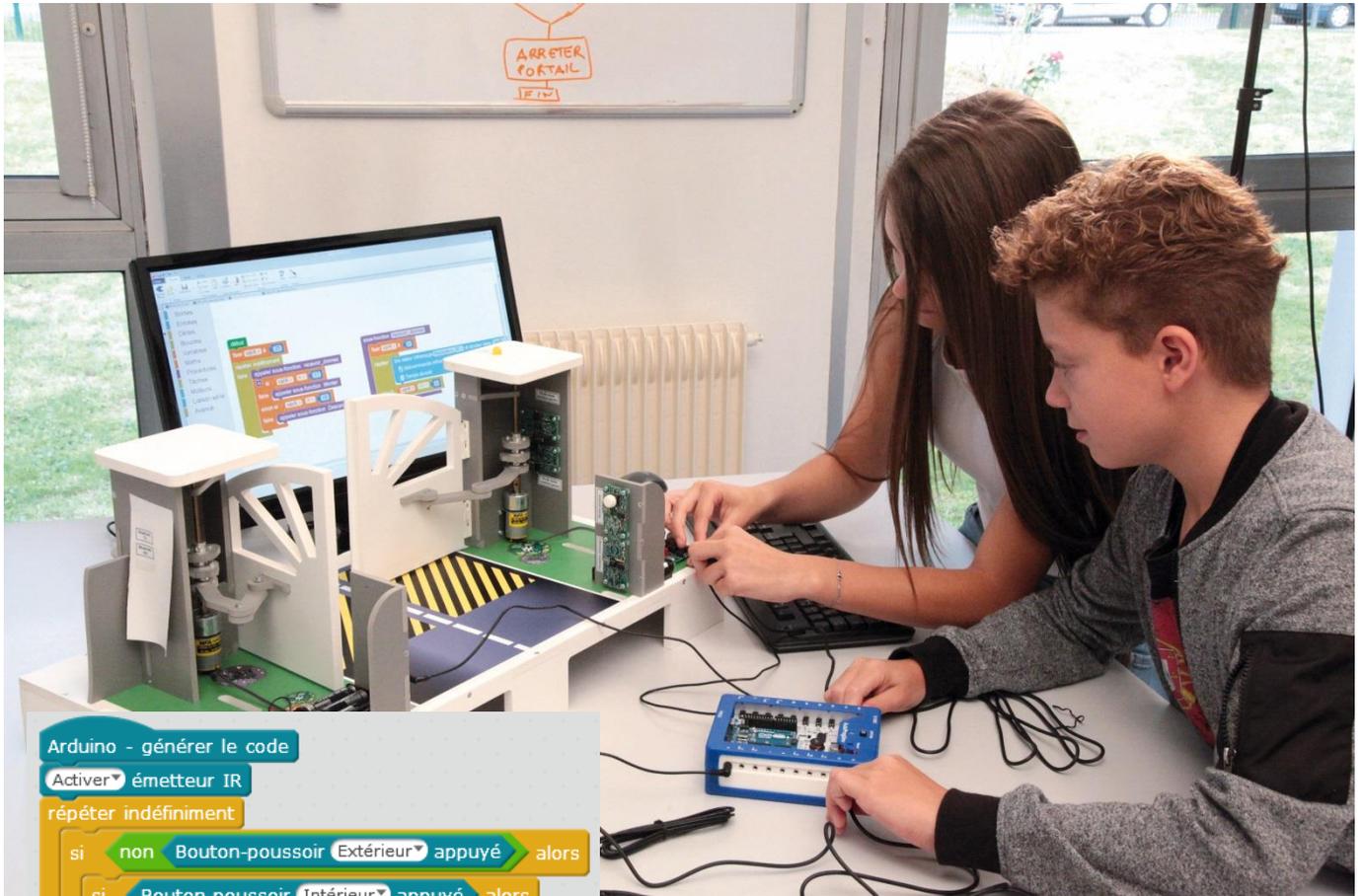


Portail battant deux vantaux

Maquette programmable avec mBlock



Arduino - générer le code

Activer émetteur IR

répéter indéfiniment

```
si non Bouton-poussoir Extérieur appuyé alors
si Bouton-poussoir Intérieur appuyé alors
  Ouvrir portail
sinon
  Arrêter portail
si non Bouton-poussoir Intérieur appuyé alors
si Bouton-poussoir Extérieur appuyé alors
  Fermer portail
sinon
  Arrêter portail
si Présence détectée par capteur PIR alors
  Activer Signal Lumineux
sinon
  Désactiver Signal Lumineux
```

Ressources disponibles pour le projet

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

Portail battant deux vantaux

- 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

SOMMAIRE

Introduction	2
Portail battant deux vantaux	2
Les environnements de programmation graphique	2
Le dossier	2
Les fiches exercices	3
Prérequis	3
Tableau d'affectation des entrées et sorties	4
Programmation version de base niveau 1	5
Niveau 1 - A.....	6
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
Niveau 1 - B.....	9
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur.....	9
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que	10
Exercice niveau 1 - B.3 : Utilisation d'un autre moteur.....	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
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 : ouverture/fermeture des deux battants	21
Exercice niveau 2 - A.4 : Contrôle ouverture/fermeture avec BP et signal de sécurité	22
Exercice niveau 2 - A.5 : Amélioration de la vitesse	23
Exercice niveau 2 - B.1 : Gestion automatique du portail via capteurs IR	24
Programmation niveau 3.....	25
Option : Module Bluetooth.....	26
Configuration	26
Mise en place des programmes et procédure de connexion.....	27
Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth	28
Exercice niveau 3 - B.2 : Contrôle du portail par Smartphone	30
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone	32
Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone	33

Introduction

Portail battant deux vantaux

La maquette portail battant deux vantaux (BE-APORT-2BAT) est une reproduction homothétique d'un portail battant automatisé réel : boutons d'activation, capteurs fin de course, barrière optique, clignotant de sécurité, signal sonore, etc. 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 du portail:

- 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 : télécommande infrarouge, module Bluetooth.

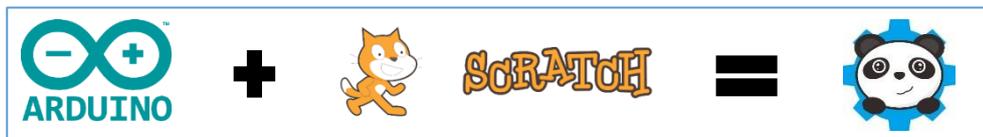
Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette 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_2_vantaux_Extension** (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 du module Bluetooth.

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 :

- **PB2** pour Portail Battant deux vantaux
- **N** : niveau de programmation 1-2-3
- **A-B-C** : jeu d'instructions du plus simple au plus avancé

Exemple : PB2_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_2_vantaux_Extension** (fichier zip) dans mBlock.
- **Maquette** Portail battant deux vantaux (Réf. BE-APORT-2BAT).
- **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.

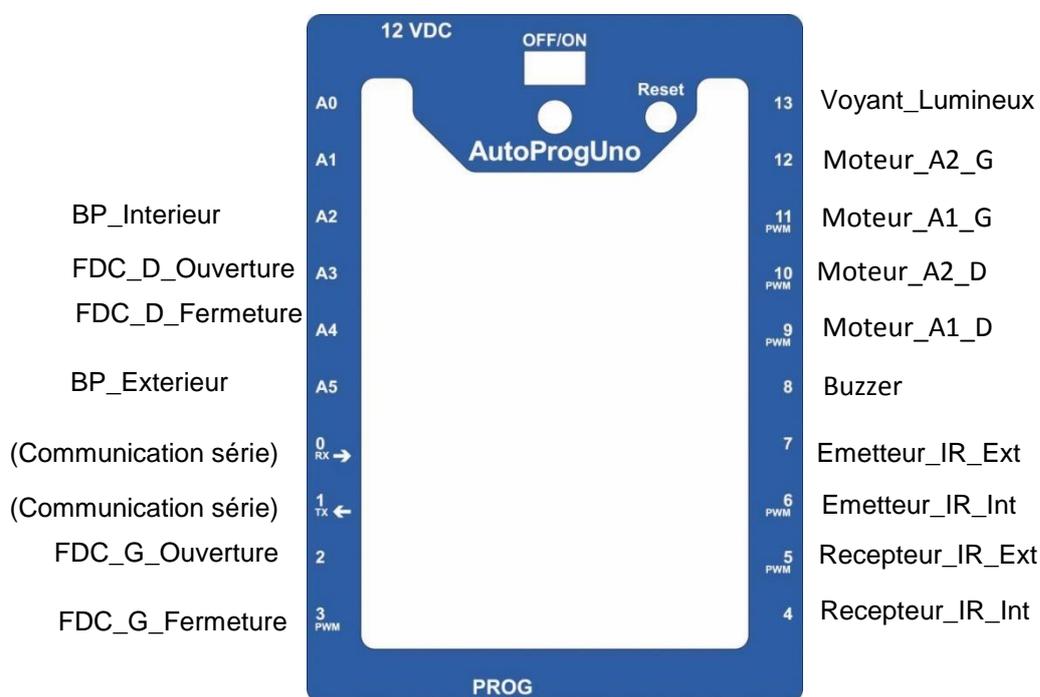
Pour l'option Bluetooth :

- **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		Nom mBlock
MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
2	Fin de course ouverture gauche	FDC_G_Ouverture
3	Fin de course fermeture gauche	FDC_G_Fermeture
4	Communication IR Intérieur	Recepteur_IR_Int
5	Communication IR Extérieur	Recepteur_IR_Ext
MODULES ACTIONNEURS POUR SORTIES NUMÉRIQUES		
6	Emetteur IR intérieur	Emetteur_IR_Int
7	Emetteur IR extérieur	Emetteur_IR_Ext
8	Module buzzer	Buzzer
9	Branche A1 moteur de droite	Moteur_A1_D
10	Branche A2 moteur de droite	Moteur_A2_D
11	Branche A1 moteur de gauche	Moteur_A1_G
12	Branche A2 moteur de gauche	Moteur_A2_G
13	Module lumineux	Voyant_Lumineux
MODULE DE COMMUNICATION		
0	(communication avec ordinateur)	
1		
ENTRÉES / SORTIES LIBRES (A pour les analogiques)		
A0		
A1		
A2	Bouton poussoir intérieur	BP_Interieur
A3	Fin de course ouverture droite	FDC_D_Ouverture
A4	Fin de course fermeture droite	FDC_D_Fermeture
A5	Bouton poussoir extérieur	BP_Exterieur



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 portail, 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
Niveau 1 A		
PB2_N1_A1.sb2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordée : -Allumage/extinction du voyant lumineux Notions de programmation abordées : -séquence d'instructions -temps d'attente -boucle infinie
PB2_N1_A2.sb2	Répéter cette même action deux fois	
PB2_N1_A3.sb2	Répéter cette action à l'infini	
Niveau 1 B		
PB2_N1_B1.sb2	Activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.	Fonctionnalité matérielle abordée : -Gestion du moteur -Utilisation de Bouton-poussoir Notions de programmation abordées : -boucle qui dépend d'une entrée
PB2_N1_B2.sb2	Ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.	
PB2_N1_B3.sb2	Utiliser un autre moteur.	
Niveau 1 C		
PB2_N1_C1.sb2	Allumer le voyant lumineux à l'appui du BP.	Fonctionnalité matérielle abordée : -Gestion des modules infrarouges -Utilisation de Bouton-poussoir Notions de programmation abordées : -Le test d'une entrée (si/sinon)
PB2_N1_C2.sb2	Activer le voyant lumineux lorsque la barrière infrarouge est franchie.	
PB2_N1_C3.sb2	Contrôler l'allumage du voyant et du moteur avec des BP.	
PB2_N1_C4.sb2	Utilisation d'un buzzer.	
Niveau 1 D		
PB2_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 -Incrémentation de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage
PB2_N1_D2.sb2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	
PB2_N1_D3.sb2	Incrémenter une variable à l'appui d'un bouton poussoir.	

Niveau 1 - A

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

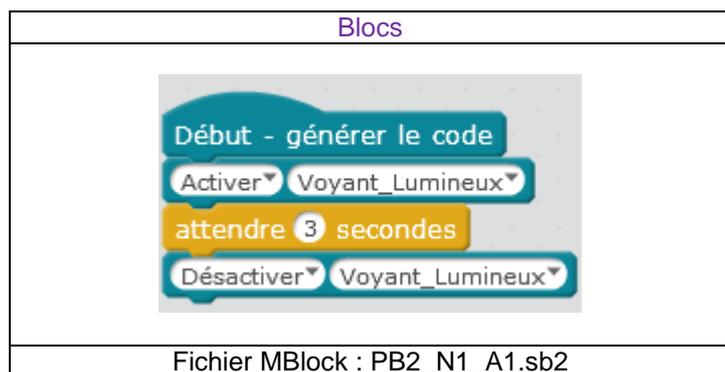
Objectif : allumer le voyant lumineux pendant 3 secondes puis l'éteindre.

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.

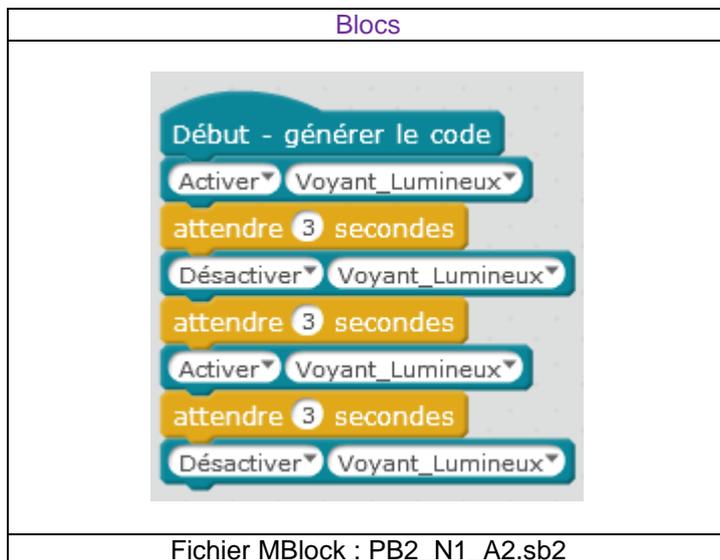
Notions abordées : séquence d'instructions, activation / désactivation d'une sortie, temps d'attente.

Instruction utilisées :



Correction :

Blocs

La correction est un programme mBlock composé des blocs suivants : 'Début - 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).

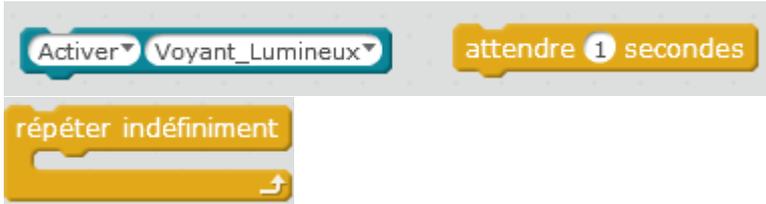
Fichier MBlock : PB2_N1_A2.sb2

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

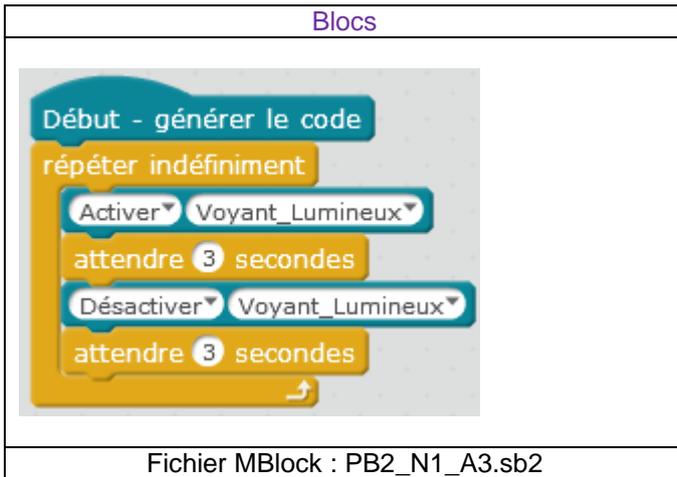
Objectif : faire clignoter le voyant lumineux avec une période de 6 secondes indéfiniment.

Notion abordée : la boucle infinie.

Instruction 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.

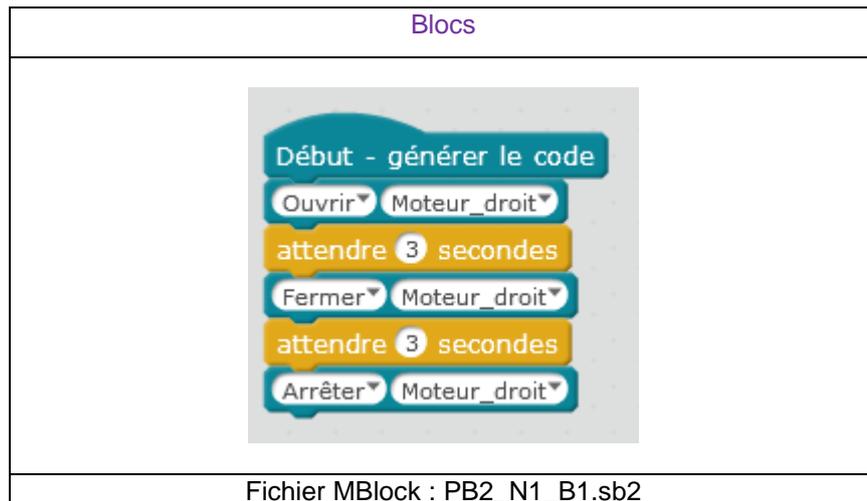
Objectif : activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.

Notion abordée : utilisation d'un moteur.

Instructions utilisées :



Correction :



ATTENTION : pour cet exercice il est recommandé de placer la barrière au milieu de sa course pour éviter tout dommage.

Il faut également activer le moteur à l'aide de l'interrupteur ON/OFF sur le module moteur (Une LED rouge indique si le moteur est allumé).

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

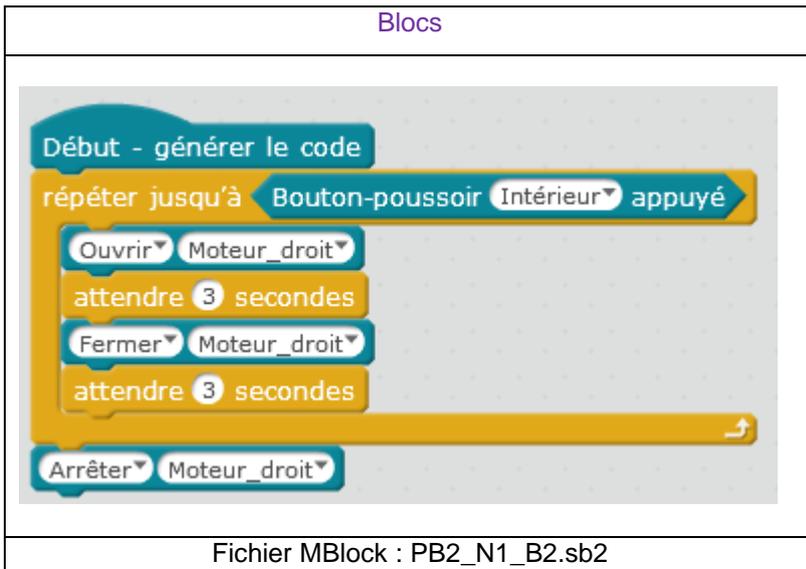
Objectif : ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.

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

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).

Exercice niveau 1 - B.3 : Utilisation d'un autre moteur

Objectif : ouvrir et fermer les deux battants du portail puis les arrêter

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

Instructions utilisées :



Correction :

Blocs

Le bloc de correction est un cadre contenant une série de blocs de programmation mBlock. Il commence par un bloc 'Début - générer le code'. Ensuite, il y a deux blocs 'Ouvrir' : 'Moteur_droit' et 'Moteur_gauche'. Cela est suivi d'un bloc 'attendre' de 3 secondes. Ensuite, deux blocs 'Fermer' : 'Moteur_droit' et 'Moteur_gauche'. Cela est suivi d'un autre bloc 'attendre' de 3 secondes. Enfin, deux blocs 'Arrêter' : 'Moteur_droit' et 'Moteur_gauche'.

Remarque : Les deux battants peuvent ne pas avoir la même vitesse, afin de régler la vitesse d'un moteur, tournez le potentiomètre situé sur un module moteur.

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 : PB2_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 :

Blocs

Fichier MBlock : PB2_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.

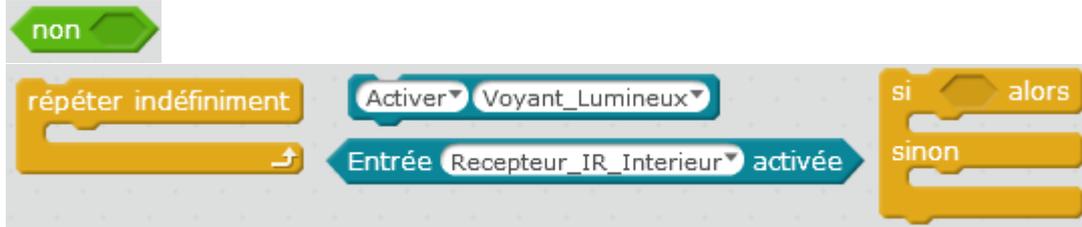
Lorsque un obstacle franchi la barrière IR, le signal n'est plus transmis et l'entrée du récepteur IR devient active. Si le programme ne marche pas, mettre le cavalier MODE de l'émetteur IR sur B et le cavalier du code sur 127.

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

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

Notion abordée : utilisation des commandes conditionnelles.

Instructions utilisées :



Correction :

Blocs

```

Début - générer le code
Activer Emetteur_IR_Interieur
répéter indéfiniment
  si non Bouton-poussoir Extérieur appuyé alors
    si Bouton-poussoir Intérieur appuyé alors
      Ouvrir Moteur_droit
    sinon
      Arrêter Moteur_droit
  si non Bouton-poussoir Intérieur appuyé alors
    si Bouton-poussoir Extérieur appuyé alors
      Fermer Moteur_droit
    sinon
      Arrêter Moteur_droit
  si Entrée Recepteur_IR_Interieur activée alors
    Activer Voyant_Lumineux
  sinon
    Désactiver Voyant_Lumineux

```

Fichier MBlock : PB2_N1_C3.sb2

Remarque : Le bloc  permet ici d'empêcher qu'on entre dans deux conditions en même temps.

Niveau 1 - D

Exercice niveau 1 - D.1 : Utilisation des variables

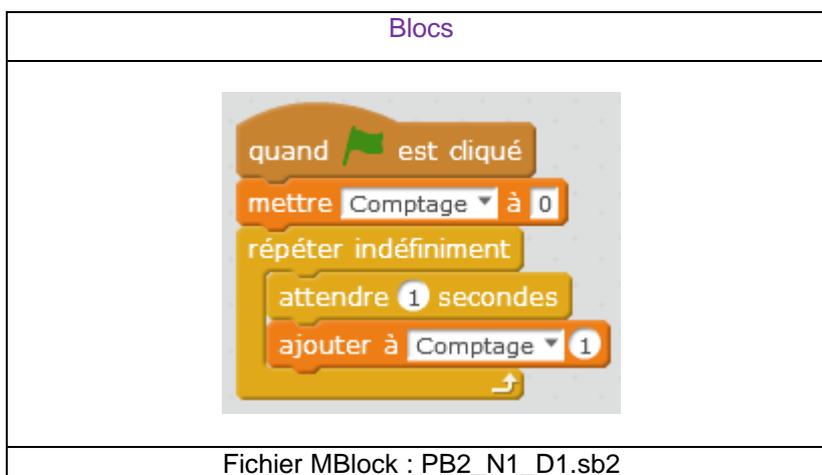
Objectif : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC.

Notions abordées : la variable : définition et incrémentation

Instructions utilisées :



Correction :



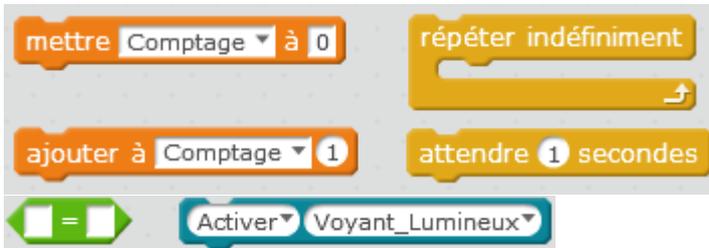
Remarques : la commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

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

A code block titled 'Début - générer le code' containing the following logic: 'mettre Comptage à 0', 'répéter jusqu'à Comptage = 10' (with a loop body containing 'attendre 1 secondes' and 'ajouter à Comptage 1'), and 'Activer Voyant_Lumineux'.

Fichier MBlock : PB2_N1_D2.sb2

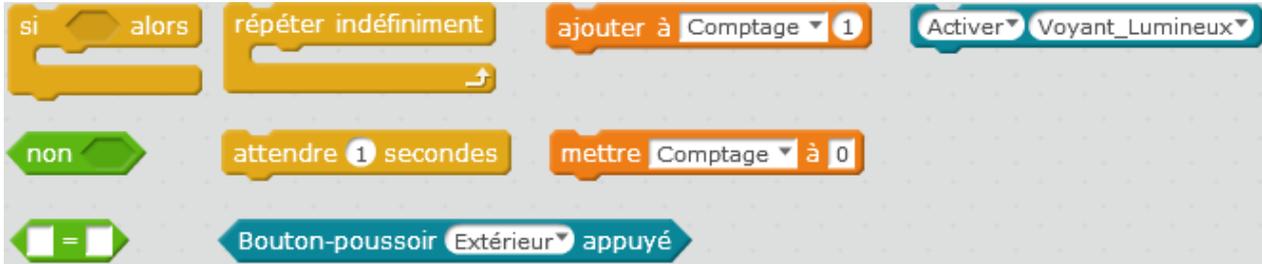
Remarque : cet exercice peut être utilisé comme un minuteur.

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

Objectif : incrémenter une variable à chaque appui sur un bouton poussoir. 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

Instructions utilisées :



Correction :

Blocs

```

Début - générer le code
mettre Comptage à 0
répéter indéfiniment
  si Bouton-poussoir Intérieur appuyé alors
    ajouter à Comptage 1
    attendre jusqu'à non Bouton-poussoir Intérieur appuyé
  si Comptage = 10 alors
    Activer Voyant_Lumineux
    attendre 3 secondes
    Désactiver Voyant_Lumineux
    mettre Comptage à 0

```

Fichier MBlock : PB2_N1_D3.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
Niveau 2 A		
PB2_N2_A1.sb2	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 fins de course
PB2_N2_A2.sb2	Ouverture du portail à l'appui sur BP_Exterieur. Fermeture du portail à l'appui sur BP_Interieur.	
PB2_N2_A3.sb2	Gestion des deux battants.	
PB2_N2_A4.sb2	Ouvrir et fermer le portail à l'aide des BP sans distinction.	
PB2_N2_A5.sb2	Amélioration de la vitesse d'ouverture et de fermeture du portail.	
Niveau 2 B		
PB2_N2_B1.sb2	Gestion du portail grâce aux capteurs IR.	Automatisation de la maquette

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

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

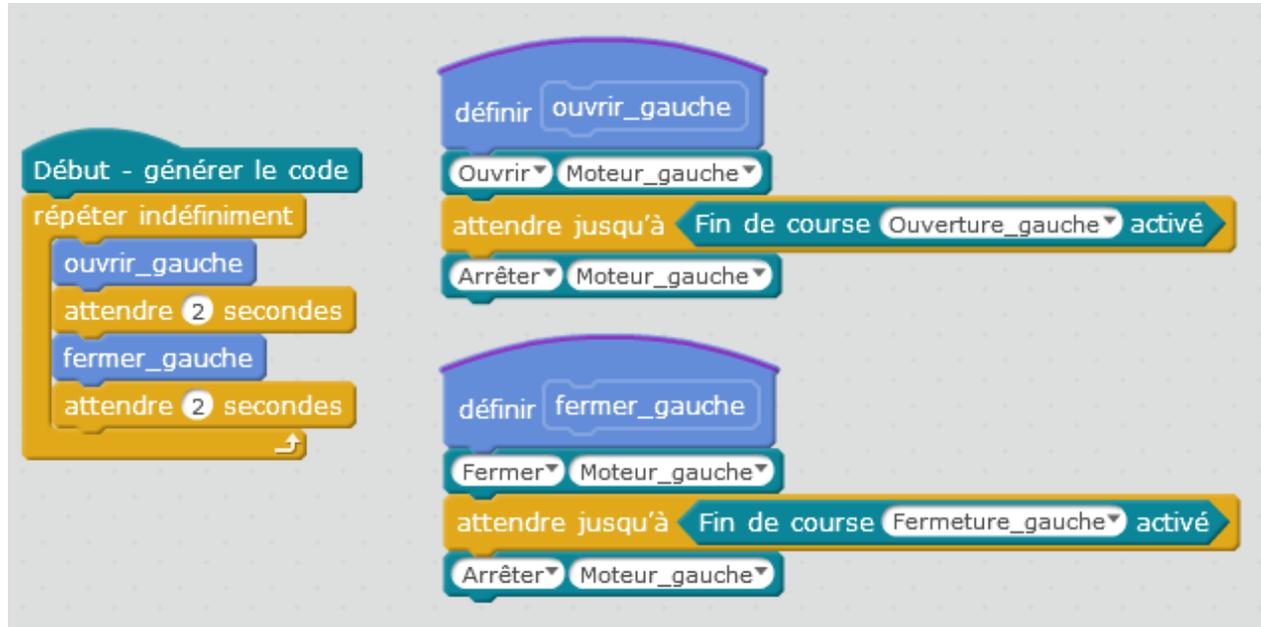
Notions abordées : utilisation des fins de course, procédures (sous-fonctions)

Instructions utilisées :



Correction :

Blocs

Le code de programmation est présenté dans un environnement graphique. Il commence par un bloc 'Début - générer le code'. Ensuite, un bloc 'répéter indéfiniment' encadre deux sous-fonctions. La première sous-fonction, 'ouvrir_gauche', est définie avec les blocs suivants : 'Ouvrir' Moteur_gauche, 'attendre jusqu'à' Fin de course Ouverture_gauche activé, et 'Arrêter' Moteur_gauche. La seconde sous-fonction, 'fermer_gauche', est définie avec : 'Fermer' Moteur_gauche, 'attendre jusqu'à' Fin de course Fermeture_gauche activé, et 'Arrêter' Moteur_gauche. Dans le code principal, les sous-fonctions sont appelées 'ouvrir_gauche' et 'fermer_gauche', chacune précédée d'un bloc 'attendre 2 secondes'.

Fichier MBlock : PB2_N2_A1.sb2

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

Les capteurs fin de course peuvent parfois se décaler (si on manipule manuellement la barrière ou si on force dessus). Pour les remettre en place, placer la barrière en fin de course à l'aide du programme PB1_N1_C3.sb2. Appuyer sur un bouton poussoir jusqu'à une fin de course, et tourner la rondelle jusqu'à entendre le clic du capteur fin de course. Ensuite appuyer sur l'autre bouton jusqu'à la seconde fin de course et refaire le même processus.

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

Objectif : ouverture d'un battant à l'appui sur BP_Exterieur. Fermeture du portail à l'appui sur BP_Interieur

Correction :

Blocs

```
graph TD
    Start[Début - générer le code] --> Loop[ répéter indéfiniment ]
    Loop --> WaitInt[ attendre jusqu'à Bouton-poussoir Intérieur appuyé ]
    WaitInt --> Open[ ouvrir_gauche ]
    Open --> WaitExt[ attendre jusqu'à Bouton-poussoir Extérieur appuyé ]
    WaitExt --> Close[ fermer_gauche ]
    Loop --> Loop
    DefineOpen[ définir ouvrir_gauche ]
    DefineOpen --> SetMotor[ Ouvrir Moteur_gauche ]
    SetMotor --> WaitOpen[ attendre jusqu'à Fin de course Ouverture_gauche activé ]
    WaitOpen --> StopMotor[ Arrêter Moteur_gauche ]
    DefineClose[ définir fermer_gauche ]
    DefineClose --> SetMotor2[ Fermer Moteur_gauche ]
    SetMotor2 --> WaitClose[ attendre jusqu'à Fin de course Fermeture_gauche activé ]
    WaitClose --> StopMotor2[ Arrêter Moteur_gauche ]
```

Fichier MBlock : PB2_N2_A2.sb2

Exercice niveau 2 - A.3 : ouverture/fermeture des deux battants

Objectif : Ouvrir et fermer les deux battants du portail à la suite en utilisant les capteurs de fin de course. Attention à l'ordre, le battant de gauche doit être ouvert en deuxième et fermé en premier.

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

Correction :

Blocs

```
graph TD
    Start[Début - générer le code] --> Loop[répéter indéfiniment]
    Loop --> OuvreD[ouvrir_droite]
    Loop --> OuvreG[ouvrir_gauche]
    Loop --> Att2S1[attendre 2 secondes]
    Loop --> FermeG[fermer_gauche]
    Loop --> FermeD[fermer_droite]
    Loop --> Att2S2[attendre 2 secondes]
    Loop --> Loop
    
    subgraph "définir ouvrir_gauche"
        OuvreGDef[Ouvrir Moteur_gauche]
        AttJusG[attendre jusqu'à Fin de course Ouverture_gauche activé]
        ArrêteG[Arrêter Moteur_gauche]
    end
    
    subgraph "définir fermer_gauche"
        FermeGDef[Fermer Moteur_gauche]
        AttJusF[attendre jusqu'à Fin de course Fermeture_gauche activé]
        ArrêteF[Arrêter Moteur_gauche]
    end
    
    subgraph "définir ouvrir_droite"
        OuvreDDef[Ouvrir Moteur_droit]
        AttJusD[attendre jusqu'à Fin de course Ouverture_droit activé]
        ArrêteD[Arrêter Moteur_droit]
    end
    
    subgraph "définir fermer_droite"
        FermeDDef[Fermer Moteur_droit]
        AttJusF2[attendre jusqu'à Fin de course Fermeture_droit activé]
        ArrêteF2[Arrêter Moteur_droit]
    end
```

Fichier MBlock : PB2_N2_A2.sb2

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

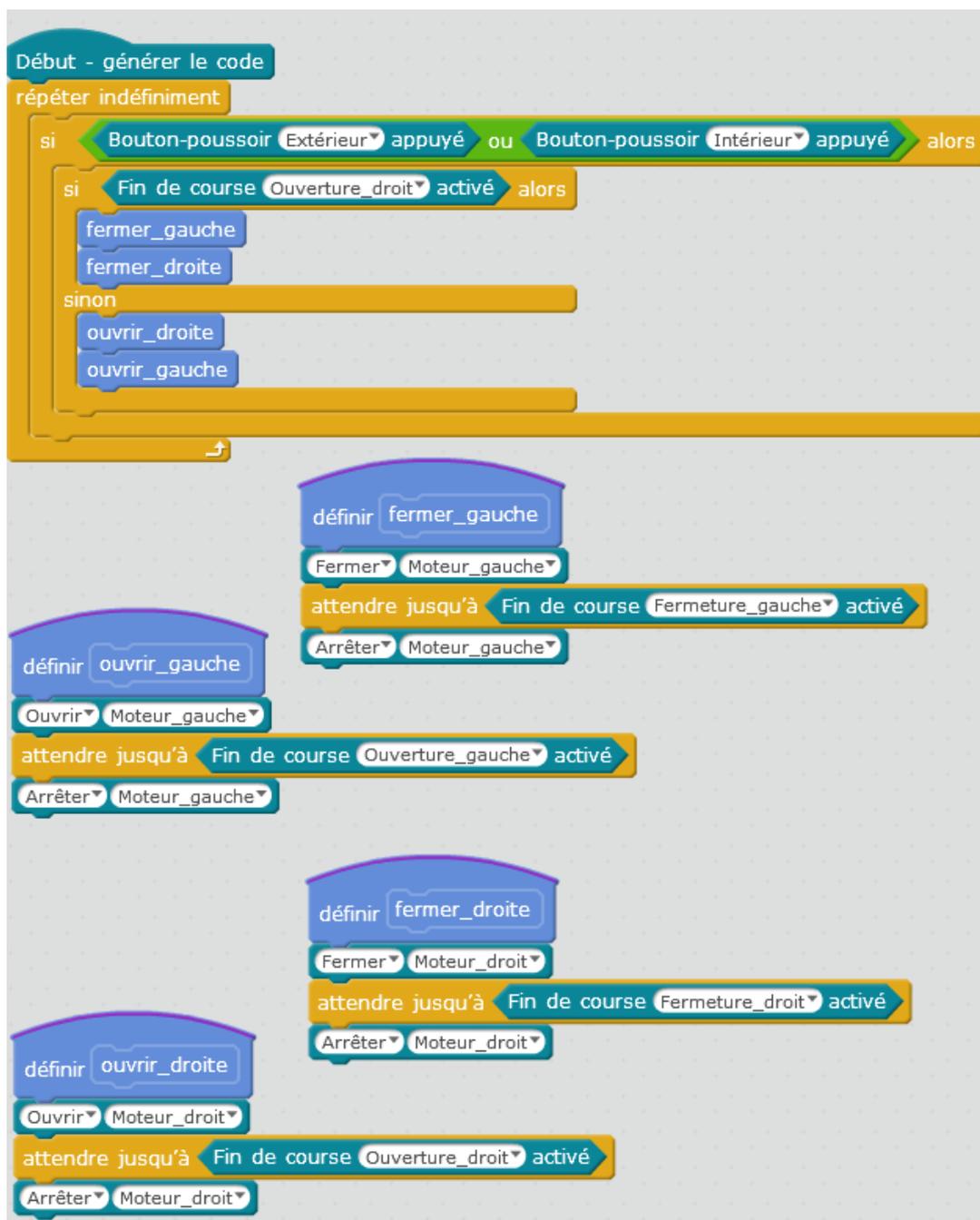
Notions abordées : utilisation d'opérateur logique OU (+)

Instructions utilisées :



Correction :

Blocs



Fichier MBlock : PB2_N2_A4.sb2

Exercice niveau 2 - A.5 : Amélioration de la vitesse

Objectif : Mettre l'ouverture et la fermeture des deux battants dans deux sous-fonctions ouvrir et fermer. Mettre un délai entre l'activation des deux battants, et les arrêter sur leur capteur fin de course respectif. Basculer le voyant lumineux lorsque le deuxième battant se met en marche.

Correction :

Blocs

Fichier MBlock : PB2_N2_A5.sb2

Remarque : En fonction de la vitesse de vos moteurs, le temps d'attente dans les fonctions peut augmenter. Nous vous conseillons de commencer avec un temps de 4 secondes et de régler la vitesse de vos moteurs avant de descendre cette valeur.

Exercice niveau 2 - B.1 : Gestion automatique du portail via capteurs IR

Objectif : Ouvrir le portail lors d'un passage entre les capteurs infrarouge à l'intérieur, le fermer lors d'un passage entre les capteurs infrarouges à l'extérieur après 3 secondes.

Instructions utilisées :



Correction :

Blocs

```

Début - générer le code
Activer Emetteur_IR_Interieur
Activer Emetteur_IR_Exterieur
répéter indéfiniment
  si Entrée Recepteur_IR_Interieur activée alors
    si non Fin de course Ouverture_droit activé alors
      ouvrir
    si Entrée Recepteur_IR_Exterieur activée alors
      si non Fin de course Fermeture_droit activé alors
        fermer
  fin

définir fermer
  Fermer Moteur_gauche
  attendre 3 secondes
  Fermer Moteur_droit
  répéter jusqu'à Fin de course Fermeture_gauche activé
    Activer Voyant_Lumineux
    attendre 0.2 secondes
    Désactiver Voyant_Lumineux
    attendre 0.2 secondes
  Arrêter Moteur_gauche
  attendre jusqu'à Fin de course Fermeture_droit activé
  Arrêter Moteur_droit

définir ouvrir
  Ouvrir Moteur_droit
  attendre 3 secondes
  Ouvrir Moteur_gauche
  répéter jusqu'à Fin de course Ouverture_droit activé
    Activer Voyant_Lumineux
    attendre 0.2 secondes
    Désactiver Voyant_Lumineux
    attendre 0.2 secondes
  Arrêter Moteur_droit
  attendre jusqu'à Fin de course Ouverture_gauche activé
  Arrêter Moteur_gauche

```

Fichier MBlock : PB2_N2_B1.sb2

Programmation niveau 3

Objectif : Utiliser les modules plus complexes : pilotage à distance ...

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 3 B : module Bluetooth		
PB2_N3_B1.sb2	Contrôler l'ouverture et la fermeture du portail à l'aide de 2 boutons présent sur l'application Android.	Fonctionnalité matérielle abordé : - module Bluetooth Notions de programmation abordées : - liaison série (hserin/hserout)
PB2_N3_B2.sb2	Ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.	
PB2_N3_B3.sb2	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.	
PB2_N3_B4.sb2	Gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.	

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.

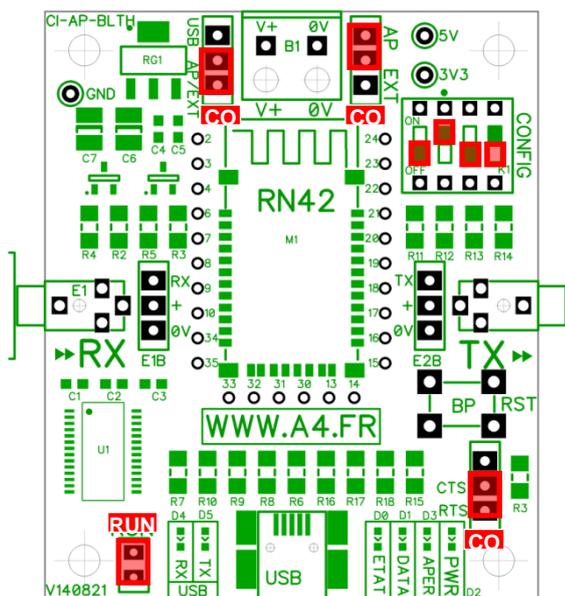
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

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.

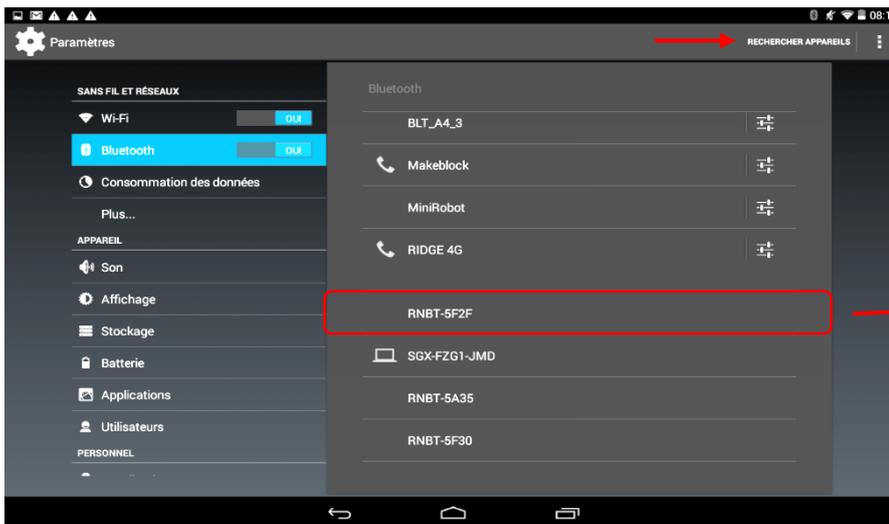
USB RX indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

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 - B.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.

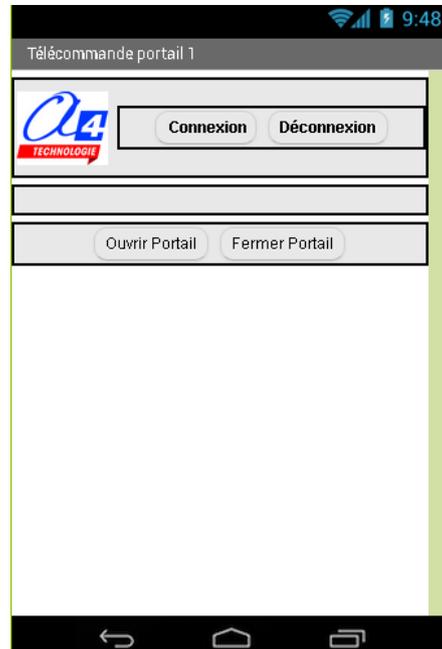
Instructions utilisées :

Donnée Bluetooth reçue

Envoyer donnée Bluetooth 0

Application Android : Portail_1.apk

Fichier App Inventor : Portail_1.aia



Correction :

Blocs

```

Début - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    si non Fin de course Ouverture_droit activé alors
      ouvrir
    sinon
      si Donnée Bluetooth reçue = 2 alors
        si non Fin de course Fermeture_droit activé alors
          fermer
        sinon
          définir fermer
            Fermer Moteur_gauche
            attendre 3 secondes
            Fermer Moteur_droit
            répéter jusqu'à Fin de course Fermeture_gauche activé
              Activer Voyant_Lumineux
              attendre 0.2 secondes
              Désactiver Voyant_Lumineux
              attendre 0.2 secondes
            Arrêter Moteur_gauche
            attendre jusqu'à Fin de course Fermeture_droit activé
            Arrêter Moteur_droit
          définir ouvrir
            Ouvrir Moteur_droit
            attendre 3 secondes
            Ouvrir Moteur_gauche
            répéter jusqu'à Fin de course Ouverture_droit activé
              Activer Voyant_Lumineux
              attendre 0.2 secondes
              Désactiver Voyant_Lumineux
              attendre 0.2 secondes
            Arrêter Moteur_droit
            attendre jusqu'à Fin de course Ouverture_gauche activé
            Arrêter Moteur_gauche
  fin

```

Fichier MBlock : PB2_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



Correction :

Blocs

```

Début - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    si non Fin de course Ouverture_droit activé alors
      ouvrir
    sinon
      fermer
  ↻

définir fermer
  Fermer Moteur_gauche
  attendre 3 secondes
  Fermer Moteur_droit
  répéter jusqu'à Fin de course Fermeture_gauche activé
    Activer Voyant_Lumineux
    attendre 0.2 secondes
    Désactiver Voyant_Lumineux
    attendre 0.2 secondes
  ↻
  Arrêter Moteur_gauche
  attendre jusqu'à Fin de course Fermeture_droit activé
  Arrêter Moteur_droit

définir ouvrir
  Ouvrir Moteur_droit
  attendre 3 secondes
  Ouvrir Moteur_gauche
  répéter jusqu'à Fin de course Ouverture_droit activé
    Activer Voyant_Lumineux
    attendre 0.2 secondes
    Désactiver Voyant_Lumineux
    attendre 0.2 secondes
  ↻
  Arrêter Moteur_droit
  attendre jusqu'à Fin de course Ouverture_gauche activé
  Arrêter Moteur_gauche

```

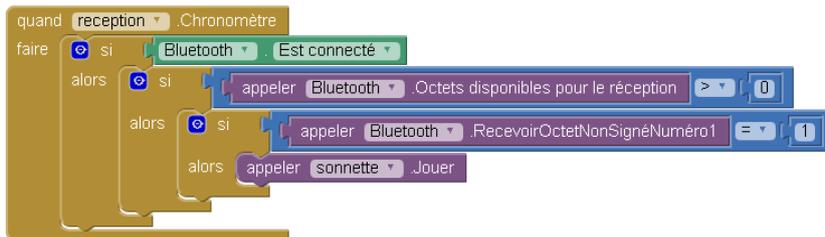
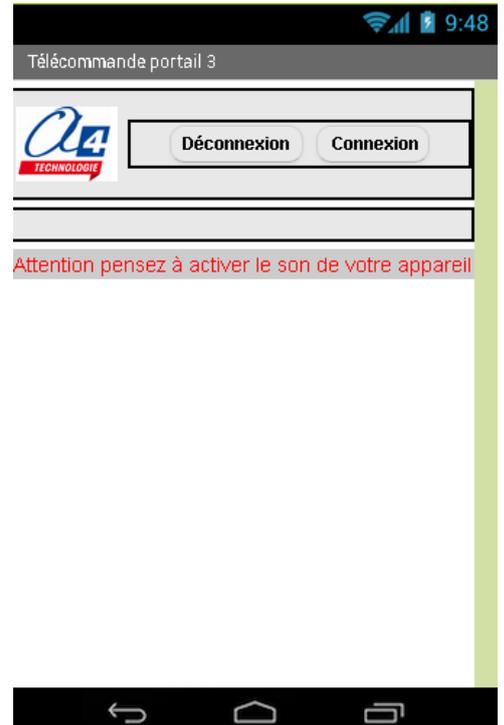
Fichier MBlock : PB2_N3_B2.sb2

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



Correction :

Blocs

Fichier MBlock : PB2_N3_B3.sb2

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. Appuyer sur un bouton poussoir pour demander une ouverture. Refermer le portail après 3 secondes lorsqu'il est ouvert.

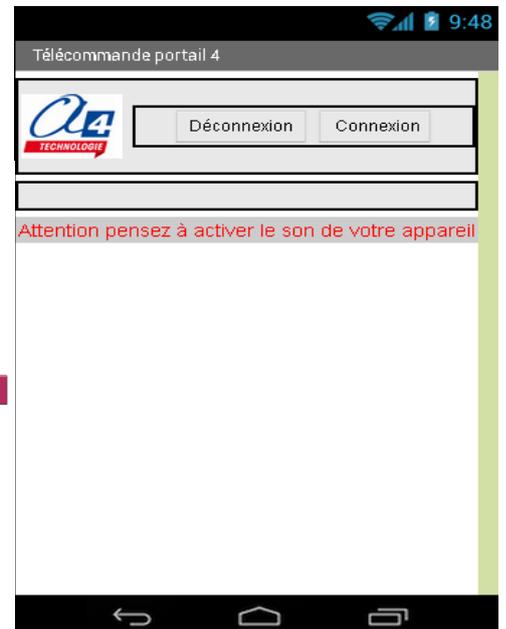
Notion abordée : envoyer et recevoir des informations à l'aide du module Bluetooth à une application.

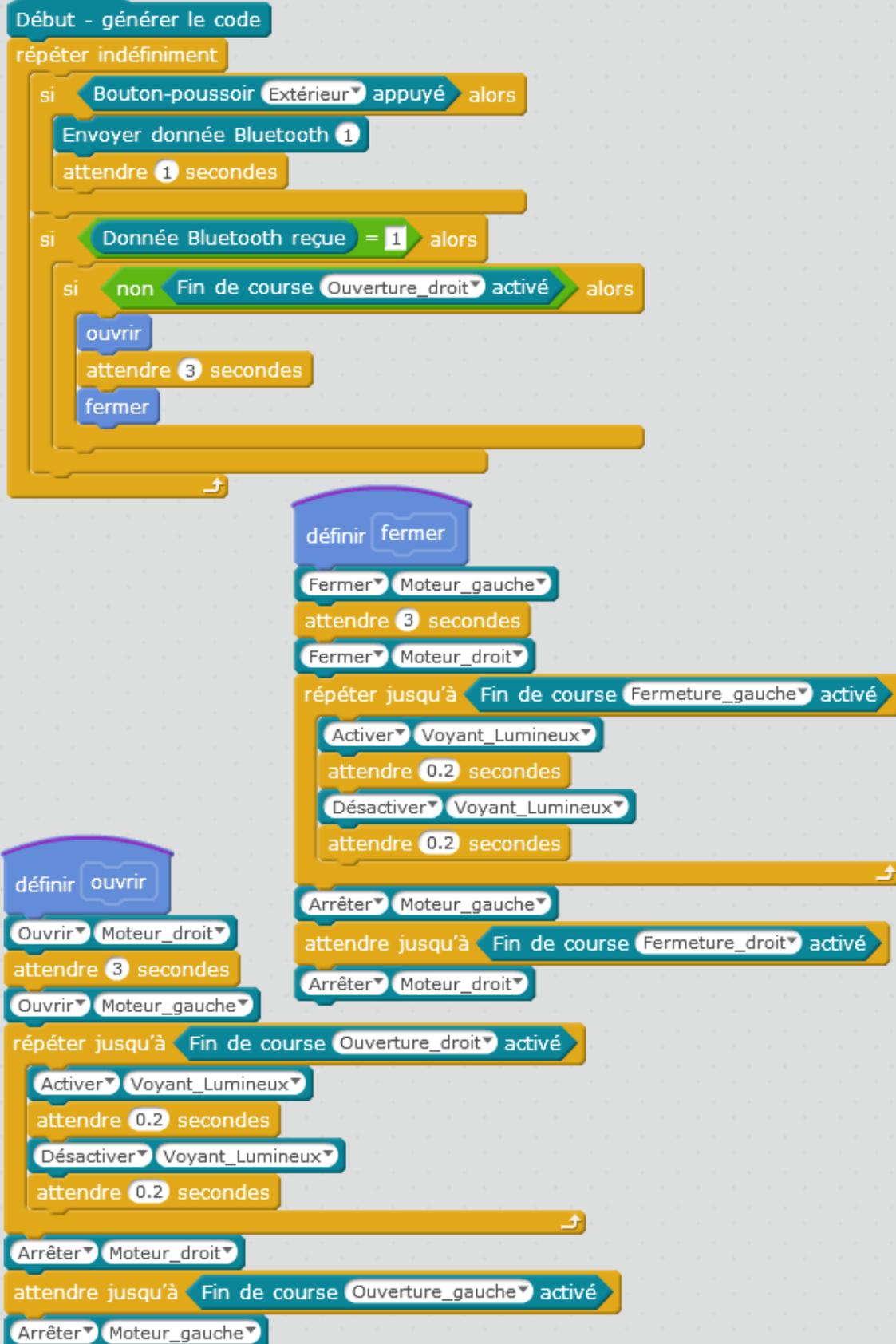
Application Android : Portail_4.apk

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 EnvoyerOctet
        nombre 1
```





Fichier MBlock : PB2_N3_B4.sb2



CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES