronique de base dont la principale caractéristique est d'opposer une plus ou moins grande résistance (mesurée en ohms) à la circulation du courant électrique. Son unité est donc le **Ohm (Ω)** et se note **R**. Exemple de résistance :



### Loi d'ohm

La loi d'Ohm permet de lier par une relation simple l'intensité, la résistance et la tension.

Avec la carte Arduino, il vous faudra protéger vos LED avec une résistance d'une valeur comprise entre 200Ω et 1kΩ.



$$U = R \cdot I$$

tension (volt)      résistance (ohm)      intensité (ampère)





## LA PROGRAMMATION EXPLIQUÉE AUX ENFANTS / ADOLESCENTS

### I) Est-ce qu'un ordinateur est intelligent ?

*[Poser la question : Qui croit qu'il est plus intelligent qu'un ordinateur et faire un vote à main levée en demandant à chacun d'expliquer sa position.]*

*Puis demander à un des participants d'exécuter une série d'instructions :*

- *S'asseoir, se lever plusieurs fois d'affilée*
- *Avancer d'un pas, puis d'un autre pas, et ainsi de suite jusqu'à arriver devant le mur et continuer de demander au participant d'avancer d'un pas*

*Puis demander aux autres enfants, si exécuter ces tâches requièrent de l'intelligence.*

#### **Explication :**

Un ordinateur calcule très vite, il peut répéter une action plusieurs millions de fois sans se lasser, mais il n'est pas intelligent.

La différence entre un humain et un ordinateur est que face à une situation nouvelle, l'être humain peut s'adapter : il essaiera de trouver des similitudes entre son expérience et cette nouvelle situation, expérimentera, fera des suppositions, bref il peut improviser.

Alors qu'un ordinateur lui est incapable d'agir en dehors de ce pourquoi il a été programmé. C'est d'ailleurs un des enjeux du développement des intelligences artificielles.

### II) Vous avez dit programmation ?

Un ordinateur, on vient de le voir, ne fait qu'exécuter les instructions qu'on lui a données.

Derrière chaque programme de l'ordinateur, une personne lui a dit quoi faire et comment le faire.

Et cette personne qui lui a dit quoi faire, c'est un programmeur.

Que ça soit pour créer un jeu vidéo, ou un site internet ou même une application de téléphone : il y a toujours un ou plusieurs développeurs qui ont expliqué aux ordinateurs, téléphones, tablettes quoi faire.

### III) Qu'est-ce que le langage informatique ?

*[Poser la question : qui sait quelle langue parle l'ordinateur ?]*

Alors le souci quand on veut expliquer quoi faire à un ordinateur, c'est qu'il parle une langue qui s'appelle le binaire : des 0 et des 1 et rien d'autre !

Le binaire, l'ordinateur le comprend très bien mais pour les humains, c'est compliqué à parler.

[Voici un petit exemple :

J'ai un ami néerlandais qui parle néerlandais mais qui ne parle pas français et moi je parle français mais pas néerlandais. Comment pouvons-nous faire pour nous comprendre l'un l'autre ?

Peut-être que nous connaissons une langue commune. Effectivement, lui et moi parlons anglais : on va pouvoir se comprendre !]

Avec l'ordinateur c'est pareil, il comprend le binaire, moi pas, on va donc trouver un langage commun : un langage informatique que lui et moi pourrons comprendre.

Des langages informatiques, il y en a pleins mais ils ont tous la même fonction : donner des instructions.

#### IV) L'algorithme

Mais même si on peut communiquer avec l'ordinateur, il a sa façon à lui de penser : le binaire. 0 et 1, on peut les assimiler à « Oui » et « Non ». Alors pour expliquer à quelqu'un qui ne comprend que les oui et non, comment faire pour jouer à *Super Mario* ? Il va falloir structurer ce qu'on lui demande, être clair, précis et méthodique. Et pour cela, on utilise les algorithmes.

Derrière ce nom un peu effrayant se cache en réalité quelque chose de très commun.

Un algorithme est une succession d'actions (difficile de faire plus bref comme définition).

Et le meilleur exemple d'algorithme : une recette de cuisine !

*Prendre des carottes.  
Les éplucher.  
Les râper.  
Les mettre dans un saladier.  
Ajouter de la vinaigrette.  
Servir.*

Voici l'algorithme des carottes râpées.

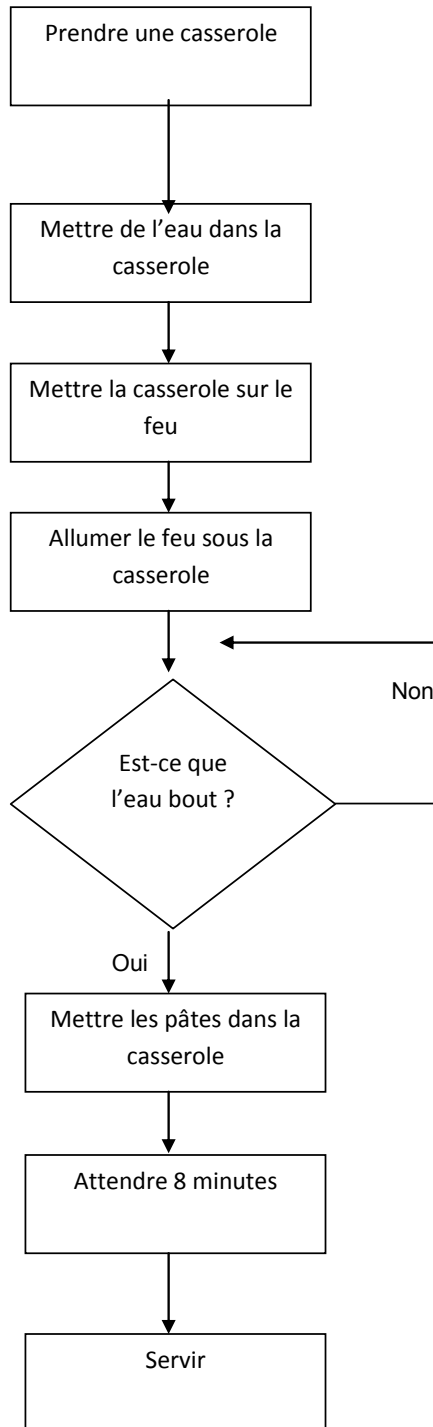
Alors vous vous doutez bien, que tous les algorithmes ne sont pas aussi simples. En réalité, il existe ce qu'on appelle des structures de contrôle qui permettent de les complexifier :

Voici les principales :

- Les conditions,
- Les boucles,
- Les variables.

**L'algorithme de la cuisson des pâtes**

Demandez à votre public d'écrire la recette pour faire des pâtes



## Les conditions :

Très souvent, nos actions dépendent de plusieurs paramètres :

*S'il fait beau → Je ne prends pas de pull.*

Les conditions permettent à un programme de faire une action en fonction d'une ou plusieurs informations. Les termes utilisés sont « Si » et « Sinon » (« If » et « Else » en anglais). Les conditions peuvent s'imbriquer les unes dans les autres.

S'il fait beau et s'il fait chaud → Alors je sors en Tee-shirt.

**Dans notre exemple :** **SI** l'eau bout, **alors** je mets les pâtes.

## Les boucles :

En programmation, un des maîtres-mots est **optimisation**. Alors souvent plutôt que de réécrire des lignes de codes, on demande à l'ordinateur de répéter une opération en utilisant une boucle.

Une boucle peut se répéter un certain nombre de fois, ou tant qu'une condition n'est pas vérifiée.

**Dans notre exemple :** tant que l'eau ne bout pas, il se repose en boucle la question (« Est-ce que l'eau bout ? »).

Un autre exemple :

Imaginons que vous n'avez une casserole ne pouvant faire des pâtes que pour 2 personnes et que vous êtes 4, il faudra donc répéter 2 fois, votre recette.

## Les variables :

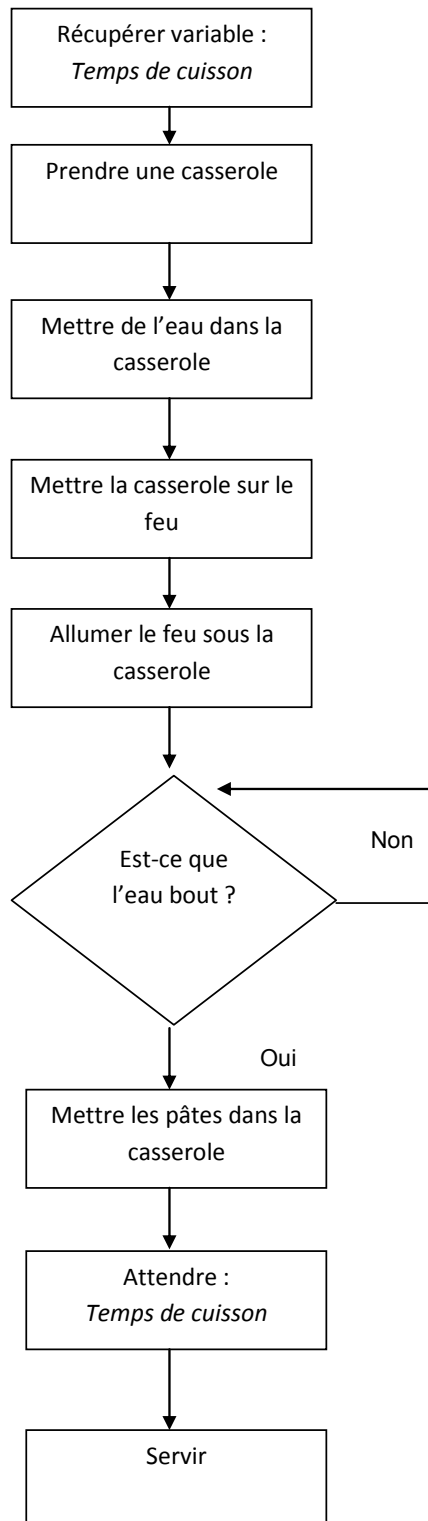
Les variables sont des espaces de mémoire dans l'ordinateur dans lesquels on peut lui demander de garder une information pour nous (soit sous la forme d'un nombre soit sous la forme de mots).

Dans un jeu vidéo, votre nombre de vie ou votre score sont des variables :

- Vous pouvez commencer avec votre variable « Vie » = 3,
- Quand vous touchez un ennemie, retire 1 à votre variable « Vie »
- Si vous tombez à variable « Vie » = 0, vous avez perdu

**Dans le cas de notre exemple :** la recette ne fonctionne que pour des pâtes ayant un temps de cuisson de 8 minutes. Pour du riz longue cuisson, le programme ne fonctionne pas.

Ce que nous allons donc faire, c'est qu'au début de notre recette, nous allons créer une variable que nous appellerons « temps de cuisson » qui va dépendre de ce que l'on veut faire cuire et après avoir versé le riz dans la casserole, on va « attendre : temps de cuisson ».





## Algorithme VS organigramme de programmation

Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat (<https://fr.wikipedia.org/wiki/Algorithme>).

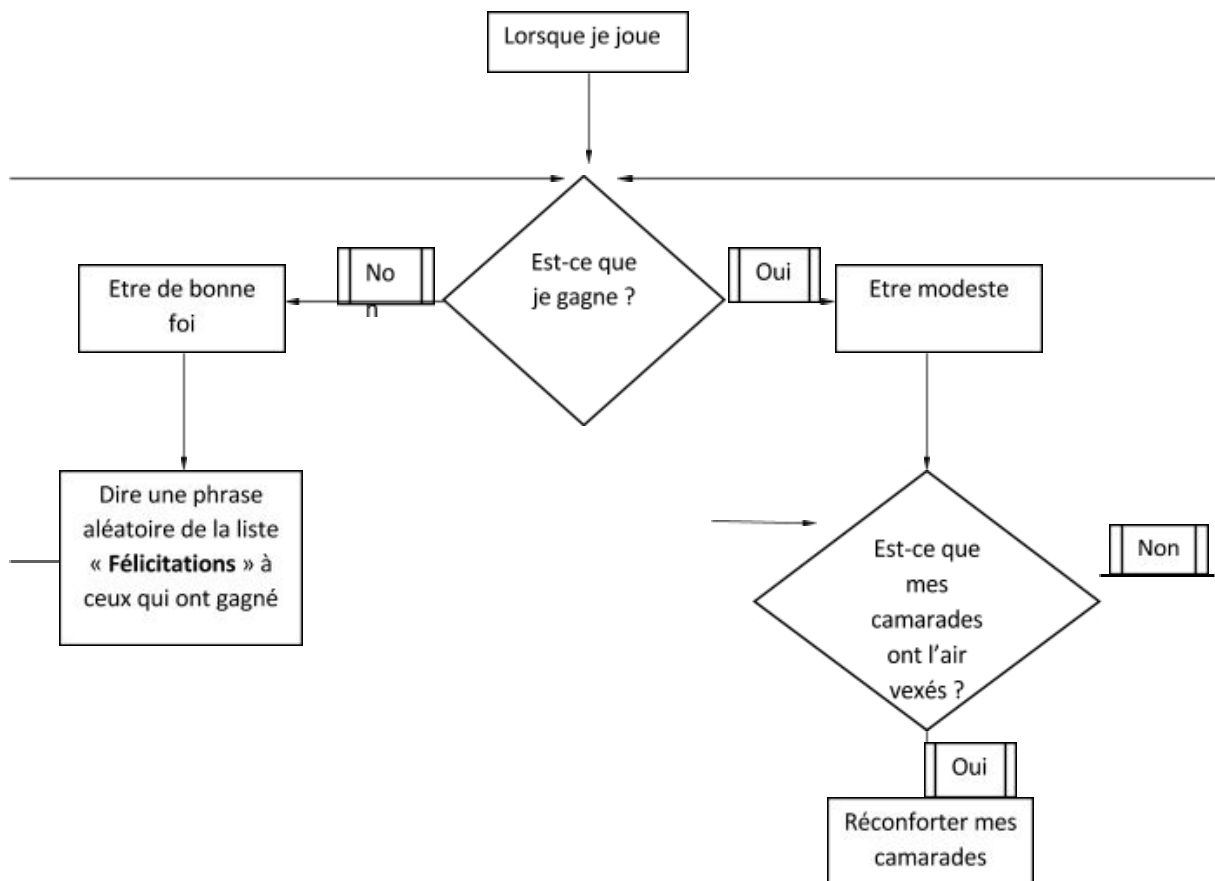
Un **organigramme de programmation** (parfois appelé algorigramme, logigramme ou plus rarement ordinogramme) est une représentation graphique normalisée de l'enchaînement des opérations et des décisions effectuées par un programme d'ordinateur. ([https://fr.wikipedia.org/wiki/Organigramme\\_de\\_programmation](https://fr.wikipedia.org/wiki/Organigramme_de_programmation)).

Une **différence** significative entre algorithme et programme (représenté par un organigramme) est que l'exécution d'un algorithme doit toujours se terminer avec un résultat, alors que celle d'un programme peut conduire à une boucle infinie (ne jamais s'arrêter). On peut donc utiliser des organigrammes de programmation pour représenter un algorithme néanmoins un organigramme ne représente pas forcément un algorithme.

Pour aller plus loin: <https://openclassrooms.com/courses/introduction-aux-algorigrammes>



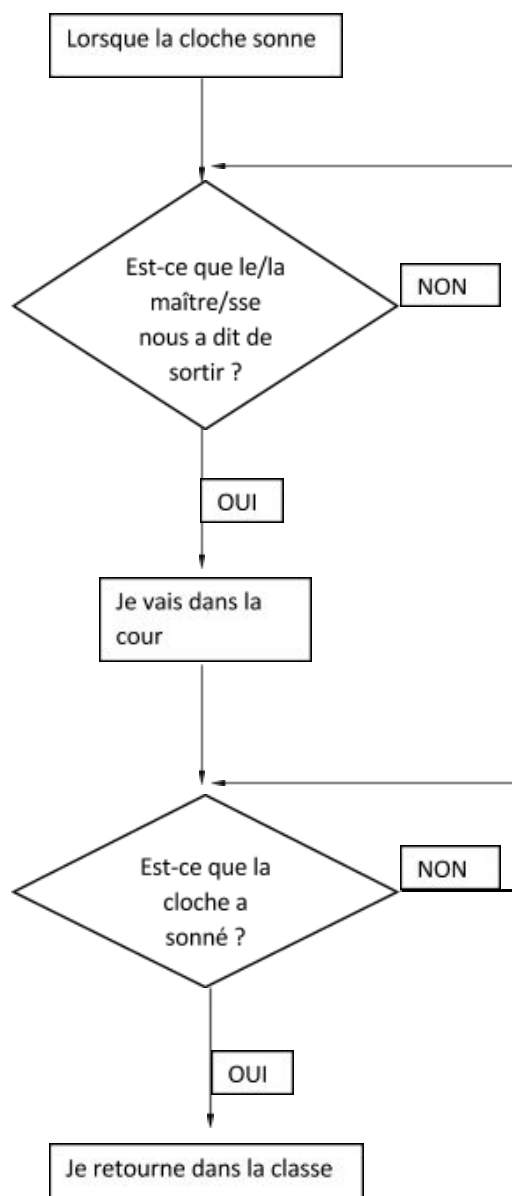
### Organigramme de programmation: bon(ne) perdant(e)



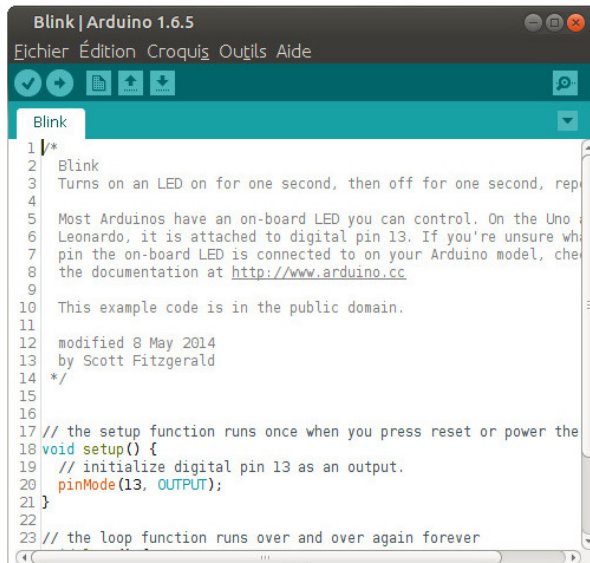
#### **Félicitations :**

- « Très bien, tu as réussi ! »
- « Bravo ! »
- « Félicitation ! »

## Organigramme de programmation : La récréation



## Installation de la bibliothèque NewPing



```

Blink | Arduino 1.6.5
Fichier Édition Croquis Outils Aide
Blink
1 /*
2  * Blink
3  * Turns on an LED on for one second, then off for one second, repeatedly.
4  *
5  * Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure which pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc.
6  *
7  * This example code is in the public domain.
8  *
9  * modified 8 May 2014
10 * by Scott Fitzgerald
11 */
12
13 // the setup function runs once when you press reset or power the board on
14 void setup() {
15   // initialize digital pin 13 as an output.
16   pinMode(13, OUTPUT);
17 }
18
19 // the loop function runs over and over again forever
20 void loop() {
21   digitalWrite(13, HIGH);   // turn the LED on (HIGH is the positive voltage)
22   delay(1000);              // wait for a second
23   digitalWrite(13, LOW);    // turn the LED off by making the pin LOW (no voltage)
24   delay(1000);              // wait for a second
25 }





```

Bien qu'il soit possible de programmer la carte Arduino avec Ardublock, Scratch, ou encore Blockly nous allons utiliser l'environnement Arduino.

### Lancer l'IDE Arduino

### Késako une bibliothèque sous Arduino ?

Les bibliothèques (ou librairie) permettent d'appeler des fonctions toutes prêtes par rapport à un shield ou un capteur. La bibliothèque NewPing permet d'utiliser facilement le capteur de distance (le HC-SR04).

<b>Bibliothèque NewPing</b>	 Télécharger le fichier <a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:newping.zip">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:newping.zip</a>
<b>Installation d'une bibliothèque</b>	<a href="http://rosa.lph.bzh/installation_librairie.html">http://rosa.lph.bzh/installation_librairie.html</a>  Télécharger le fichier <a href="http://rosa.lph.bzh/videos/installation_librairie.zip">http://rosa.lph.bzh/videos/installation_librairie.zip</a>
<b>Code test des moteurs</b>	 Télécharger le fichier <a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:test_moteurs.ino.zip">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:test_moteurs.ino.zip</a>
<b>Code du robot</b>	 Télécharger le fichier <a href="https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:robot_ligue_19110.ino.zip">https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:robot_ligue_19110.ino.zip</a>



Retrouvez cette annexe sur :  
<https://frama.link/programmationRosa>

## Le Code de ROSA



### 1 Déclaration des constantes<sup>1</sup> et variables<sup>2</sup>

`#include <NewPing.h>` On importe la bibliothèque New Ping. Elle possède une fonction qui mesure la distance.

On définit les broches du capteur ultrason :

`#define trigPin 12` la broche (pin) Trigger est branchée sur la broche 12 de l'Arduino

`#define echoPin 11` la broche (pin) Echo est branchée sur la broche 11 de l'Arduino

`NewPing distanceCM (trigPin, echoPin);` on initialise la fonction DistanceCM

`int maximumDistance = 50;` distance maximale acceptée (de 0-450 cm)

`int minimumDistance = 0;` distance minimale acceptée (en cm)

`boolean PasObstacle = false;` valeur à `false` (ou 0) si pas d'obstacle détecté  
sinon la valeur prend `true` (ou 1)

`const int distanceObstacle = 15;` on définit la distance de détection d'un obstacle, ici 15 cm

`int attenteCapteur = 100;` valeur en ms

### Déclaration des variables pour la vitesse de chaque moteur

Les moteurs ne tournant pas exactement à la même vitesse, vous pouvez modifier les valeurs suivantes.

La valeur maximale acceptée est 255 (sans unité)

Analogie avec l'électricité : 255 correspond à 5 volts

`#define vitesse_MG 150` vitesse du moteur gauche

`#define vitesse_MD 150` vitesse du moteur droit

<sup>1</sup> Une constante est une variable dont la valeur est inchangeable lors de l'exécution d'un programme.

<sup>2</sup> Une variable est un espace de stockage qui associe un nom à une valeur. Les variables peuvent changer de valeur au cours du temps.

On déclare les broches 5, 6, 9, 10 en sortie pour piloter les 2 moteurs

Moteur A (moteur Gauche)

```
int AIA = 5;    connecté à la broche 5 de l'Arduino  
int AIB = 6;    connecté à la broche 6 de l'Arduino
```

Moteur B (moteur Droit)

```
int BIA = 9;    connecté à la broche 9 de l'Arduino  
int BIB = 10;   connecté à la broche 10 de l'Arduino
```

*\* Vous pouvez inverser les moteurs, en respectant bien le branchement sur la carte Arduino. Ex. le moteur gauche devient le moteur droit.*

## 2 La fonction SETUP

La fonction `setup()` est appelée au démarrage du programme. Cette fonction est utilisée pour initialiser les variables, le sens des broches, les bibliothèques utilisées. La fonction `setup` n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino.

```
void setup()  
{
```

Moteur gauche

```
pinMode(AIA, OUTPUT);  
pinMode(AIB, OUTPUT);
```

Moteur droit

```
pinMode(BIA, OUTPUT);  
pinMode(BIB, OUTPUT);
```

```
stopRobot(); le robot est à l'arrêt  
delay(300); pendant 300 ms
```

```
}
```

### 3 La fonction LOOP

La boucle (loop) = le programme principal. Le code dans cette fonction est exécuté en boucle. C'est la partie qu'on utilisera avec les enfants.

```
void loop()
{
  int distance = mesureDistance();
```

On stocke dans la variable distance la valeur du capteur en cm

```
  delay(attenteCapteur);
```

Attente en ms entre chaque mesure du capteur

```
  if (PasObstacle == false)
  {
```

S'il n'y a pas d'obstacle,

```
    avanceRobot();
  }
```

le robot avance.

```
  else
  {
```

Sinon, un obstacle est détecté,

```
    stopRobot();
    delay(300);
    reculeRobot();
```

on arrête le robot  
pendant 300 ms (cette valeur est modifiable)  
on recule le robot

```
    delay(200);
    tourneDroite();
    delay(200);
```

pendant 200 ms (on augmente la valeur  
si on souhaite reculer plus longtemps)  
le robot tourne à droite  
pendant 200ms

```
  }
```

Fin de la boucle principale

```
}
```

### 4 Les fonctions du robot préprogrammées

Nous pouvons utiliser les fonctions suivantes avec les enfants et changer le code principal qui se trouve dans la fonction LOOP :

- avanceRobot()
- reculeRobot()
- tourneDroite()
- tourneGauche()
- robotSurPlaceGauche()
- robotSurPlaceDroite()
- stopRobot()



Pour appeler une fonction dans la boucle principale (`loop`) on utilise la syntaxe suivante :

```
avanceRobot();
```

On rajoute un point-virgule en fin de ligne.

## 5 Écriture du code pour les fonctions du robot

```
void avanceRobot()
{
```

Moteur A

```
  analogWrite(AIA, vitesse_MG);
  analogWrite(AIB, LOW);
```

Moteur B

```
  analogWrite(BIA, vitesse_MD);
  analogWrite(BIB, LOW);
```

```
}
```

```
void reculeRobot()
{
```

Moteur A

```
  analogWrite(AIA, LOW);
  analogWrite(AIB, vitesse_MG);
```

Moteur B

```
  analogWrite(BIA, LOW);
  analogWrite(BIB, vitesse_MD);
```

```
}
```

```
void tourneDroite()
{
```

Moteur A

```
  analogWrite(AIA, vitesse_MG);
  analogWrite(AIB, LOW);
```

Moteur B

```
  analogWrite(BIA, LOW);
  analogWrite(BIB, LOW);
```

```
}
```

```
void tourneGauche()
{
```

Moteur A

```
  analogWrite(AIA, LOW);
  analogWrite(AIB, LOW);
```

Moteur B

```
  analogWrite(BIA, vitesse_MD);
  analogWrite(BIB, LOW);
```

```
}
```

```
void robotSurPlaceGauche()
{
```

Moteur A

```
  analogWrite(AIA, LOW);
  analogWrite(AIB, vitesse_MG);
```

Moteur B

```
  analogWrite(BIA, vitesse_MD);
  analogWrite(BIB, LOW);
```

```
}
```

```
void robotSurPlaceDroite()
{
```

Moteur A

```
  analogWrite(AIA, vitesse_MG);
  analogWrite(AIB, LOW);
```

Moteur B

```
  analogWrite(BIA, LOW);
  analogWrite(BIB, vitesse_MD);
```

```
}
```

```
void stopRobot()
{
```

```
  digitalWrite(AIA, LOW);
  digitalWrite(AIB, LOW);
  digitalWrite(BIA, LOW);
  digitalWrite(BIB, LOW);
```

```
}
```

Fonction qui retourne la valeur en cm du capteur ultrason

```
unsigned int measureDistance()
{
```

déclaration de la variable locale où sera stockée la distance en cm

```
int cm = distanceCM.ping_cm();
```

 on lit la valeur du capteur ultrason

on définit la plage de détection du capteur

```
if (cm > distanceObstacle || cm <= minimumDistance)
{
```

<pre>PasObstacle = false;</pre>	on renvoie <code>false</code> car il n'y a pas d'obstacle
<pre>}</pre>	
<pre>else</pre>	sinon
<pre>{</pre>	
<pre>PasObstacle = true;</pre>	on renvoie <code>true</code> si un obstacle est détecté
<pre>}</pre>	
<pre>return cm;</pre>	on retourne la distance du capteur en cm

```
}
```

## 6 Récapitulatif des instructions :

- **Déclaration d'une variable** : on vient avec cette ligne stocker la valeur à droite du signe égal (=) dans la variable à gauche du signe égal.

```
int maximumDistance = 50;
```

Dans notre cas, cela signifie que la variable appelée `maximumDistance` viendra prendre la valeur 50. Le mot clé `int` en début de phrase signifie que la variable sera un nombre entier.

- **Les blocs d'instructions** : `setup` regroupe toutes les instructions qui seront exécutées au démarrage du programme. La fonction `setup` n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino. `loop` (boucle en anglais) contient les instructions que l'on souhaite voir exécutées encore et encore tant que l'Arduino est branché.

```
void setup() { }
```

```
void loop() { }
```

- **Les fonctions** : sont des instructions qui permettent d'exécuter une ou plusieurs actions. Les fonctions sont définies avec :

Un nom : ce qu'on devra taper pour appeler la fonction.

Une ou des entrées : ce sont des variables passées à la fonction appelées paramètres ou arguments. Ces arguments sont placés entre parenthèses.

Une sortie : le résultat de la fonction qui peut être stocké dans une variable.

Prenons l'exemple de la fonction suivante :

```
analogWrite(AIB, LOW);
```

Dans ce cas, le nom de la fonction est `analogWrite`. Nous passons deux paramètres à la fonction : `AIB` et `LOW`. La fonction `analogWrite` n'a pas de sortie. Avec cette fonction, nous éteignons la broche située sur la broche passée avec le premier paramètre (qui peut être un nombre ou une variable). Lorsque le second argument est placé à `LOW`, on vient d'arrêter le moteur. Tandis qu'on mettra en marche le moteur si le second argument utilise un nombre entier supérieur à `0`. Ce nombre est compris entre `0` et `255`. Faisons l'analogie avec l'électricité :

`0` correspond à `0 volt`.

`255` correspond à `5 volts`.

Plus la tension est élevée plus le moteur tournera vite.

#### • Autres fonctions

`pinMode` configure la broche spécifiée dans le premier paramètre pour qu'elle se comporte soit en entrée (`INPUT`), soit en sortie (`OUTPUT`) passée avec le second paramètre :

```
pinMode(AIA, OUTPUT);
```

`delay` correspond au temps d'exécution d'une fonction. La durée est mesurée en millisecondes:

```
avanceRobot()  
delay(200);
```

Le robot avancera pendant 200ms

Pour aller plus loin : Compléments sur la programmation Arduino par Eskimon :  
<http://eskimon.fr/category/arduino/partie-2>

Arnaud Reungoat  
<http://mdl29.net>

Juin 2016  
License : GNU General Public License



Retrouvez cette annexe sur :  
<https://frama.link/codeRosa>

## Fiche individuelle

Mon projet prend comme base un robot  
(cocher les bonnes cases)

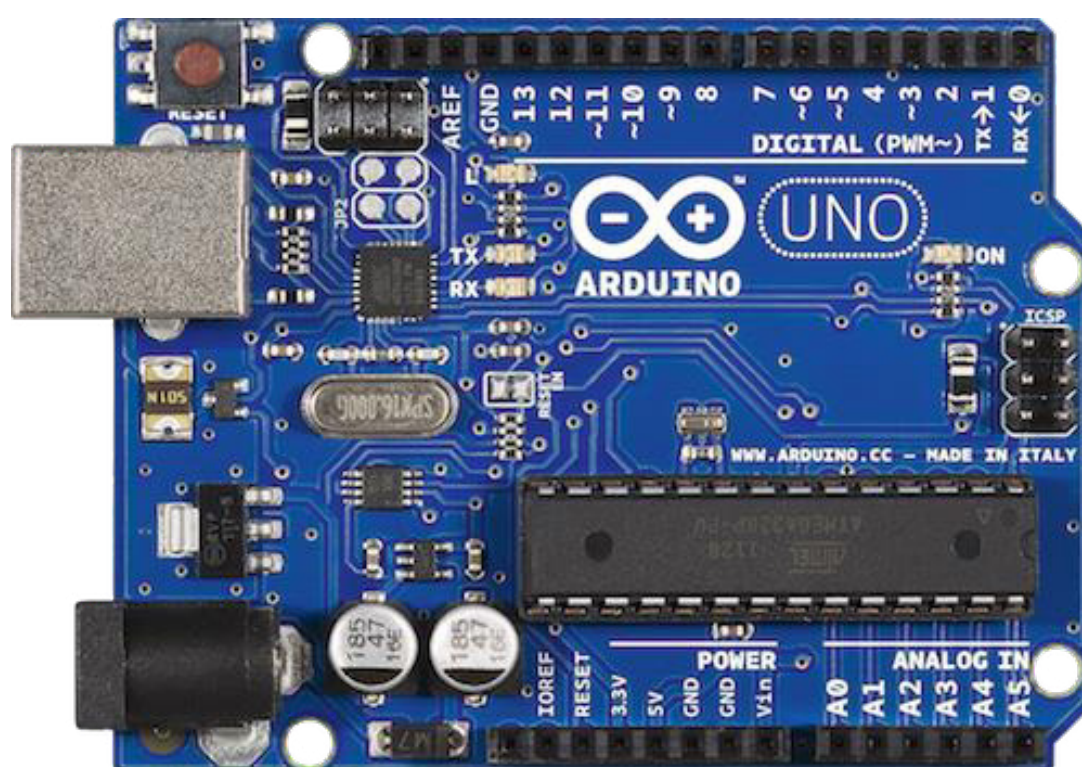
- humanoïde
- mékhano centré
- zoo centré
- autres, à préciser

Quel est le thème ou scénario ?

Le robot sera capable de  
(cocher les bonnes cases)

- avancer
- reculer
- tourner à droite
- tourner à gauche
- détecter un obstacle
- tourner sur place  
à droite
- tourner sur place  
à gauche

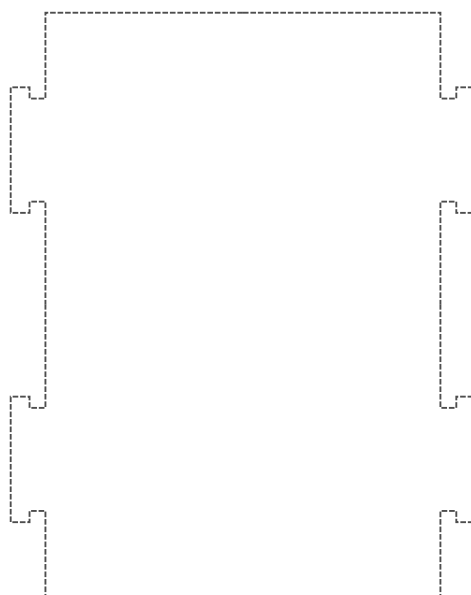
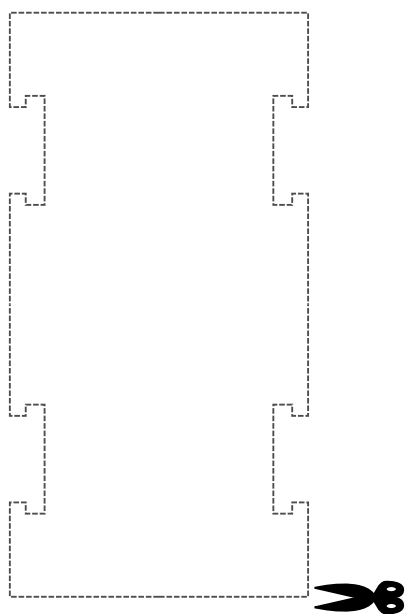
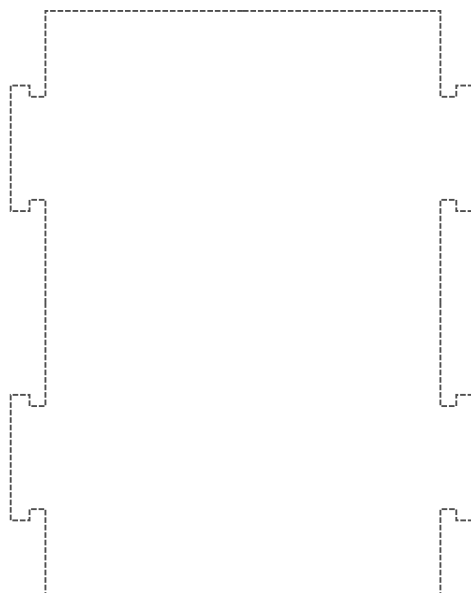
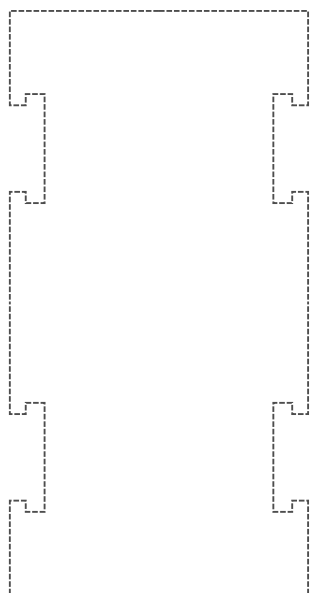
NOTRE PROJET DE ROBOTIQUE		ANNEXE 24
Nom de notre projet		
Nom de notre studio		
Nom des personnes composant notre studio		
Description de notre projet		
Notre projet prend comme base un robot (cocher les bonnes cases)	humanoïde	<input type="checkbox"/>
	mékhano centré	<input type="checkbox"/>
	zoo centré	<input type="checkbox"/>
	autres, à préciser	<input type="checkbox"/>
Il lui apporte les transformations suivantes (personnalisation)		
Quel est le thème ou scénario ?		
Le robot sera capable de (cocher les bonnes cases)	avancer	<input type="checkbox"/>
	reculer	<input type="checkbox"/>
	tourner à droite	<input type="checkbox"/>
	tourner à gauche	<input type="checkbox"/>
	détecter un obstacle	<input type="checkbox"/>
	tourner sur place à droite	<input type="checkbox"/>
	tourner sur place à gauche	<input type="checkbox"/>
Les grandes étapes (lister les différentes choses à faire pour la réalisation du projet)		
Qui a fait quoi ? (répartition du travail entre les différents membres de l'équipe)	Nom	Responsable de



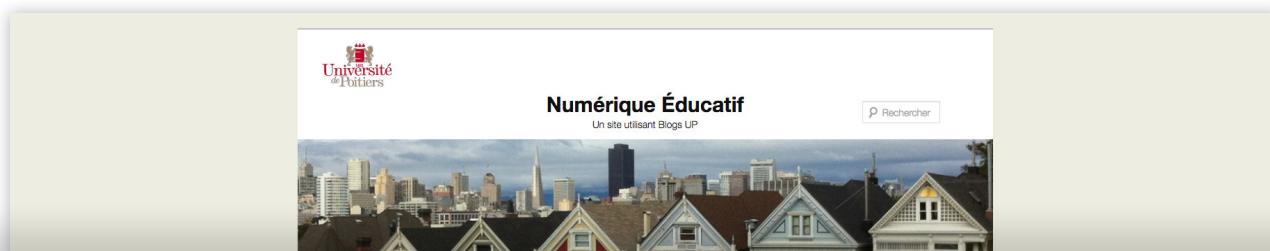


2016

# Robot boîte



PRÉSENTATION DU PROJET	ANNEXE 27
<b>Nom du robot</b>	
<b>Présentation du robot et de l'environnement:</b>	
<b>Quels sont les commandes informatiques utilisées :</b>	
<b>Quel est le but de notre robot:</b>	
<b>Comment avons-nous créé ce robot (forme déjà existante, robot créé complètement, quelle construction du décor etc.) ?</b>	



## Tutoriel Prezi

Actualisation de ce tutoriel en ligne avec les dernières modifications de l'interface Prezi le 23 septembre 2014.

Pour pouvoir faire une utilisation pertinente d'un outil, il est indispensable de le connaître, de s'entraîner, de le pratiquer. Après et seulement après on pourra choisir celui-ci, ou un autre. La présentation Prezi qui suit permet une prise en main rapide de cet outil de présentation dynamique.

Retrouver le tutoriel sur :

<http://blogs.univ-poitiers.fr/t-roy/2014/04/10/tutoriel-prezi-en-francais/>

Université de Poitiers - 15, rue de l'Hôtel Dieu - 86034 POITIERS Cedex - France Tél : (33) (0)5 49 45 30 00 - Fax : (33) (0)5 49 45 30 50 - [webmaster@univ-poitiers.fr](mailto:webmaster@univ-poitiers.fr)

Featuring WPMU Bloglist Widget by YD WordPress Developer



Retrouvez cette annexe sur :

<http://blogs.univ-poitiers.fr/t-roy/2014/04/10/tutoriel-prezi-en-francais/>



## Créer

### Tutoriels

- Votre première présentation  
(<http://sozi.baierouge.fr/pages/tutorial-first-fr.html>)
- Utiliser les calques  
(<http://sozi.baierouge.fr/pages/tutorial-layers-fr.html>)
- Les effets de transition  
(<http://sozi.baierouge.fr/pages/tutorial-transitions-fr.html>)
- Créer un lien vers une vue ou une URL  
(<http://sozi.baierouge.fr/pages/tutorial-links-fr.html>)
- Insérer une présentation Sozi dans une page HTML  
(<http://sozi.baierouge.fr/pages/tutorial-embedding-fr.html>)
- Montrer et cacher des objets  
(<http://sozi.baierouge.fr/pages/tutorial-showing-hiding-fr.html>)
- Insérer de l'audio ou une vidéo  
(<http://sozi.baierouge.fr/pages/tutorial-media-fr.html>)
- Convertir les présentations Sozi en PDF ou en vidéo  
(<http://sozi.baierouge.fr/pages/tutorial-converting-fr.html>)
- Améliorer les performances  
(<http://sozi.baierouge.fr/pages/tutorial-performance-fr.html>)

### Obtenir de l'aide et signaler un problème

- Foire Aux Questions et résolution des problèmes  
(<http://sozi.baierouge.fr/pages/faq-fr.html>)
- Rejoindre le groupe de discussion des utilisateurs de Sozi  
(<http://groups.google.com/group/sozi-users>)
- Signaler un problème et proposer de nouvelles fonctionnalités  
(<http://github.com/senshu/Sozi/issues>)

### Partager vos présentations

Il n'y a actuellement aucune plate-forme de partage des présentations Sozi. Vous pouvez trouver des exemples de présentations et ajouter des liens vers vos propres présentations sur le site Sozi Community Wiki (<http://sozi.wikidot.com/>).

News  
Releases

Inkscape  
Ext-Sozi  
lua-present

Atom feed  
GitHub  
Twitter  
Facebook  
Google+  
Ohloh



Retrouvez cette annexe sur :  
<http://sozi.baierouge.fr/pages/30-create-fr.html>

## La réalisation de ce parcours éducatif a été confiée à la Ligue de l'enseignement :

Antonin Cois,  
Responsable « D-Clics numériques »

Melpomeni Papadopoulou,  
Chargée de projet « Ingénierie pédagogique pour l'éducation au et par le numérique »

## Le parcours a mobilisé des experts locaux et nationaux :



Arnaud Reungoat  
Membre fondateur de l'association La Maison du Libre  
Animateur Petits Hackers  
<https://mdl29.net/>



Brian Benatier  
Coordinatrice départementale  
Francas de Seine Maritime  
[www.francas.asso.fr](http://www.francas.asso.fr)



Eric Chaumet  
Médiateur numérique,  
Réseau CANOPE  
[www.reseau-canope.fr](http://www.reseau-canope.fr)



Régis Leloup  
Coordonnateur technique et pédagogique  
Colommbus  
[www.colommbus.org](http://www.colommbus.org)



Stéphane Brunel  
Président de la Ligue de l'Enseignement de la Gironde  
Maître de conférences - Université de Bordeaux - Laboratoire IMS UMR 5218 CNRS  
Enseignant chercheur à l'ESPE de l'Académie de Bordeaux  
[www.laligue.org](http://www.laligue.org)



Adrien PAYET  
Directeur du CRREP  
Centre de Ressources en Robotique Educative et Professionnelle

Ainsi que les 22 personnes du groupe de travail réunies les 12, 13 et 14 septembre 2016 à Paris.

## Votre avis est important et nous permet d'améliorer la qualité de notre parcours éducatif.

Prenez quelques minutes pour nous le laisser sur :

[www.frama.link/satisfactionmediassociaux](http://www.frama.link/satisfactionmediassociaux)

Les contenus sont en Licence Creative Commons  
<https://creativecommons.org/choose> partagé dans les mêmes conditions et pas d'autorisation pour une utilisation commerciale



Cette licence permet aux autres de remixer, arranger, et adapter cette œuvre à des fins non commerciales tant qu'on nous crédite en citant nos noms et que les nouvelles œuvres sont diffusées selon les mêmes conditions.

**D-clicsnumeriques.org**

Le projet national est coordonné par

