

# Barrière de parking

## Maquette programmable avec mBlock



```
Arduino - générer le code
Oled sur Afficheur vider l'écran
mettre Comptage à 0
Oled sur Afficheur écrire regroupe Valeur = Comptage sur la ligne 1 colonne 0
répéter indéfiniment
si Bouton-poussoir Bouton_Poussoir_A appuyé alors
  ajouter à Comptage 1
  Oled sur Afficheur écrire regroupe Valeur = Comptage sur la ligne 1 colonne 0
  attendre jusqu'à non Bouton-poussoir Bouton_Poussoir_A appuyé
```

# Ressources disponibles pour le projet

Autour du projet, nous vous proposons un ensemble de **ressources téléchargeables gratuitement sur le wiki**.

## Barrière de parking

- Fichiers **3D** (SolidWorks, Edrawings et Parasolid) de la maquette et de ses options.
- Dossier **technique** pour la mise en œuvre de la maquette ;
- Une notice d'utilisation de l'**option Bluetooth** ;

## Logiciels mBlock et App Inventor

- Notice d'installation d'une extension dans mBlock.
- Notice d'utilisation App Inventor 2.

## Activités / Programmation

- Fichiers modèles et fichiers de correction des programmes pour mBlock et AppInventor.

**NOTE** : Certains fichiers sont donnés sous forme de fichier.zip.



**Les documents techniques et pédagogiques signés A4 Technologie sont diffusés librement sous licence Creative Commons BY-NC-SA :**

- **BY** : Toujours citer A4 Technologie comme source (paternité).
- **NC** : Aucune utilisation commerciale ne peut être autorisée sans l'accord préalable de la société A4 Technologie.
- **SA** : La diffusion des documents éventuellement modifiés ou adaptés doit se faire sous le même régime.

**Consulter le site <http://creativecommons.fr/>**

*Note : la duplication de ce dossier est donc autorisée sans limite de quantité au sein des établissements scolaires, aux seules fins pédagogiques, à condition que soit cité le nom de l'éditeur A4 Technologie.*

**Logiciels, programmes, manuels utilisateurs téléchargeables gratuitement sur [www.a4.fr](http://www.a4.fr)**

# SOMMAIRE

<b>Introduction .....</b>	<b>2</b>
Barrière de parking .....	2
Les environnements de programmation graphique .....	2
Le dossier .....	2
Les fiches exercices .....	3
Prérequis .....	3
Tableau d'affectation des entrées et sorties .....	4
<b>Programmation version de base niveau 1 .....</b>	<b>5</b>
<b>Niveau 1 - A.....</b>	<b>6</b>
Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux.....	6
Exercice niveau 1 - A.2: Répéter une action deux fois.....	7
Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment.....	8
<b>Niveau 1 - B.....</b>	<b>9</b>
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur.....	9
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que .....	10
<b>Niveau 1 - C.....</b>	<b>11</b>
Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir.....	11
Exercice niveau 1 - C.2 : Instruction conditionnelle et barrière infrarouge.....	12
Exercice niveau 1 - C.3 : Contrôle moteur ET voyant lumineux.....	13
<b>Niveau 1 - D.....</b>	<b>14</b>
Exercice niveau 1 - D.1 : Utilisation des variables .....	14
Exercice niveau 1 - D.2 : Utiliser et tester une variable.....	15
Exercice niveau 1 - D.3 : Contrôler la valeur d'une variable à l'aide des boutons poussoirs.....	16
<b>Programmation version de base niveau 2 .....</b>	<b>18</b>
Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses .....	19
Exercice niveau 2 - A.3 : Contrôle de l'ouverture et de la fermeture avec bouton et barrière infrarouge .....	21
<b>Programmation version de base niveau 3 (OPTION) .....</b>	<b>22</b>
<b>Option : Module afficheur OLED .....</b>	<b>23</b>
Exercice niveau 3 - A.1 : Afficher un message.....	24
Exercice niveau 3 - A.2 : Afficher une variable.....	25
Exercice niveau 3 - A.3 : Afficher un message en fonction de l'état de la barrière .....	26
<b>Option : Module Bluetooth.....</b>	<b>27</b>
Exercice niveau 3 - C.1 : Ouvrir/fermer avec application Bluetooth.....	29
Exercice niveau 3 - C.2 : Contrôle du portail par Smartphone .....	30
Exercice niveau 3 - C.3 : Envoyer des données vers un Smartphone.....	31
Exercice niveau 3 - C.4 : Envoyer et recevoir des données provenant d'un Smartphone .....	32
<b>Option : Module Capteur RFID.....</b>	<b>33</b>
Exercice niveau 3 – E.1 : Utilisation du module capteur RFID.....	34
Exercice niveau 3 – E.2 : Utilisation du module capteur RFID (2) .....	35

# Introduction

---

## Barrière de parking

La maquette Barrière de Parking (BE-APARK) est une reproduction homothétique d'une barrière de parking automatisée réelle : bouton d'activation, capteurs fin de course, barrière optique, clignotant de sécurité, moteur. Programmable et piloté par les systèmes AutoProgX2 ou AutoProgUno, il permet une activité de programmation complète par rapport aux attendus de fin de cycle collège : l'algorithmique en maths, l'étude de scénarios, la programmation et la mise en œuvre en Technologie.

Vous trouverez dans ce document tout le nécessaire pour démarrer des activités de programmation autour de la barrière :

- La mise en œuvre de la maquette : câblage et configuration des modules.
- Différents scénarios de programmation, du plus simple au plus complexe, avec des exemples de programmes tout faits en langage par blocs.
- Des exercices complémentaires pour les différents modules en option : afficheur, Bluetooth et capteur RFID.

## Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette AutoAlarme ont été réalisés sous **mBlock**.



mBlock est un IDE développé par Makeblock, reprenant la base de Scratch avec l'ajout de blocs permettant le contrôle d'une carte Arduino.

MBlock permet également de créer ses propres blocs dans une extension **A4\_Barriere** (fichier zip), des blocs simples et intuitifs présents permettant de prendre en main la maquette rapidement.



Pour les activités menées avec un smartphone ou une tablette, les programmes et applications ont été réalisés sous **App Inventor 2**.

Il s'agit d'un environnement de développement pour concevoir des applications pour smartphone ou tablette Android. Il a été développé par le MIT pour l'éducation. Il est gratuit et fonctionne via internet avec MBlock.

## Le dossier

Ce document propose un parcours progressif pour découvrir et se perfectionner avec la programmation en se basant sur une série d'exemples ludiques autour de la maquette grâce à ses capteurs et actionneurs. Il est organisé en fonction des niveaux de programmation.

### Niveau 1 :

Découverte progressive du jeu d'instructions et des fonctionnalités de base de la maquette et maîtrise des principes fondamentaux pour concevoir un programme : séquences, boucles, structures conditionnelles (test) et variables.

### Niveau 2 :

Approfondissement des principes de programmation abordés dans le niveau 1 en concevant des programmes plus élaborés qui répondent à des cas concrets d'utilisation de la maquette (version de base).

### Niveau 3 :

Exemples d'utilisation des différentes options proposées : afficheur, Bluetooth et capteur RFID.

# Les fiches exercices

Pour chaque niveau de programmation, nous vous proposons des fiches exercices avec :

- un objectif : ce que doit faire le programme ;
- un fichier de correction qui propose un exemple de programme réalisé sous mBlock (extension .sb2).

Deux approches :

- Avec les exemples de programmes, les utilisateurs découvrent les principes de la programmation graphique en blocs : chargement d'un programme, modification d'un programme et vérification sur le matériel (ex : modification des temps d'attente, etc.).
- Les utilisateurs conçoivent eux-mêmes le programme pour atteindre l'objectif proposé, en organigrammes ou en blocs (à partir du fichier modèle). Ils peuvent ensuite le comparer au fichier de correction.

Principe de nommage des fichiers :

- **BP** pour Barrière de parking
- **N** : niveau de programmation 1-2-3
- **A-B-C** : jeu d'instructions du plus simple au plus avancé

Exemple : VR\_N3\_A1.sb2

Correspond au niveau 3 avec le jeu d'instructions A, adapté aux objectifs « avancés » de ce niveau.

## Prérequis

Pour la version de base :

- Installer le logiciel **mBlock**.
- Installer l'extension **A4\_Barriere** (fichier zip) dans mBlock.
- **Maquette** Barrière de parking (Réf. BE-APARK).
- **Câble de programmation** USB (Réf : CABL-IMPUSB).
- **Interface programmable** AutoProgUno (Réf. K-AP-UNO).
- **Cordons de liaison** jack compatibles AutoProg pour établir les liaisons entre l'interface programmable et la maquette.

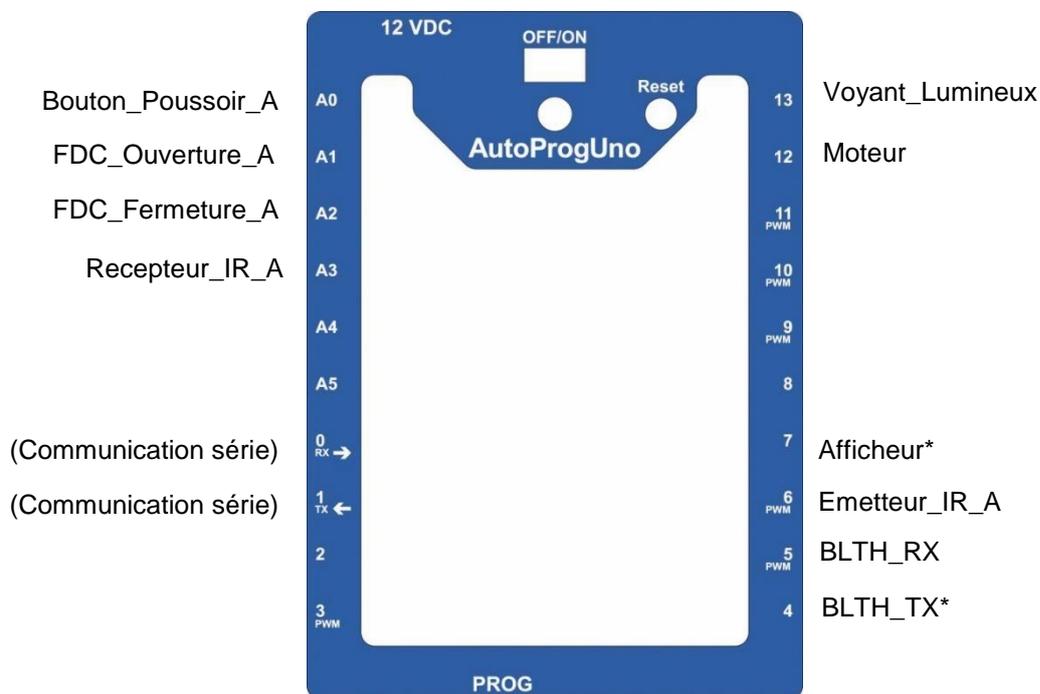
Pour l'option Bluetooth :

- **Tablette ou smartphone** Android 5 ou + équipés de Bluetooth V3.
- Connexion internet pour accéder à **App Inventor** : <http://ai2.appinventor.mit.edu/>
- Compte Gmail requis.

**Le guide de montage ainsi que les caractéristiques techniques des composants sont détaillés dans le dossier technique disponible sur le wiki.**

## Tableau d'affectation des entrées et sorties

AutoProgUno	Maquette	Nom mBlock
<b>MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES</b>		
2		
3		
4	Module Bluetooth (option)	BLTH_TX*
5	Module Bluetooth (option)	BLTH_RX
<b>MODULES ACTIONNEURS POUR SORTIES NUMÉRIQUES</b>		
6	Emetteur infrarouge A	Emetteur_IR_A
7	Module afficheur OLED	Afficheur*
8		
9		
10		
11		
12	Module moteur	Moteur
13	Module lumineux jaune	Voyant_Lumineux
<b>MODULE DE COMMUNICATION</b>		
0	(communication avec ordinateur)	
1		
<b>ENTRÉES / SORTIES LIBRES (A pour les analogiques)</b>		
A0	Bouton poussoir A / Capteur RFID	Bouton_Poussoir_A
A1	Fin de course ouverture A	FDC_Ouverture_A
A2	Fin de course fermeture A	FDC_Fermeture_A
A3	Recepteur infrarouge A	Recepteur_IR_A
A4		
A5		



# Programmation version de base niveau 1

## Objectifs :

- Découvrir et maîtriser le matériel avec des exemples très simples pour débiter en programmation.
- Appréhender les différentes fonctionnalités du matériel.

Ce niveau permet de découvrir toutes les fonctionnalités de base du volet automatique, en apprenant les structures de base de la programmation. Et en particulier celles demandées dans les nouveaux programmes : séquences, boucles, structures conditionnelles et enfin les variables.

Nom du fichier	Description	Objectif
<b>Niveau 1 A</b>		
BP_N1_A1.sb2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordé : -Allumage/extinction du voyant lumineux Notions de programmation abordées : -séquence d'instructions -temps d'attente -boucle infinie
BP_N1_A2.sb2	Répéter cette même action deux fois.	
BP_N1_A3.sb2	Répéter cette action à l'infini.	
<b>Niveau 1 B</b>		
BP_N1_B1.sb2	Activer le moteur pendant 5 secondes.	Fonctionnalité matérielle abordé : -Gestion du moteur -Utilisation de Bouton-poussoir Notions de programmation abordées : -boucle qui dépend d'une entrée
BP_N1_B2.sb2	Activer le moteur jusqu'à l'appui d'un bouton poussoir.	
<b>Niveau 1 C</b>		
BP_N1_C1.sb2	Allumer le voyant lumineux à l'appui du BP.	Fonctionnalité matérielle abordé : -Gestion des modules infra-rouge -Utilisation de Bouton-poussoir Notions de programmation abordées : -Le test d'une entrée (si/sinon)
BP_N1_C2.sb2	Activer le voyant lumineux lorsque la barrière infrarouge est franchie.	
BP_N1_C3.sb2	Contrôler le moteur avec le bouton poussoir et allumer le voyant sur le franchissement de la barrière infrarouge.	
<b>Niveau 1 D</b>		
BP_N1_D1.sb2	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).	Notions de programmation abordées : -Définition de variable -Incrémement de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage
BP_N1_D2.sb2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	
BP_N1_D3.sb2	Incrémenter une variable à l'appui d'un bouton poussoir, la décrémenter à l'appui de l'autre bouton poussoir.	
BP_N1_D4.sb2	Incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.	

# Niveau 1 - A

## Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux

Fichier modèle : BP\_BASE.sb2

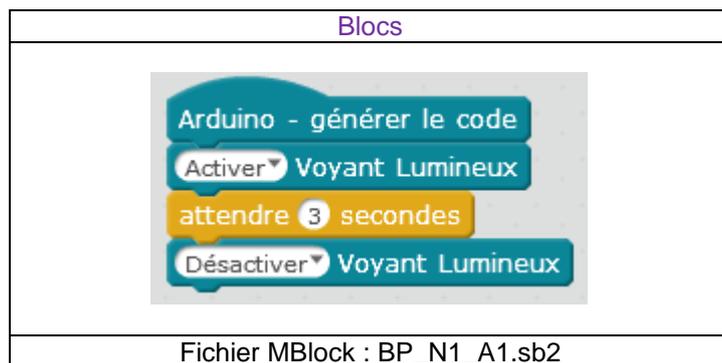
**Objectif** : allumer le voyant lumineux pendant 3 secondes puis l'éteindre.

**Notions abordées** : séquence d'instructions, activation / désactivation d'une sortie, temps d'attente.

**Instruction utilisées** :



**Correction** :



**Remarque** : avec le langage de programmation par blocs la dernière instruction exécutée marque la fin du programme.

## Exercice niveau 1 - A.2: Répéter une action deux fois

**Objectif** : allumer le voyant lumineux pendant 3 secondes puis l'éteindre, recommencer.

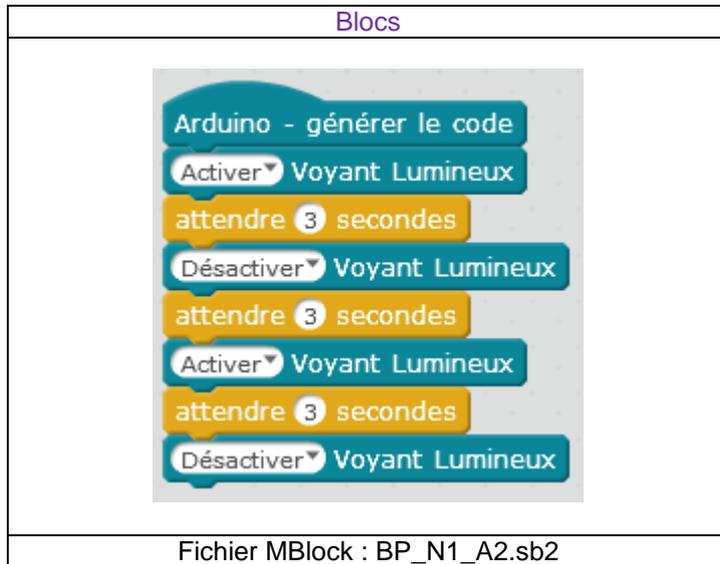
**Notions abordées** : séquence d'instructions, activation / désactivation d'une sortie, temps d'attente.

**Instruction utilisées** :



**Correction** :

Blocs

Le code de correction est une séquence de blocs mBlock :

- Arduino - générer le code
- Activer (menu déroulant) Voyant Lumineux (menu déroulant)
- attendre 3 secondes
- Désactiver (menu déroulant) Voyant Lumineux (menu déroulant)
- attendre 3 secondes
- Activer (menu déroulant) Voyant Lumineux (menu déroulant)
- attendre 3 secondes
- Désactiver (menu déroulant) Voyant Lumineux (menu déroulant)

Fichier MBlock : BP\_N1\_A2.sb2

# Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment

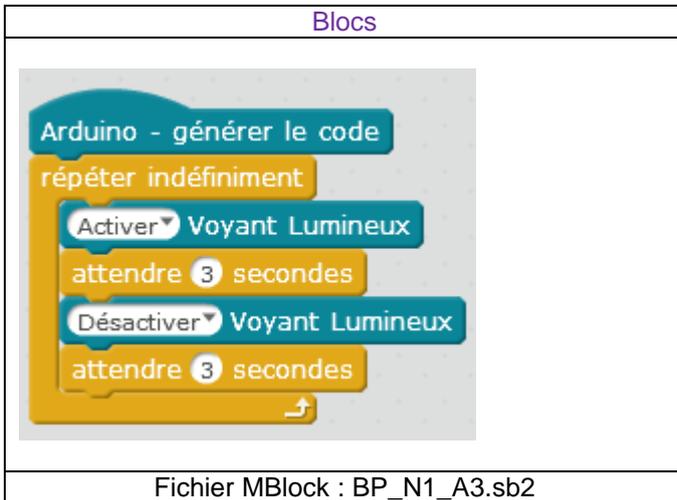
**Objectif** : faire clignoter le voyant lumineux avec une période de 6 secondes indéfiniment.

**Notion abordée** : la boucle infinie.

**Instructions utilisées** :



**Correction** :



**Remarque** : le programme ne peut s'arrêter lorsqu'il est dans une boucle infinie. Le seul moyen de sortir de la boucle est de faire un Reset ou d'éteindre et rallumer le boîtier AutoProg.

# Niveau 1 - B

---

## Exercice niveau 1 - B.1 : Maitriser la rotation du moteur

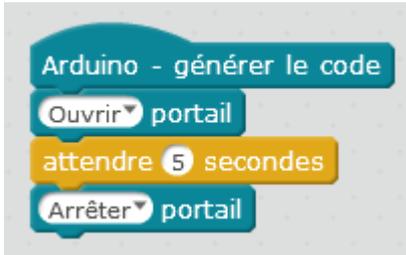
**Objectif** : activer le moteur pendant 5 secondes

**Notion abordée** : utilisation d'un moteur.

**Instructions utilisées** :



**Correction** :

Blocs

Fichier MBlock : BP_N1_B1.sb2

## Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que

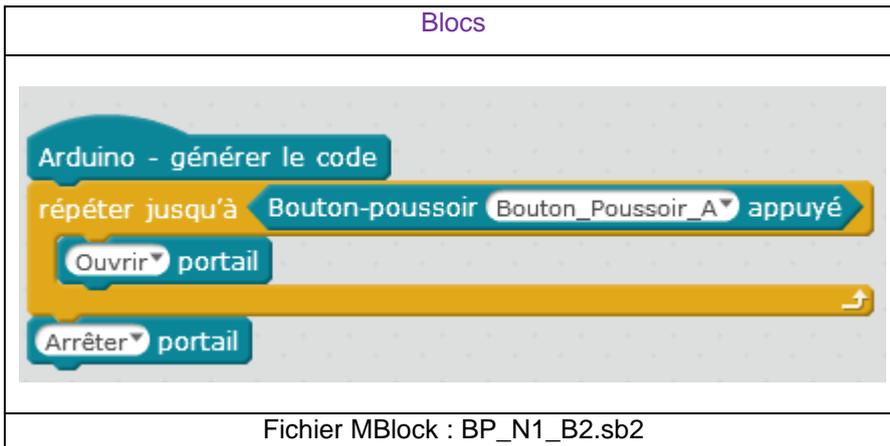
**Objectif :** Activer le moteur jusqu'à l'appui d'un bouton poussoir

**Notion abordée :** exécuter une boucle qui dépend de l'état d'une entrée.

**Instructions utilisées :**



**Correction :**



**Remarque :** Le programme ne peut sortir de la boucle qu'une fois le test sur le bouton-poussoir validé. Le test sur le bouton poussoir se fait qu'une seule fois en début de séquence, avant de commencer l'ouverture. Si un appui est effectué pendant la séquence, aucun effet n'aura lieu sur le programme. Afin de vérifier à tout moment le changement d'état d'une entrée dans une séquence, l'utilisation des interruptions est indispensable (voir ex sur interruption).

Si vous possédez un capteur RFID à la place du bouton poussoir, veuillez-vous référencer dans un premier temps à l'exercice 1 de l'option Module Capteur RFID en cliquant [ici](#)

# Niveau 1 - C

## Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir

**Objectif** : allumer le voyant lumineux à l'appui du BP.

**Notion abordée** : utilisation des commandes conditionnelles (si/sinon).

**Instructions utilisées** :



**Correction** :

Blocs

Fichier MBlock : BP\_N1\_C1.sb2

**Remarque** : les blocs de couleur bleu claires représente des commandes concernant l'utilisation des entrées.

# Exercice niveau 1 - C.2 : Instruction conditionnelle et barrière infrarouge

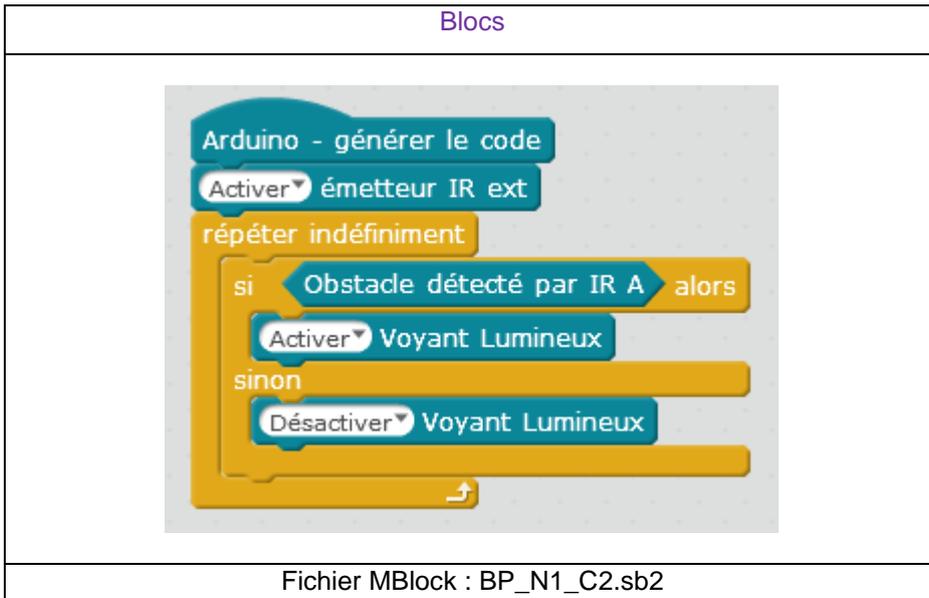
**Objectif** : activer le voyant lumineux lorsque la barrière infrarouge est franchie.

**Notions abordées** : utilisation des commandes conditionnelles (si/sinon) / utilisation d'une barrière infrarouge.

**Instructions utilisées** :



**Correction** :



**Remarque** : l'entrée du récepteur IR est activée d'origine et se désactive lors de la réception du signal de l'émetteur IR.

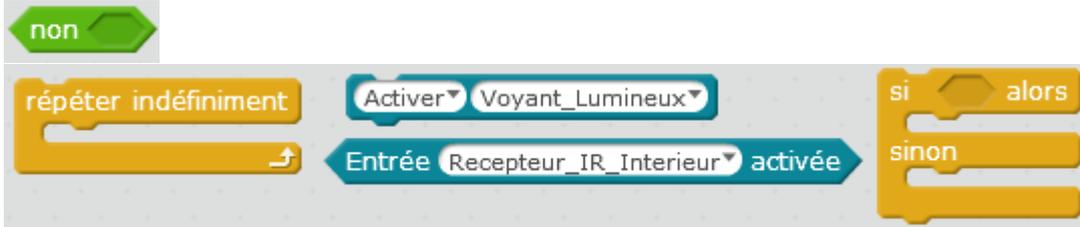
Lorsque un obstacle franchi la barrière IR, le signal n'est plus transmis et l'entrée du récepteur IR devient active. Si le programme ne marche pas, mettre le cavalier MODE de l'émetteur IR sur B et le cavalier du code sur 127.

## Exercice niveau 1 - C.3 : Contrôle moteur ET voyant lumineux

**Objectif :** contrôler le voyant avec le bouton poussoir et allumer le moteur sur le franchissement de la barrière infrarouge.

**Notion abordée :** utilisation des commandes conditionnelles.

**Instructions utilisées :**



**Correction :**

Blocs

Correction complète du programme mBlock. Le code commence par 'Arduino - générer le code' et 'Activer émetteur IR ext'. Il est suivi d'un bloc 'répéter indéfiniment' qui contient deux conditions 'si'. La première condition 'si' vérifie 'Obstacle détecté par IR A' ; si vrai, elle exécute 'Ouvrir portail', sinon 'Arrêter portail'. La seconde condition 'si' vérifie 'Bouton-poussoir Bouton\_Poussoir\_A appuyé' ; si vrai, elle exécute 'Activer Voyant Lumineux', sinon 'Désactiver Voyant Lumineux'.

Fichier MBlock : BP\_N1\_C3.sb2

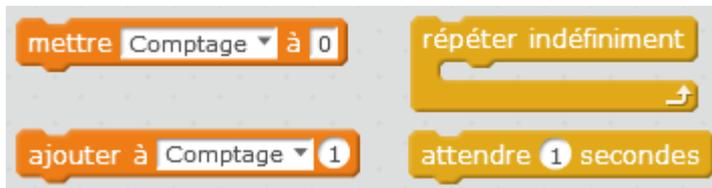
# Niveau 1 - D

## Exercice niveau 1 - D.1 : Utilisation des variables

**Objectif** : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC.

**Notions abordées** : la variable : définition et incrémentation

**Instructions utilisées** :



**Correction** :

Blocs

Fichier MBlock : BP_N1_D1.sb2

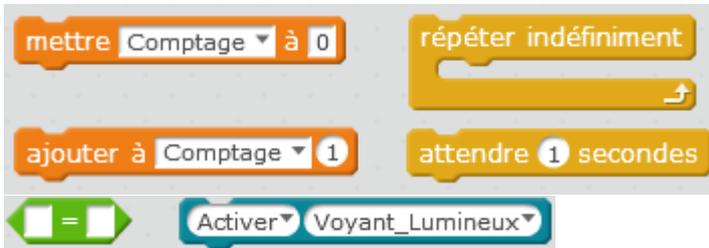
**Remarques** : la commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

## Exercice niveau 1 - D.2 : Utiliser et tester une variable

**Objectif** : incrémenter une variable au cours du temps. Lorsque la variable est supérieure à 10, activer le voyant.

**Notion abordée** : boucle tant que dépendant d'une variable

**Instructions utilisées** :



**Correction** :

Blocs

The image shows a corrected code sequence within a 'Blocs' container. It starts with a blue 'Arduino - générer le code' block, followed by an orange 'mettre Comptage à 0' block, a yellow 'répéter jusqu'à Comptage = 10' loop block, a yellow 'attendre 1 secondes' block, an orange 'ajouter à Comptage 1' block, and finally a blue 'Activer Voyant Lumineux' block.

Fichier MBlock : BP\_N1\_D2.sb2

**Remarque** : cet exercice peut être utilisé comme un minuteur.

# Exercice niveau 1 - D.3 : Contrôler la valeur d'une variable à l'aide des boutons poussoirs

**Objectif :** incrémenter une variable à l'appui d'un bouton poussoir

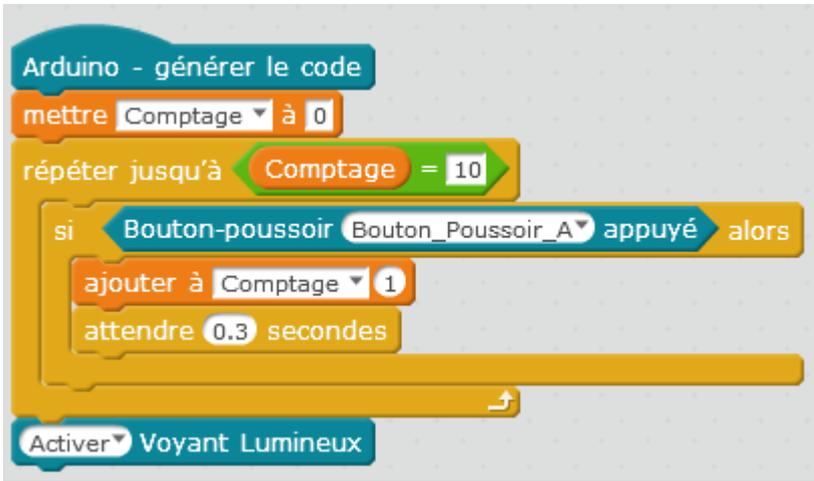
**Notions abordées :** test sur entrées et incrémentation/décrémentation contrôlée d'une variable

**Instructions utilisées :**



**Correction :**

Blocs



Fichier MBlock : BP\_N1\_D3.sb2

**Remarques :** deux tests sont insérés l'un après l'autre. La vitesse d'exécution du programme donne l'impression que les commandes sont exécutées en même temps.

La commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

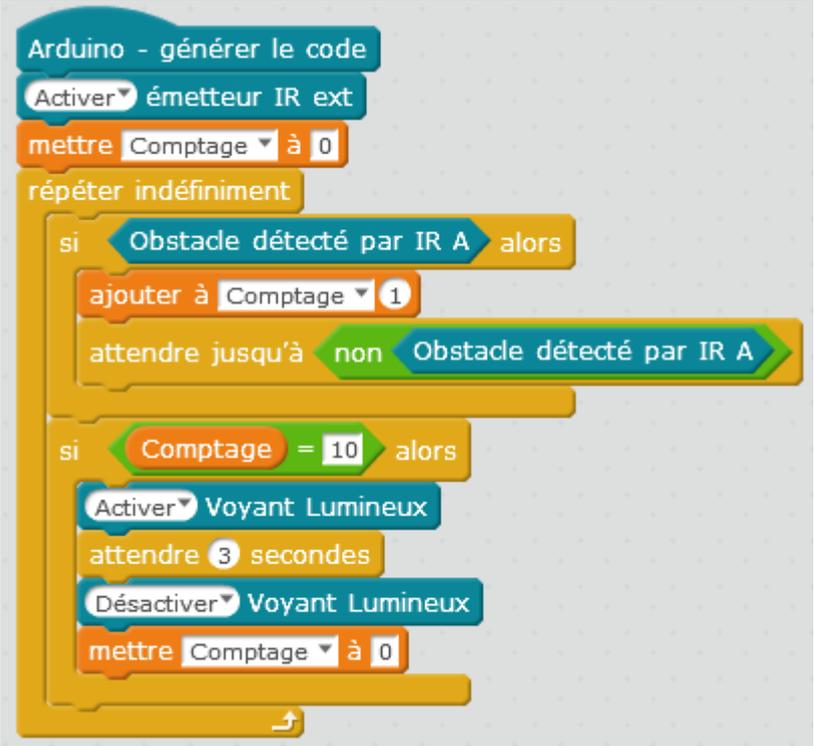
## Exercice niveau 1 - D.4 : Tests /variables/ modules IR

**Objectif** : incrémenter une variable à chaque passage sur la barrière IR. Lorsque le compteur arrive à 10, activer le voyant lumineux 3 secondes et remettre la variable à zéro

**Notion abordée** : test dépendant d'une variable

**Correction** :

Blocs



The screenshot shows the following code blocks in mBlock:

- Arduino - générer le code
- Activer émetteur IR ext
- mettre Comptage à 0
- répéter indéfiniment
  - si Obstacle détecté par IR A alors
    - ajouter à Comptage 1
    - attendre jusqu'à non Obstacle détecté par IR A
  - si Comptage = 10 alors
    - Activer Voyant Lumineux
    - attendre 3 secondes
    - Désactiver Voyant Lumineux
    - mettre Comptage à 0

Fichier MBlock : BP\_N1\_D4.sb2

# Programmation version de base niveau 2

## Objectifs :

- Utilisation concrète de la maquette.
- Utilisation de tous les modules.
- Appréhension des différentes fonctionnalités du matériel ainsi que certaines notions de sécurité.

Ce niveau permet de mettre en œuvre la maquette, au fur et à mesure des exercices vous allez utiliser de plus en plus de modules et enrichir votre code pour obtenir à la fin du niveau une maquette qui marche parfaitement et qui respecte une logique de fonctionnement calquée sur le réel.

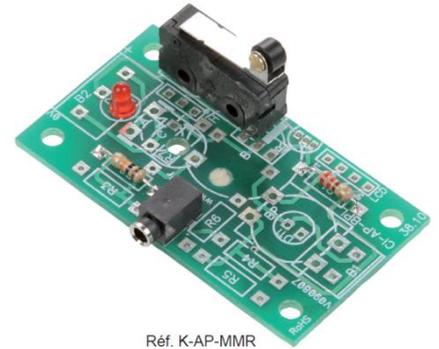
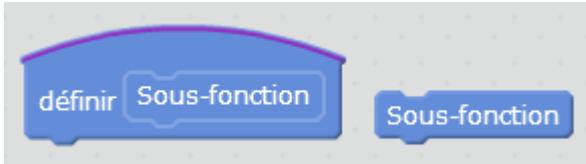
Nom du fichier	Description	Objectif
<b>Niveau 2 A</b>		
BP_N2_A1.sb2	Allumer le moteur et éteindre avec 2 secondes d'attente sur chaque capteur fin de course. Le voyant doit d'allumer lorsque le moteur tourne. Utiliser des sous-fonctions.	Notions de programmation abordées : -Utilisation des fins de course
BP_N2_A2.sb2	Ouvrir ou fermer la barrière sur l'appui du bouton poussoir.	
BP_N2_A3.sb2	Ouvrir la barrière sur l'appui d'un bouton poussoir ou sur le franchissement de la barrière infrarouge, puis fermer la barrière après 5 secondes.	

## Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses

**Objectif :** allumer le moteur et éteindre avec 2 secondes d'attente sur chaque capteur fin de course. Le voyant doit d'allumer lorsque le moteur tourne. Utiliser des sous-fonctions

**Notions abordées :** utilisation des fins de course, procédures (sous-fonctions)

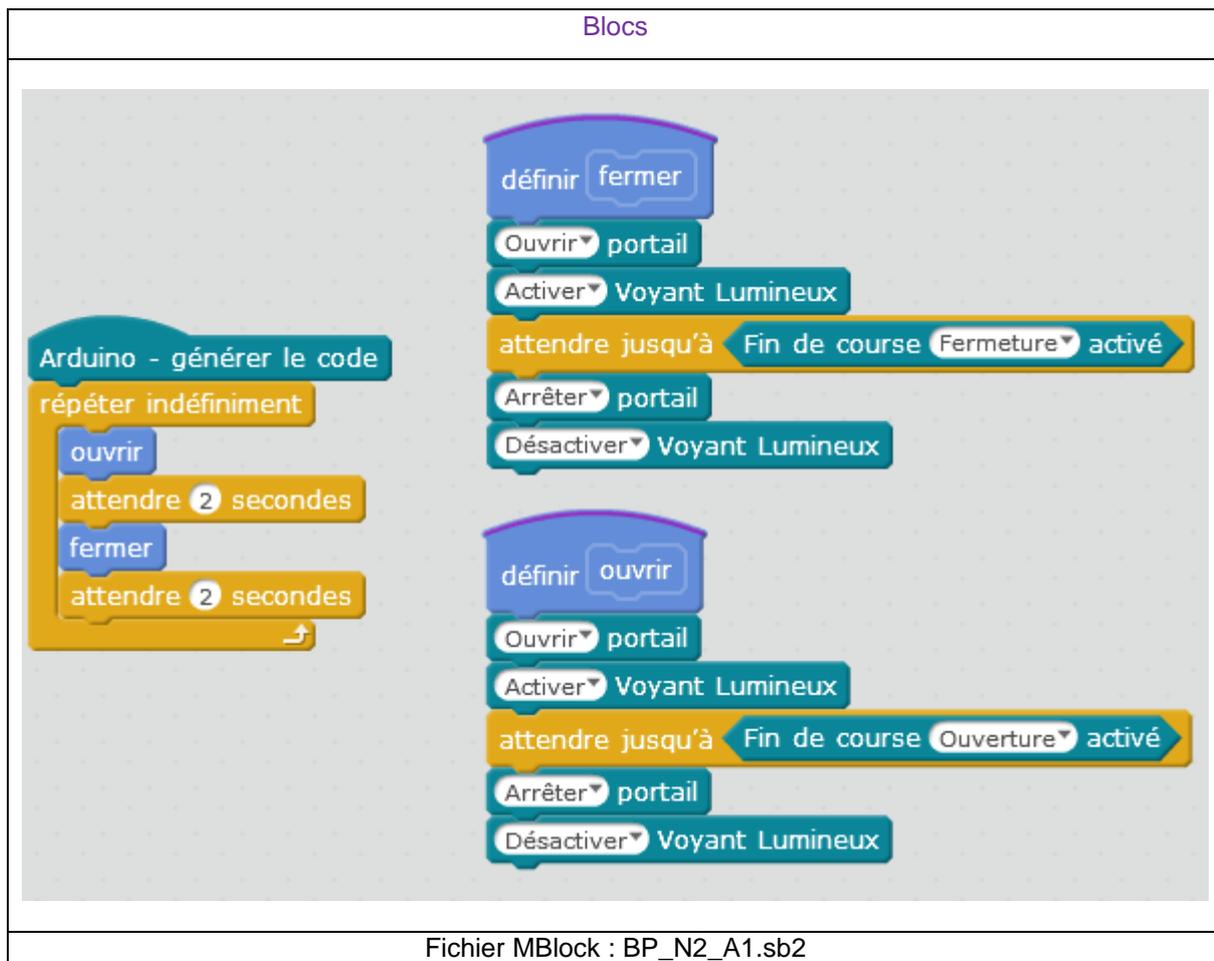
**Instructions utilisées :**



Réf. K-AP-MMR

**Correction :**

Blocs



```
Arduino - générer le code
répéter indéfiniment
  ouvrir
  attendre 2 secondes
  fermer
  attendre 2 secondes

définir fermer
  Ouvrir portail
  Activer Voyant Lumineux
  attendre jusqu'à Fin de course Fermeture activé
  Arrêter portail
  Désactiver Voyant Lumineux

définir ouvrir
  Ouvrir portail
  Activer Voyant Lumineux
  attendre jusqu'à Fin de course Ouverture activé
  Arrêter portail
  Désactiver Voyant Lumineux
```

Fichier MBlock : BP\_N2\_A1.sb2

## Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture

**Objectif** : ouvrir ou fermer la barrière sur l'appui du bouton poussoir

**Notions abordées** :

**Correction** :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  attendre jusqu'à Bouton-poussoir Bouton_Poussoir_A appuyé
  ouvrir
  attendre jusqu'à Bouton-poussoir Bouton_Poussoir_A appuyé
  fermer

définir fermer
  Ouvrir portail
  Activer Voyant Lumineux
  attendre jusqu'à Fin de course Fermeture activé
  Arrêter portail
  Désactiver Voyant Lumineux

définir ouvrir
  Ouvrir portail
  Activer Voyant Lumineux
  attendre jusqu'à Fin de course Ouverture activé
  Arrêter portail
  Désactiver Voyant Lumineux
```

Fichier MBlock : BP\_N2\_A3.sb2

## Exercice niveau 2 - A.3 : Contrôle de l'ouverture et de la fermeture avec bouton et barrière infrarouge

**Objectif** : ouvrir la barrière sur l'appui d'un bouton poussoir, la laisser ouverte sur le franchissement de la barrière infrarouge, puis fermer la barrière après 3 secondes sans qu'il n'y ait d'obstacle sur la barrière

**Notions abordées** :

**Correction** :

Blocs

```
Arduino - générer le code
Activer émetteur IR ext
répéter indéfiniment
  attendre jusqu'à Bouton-poussoir Bouton_Poussoir_A appuyé
  ouvrir
  attendre jusqu'à non Obstacle détecté par IR A
  attendre 3 secondes
  fermer

définir fermer
Ouvrir portail
Activer Voyant Lumineux
attendre jusqu'à Fin de course Fermeture activé
Arrêter portail
Désactiver Voyant Lumineux

définir ouvrir
Ouvrir portail
Activer Voyant Lumineux
attendre jusqu'à Fin de course Ouverture activé
Arrêter portail
Désactiver Voyant Lumineux
```

Fichier MBlock : BP\_N2\_A3.sb2

# Programmation version de base niveau 3 (OPTION)

Objectif : Utiliser les modules plus complexes : afficheur, pilotage à distance, capteur RFID...

Le niveau 3 n'intègre pas de nouvelles notions de programmation mais de nouveaux blocs permettant d'utiliser les modules options.

Nom du fichier	Description	Objectif
<b>Niveau 3 A : Afficheur OLED</b>		
BP_N3_A1.sb2	Afficher un message sur les deux lignes de l'afficheur OLED.	Fonctionnalité matérielle abordée : Utilisation d'un afficheur OLED  Notions de programmation abordées : Utilisation d'un block dédié à l'affichage sur OLED
BP_N3_A2.sb2	Afficher une variable qui changera à chaque incrémentation.	
BP_N3_A3.sb2	Reprendre le programme <b>BP_N2_A4</b> et afficher l'état de la barrière (en mouvement ou à l'arrêt) sur l'afficheur OLED.	
<b>Niveau 3 C : Capteur Bluetooth</b>		
BP_N3_C1.sb2	Contrôler l'ouverture et la fermeture à l'aide de 2 boutons sur application Android.	Fonctionnalité matérielle abordée : - module Bluetooth  Notions de programmation abordées : liaison série (hserin/hserout)
BP_N3_C2.sb2	Contrôler l'ouverture et la fermeture à l'aide d'un seul bouton sur application Android.	
BP_N3_C3.sb2	Jouer une sonnerie sur Smartphone à l'appui d'un bouton poussoir.	
BP_N3_C4	Lors de l'appui sur un bouton de la maquette, envoyer une information au Smartphone qui devra renvoyer une information à la maquette.	
<b>Niveau 3 E : Capteur RFID</b>		
BP_N3_E1.sb2	Basculer le voyant lumineux lors de la détection d'une carte sur le capteur RFID.	Fonctionnalité matérielle abordée : Utilisation d'un capteur magnétique RFID
BP_N3_E2.sb2	Ouvrir puis fermer la barrière lors du déclenchement du capteur RFID.	

# Option : Module afficheur OLED



Afficheur OLED 2 lignes (ref : RAX 133Y)

Ces afficheurs permettent à son utilisateur de rentrer un texte dans un programme qui sera chargé d'afficher ce texte.

L'afficheur utilisé pour cette maquette possède 2 lignes sur lesquelles il est possible d'insérer au maximum 16 caractères (espaces compris)

Trois blocs sont déjà disponibles sur PICAXE Editor afin de simplifier l'utilisation de ce module :



Il est nécessaire d'effacer le texte déjà présent sur l'afficheur lorsqu'on veut y mettre un autre message, car celui-ci garde en mémoire chaque caractère jusqu'à ce que tout soit effacé. Ce qui signifie que si nous écrivons sur 5 caractères là où il y en avait déjà 10 avant la programmation, nous verrons nos 5 caractères + les 5 derniers déjà présents auparavant, ou alors le message d'affichera plus loin qu'au début de l'afficheur.

La commande positionner le curseur à la ligne 1 signifie simplement que nous écrivons sur la première ligne. Nous pouvons également choisir la deuxième ligne sur cet afficheur.

Ensuite il vous suffit seulement d'afficher le message souhaité, tant qu'il ne dépasse pas 16 caractères.

## Exercice niveau 3 - A.1 : Afficher un message

**Objectif :** Afficher un message utilisant les deux lignes sur l'afficheur OLED

**Correction :**

Blocs



```
Arduino - générer le code
Oled sur Afficheur vider l'écran
Oled sur Afficheur écrire hello world 1 sur la ligne 1 colonne 0
Oled sur Afficheur écrire hello world 2 sur la ligne 2 colonne 0
```

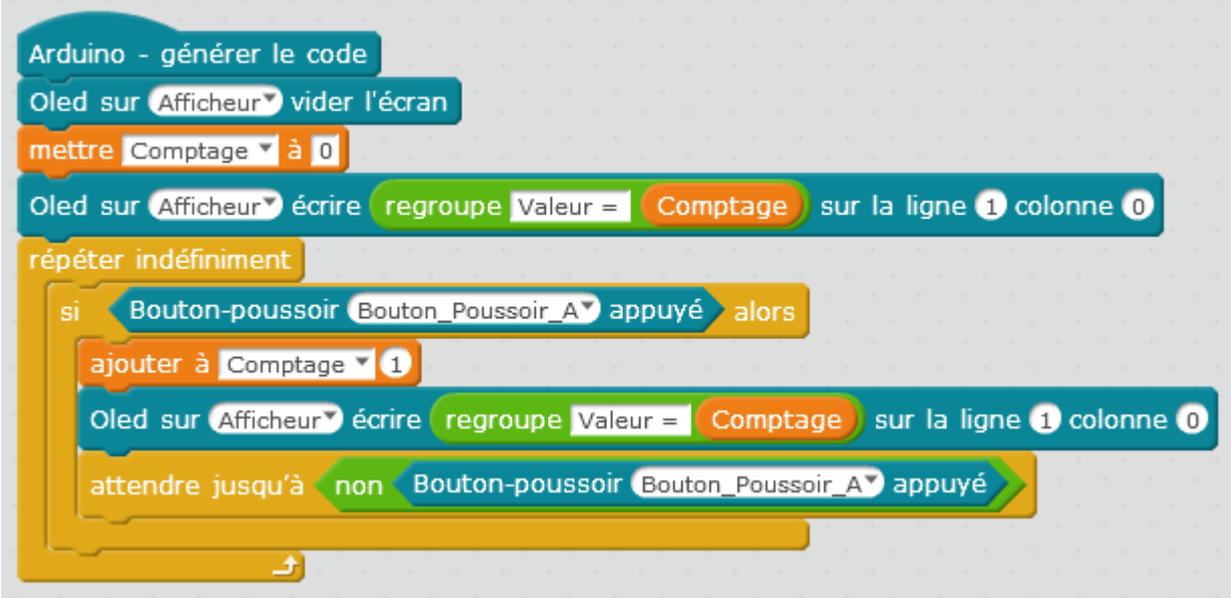
Fichier MBlock : BP\_N3\_A1.sb2

## Exercice niveau 3 - A.2 : Afficher une variable

**Objectif :** Afficher un message sur l'afficheur avec une variable. La variable augmente à chaque appui sur le bouton poussoir.

**Correction :**

Blocs



```
Arduino - générer le code
Oled sur Afficheur vider l'écran
mettre Comptage à 0
Oled sur Afficheur écrire regroupe Valeur = Comptage sur la ligne 1 colonne 0
répéter indéfiniment
  si Bouton-poussoir Bouton_Poussoir_A appuyé alors
    ajouter à Comptage 1
    Oled sur Afficheur écrire regroupe Valeur = Comptage sur la ligne 1 colonne 0
    attendre jusqu'à non Bouton-poussoir Bouton_Poussoir_A appuyé
```

Fichier MBlock : BP\_N3\_A2.sb2

## Exercice niveau 3 - A.3 : Afficher un message en fonction de l'état de la barrière

**Objectif :** Reprendre l'exercice BP\_N2\_A3

Afficher un message dépendant de l'état de la barrière. Si elle est ouverte, il doit afficher « ouverte », si elle est fermée, il doit afficher « fermer », si elle se ferme, il doit afficher « fermeture » et « ouverture » lorsqu'elle s'ouvre.

**Correction :**

Blocs

```
Arduino - générer le code
Activer émetteur IR ext
Oled sur Afficheur vider l'écran
répéter indéfiniment
  attendre jusqu'à Bouton-poussoir Bouton_Poussoir_A appuyé
  ouvrir
  attendre jusqu'à non Obstacle détecté par IR A
  attendre 3 secondes
  fermer
  ↵

définir fermer
  Actionner Barriere
  Activer Voyant Lumineux
  Oled sur Afficheur vider l'écran
  Oled sur Afficheur écrire Fermeture sur la ligne 1 colonne 0
  attendre jusqu'à Fin de course Fermeture activé
  Arrêter Barriere
  Désactiver Voyant Lumineux
  Oled sur Afficheur vider l'écran
  Oled sur Afficheur écrire Barrière fermée sur la ligne 1 colonne 0

définir ouvrir
  Actionner Barriere
  Activer Voyant Lumineux
  Oled sur Afficheur vider l'écran
  Oled sur Afficheur écrire Ouverture sur la ligne 1 colonne 0
  attendre jusqu'à Fin de course Ouverture activé
  Arrêter Barriere
  Désactiver Voyant Lumineux
  Oled sur Afficheur vider l'écran
  Oled sur Afficheur écrire Barrière ouverte sur la ligne 1 colonne 0
```

Fichier MBlock : BP\_N3\_A3.sb2

# Option : Module Bluetooth

Le module Bluetooth développé par A4 Technologie permet de convertir le protocole Bluetooth en protocole de communication type Série qui est le mode de communication classique utilisé avec PICAXE ou Arduino. Ce module accepte différentes configurations.

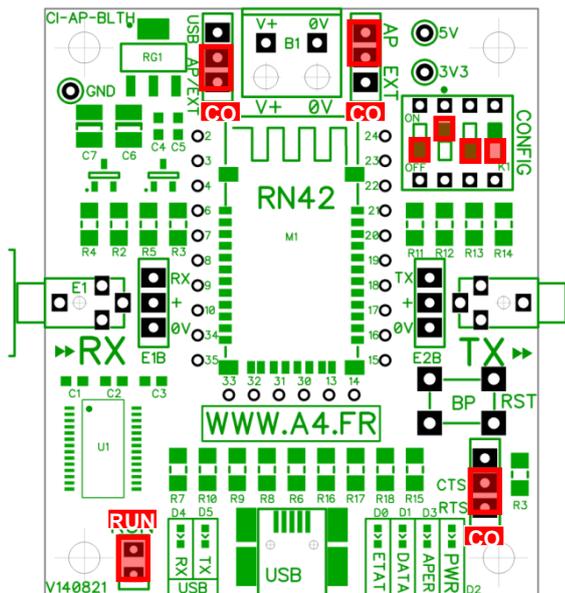
En mode avancé, il peut être configuré au travers d'une liaison par connexion USB à un PC ou par l'envoi de commandes au travers de ses liaisons RX et TX.

La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module. Elle est téléchargeable sur [http://a4.fr/wiki/index.php/Module Bluetooth - K-AP-MBLTH / S-113020008](http://a4.fr/wiki/index.php/Module_Bluetooth_-_K-AP-MBLTH_/S-113020008).

Les informations seront envoyées via un smartphone ou une tablette possédant la technologie bluetooth à l'aide d'une application développée sous Applinventor par l'équipe technique de A4.

## Configuration

Positionner les cavaliers et interrupteurs comme indiqué par les positions repérées en rouge ci-dessous.



Le cavalier repéré **RUN** est utilisé lors de la mise au point de programmes avec **Arduino**. Il doit être ôté pour permettre le téléversement du programme puis doit être remis lors de l'utilisation.

La mise au point de programmes avec **PICAXE** ne nécessite pas d'ôter ce cavalier pour transférer le programme. Les cavaliers **CO1** et **CO2** permettent de sélectionner le mode d'alimentation du module Bluetooth. Dans la configuration ci-dessus, son alimentation provient directement de l'interface AutoProg ou AutoProgUno au travers des cordons de liaison avec le module ; ils sont positionnés respectivement sur AP et sur AP/EXT.

Le cavalier **CO3** est utilisé en mode avancé pour relier ou dissocier les signaux CTS et RTS nécessaires au fonctionnement du module Bluetooth. Ici, il est positionné sur CTS/RTS.

Les interrupteurs **CONFIG** permettent de paramétrer le mode de fonctionnement du module Bluetooth. Ici, l'interrupteur n°2 est positionné sur ON pour sélectionner une vitesse de transmission des données à 9600 bauds.

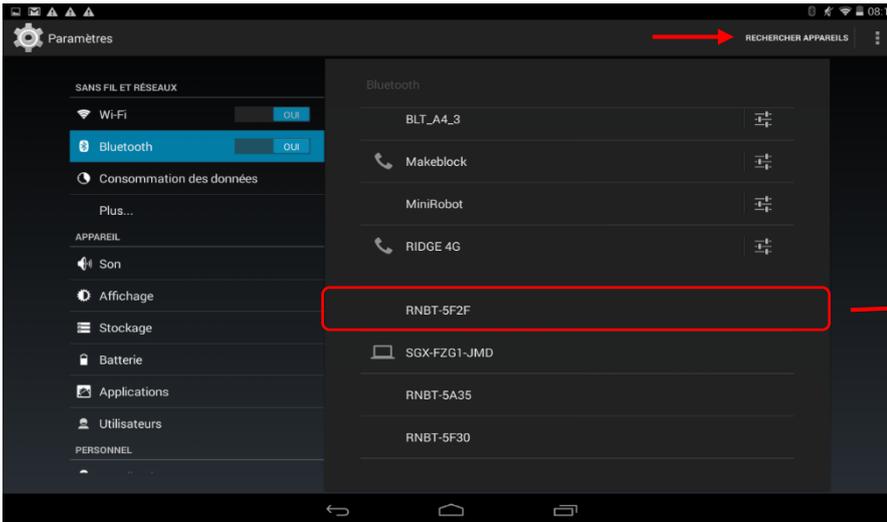
## Témoins lumineux

- PWR** indique que le module est sous tension.
- APER** indique que le module est associé avec un matériel Bluetooth.
- DATA** indique qu'il y a un flux de données entre le module et l'appareil avec lequel il est connecté.
- ETAT** indique que le module est opérationnel. L'affichage clignotant indique qu'il n'est pas opérationnel.
- USB RX** indique qu'il y a un flux de données sur la liaison USB du PC vers le module.
- USB TX** indique qu'il y a un flux de données sur la liaison USB du module vers le PC.

## Mise en place des programmes et procédure de connexion

Avant de commencer à tester les programmes il faut d'abord appairer le smartphone ou la tablette au module bluetooth.

Pour cela rendez-vous dans les réglages bluetooth et lancer une recherche d'appareils (la maquette doit être allumée pour alimenter le module). Le nom de votre module s'appelle : RNBT + les 4 derniers chiffres de l'adresse mac du module notés sur le composant. Sélectionnez le et un message proposant de vous connecter à lui devrait s'afficher.



Une fois cette étape passée vous pourrez vous connecter au module à partir du programme Applinventor à chaque fois.

Lorsque la connexion est réalisée, le bouton **Déconnexion** apparaît dans l'application.

Le témoin vert **DATA** s'allume sur le module dès qu'une donnée est émise ou reçue par le module Bluetooth.

L'appui sur le bouton d'envoi de données, dans cet exemple **Commande portail**, déclenche l'allumage fugitif de ce témoin.



# Exercice niveau 3 - C.1 : Ouvrir/fermer avec application Bluetooth

**Objectif :** contrôler l'ouverture et la fermeture du portail à l'aide de 2 boutons présent sur l'application Android.

**Notion abordée :** réception de données Bluetooth envoyées par un Smartphone.

**Instructions utilisées :**

Donnée Bluetooth reçue

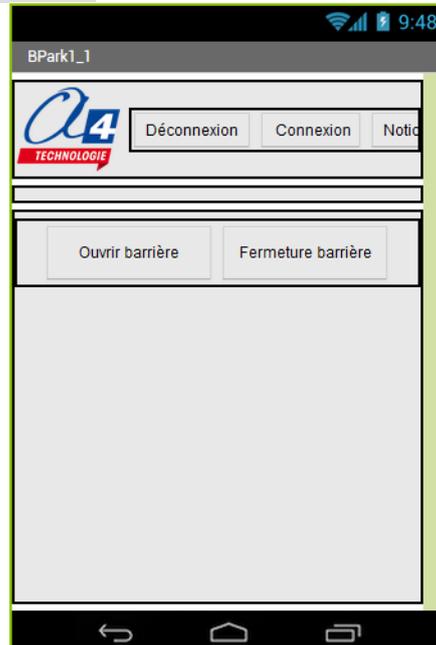
Envoyer donnée Bluetooth 0

**Application Android :** BPark1\_1.apk

**Fichier App Inventor :** BPark1\_1.aia

```
quand Ouvrir .Clic
faire appeler Bluetooth .Envoyer1Octet
      nombre 1

quand Fermer .Clic
faire appeler Bluetooth .Envoyer1Octet
      nombre 2
```



**Correction :**

## Blocs

```
Arduino - générer le code
répéter indéfiniment
si Donnée Bluetooth reçue = 1 alors
  ouvrir
si Donnée Bluetooth reçue = 2 alors
  fermer

définir fermer
  Ouvrir portail
  Activer Voyant Lumineux
  attendre jusqu'à Fin de course Fermeture activé
  Arrêter portail
  Désactiver Voyant Lumineux

définir ouvrir
  Ouvrir portail
  Activer Voyant Lumineux
  attendre jusqu'à Fin de course Ouverture activé
  Arrêter portail
  Désactiver Voyant Lumineux
```

Fichier MBlock : BP\_N3\_C1.sb2

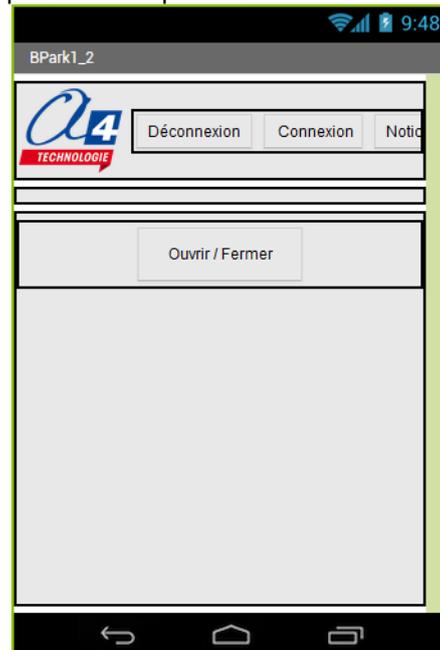
## Exercice niveau 3 - C.2 : Contrôle du portail par Smartphone

**Objectif** : ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.

**Notion abordée** : réception de données Bluetooth envoyées par un Smartphone.

**Application Android** : BPark1\_2.apk

**Fichier App Inventor** : BPark1\_2.aia



```
quand Ouvrir_fermer .Clic
faire appeler Bluetooth .Envoyer1Octet
      nombre 1
```

**Correction** :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    si Fin de course Fermeture activé alors
      ouvrir
    sinon
      fermer
  si Fin de course Ouverture activé alors
    fermer
  sinon
    ouvrir

définir fermer
Ouvrir portail
Activer Voyant Lumineux
attendre jusqu'à Fin de course Fermeture activé
Arrêter portail
Désactiver Voyant Lumineux

définir ouvrir
Ouvrir portail
Activer Voyant Lumineux
attendre jusqu'à Fin de course Ouverture activé
Arrêter portail
Désactiver Voyant Lumineux
```

Fichier MBlock : BP\_N3\_C2.sb2

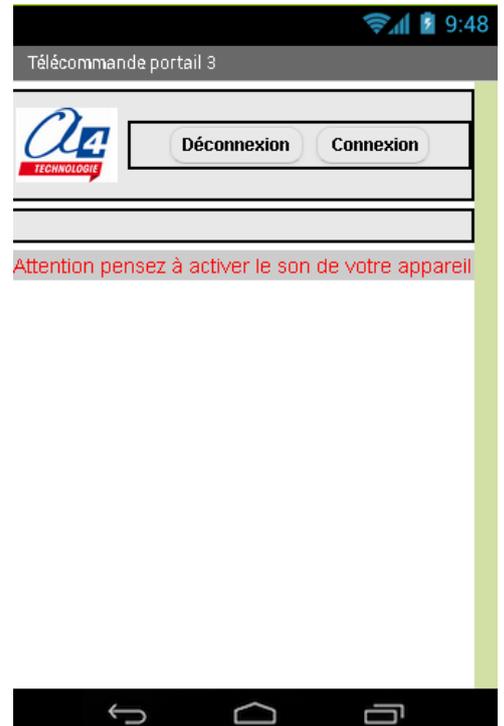
# Exercice niveau 3 - C.3 : Envoyer des données vers un Smartphone

**Objectif** : jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.

**Notion abordée** : envoyer des informations à un Smartphone par Bluetooth.

**Application Android** : BPark1\_3.apk

**Fichier App Inventor** : Portail\_3.aia



```
quand reception Chronomètre
faire
  si Bluetooth Est connecté
  alors
    si appeler Bluetooth Octets disponibles pour le réception > 0
    alors
      si appeler Bluetooth RecevoirOctetNonSignéNuméro 1
      alors
        appeler sonnette Jouer
```

**Correction** :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir Bouton_Poussoir_A appuyé alors
    Envoyer donnée Bluetooth 1
    attendre 1 secondes
```

Fichier MBlock : BP\_N3\_C3.sb2

# Exercice niveau 3 - C.4 : Envoyer et recevoir des données provenant d'un Smartphone

**Objectif :** gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.

**Notion abordée :** envoyer et recevoir des informations à l'aide du module Bluetooth à une application.

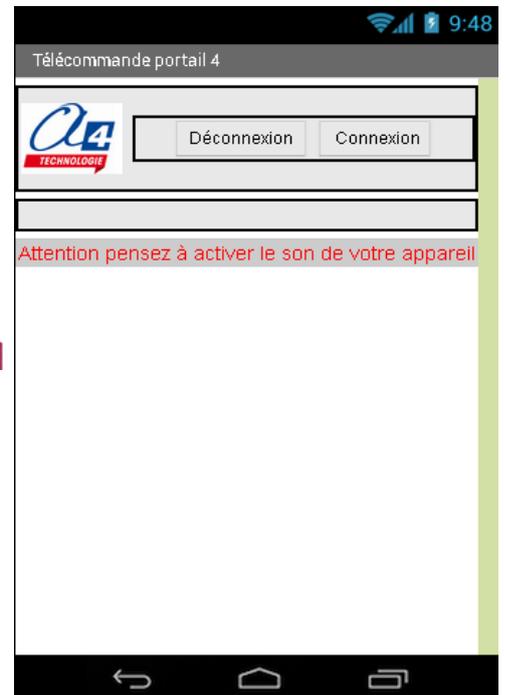
**Application Android :** BPark1\_4.apk

**App Inventor :** Portail\_4.aia

```

quand reception Chronomètre
faire
  si Bluetooth Est connecté
  alors
    si Bluetooth Octets disponibles pour la réception > 0
    alors
      si Bluetooth RecevoirOctetNonSignéNuméro1 = 1
      alors
        appeler sonnette Jouer
        appeler evenement Afficher fenêtre choix
          message " Voulez-vous monter la barrière ? "
          Titre " Bouton activé "
          Texte bouton 1 " Ouvrir la barrière "
          Texte bouton 2 " Ne rien faire "
          annulable faux
    fin si
  fin si
fin si

quand evenement Après choix
faire
  si Choix = " Ouvrir la barrière "
  alors
    appeler Bluetooth Envoyer1Octet
      nombre 1
  fin si
fin si
  
```



**Correction :**

Blocs

```

Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir Bouton_Poussoir_A appuyé alors
  Envoyer donnée Bluetooth 1
  attendre 0.5 secondes
  si Donnée Bluetooth reçue = 1 alors
  ouvrir
  attendre 5 secondes
  fermer
  fin si
  fin si
  
```

```

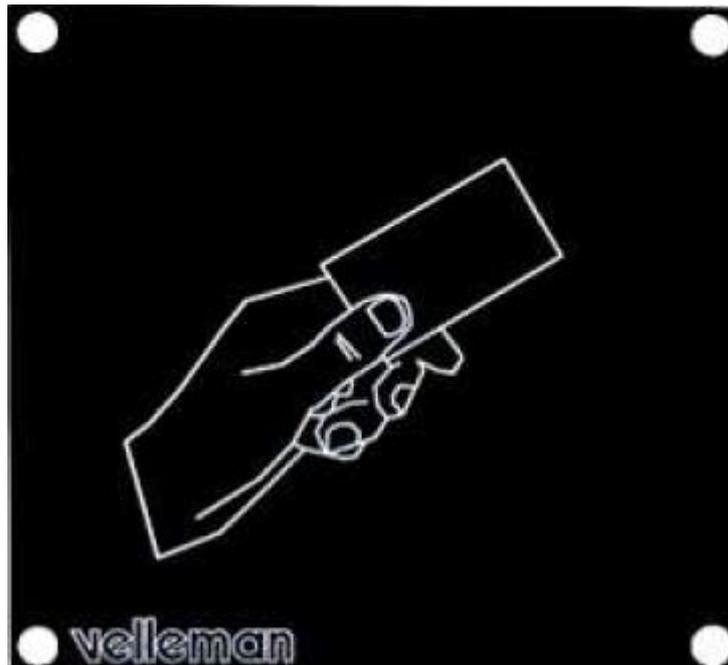
définir fermer
Ouvrir portail
Activer Voyant Lumineux
attendre jusqu'à Fin de course Fermeture activé
Arrêter portail
Désactiver Voyant Lumineux

définir ouvrir
Ouvrir portail
Activer Voyant Lumineux
attendre jusqu'à Fin de course Ouverture activé
Arrêter portail
Désactiver Voyant Lumineux
  
```

Fichier MBlock : BP\_N3\_C4.sb2

## Option : Module Capteur RFID

---



Module capteur RFID (ref : V-VM179)

Le module capteur RFID est une entrée tout ou rien répondant à une carte magnétique. Il agit donc comme un simple bouton poussoir lorsqu'on approche une carte magnétique spécifique.

Vous avez à disposition des cartes magnétiques permettant d'activer le module.

Il est nécessaire de fournir indépendamment une tension à ce module via une alimentation 12V à brancher sur la carte d'alimentation avec fusible et allumer le module grâce à l'interrupteur, la LED doit s'allumer.

Lorsque vous passez une carte devant le module, vous entendez un bruit signifiant qu'il a détecté quelque chose.

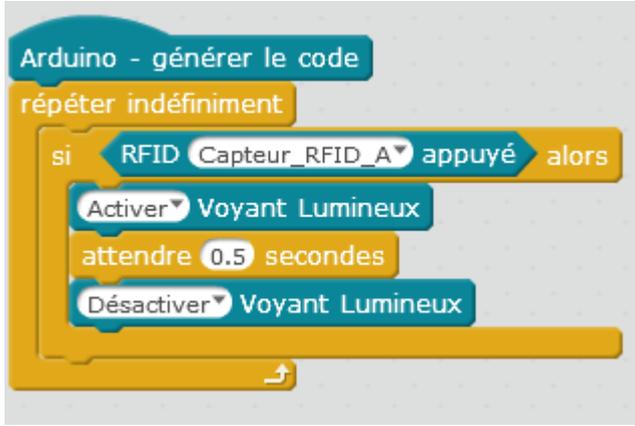
## Exercice niveau 3 – E.1 : Utilisation du module capteur RFID

**Objectif** : Basculer le voyant lumineux lorsque le module détecte une carte magnétique.

**Notion(s) abordée(s)** : lire une entrée numérique

**Correction** :

Blocs



```
Arduino - générer le code
répéter indéfiniment
  si RFID Capteur_RFID_A appuyé alors
    Activer Voyant Lumineux
    attendre 0.5 secondes
    Désactiver Voyant Lumineux
```

Fichier MBlock : BP\_N3\_E1.sb2

## Exercice niveau 3 – E.2 : Utilisation du module capteur RFID (2)

**Objectif :** Ouvrir, attendre 5 secondes, et fermer la barrière lors du déclenchement du capteur RFID

**Notion(s) abordée(s) :** Condition dépendant d'une entrée numérique

**Correction :**

Blocs

The image shows a screenshot of the mBlock programming environment with the following code blocks:

- Arduino - générer le code**
- répéter indéfiniment**
  - si** RFID Capteur\_RFID\_A appuyé alors
    - si** non Fin de course Ouverture activé alors
      - ouvrir**
      - attendre 5 secondes**
      - fermer**

- définir ouvrir**
- Ouvrir** Barriere
- répéter jusqu'à** Fin de course Ouverture activé
  - Activer** Voyant Lumineux
  - attendre 0.2 secondes**
  - Désactiver** Voyant Lumineux
  - attendre 0.2 secondes**
- Arrêter** Barriere
- définir fermer**
- Ouvrir** Barriere
- répéter jusqu'à** Fin de course Fermeture activé
  - Activer** Voyant Lumineux
  - attendre 0.2 secondes**
  - Désactiver** Voyant Lumineux
  - attendre 0.2 secondes**
- Arrêter** Barriere

Fichier MBlock : BP\_N3\_E2.sb2



CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES