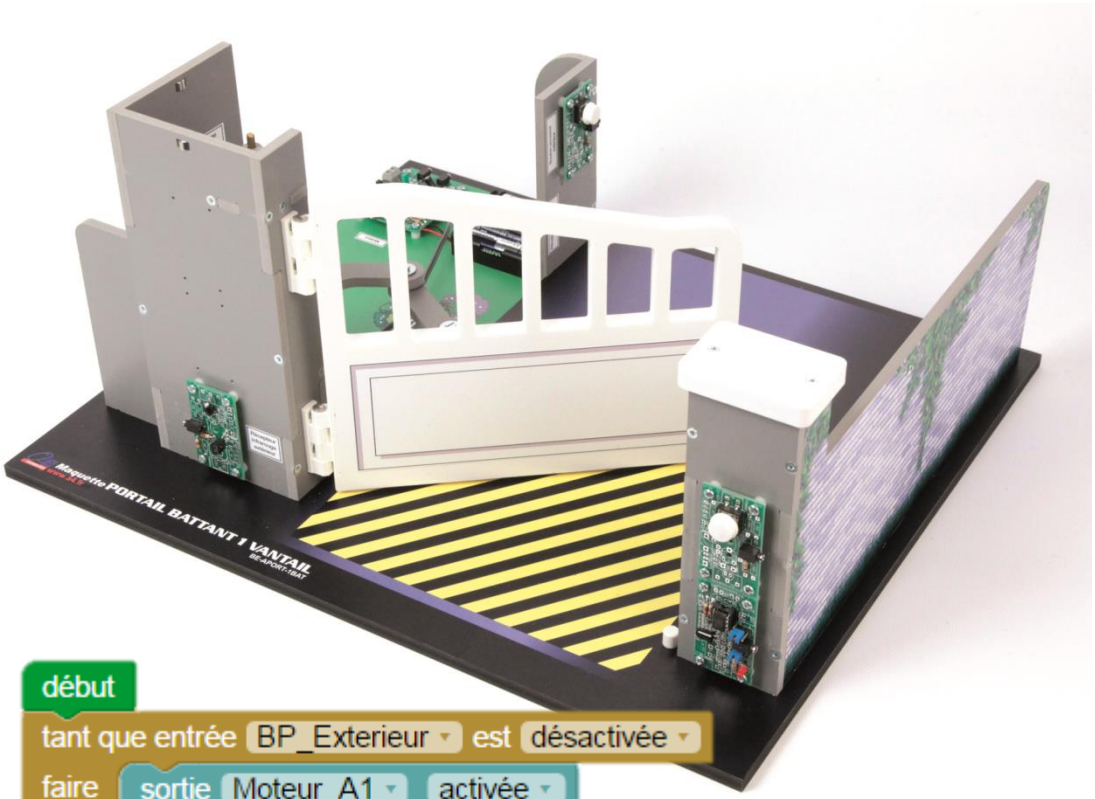
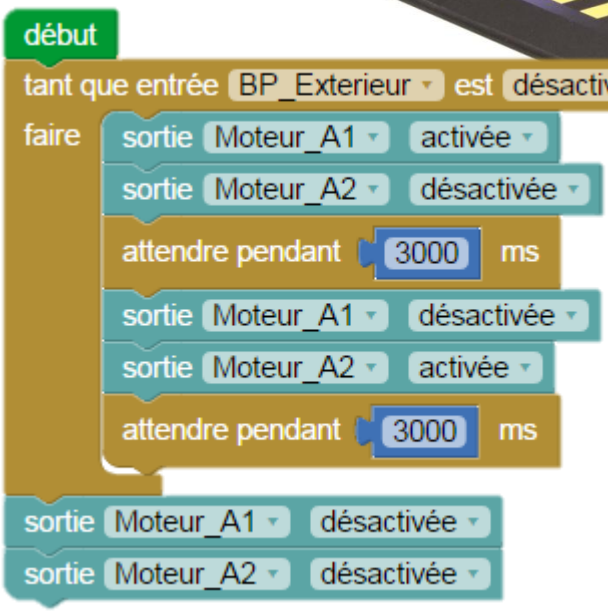
Portail battant un vantail

Maquette programmable avec Editor / Blockly





TECHNOLOGIE

Ressources disponibles pour le projet Portail battant un vantail

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

Portail battant un vantail

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

Logiciels Picaxe Editor 6 / Blockly et App Inventor

- Procédure d'installation du driver pour le câble de programmation.
- Manuel d'utilisation Picaxe Editor 6.
- Notice d'utilisation App Inventor 2.

Activités / Programmation

- Fichiers modèles et fichiers de correction des programmes pour Picaxe EDITOR 6 (organigrammes et blocs) et AppInventor.

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

Logiciels, programmes, manuels utilisateurs téléchargeables gratuitement sur www.a4.fr

SOMMAIRE

Introduction

Portail battant un vantail

La maquette portail battant 1 vantail (BE-APORT-1BAT) est une reproduction homothétique d'un portail battant automatisé réel : Boutons d'activation, capteurs fin de course, barrière optique, clignotant de sécurité, 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, module buzzer et modules infrarouge complémentaires

Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette AutoLumi ont été réalisés sous **PICAXE Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs **avantages** :

- Gratuit
- Blocs et organigrammes (proche algorigrammes).
- Personnalisation des noms des entrées/sorties.
- Personnalisation du jeu d'instructions.
- Mode de simulation visuelle à l'écran pour mettre au point et débugger les programmes.

Vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

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 Blockly.

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 AutoLumi 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 : télécommande infrarouge, module de détection de mouvement PIR, 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 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 Picaxe Editor 6 en blocs (extension .xml) et en organigrammes pour le niveau 1 uniquement (extension .plf).

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 organigrammes ou 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 :

- **PB1** pour Portail Battant 1 vantail
- **N** : niveau de programmation 1-2-3
- **A-B-C** : jeu d'instructions du plus simple au plus avancé

Exemple : PB1_N1_C3.xml
Correspond au niveau 1 avec le jeu d'instructions C, adapté aux objectifs « avancés » de ce niveau.

Prérequis

Pour la version de base :

- Installer le logiciel **Picaxe Editor 6** ou **Blockly for Picaxe :** http://www.picaxe.com/Software
- **Maquette** Portail battant un vantail (Réf. BE-APORT-1BAT).
- **Câble de programmation** Picaxe USB (Réf : CABLE-USBPICAXE).
- **Interface programmable** AutoProgX1 ou X2 (Réf. K-APV2).
- 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.

Caractéristiques techniques

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

Environnement de programmation graphique

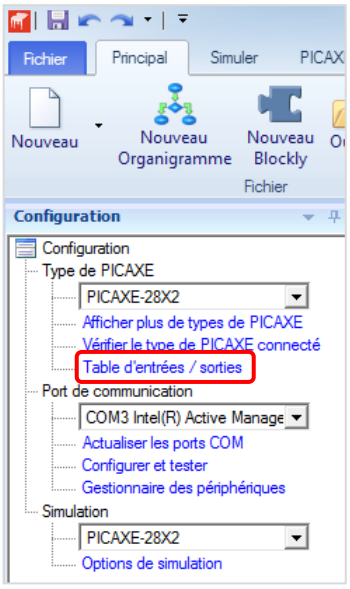
Tous les programmes correspondant aux activités menées autour du Portail battant un vantail ont été réalisés sous **PICAXE Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs avantages :

- Gratuit
- Blocs et organigrammes (proche algorigrammes).
- Personnalisation des noms des entrées/sorties.
- Personnalisation du jeu d'instructions.
- Mode de simulation visuelle à l'écran pour mettre au point et débugger les programmes.

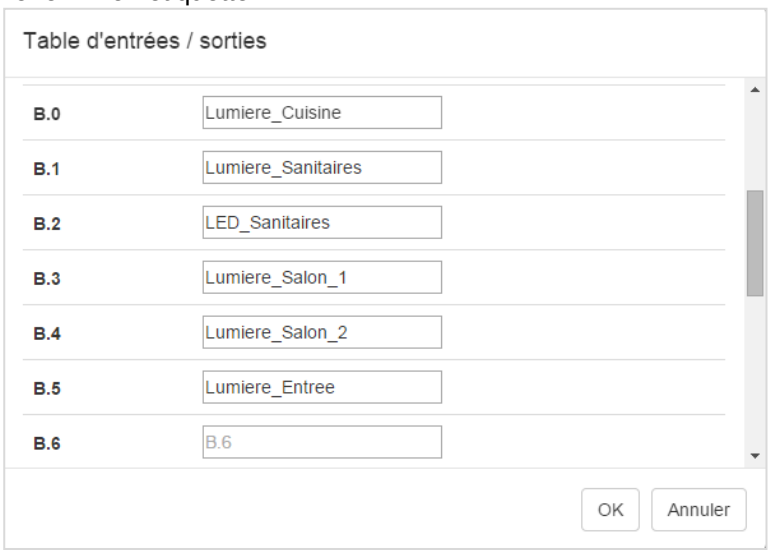
Note : vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

Personnalisation des entrées/ sorties

Nous vous proposons le fichier **PB1_BASE.xml** dans lequel les noms des entrées/sorties ont été personnalisés pour une utilisation avec le Portail battant un vantail. Tous les programmes et activités proposés dans ce document se basent sur cette liste. Celle-ci reste modifiable à tout moment.
A partir de Picaxe Editor 6, dans l'explorateur d'espace de travail cliquer sur **Table d'entrées / sorties**.



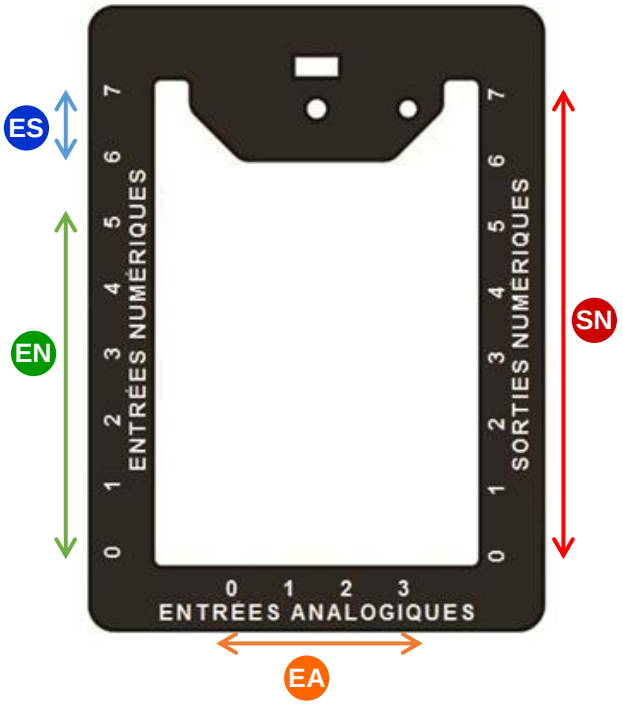
Une fenêtre apparaît à partir de laquelle vous pouvez modifier les noms de toutes les entrées et sorties dans la zone « Mon étiquette ».



Valider en cliquant sur **OK**.

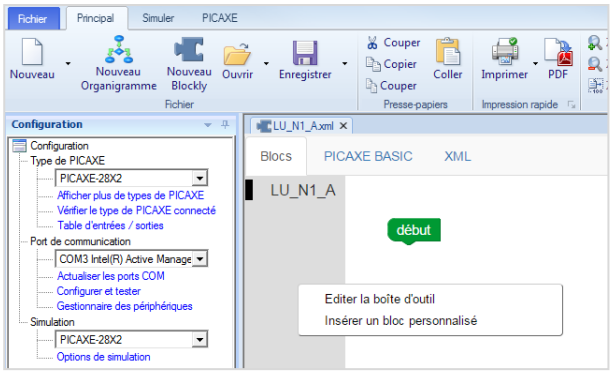
Tableau d'affectation des entrées et sorties

	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
ES 7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	**MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES**		
5	Récepteur barrière infrarouge intérieur (option)	C.5	Recepteur_IR_Int*
4	Récepteur barrière infrarouge extérieur	C.4	Recepteur_IR_Ext
3	Bouton poussoir extérieur	C.3	BP_Exterieur
2	Capteur de fin de course fermeture du portail	C.2	FDC_Fermeture
1	Capteur de fin de course ouverture du portail	C.1	FDC_Ouverture
0	Bouton poussoir intérieur	C.0	BP_Interieur
EA	**MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES**		
	(libre)	A.3	
2	*(libre)*	A.2	
1	*(libre)*	A.1	
0	*(libre)*	A.0	
SN	**MODULES ACTIONNEURS SORTIES NUMÉRIQUES**		
	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	Module sonore Buzzer (option)	B.3	Buzzer*
2	Emetteur barrière infrarouge intérieur (option)	B.2	Emetteur_IR_int*
1	Emetteur barrière infrarouge extérieur	B.1	Emetteur_IR_Ext
0	Module signal LED jaune	B.0	Voyant_Lumineux

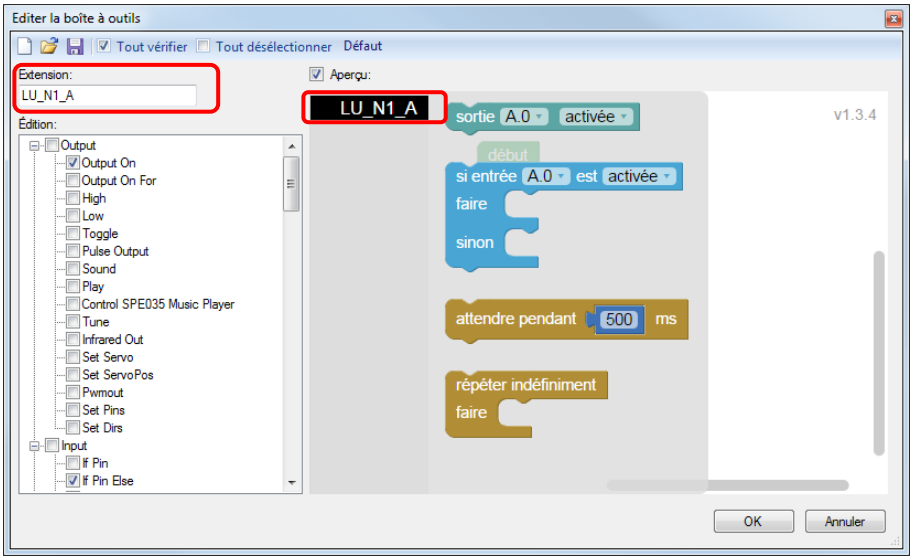


Personnalisation du jeu d'instructions

Vous pouvez personnaliser l'affichage du jeu d'instructions pour en limiter la quantité afin de faciliter la l'analyse et la correction des erreurs. Faire un clic droit sur la zone des blocs puis cliquer sur **Editer la boite d'outil**.



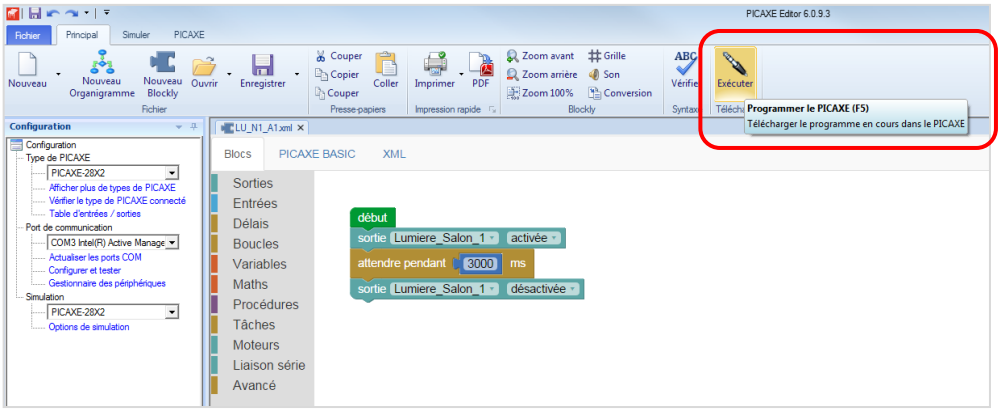
Une fenêtre s'ouvre à partir de laquelle vous pouvez sélectionner ou désélectionner les instructions de votre choix. Vous pouvez renommer le jeu d'instructions dans la zone « **Extension** ».



Valider en cliquant sur **OK**.

Procédure de chargement d'un programme

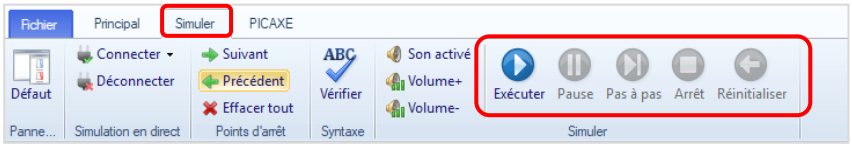
Commencer par relier le Loupiot à l'ordinateur avec le câble de programmation USB et le mettre sous tension.
A partir du Picaxe Editor 6, ouvrir un programme.



A partir du menu **Principal** ou du menu **PICAXE**, cliquer sur le bouton **Exécuter**. Vous pouvez également utiliser la touche **F5** de votre clavier.
Note : un programme téléchargé écrase le précédent.

Mode simulation

La simulation sur Picaxe EDITOR 6 permet de tester un programme avant de le téléverser dans la maquette.
Pour lancer et contrôler une simulation, utiliser les boutons **Exécuter / Pause / Pas à pas / Arrêt** à partir du menu **Simuler**.



La simulation surligne les blocs dans l'espace de travail pour vous montrer où en est le programme.

Programmation version de base niveau 1

Objectifs :

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

Ce niveau permet de découvrir toutes les fonctionnalités de base du portail Battant, 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 « PB1_N1_A1.xml », charger le programme modèle « PB1_BASE.xml ».

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 complétement le portail Battant.

<table>
<tr><th>Nom du fichier</th><th>Description</th><th>Objectif</th></tr>
<tr><td colspan="3">Niveau 1 A (Modèle : PB1_N1_A.xml)</td></tr>
<tr><td>PB1_N1_A1.xml</td><td>Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.</td><td rowspan="3">Fonctionnalité matérielle abordée :
-Allumage/extinction du voyant lumineux
Notions de programmation abordées :
-séquence d'instructions
-temps d'attente
-boucle infinie</td></tr>
<tr><td>PB1_N1_A2.xml</td><td>Répéter cette même action deux fois</td></tr>
<tr><td>PB1_N1_A3.xml</td><td>Répéter cette action à l'infini</td></tr>
<tr><td colspan="3">Niveau 1 B (Modèle : PB1_N1_B.xml)</td></tr>
<tr><td>PB1_N1_B1.xml</td><td>Activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.</td><td rowspan="2">Fonctionnalité matérielle abordée :
-Gestion du moteur
-Utilisation de Bouton-poussoir
Notions de programmation abordées :
-boucle qui dépend d'une entrée</td></tr>
<tr><td>PB1_N1_B2.xml</td><td>Ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.</td></tr>
<tr><td colspan="3">Niveau 1 C (Modèle : PB1_N1_C.xml)</td></tr>
<tr><td>PB1_N1_C1.xml</td><td>allumer le voyant lumineux à l'appui du BP.</td><td rowspan="3">Fonctionnalité matérielle abordée :
-Gestion des modules infrarouge
-Utilisation de Bouton-poussoir

Notions de programmation abordées :
-Le test d'une entrée (si/sinon)</td></tr>
<tr><td>PB1_N1_C2.xml</td><td>activer le voyant lumineux lorsque la barrière infrarouge est franchie</td></tr>
<tr><td>PB1_N1_C3.xml</td><td>contrôler l'allumage du voyant et du moteur avec des BP</td></tr>
<tr><td colspan="3">Niveau 1 D (Modèle : PB1_N1_D.xml)</td></tr>
<tr><td>PB1_N1_D1.xml</td><td>Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage)</td><td rowspan="4">Fonctionnalité matérielle abordé :

Notions de programmation abordées :
-Définition de variable
-Incrémentation de variable
-Test (si/sinon) de variable
-Test (juste si) d'entrée
-Débogage</td></tr>
<tr><td>PB1_N1_D2.xml</td><td>Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.</td></tr>
<tr><td>PB1_N1_D3.xml</td><td>Incrémenter une variable à l'appui d'un bouton poussoir, la décrémenter à l'appui de l'autre bouton poussoir.</td></tr>
<tr><td>PB1_N1_D4.xml</td><td>incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.</td></tr>
</table>

Niveau 1 - A

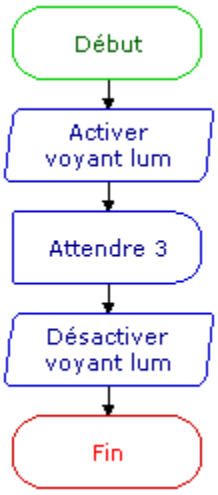
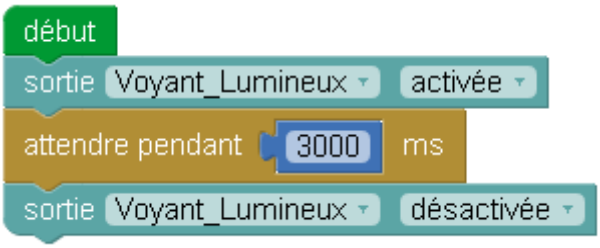
Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux

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

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

Instructions utilisées :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_A1_Organigramme.plf	Fichier Blockly : PB1_N1_A1.xml

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

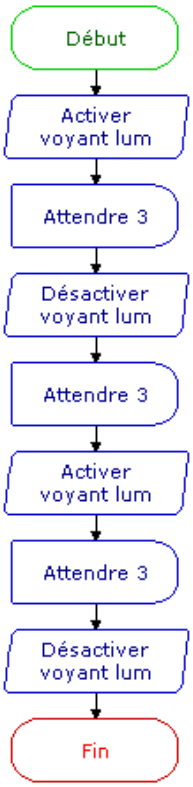
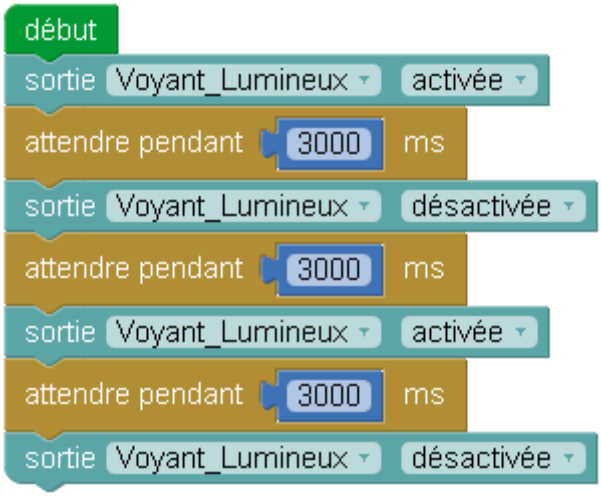
Exercice niveau 1 - A.2: Répéter une action deux fois

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.

Instructions utilisées :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_A2_Organigramme.plf	Fichier Blockly : PB1_N1_A2.xml

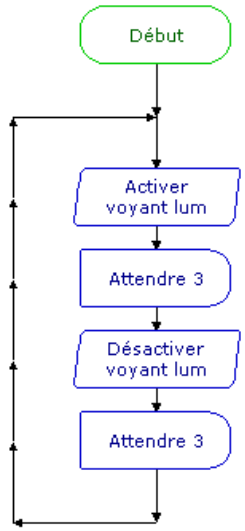
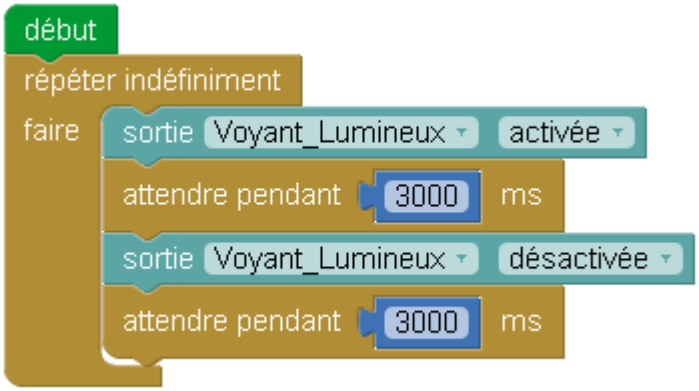
Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment

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

Notion abordée : la boucle infinie.

Instructions utilisées :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_A3_Organigramme.plf	Fichier Blockly : PB1_N1_A3.xml

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 AutoProgX2.

Niveau 1 - B

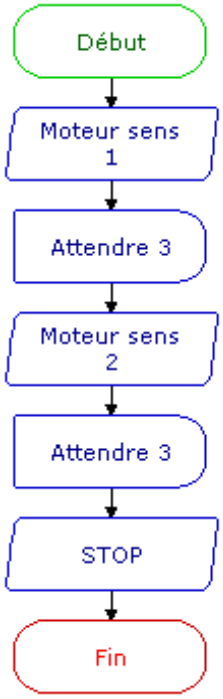
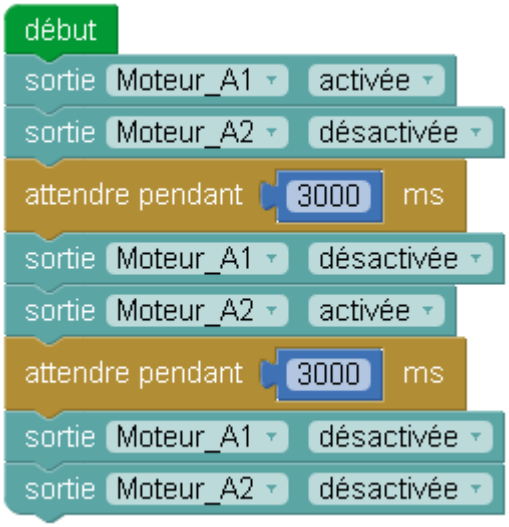
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur.

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

Notion abordée : utilisation d'un moteur.

Instructions utilisées :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_B1_Organigramme.plf	Fichier Blockly : PB1_N1_B1.xml

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).

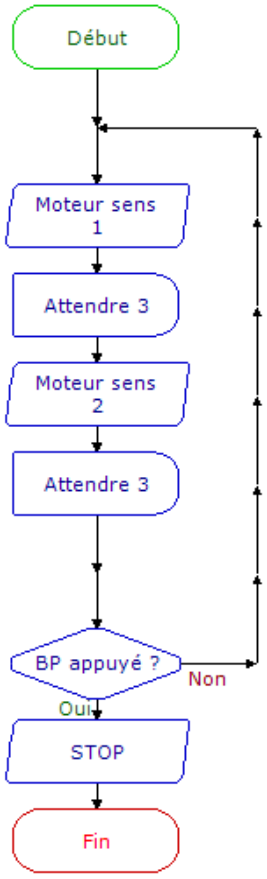
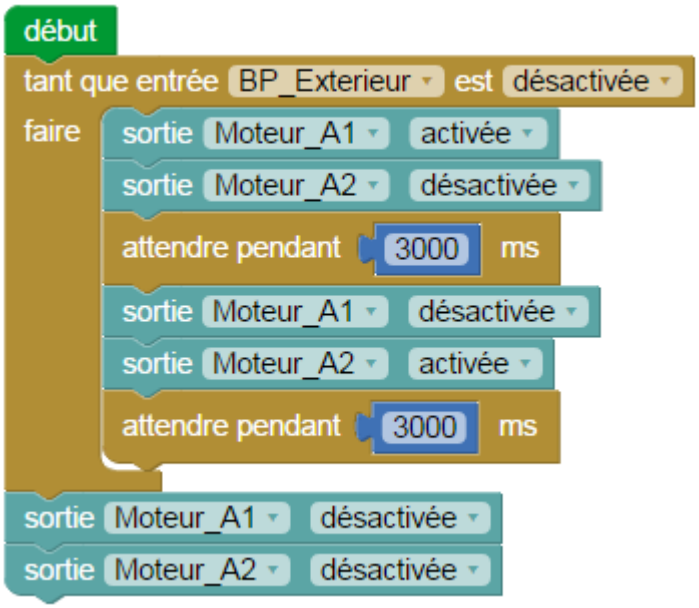
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que

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 :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_B2_Organigramme.plf	Fichier Blockly : PB1_N1_B2.xml

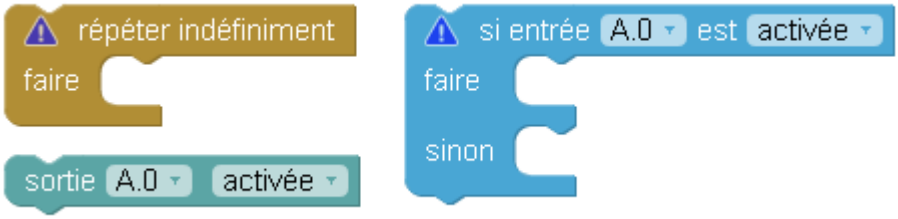
Remarque : Le programme ne peut sortir de la boucle qu'une fois le test sur le bouton-poussoir validé. Le test sur le bouton poussoir se fait qu'une seule fois en début de séquence, avant de commencer l'ouverture. Si un appui est effectué pendant la séquence, aucun effet n'aura lieu sur le programme. 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 - C.1 : Instruction conditionnelle et bouton-poussoir

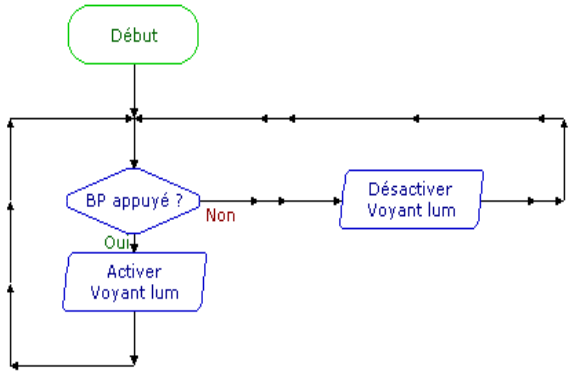
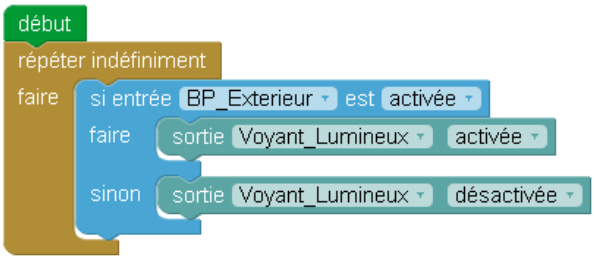
Objectif : allumer le voyant lumineux à l'appui du BP.

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

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_C1_Organigramme.plf	Fichier Blockly : PB1_N1_C1.xml

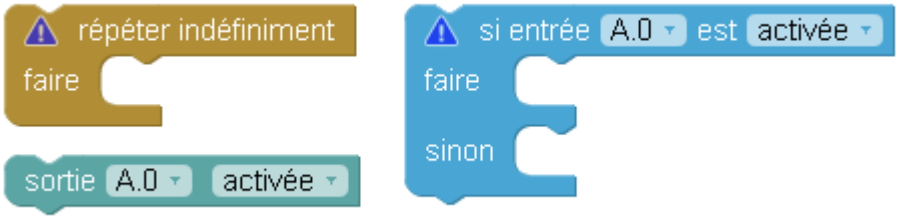
Remarque : les blocs de couleur bleu claires représente des commandes concernant l'utilisation des entrées.

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

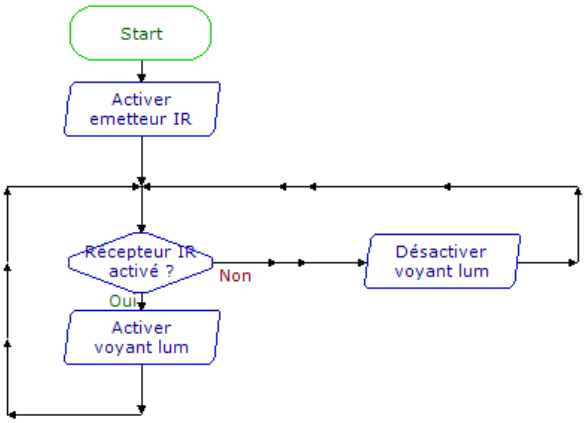
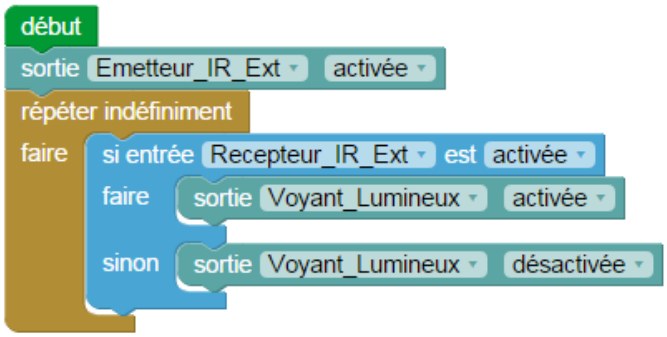
Objectif : activer le voyant lumineux lorsque la barrière infrarouge est franchie.

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

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_C2_Organigramme.plf	Fichier Blockly : PB1_N1_C2.xml

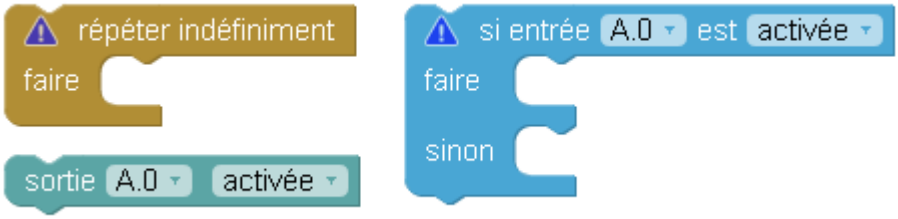
Remarque : l'entrée du récepteur IR est activée d'origine et se désactive lors de la réception du signal de l'émetteur IR.
Lorsque un obstacle franchi la barrière IR, le signal n'est plus transmis et l'entrée du récepteur IR devient active.
Si le programme ne marche pas, mettre le cavalier MODE de l'émetteur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

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

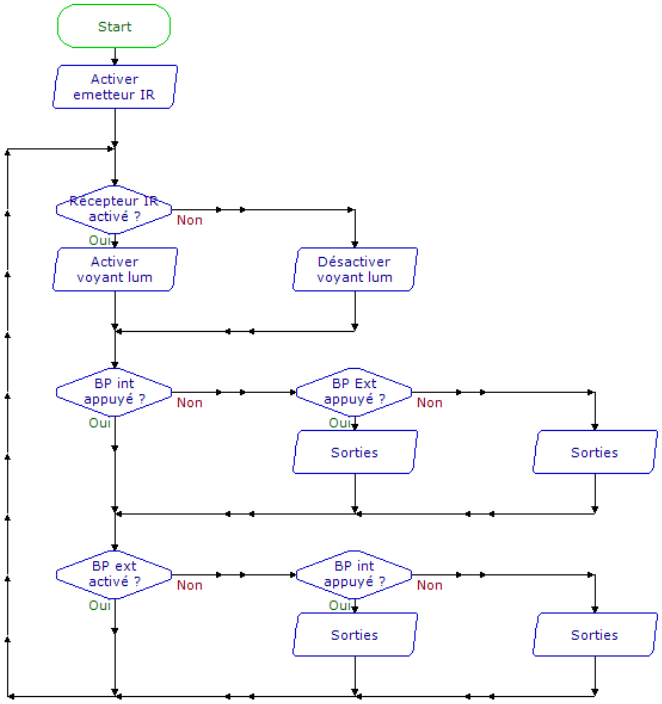
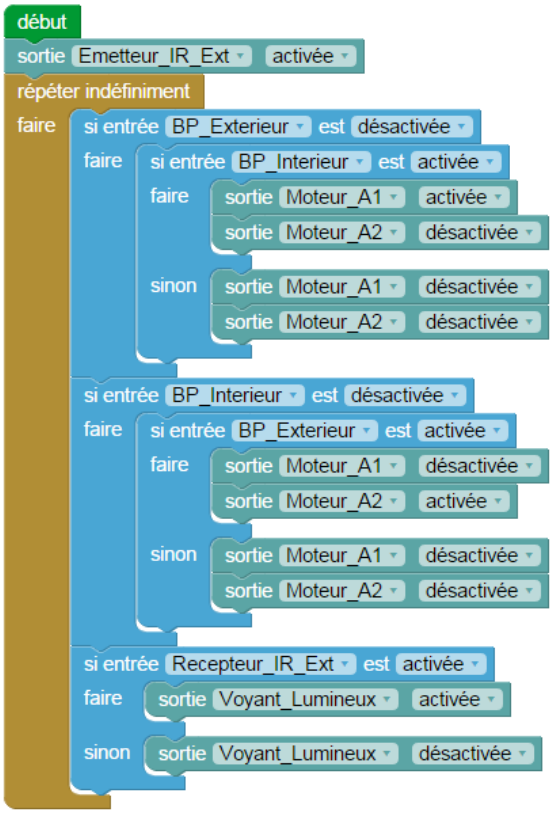
Objectif : contrôler le moteur avec les boutons poussoirs et allumer le voyant sur le franchissement de la barrière infrarouge.

Notion abordée : utilisation des commandes conditionnelles.

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_C3_Organigramme.plf	Fichier Blockly : PB1_N1_C3.xml

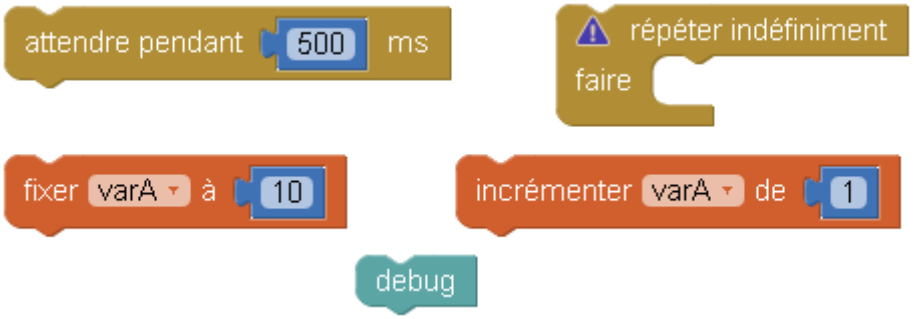
Niveau 1 - D

Exercice niveau 1 - D.1 : Utilisation des variables

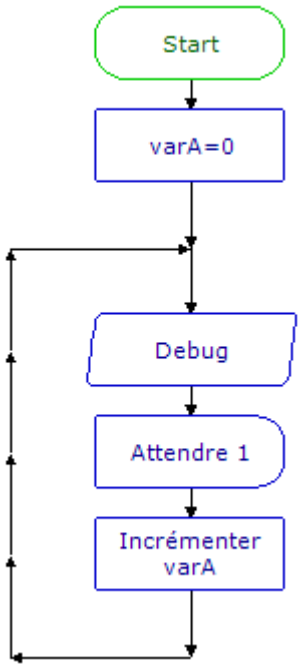
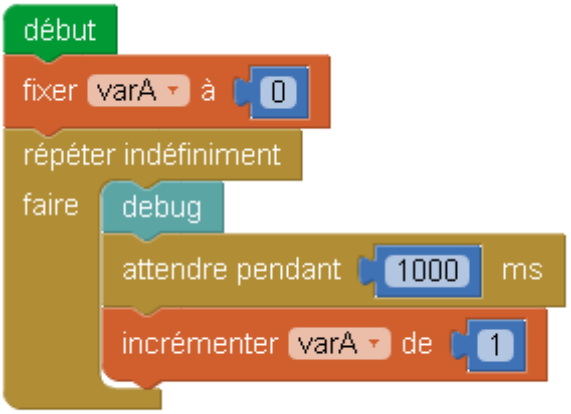
Objectif : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).

Notions abordées : la variable : définition et incrémentation, debug.

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_D1_Organigramme.plf	Fichier Blockly : PB1_N1_D1.xml

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.

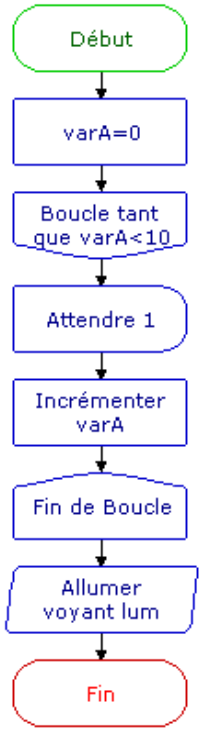
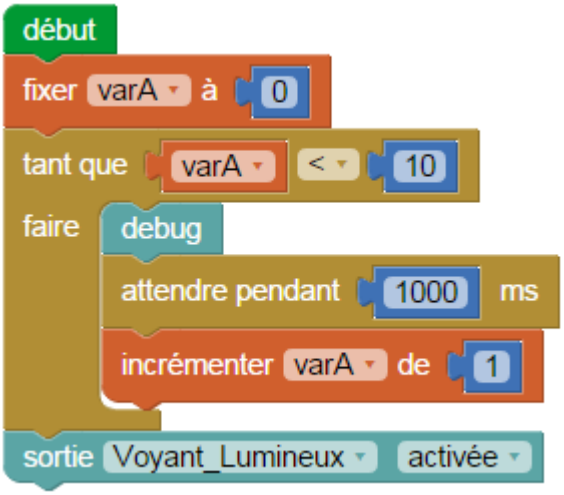
Exercice niveau 1 - D.2 : Utiliser et tester une variable

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 :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_D2_Organigramme.plf	Fichier Blockly : PB1_N1_D2.xml

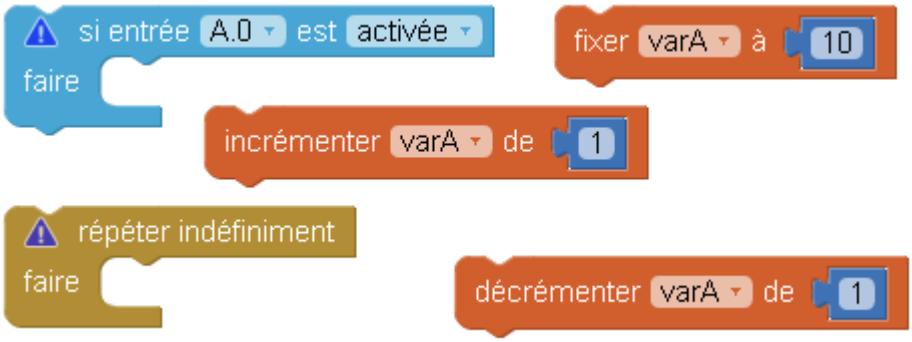
Remarque : cet exercice peut être utilisé comme un minuteur.

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

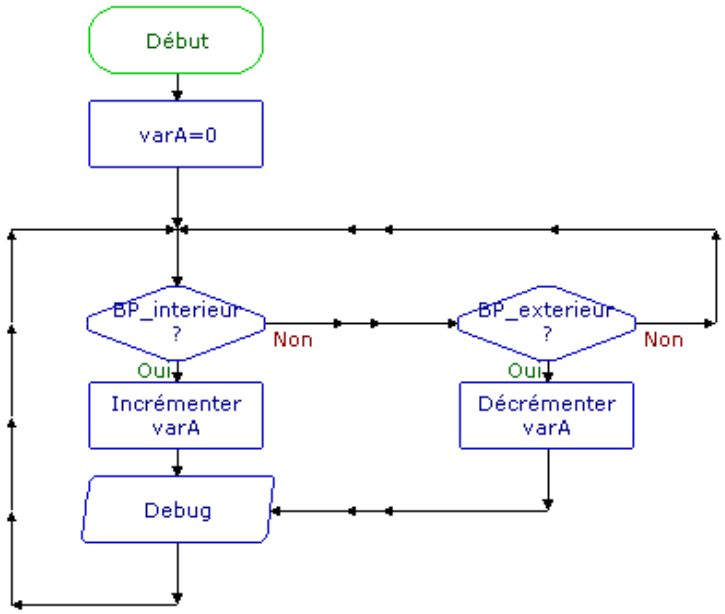
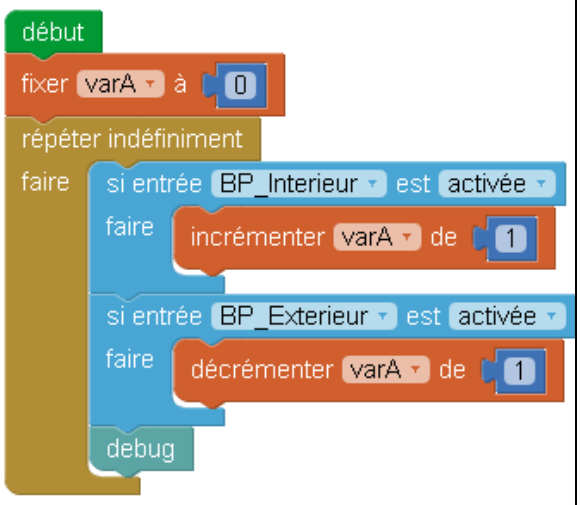
Objectif : incrémenter une variable à l'appui d'un bouton poussoir, décrémenter la même variable à l'appui de l'autre bouton poussoir.

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

Instruction utilisée :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_D3_Organigramme.plf	Fichier Blockly : PB1_N1_D3.xml

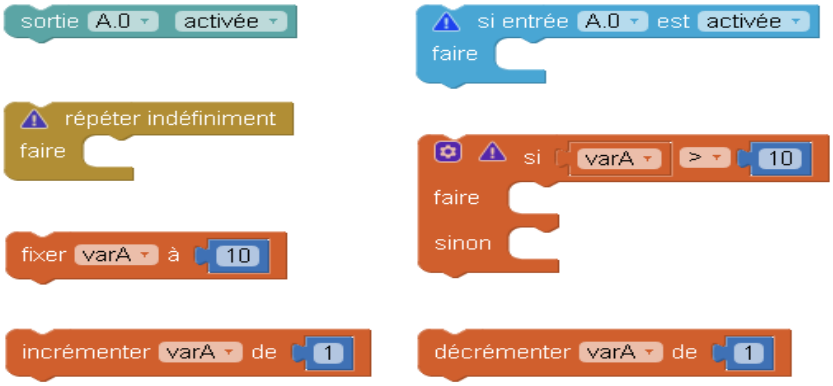
Remarques : deux tests sont insérés l'un après l'autre. La vitesse d'exécution du programme donne l'impression que les commandes sont exécutées en même temps.
La commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

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

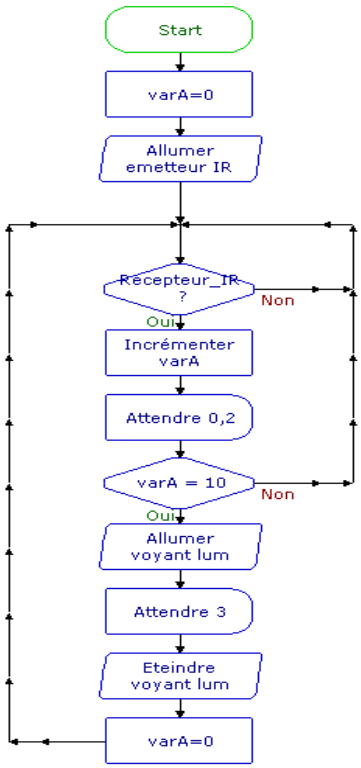
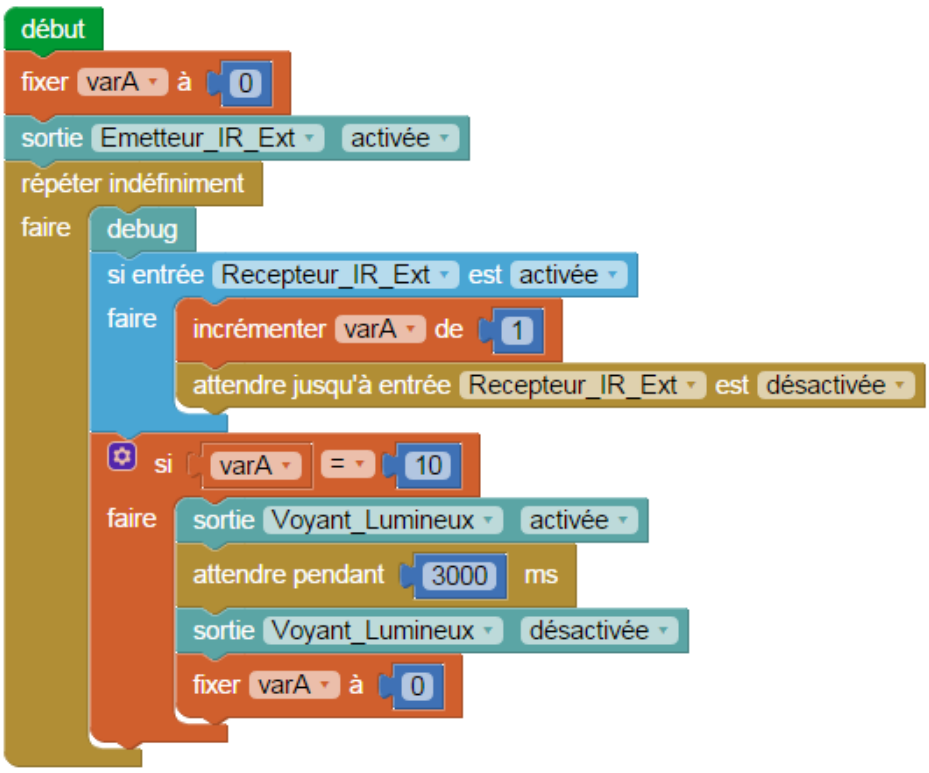
Objectif : incrémenter une variable à chaque passage sur la barrière IR. Lorsque le compteur arrive à 10, activer le voyant lumineux 3 secondes et remettre la variable à zéro

Notion abordée : test dépendant d'une variable

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PB1_N1_D4_Organigramme.plf	Fichier Blockly : PB1_N1_D4.xml

Programmation version de base niveau 2

Objectifs :

- Utilisation concrète du portail battant
- 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 battant, au fur et à mesures 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 (Modèle : PB1_N1_A.xml)		
PB1_N2_A1.xml	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.	Fonctionnalité matérielle abordé : Utilisation des FDC
PB1_N2_A2.xml	Ouverture du portail à l'appui sur BP_Exterieur. Fermeture du portail à l'appui sur BP_Interieur	
PB1_N2_A3.xml	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.	
PB1_N2_A4.xml	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.	

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

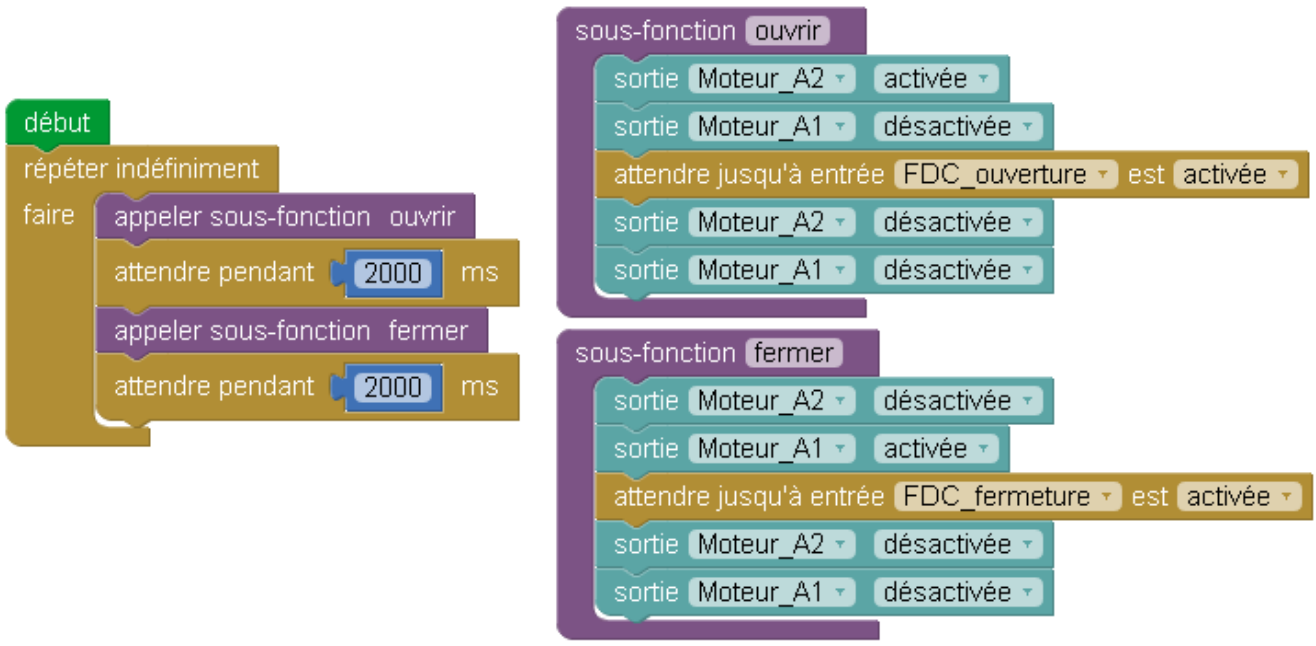
Objectif : ouvrir et fermer le portail avec 2 secondes d'attente entre chaque mouvement. Utiliser les capteurs fins de course pour contrôler l'ouverture et la fermeture.

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

Réf. K-AP-MMR

Correction :

Blocs



Fichier Blockly : PB1_N2_A1.xml

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.xml 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.

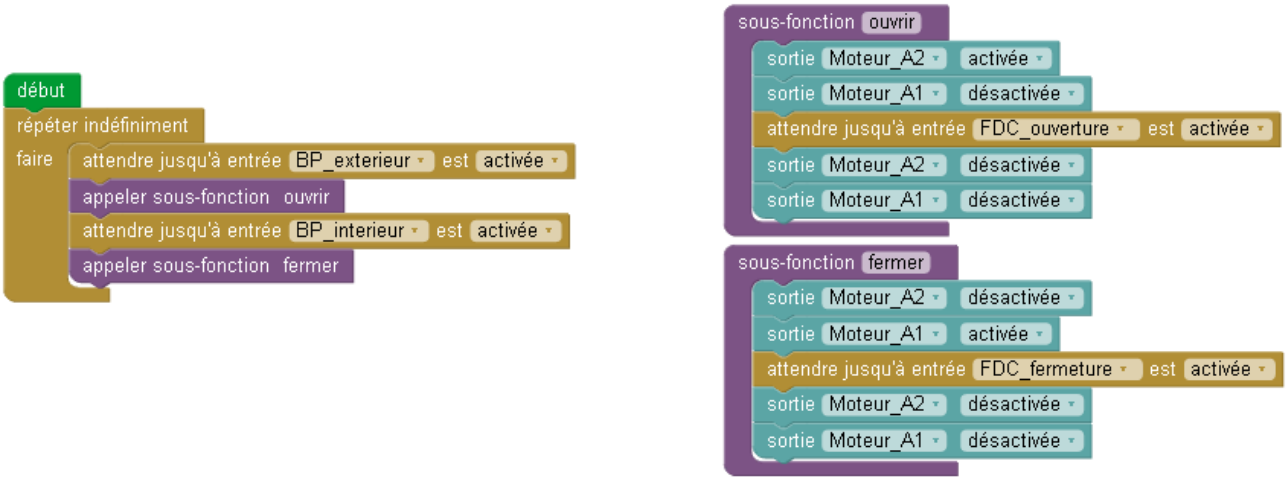
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture

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

Notions abordées :

Correction :

Blocs



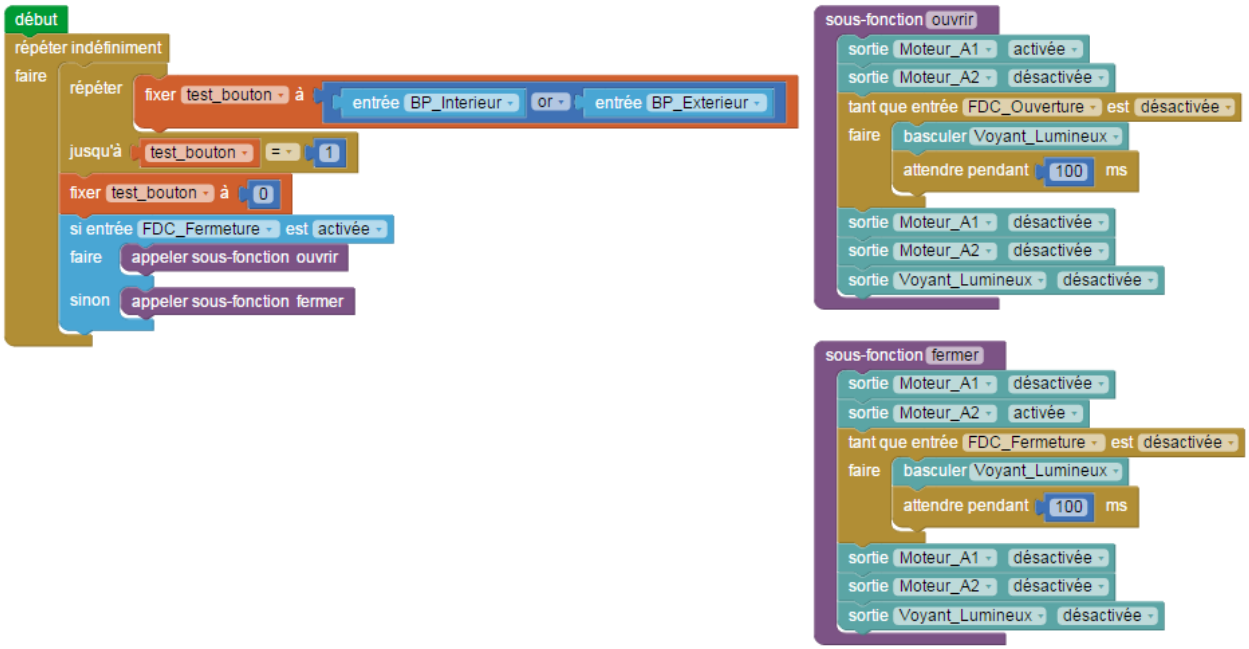
Fichier Blockly : PB1_N2_A2.xml

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

Fichier Blockly : PB1_N2_A3.xml

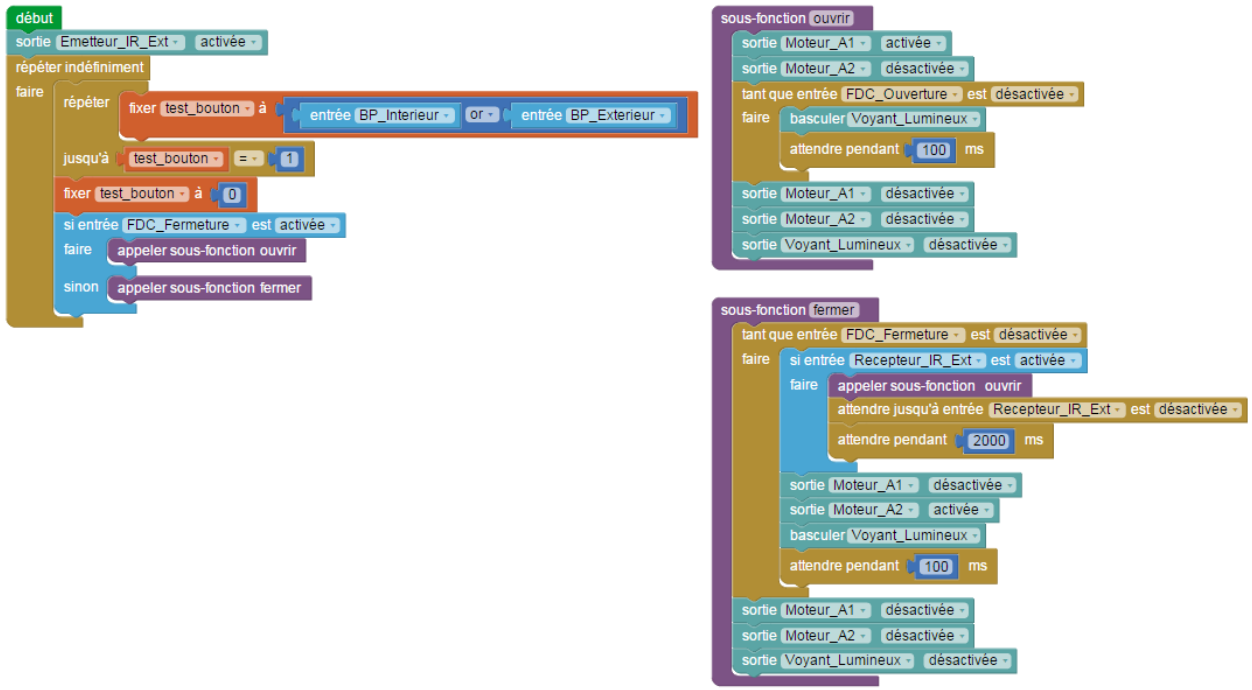
Exercice niveau 2 - A.4 : Contrôle d'ouverture/fermeture avec BP, signal de sécurité

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 via les capteurs infrarouge lors la fermeture du portail.

Notions abordées : utilisation d'une procédure de sécurité

Correction :

Blocs



Fichier Blockly : PB1_N2_A4.xml

Programmation version de base niveau 3 (OPTION)

Objectif :

- Utiliser les modules plus complexes : pilotage à distance, contrôle par le courant…

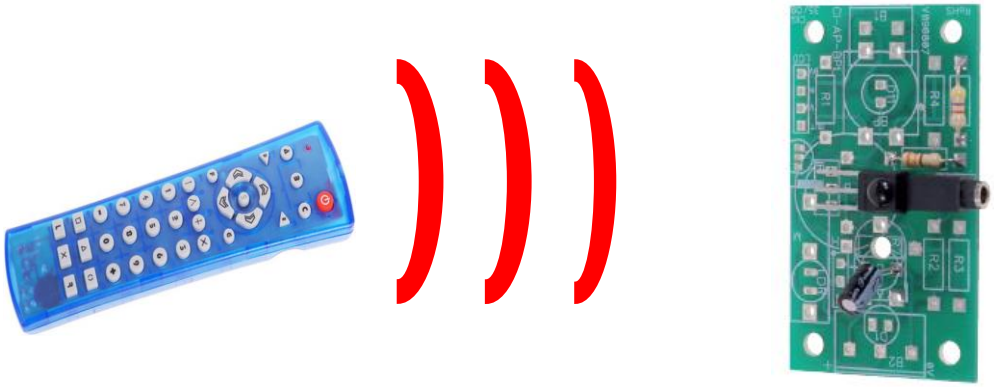
Le niveau 3 n'intègre pas de nouvelles notions de programmation mais de nouveaux blocs permettant d'utiliser les modules options.

Nom du fichier	Description	Objectif
Niveau 3 A : Télécommande IR		
PB1_N3_A1.xml	Ouvrir et fermer le portail à l'aide de la télécommande IR.	Fonctionnalité matérielle abordé : Utilisation de la télécommande IR Notions de programmation abordées : Utilisation d'un block dédié à la communication IR
PB1_N3_A2.xml	Ouvrir le portail dès la réception d'un code spécifique envoyée par la télécommande. Le fermer 2 secondes plus tard.	
PB1_N3_A3.xml	Reprendre le programme **PB1_N2_A4** et intégrer l'option ouverture par la télécommande.	
Niveau 3 B : module Bluetooth		
PB1_N3_B1.xml	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)
PB1_N3_B2.xml	Ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.	
PB1_N3_B3.xml	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.	
PB1_N3_B4.xml	Gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.	
Niveau 3 C : Modules infrarouge complémentaires		
PB1_N3_C1.xml	Allumer le voyant si une des deux barrières infrarouges est activée	Notions de programmation abordées : - Gestion de plusieurs entrées
PB1_N3_C2.xml	Ouvrir le portail lors d'un passage sur la barrière intérieure et le fermer après un passage sur la barrière extérieure	
Niveau 3 D : Module Buzzer		
PB1_N3_D1.xml	Jouer un son sur le buzzer	Fonctionnalité matérielle abordé : - Buzzer
PB1_N3_D2.xml	Jouer un signal sonore sur un mouvement de portail	

Option : Module télécommande infrarouge

Télécommande RAX-TVR10

La télécommande universelle infrarouge associée à un capteur infrarouge approprié permet de piloter à distance une carte PICAXE.
Afin d'assurer la compatibilité de fonctionnement avec le système PICAXE il est nécessaire de l'initialiser avec le mode de fonctionnement au standard *« Sony TV ».*



Attention : L'émetteur infrarouge doit toujours être désactivé lors de l'utilisation de la télécommande infrarouge.

Procédure de mise en service pour PICAXE

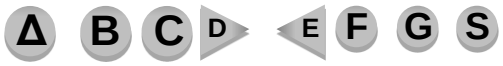
1. Insérer 2 piles AAA dans le logement au dos de la télécommande.

2. Appuyer simultanément sur S et B.
= La LED s'allume.

3. Taper le code 0 - 1 - 3
= La LED clignote brièvement à chaque appui des touches « 0 » et « 1 » puis s'éteint après l'appui sur la touche « 3 ».

4. Appuyer sur le bouton de mise en service. 
= La télécommande est opérationnelle.

Les touches suivantes risquent de déprogrammer :

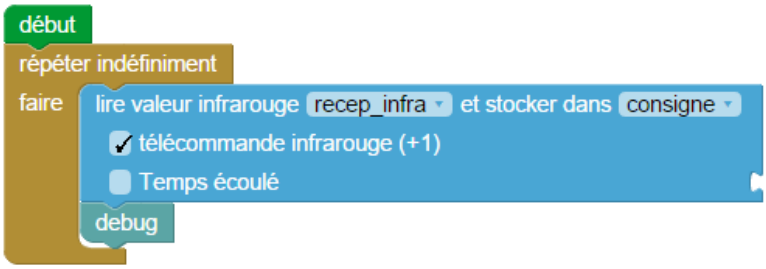


Conseil : Si la télécommande ne fonctionne plus, appuyer sur  pour revenir à la configuration compatible PICAXE

Remarque : Le guide d'utilisation complet de la télécommande est disponible ici :
http://www.a4telechargement.fr/RAX-TVR010/RAX-TRV10_Telecommande_InfraRouge.pdf

Tester la télécommande

Charger les programmes de test de la télécommande : « test_infra_bloc.xml » ou « test_infra_org.plf ».
Respecter le plan de câblage vu précédemment dans le dossier.



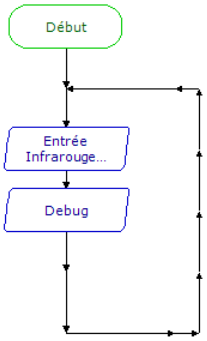


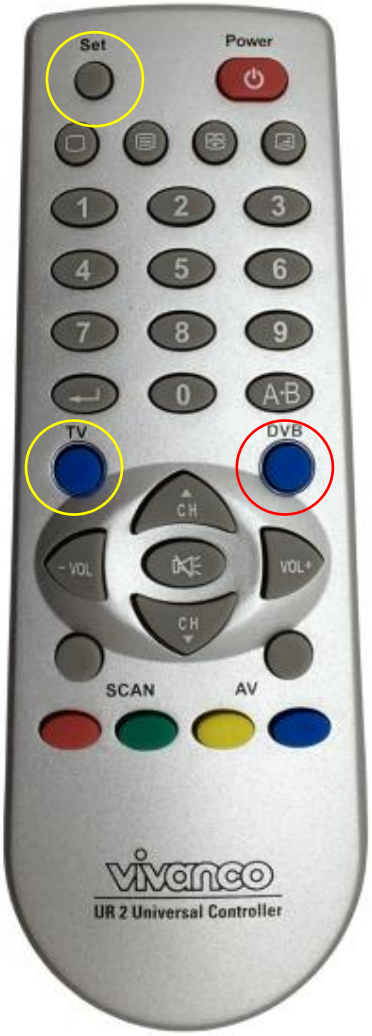
Tableau de correspondance des touches

Diriger la télécommande vers le récepteur infrarouge et vérifier dans la partie « Variables » de Picaxe Editor que les données reçues sont correctes.
Ci-dessous, le tableau des valeurs renvoyées par les différents boutons de la télécommande :

Touche	1	2	3	4	5	6	7	8	9	0	
Code émis	0	1	2	3	4	5	6	7	8	9	21
Code émis	16	17	19	18	96	54	37	20	98	11	

Télécommande TELEC-IR-UNIV

Procédure de mise en service pour PICAXE



1. Insérer 2 piles AAA dans le logement au dos de la télécommande.

2. Appuyer simultanément sur les boutons **Set** et **TV**.
   Le bouton **Power** s'allume.

3. Taper le code 0-7-7.
   Le bouton **Power** clignote brièvement à chaque appui, puis s'éteint.

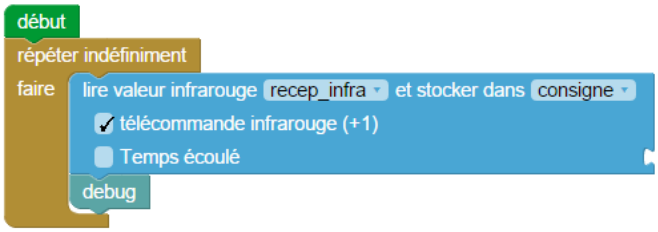
4. Appuyer sur le bouton **Power**.
   La télécommande est opérationnelle.

ATTENTION !
La touche **DVB** risque de changer le mode. Appuyer sur **TV** pour revenir dans le bon mode.

Conseil : Si la télécommande ne fonctionne plus, appuyer sur **TV** pour revenir à la configuration compatible PICAXE.

Tester la télécommande

Charger les programmes de test de la télécommande : « test_infra_bloc.xml » ou « test_infra_org.plf ».
Respecter le plan de câblage vu précédemment dans le dossier.



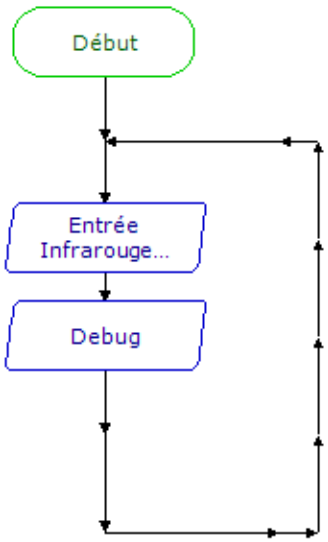
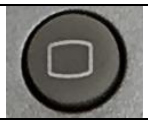
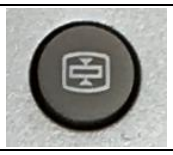
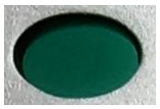


Tableau de correspondance des touches

L'appui sur une touche provoque l'émission d'un signal infrarouge qui véhicule un code correspondant à la touche. Par défaut : appui sur la touche 1 = envoi du code 0.

Pour simplifier l'utilisation de la télécommande infrarouge, il est possible d'activer la compatibilité entre le N° des touches et le code envoyé. Dans ce cas, appui sur la touche 1 = envoi du code 1.

Touche					
Code émis standard	**9**	**0**	**1**	**2**	**3**
Compatibilité activée	**10**	**1**	**2**	**3**	**4**
Touche					
Code émis standard	**4**	**5**	**6**	**7**	**8**
Compatibilité activée	**5**	**6**	**7**	**8**	**9**
Touche					
Code émis standard	**12**	**37**	**16**	**37**	**56**
Compatibilité activée	**13**	**38**	**17**	**38**	**57**
Touche					
Code émis standard	**63**	**74**	**29**	**21**	**76**
Compatibilité activée	**64**	**75**	**30**	**22**	**77**
Touche					
Code émis standard	**77**	**78**	**79**	**18**	**19**
Compatibilité activée	**78**	**79**	**80**	**19**	**20**
Touche					
Code émis standard	**20**	**16**	**17**		
Compatibilité activée	**21**	**17**	**18**		

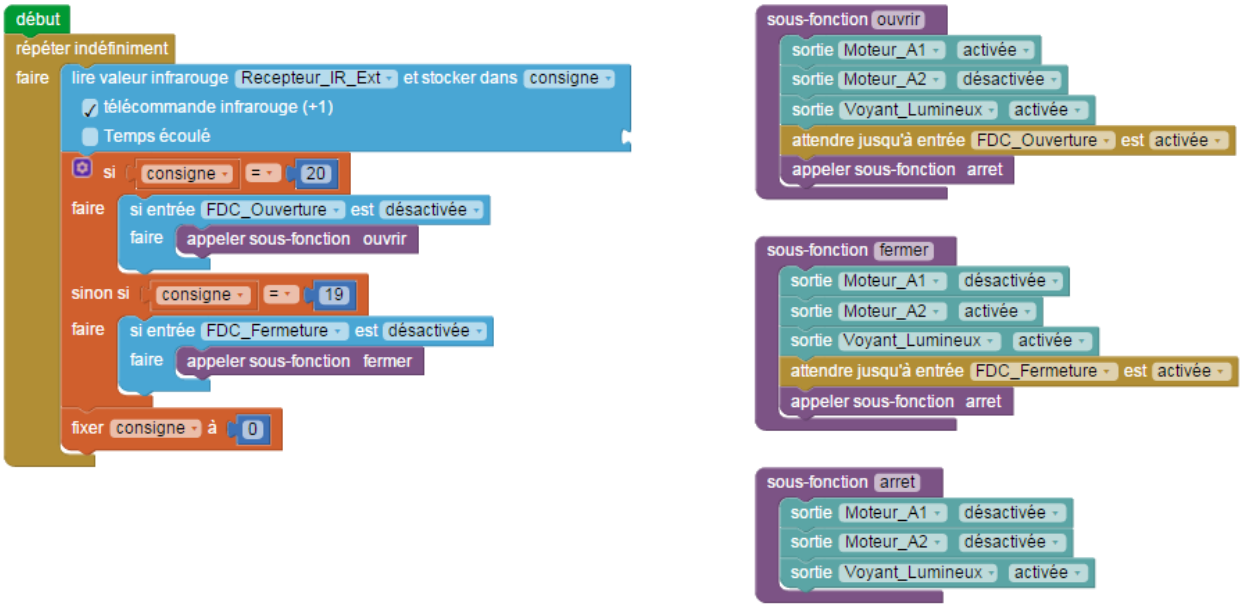
Exercice niveau 3 - A.1 : Contrôle d'ouverture/fermeture avec la télécommande IR

Objectif : ouvrir et fermer le portail à l'aide de la télécommande IR.

Notion abordée : gestion d'une liaison infra-rouge : télécommande/AutoProg à l'aide du bloc prévu à cet effet.

Correction :

Blocs



Fichier Blockly : PB1_N3_A1.xml

Remarque : Cocher la case +1 permet d'avoir une concordance entre la touche pressée et la consigne reçue.

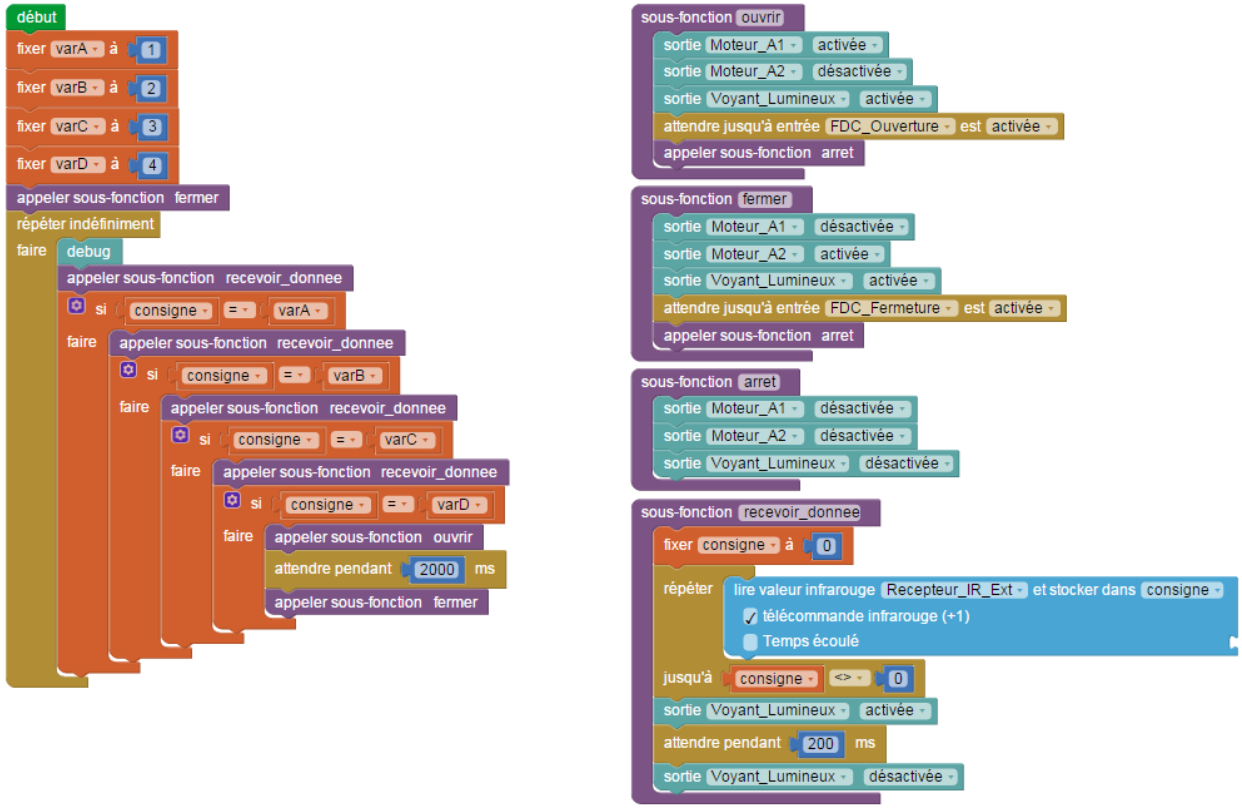
Exercice niveau 3 - A.2 : Ouvrir le portail avec un code

Objectif : ouvrir le portail dès la réception d'un code spécifique envoyée par la télécommande. Le fermer 2 secondes plus tard.

Notion abordée : gestion d'une liaison infra-rouge : télécommande/AutoProg.

Correction :

Blocs



Fichier Blockly : PB1_N3_A2.xml

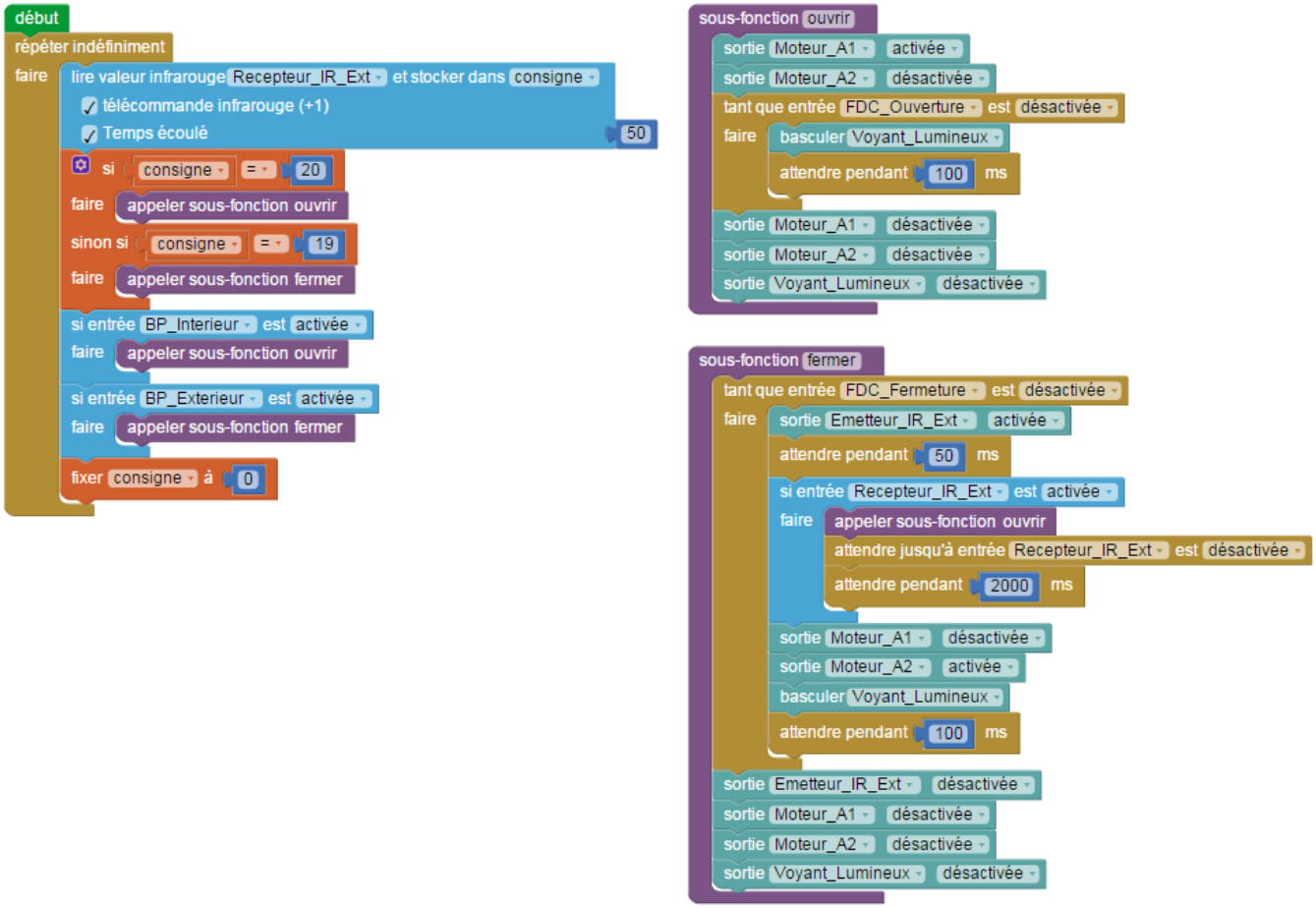
Exercice niveau 3 - A.3 : Télécommande IR + BP + Barrière IR

Objectif : reprendre le programme **PB2_N2_A4** et intégrer l'option ouverture par la télécommande.

Notion abordée : Gestion d'une liaison infra-rouge : télécommande/AutoProg en cohabitation avec la barrière IR (émetteur + récepteur).

Correction :

Blocs



Fichier Blockly : PB1_N3_A3.xml

Remarques : Dans cet exercice, la case « Temps écoulé » du bloc de lecture infrarouge est cochée. Cette option permet de sauter la commande si aucune touche n'a été appuyé par la télécommande pendant un temps de 50ms

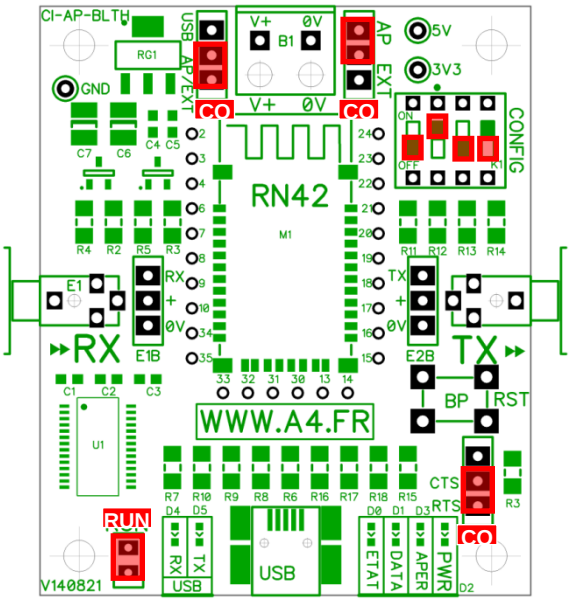
Option : Module Bluetooth

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 .
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.



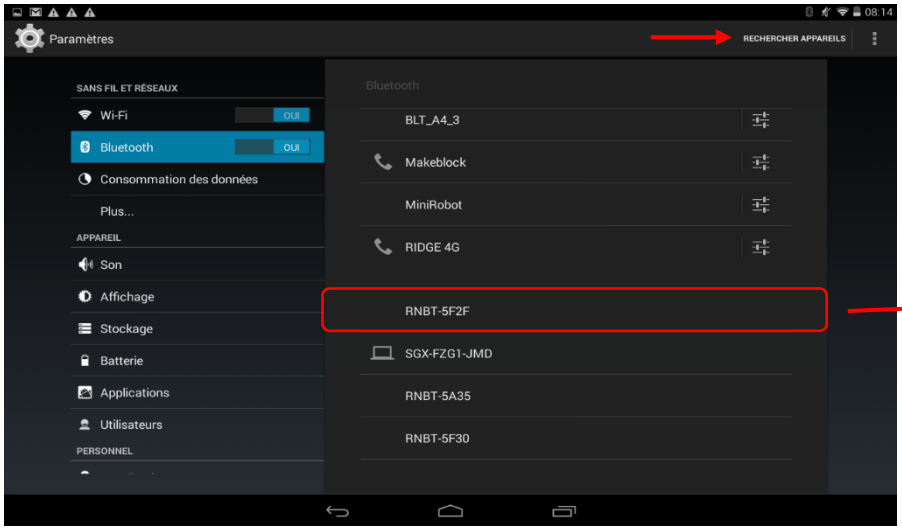
Le cavalier repéré **RUN** est utilisé lors de la mise au point de programmes avec **Arduino**. Il doit être ôté pour permettre le téléversement du programme puis doit être remis lors de l'utilisation.
La mise au point de programmes avec **PICAXE** ne nécessite pas d'ôter ce cavalier pour transférer 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

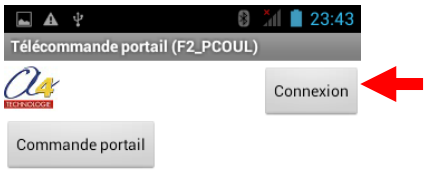
PWR indique que le module est sous tension.
APER indique que le module est associé avec un matériel Bluetooth.
DATA indique qu'il y a un flux de données entre le module et l'appareil avec lequel il est connecté.
ETAT indique que le module est opérationnel. L'affichage clignotant indique qu'il n'est pas 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 allimenter le module). Le nom de votre module s'appelle : RNBT + les 4 derniers chiffres de l'adresse mac du module notés sur le composant. Selectionnez 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 AppInventor 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.



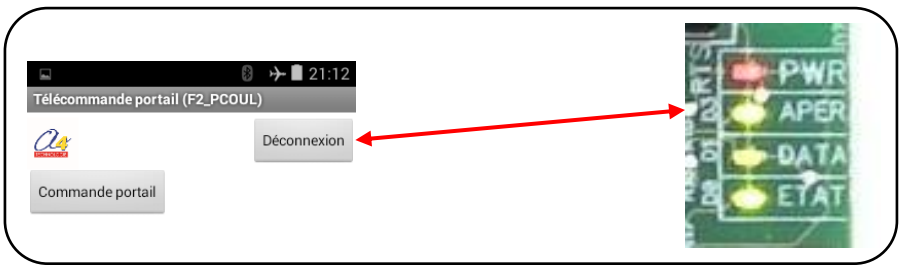
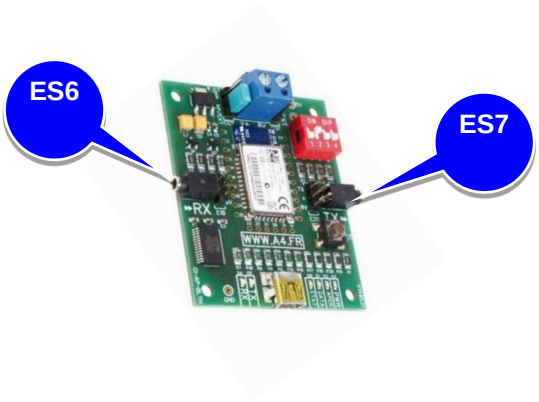


Tableau d'affectation des entrées et sorties

	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
ES			
7	**Communication Bluetooth envoi de données**	**C.7**	**BLTH_TX***
6	**Communication Bluetooth réception de données**	**C.6**	**BLTH_RX***
EN	**MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES**		
5	Récepteur barrière infrarouge intérieur (option)	C.5	Recepteur_IR_Int*
4	Récepteur barrière infrarouge extérieur	C.4	Recepteur_IR_Ext
3	Bouton poussoir extérieur	C.3	BP_Exterieur
2	Capteur de fin de course fermeture du portail	C.2	FDC_Fermeture
1	Capteur de fin de course ouverture du portail	C.1	FDC_Ouverture
0	Bouton poussoir intérieur	C.0	BP_Interieur
EA	**MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES**		
	(libre)	A.3	
2	*(libre)*	A.2	
1	*(libre)*	A.1	
0	*(libre)*	A.0	
SN	**MODULES ACTIONNEURS SORTIES NUMÉRIQUES**		
	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	Module sonore Buzzer (option)	B.3	Buzzer
2	Emetteur barrière infrarouge intérieur (option)	B.2	Emetteur_IR_int*
1	Emetteur barrière infrarouge extérieur	B.1	Emetteur_IR_Ext
0	Module signal LED jaune	B.0	Voyant_Lumineux

Câblage du module bluetooth (K-AP-MBLTH)



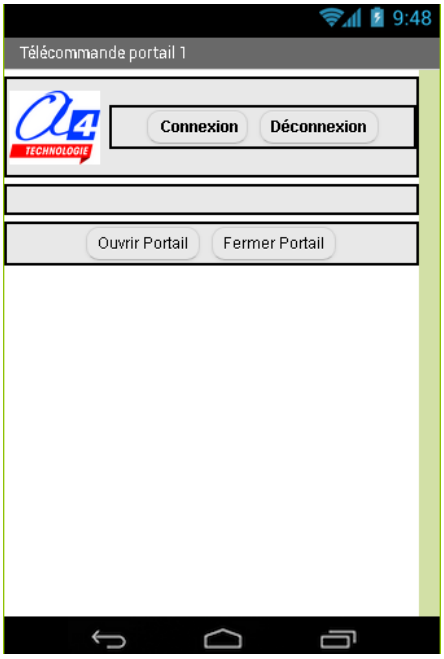
Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth

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.

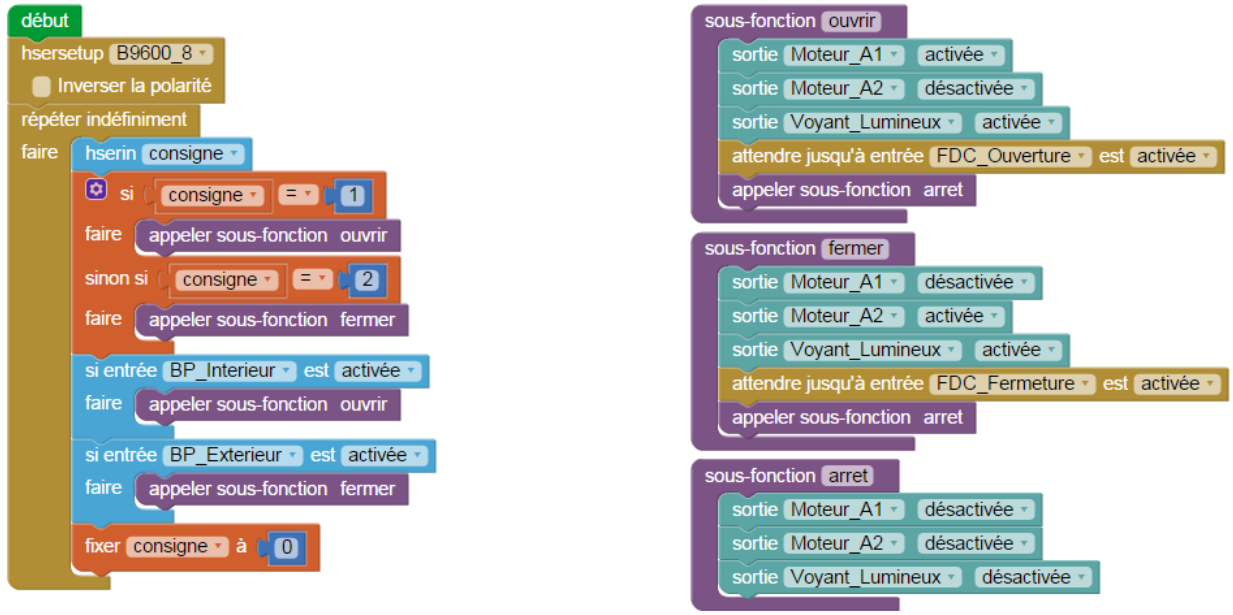
Application Android : Portail_1.apk

Fichier App Inventor : Portail_1.aia



Correction :

Blocs



Fichier Blockly : PB1_N3_B1.xml

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.

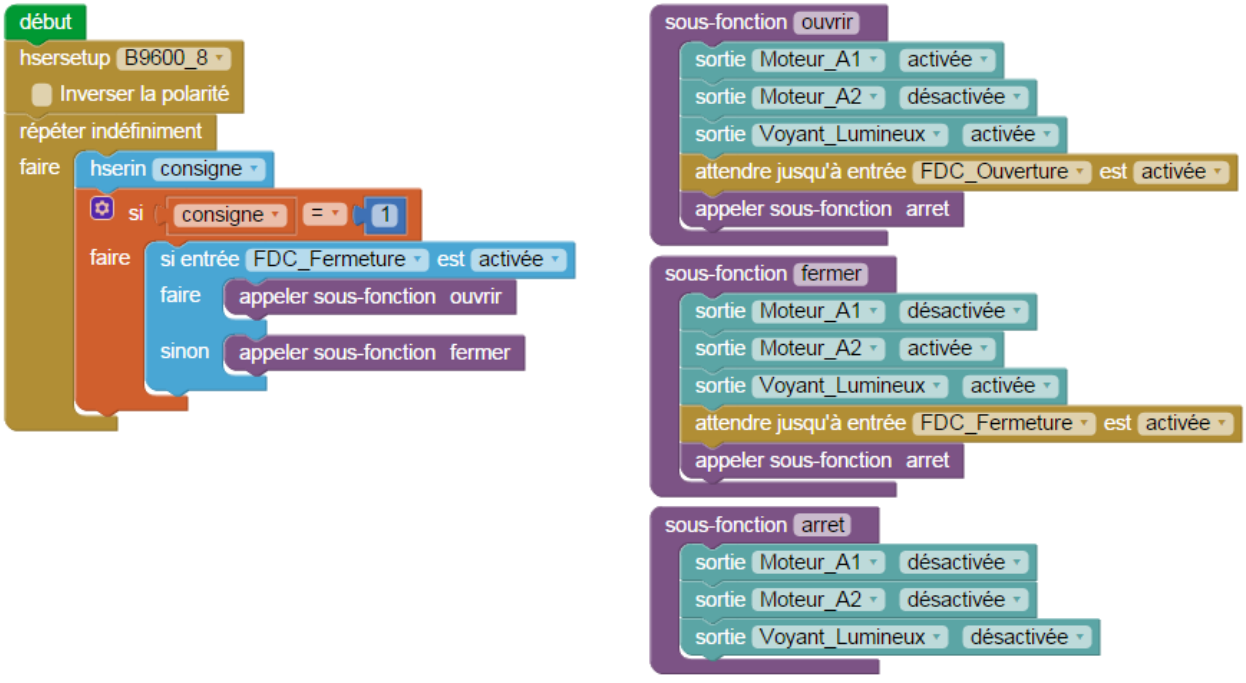
Notion abordée : réception de données Bluetooth envoyées par un Smartphone.

Application Android : Portail_2.apk

Fichier App Inventor : Portail_2.aia

Correction :

Blocs



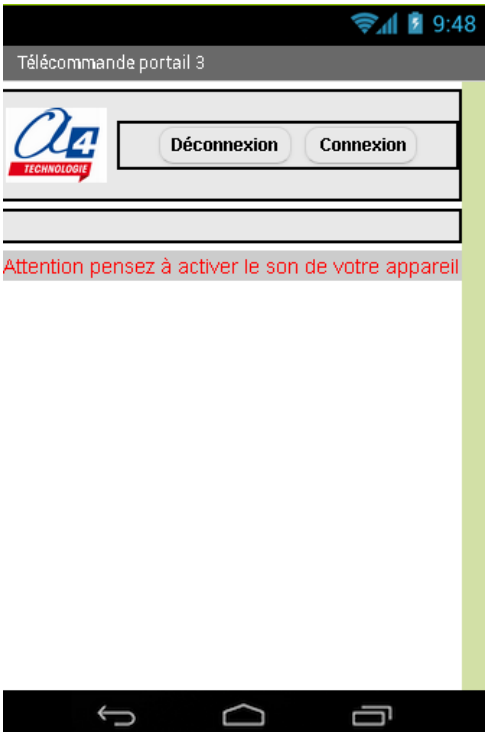
Fichier Blockly : PB1_N3_B2.xml

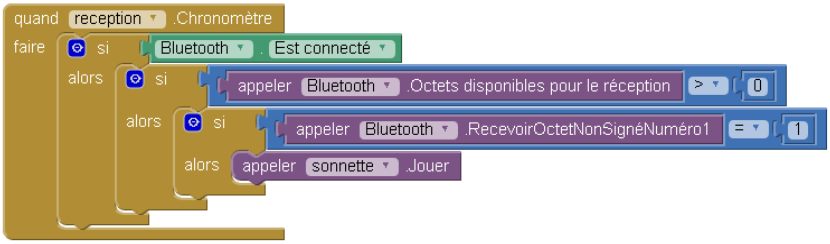
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone

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

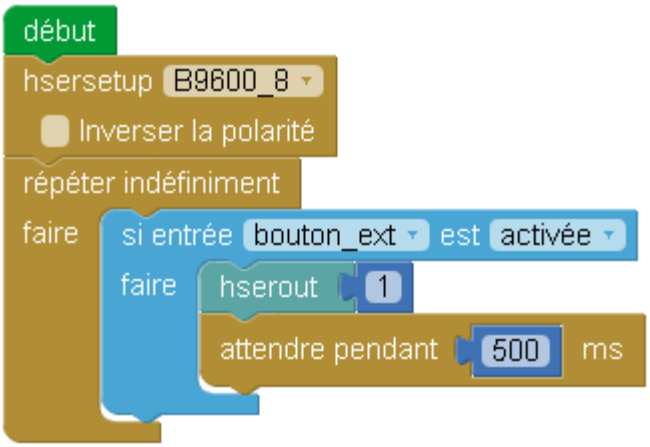
Notion abordée : envoyer des informations à un Smartphone par Bluetooth.

Application Android : Portail_3.apk





Correction :

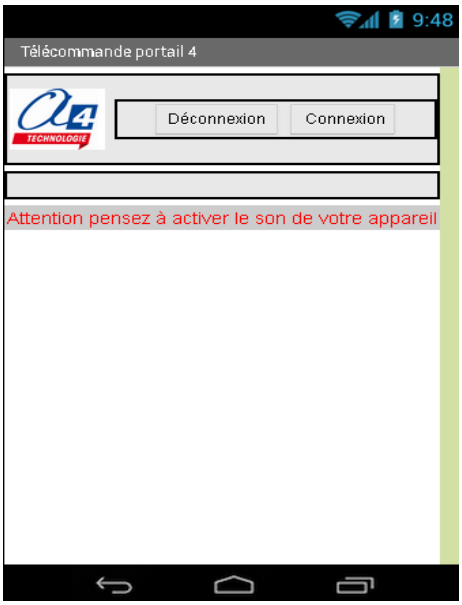
Blocs

Fichier Blockly : PB1_N3_B3.xml

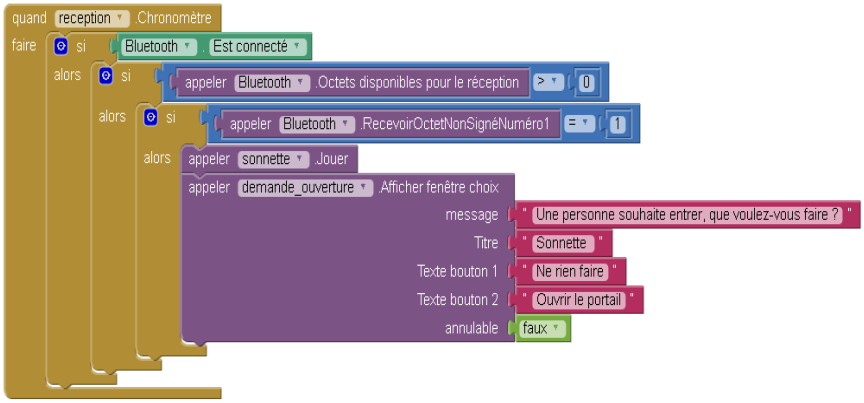
Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone

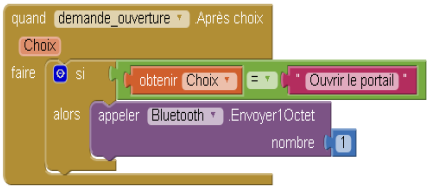
Objectif : gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.

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

Application Android : Portail_4.apk

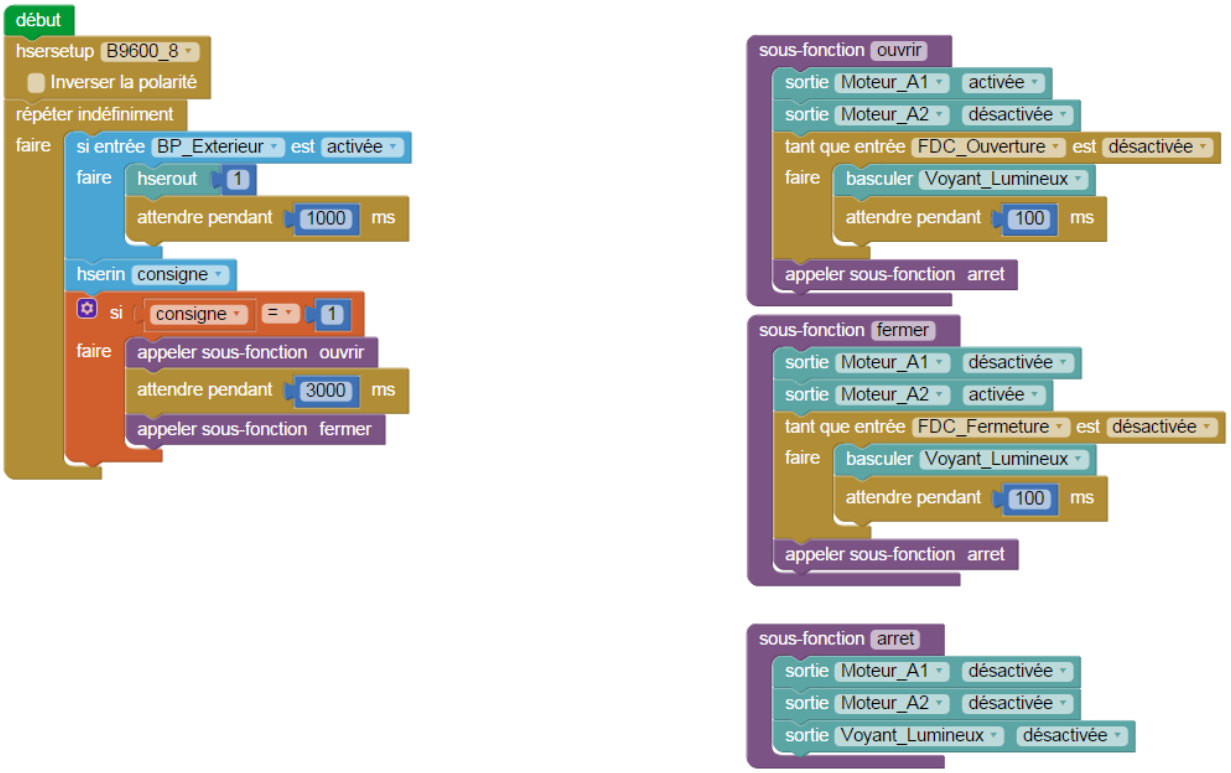






Correction :

Blocs



Fichier Blockly : PB1_N3_B4.xml

Option : Modules infrarouge intérieur

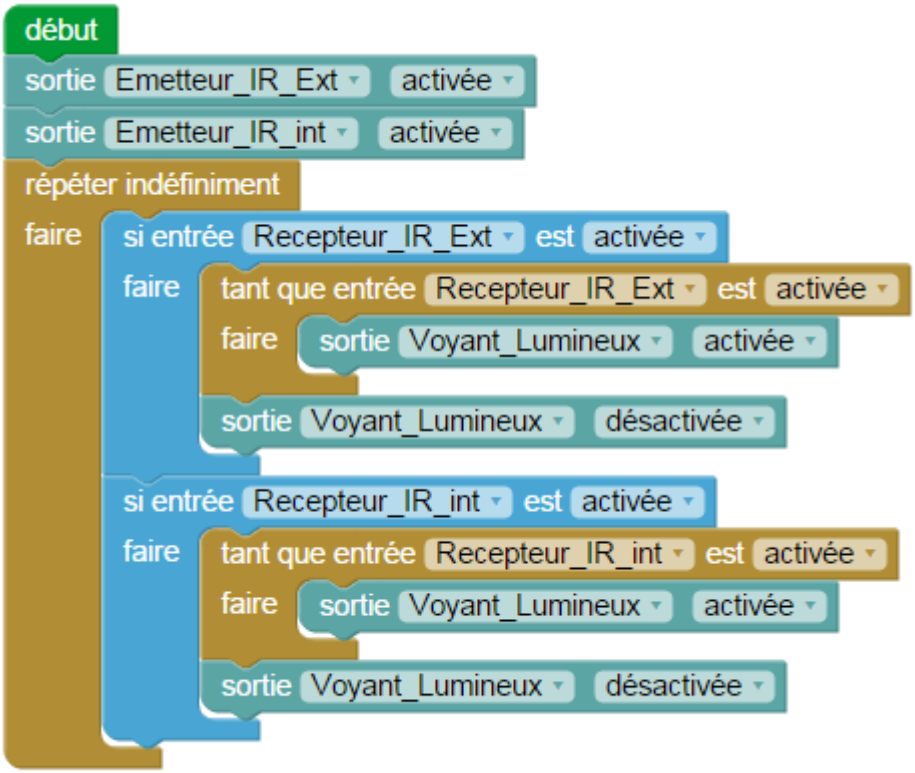
Exercice niveau 3 - C.1 : Utilisation de deux barrières optiques

Objectif : Allumer le voyant lumineux lorsqu'un obstacle franchit la barrière infrarouge intérieure ou extérieure. L'éteindre lorsqu'il n'y a pas d'obstacles entre les deux.

Notion abordée : gestion de deux groupes de capteurs

Correction :

Blocs



Fichier Blockly : PB1_N3_C1.xml

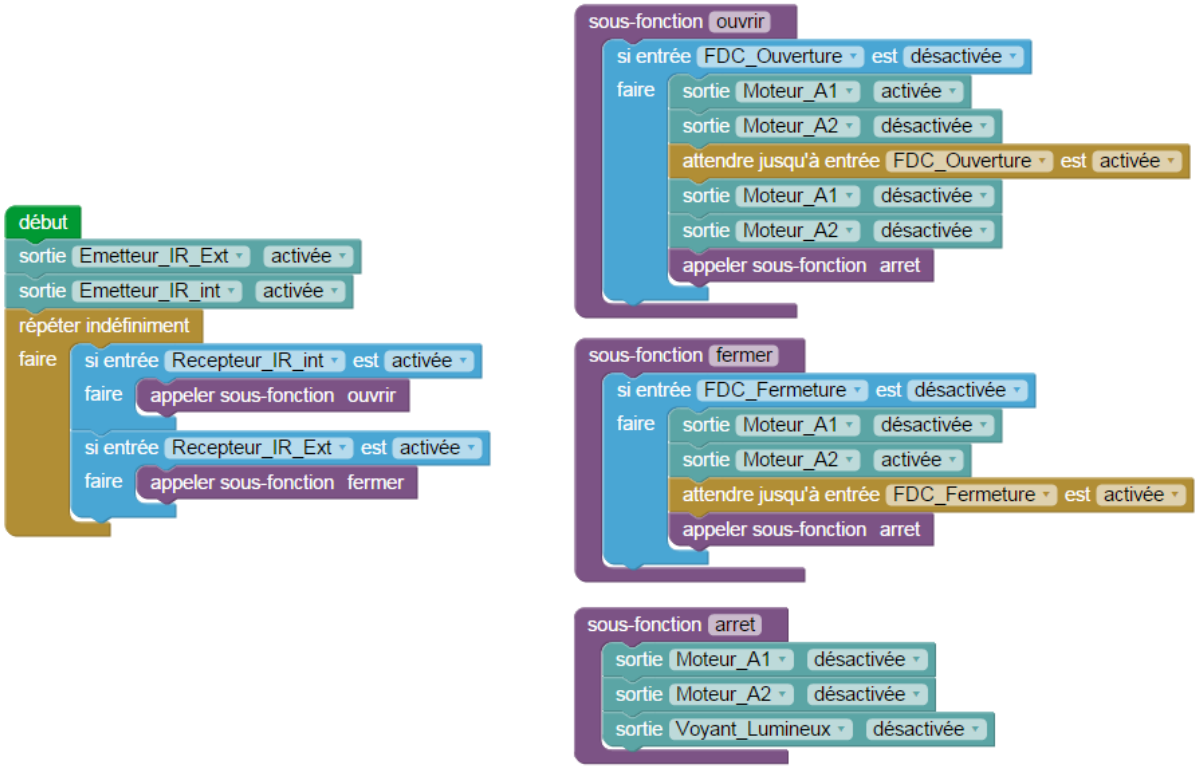
Exercice niveau 3 - C.2 : Utilisation de deux barrières optiques

Objectif : Ouvrir le portail lors d'un passage entre l'émetteur et le récepteur infrarouge intérieur et fermer le portail lors du passage entre les capteurs à l'extérieur.

Notion abordée : gestion de deux groupes de capteurs

Correction :

Blocs



Fichier Blockly : PB1_N3_C2.xml

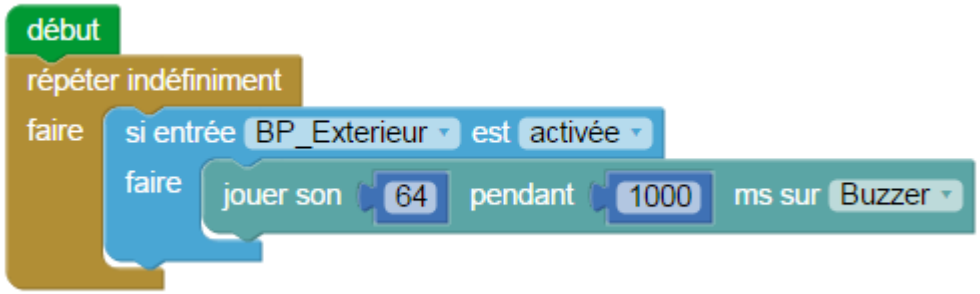
Option : Module buzzer

Exercice niveau 3 - D.1 : Jouer un son

Objectif : Grâce à une fonction déjà faite sur PICAXE Editor, jouez un son de 1 seconde sur le buzzer à chaque fois qu'un appui sur un bouton poussoir est effectué

Instruction utilisée :

Correction :

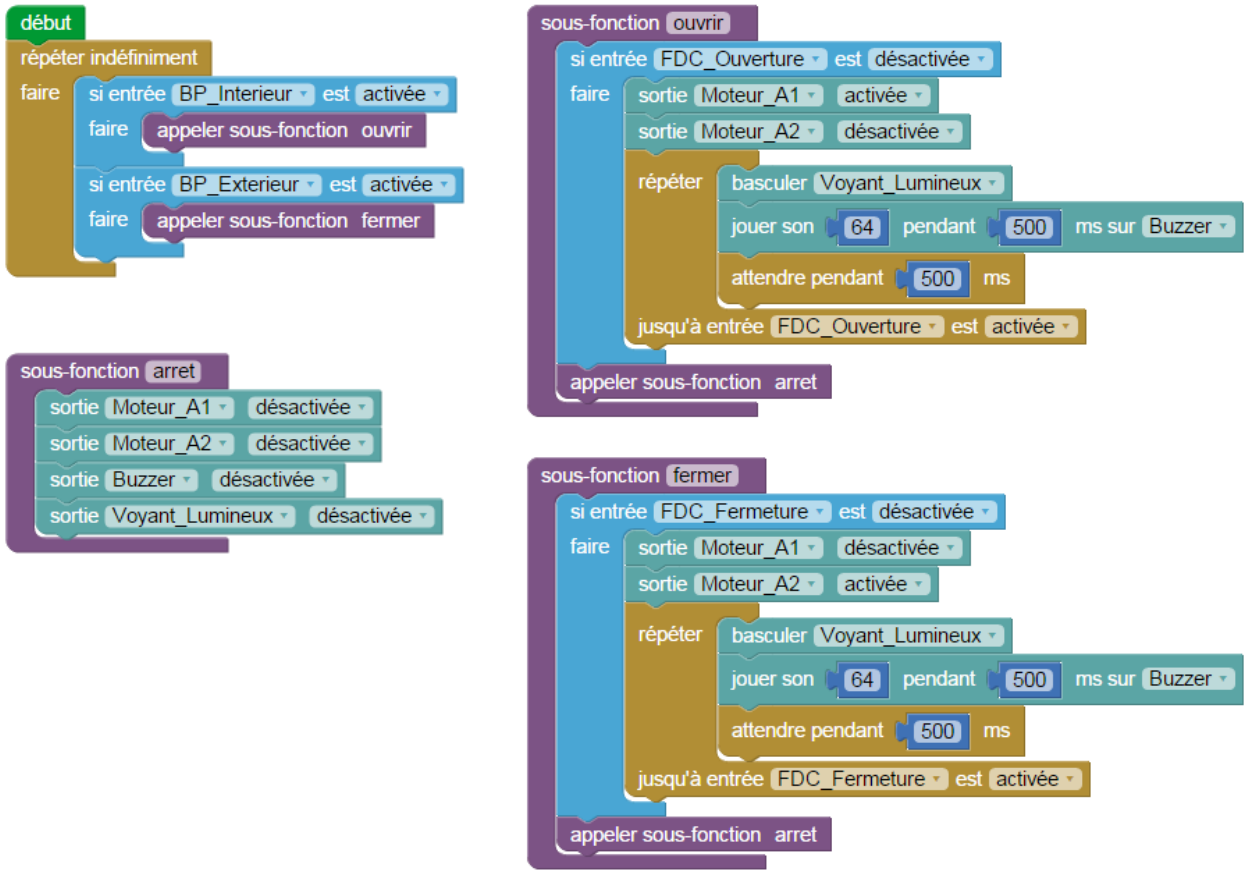
Blocs

Fichier Blockly : PB1_N3_D1.xml

Exercice niveau 3 - D.2 : Signal sonore

Objectif : Lors d'un mouvement du portail (ouverture/fermeture) jouer un son toutes les 500ms jusqu'à une fin de course.

Correction :

Blocs



Fichier Blockly : PB1_N3_D2.xml

CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES