

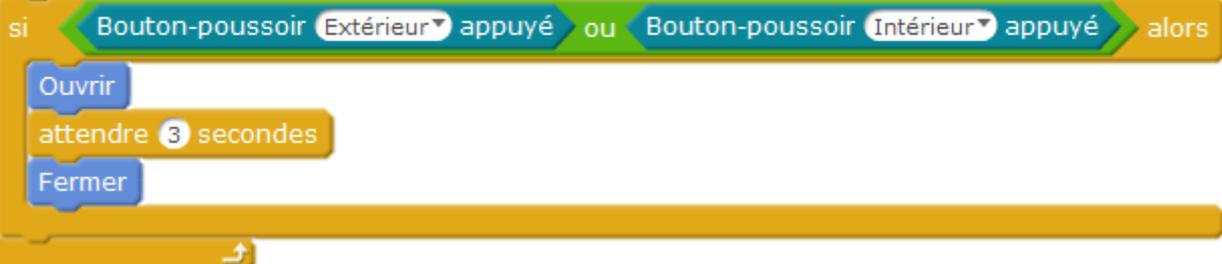
# Portail coulissant

## Maquette motorisée programmable avec mBlock 5



Arduino générer le code

répéter indéfiniment



# Ressources disponibles pour le projet Portail coulissant

Autour du projet Portail coulissant, nous vous proposons un ensemble de **ressources téléchargeables gratuitement sur [www.a4.fr](http://www.a4.fr)**.

## Portail coulissant

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

## Logiciels mBlock 5 et App Inventor 2

- Extension A4\_Portail pour animer la maquette Portail coulissant ;
- 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 5 et AppInventor 2.

**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>
Les environnements de programmation graphique .....	2
Le dossier .....	2
Les fiches exercices .....	3
Prérequis.....	3
Tableau d'affectation des entrées et sorties .....	4
Plan de câblage du portail coulissant .....	5
<b>Programmation version de base niveau 1 .....</b>	<b>6</b>
<b>Niveau 1 - A.....</b>	<b>7</b>
Exercice niv1 - A.1 : Activer / désactiver un témoin lumineux .....	7
Exercice niv1 - A.2 : Répéter une action deux fois.....	8
Exercice niv1 - A.3 : Répéter une séquence indéfiniment.....	9
<b>Niveau 1 - B.....</b>	<b>10</b>
Exercice niv1 - B.1 : Maitriser la rotation du moteur.....	10
Exercice niv1 - B.2 : Utilisation d'une boucle tant que.....	11
<b>Niveau 1 - C.....</b>	<b>12</b>
Exercice niv1 - C.1 : Instruction conditionnelle et bouton-poussoir .....	12
Exercice niv1 - C.2 : Instruction conditionnelle et barrière infrarouge .....	13
Exercice niv1 – C.3 : Contrôle moteur ET voyant lumineux .....	14
<b>Niveau 1 - D.....</b>	<b>15</b>
Exercice niv1 - D.1 : Utilisation des variables.....	15
Exercice niv1 - D.2 : Utiliser et tester une variable .....	16
Exercice niv1 - D.3 : Tests /variables/ modules IR .....	17
<b>Programmation version de base niveau 2 .....</b>	<b>18</b>
<b>Niveau 2 - A.....</b>	<b>19</b>
Exercice niv2 - A.1 : ouverture/fermeture entre fins de courses .....	19
Exercice niv2 - A.2 : Contrôle de l'ouverture et de la fermeture.....	20
Exercice niv2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité.....	21
Exercice niv2 - A.4 : Contrôle ouverture/fermeture avec BP et signal de sécurité.....	22
<b>Option : Module Bluetooth.....</b>	<b>24</b>
Configuration .....	24
Mise en place des programmes et procédure de connexion.....	25
Tableau d'affectation des entrées et sorties .....	26
Schéma de câblage .....	27
Exercice niv3 - B.1 : Ouvrir/fermer avec application Bluetooth .....	28
Exercice niv3 - B.2 : Contrôle du portail par Smartphone .....	29
Exercice niv3 - B.3 : Envoyer des données vers un Smartphone .....	30
Exercice niv3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone.....	31
<b>Option : Module capteur PIR.....</b>	<b>33</b>
Tableau d'affectation des entrées et sorties .....	34
Schéma de câblage .....	35
Exercice niv3 – D.1 : Utilisation du capteur PIR .....	36
Exercice niv3 – D.2 : Ouverture contrôlée à l'aide du capteur PIR .....	37

# Introduction

La maquette portail coulissant (BE-APORT-COUL) est une reproduction homothétique d'un portail coulissant automatisé réel : roue dentée, crémaillère, capteurs fin de course, barrière optique, clignotant de sécurité, etc. Programmable et pilotée par les systèmes AutoProgX2 ou AutoProgUno, elle 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 du Portail coulissant :

- 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 : module Bluetooth et capteur PIR.

## Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette Portail coulissant ont été réalisés sous **mBlock 5**.



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\_Portail\_Coulissant** (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 Portail coulissant 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 : module Bluetooth, capteur PIR (**détection de mouvement**).

## 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 modèle : un programme vide avec un jeu d'instructions limité (suffisant pour réaliser l'exercice) ;
- un fichier de correction qui propose un exemple de programme réalisé sous mBlock (extension .sb2).

Intérêt du fichier modèle :

- il évite aux utilisateurs de se perdre dans une multitude d'instructions ;
- il limite les propositions possibles ;
- il facilite la correction et l'analyse des erreurs.

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 :

- **PC** pour Portail coulissant ;
- **N** : niveau de programmation 1-2-3 ;
- **A-B-C** : jeu d'instructions du plus simple au plus avancé

Exemple : PC\_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 5**;
- **installer** l'extension **A4\_Portail** dans mBlock. (notice d'installation sur [www.a4.fr](http://www.a4.fr)) ;
- **maquette** Portail coulissant (Réf. BE-APOR-COUL) ;
- **câble de programmation** USB (Réf : CABL-IMPUSB) ;
- **interface programmable** AutoProgUno (Réf. K-AP-UNO) ;
- 13 **cordons de liaison** jack compatibles AutoProg pour établir les liaisons entre l'interface programmable et la maquette.

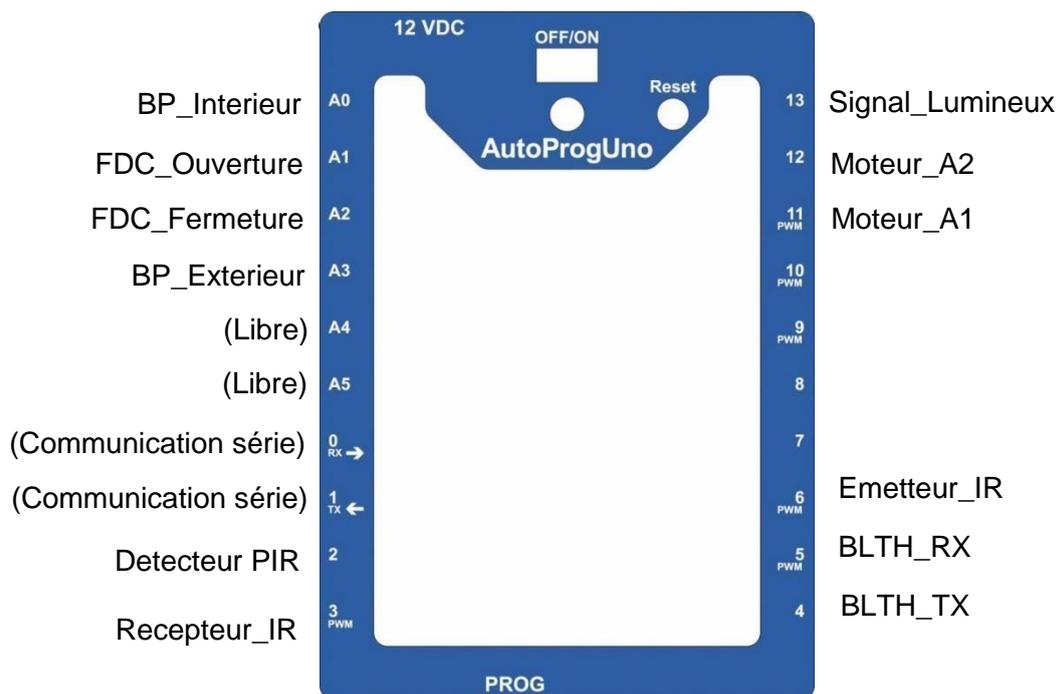
Pour l'option Bluetooth :

- **tablette ou smartphone** Android 5 ou + équipés de Bluetooth V3 (3.0) ;
- 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 [www.a4.fr](http://www.a4.fr)**

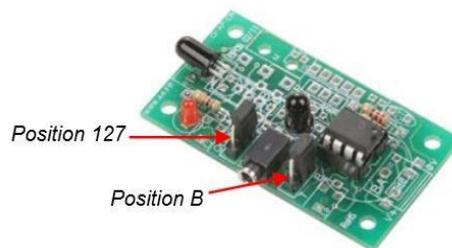
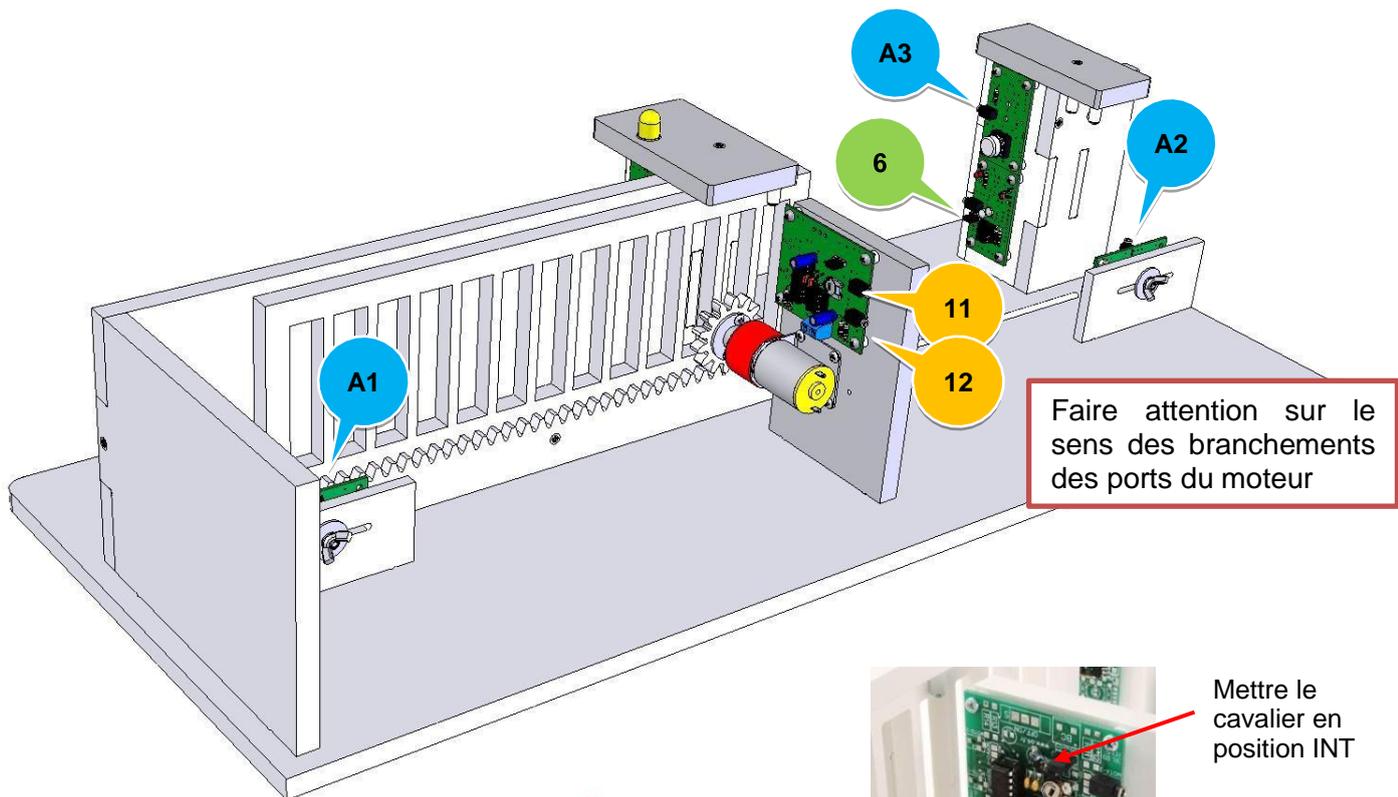
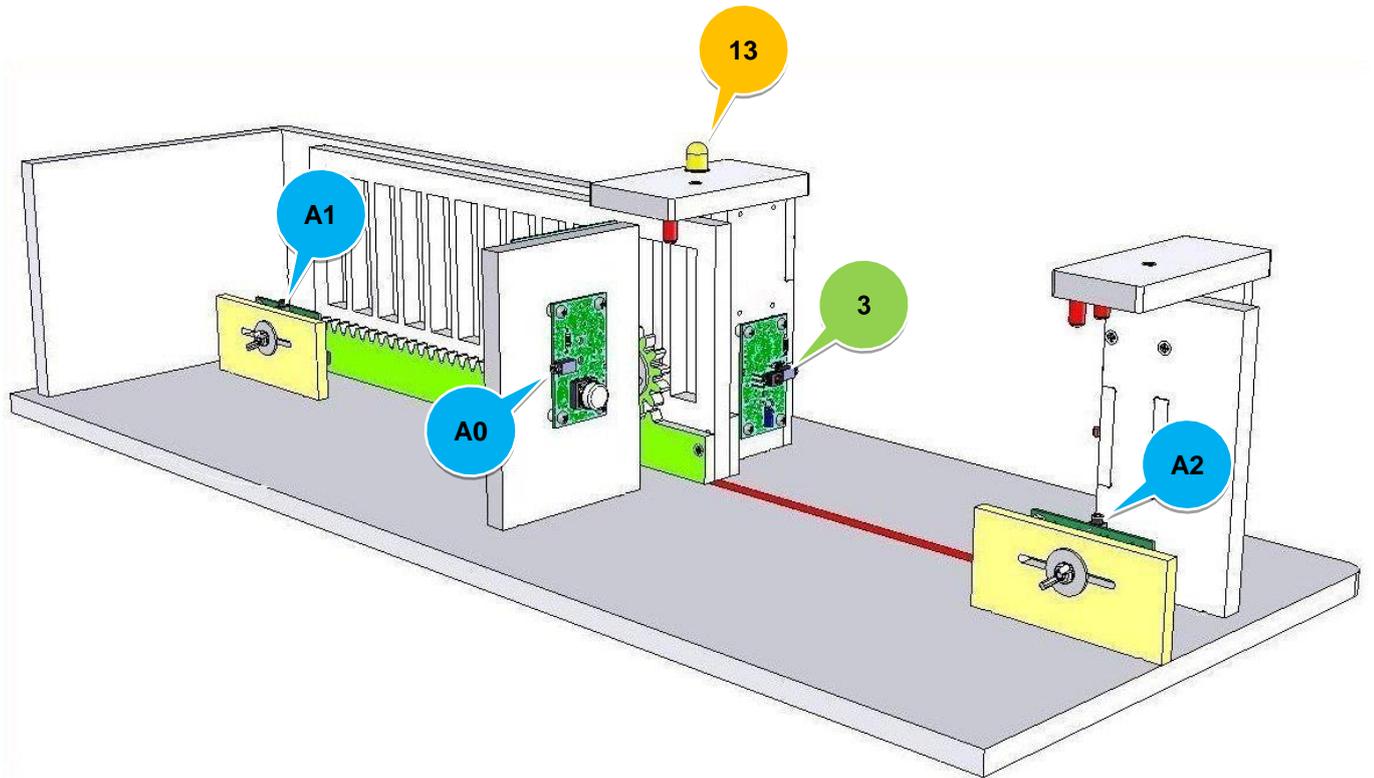
## Tableau d'affectation des entrées et sorties

AutoProgUno	Portail coulissant	Nom mBlock
<b>Modules capteurs pour entrées numériques</b>		
3	Récepteur infrarouge	Récepteur_IR
4 (option)	Module Bluetooth sortie (TX)	BLTH_TX
5 (option)	Module Bluetooth entrée (RX)	BLTH_RX
6	Émetteur infrarouge	
<b>Modules actionneurs pour sorties numériques</b>		
11	MOTA-1	Moteur_A1
12	MOTA-2	Moteur_A2
13	LED Gyrophare	Signal_Lumineux
<b>Module de communication</b>		
1	(communication avec ordinateur)	
2 (option)	Détecteur PIR	
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0	Bouton-poussoir intérieur	BP_Interieur
A1	Fin de course portail ouvert	FDC_Ouverture
A2	Fin de course portail fermé	FDC_Fermeture
A3	Bouton-poussoir extérieur	BP_Exterieur



r  
o  
p  
h

# Plan de câblage du portail coulissant



Mettre le cavalier en position INT

# 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.

Nous vous conseillons pour chaque exercice d'essayer d'écrire le programme vous-même, en partant du modèle de base (fournit avec les exercices), avant de regarder la correction et l'explication de chaque programme.

Par exemple pour le programme « PC\_N1\_A1.sb2 », charger le programme modèle « PC\_N1\_A.sb2 ».

Dans chaque programme modèle du niveau 1 vous trouverez la liste de blocs nécessaires à la réalisation des exercices des sous niveaux A, B, C et D. Au fur et à mesure de l'avancement dans les sous niveaux, la liste de blocs s'agrandit jusqu'à retrouver tous les blocs nécessaires pour piloter complètement la maquette.

Nom du fichier	Description	Objectif
<b>Niveau 1 A</b> <b>Fichier modèle : PC_mB5_N1_A</b>		
PC_mB5_N1_A1	Allumer la <b>LED Gyrophare</b> pendant 3 secondes puis l'éteindre.	<b>Fonctionnalité matérielle abordée :</b> - <b>allumage</b> /extinction du voyant lumineux <b>Notions de programmation abordées :</b> - séquence d'instructions - temps d'attente - boucle infinie
PC_mB5_N1_A2	Répéter cette même action deux fois.	
PC_mB5_N1_A3	Répéter cette action à l'infini.	
<b>Niveau 1 B</b> <b>Fichier modèle : PC_mB5_N1_B</b>		
PC_mB5_N1_B1	Activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.	<b>Fonctionnalité matérielle abordée :</b> - <b>gestion</b> du moteur - <b>utilisation</b> de Bouton-poussoir <b>Notions de programmation abordées :</b> - <b>boucle</b> qui dépend d'une entrée
PC_mB5_N1_B2	<b>allumer la LED Gyrophare en continue jusqu'à l'appui du bouton-poussoir puis la faire clignoter.</b>	
<b>Niveau 1 C</b> <b>Fichier modèle : PC_mB5_N1_C</b>		
PC_mB5_N1_C1	allumer la <b>LED Gyrophare</b> à l'appui du BP.	<b>Fonctionnalité matérielle abordée :</b> - <b>gestion</b> des modules infra-rouge - <b>utilisation</b> de Bouton-poussoir <b>Notions de programmation abordées :</b> - <b>le</b> test d'une entrée (si/sinon).
PC_mB5_N1_C2	activer la <b>LED Gyrophare</b> lorsque la barrière infrarouge est franchie.	
PC_mB5_N1_C3	contrôler l'allumage de la <b>LED</b> et du moteur avec des BP.	
<b>Niveau 1 D</b> <b>Fichier modèle : PC_mB5_N1_D</b>		
PC_mB5_N1_D1	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débugage).	<b>Notions de programmation abordées :</b> - <b>définition</b> de variable - <b>incrément</b> de variable - <b>test</b> (si/sinon) de variable - <b>test</b> (juste si) d'entrée - <b>debugage</b> .
PC_mB5_N1_D2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer la <b>LED</b> .	
PC_mB5_N1_D3	incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état de la <b>LED</b> .	

# Niveau 1 - A

---

## Exercice niv1 - A.1 : Activer / désactiver un témoin lumineux

**Objectif** : allumer la LED Gyrophare pendant 3 secondes puis l'éteindre.

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

**Attention** à bien téléverser le programme via « Téléverser dans l'Arduino »

Retour

Téléverser dans l'Arduino

**Correction** : Fichier mBlock : PC\_mB5\_N1\_A1



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

## Exercice niv1 - A.2 : Répéter une action deux fois

**Objectif :** allumer la LED Gyrophare pendant 3 secondes puis l'éteindre, recommencer.

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

**Correction :** Fichier mBlock : PC\_mB5\_N1\_A2



## Exercice niv1 - A.3 : Répéter une séquence indéfiniment

**Objectif** : faire clignoter la LED Gyrophare avec une période de 6 secondes indéfiniment.

**Notion abordée** : la boucle infinie.

**Correction** : Fichier mBlock: PC\_mB5\_N1\_A3



**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 niv1 - B.1 : Maitriser la rotation du moteur

**Objectif** : activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.

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

**Correction** : Fichier mBlock : PC\_N1\_B1



**(ATTENTION** : pour cet exercice, il est recommandé d'enlever la barrière du portail pour éviter tout dommage. Il faut également activer le moteur à l'aide de l'interrupteur (Une LED rouge indique si le moteur est allumé).)

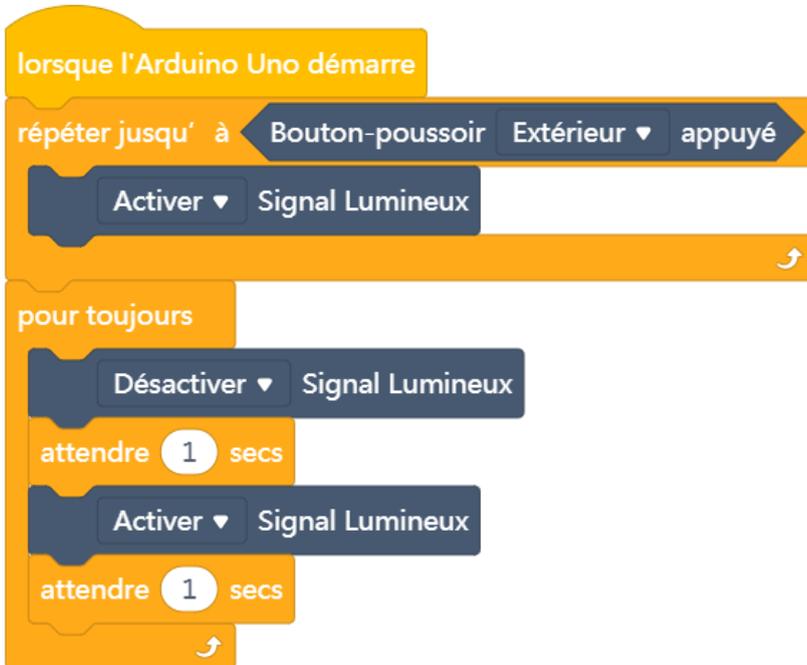
**Information** : des blocs spécifiques sont disponibles pour contrôler le portail dans l'extension A4\_Portail.

## Exercice niv1 - B.2 : Utilisation d'une boucle tant que

**Objectif :** allumer la LED Gyrophare en continue jusqu'à l'appui du bouton-poussoir. Une fois le bouton-poussoir appuyé, faire clignoter la LED Gyrophare.

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

**Correction :** Fichier mBlock : PC\_mB5\_N1\_B2



# Niveau 1 - C

---

## Exercice niv1 - C.1 : Instruction conditionnelle et bouton-poussoir

**Objectif** : allumer la **LED Gyrophare** à l'appui du BP.

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

**Correction** : Fichier mBlock : PC\_mB5\_N1\_C1



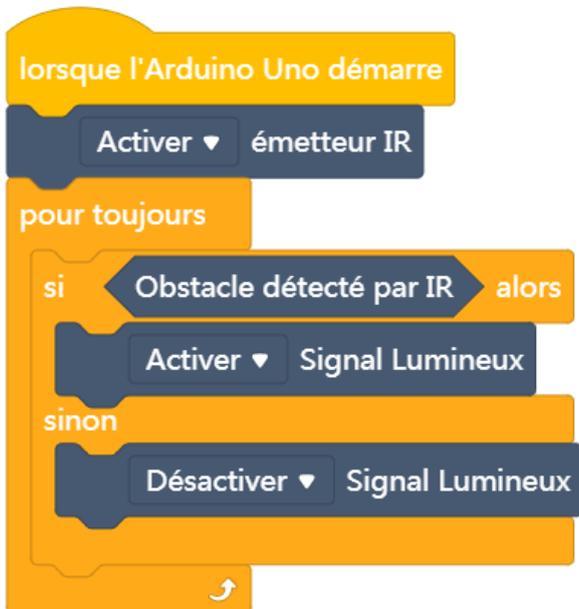
**Remarque** : Le récepteur est activé lorsqu'il n'y a rien, c'est à dire qu'il est désactivé lorsqu'il est en face d'un émetteur.

## Exercice niv1 - C.2 : Instruction conditionnelle et barrière infrarouge

**Objectif** : activer la **LED Gyrophare** lorsque la barrière infrarouge est franchie.

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

**Correction** : Fichier mBlock : PC\_mB5\_N1\_C2



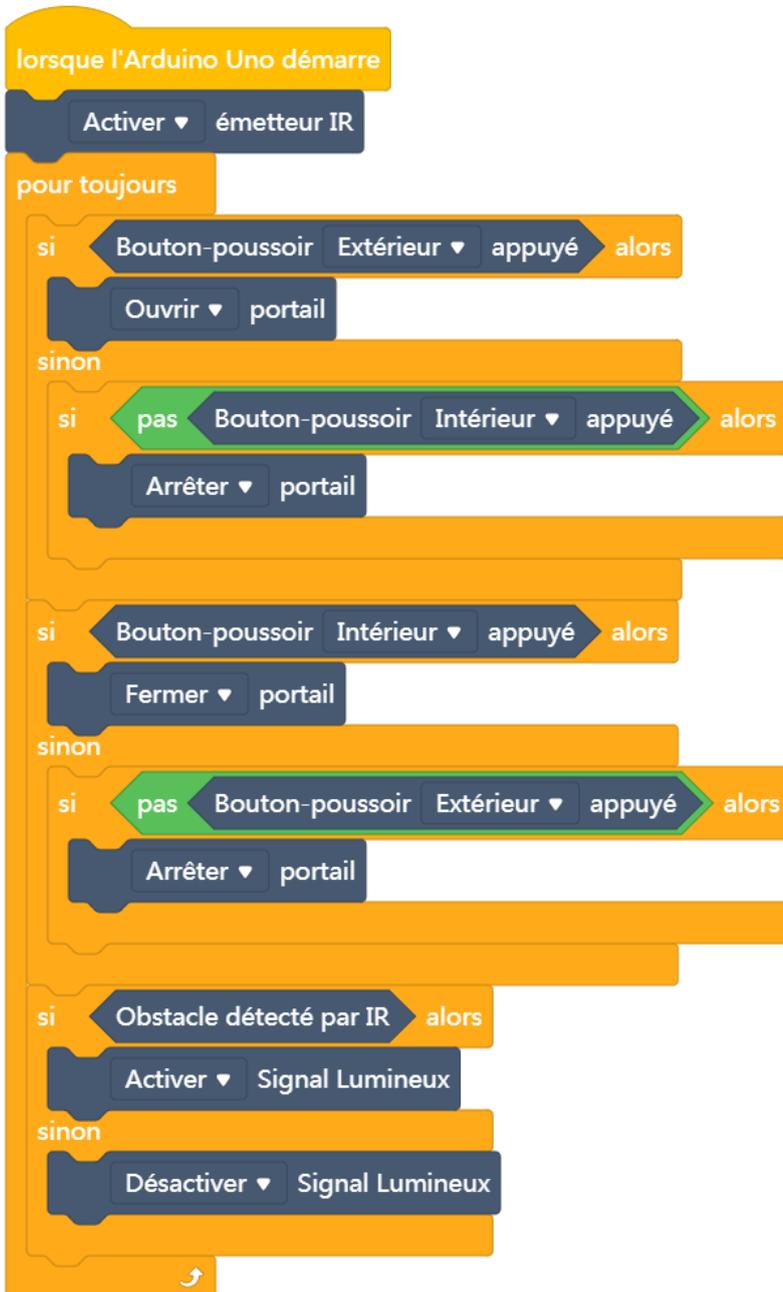
**Remarque** : l'entrée du récepteur IR est activée d'origine et se désactive lors de la réception du signal de l'émetteur IR. Lorsque un obstacle franchi la barrière IR, le signal n'est plus transmis et l'entrée du récepteur IR devient active.

## Exercice niv1 – C.3 : Contrôle moteur ET voyant lumineux

**Objectif** : contrôler le moteur avec les boutons poussoirs et allumer la LED Gyrophare sur le franchissement de la barrière infrarouge.

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

**Correction** : Fichier mBlock : PC\_mB5\_N1\_C3



# Niveau 1 - D

## Exercice niv1 - D.1 : Utilisation des variables

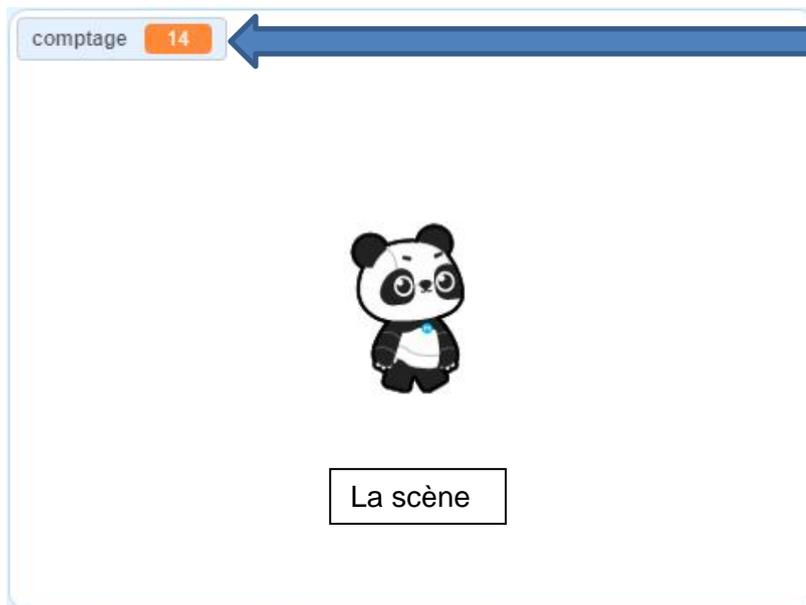
**Objectif** : incrémenter une variable **toutes les secondes** et observer sa valeur **sur la scène** (comptage).

**Notions abordées** : la variable (définition et incrémentation).

**Correction** : Fichier **mBlock** : PC\_mB5\_N1\_D1



Pour ce type de programme, il faut utiliser cette instruction de départ.



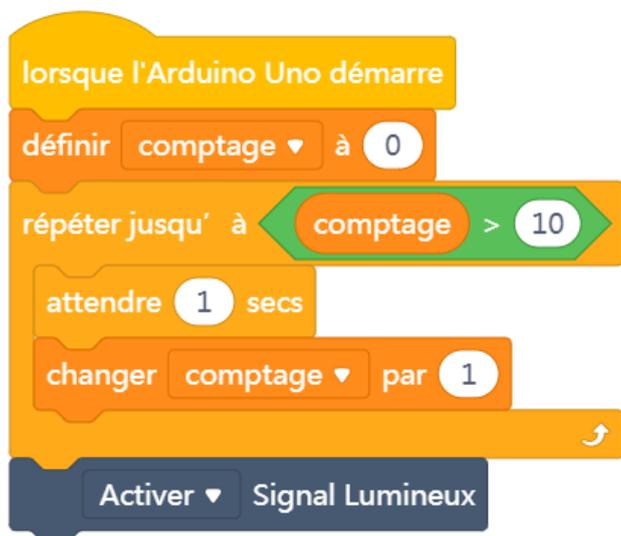
Après déclaration de la variable « Comptage », l'incrémement s'effectuera à cet emplacement sur la scène.

## Exercice niv1 - D.2 : Utiliser et tester une variable

**Objectif** : incrémenter une variable **chaque seconde**. Lorsque la variable est supérieure à 10, activer la **LED Gyrophare**.

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

**Correction** : Fichier **mBlock** : PC\_mB5\_N1\_D2



**Remarque** : cet exercice peut être utilisé comme un minuteur. **L'incrémentation ne pourra être visualisée sur la scène, car nous interagissons ici avec le portail.**

## Exercice niv1 - D.3 : Tests /variables/ modules IR

**Objectif** : incrémenter une variable chaque seconde s'il y a un obstacle sur la barrière IR. Lorsque le compteur arrive à 10, activer la LED Gyrophare durant 3 secondes et remettre la variable à zéro.

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

**Correction** : Fichier mBlock : PC\_mB5\_N1\_D3



# Programmation version de base niveau 2

## Objectifs :

- utilisation concrète du portail coulissant ;
- utilisation de tous les modules de la maquette ;
- appréhension des différentes fonctionnalités du matériel ainsi que certaines notions de sécurité.

Ce niveau permet de mettre en œuvre le portail coulissant, 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 un portail qui marche parfaitement et qui respecte une logique de fonctionnement calquée sur le réel.

Nom du fichier	Description	Objectif
<b>Niveau 2 A</b> <b>Fichier modèle : PC_mB5_N2_A</b>		
PC_mB5_N2_A1	Ouvrir et fermer le portail avec 2 secondes d'attente entre chaque mouvement. Utiliser les capteurs fins decourse pour contrôler l'ouverture et la fermeture.	<b>Notions de programmation abordées :</b> - utilisation des FDC (Fin De Course)
PC_mB5_N2_A2	Ouverture du portail à l'appui sur BP_Exterieur. Fermeture du portail à l'appui sur BP_Interieur.	
PC_mB5_N2_A3	Ouvrir et fermer le portail à l'aide des BP sans distinction, faire en sorte que la LED Gyrophare clignote lors d'une manœuvre de la barrière.	
PC_mB5_N2_A4	Ouvrir et fermer le portail à l'aide des BP sans distinction, la LED Gyrophare doit clignoter lors d'une manœuvre de la barrière. Inclure une gestion de sécurité lors la fermeture du portail.	

# Niveau 2 - A

## Exercice niv2 - A.1 : ouverture/fermeture entre fins de courses

**Objectif** : ouvrir puis fermer le portail avec 2 secondes d'attente entre chaque mouvement.  
Utiliser les capteurs fins de course pour contrôler l'ouverture et la fermeture.

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

**Correction** : Fichier mBlock : PC\_mB5\_N2\_A1



Ref. K-AP-MMR

```
lorsque l'Arduino Uno démarre
pour toujours
  ouvrir
  attendre 2 secs
  fermer
  attendre 2 secs
```

```
définir ouvrir
  Ouvrir portail
  attendre jusqu' à Fin de course Ouverture activé
  Arrêter portail
```

```
définir fermer
  Fermer portail
  attendre jusqu' à Fin de course Fermeture activé
  Arrêter portail
```

**Remarque** : l'utilisation des sous-fonctions « fermer » et « ouvrir » facilite la lecture du programme.

## Exercice niv2 - A.2 : Contrôle de l'ouverture et de la fermeture

**Objectif :** ouverture du portail à l'appui sur BP\_Exterieur / fermeture du portail à l'appui sur BP\_Interieur

**Notions abordées :**

**Correction :** Fichier **mBlock** : PC\_mB5\_N2\_A2

```
lorsque l'Arduino Uno démarre
pour toujours
  attendre jusqu' à Bouton-poussoir Extérieur ▼ appuyé
  ouvrir
  attendre jusqu' à Bouton-poussoir Intérieur ▼ appuyé
  fermer
  ↻
```

```
définir ouvrir
Ouvrir ▼ portail
attendre jusqu' à Fin de course Ouverture ▼ activé
Arrêter ▼ portail
```

```
définir fermer
Fermer ▼ portail
attendre jusqu' à Fin de course Fermeture ▼ activé
Arrêter ▼ portail
```

## Exercice niv2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité

**Objectif** : ouvrir puis fermer le portail à l'aide des BP sans distinction, faire en sorte que la **LED Gyrophare** clignote lors d'une manœuvre de la barrière.

**Notions abordées** : utilisation d'opérateur logique OU (+)

**Correction** : Fichier **mBlock** : PC\_mB5\_N2\_A3

```
lorsque l'Arduino Uno démarre
  Arrêter ▼ portail
  pour toujours
    si Bouton-poussoir Extérieur ▼ appuyé ou Bouton-poussoir Intérieur ▼ appuyé alors
      ouvrir
      attendre 3 secs
      fermer
```

```
définir ouvrir
  Ouvrir ▼ portail
  répéter jusqu' à Fin de course Ouverture ▼ activé
    Activer ▼ Signal Lumineux
    attendre 0.1 secs
    Désactiver ▼ Signal Lumineux
    attendre 0.1 secs
  Arrêter ▼ portail
```

```
définir fermer
  Fermer ▼ portail
  répéter jusqu' à Fin de course Fermeture ▼ activé
    Activer ▼ Signal Lumineux
    attendre 0.1 secs
    Désactiver ▼ Signal Lumineux
    attendre 0.1 secs
  Arrêter ▼ portail
```

# Exercice niv2 - A.4 : Contrôle ouverture/fermeture avec BP et signal de sécurité

**Objectif** : ouvrir puis fermer le portail à l'aide des BP sans distinction, la LED Gyrophare doit clignoter lors d'une manœuvre de la barrière. Inclure une gestion de sécurité lors la fermeture du portail.

**Notions abordées** : utilisation d'une procédure de sécurité.

**Correction** : Fichier mBlock : PC\_mB5\_N2\_A4

```
lorsque l'Arduino Uno démarre
  Activer émetteur IR
  fermer
  pour toujours
    si Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé alors
      si Fin de course Ouverture activé alors
        fermer
      sinon
        ouvrir
```

```
définir fermer
  Fermer portail
  répéter jusqu' à Fin de course Fermeture activé
    si Obstacle détecté par IR alors
      ouvrir
      attendre jusqu' à Fin de course Ouverture activé
      attendre 2 secs
    si Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé alors
      Fermer portail
      Activer Signal Lumineux
      attendre 0.1 secs
      Désactiver Signal Lumineux
      attendre 0.1 secs
  Arrêter portail
```

```
définir ouvrir
  Ouvrir portail
  répéter jusqu' à Fin de course Ouverture activé
    Activer Signal Lumineux
    attendre 0.1 secs
    Désactiver Signal Lumineux
    attendre 0.1 secs
  Arrêter portail
```

# Programmation version de base niveau 3 (OPTIONS)

Objectif :

- utiliser les modules plus complexes : pilotage à distance, contrôle par le courant...

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 B</b>		
PC_mB5_N3_B1	Contrôler l'ouverture et la fermeture du portail à l'aide de 2 boutons présent sur l'application Android.	<b>Fonctionnalité matérielle abordée :</b> - module Bluetooth  <b>Notions de programmation abordées :</b> - liaison série (hserin/hserout)
PC_mB5_N3_B2	Ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.	
PC_mB5_N3_B3	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.	
PC_mB5_N3_B4	Gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.	
<b>Niveau 3 D</b>		
PC_mB5_N3_D1	Allumer la LED Gyrophare lorsqu'une présence est détectée par le capteur PIR.	<b>Fonctionnalité matérielle abordée :</b> - capteur PIR
PC_mB5_N3_D2	Gestion complète du portail avec ouverture par détection PIR.	

# 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.

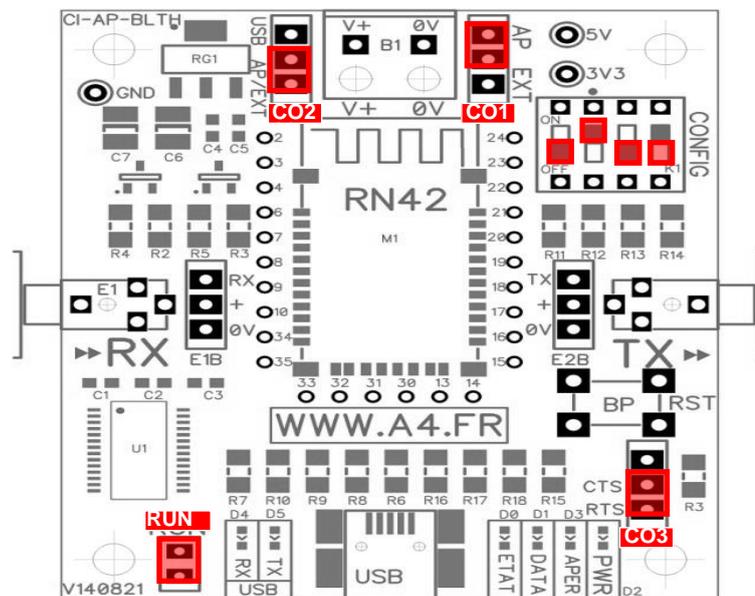
Ce module accepte différentes configurations.

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 (réception) et TX (transmission). Les informations seront envoyées via un smartphone ou une tablette possédant la technologie Bluetooth à l'aide d'une application développée sous AppInventor 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.

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.

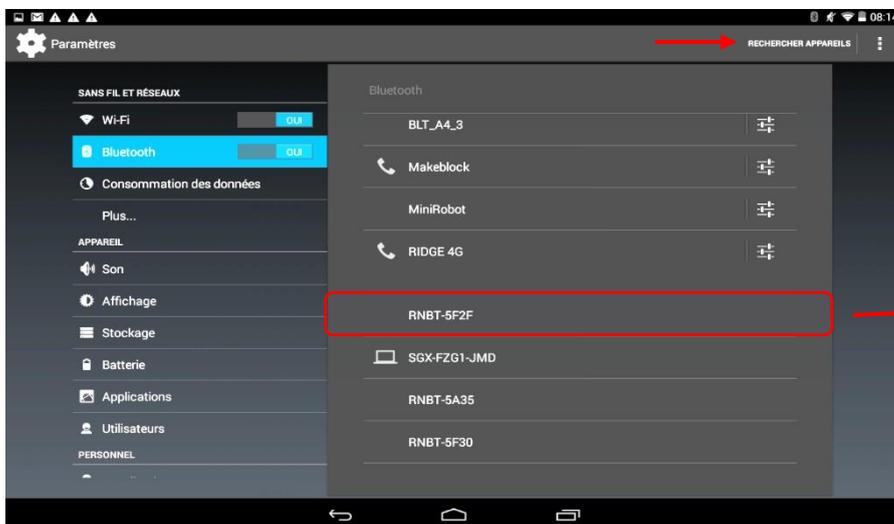
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

Disponible sur [www.a4.fr](http://www.a4.fr)

# 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.)



**Remarque** : l'appairage entre le smartphone ou la tablette au module Bluetooth s'effectue via l'application (fichier apk) à télécharger sur le site [www.a4.fr](http://www.a4.fr).

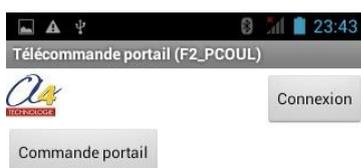
Une fois cette étape passée vous pourrez vous connecter au module Bluetooth à partir des programmes que nous devons réaliser sur le logiciel ApplInventor 2.

Pour en savoir davantage sur le logiciel ApplInventor 2 et de son utilisation, nous avons créé un tutoriel de prise en main du logiciel sur le site [www.a4.fr](http://www.a4.fr).

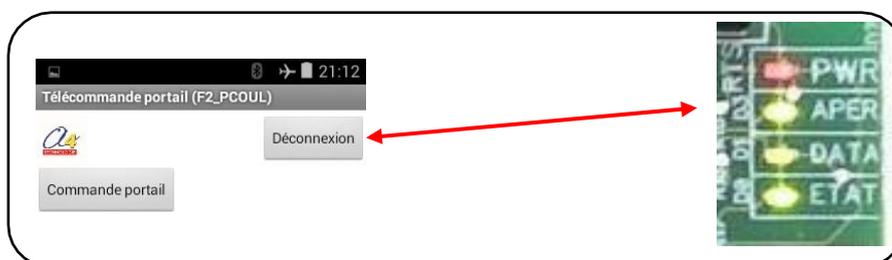
Lorsque la connexion est réalisée, le bouton **Déconnexion** apparaît dans l'application.

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.



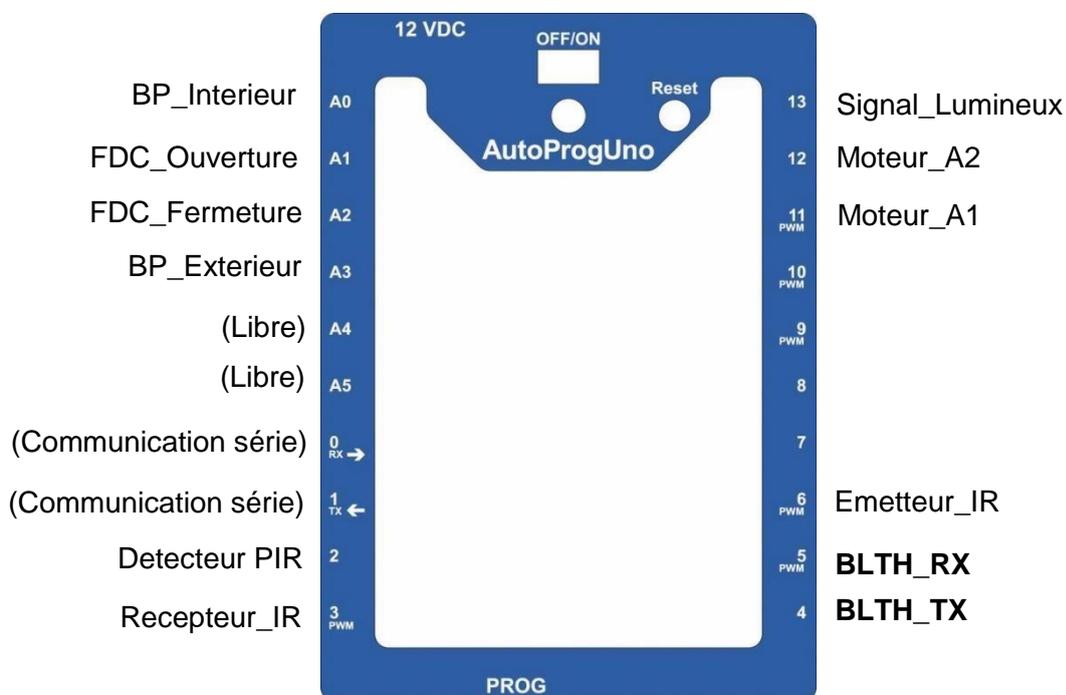
Appuyer sur le bouton « Connexion » et choisissez le nom du module. L'appairage avec le module prend quelques secondes.



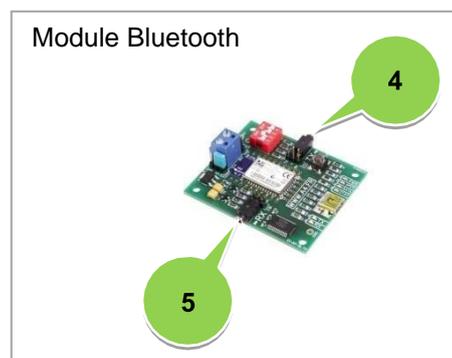
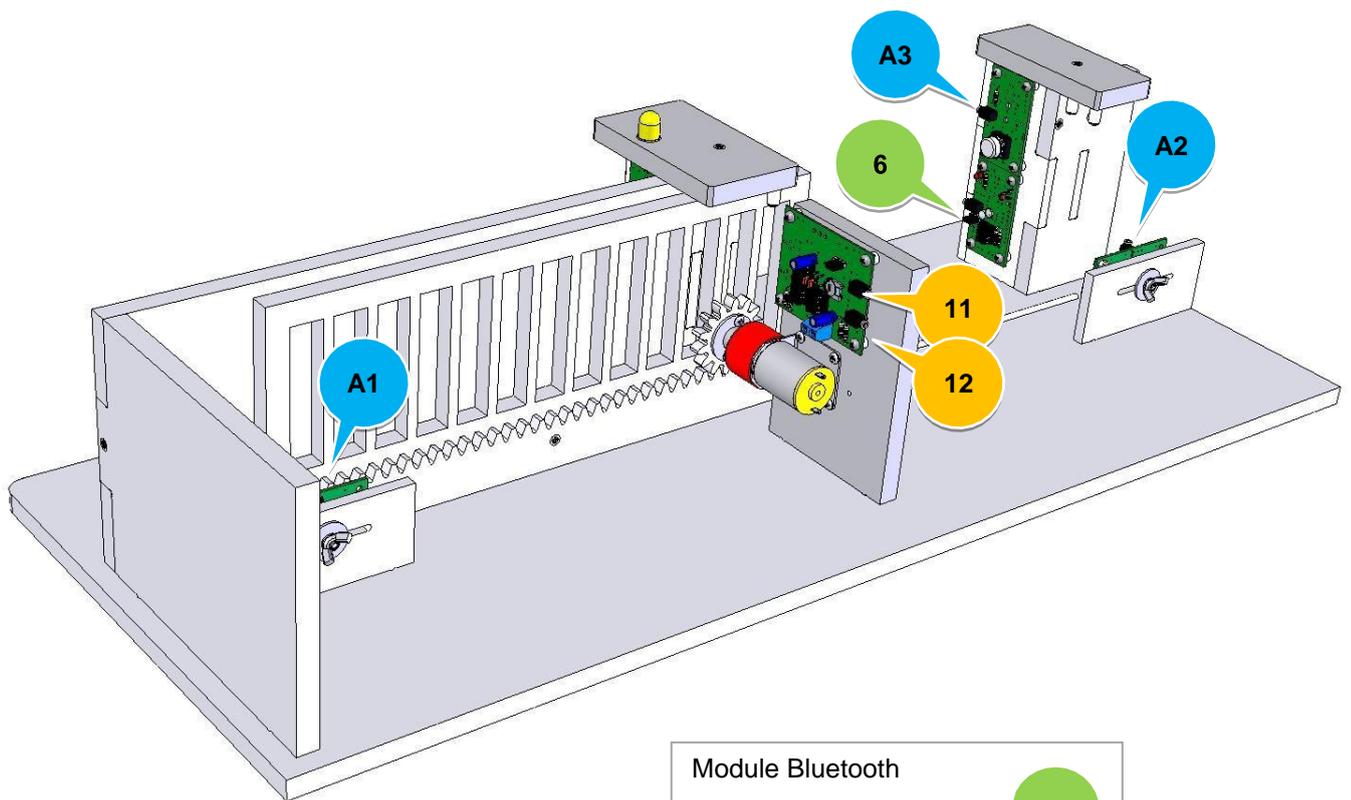
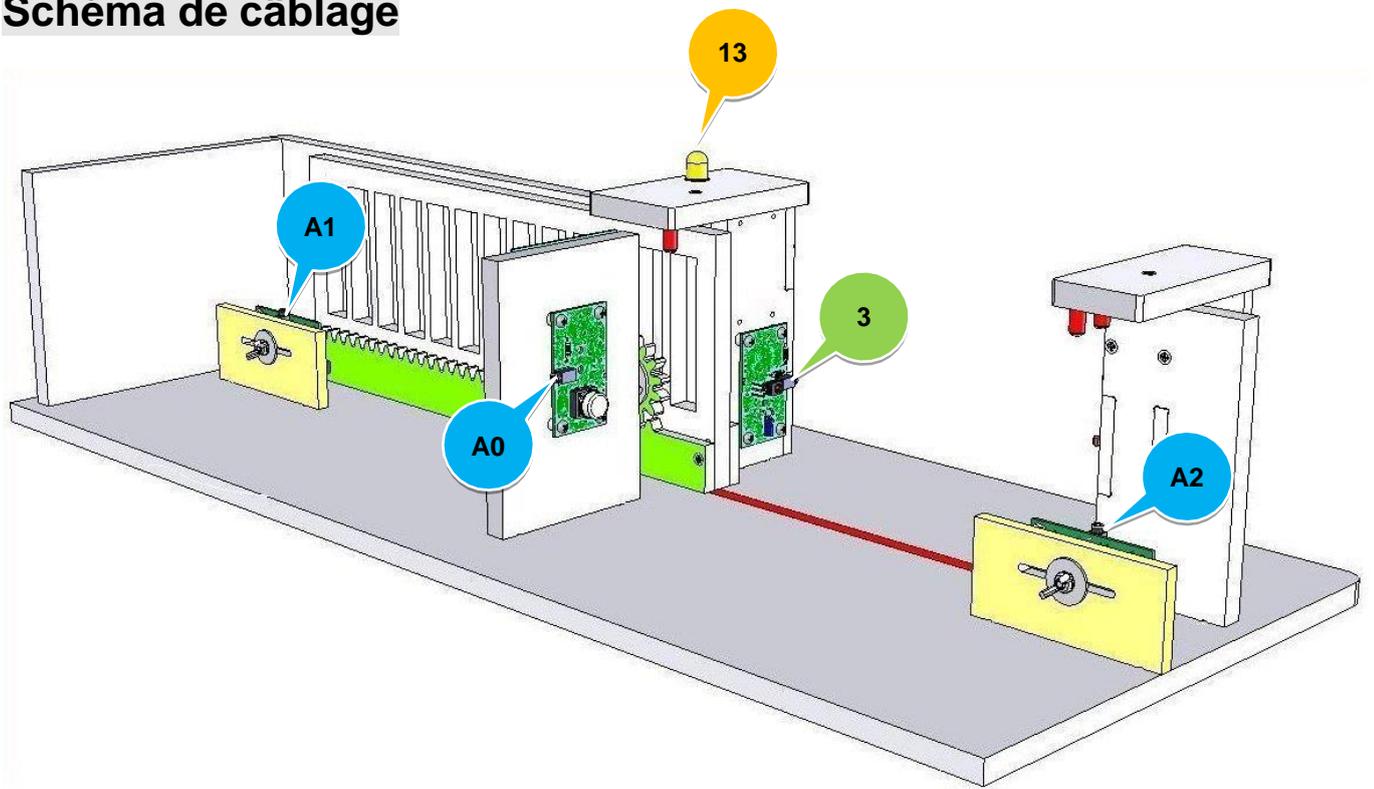
Connexion établie avec succès

## Tableau d'affectation des entrées et sorties

AutoProgUno	Portail coulissant	Nom mBlock
<b>Modules capteurs pour entrées numériques</b>		
3	Récepteur infrarouge	Récepteur_IR
4 (option)	Module Bluetooth sortie (TX)	BLTH_TX
5 (option)	Module Bluetooth entrée (RX)	BLTH_RX
6	Émetteur infrarouge	
<b>Modules actionneurs pour sorties numériques</b>		
11	MOTA-1	Moteur_A1
12	MOTA-2	Moteur_A2
13	LED Gyrophare	Signal_Lumineux
<b>Module de communication</b>		
1	(communication avec ordinateur)	
2 (option)	Détecteur PIR	
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0	Bouton-poussoir intérieur	BP_Interieur
A1	Fin de course portail ouvert	FDC_Ouverture
A2	Fin de course portail fermé	FDC_Fermeture
A3	Bouton-poussoir extérieur	BP_Exterieur



# Schéma de câblage



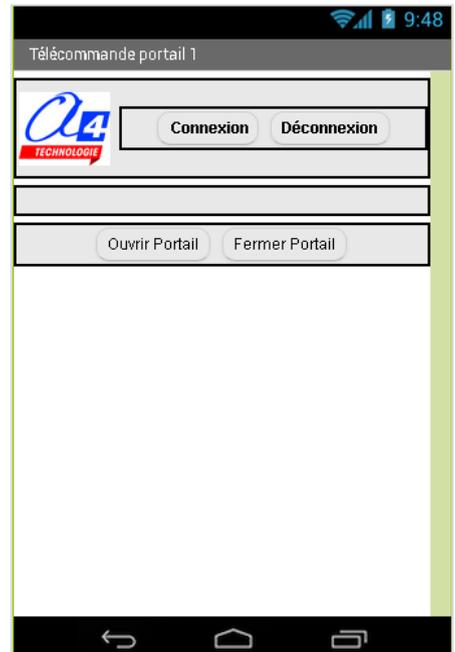
# Exercice niv3 - B.1 : Ouvrir/fermer avec application Bluetooth

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

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

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

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



```

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

```

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

**Correction :** Fichier mBlock : PC\_mB5\_N3\_B1

```

    lorsque l'Arduino Uno démarre
    Activer émetteur IR
    fermer
    pour toujours
    définir BLTH à Donnée Bluetooth reçue
    si BLTH = 1 alors
    ouvrir
    sinon
    si BLTH = 2 alors
    fermer
  
```

```

    définir ouvrir
    Ouvrir portail
    répéter jusqu' à Fin de course Ouverture activé
    Activer Signal Lumineux
    attendre 0.1 secs
    Désactiver Signal Lumineux
    attendre 0.1 secs
    Arrêter portail
  
```

```

    définir fermer
    Fermer portail
    répéter jusqu' à Fin de course Fermeture activé
    Activer Signal Lumineux
    attendre 0.1 secs
    Désactiver Signal Lumineux
    attendre 0.1 secs
    Arrêter portail
  
```

# Exercice niv3 - B.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 :** Portail\_2.apk

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



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

**Correction :** Fichier mBlock : PC\_mB5\_N3\_B2

```
lorsque l'Arduino Uno démarre
  Activer ▼ émetteur IR
  Fermer ▼ portail
  pour toujours
    définir BLTH ▼ à Donnée Bluetooth reçue
    si BLTH = 1 alors
      si Fin de course Fermeture ▼ activé alors
        ouvrir
      sinon
        fermer
```

```
définir ouvrir
  Ouvrir ▼ portail
  répéter jusqu' à Fin de course Ouverture ▼ activé
    Activer ▼ Signal Lumineux
    attendre 0.1 secs
    Désactiver ▼ Signal Lumineux
    attendre 0.1 secs
  Arrêter ▼ portail
```

```
définir fermer
  Fermer ▼ portail
  répéter jusqu' à Fin de course Fermeture ▼ activé
    si Obstacle détecté par IR alors
      ouvrir
      attendre jusqu' à Fin de course Ouverture ▼ activé
    si Bouton-poussoir Extérieur ▼ appuyé ou Bouton-poussoir Intérieur ▼ appuyé alors
      Fermer ▼ portail
      Activer ▼ Signal Lumineux
      attendre 0.1 secs
      Désactiver ▼ Signal Lumineux
      attendre 0.1 secs
  Arrêter ▼ portail
```

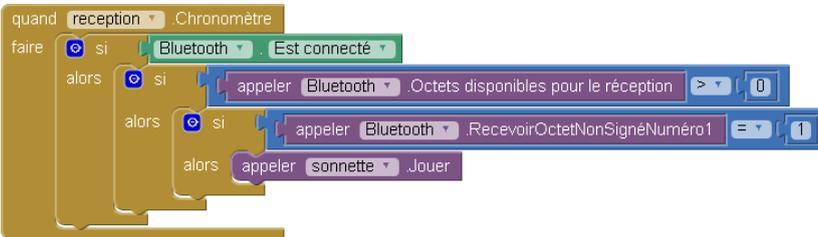
## Exercice niv3 - B.3 : Envoyer des données vers un Smartphone

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

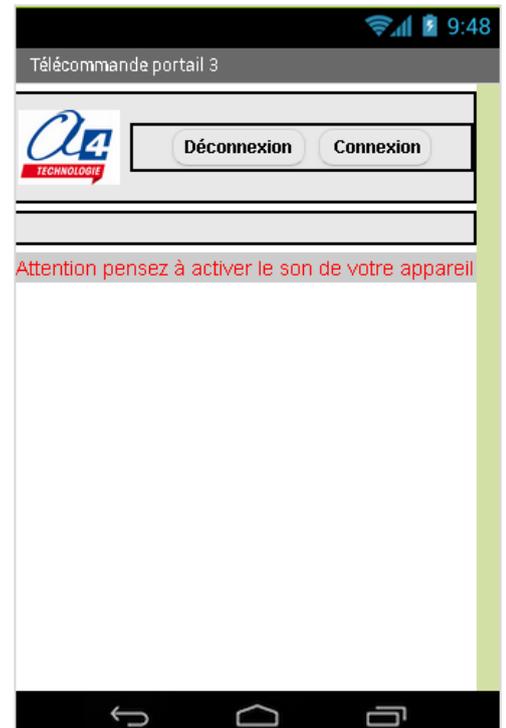
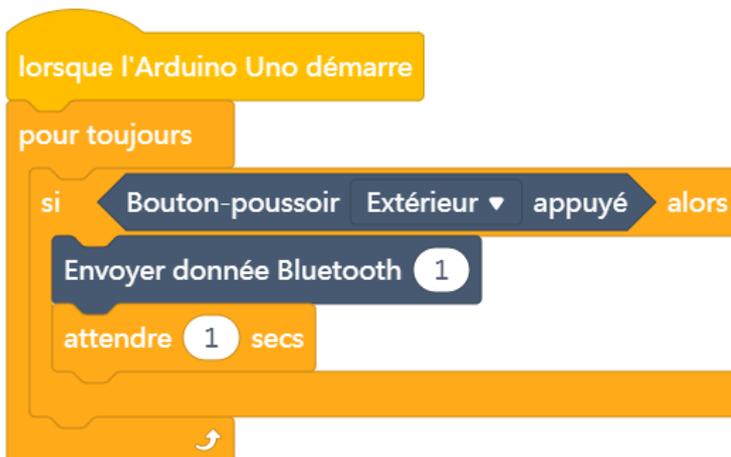
**Notion abordée** : envoyer des informations à un Smartphone par Bluetooth.

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

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



**Correction** : Fichier mBlock : PC\_mB5\_N3\_B3



# Exercice niv3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone

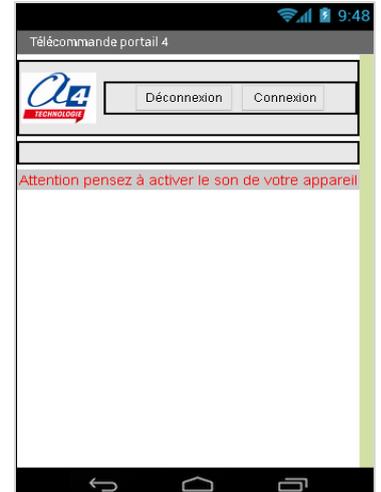
**Objectif :** gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.

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

**Application Android :** Portail\_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 demande_ouverture Afficher fenêtre choix
          message "Une personne souhaite entrer, que voulez-vous faire ?"
          Titre "Sonnette"
          Texte bouton 1 "Ne rien faire"
          Texte bouton 2 "Ouvrir le portail"
          annulable faux
```



```
quand demande_ouverture Après choix
ChoiX
faire
  si obtenir Choix = "Ouvrir le portail"
  alors
    appeler Bluetooth Envoyer1Octet
      nombre 1
```

**Correction :** Fichier mBlock : PC\_mB5\_N3\_B4

```
lorsque l'Arduino Uno démarre
  Activer émetteur IR
  fermer
  pour toujours
    définir BLTH à Donnée Bluetooth reçue
    si Bouton-poussoir Extérieur appuyé alors
      Envoyer donnée Bluetooth 1
      attendre 1 secs
    si BLTH = 1 alors
      ouvrir
      attendre 3 secs
      fermer
```

```

définir ouvrir
  Ouvrir ▾ portail
  répéter jusqu' à Fin de course Ouverture ▾ activé
    Activer ▾ Signal Lumineux
    attendre 0.1 secs
    Désactiver ▾ Signal Lumineux
    attendre 0.1 secs
  Arrêter ▾ portail

```

```

définir fermer
  Fermer ▾ portail
  répéter jusqu' à Fin de course Fermeture ▾ activé
    si Obstacle détecté par IR alors
      ouvrir
      attendre jusqu' à Fin de course Ouverture ▾ activé
    si Bouton-poussoir Extérieur ▾ appuyé ou Bouton-poussoir Intérieur ▾ appuyé alors
      Fermer ▾ portail
      Activer ▾ Signal Lumineux
      attendre 0.1 secs
      Activer ▾ Signal Lumineux
      attendre 0.1 secs
  Arrêter ▾ portail

```

## Option : Module capteur PIR

---

Le module PIR est équipé d'un capteur pyroélectrique. Il réagit aux faibles variations de température et permet de détecter la présence (mouvement) d'une personne jusqu'à 5 m. Son champ de détection est de 60° jusqu'à 2,5 m et 20° à 5 m.

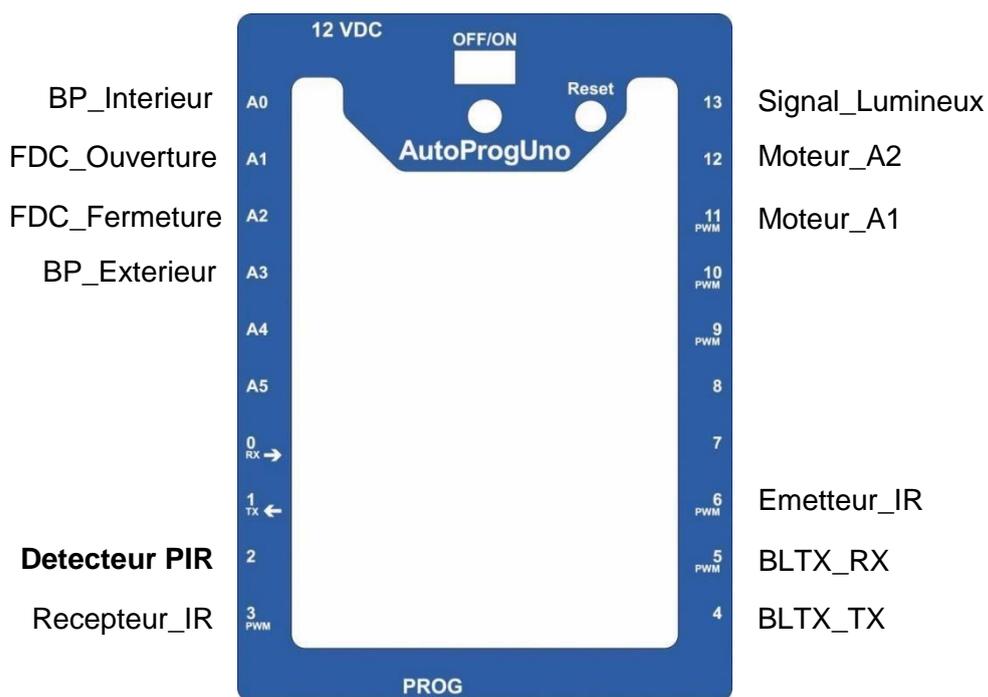
Le capteur réagit comme un bouton poussoir actif lors d'une détection d'un mouvement. Son activation est retardée d'environ 20 secondes après la mise sous tension afin d'éviter les détections intempestives.

Par ailleurs, le capteur est sensible aux variations de températures brutales, aux vibrations ou aux chocs importants. Il ne faut pas l'exposer à la lumière directe du soleil, à l'air pulsé d'un radiateur ou d'un climatiseur. Il est conçu pour une utilisation en intérieur ; pour une utilisation en extérieur, une protection anti humidité est nécessaire.

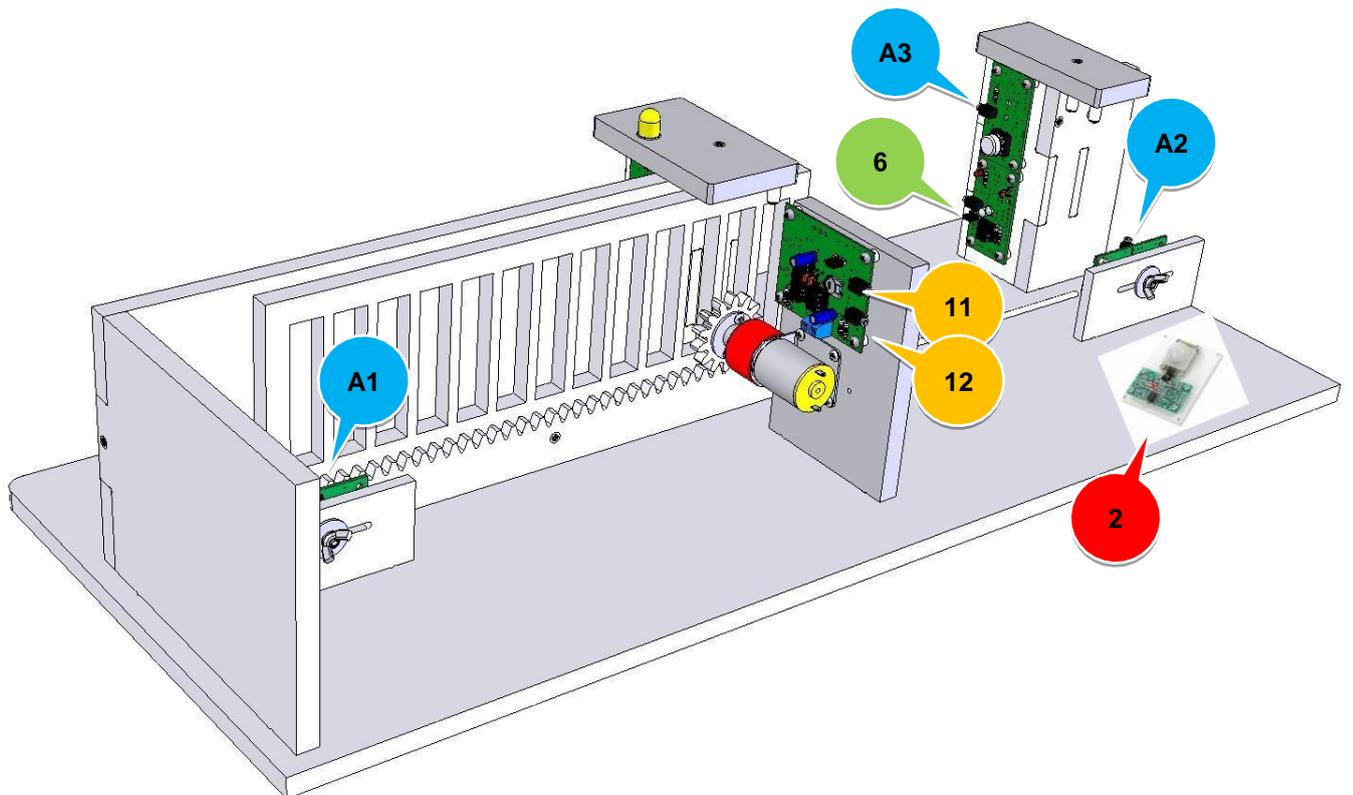
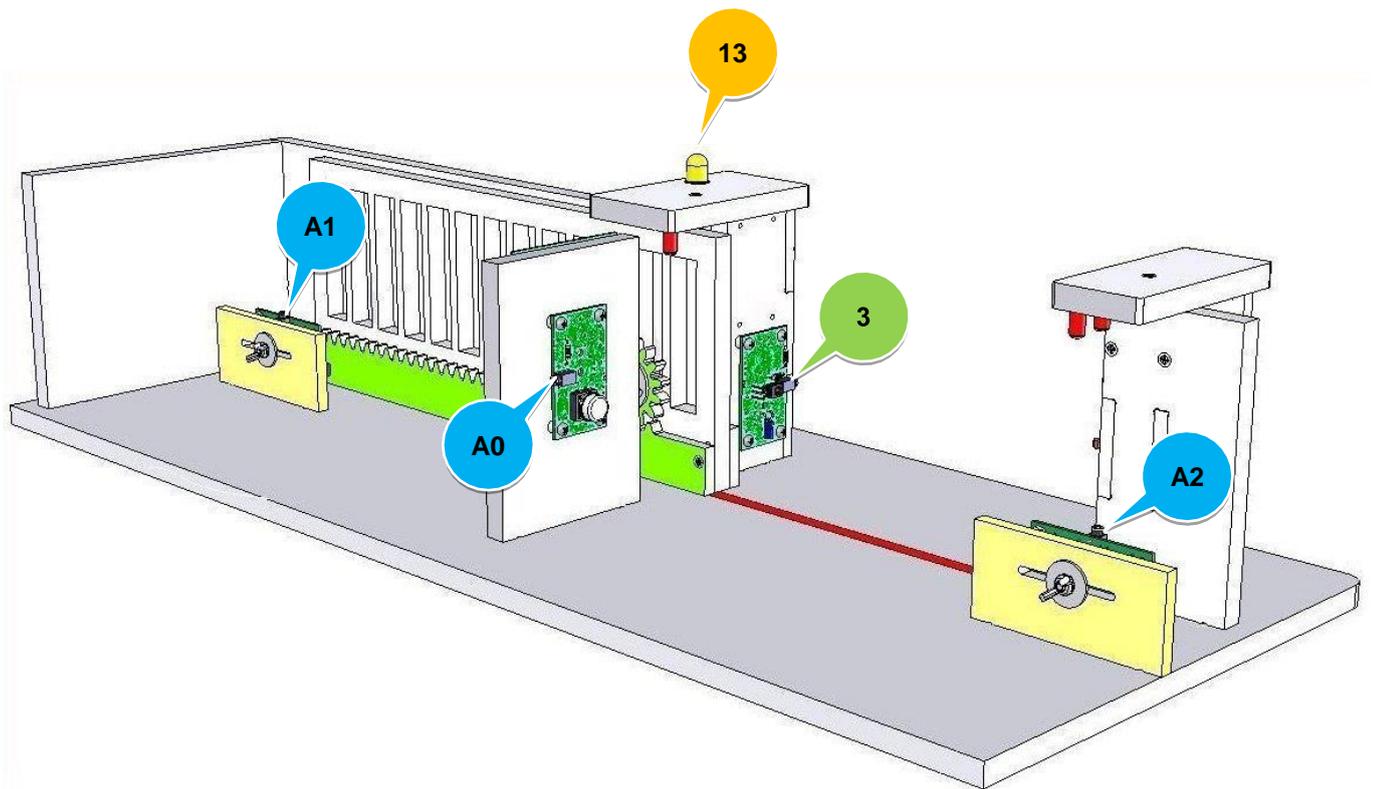


## Tableau d'affectation des entrées et sorties

AutoProgUno	Portail coulissant	Nom mBlock
<b>Modules capteurs pour entrées numériques</b>		
3	Récepteur infrarouge	Recepteur_IR
4	Module Bluetooth sortie (TX)	BLTH_TX
5	Module Bluetooth entrée (RX)	BLTH_RX
6	Emetteur infrarouge	
<b>Modules actionneurs pour sorties numériques</b>		
11	MOTA-1	Moteur_A1
12	MOTA-2	Moteur_A2
13	LED gyrophare	Signal_Lumineux
<b>Module de communication</b>		
1	(communication avec ordinateur)	
2	Détecteur PIR	
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0	Bouton-poussoir intérieur	BP_Interieur
A1	Fin de course portail ouvert	FDC_Ouverture
A2	Fin de course portail fermé	FDC_Fermeture
A3	Bouton-poussoir extérieur	BP_Exterieur



## Schéma de câblage



## Exercice niv3 – D.1 : Utilisation du capteur PIR

**Objectif** : allumer la **LED Gyrophare** lorsqu'une présence est détectée par le capteur PIR.

**Notion abordée** : utilisation du capteur de présence PIR.

**Correction** : Fichier **mBlock** : PC\_mB5\_N3\_D1

**Remarque** : Une attente d'environ 30 secondes après l'alimentation du module est nécessaire pour que le module se mette en route correctement.



## Exercice niv3 – D.2 : Ouverture contrôlée à l'aide du capteur PIR

**Objectif :** reprendre le programme PC\_N2\_A4 et ajouter la **détection** par le capteur PIR pour ouvrir le portail.

**Correction :** Fichier mBlock : PC\_mB5\_N3\_D2

```
lorsque l'Arduino Uno démarre
  Activer émetteur IR
  fermer
  pour toujours
    si Bouton-poussoir Extérieur ▼ appuyé ou Bouton-poussoir Intérieur ▼ appuyé alors
      si Fin de course Ouverture ▼ activé alors
        fermer
      sinon
        ouvrir
    si Présence détectée par capteur PIR alors
      si Fin de course Fermeture ▼ activé alors
        ouvrir
```

```
définir ouvrir
  Ouvrir portail
  répéter jusqu' à Fin de course Ouverture ▼ activé
    Activer Signal Lumineux
    attendre 0.1 secs
    Désactiver Signal Lumineux
    attendre 0.1 secs
  Arrêter portail
```

```
définir fermer
  Fermer ▾ portail
  répéter jusqu' à Fin de course Fermeture ▾ activé
  si Obstacle détecté par IR alors
    ouvrir
    attendre jusqu' à Fin de course Ouverture ▾ activé
    attendre 2 secs
  si Bouton-poussoir Extérieur ▾ appuyé ou Bouton-poussoir Intérieur ▾ appuyé alors
    Fermer ▾ portail
    Activer ▾ Signal Lumineux
    attendre 0.1 secs
    Désactiver ▾ Signal Lumineux
    attendre 0.1 secs
  Arrêter ▾ portail
```



[www.a4.fr](http://www.a4.fr)

Concepteur et fabricant de matériels pédagogiques