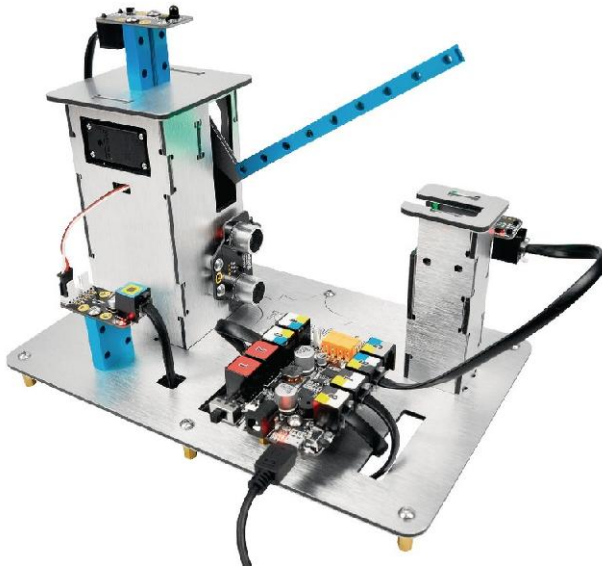


# Barrière

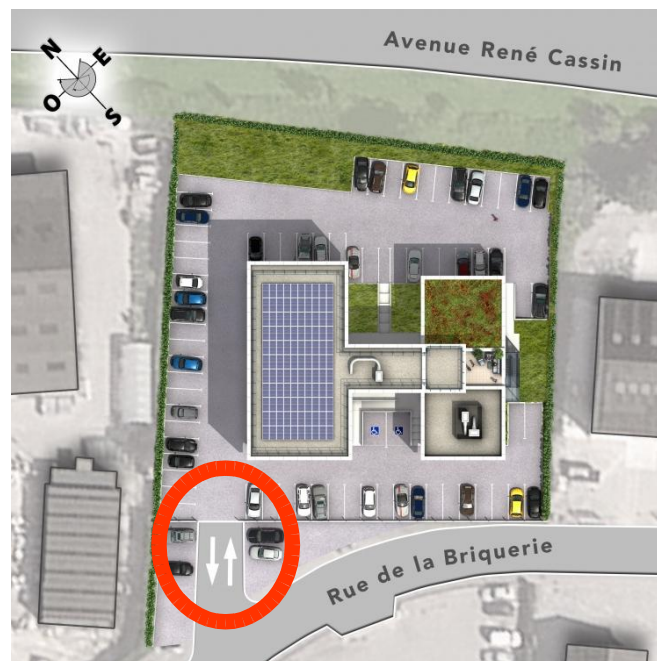
Comment contrôler l'entrée d'un parking ?



Makeblock



Un immeuble de centre ville dispose d'un parking aérien pour ses habitants. Ce parking est ouvert sur une rue fréquentée et il est souvent « squatté » par des véhicules extérieurs ce qui énerve les habitants de l'immeuble. Pour remédier à ce **problème** l'association des habitants s'est réunie et a voté pour l'installation d'une **barrière automatique**. Cette barrière doit répondre aux préoccupations des habitants de l'immeuble et elle doit respecter toutes les normes européennes en vigueur.



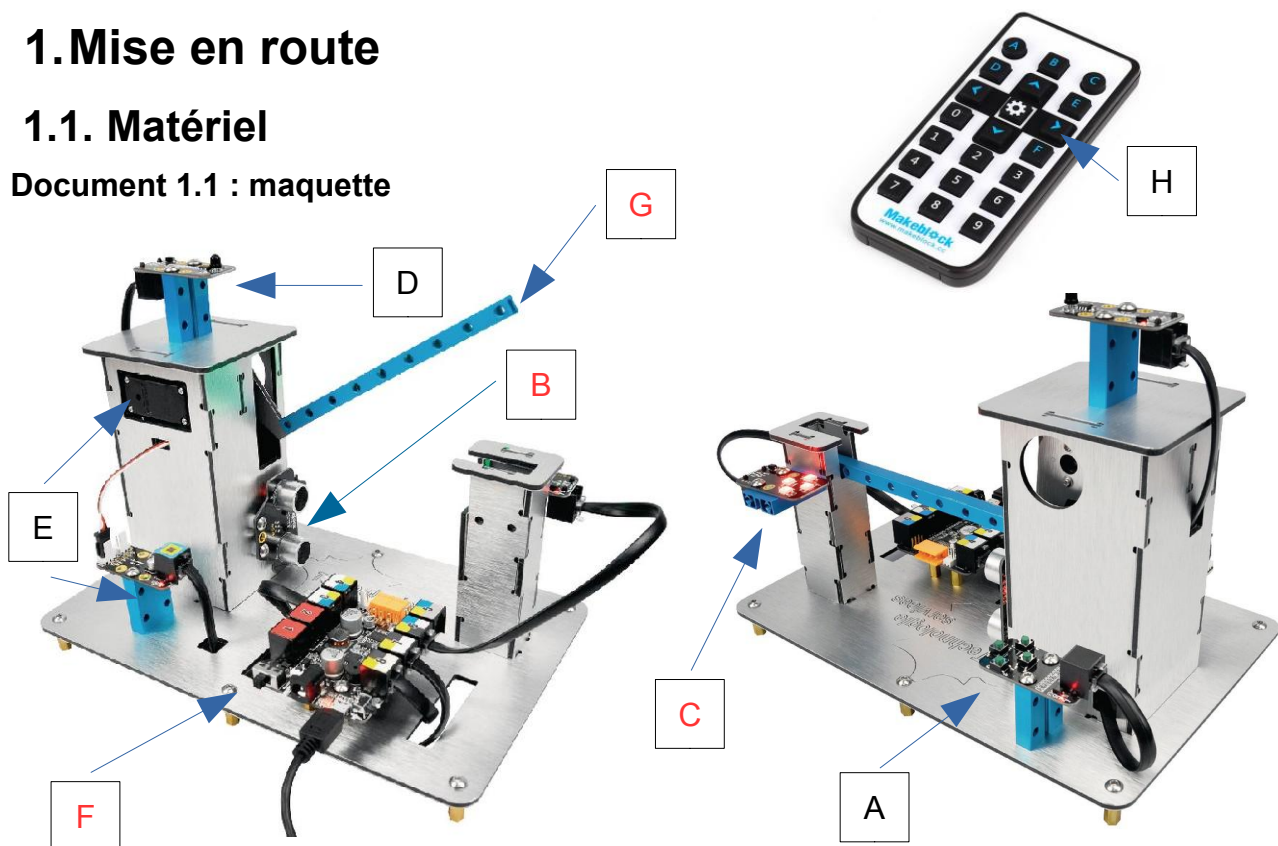
## Table des matières

1. Mise en route.....	3
1.1. Matériel.....	3
1.2. Connexions.....	5
1.3. Logiciel.....	6
1.4. Programmation.....	7
2. Un objet technique utile.....	8
2.1. Expression du besoin.....	8
2.2. « Bête à corne ».....	9
3. Un objet technique fonctionnel.....	10
3.1. Cahier des charges fonctionnel.....	10
3.2. Diagramme « pieuvre ».....	10
4. Principe de fonctionnement.....	11
5. Programmation de la barrière.....	12
5.1. Télécommande.....	12
5.2. Couleurs du gyrophare.....	13
6. Améliorations (en option).....	15
6.1. Signalétiques.....	15
6.2. Mise à niveau des sécurités.....	15
7. Synthèse.....	16
8. Auto-évaluation.....	17
9. Annexe.....	18
9.1. Automates.....	18
9.2. Programmes informatiques.....	19
9.3. Bug informatique.....	20
9.4. Méthode APTE.....	21
a) But.....	21
b) Historique.....	21
c) Les principes de base de la méthode.....	21

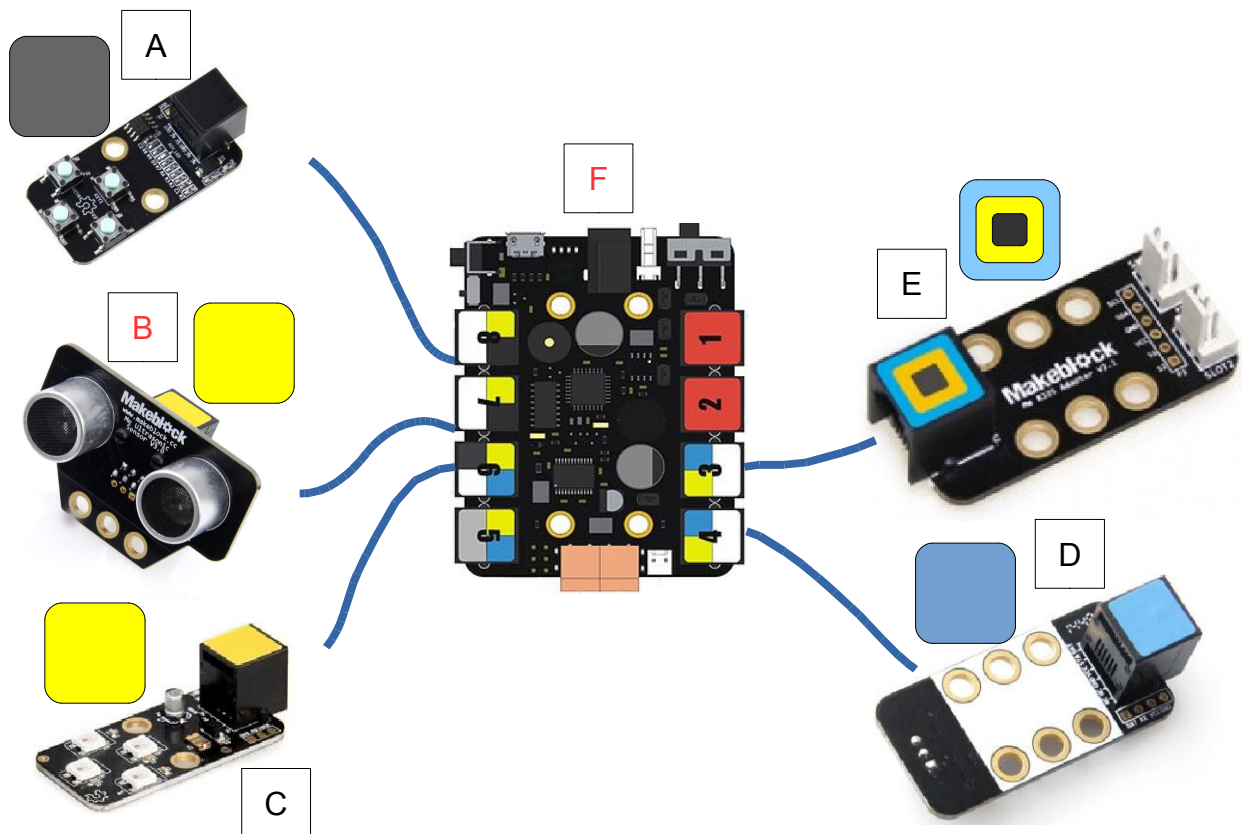
# 1. Mise en route

## 1.1. Matériel




Document 1.1 : maquette

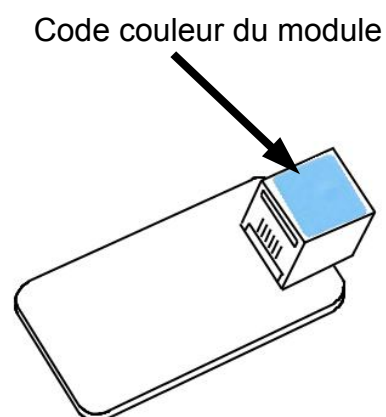


Document 1.2 : carte et modules électroniques de la maquette



### Document 1.3 : code couleur des connecteurs des modules

Couleurs	Fonctions	Modules utilisé pour ce port
	Port digital simple	<ul style="list-style-type: none"><li>Détecteur à ultrason</li><li>Module 4 LED RGB</li></ul>
	Port digital double	<ul style="list-style-type: none"><li>Récepteur IR</li></ul>
	Port analogique	<ul style="list-style-type: none"><li>Module à 4 boutons</li></ul>



### Document 1.4 : nomenclature des documents 1.1 et 1.2

H	Télécommande	
G	Lisse de la barrière	
F	Carte électronique Orion	
E	Adaptateur RJ25 et Servomoteur	oui
D	Récepteur des signaux de la télécommande	non
C	Module 4 LED RGB	oui
B	Détecteur d'objets à ultrason	non
A	Module à 4 boutons	non
Repères	Désignations	C'est un actionneur ? *

\* Un actionneur est un élément qui réalise une action (lumière, son, mouvement...) à partir de l'énergie qu'il reçoit (souvent électrique). Attention ! Un interrupteur n'est pas un actionneur ! Il déclenche une action mais il ne la réalise pas !

a) Vous devez compléter les étiquettes A , B ... sur les documents 1.1 et 1.2 à l'aide de la nomenclature du document 1.4.

b) Vous devez les compléter à l'aide du document 1.2 en observant la maquette et en vous aidant du document 1.3.

c) Dans le document 1.3, il est spécifié que certains ports sont digitaux. A votre avis cela signifie quoi ?

*Cela signifie que les données qui transitent par ces ports de communication sont digitales, c'est à dire que ces données ne peuvent prendre que 2 états possibles : 0 ou 1.*

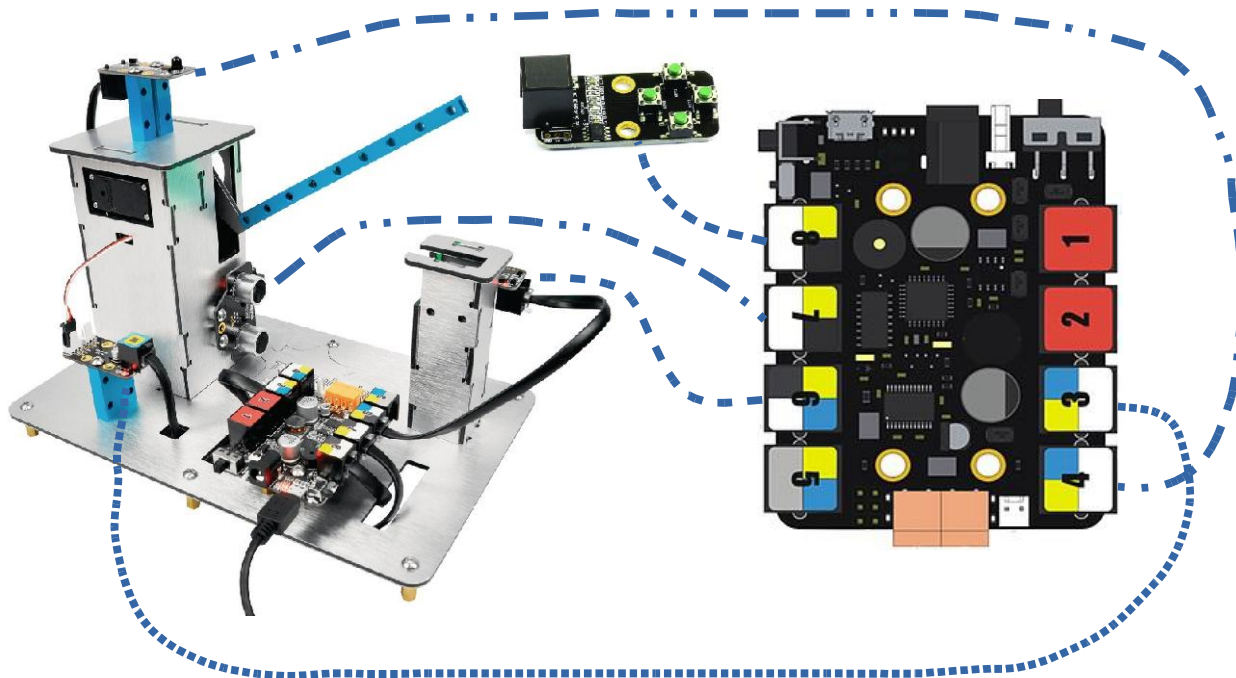
d) Donnez votre définition d'un actionneur. Attention ! Nous ne vous demandons pas ici un exemple mais bien une définition :

*Un actionneur est un élément de la partie opérative d'un automate qui agit physiquement à partir de l'énergie qu'il reçoit.*

## 1.2. Connexions

Vous allez faire fonctionner la maquette en respectant les différentes connexions.

a) Vous devez brancher la carte « Orion » aux modules de la maquette de la façon suivante :



N° port	Capteurs/actionneurs	E/S	Taille du câble
3	Adaptateur pour le Servomoteur	sortie	20cm
4	Récepteur des signaux de la télécommande	entrée	50cm
6	Module 4 LED RGB	sortie	20cm
7	Détecteur d'objets à ultrason	entrée	50cm
8	Module à 4 boutons	entrée	50cm



b) Vous devez relier la carte « Orion » à un port USB de votre ordinateur à l'aide du câble USB. Une alimentation extérieure n'est pas nécessaire !

Puisqu'une alimentation extérieure n'est pas utile, d'où provient l'énergie électrique à votre avis ?

*La mise en énergie de la carte et de la maquette se fait via le câble USB.*

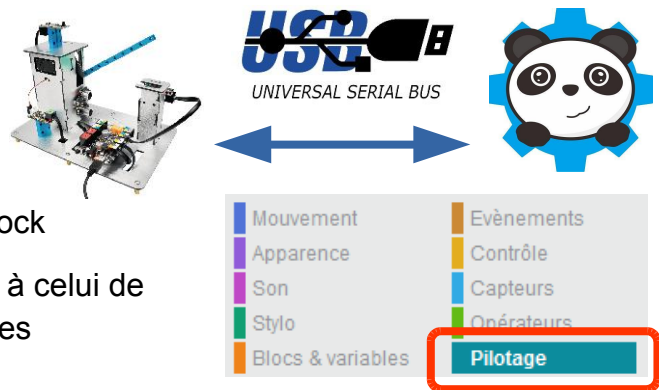


### 1.3. Logiciel

Vous devez suivre la procédure suivante pour programmer la carte :

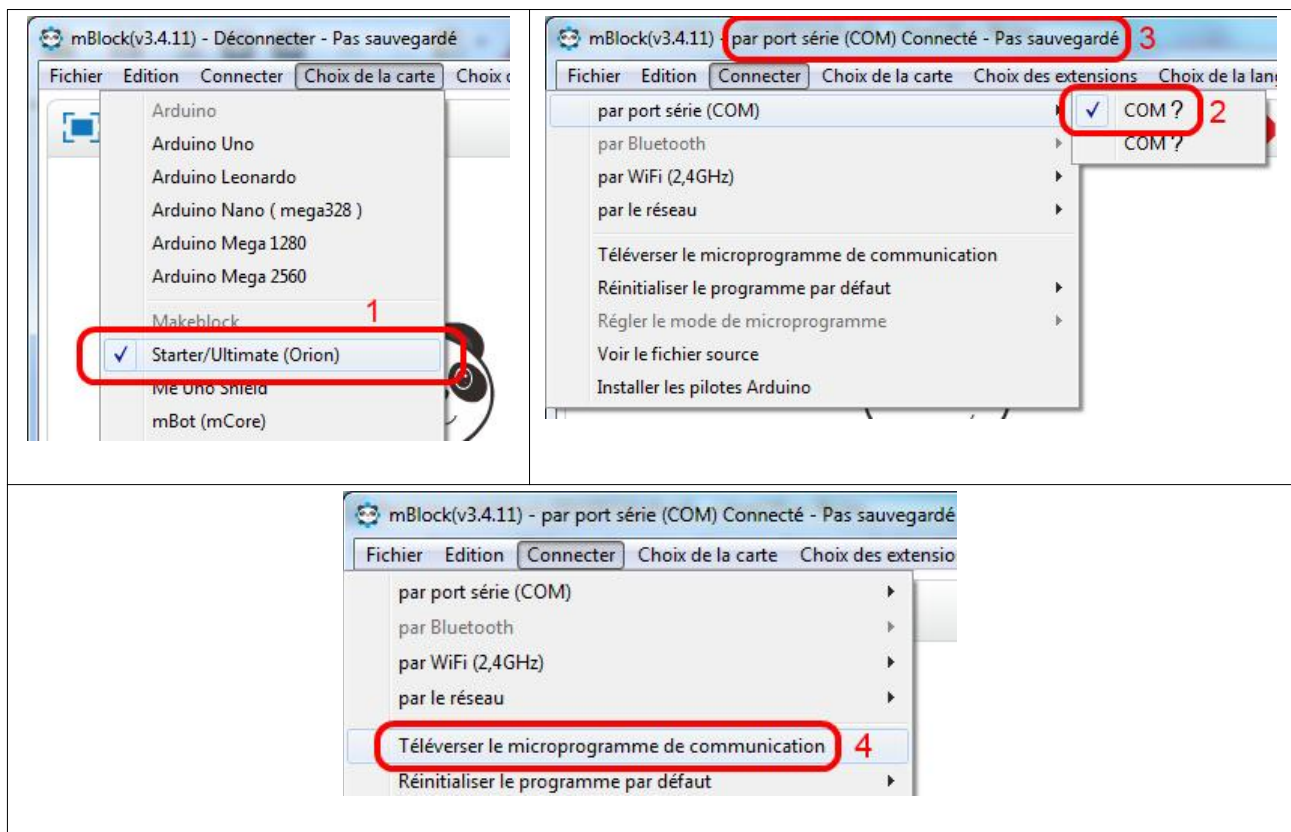
a) Reliez votre maquette à un ordinateur avec un câble USB et ouvrez le logiciel mBlock

L'environnement de ce logiciel est identique à celui de « Scratch », seul la programmation des cartes électroniques est en plus.



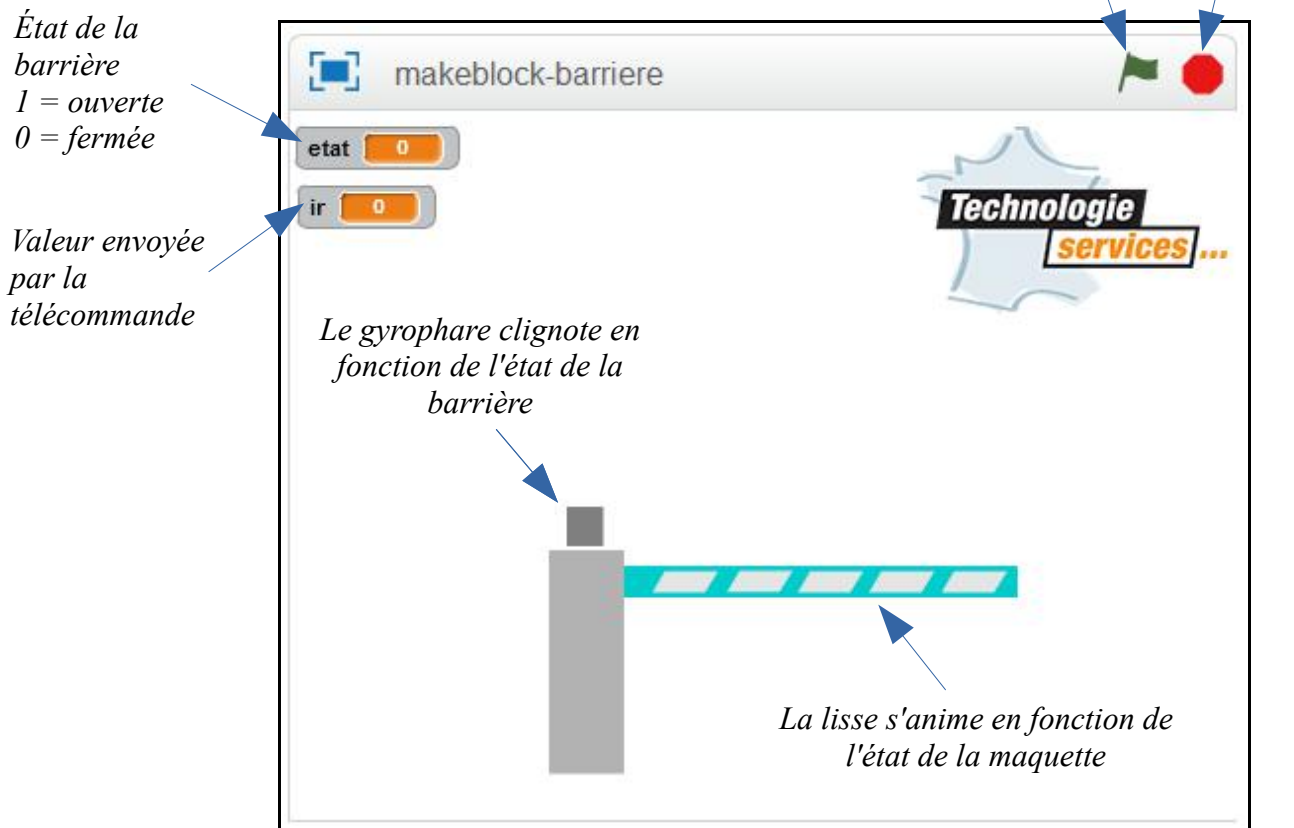
b) Avant de commencer, vous devez configurer le moyen de communication avec la bonne carte en utilisant la procédure suivante :

1. dans le menu « Choix de la carte » cliquez sur « Starter/Ultimate (Orion) »,
2. dans le menu « Connecter », cliquez sur le sous-menu « par port série (COM) » et choisissez un port valide (le numéro de port de communication dépend de votre système informatique),
3. vérifiez que la connexion est bien établie grâce à la mention écrite dans le bandeau de la fenêtre,
4. dans le menu « Connecter », cliquez sur « Téléverser le microprogramme de communication ».



## 1.4. Programmation

a) Vous devez ouvrir le fichier : « makeblock-barriere »



b) Le programme de la barrière fonctionne selon l'algorithme suivant :

1. la barrière s'initialise en position haute,
2. le programme mémorise le code émise par la télécommande,
3. si l'utilisateur a appuyé sur la touche ↑ de la télécommande ou la touche 4 de la maquette et que la barrière est *fermée* alors le programme OUVRE la barrière,
4. si l'utilisateur a appuyé sur la touche ↓ de la télécommande ou la touche 2 de la maquette et que la barrière est *ouverte* alors le programme FERME la barrière,
5. retour à l'étape 2



## 2. Un objet technique utile

De nos jours l'énergie est chère, les matériaux sont rares et polluants, c'est pourquoi nous devons créer des objets rationnels et utiles !

(voir document 2.1)

La démarche suivante est basée sur une méthode utilisée dans l'industrie. Elle vous aidera à justifier l'utilité d'une barrière automatisée dans le cadre d'un parking d'un immeuble.

Cette démarche s'appelle : méthode APTE (voir en annexe).

**Document 2.1 :**

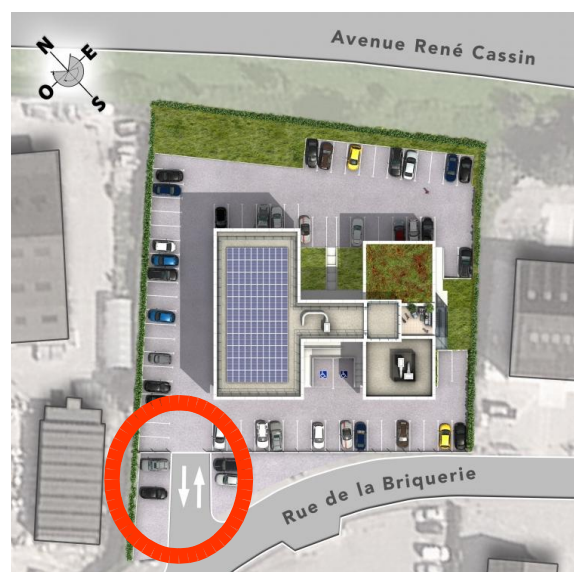


### 2.1. Expression du besoin

**Document 2.2 :** Ce que souhaitent les habitants de immeuble

Un immeuble de centre ville dispose d'un parking aérien pour ses habitants. Ce parking est ouvert sur une rue fréquentée et il est souvent « squatté » par des véhicules extérieurs ce qui énerve les habitants de l'immeuble. Pour remédier à ce **problème** l'association des habitants s'est réunie et a voté pour l'installation d'une **barrière automatique**.

Les habitants de l'immeuble veulent que cette barrière soit électrique, qu'elle puisse être commandée à distance avec une télécommande ou un clavier de secours, qu'elle ne puisse pas se refermer sur une voiture ou un passant et qu'elle respecte toutes les normes européennes en vigueur.



Vous devez répondre aux questions suivantes à partir des informations contenues dans le document 2.2 :

a) qui demande l'installation d'une barrière automatique ?

*L'installation d'une barrière automatique est demandée par les habitants de l'immeuble.*

.....



b) Cette installation doit agir et réguler quoi ?

*La barrière automatique doit réguler l'accès des voitures au parking aérien.*

.....

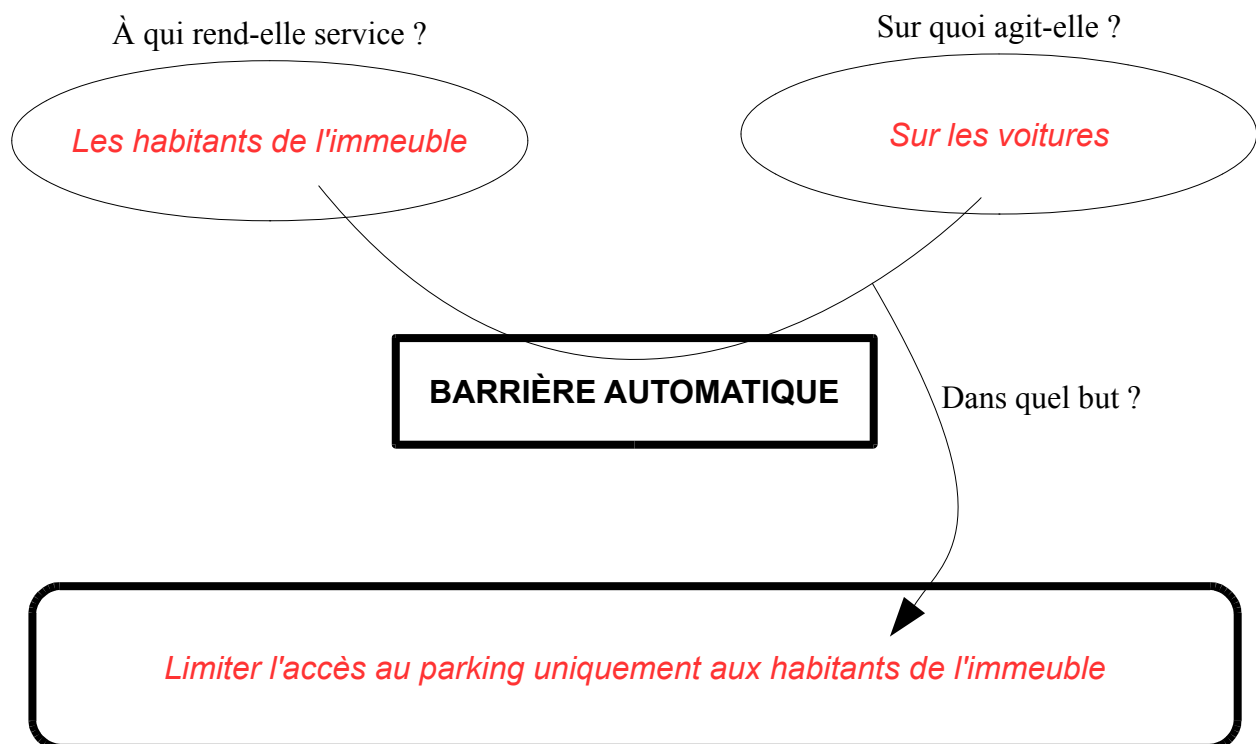
c) Pourquoi ces gens demandent-ils une telle installation ?

*Ils demandent l'installation d'une barrière automatique pour éviter les « squatteurs » d'utiliser leur parking.*

.....

## 2.2. « Bête à corne »

Vous devez compléter le diagramme suivant à partir des réponses du chapitre précédent :



### 3. Un objet technique fonctionnel

Un objet utile capable de rendre un réel service c'est bien mais faut-il qu'il soit pratique et fonctionnel ! (voir document 3.2)

La démarche suivante est la suite de la méthode APTE (voir en annexe). Elle a pour but de faire correspondre à chaque fonction une attente des utilisateurs et surtout une solution technique.

Document 3.2 :



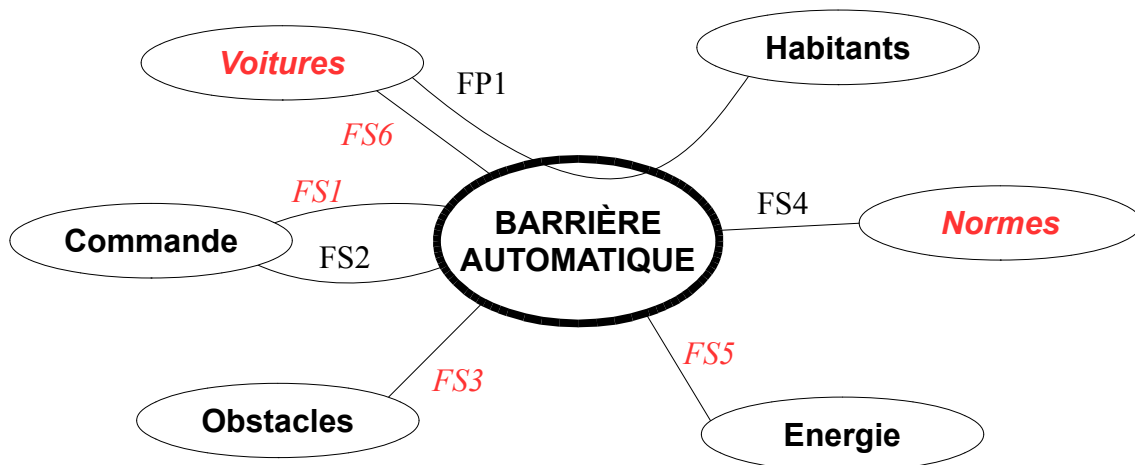
#### 3.1. Cahier des charges fonctionnel

Vous devez vous approprier le cahier des charges suivant et le compléter à l'aide du document 2.2 :

N°	Fonctions	Niveaux	Critères
Fonction principale 1	Limiter l'accès au parking	- <i>gabarit de véhicule</i>	- passage pour une voiture
Fonction de service 1	Commander l'ouverture à distance	- <i>faible distance</i> - <i>type de signal</i>	- 5m maximum - rayons infrarouges
Fonction de service 2	Commander l'ouverture avec un digicode de secours	- <i>code simple</i>	- 4 touches
Fonction de service 3	Ne pas se refermer sur un obstacle	- <i>voitures</i> - <i>passants</i>	- capteur à ultrason - capteur sur lisse
Fonction de service 4	Respecter les normes	- <i>types de normes</i>	- européennes
Fonction de service 5	Être alimentée en énergie	- <i>type d'énergie</i>	- électrique (5v pour la maquette)
Fonction de service 6	Être visible	- <i>la nuit</i>	- revêtement réfléchissant

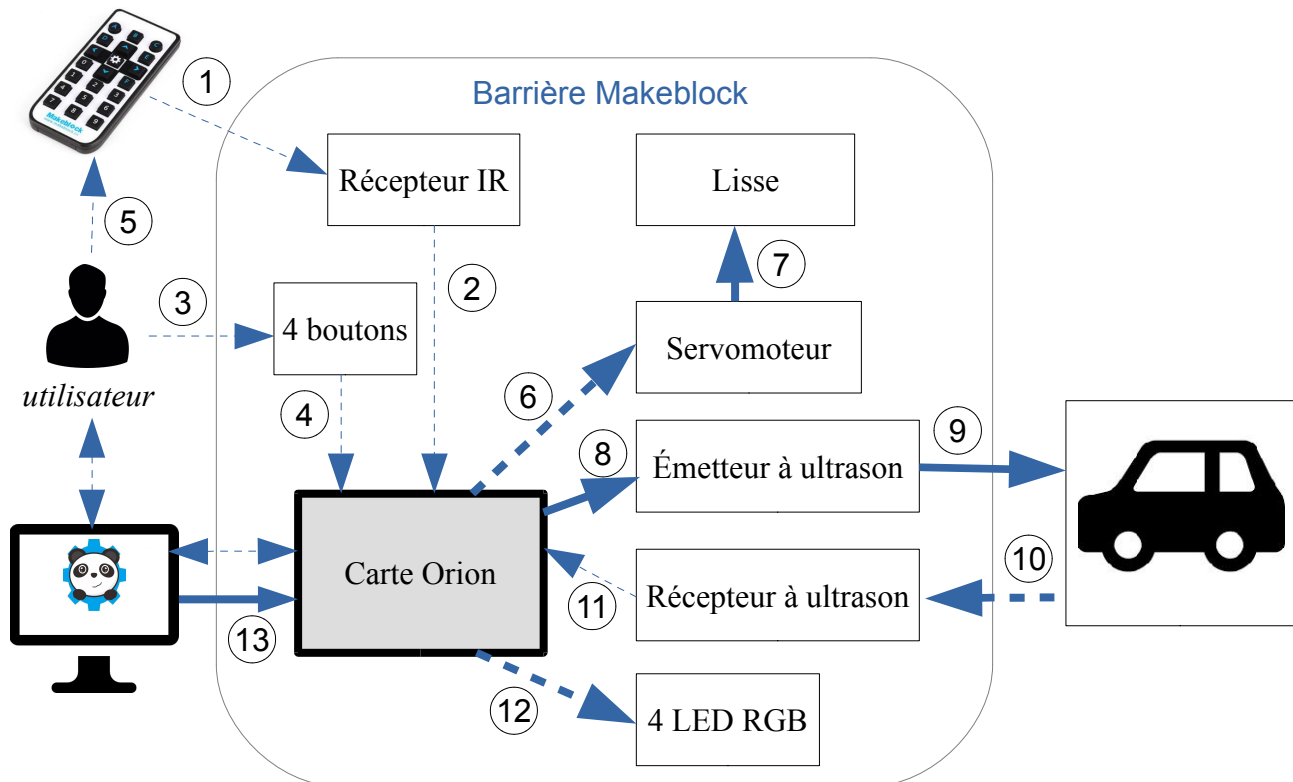
#### 3.2. Diagramme « pieuvre »

Vous devez compléter le diagramme suivant à l'aide du cahier des charges précédent :



## 4.Principe de fonctionnement

Document 4.1 : chaînes de l'énergie et de l'information



a) Vous devez compléter le tableau suivant en mettant, pour chaque flèche numérotée, une croix pour identifier la nature et le type de flux de chaque liaison.

Lien	Nature de l'interaction					Type de flux	
	Flèches	Signal sonore	Signal lumineux	Courant électrique	Mécanique	Consigne utilisateur	
1			X				X
2				X			X
3						X	X
4				X			X
5						X	X
6				X			X
7					X		X
8				X			X
9		X					X
10		X					X
11				X			X
12				X			X
13				X			X

## 5. Programmation de la barrière

### 5.1. Télécommande

a) Vous devez ouvrir le fichier : « makeblock-telecommande »

b) Exécutez le programme en appuyant sur le drapeau vert.

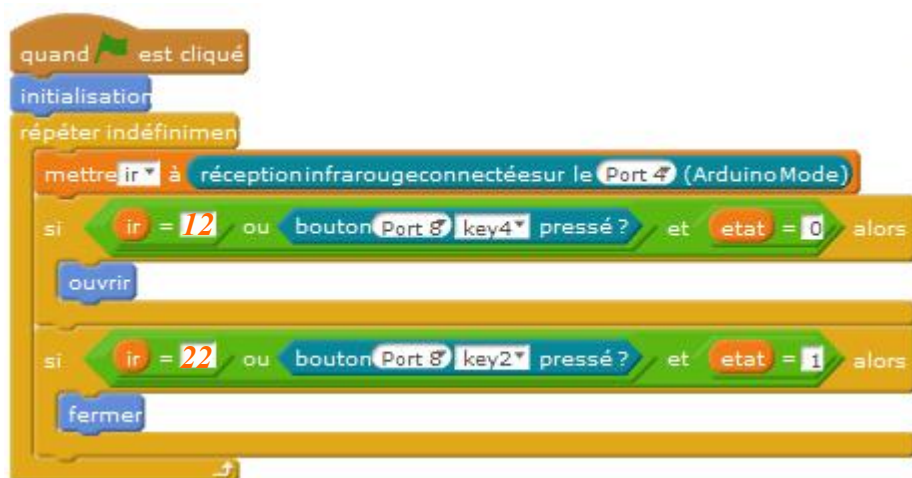
c) Appuyez sur les différentes touches de la télécommande et notez la valeur correspondante reçue par le programme :



Touches	Valeurs	Touches	Valeurs	Touches	Valeurs
A	69	0	22	7	66
B	70	1	12	8	82
C	71	2	24	9	74
D	68	3	94	→	9
E	67	4	8	↓	25
F	13	5	28	↑	64
⚙️	21	6	90	←	7

d) ouvrez le programme : « makeblock-barriere »

e) modifiez ce programme pour que ce soit la touche « 1 » de la télécommande qui commande l'ouverture de la barrière et la touche « 0 » la fermeture :

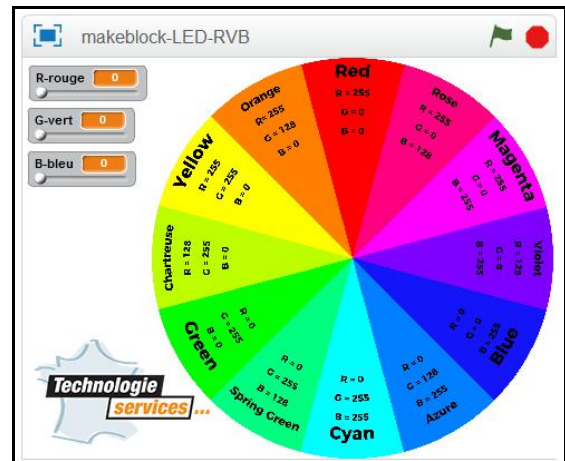


## 5.2. Couleurs du gyrophare

a) Vous devez ouvrir le fichier : « makeblock-LED-RVB »

b) Exécutez le programme en appuyant sur le drapeau vert.

c) Vous devez modifier les valeurs R,G et B et observer si les couleurs du gyrophare de la maquette prennent bien les teintes attendues



Complétez le tableau suivant :

Couleurs attendues	R	G	B	Résultats
Cyan	0	255	255	Bleu clair
Vert	0	255	0	Vert
Jaune	255	255	0	Jaune
Orange	255	128	0	Orange
Rouge	255	0	0	Rouge
Magenta	255	0	255	Rose vif
Bleu	0	0	255	Bleu

d) Indiquez le réglage de R, G et B qui permet d'obtenir une couleur proche du blanc :

*Pour obtenir un blanc il faut le réglage suivant :  $R = G = B = 255$ .*

e) ouvrez le programme : « makeblock-barriere »

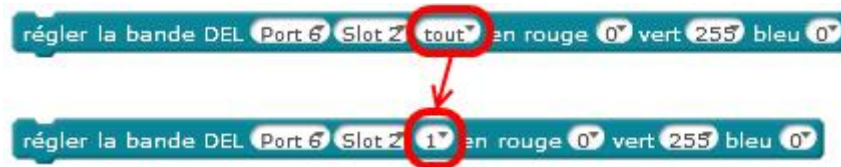
f) Dans le programme initial, la fermeture de la barrière est annoncée par un clignotement vert du gyrophare. Vous devez modifier le bloc suivant pour changer la couleur « verte » en couleur « orange » :





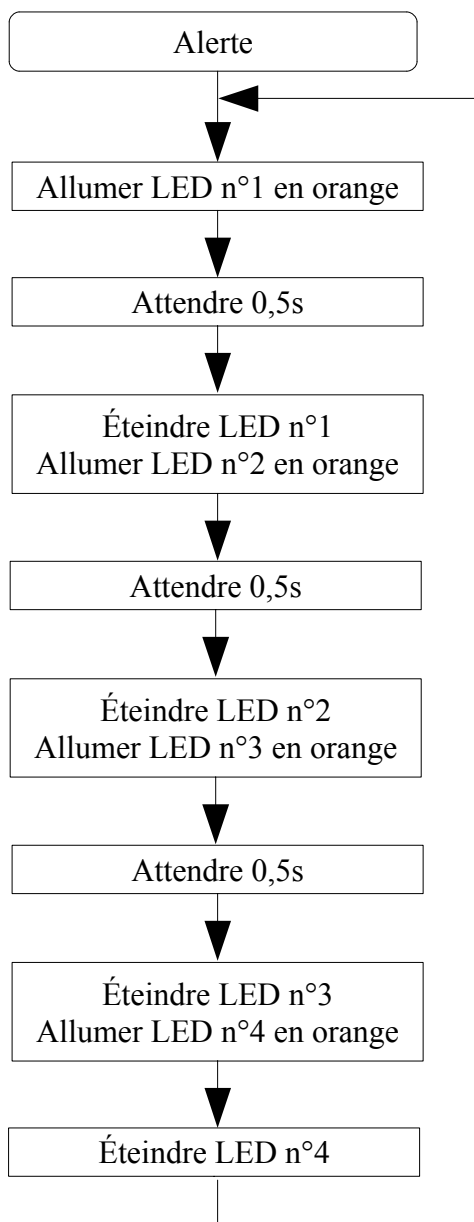
g) Le gyrophare est composé de 4 LED et pour l'instant toutes les LED sont commandées en même temps ce qui ne donne pas une impression de mouvement. Pour donner un effet de rotation au gyrophare il faudrait allumer et éteindre les 4 LED les unes après les autres.

Pour commander une LED à la fois, voici la partie du script à modifier :



*Vous devez remplacer la valeur « tout » par « 1 » pour commander la LED n°1  
ou par « 2 » pour la N°2...*

Nous vous proposons l'organigramme suivant, ça sera à vous de modifier le script initial pour obtenir cette rotation du gyrophare :



X4



## 6. Améliorations (en option)

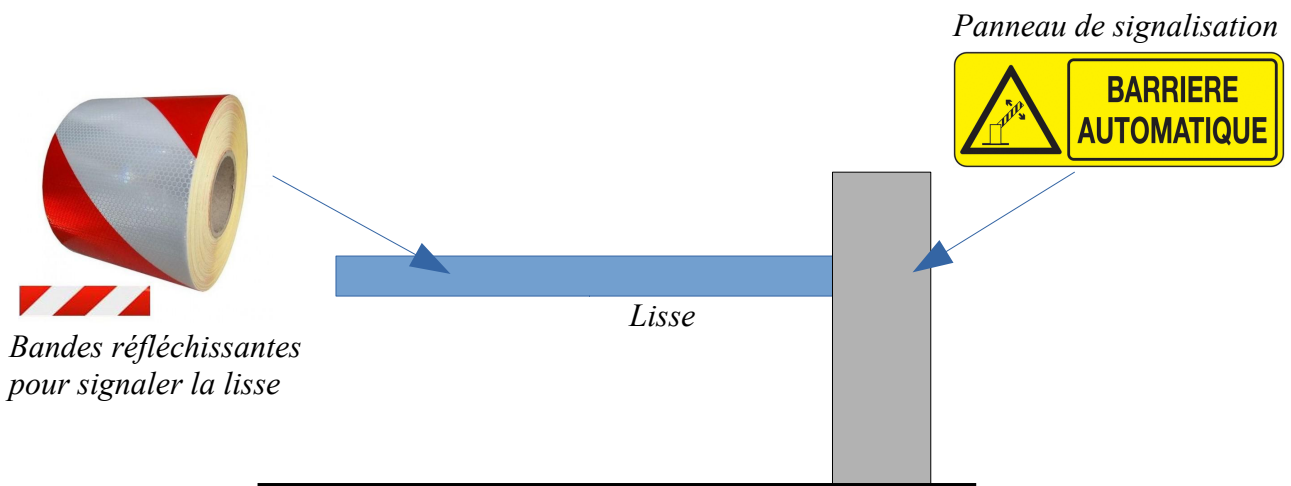
Pour aller plus loin dans cette activité, nous vous proposons d'améliorer cette maquette. En effet, pour que cette barrière respecte les normes de sécurité, vous devez ajouter une signalétique et une sécurité supplémentaire qui fassent remonter la lisse en cas de contact avec un obstacle non détecté par le capteur à ultrason.

### 6.1. Signalétiques

Dans cette partie vous devez concevoir et coller des panneaux de signalisation et des bandes réfléchissantes pour mettre en conformité la barrière.

a) Cette signalétique est la solution technique de quelles fonctions dans le cahier des charges ?

*Il s'agit des fonctions de service FS6 et FS4 du cahier des charges.*



### 6.2. Mise à niveau des sécurités

Actuellement la barrière est capable de détecter la présence d'un gros obstacle (voiture, camionnette) à l'aide du capteur à ultrason mais pour respecter les normes il faut qu'il y ait un capteur directement lié à la lisse capable de détecter des obstacles « légers » (passant, cycliste...).

a) Cette mise à niveau des sécurités est la solution technique de quelles fonctions dans le cahier des charges ?

*Il s'agit des fonctions de service FS3 et FS4 du cahier des charges.*

b) Vous devez regarder le film suivant pour ajouter un capteur de fin de course sur la lisse de la barrière.

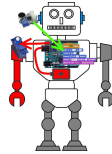


« barriere-makeblock-  
ameloration.mov »

*Liste des pièces :*

- fin de course ref : 276398
- vis makeblock M4 ref : 751263
- pièce à imprimer : « fixation.stl »

## 7.Synthèse



**Automate :**

**Un automate c'est un « robot » !** C'est un système qui réalise des tâches plus au moins complexes à partir d'une consigne en utilisant de l'énergie.

**Capteurs :**



**Les capteurs ce sont les « oreilles » du « robot » !** Ce sont des éléments de l'automate qui lui permettent de connaître son environnement proche à travers des mesures physiques (position, température, niveau sonore...).



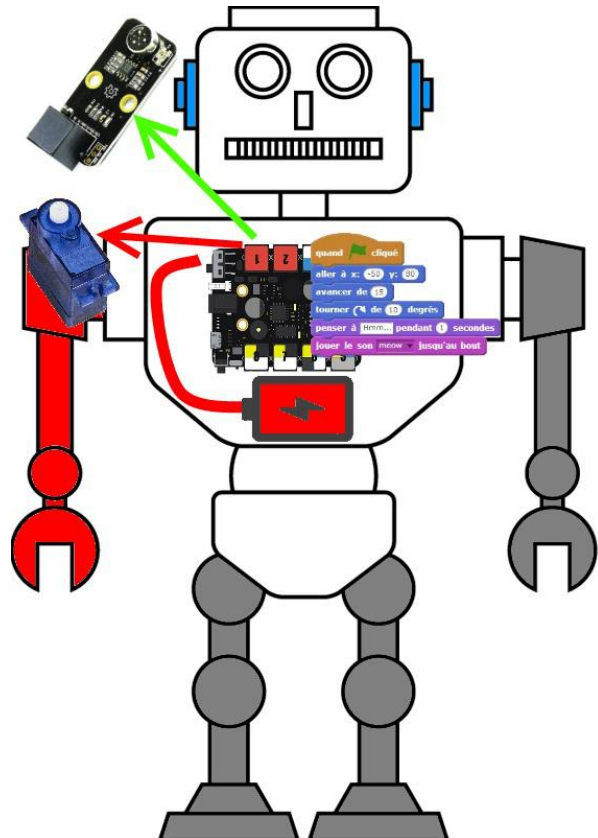
**Actionneurs :**

**Les actionneurs ce sont les « bras » du « robot » !** Ce sont des éléments de l'automate qui lui permettent de réaliser une action à partir de l'énergie qu'il reçoit.

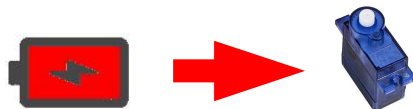


**Programme :**

**Le programme c'est la « liste des choses à faire » du « robot » !** C'est une suite d'instructions qui spécifie étape par étape les opérations à exécuter par l'automate.

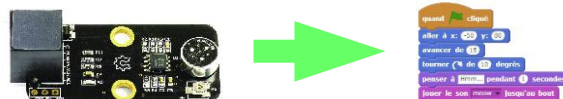


**Chaîne de l'énergie :**



C'est la suite chaînée des éléments de l'automate qui part de la **source d'énergie** pour aller vers les **actionneurs**.

**Chaîne de l'information :**






C'est la suite chaînée des éléments de l'automate qui partent de ses **capteurs** pour aller vers son **programme**.

## 8.Auto-évaluation

Vous devez vous auto-évaluer respectant les indicateurs.

**MI** = Maîtrise Insuffisante, **MF** = Maîtrise Faible, **MS** = Maîtrise Satisfaisante et

**TBM** = Très Bonne Maîtrise

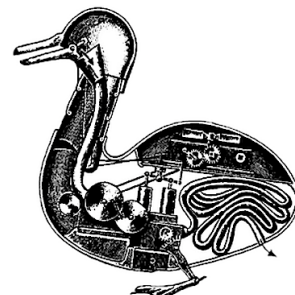
<div>  <b>Domaine 1</b>  Les langages pour penser et communiquer </div>						
N°	Compétences travaillées	Indicateurs	MI	MF	MS	TBM
CT 4.1	Décrire, en utilisant les outils et langages de descriptions adaptés, la structure et le comportement des objets.	<b>TBM</b> = utiliser un vocabulaire technique et être capable d'en donner une définition. <b>MS</b> = utiliser un vocabulaire technique.				
<div>  <b>Domaine 2</b>  Les méthodes et outils pour apprendre </div>						
N°	Compétences travaillées	Indicateurs	MI	MF	MS	TBM
CT 5.2	Organiser, structurer et stocker des ressources numériques.	<b>TBM</b> = dossiers et sous dossiers <b>MS</b> = tous les fichiers dans le même dossier				
CT 5.4	Piloter un système connecté localement ou à distance.	<b>TBM</b> = programmer seul en partant de rien <b>MS</b> = modifier un programme existant				
<div>  <b>Domaine 4</b>  Les systèmes naturels et techniques </div>						
N°	Compétences travaillées	Indicateurs	MI	MF	MS	TBM
CT 1.2	Mesurer des grandeurs de manière directe ou indirecte.	<b>TBM</b> = mesurer une grandeur en utilisant le bon matériel et la bonne unité. <b>MS</b> = mesurer une grandeur en utilisant le bon matériel.				
CS 1.6	Analyser le fonctionnement et la structure d'un objet, identifier les entrées et les sorties.	<b>TBM</b> = identifier complètement les chaînes de l'information d'un automate <b>MS</b> = identifier partiellement les chaînes de l'information d'un automate				
CS 2.3	S'approprier un cahier des charges.	<b>TBM</b> = créer des cahiers des charges <b>MS</b> = compléter le cahier des charges				
CT 2.6	Réaliser, de manière collaborative, le prototype de tout ou partie d'un objet pour valider une solution.	<b>TBM</b> = aide technique apportée au groupe sans intervention du professeur. <b>MS</b> = aide technique avec le soutien du professeur.				
CT 2.7	Imaginer, concevoir et programmer des applications informatiques nomades.	<b>TBM</b> = créer des programmes <b>MS</b> = modifier des programmes				

## 9. Annexe

### 9.1. Automates

En **-380 Archytas de Tarente** a construit un pigeon mécanique en bois capable de tourner sur lui-même suspendu à une barre. Depuis les hommes rêvent de donner une vie à des mécanismes.

Les plus décidés veulent reproduire des êtres vivants, c'est le cas de **Jacques de Vaucanson** qui a reproduit en **1738** un canard mécanique capable de manger, de digérer, de battre des ailes et de caqueter.



Les plus pragmatiques veulent simplifier le travail en automatisant des tâches répétitives. **Joseph Marie Jacquard** a remplacé en **1801** les tisserands par des métiers mécaniques programmables à partir de cartons perforés.

Les plus imaginatifs utilisent le cinéma pour satisfaire leurs fantasmes ou pour mettre en garde ceux qui se flattent d'instrumentaliser la nature. La liste des films qui mettent en scène des créatures vivantes faites de matière plus au moins morte est longue. En tout cas cette liste commence en **1927** par le robot de **Metropolis** par **Fritz Lang**.

Les plus convaincus veulent transformer les maisons en robot géant autonome capable de se défendre, d'économiser son énergie, d'ouvrir ses issues...

**En résumé** les automates sont les membres d'une vieille et grande famille dont la devise est : « agit par toi-même ! »



Sources :

[http://fr.wikipedia.org/wiki/Archytas\\_de\\_Tarente](http://fr.wikipedia.org/wiki/Archytas_de_Tarente)

[http://fr.wikipedia.org/wiki/Jacques\\_de\\_Vaucanson](http://fr.wikipedia.org/wiki/Jacques_de_Vaucanson)

<http://fr.wikipedia.org/wiki/Automate>

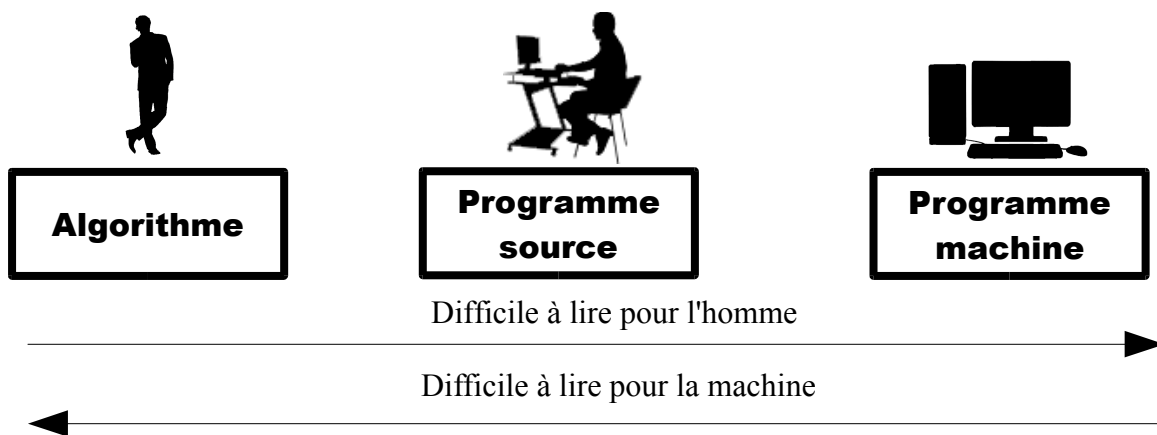
<http://www.syfy.fr/dossier/salade-dautomates>



## 9.2. Programmes informatiques

Un **programme** est une suite d'instructions qui spécifie étape par étape les opérations à exécuter par un ordinateur ou un automate.

Un **langage de programmation** est une notation utilisée pour exprimer des algorithmes et les faire exécuter par un ordinateur ou un automate. Un algorithme est un procédé pour obtenir un résultat par une succession de calculs, décrits sous forme de pictogrammes et de termes simples dans une langue naturelle. Jusqu'en 1950 les programmeurs exprimaient les programmes dans le langage des machines, un langage fait de termes bizarres qui rendait le travail pénible et le résultat sujet à de nombreuses erreurs. Dès 1950 les programmes ont été décrits dans un langage différent - un langage de programmation, ce qui rendait les opérations plus simples à exprimer. Le programme était ensuite traduit automatiquement sous une forme qui permet d'être exécutée par la machine.



**La programmation** est un sujet central en informatique. Un ordinateur ou un automate doit exécuter des instructions exprimées de manière **précise et non ambiguë**. Pour ce faire, les langages de programmation combinent la lisibilité de l'anglais avec l'exactitude des mathématiques. Les programmes sont créés par des programmeurs, ou des ingénieurs logiciels. La création d'un programme comprend une série d'activités telles que la conception, l'écriture, le test et la documentation. En vue d'obtenir un programme de meilleure qualité, le travail de programmation se fait selon une démarche systématique et planifiée.

**En résumé** si les instructions sont lisibles pour nous alors il s'agit d'un algorithme. Si elles sont faciles à comprendre alors il s'agit d'un programme source codé avec un langage de programmation graphique ou écrit. Et enfin s'il s'agit d'une suite de commandes totalement incompréhensibles pour nous alors c'est un programme machine.

**Le langage Sysml** est la frontière entre l'algorithme et le programme source. Il s'agit d'un langage graphique évolué.

Sources :

[http://fr.wikipedia.org/wiki/Programme\\_informatique](http://fr.wikipedia.org/wiki/Programme_informatique)

[http://fr.wikipedia.org/wiki/Carte\\_perfor%C3%A9e](http://fr.wikipedia.org/wiki/Carte_perfor%C3%A9e)

### 9.3. Bug informatique

En informatique, un **bug** (de l'anglais *bug*, « insecte ») est un défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement.

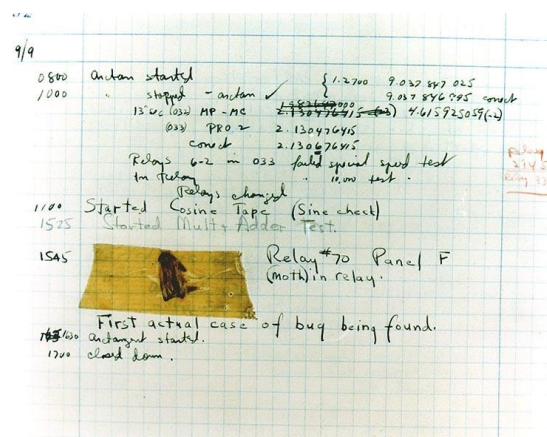
Ce nom vient du tout premier incident informatique qui a été causé par un insecte. La gravité du dysfonctionnement peut aller de bénigne (défauts d'affichage mineurs) à majeure (crash système pouvant entraîner de graves accidents : par exemple, l'explosion du Vol 501 d'Ariane 5).

Un bug peut résider dans un logiciel applicatif, dans les logiciels tiers utilisés par ce logiciel applicatif, voire dans le firmware d'un composant matériel comme ce fut le cas du bug de la division du Pentium3. Un *patch* (terme francisé en « retouche » ou « correctif ») est un morceau de logiciel destiné à corriger un ou plusieurs bugs.

Le mot anglais *bug* (insecte) vient du jargon des ingénieurs de matériel et représentant les problèmes qui y survenaient. L'utilisation du terme pour décrire les défauts de systèmes mécaniques date d'au moins avant les années 1870. Thomas Edison, entre autres, utilisait le mot dans ses notes. Si l'origine précise du mot est donc incertaine, le rapprochement avec les dysfonctionnements dus à la présence d'un insecte dans le système semble évident.

Le terme bug vient du fait que dans les années 1930, les premiers ordinateurs fonctionnaient à l'aide d'amplificateurs à lampe (ancêtres du transistor). Ces ordinateurs prenaient la place d'une pièce entière. En été, la chaleur dégagée par les tubes était telle qu'il fallait ouvrir les fenêtres donc l'intrusion des insectes était inévitable. Ces ordinateurs étaient alors vulnérables à ces insectes volants qui pouvaient se coller à l'un des tubes. Cela provoquait un choc thermique mettant le tube hors service et produisant un mauvais fonctionnement de la machine, d'où le terme *bug* employé par les anglo-saxons.

Le terme est parfois faussement attribué à Grace Hopper : une anecdote raconte qu'elle aurait découvert qu'un insecte (*bug*), coincé entre deux contacts d'un relais, causait le mauvais fonctionnement du Harvard Mark II, l'un des premiers ordinateurs électromécaniques. Son journal d'entretien, conservé à la Smithsonian Institution, contient encore en date du 9 septembre 1947, 15h45, le corps de la mite retiré du relais défectueux, avec l'annotation « premier cas avéré de bug ayant été trouvé ». Cette anecdote a popularisé l'expression *bug* pour représenter les erreurs dans un programme.



**Illustration 1: Photo du « premier cas véritable de découverte de bug », dans le journal d'entretien du Harvard Mark II conservé à la Smithsonian Institution.**

Source :

[https://fr.wikipedia.org/wiki/Bug\\_\(informatique\)](https://fr.wikipedia.org/wiki/Bug_(informatique))

## 9.4. Méthode APTE

La méthode APTE désigne une méthode d'analyse fonctionnelle et d'analyse de la valeur pour la conduite de projets d'innovation ou d'optimisation. Elle est une marque déposée par la société APTE.

### a) But

Cette méthode est tirée des principes de l'analyse de la valeur de Larry Miles. Le but de cette méthode est double : il s'agit à la fois d'accroître la qualité (c'est-à-dire d'aboutir à une meilleure adéquation avec les besoins de l'utilisateur) et de diminuer le coût de ce que l'on étudie. Cette méthode s'applique aussi bien aux produits, aux procédés de fabrication, aux équipements qu'aux organisations.

La Méthode APTE a été formalisée par Bertrand de la Bretesche dans le livre La méthode APTE : Analyse de la valeur, analyse fonctionnelle<sup>1</sup>. Elle apparaît dans les programmes pédagogiques dès le collège et est enseignée dans les écoles d'ingénieur, les écoles de commerce et les filières universitaires<sup>2</sup>. Elle est citée dans de nombreux ouvrages.

### b) Historique

Au sortir de la guerre, face à la hausse du prix de revient des produits industriels et à la part prépondérante que prenaient les achats dans ces derniers, la compagnie General Electric missionna Larry Miles, alors aux achats, pour définir une méthode d'optimisation. Larry Miles rapprocha les coûts d'un produit de la liste des services que l'utilisateur en attend. Il en tira deux principales constatations : premièrement la valeur d'utilisation du produit se pose en fonctions d'usage et fonctions d'estime, deuxièmement 70 à 80 % du coût d'un produit est lié à la façon de rendre le service (solution de conception) et non au service lui-même. À partir de là, Larry Miles développa une nouvelle méthode d'optimisation : l'analyse de la valeur, en anglais value analysis. Plus tard, Miles conseilla d'appliquer sa méthode dès la conception : c'est ce qu'on appelle le value engineering. Aujourd'hui, ces méthodes sont connues sous le nom de VA-VE (Value Analysis - Value Engineering).

Au début des années soixante, Gilbert Barbey, alors consultant en France au sein du cabinet KBWhite, créa la méthode APTE à partir des principes de l'analyse de la valeur. Depuis la création en 1964 du cabinet APTE, la méthode est un ensemble formalisé et cohérent de concepts logiques de raisonnement et d'outils méthodologiques. Elle est devenue en France une méthode pour l'optimisation des produits mais aussi des procédés, des équipements et des organisations.

### c) Les principes de base de la méthode

La Méthode APTE c'est tout d'abord raisonner par rapport aux finalités exprimées indépendamment des solutions, ce qui implique :

- de définir des problèmes en tant qu'objectifs à atteindre.

- l'obtention d'un cadre de réflexion consensuel entre les différents responsables sur les services à rendre.

- l'objectivité qui évite la comparaison entre solutions, en proposant au contraire le jugement d'une solution par rapport à l'objectif.

- et enfin la créativité en rouvrant complètement le champ des choix possibles.

Que ce soit en spécification, conception ou diagnostic, la Méthode APTE établit la distinction entre ce qui est « utile » ou « inutile » en distinguant :

- ce qui participe directement aux finalités (fonctions) : le « juste nécessaire » ;

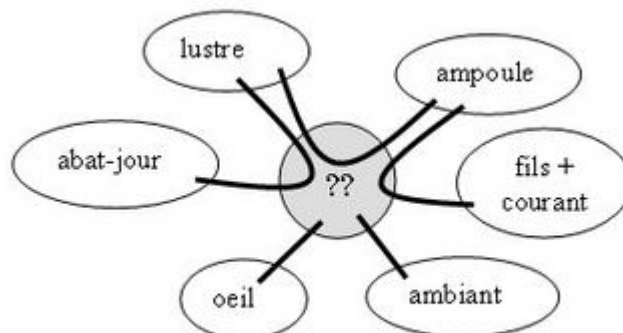
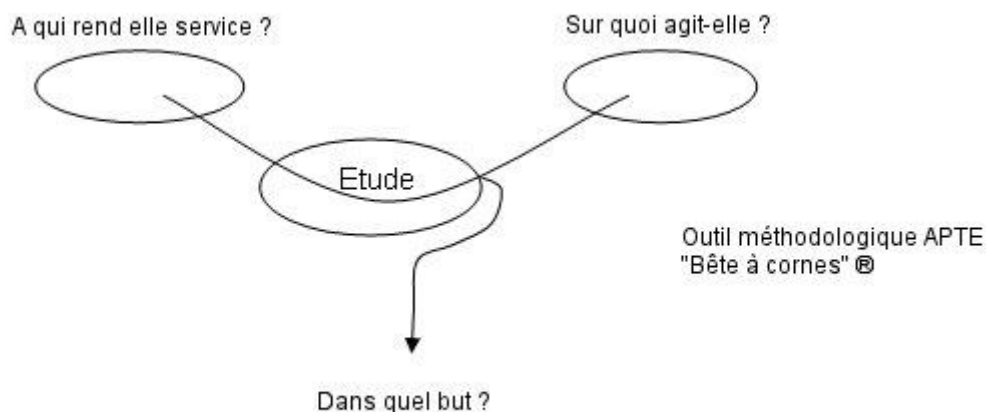
- de ce qui ne dépend que des solutions : la « fonction de conception ».

Il s'agit ensuite de comprendre et d'analyser

- les écarts et les causes de ces écarts de qualité entre les services exprimés et l'objet de l'étude à optimiser (diagnostic valeur)

- les écarts et les causes de ces écarts entre le juste nécessaire à dépenser pour satisfaire les fonctions et les coûts de l'actuelle solution (diagnostic coût)

Il suffit ensuite d'organiser la recherche de la ou des solutions(s) optimum.



Source :

[https://fr.wikipedia.org/wiki/M%C3%A9thode\\_APTE](https://fr.wikipedia.org/wiki/M%C3%A9thode_APTE)