

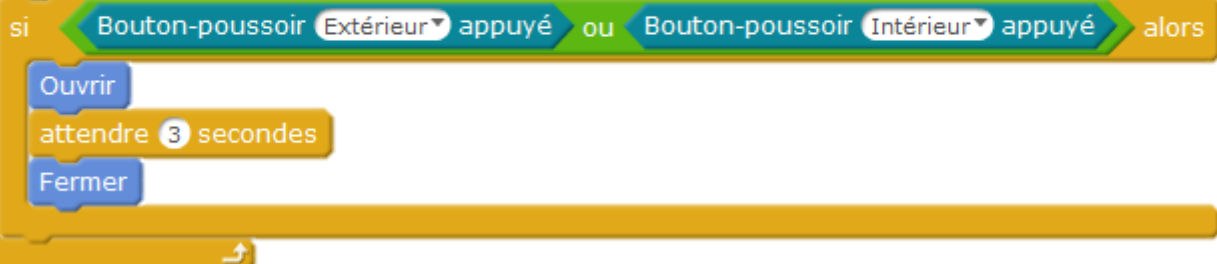
Portail coulissant

Maquette motorisée programmable avec mBlock



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 le wiki**.

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 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**

SOMMAIRE

Introduction	2
Portail coulissant.....	2
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
Programmation version de base niveau 1	6
Niveau 1 - A.....	7
Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux.....	7
Exercice niveau 1 - A.2 : Répéter une action deux fois.....	8
Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment.....	9
Niveau 1 - B.....	10
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur.....	10
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que	11
Niveau 1 - C.....	12
Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir.....	12
Exercice niveau 1 - C.2 : Instruction conditionnelle et barrière infrarouge.....	13
Exercice niveau 1 - C.3 : Contrôle moteur ET voyant lumineux.....	14
Niveau 1 - D.....	15
Exercice niveau 1 - D.1 : Utilisation des variables	15
Exercice niveau 1 - D.2 : Utiliser et tester une variable.....	16
Exercice niveau 1 - D.3 : Tests /variables/ modules IR.....	17
Programmation version de base niveau 2	18
Niveau 2 - A.....	19
Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses	19
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture.....	20
Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité	21
Exercice niveau 2 - A.4 : Contrôle ouverture/fermeture avec BP et signal de sécurité	22
Option : Module Bluetooth.....	24
Configuration	24
Mise en place des programmes et procédure de connexion.....	25
Tableau d'affectation des entrées et sorties.....	26
Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth	27
Exercice niveau 3 - B.2 : Contrôle du portail par Smartphone	28
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone	29
Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone	30
Option : Module capteur PIR	32
Tableau d'affectation des entrées et sorties.....	33
Schéma de câblage	34
Exercice niveau 3 - D.1 : Utilisation du capteur PIR	35
Exercice niveau 3 - D.2 : Ouverture contrôlée à l'aide du PIR	36

Introduction

Portail coulissant

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**.



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.

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**.
- Installer l'extension **A4_Portail_Coulissant** (fichier zip) dans mBlock.
- **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.

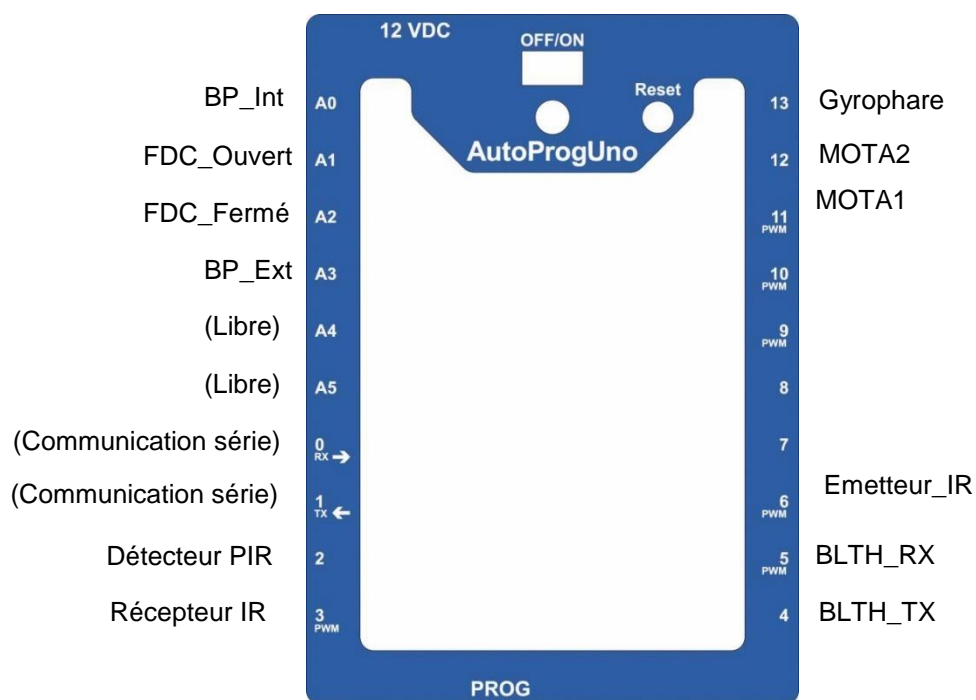
Pour l'option Bluetooth :

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

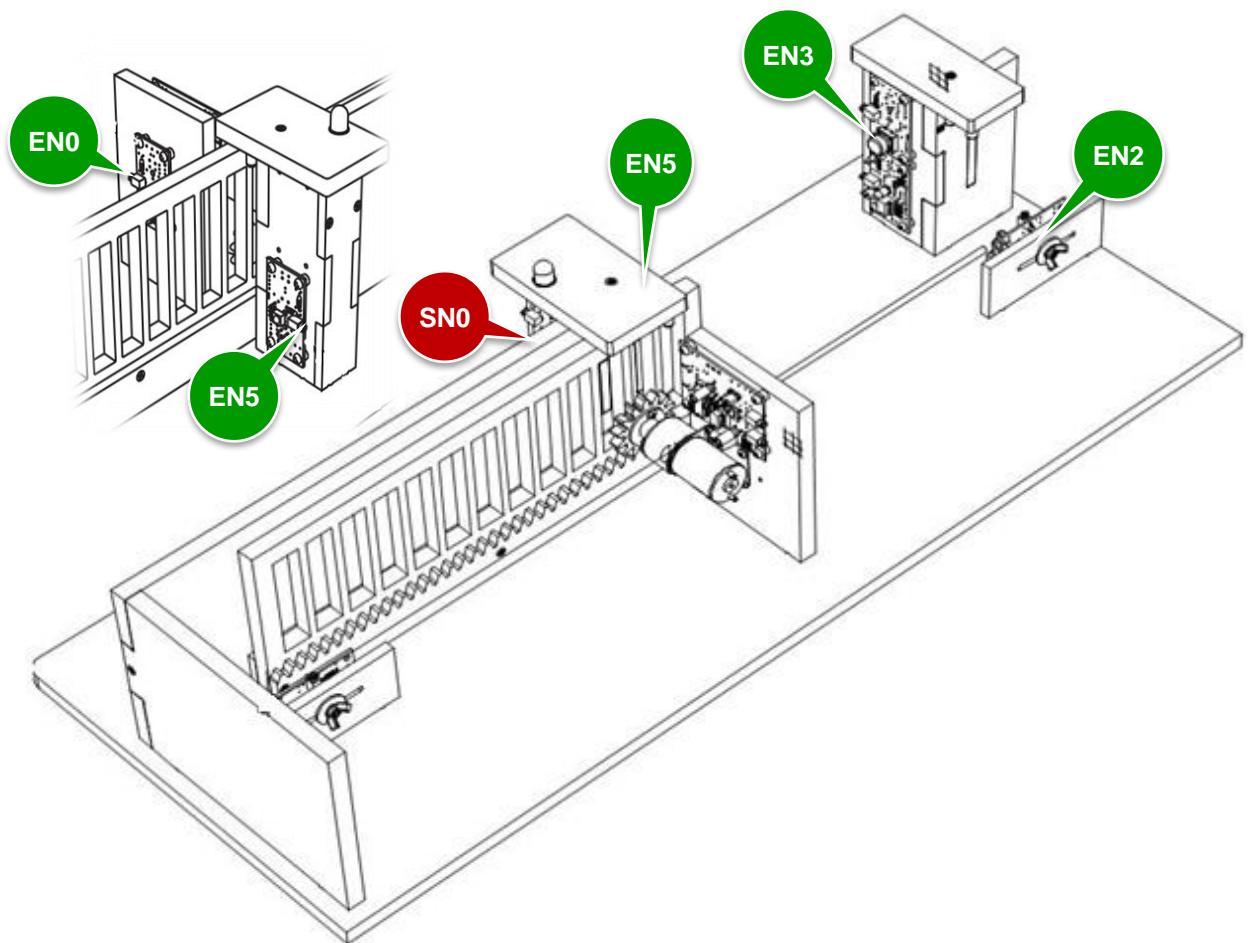
Le guide de montage ainsi que les caractéristiques techniques des composants sont détaillés dans le dossier technique disponible sur le wiki.

Tableau d'affectation des entrées et sorties

AutoProgUno	Portail coulissant	Nom mBlock
Modules capteurs pour entrées numériques		
2		
3	Récepteur infrarouge	Récepteur_IR
4	Module Bluetooth sortie (TX)	BLTH_TX
5	Module Bluetooth entrée (RX)	BLTH_RX
6	Emetteur infrarouge	
Modules actionneurs pour sorties numériques		
9		
11	MOTA-1	MOTA1
12	MOTA-2	MOTA2
13	LED gyrophare	Gyrophare
Module de communication		
1	(communication avec ordinateur)	
2	Détecteur PIR	
6		
7		
Entrées / sorties libres (A pour les analogiques)		
A0	Bouton-poussoir intérieur	BP_Int
A1	Fin de course portail ouvert	FDC_Ouvert
A2	Fin de course portail fermé	FDC_Fermé
A3	Bouton-poussoir extérieur	BP_Ext
A4		
A5		



Plan de câblage du portail coulissant

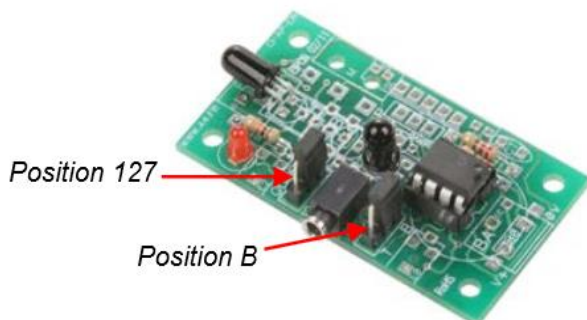


Position INT

RAPPEL

Configuration du module moteur :

Le cavalier de configuration est sur la position INT par défaut (alimentation du moteur par AutoProg).



Configuration du module émetteur infrarouge :

Les cavaliers de configuration sont sur les positions B et 127 (mode barrière IR).

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
Niveau 1 A Fichier modèle : PC_N1_A.sb2		
PC_N1_A1	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordé : -Allumage/extinction du voyant lumineux
PC_N1_A2	Répéter cette même action deux fois.	Notions de programmation abordées : -séquence d'instructions
PC_N1_A3	Répéter cette action à l'infini.	-temps d'attente -boucle infinie
Niveau 1 B Fichier modèle : PC_N1_B.sb2		
PC_N1_B1	Activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.	Fonctionnalité matérielle abordé : -Gestion du moteur -Utilisation de Bouton-poussoir
PC_N1_B2	Ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.	Notions de programmation abordées : -boucle qui dépend d'une entrée.
Niveau 1 C Fichier modèle : PC_N1_C.sb2		
PC_N1_C1	allumer le voyant lumineux à l'appui du BP.	Fonctionnalité matérielle abordé : -Gestion des modules infra-rouge
PC_N1_C2	activer le voyant lumineux lorsque la barrière infrarouge est franchie.	-Utilisation de Bouton-poussoir
PC_N1_C3	contrôler l'allumage du voyant et du moteur avec des BP.	Notions de programmation abordées : -Le test d'une entrée (si/sinon).
Niveau 1 D Fichier modèle : PC_N1_D.sb2		
PC_N1_D1	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).	Notions de programmation abordées : -Définition de variable
PC_N1_D2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	-Incrémentation de variable -Test (si/sinon) de variable
PC_N1_D3	incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.	-Test (juste si) d'entrée -Débogage.

Niveau 1 - A

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

Fichier modèle : PC_N1_A.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.

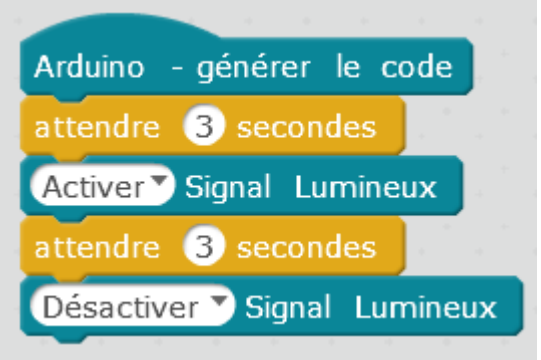
Attention à bien téléverser le programme via « Téléverser dans l'Arduino »

[Retour](#)

[Téléverser dans l'Arduino](#)

Correction :

Blocs



The screenshot shows a sequence of five mBlock code blocks for an Arduino program. The first block is a blue 'Arduino - générer le code' block. The second is a yellow 'attendre 3 secondes' block. The third is a blue 'Activer Signal Lumineux' block with a dropdown arrow. The fourth is another yellow 'attendre 3 secondes' block. The fifth is a blue 'Désactiver Signal Lumineux' block with a dropdown arrow.

Fichier mBlock : PC_N1_A1.sb2

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


Fichier modèle : PC_N1_A.sb2

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.

Correction :

Blocs



The image shows a sequence of mBlock code blocks for an Arduino project. The sequence starts with a blue block labeled 'Arduino - générer le code'. This is followed by a yellow 'attendre 3 secondes' block, a blue 'Activer Signal Lumineux' block, another yellow 'attendre 3 secondes' block, a blue 'Désactiver Signal Lumineux' block, a third yellow 'attendre 3 secondes' block, a blue 'Activer Signal Lumineux' block, a fourth yellow 'attendre 3 secondes' block, and finally a blue 'Désactiver Signal Lumineux' block. The blocks are arranged in a vertical stack, with the 'Attendre' blocks acting as separators between the 'Activer' and 'Désactiver' blocks.

Fichier mBlock : PC_N1_A2.sb2

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

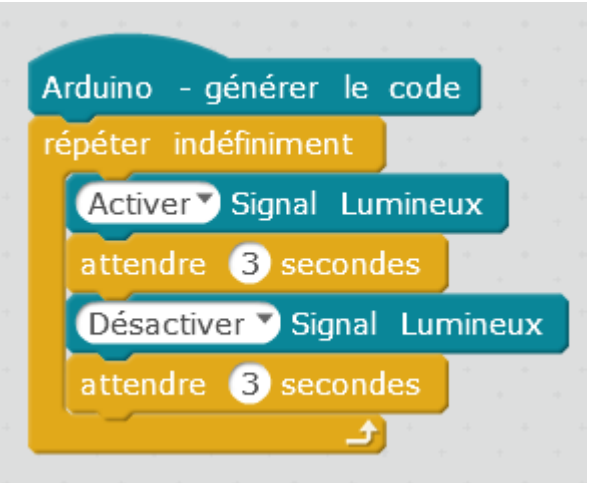
Fichier modèle : PC_N1_A.sb2

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

Notion abordée : la boucle infinie.

Correction :

Blocs



Fichier mBlock: PC_N1_A3.sb2

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

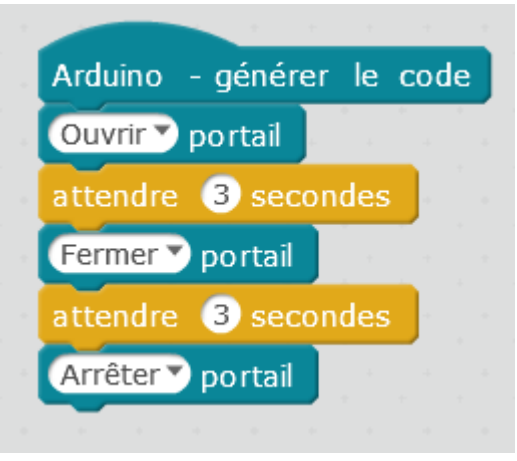
Fichier modèle : PC_N1_B.sb2

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

Notion abordée : utilisation d'un moteur.

Correction :

Blocs



Fichier mBlock : PC_N1_B1.sb2

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 le menu A4_Portail

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

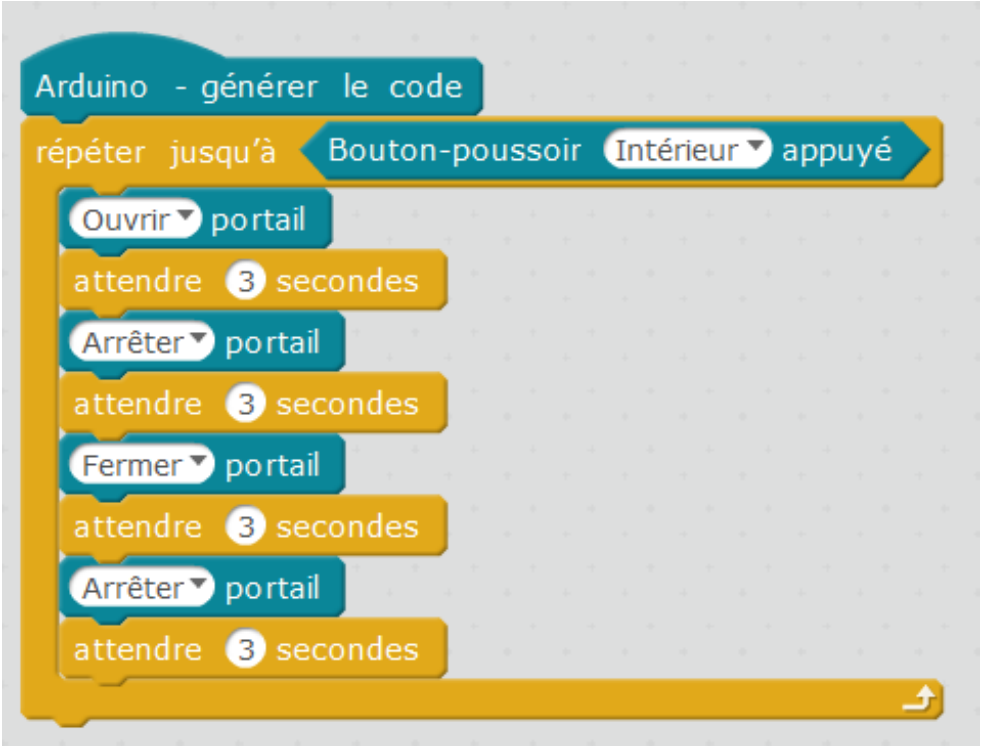
Fichier modèle : PC_N1_B.sb2

Objectif : ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.

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

Correction :

Blocs



The screenshot shows an mBlock code editor with the following blocks:

- Arduino - générer le code
- répéter jusqu'à Bouton-poussoir Intérieur appuyé
- Ouvrir portail
- attendre 3 secondes
- Arrêter portail
- attendre 3 secondes
- Fermer portail
- attendre 3 secondes
- Arrêter portail
- attendre 3 secondes

Fichier mBlock : PC_N1_B2.sb2

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).

Niveau 1 - C

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

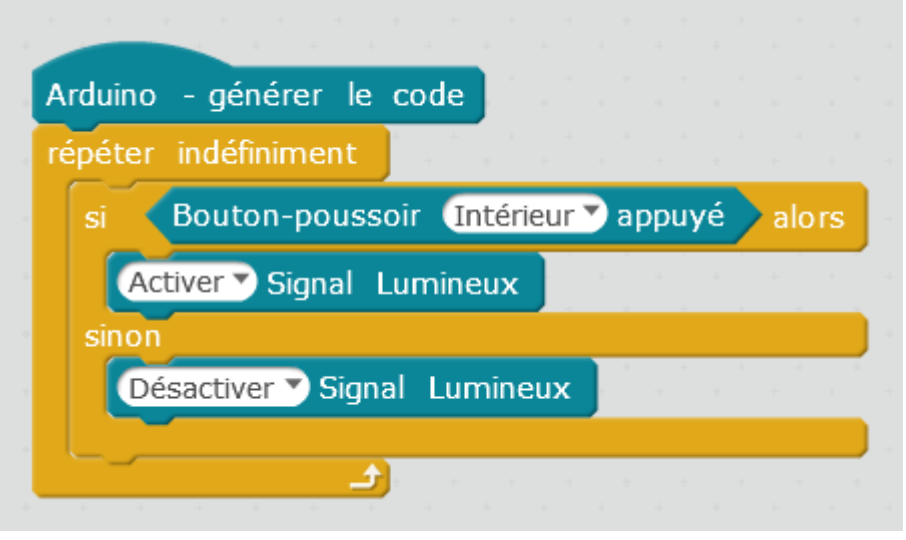
Fichier modèle : PC_N1_C.sb2

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

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

Correction :

Blocs



The screenshot shows an mBlock code editor with the following logic:

- Arduino - générer le code
- répéter indéfiniment
- si Bouton-poussoir Intérieur appuyé alors
 - Activer Signal Lumineux
- sinon
 - Désactiver Signal Lumineux

Fichier mBlock : PC_N1_C1.sb2

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

Fichier modèle : PC_N1_C.sb2

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

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

Correction :

Blocs

```
Arduino - générer le code
Activer émetteur IR
répéter indéfiniment
  si Obstacle détecté par IR alors
    Activer Signal Lumineux
  sinon
    Désactiver Signal Lumineux
```

Fichier mBlock : PC_N1_C2.sb2

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 niveau 1 – C.3 : Contrôle moteur ET voyant lumineux

Fichier modèle : PC_N1_C.sb2

Objectif : contrôler le moteur avec les boutons-poussoirs et allumer le voyant sur le franchissement de la barrière infrarouge.

Notion abordée : utilisation des commandes conditionnelles.

Correction :

Blocs

```
Arduino - générer le code
Activer émetteur IR
répéter indéfiniment
  si Bouton-poussoir Extérieur appuyé alors
    Ouvrir portail
  sinon
    si non Bouton-poussoir Intérieur appuyé alors
      Arrêter portail
    si Bouton-poussoir Intérieur appuyé alors
      Fermer portail
    sinon
      si non Bouton-poussoir Extérieur appuyé alors
        Arrêter portail
  si Obstacle détecté par IR alors
    Activer Signal Lumineux
  sinon
    Désactiver Signal Lumineux
```

Fichier mBlock : PC_N1_C3.sb2



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.

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.

Niveau 1 - D

Exercice niveau 1 - D.1 : Utilisation des variables

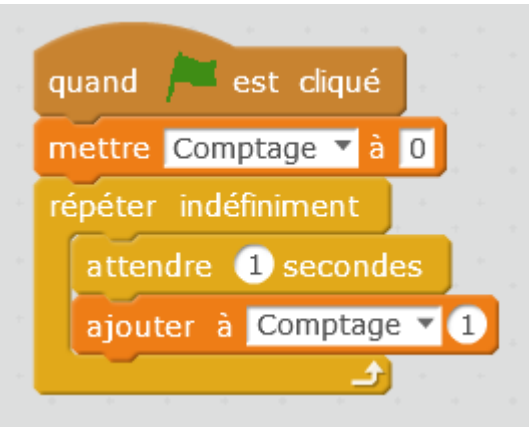
Fichier modèle : PC_N1_D.sb2

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

Notions abordées : la variable : définition et incrémentation.

Correction :

Blocs



The image shows a Scratch script for a counter. It starts with a 'when green flag is clicked' block. This is followed by a 'set Comptage to 0' block. Then, there is a 'repeat indefinitely' loop containing two blocks: 'wait 1 seconds' and 'add 1 to Comptage'. The script is contained within a box labeled 'Blocs'.

Fichier MBlock : PC_N1_D1.sb2

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

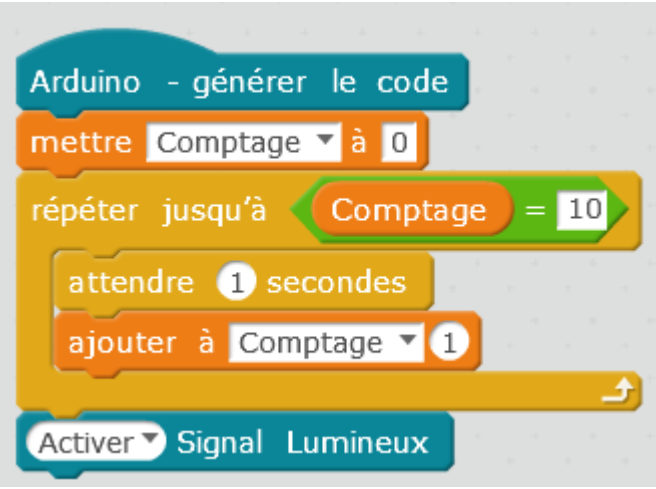
Fichier modèle : PC_N1_D.sb2

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

Correction :

Blocs



```
Arduino - générer le code
mettre Comptage à 0
répéter jusqu'à Comptage = 10
  attendre 1 secondes
  ajouter à Comptage 1
Activer Signal Lumineux
```

Fichier MBlock : PC_N1_D2.sb2

Remarque : cet exercice peut être utilisé comme un minuteur.

Exercice niveau 1 - D.3 : Tests /variables/ modules IR

Fichier modèle : PC_N1_D.sb2

Objectif : incrémenter une variable chaque seconde s'il y a un obstacle 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

```
Arduino - générer le code
mettre Comptage à 0
Activer émetteur IR
répéter indéfiniment
  si Obstacle détecté par IR alors
    ajouter à Comptage 1
    attendre 1 secondes
  si Comptage = 10 alors
    Activer Signal Lumineux
    attendre 3 secondes
    Désactiver Signal Lumineux
    mettre Comptage à 0
  ↑
```

Fichier MBlock : PC_N1_D3.sb2

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
Niveau 2 A Fichier modèle : PC_N2_A.sb2		
PC_N2_A1	Ouvrir et 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 de programmation abordées : -Utilisation des FDC
PC_N2_A2	Ouverture du portail à l'appui sur BP_Exterieur. Fermeture du portail à l'appui sur BP_Interieur.	
PC_N2_A3	Ouvrir et fermer le portail à l'aide des BP sans distinction, faire en sorte que le voyant lumineux clignote lors d'une manœuvre de la barrière.	
PC_N2_A4	Ouvrir et fermer le portail à l'aide des BP sans distinction, le voyant lumineux doit clignoter lors d'une manœuvre de la barrière. Inclure une gestion de sécurité lors la fermeture du portail.	

Niveau 2 - A

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

Objectif : ouvrir et 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 :

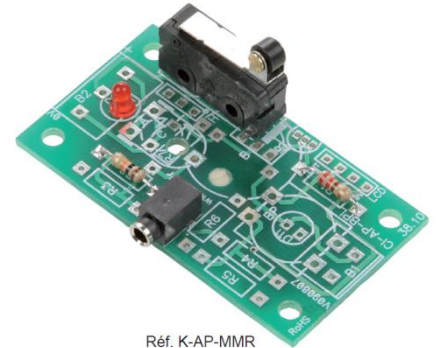
Blocs

```
Arduino - générer le code
répéter indéfiniment
  Ouvrir
  attendre 2 secondes
  Fermer
  attendre 2 secondes

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

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

Fichier MBlock : PC_N2_A1.sb2



Réf. K-AP-MMR

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

Exercice niveau 2 - 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 :

Blocs

The screenshot shows the mBlock code editor with the following blocks:

- Arduino - générer le code
- répéter indéfiniment
 - 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

Fichier MBlock : PC_N2_A2.sb2

Remarque :

Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité

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

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

Correction :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé alors
    Ouvrir
    attendre 3 secondes
    Fermer

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

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

Fichier MBlock : PC_N2_A3.sb2

Remarque : La fonction **Basculer** permet de passer d'un état logique à un autre.

Exercice niveau 2 - A.4 : Contrôle ouverture/fermeture avec BP et signal de sécurité

Objectif : ouvrir et fermer le portail à l'aide des BP sans distinction, le voyant lumineux 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 :

Blocs

```
Arduino - générer le code
Activer émetteur IR
Fermer
répéter indéfiniment
  si Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé alors
    si Fin de course Ouverture activé alors
      Fermer
    sinon
      Ouvrir
  fin
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 secondes
    si Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé alors
      Fermer portail
  fin
  Activer Signal Lumineux
  attendre 0.1 secondes
  Désactiver Signal Lumineux
  attendre 0.1 secondes
  Arrêter portail
définir Ouvrir
  Ouvrir portail
  répéter jusqu'à Fin de course Ouverture activé
    Activer Signal Lumineux
    attendre 0.1 secondes
    Désactiver Signal Lumineux
    attendre 0.1 secondes
  fin
  Arrêter portail
```

Fichier MBlock : PC_N2_A4.sb2

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
Niveau 1 B		
PC_N3_B1	Contrôler l'ouverture et la fermeture du portail à l'aide de 2 boutons présent sur l'application Android.	Fonctionnalité matérielle abordée : - module Bluetooth Notions de programmation abordées : - liaison série (hserin/hserout)
PC_N3_B2	Ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.	
PC_N3_B3	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.	
PC_N3_B4	Gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.	
Niveau 1 D		
PC_N3_D1	Allumer le voyant lumineux lorsqu'une présence est détectée par le capteur PIR.	Fonctionnalité matérielle abordée : Capteur_PIR
PC_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 et TX.

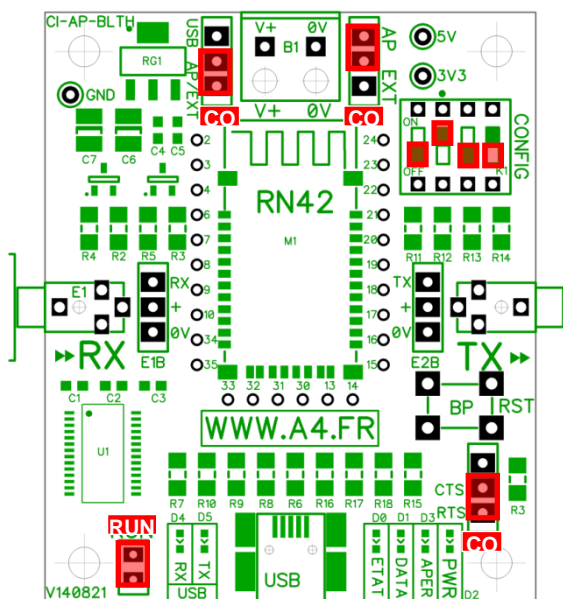
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

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.

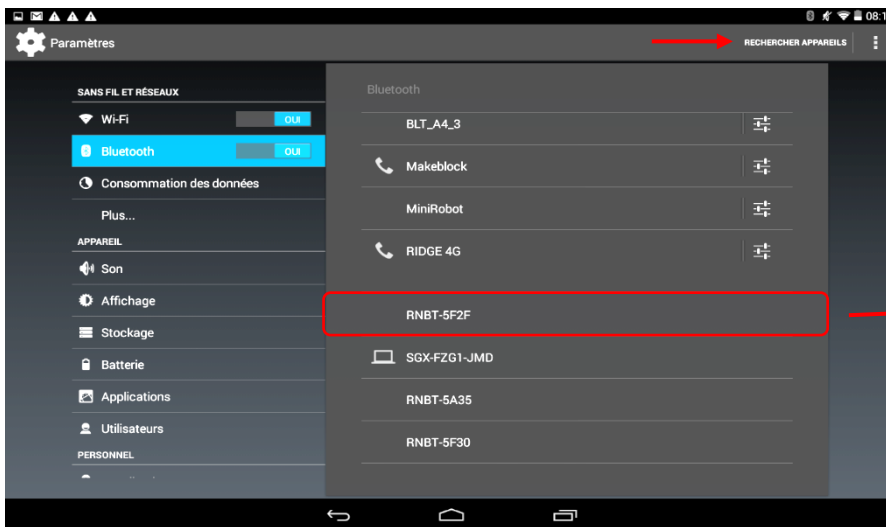
Témoins lumineux

- 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.



Tableau d'affectation des entrées et sorties

AutoProgUno	Portail coulissant	mBlock	Nom mBlock
Modules capteurs pour entrées numériques			
5	Récepteur barrière infrarouge	C.5	
4	Capteur détection de présence (option)	C.4	
3	Bouton poussoir extérieur	C.3	
2	Capteur de fin de course fermeture du portail	C.2	
1	Capteur de fin de course ouverture du portail	C.1	
0	Bouton poussoir intérieur	C.0	
Modules actionneurs pour sorties numériques			
7	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.7	Moteur_A2
6	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.6	Moteur_A1
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	(libre)	B.2	
1	Emetteur barrière infrarouge	B.1	Emetteur_IR
0	Module signal LED jaune	B.0	voyant_Lumineux
Module de communication			
7	Communication Bluetooth envoi de données	C.7	
6	Communication Bluetooth réception de données	C.6	
Entrées / sorties libres (A pour les analogiques)			

Exercice niveau 3 - 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.

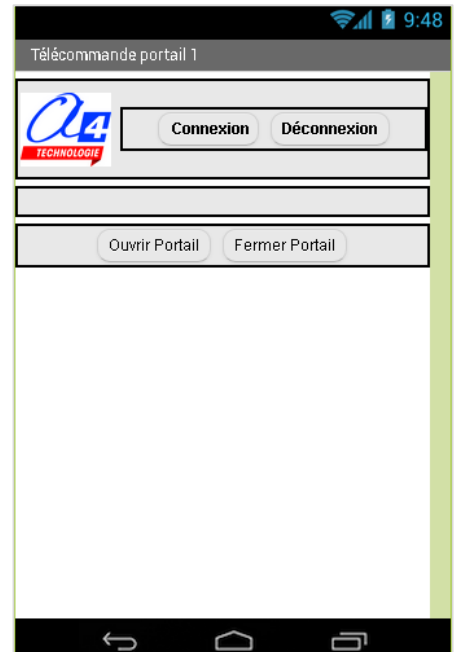
Notion abordée : réception de données Bluetooth envoyées par un Smartphone.

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 :

```
Blocs

Arduino - générer le code
Activer émetteur IR
Fermer
répéter indéfiniment
  mettre BLTH à Donnée Bluetooth reçue
  si BLTH = 1 alors
    Ouvrir
  sinon
    si BLTH = 2 alors
      Fermer
  si Bouton-poussoir Intérieur appuyé alors
    Ouvrir
  si Bouton-poussoir Extérieur appuyé alors
    Fermer

définir Fermer
Fermer portail
répéter jusqu'à Fin de course Fermeture activé
  si Recepteur_IR = 1 alors
    Ouvrir
    attendre jusqu'à Recepteur_IR = 0
    attendre 2000 secondes
    Activer Signal Lumineux
    attendre 0.1 secondes
    Désactiver Signal Lumineux
    attendre 0.1 secondes
  Arrêter portail

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

Fichier MBlock : PC_N3_B1.sb2
```

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

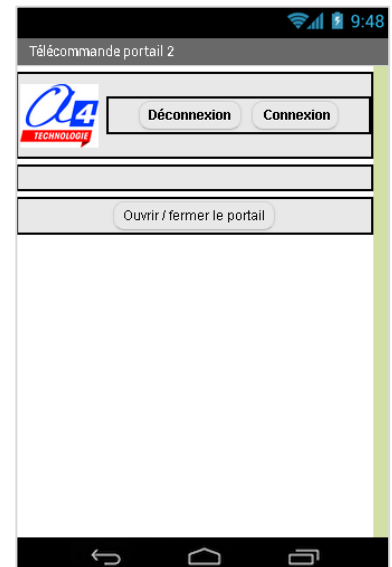
Objectif : ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.

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 :

Blocs

```
Arduino - générer le code
Activer émetteur IR
Fermer
répéter indéfiniment
  mettre 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 secondes
  Désactiver Signal Lumineux
  attendre 0.1 secondes
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 secondes
  Désactiver Signal Lumineux
  attendre 0.1 secondes
Arrêter portail
```

Fichier MBlock : PC_N3_B2.sb2

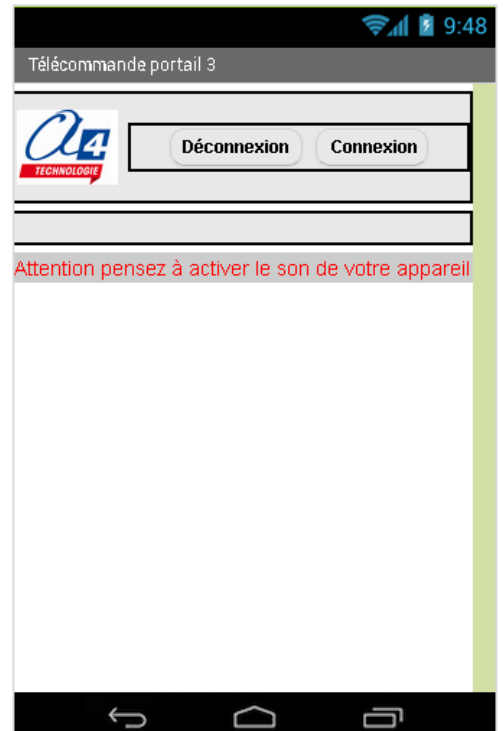
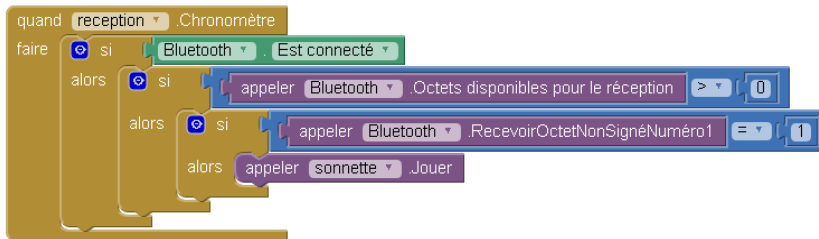
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone

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

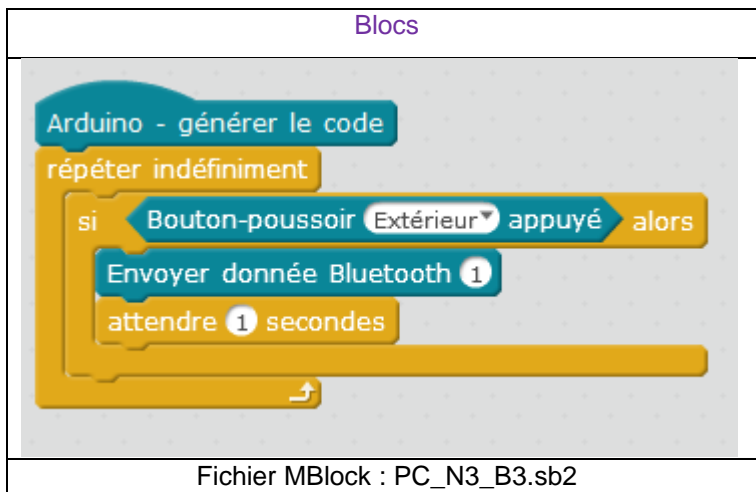
Notion abordée : envoyer des informations à un Smartphone par Bluetooth.

Application Android : Portail_3.apk

Fichier App Inventor : Portail_3.aia



Correction :



Exercice niveau 3 - 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.

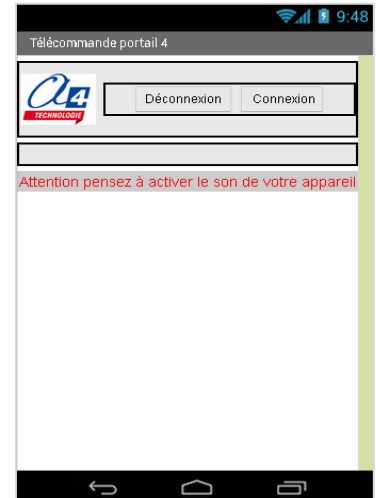
Notion abordée : envoyer et recevoir des informations à l'aide du module Bluetooth à une application.

Application Android : Portail_4.apk

App Inventor : Portail_4.aia

```
quand reception Chronomètre
faire
  si Bluetooth Est connecté
  alors
    si appeler Bluetooth .Octets disponibles pour la réception > 0
    alors
      si appeler Bluetooth RecevoirOctetNonSignéNuméro1
      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 :

Blocs

```
Arduino - générer le code
Activer émetteur IR
Fermer
répéter indéfiniment
  mettre BLTH à Donnée Bluetooth reçue
  si Bouton-poussoir Extérieur appuyé alors
    Envoyer donnée Bluetooth 1
    attendre 1 secondes
  si BLTH = 1 alors
    Ouvrir
    attendre 3 secondes
    Fermer
```


The image shows two scripts in the mBlock code editor. The first script, titled 'Ouvrir', starts with a 'définir Ouvrir' block, followed by 'Ouvrir portail'. It then enters a 'répéter jusqu'à' loop with the condition 'Fin de course Ouverture activé'. Inside the loop, it performs the following steps: 'Activer Signal Lumineux', 'attendre 0.1 secondes', 'Désactiver Signal Lumineux', and 'attendre 0.1 secondes'. After the loop, it executes 'Arrêter portail'. The second script, titled 'Fermer', starts with 'définir Fermer', followed by 'Fermer portail'. It enters a 'répéter jusqu'à' loop with the condition 'Fin de course Fermeture activé'. Inside this loop, there are two 'si' (if) conditions. The first is 'si Obstacle détecté par IR alors', which leads to 'Ouvrir' and 'attendre jusqu'à Fin de course Ouverture activé'. The second is 'si Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé alors', which leads to 'Fermer portail'. After the loop, it performs 'Activer Signal Lumineux', 'attendre 0.1 secondes', 'Désactiver Signal Lumineux', and 'attendre 0.1 secondes'. Finally, it executes 'Arrêter portail'. At the bottom right of the code area, there are icons for search, equals, and refresh.

Fichier MBlock : PC_N3_B4.sb2

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.

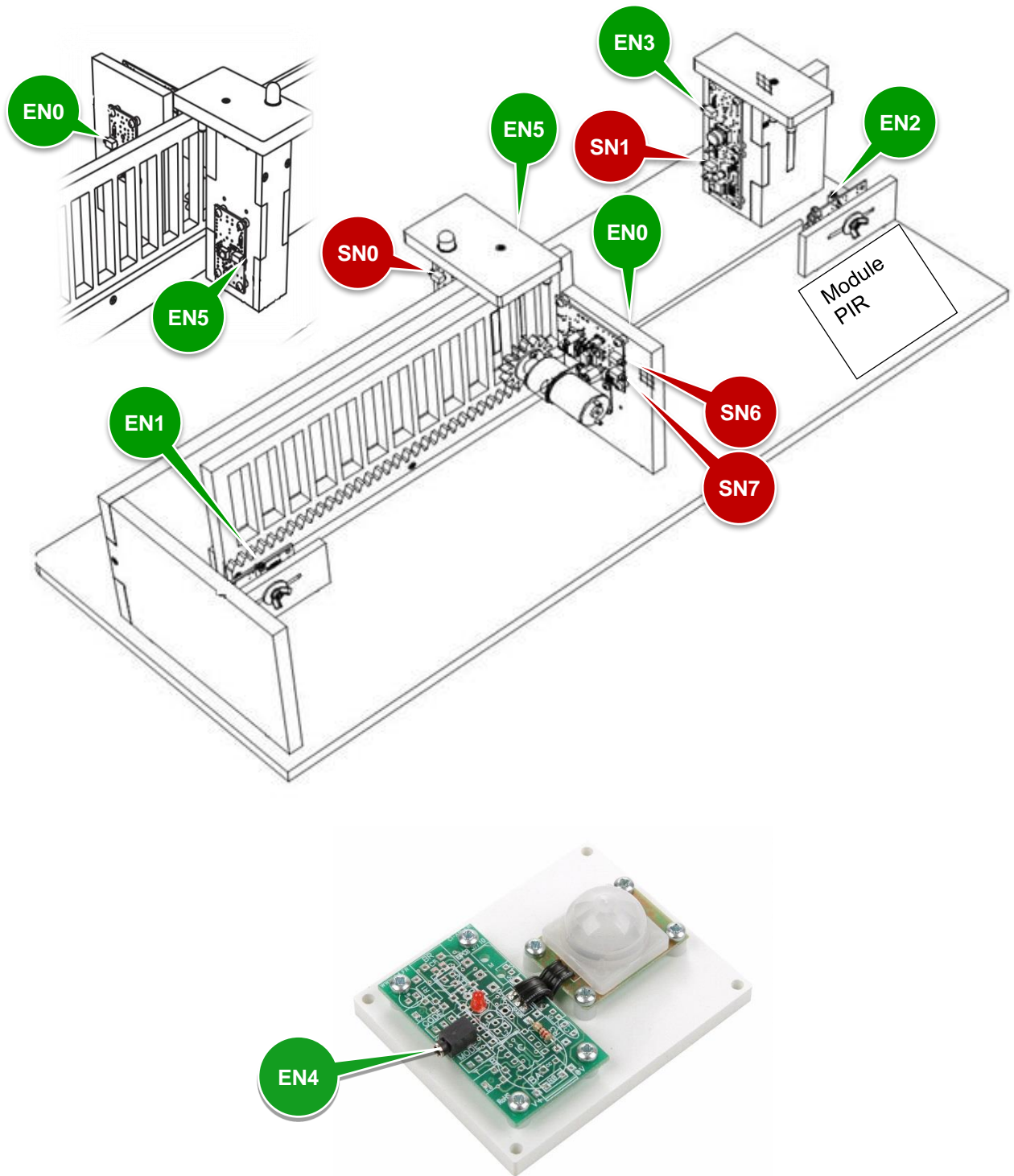
Par ailleurs, le capteur est sensible aux variations de températures brutales, aux vibrations ou aux chocs importants. Il ne faut pas l'exposer à la lumière directe du soleil, à l'air pulsé d'un radiateur ou d'un climatiseur. Il est conçu pour une utilisation en intérieur ; pour une utilisation en extérieur,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



Tableau d'affectation des entrées et sorties

AutoProgUno	Portail coulissant	mBlock	Nom mBlock
Modules capteurs pour entrées numériques			
5	Récepteur barrière infrarouge	C.5	
4	Capteur détection de présence (option)	C.4	
3	Bouton poussoir extérieur	C.3	
2	Capteur de fin de course fermeture du portail	C.2	
1	Capteur de fin de course ouverture du portail	C.1	
0	Bouton poussoir intérieur	C.0	
Modules actionneurs pour sorties numériques			
7	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.7	Moteur_A2
6	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.6	Moteur_A1
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	(libre)	B.2	
1	Emetteur barrière infrarouge	B.1	Emetteur_IR
0	Module signal LED jaune	B.0	voyant_Lumineux
Module de communication			
7	Communication Bluetooth envoi de données	C.7	
6	Communication Bluetooth réception de données	C.6	
Entrées / sorties libres (A pour les analogiques)			

Schéma de câblage




Exercice niveau 3 – D.1 : Utilisation du capteur PIR

Objectif : allumer le voyant lumineux lorsqu'une présence est détectée par le capteur PIR.

Notion abordée : utilisation du capteur de présence PIR.

Correction :

Blocs



```
Arduino - générer le code
répéter indéfiniment
  si Présence détectée par capteur PIR alors
    Activer Signal Lumineux
    attendre 0.1 secondes
    Désactiver Signal Lumineux
  attendre 0.1 secondes
  sinon
    Désactiver Signal Lumineux
```

Fichier MBlock : PC_N3_D1.sb2

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

Exercice niveau 3 – D.2 : Ouverture contrôlée à l'aide du PIR

Objectif : reprendre le programme PC_N2_A4 et ajouter la detection_PIR pour ouvrir le portail.

Correction :

Blocs

The code is organized into several functional blocks:

- Initialization:** 'Arduino- générer le code', 'Activer émetteur IR', and 'Fermer'.
- Main Loop (répéter indéfiniment):**
 - Condition: 'Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé'.
 - If 'Fin de course Ouverture activé': 'Fermer'.
 - Else: 'Ouvrir'.
 - Condition: 'Présence détectée par capteur PIR'.
 - If 'Fin de course Fermeture activé': 'Ouvrir'.
- Function: définir Fermer**
 - 'Fermer portail'.
 - Loop 'répéter jusqu'à Fin de course Fermeture activé':
 - If 'Obstacle détecté par IR': 'Ouvrir', 'attendre jusqu'à Fin de course Ouverture activé', 'attendre 2 secondes'.
 - Else: 'Bouton-poussoir Extérieur appuyé ou Bouton-poussoir Intérieur appuyé' → 'Fermer portail'.
 - 'Activer Signal Lumineux', 'attendre 0,1 secondes', 'Désactiver Signal Lumineux', 'attendre 0,1 secondes'.
 - 'Arrêter portail'.
- Function: définir Ouvrir**
 - 'Ouvrir portail'.
 - Loop 'répéter jusqu'à Fin de course Ouverture activé':
 - 'Activer Signal Lumineux', 'attendre 0,1 secondes', 'Désactiver Signal Lumineux', 'attendre 0,1 secondes'.
 - 'Arrêter portail'.

Fichier MBlock : PC_N3_D2.sb2



www.a4.fr

Concepteur et fabricant de matériels pédagogiques