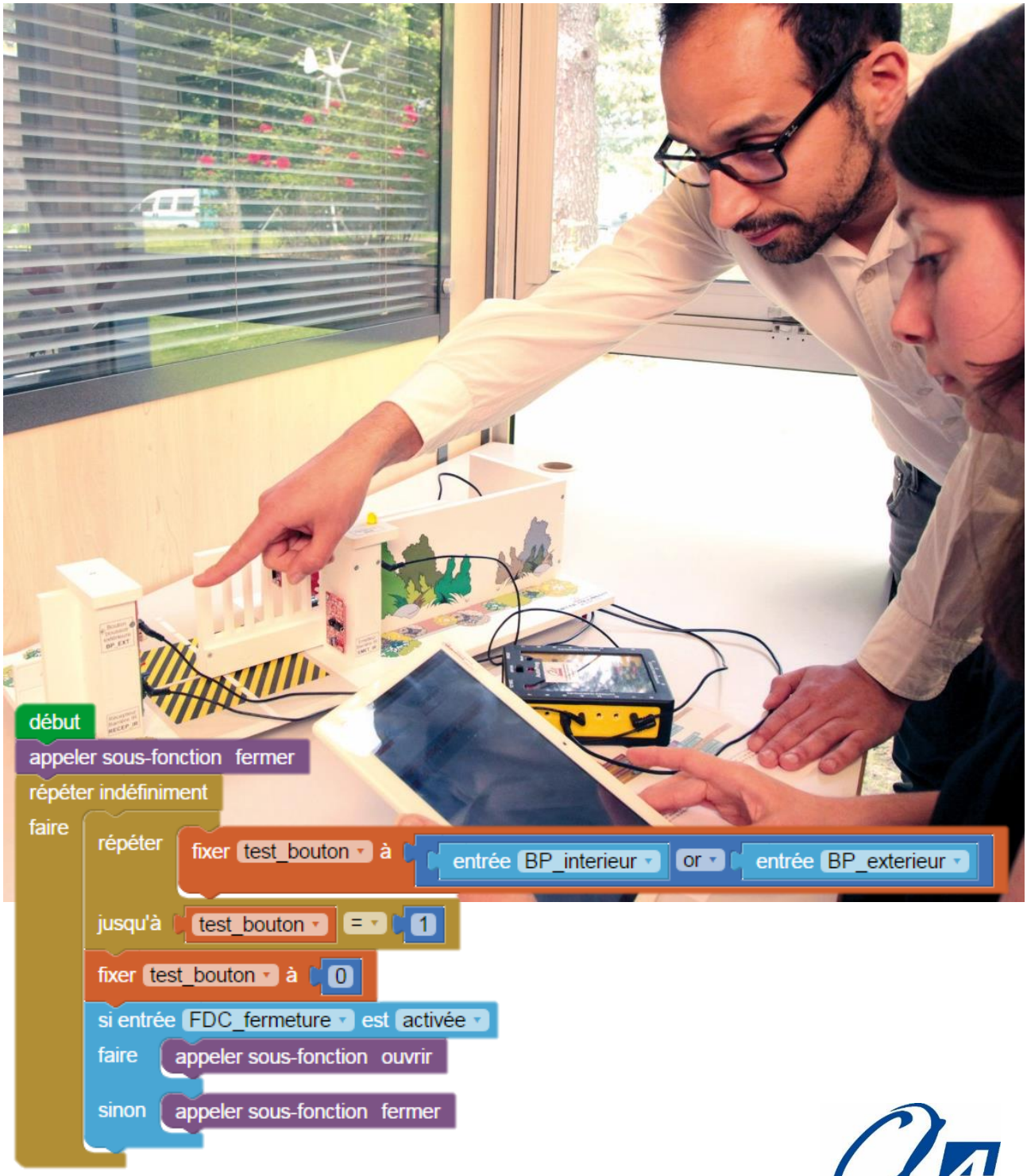
Portail coulissant

Maquette motorisée programmable avec Picaxe Editor / Blockly



A4 TECHNOLOGIE

Ressources disponibles pour le projet Portail coulissant

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

Portail coulissant

- Fichiers **3D** (SolidWorks, Edrawings et Parasolid) de la maquette et de ses options.
- Dossier technique du Portail Coulissant 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

Vous trouverez dans ce document tout le nécessaire pour démarrer des activités de programmation autour du portail coulissant.

Le portail coulissant

La maquette portail coulissant (BE-APORT-COUL) est une reproduction homothétique d'un portail coulissant automatisé réel : roue dentée, crémaillère, capteurs fin de course, barrière optique, clignotant de sécurité, etc. Programmable et pilotée par les systèmes AutoProgX2 ou AutoProgUno, elle permet une activité de programmation complète par rapport aux attendus de fin de cycle collège : l'algorithmique en maths, l'étude de scénarios, la programmation et la mise en œuvre en Technologie.

Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour du Loupiot 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 Portail coulissant grâce à ses capteurs et actionneurs.
Il est organisé en fonction des niveaux de programmation.

Niveau 1 :
Découverte progressive du jeu d'instructions et des fonctionnalités de base du portail coulissant 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 du portail (version de base).

Niveau 3 :
Exemples d'utilisation des différentes options proposées autour du portail coulissant : télécommande infrarouge, module capteur de courant, 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 :

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

Exemple : PC_N2_C3.xml
Correspond au niveau 2 avec le jeu d'instructions C, adapté aux objectifs « avancés » de ce niveau.

Prérequis

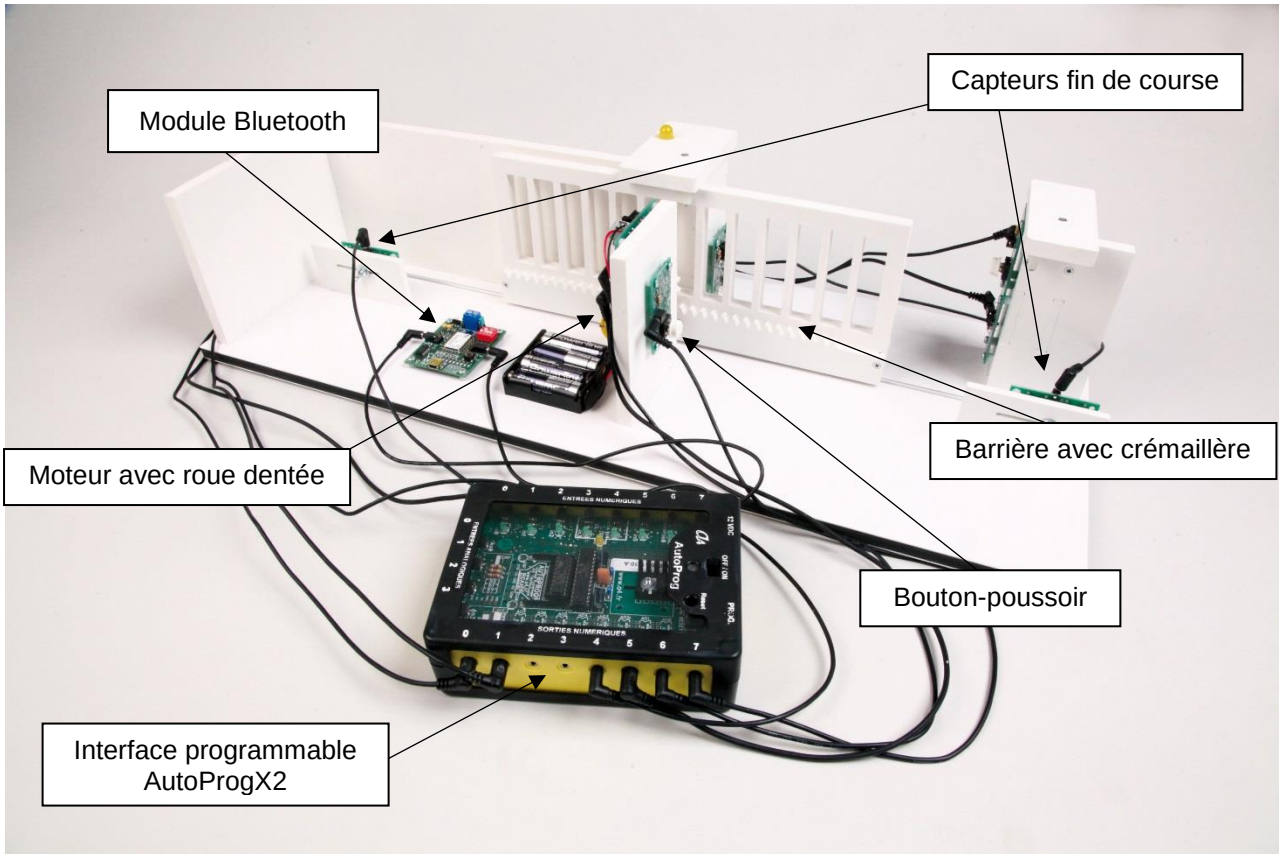
Pour la version de base :

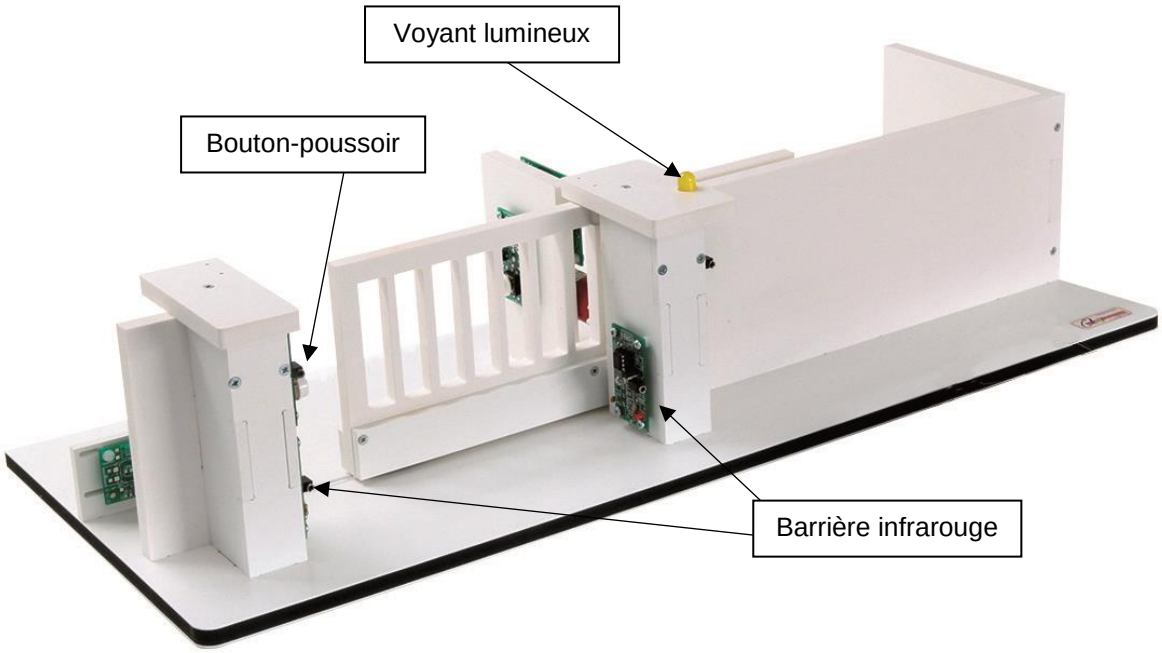
- Installer le logiciel **Picaxe Editor 6** ou **Blockly for Picaxe :** http://www.picaxe.com/Software
- **Maquette** de portail coulissant (Réf. BE-APOR-COUL).
- **Câble de programmation** Picaxe USB (Réf : CABLE-USBPICAXE).
- **Interface programmable** AutoProgX1 ou X2 (Réf. K-APV2).
- Onze **cordons 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

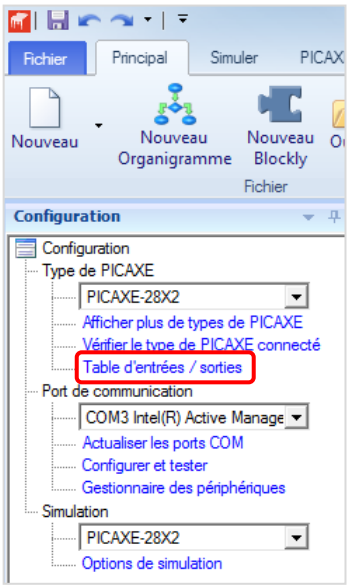
Tous les programmes correspondant aux activités menées autour du Portail coulissant ont été réalisés sous **PICAXE Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs avantages :

- 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.

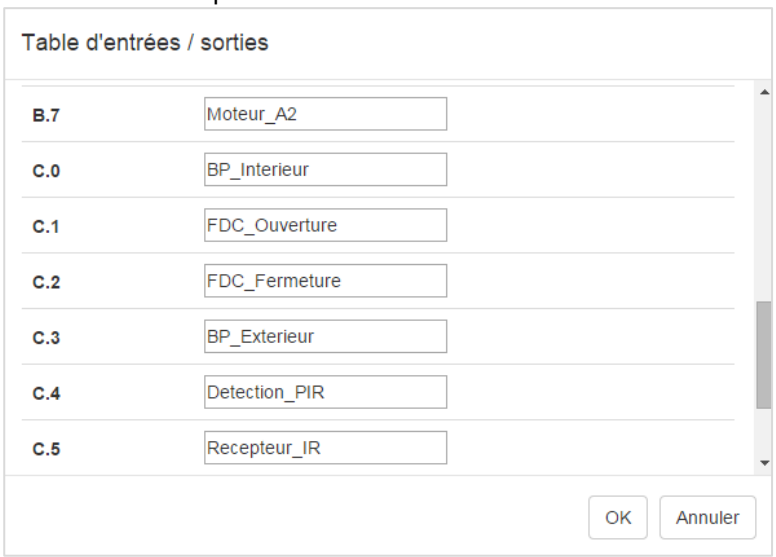
Note : vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

Personnalisation des entrées/ sorties

Tous les programmes et activités proposés dans ce document se basent sur une table d'entrées/sorties personnalisée pour une utilisation avec le Portail Coulissant. 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**.



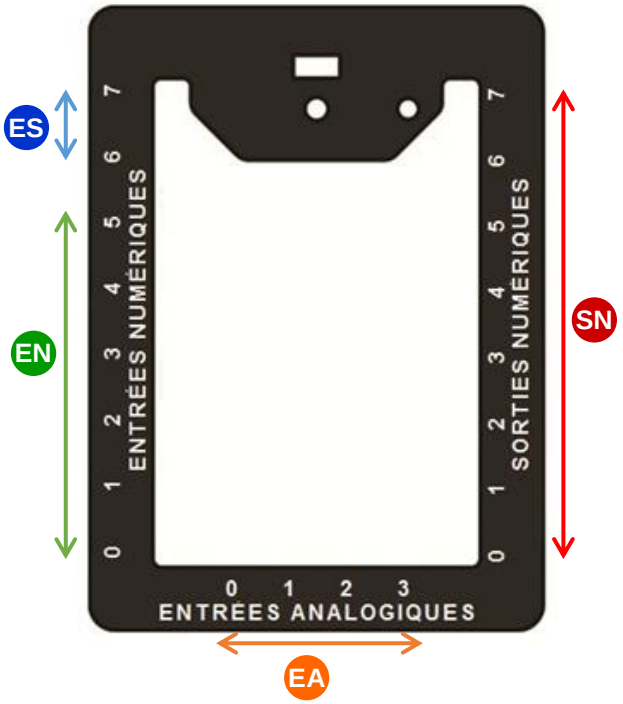
Une fenêtre apparaît à partir de laquelle vous pouvez modifier les noms de toutes les entrées et sorties dans la zone « Mon étiquette ».



Valider en cliquant sur **OK**.

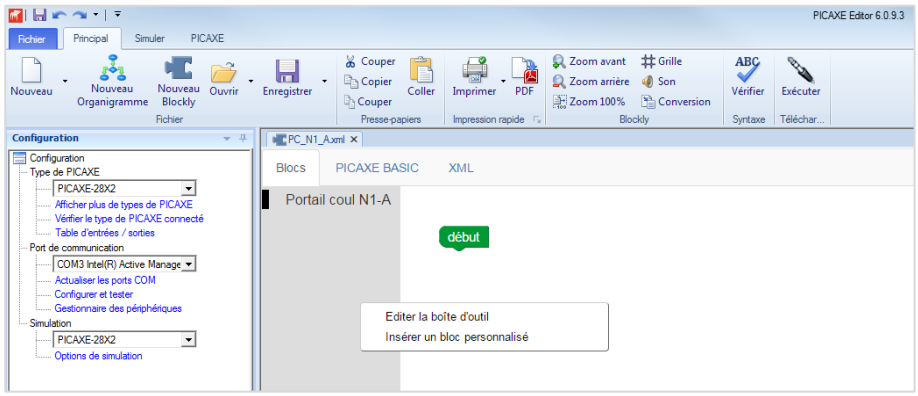
Tableau d'affectation des entrées et sorties du Portail coulissant

ES	Modules de communication pour entrées / sorties numériques	Broche	Etiquette Blockly
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	C.5	Recepteur_IR
4	Capteur détection de présence (option)	C.4	Detection_PIR*
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**		
3	*(libre)*	*A.3*	
2	*(libre)*	A.2	
1	*(libre)*	A.1	
0	Capteur de courant analogique/numérique (option)	A.0	Capteur_courant
SN	**Modules actionneurs sorties numériques**		
7	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.7	Moteur_A2
6	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.6	Moteur_A1
5	*(libre)*	B.5	
4	*(libre)*	B.4	
3	*(libre)*	B.3	
2	*(libre)*	B.2	
1	Emetteur barrière infrarouge	B.1	Emetteur_IR
0	Module signal LED jaune	B.0	Voyant_Lumineux

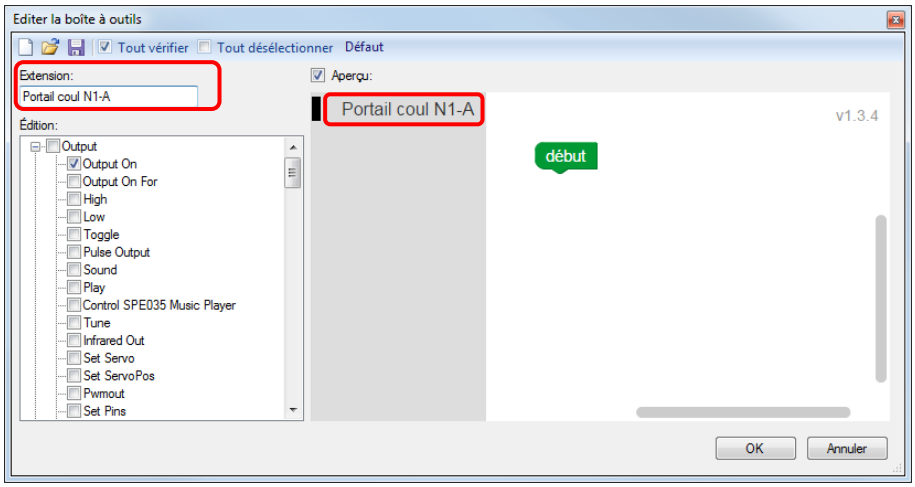


Personnalisation du jeu d'instructions

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**.



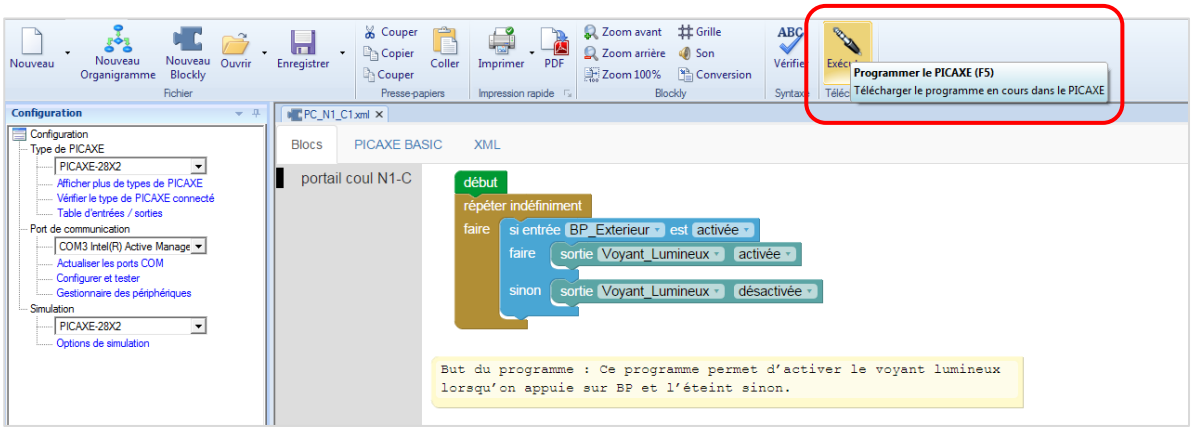
Une fenêtre s'ouvre à partir de laquelle vous pouvez sélectionner ou désélectionner les instructions de votre choix. Vous pouvez renommer le jeu d'instructions dans la zone « **Extension** ».



Valider en cliquant sur **OK**.

Procédure de chargement d'un programme

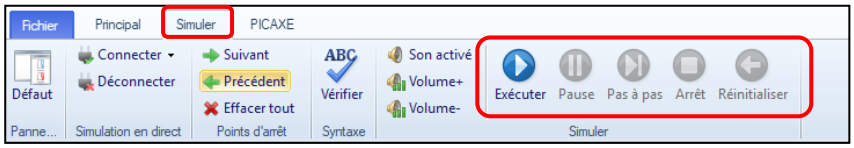
Commencer par relier le Loupiot à l'ordinateur avec le câble de programmation USB et le mettre sous tension.
A partir du Picaxe Editor 6, ouvrir un programme.



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.

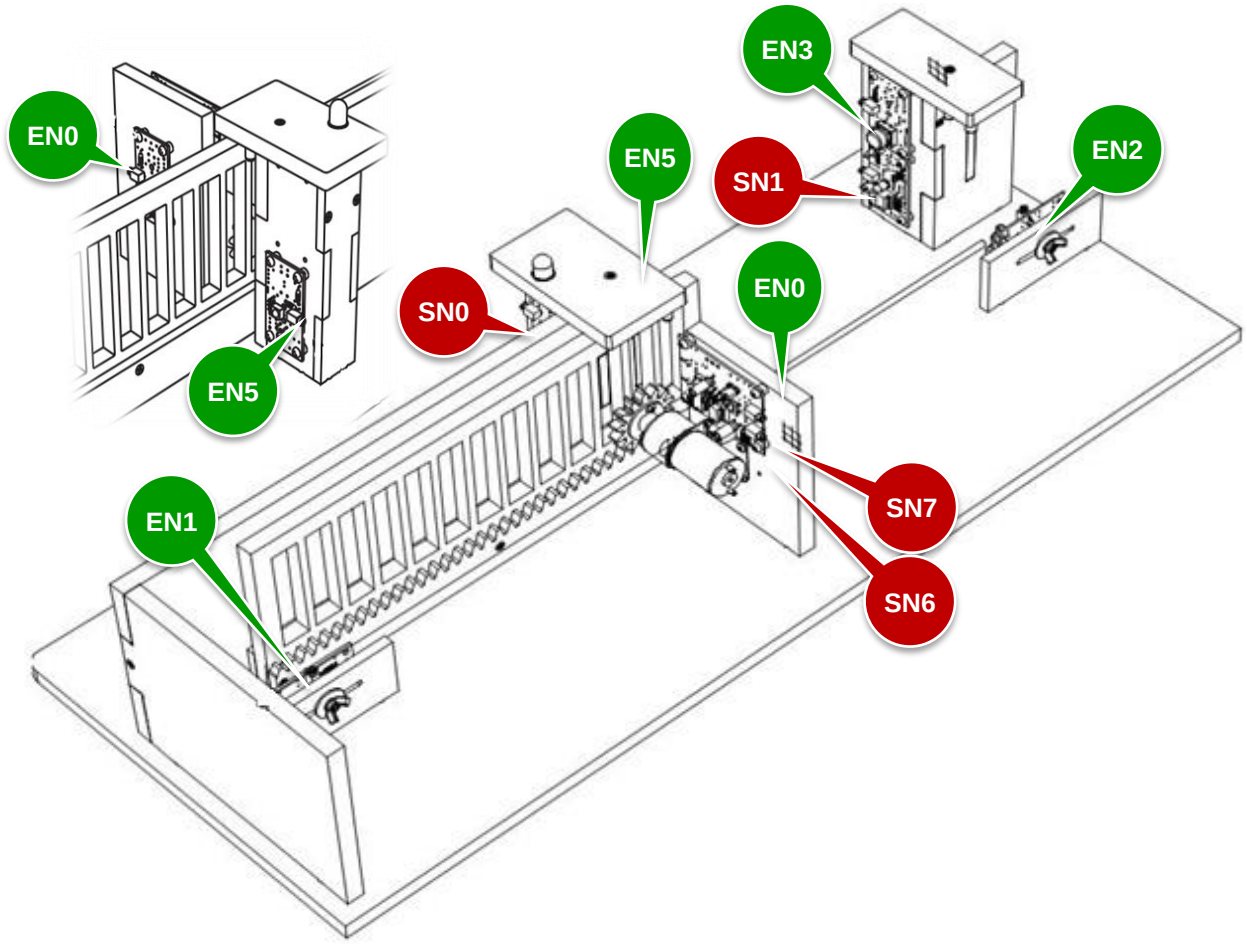
Mode simulation

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**.



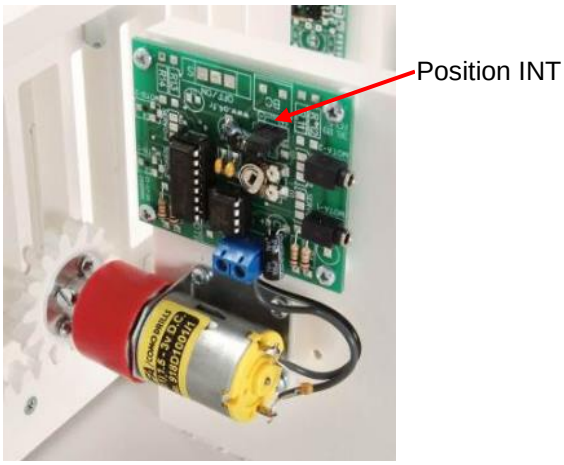
La simulation surligne les blocs dans l'espace de travail pour vous montrer où en est le programme.

Plan de câblage du portail coulissant

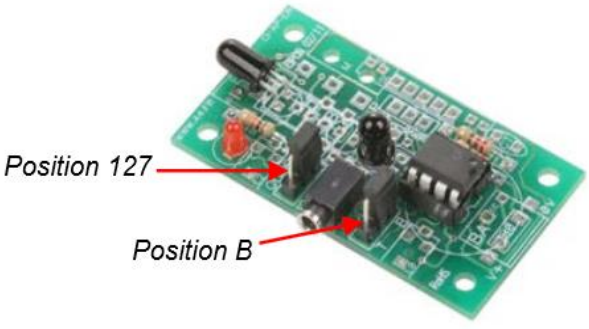


RAPPEL

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



Configuration du module récepteur infrarouge :
Les cavaliers de configuration sont sur les positions **B et 127** (mode barrière IR).



Programmes de test

Une fois votre maquette assemblée et câblée, vous pouvez procéder à cette vérification en 10 étapes afin de vous assurer du bon fonctionnement des modules du portail.
Si le programme ne se charge pas dans l'AutoProg, référez-vous à http://a4.fr/wiki/index.php/Probl%C3%A8me_de_chargement_d%27un_programme_PICAXE .
Si à une instruction vous n'obtenez pas le comportement attendu, vérifiez votre câblage.
Enfin, si vous n'arrivez pas à résoudre le problème, vous pouvez contacter notre équipe d'assistance : http://www.a4.fr/conseils_assistance/ .

Etape	Instruction	Comportement attendu
1	Retirer la barrière par sécurité	/
2	Charger le programme PC_N2_A4 dans l'AutoProg.	Le moteur doit tourner (dans le sens prévu pour fermer le portail) et le voyant lumineux doit clignoter (il clignotera à chaque fois que le moteur tourne).
3	Appuyer sur le capteur fin de course fermeture.	Le moteur doit s'arrêter de tourner et le voyant lumineux doit s'éteindre (il restera éteint à chaque fois que le moteur est à l'arrêt).
4	Maintenir le capteur fin de course fermeture appuyé et appuyer sur un des boutons-poussoirs puis relâcher le capteur fin de course.	Le moteur doit tourner dans l'autre sens.
5	Appuyer sur le capteur fin de course ouverture.	Le moteur doit s'arrêter.
6	Appuyer sur l'autre bouton-poussoir.	Le moteur doit tourner dans le sens de la fermeture.
7	Passer la main entre l'émetteur et le récepteur infrarouge.	Le moteur doit changer de sens (sens d'ouverture).
8	Appuyer sur le capteur fin de course ouverture.	Le moteur doit s'arrêter.
9	Attendre 2 secondes.	Le moteur doit tourner dans le sens de fermeture.
10	Appuyer sur le capteur fin de course fermeture.	Le moteur doit s'arrêter.

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 coulissant, en apprenant les structures de base de la programmation, 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 (fourni avec les exercices), avant de regarder la correction et l'explication de chaque programme.
Par exemple, pour le programme « PC_N1_A1.xml », charger le programme modèle « PC_N1_A.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 coulissant.

Nom du fichier	Description	Objectif
Niveau 1 A **Fichier modèle :** PC_N1_A.xml		
PC_N1_A1	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
PC_N1_A2	Répéter cette action deux fois.	
PC_N1_A3	Répéter cette action à l'infini.	
Niveau 1 B **Fichier modèle :** PC_N1_B.xml		
PC_N1_B1	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
PC_N1_B2	Ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.	
Niveau 1 C **Fichier modèle :** PC_N1_C.xml		
PC_N1_C1	allumer le voyant lumineux à l'appui du bouton-poussoir.	Fonctionnalité matérielle abordée : -Gestion des modules infra-rouge -Utilisation de Bouton-poussoir Notions de programmation abordées : -Le test d'une entrée (si/sinon)
PC_N1_C2	activer le voyant lumineux lorsque la barrière infrarouge est franchie.	
PC_N1_C3	contrôler l'allumage du voyant et du moteur avec des boutons-poussoirs.	
Niveau 1 D **Fichier modèle :** PC_N1_D.xml		
PC_N1_D1	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).	Notions de programmation abordées : -Définition de variable -Incrémentation de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage
PC_N1_D2	Incrémenter une variable au cours du temps, faire un test sur celle-ci pour activer le voyant.	
PC_N1_D3	Incrémenter une variable à l'appui d'un bouton-poussoir, la décrémenter à l'appui de l'autre bouton-poussoir.	
PC_N1_D4	incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.	

Niveau 1 - A

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

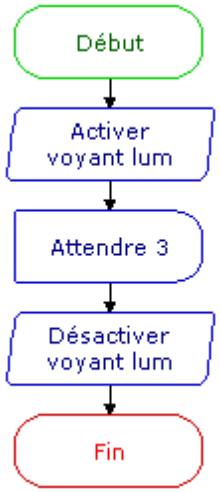
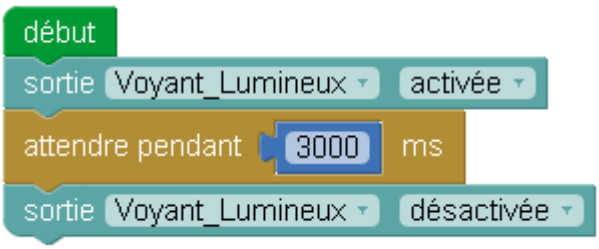
Fichier modèle : PC_N1_A.xml

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 : PC_N1_A1_Organigramme.plf	Fichier Blockly : PC_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

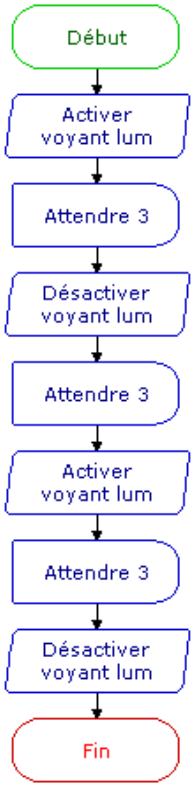
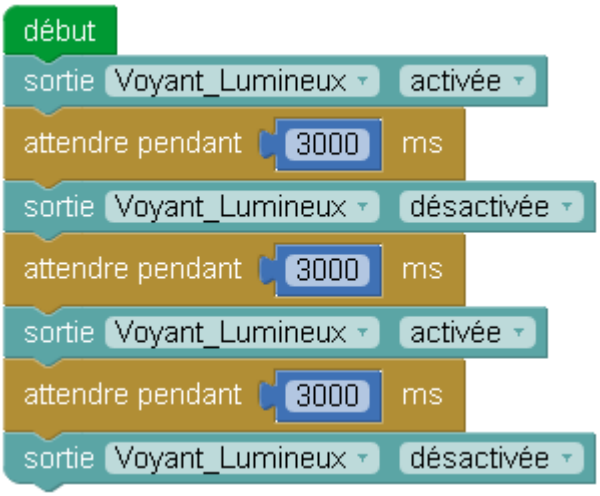
Fichier modèle : PC_N1_A.xml

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 : PC_N1_A2_Organigramme.plf	Fichier Blockly : PC_N1_A2.xml

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

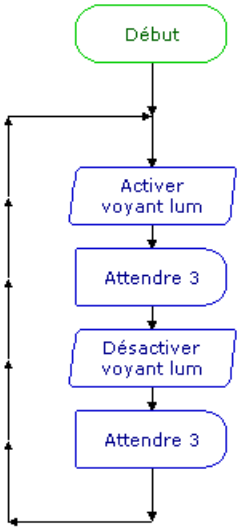
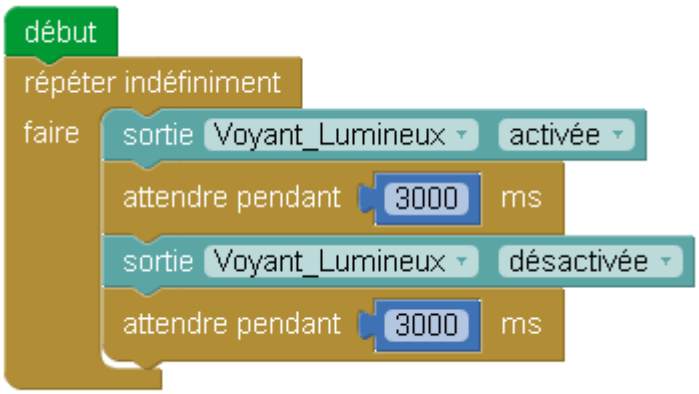
Fichier modèle : PC_N1_A.xml

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 : PC_N1_A3_Organigramme.plf	Fichier Blockly : PC_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 AutoProg.

Niveau 1 - B

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

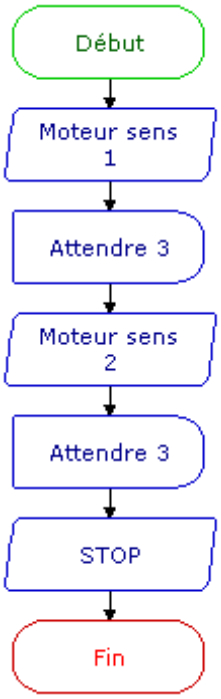
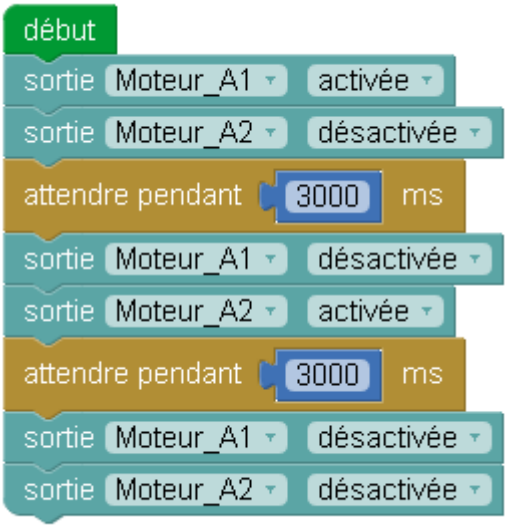
Fichier modèle : PC_N1_B.xml

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

Notion abordée : utilisation d'un moteur.

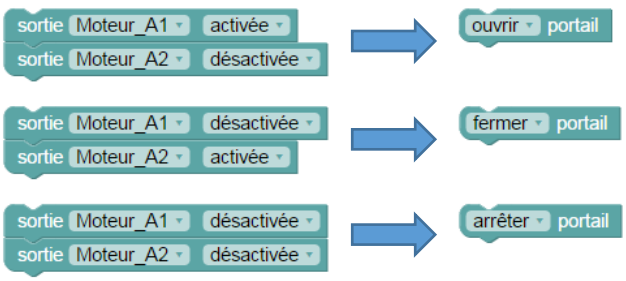
Instructions utilisées :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_B1_Organigramme.plf	Fichier Blockly : PC_N1_B1.xml

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

Information : Des blocs spécifiques sont disponibles pour contrôler le portail dans le menu extensionA4 (uniquement sur Editor 6) :



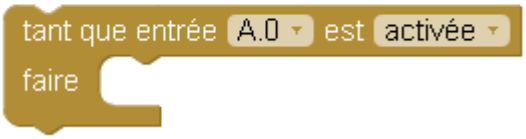
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que

Fichier modèle : PC_N1_B.xml

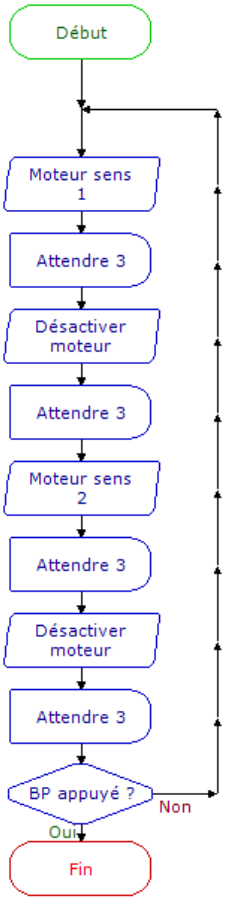
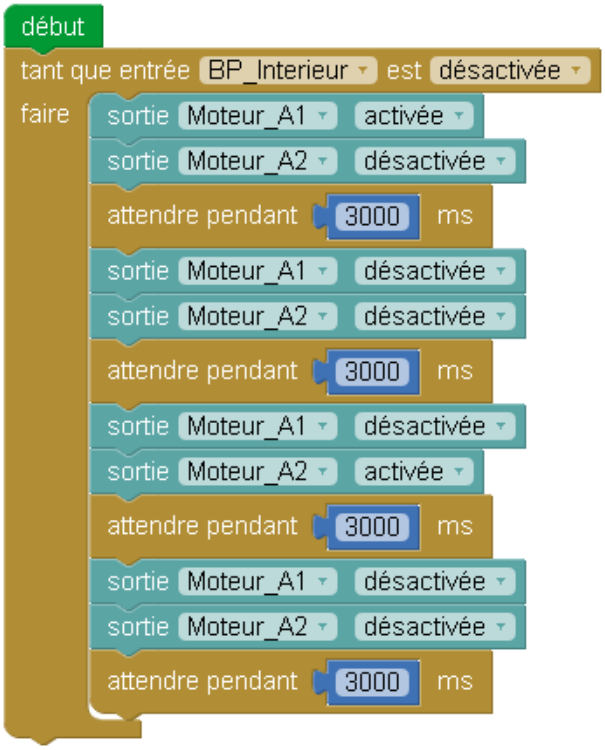
Objectif : ouvrir et fermer le portail en continu jusqu'à l'appui d'un bouton-poussoir.

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

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_B2_Organigramme.plf	Fichier Blockly : PC_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 - B.3 : Utilisation d'une boucle tant que

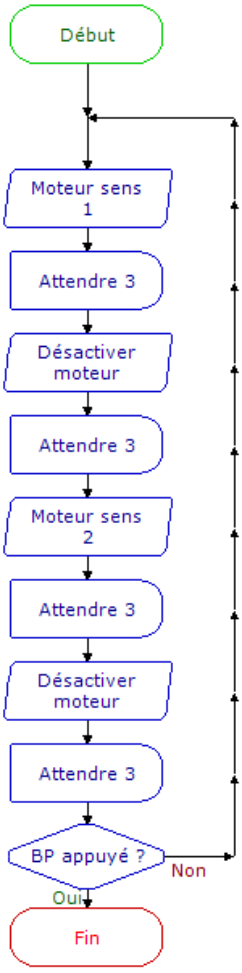
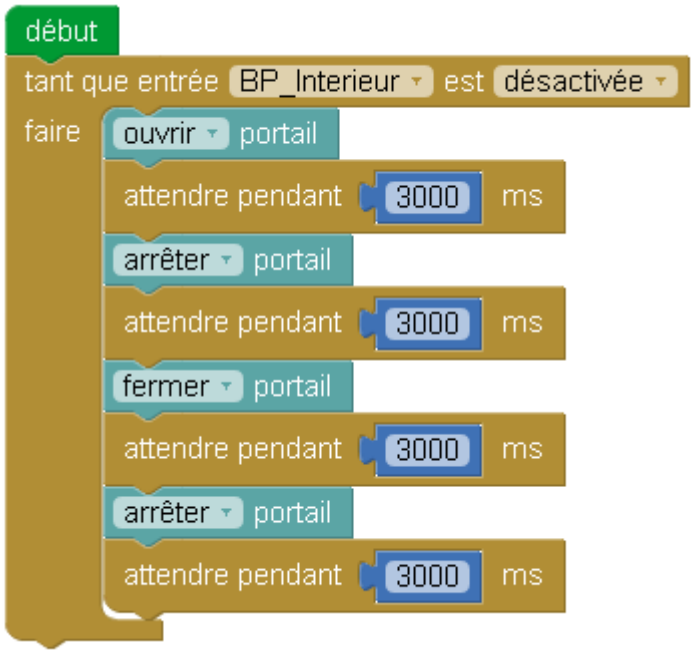
Fichier modèle : PC_N1_B.xml

Objectif : utiliser les blocs spéciaux du portail.

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

Instructions utilisées :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_B3_Organigramme.plf	Fichier Blockly : PC_N1_B3.xml

Remarque : les blocs de couleur jaune représente des commandes concernant l'utilisation des boucles et de gestion d'attente.

Niveau 1 - C

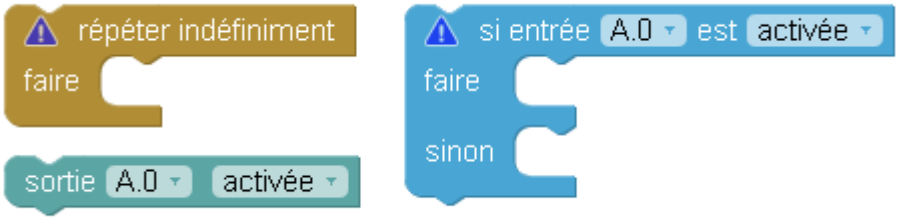
Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir

Fichier modèle : PC_N1_C.xml

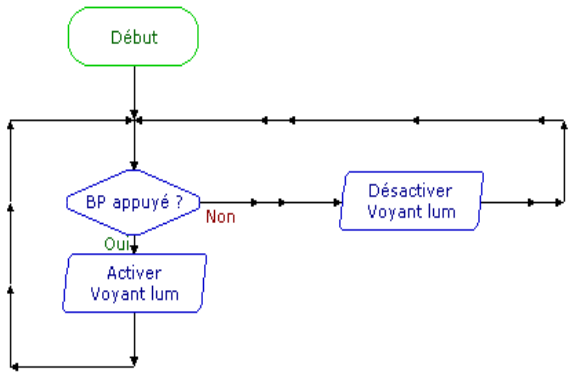
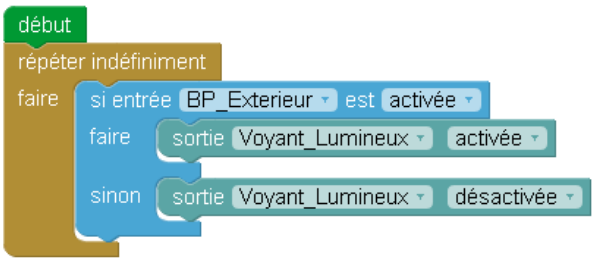
Objectif : allumer le voyant lumineux à l'appui du BP.

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

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_C1_Organigramme.plf	Fichier Blockly : PC_N1_C1.xml

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

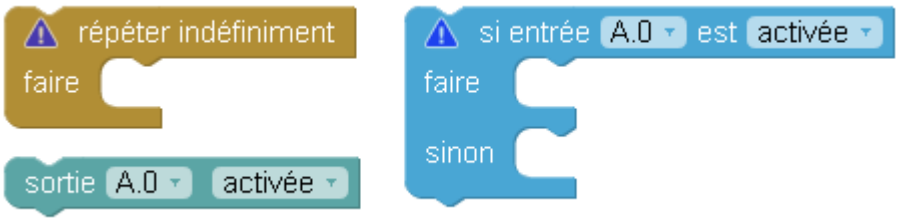
Exercice niveau 1 - C.2 : Instruction conditionnelle et barrière infrarouge

Fichier modèle : PC_N1_C.xml

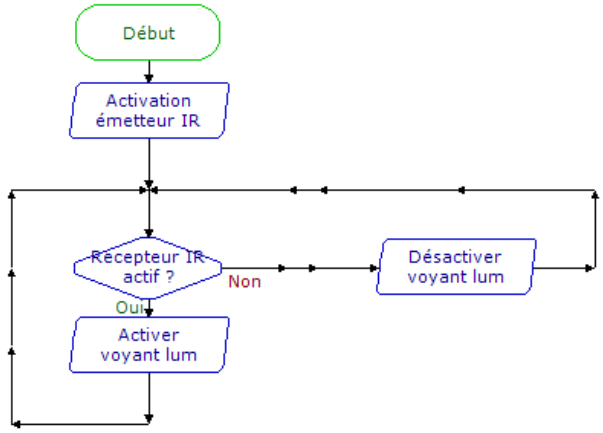
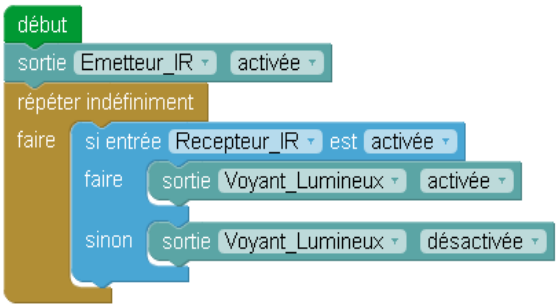
Objectif : activer le voyant lumineux lorsque la barrière infrarouge est franchie.

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

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_C2_Organigramme.plf	Fichier Blockly : PC_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.
Lorsqu'un obstacle franchit la barrière IR, le signal n'est plus transmis et l'entrée du récepteur IR devient active.

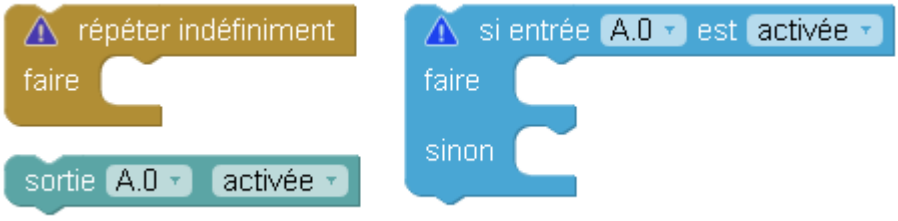
Exercice niveau 1 - C.3 : Contrôle moteur ET voyant lumineux

Fichier modèle : PC_N1_C.xml

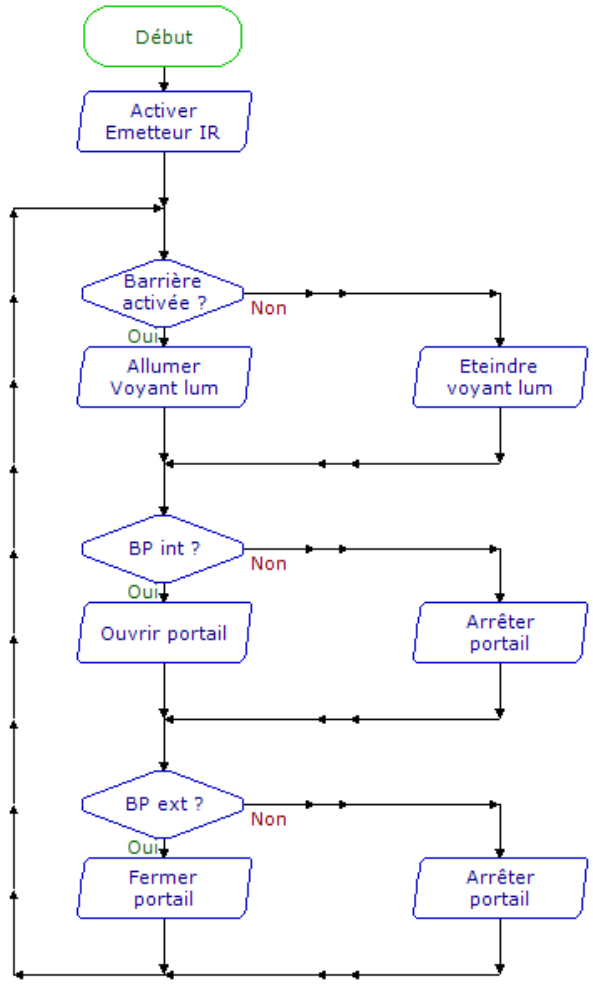
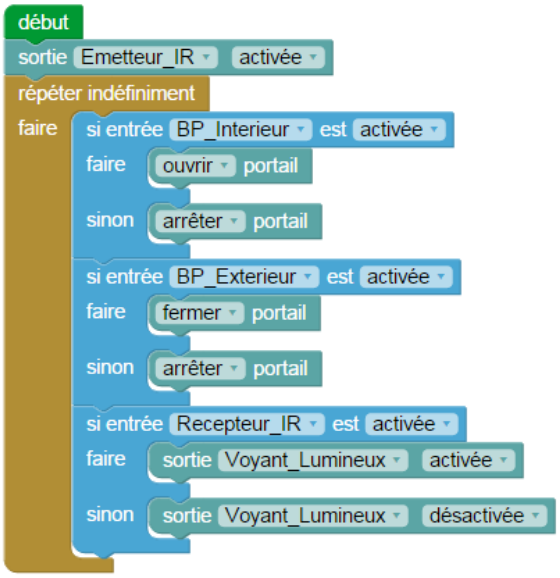
Objectif : contrôler le moteur avec les boutons-poussoirs et allumer le voyant sur le franchissement de la barrière infrarouge.

Notion abordée : utilisation des commandes conditionnelles.

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_C3_Organigramme.plf	Fichier Blockly : PC_N1_C3.xml

Niveau 1 - D

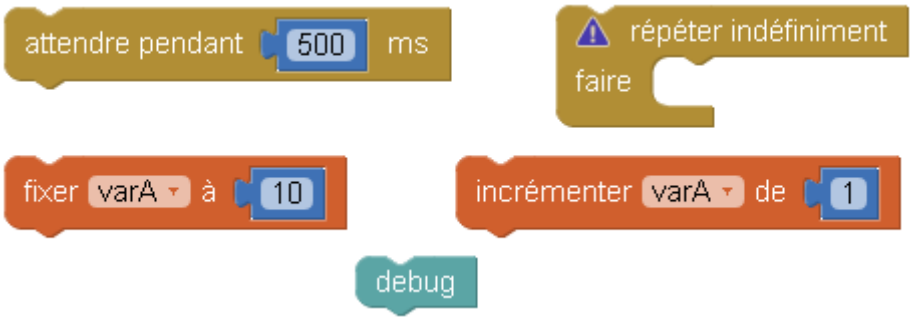
Exercice niveau 1 - D.1 : Utilisation des variables

Fichier modèle : PC_N1_D.xml

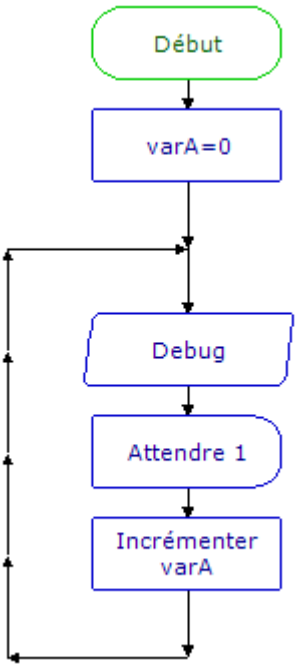
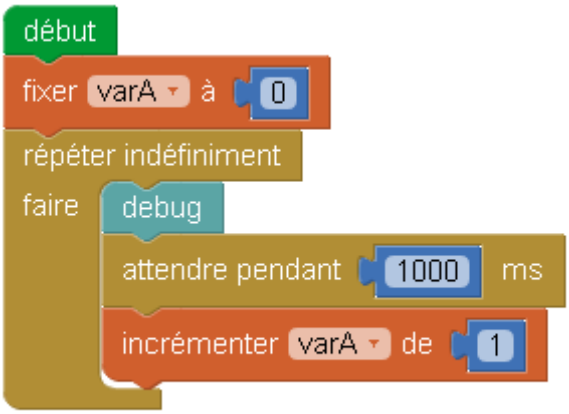
Objectif : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).

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

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_D1_Organigramme.plf	Fichier Blockly : PC_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.

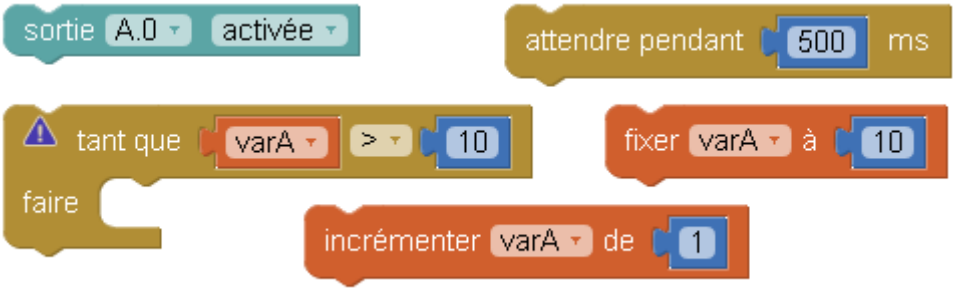
Exercice niveau 1 - D.2 : Utiliser et tester une variable

Fichier modèle : PC_N1_D.xml

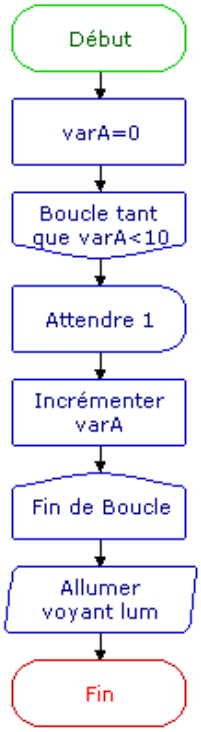
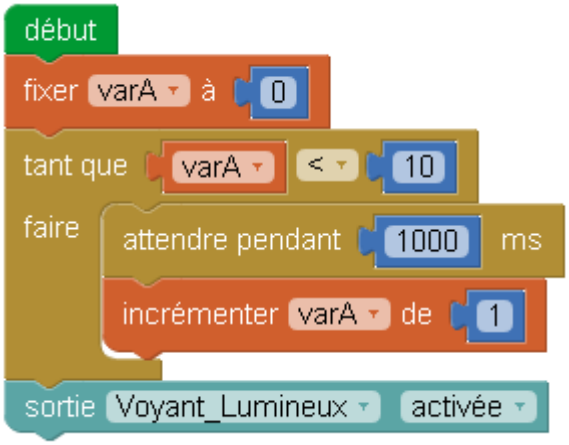
Objectif : incrémenter une variable au cours du temps. Lorsque la variable est supérieure à 10, activer le voyant.

Notion abordée : boucle tant que dépendant d'une variable

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_D2_Organigramme.plf	Fichier Blockly : PC_N1_D2.xml

Remarque : cet exercice peut être utilisé comme un minuteur.

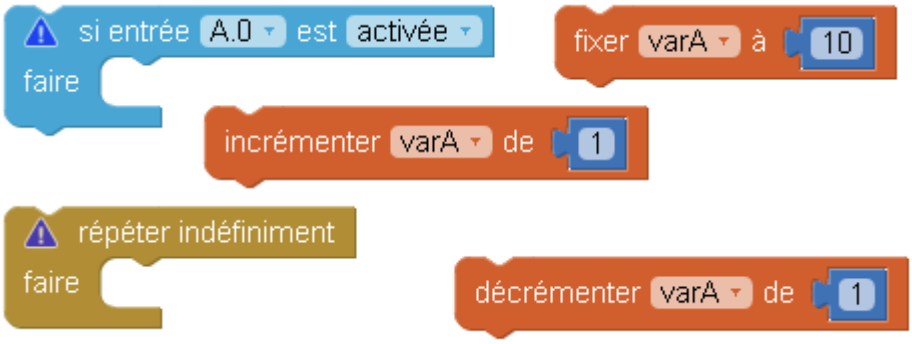
Exercice niveau 1 - D.3 : Contrôler la valeur d'une variable à l'aide des boutons-poussoirs

Fichier modèle : PC_N1_D.xml

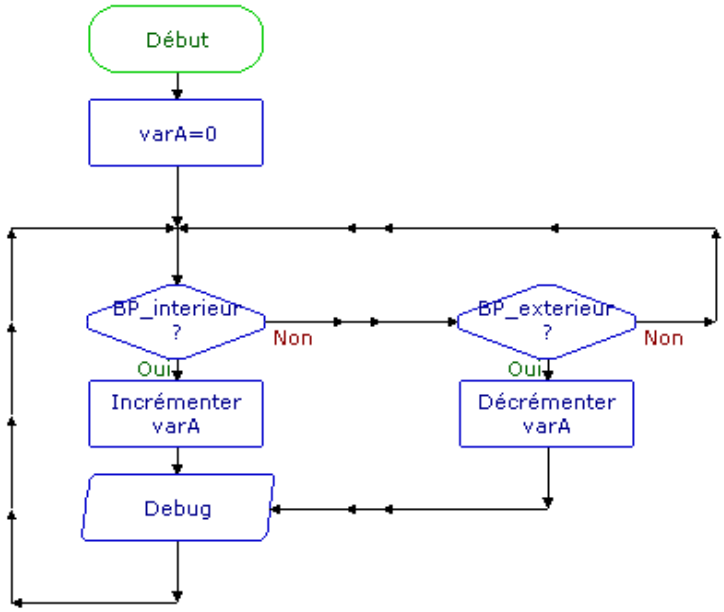
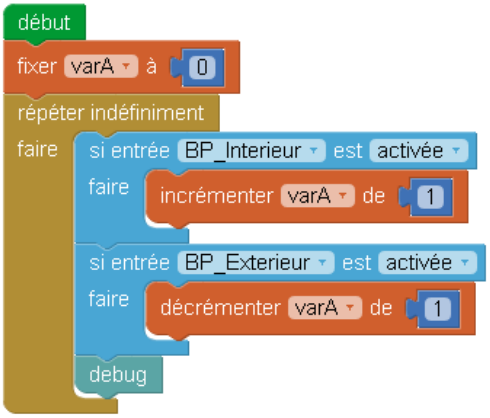
Objectif : incrémenter une variable à l'appui d'un bouton poussoir, décrémenter la même variable à l'appui de l'autre bouton poussoir.

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

Instruction utilisée :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_D3_Organigramme.plf	Fichier Blockly : PC_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.

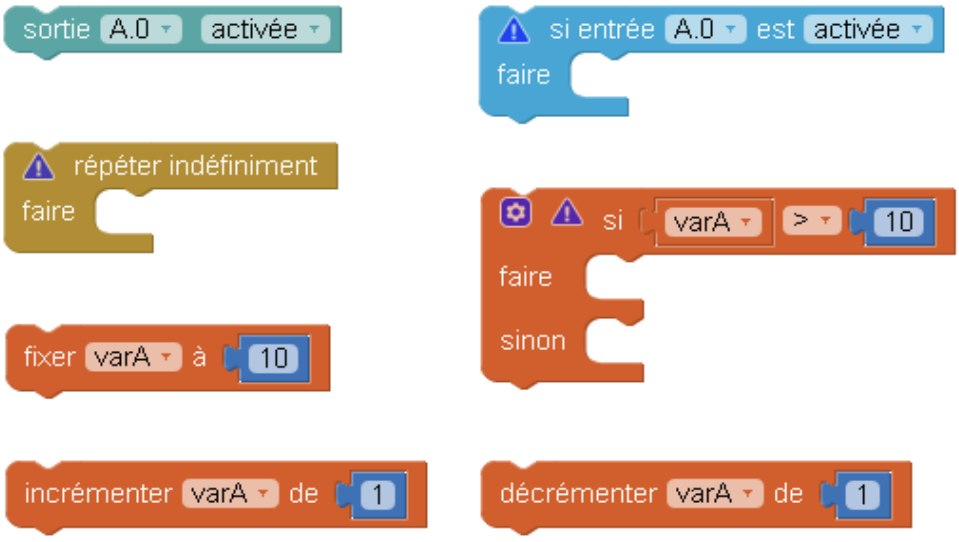
Exercice niveau 1 - D.4 : Tests /variables/ modules IR

Fichier modèle : PC_N1_D.xml

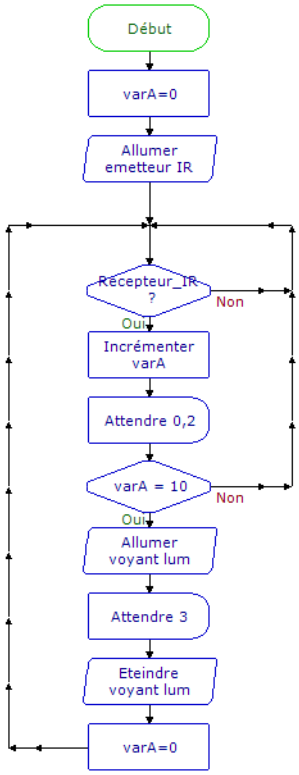
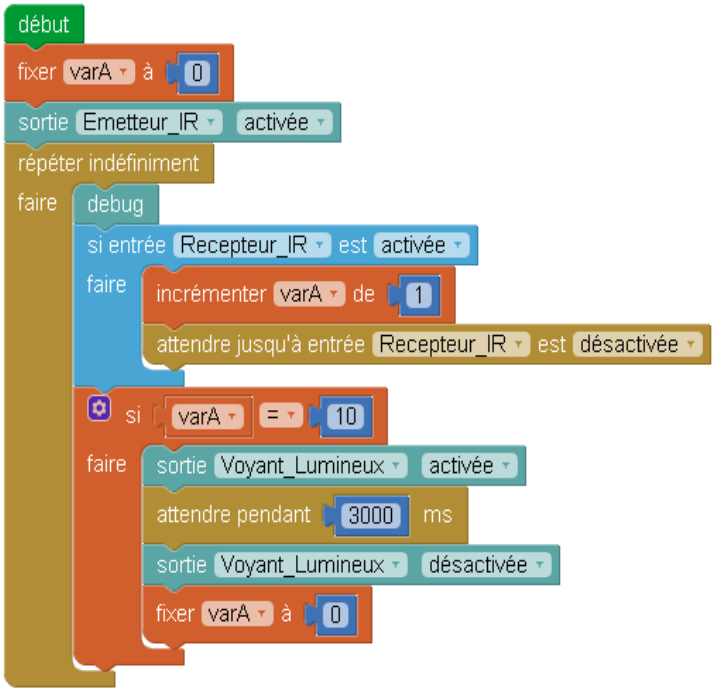
Objectif : incrémenter une variable à chaque passage sur la barrière IR. Lorsque le compteur arrive à 10, activer le voyant lumineux 3 secondes et remettre la variable à zéro

Notion abordée : test dépendant d'une variable

Instructions utilisées :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : PC_N1_D4_Organigramme.plf	Fichier Blockly : PC_N1_D4.xml

Programmation version de base niveau 2

Objectifs :

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

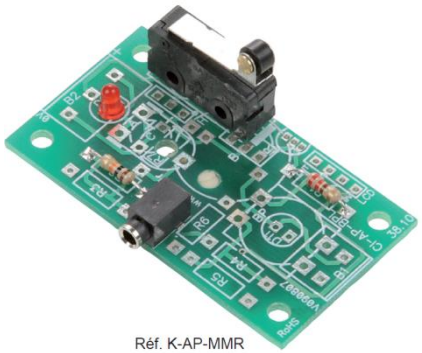
Ce niveau permet de mettre en œuvre le portail coulissant, au fur et à mesures des exercices vous allez utiliser de plus en plus de modules et enrichir votre code pour obtenir à la fin du niveau un portail qui marche parfaitement et qui respecte une logique de fonctionnement calquée sur le réel.

Nom du fichier	Description	Objectif
Niveau 2 A		
PC_N2_A1	Ouvrir et fermer le portail avec 2 secondes d'attente entre chaque mouvement. Utiliser les capteurs fins de course pour contrôler l'ouverture et la fermeture.	Fonctionnalité matérielle abordée : – Utilisation des FDC
PC_N2_A2	Ouverture du portail à l'appui sur BP_Exterieur. Fermeture du portail à l'appui sur BP_Interieur	
PC_N2_A3	Ouvrir et fermer le portail à l'aide des BP sans distinction, faire en sorte que le voyant lumineux clignote lors d'une manœuvre de la barrière.	
PC_N2_A4	Ouvrir et fermer le portail à l'aide des BP sans distinction, le voyant lumineux doit clignoter lors d'une manœuvre de la barrière. Inclure une gestion de sécurité lors la fermeture du portail.	

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.

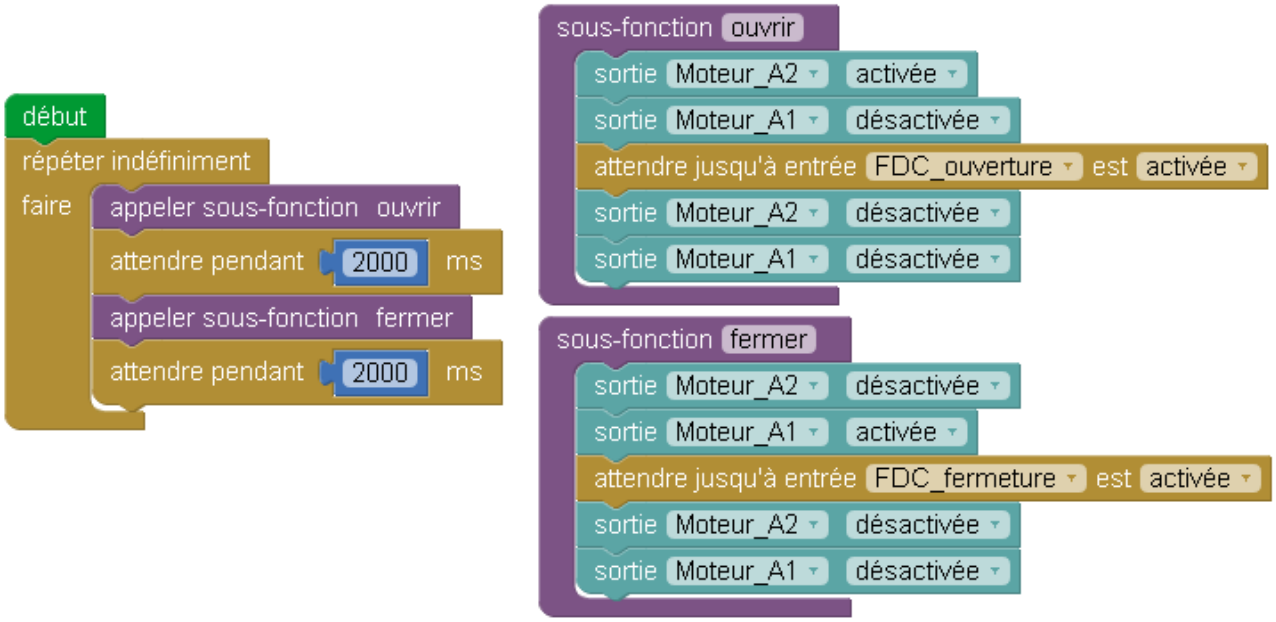
Notions abordées : utilisation des fins de course, procédures (sous-fonctions)



Réf. K-AP-MMR

Correction :

Blocs



Fichier Blockly : PC_N2_A1.xml

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

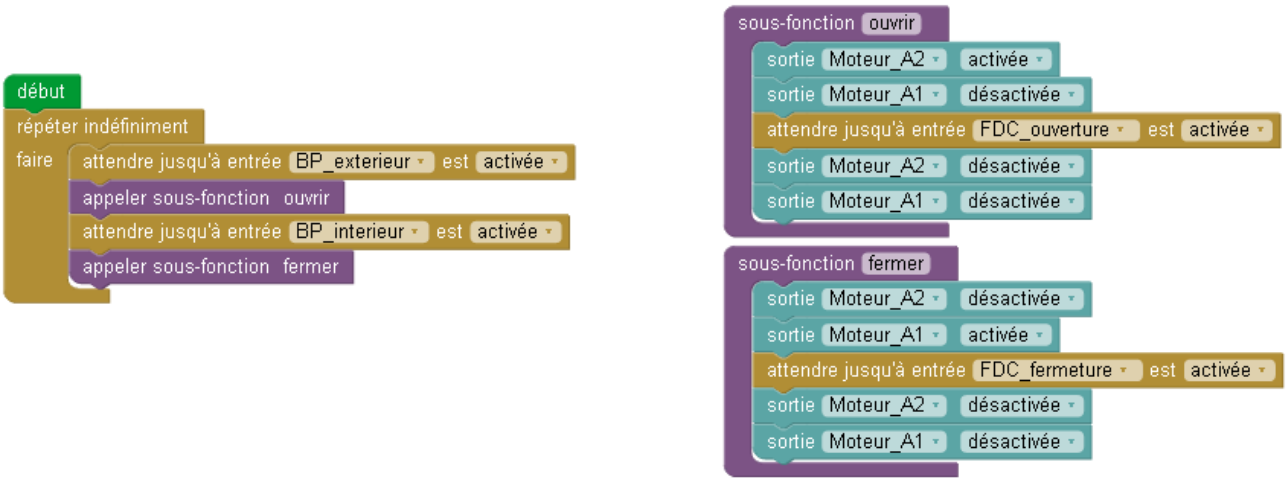
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture

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

Notions abordées :

Correction :

Blocs



Fichier Blockly : PC_N2_A2.xml

Remarque :

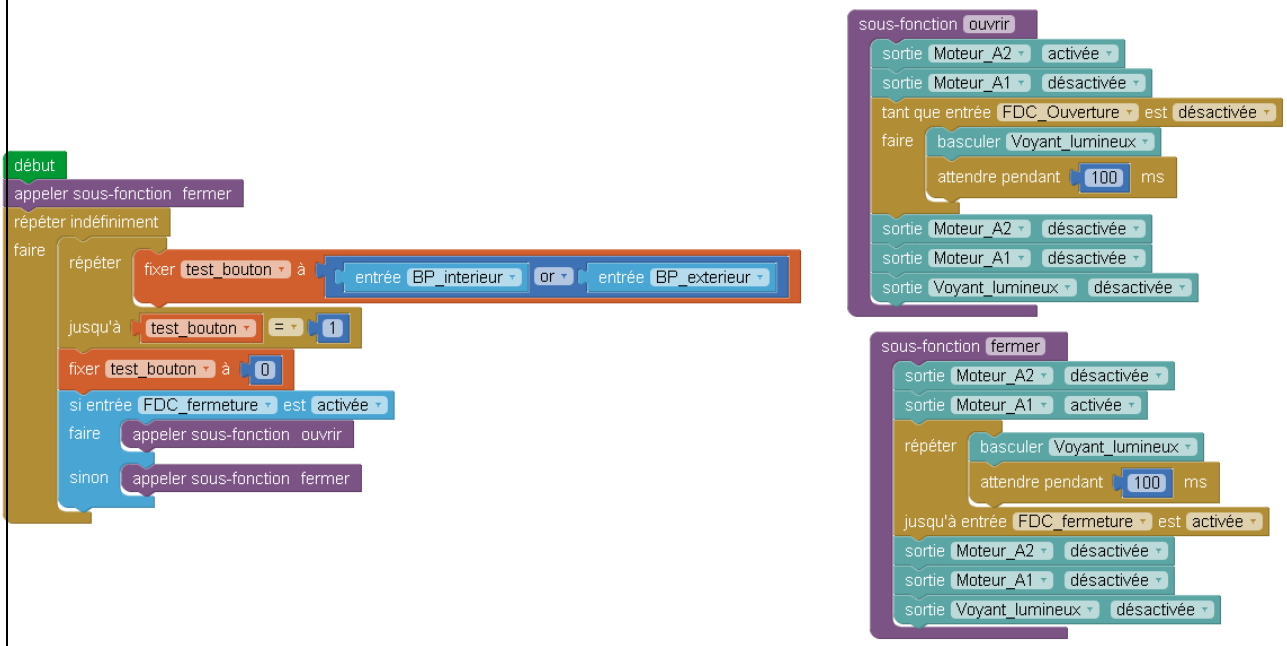
Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité

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

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

Correction :

Blocs



Fichier Blockly : PC_N2_A3.xml

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

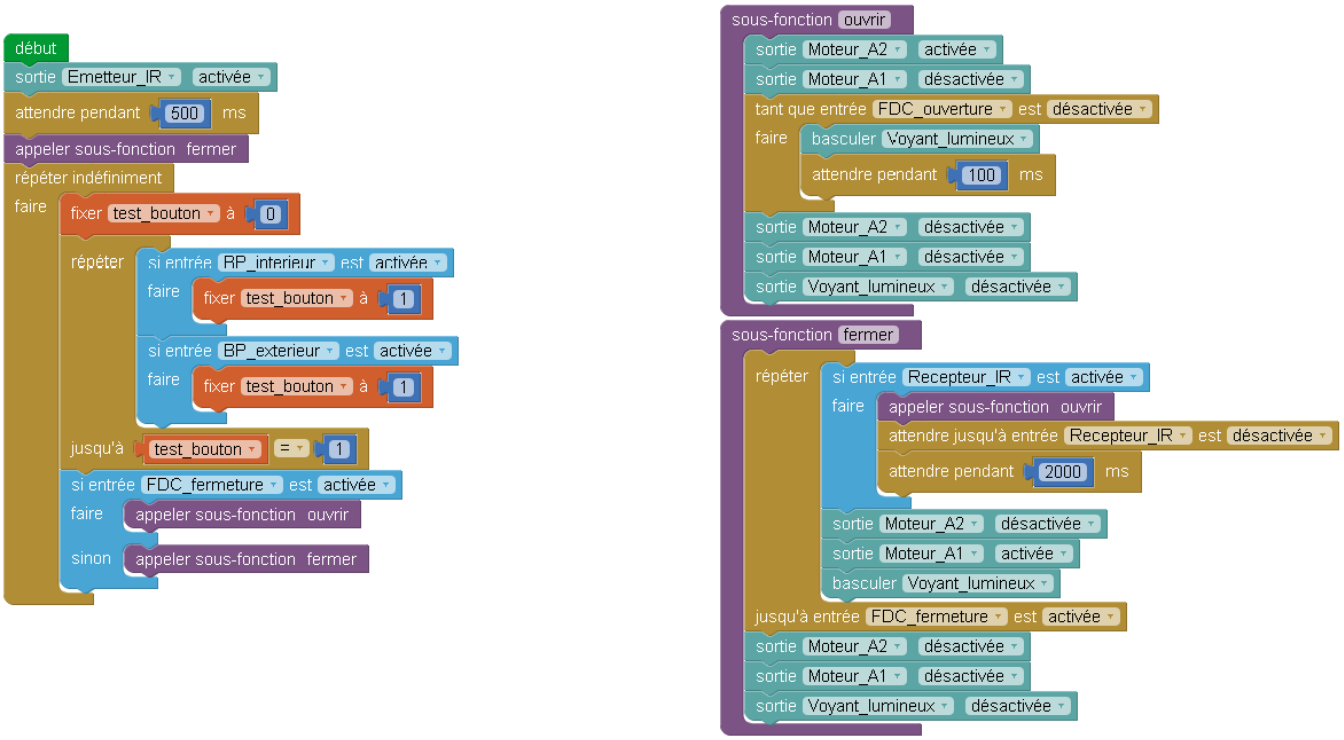
Exercice niveau 2 - A.4 : Contrôle d'ouverture/fermeture avec BP, signal de sécurité

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 lors la fermeture du portail.

Notions abordées : utilisation d'une procédure de sécurité

Correction :

Blocs



Fichier Blockly : PC_N2_A4.xml

Programmation version de base niveau 3 (options)

Objectif :

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

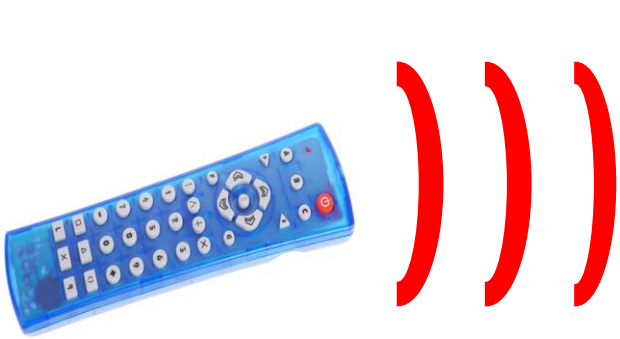
Le niveau 3 n'intègre pas de nouvelles notions de programmation mais de nouveaux blocs permettant d'utiliser les modules options.

Nom du fichier	Description	Objectif
Niveau 3A – Télécommande infrarouge		
PC_N3_A1	Ouvrir et fermer le portail à l'aide de la télécommande IR.	Fonctionnalité matérielle abordée : Utilisation de la télécommande IR Notions de programmation abordées : – Utilisation d'un bloc dédié à la communication IR
PC_N3_A2	Ouvrir le portail dès la réception d'un code spécifique envoyée par la télécommande. Le fermer 2 secondes plus tard.	
PC_N3_A3	Reprendre le programme **PC_N2_A4** et intégrer l'option ouverture par la télécommande.	
Niveau 3B – Module Bluetooth		
PC_N3_B1	Contrôler l'ouverture et la fermeture du portail à l'aide de 2 boutons présent sur l'application Android.	Fonctionnalité matérielle abordée : – module Bluetooth Notions de programmation abordées : – liaison série (hserin/hserout)
PC_N3_B2	Ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.	
PC_N3_B3	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.	
PC_N3_B4	Gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.	
Niveau 3C – Capteur de courant		
PC_N3_C1	Allumer le moteur et lire l'entrée capteur de courant.	Fonctionnalité matérielle abordée : – capteur de courant Notions de programmation abordées : – lire une entrée analogique
PC_N3_C2	Arrêter le moteur lors de la détection d'un blocage.	
PC_N3_C3	Gestion complète du portail avec sécurité par capteur de courant.	
Niveau 3D – Détection PIR		
PC_N3_D1	Allumer le voyant lumineux lorsqu'une présence est détectée par le capteur PIR.	Fonctionnalité matérielle abordée : – Capteur_PIR
PC_N3_D2	Gestion complète du portail avec ouverture par détection PIR	

Option : Module 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

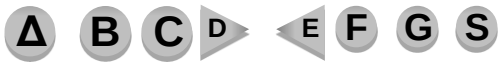
1. Insérer 2 piles AAA dans le logement au dos de la télécommande.

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 :

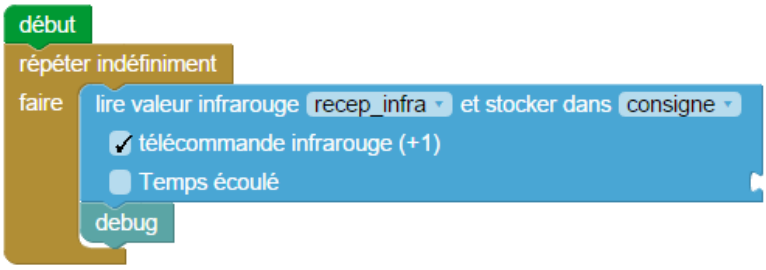


Conseil : Si la télécommande ne fonctionne plus, appuyer sur  pour revenir à la configuration compatible PICAXE

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.



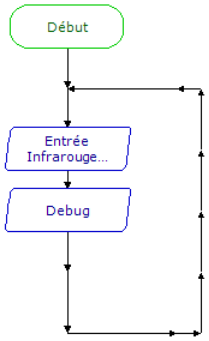


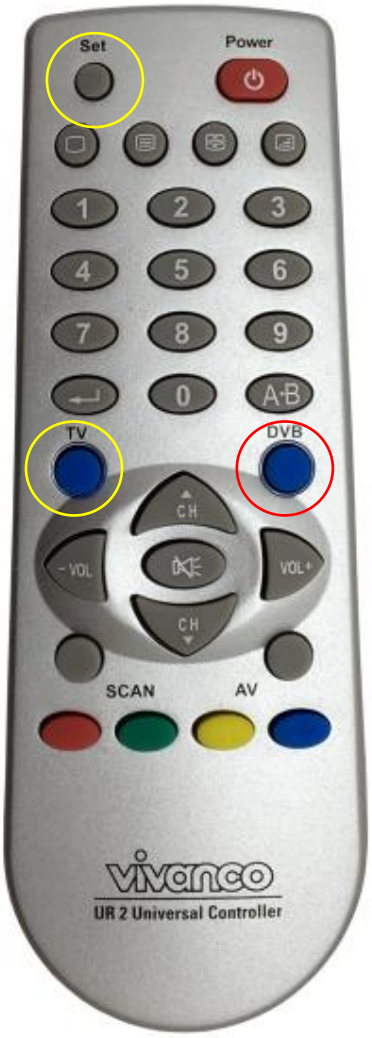
Tableau de correspondance des touches

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



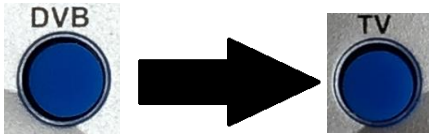
1. Insérer 2 piles AAA dans le logement au dos de la télécommande.

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.

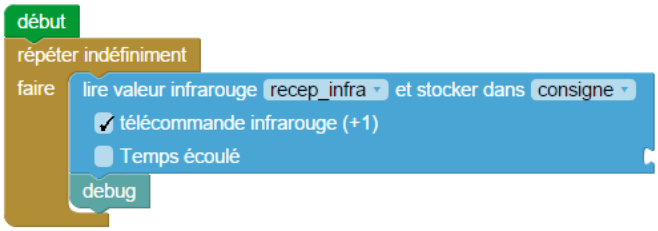
ATTENTION !
La touche **DVB** risque de changer le mode. Appuyer sur **TV** pour revenir dans le bon mode.



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.



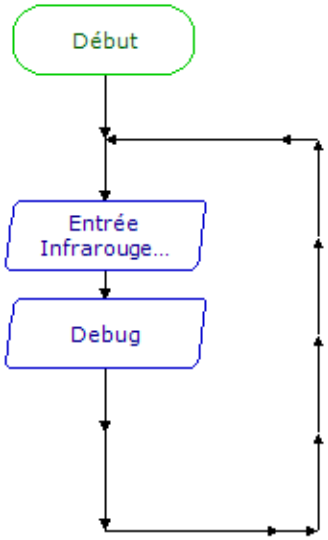
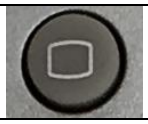
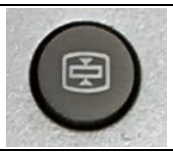
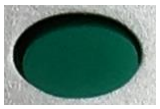
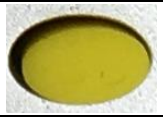


Tableau de correspondance des touches

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**		

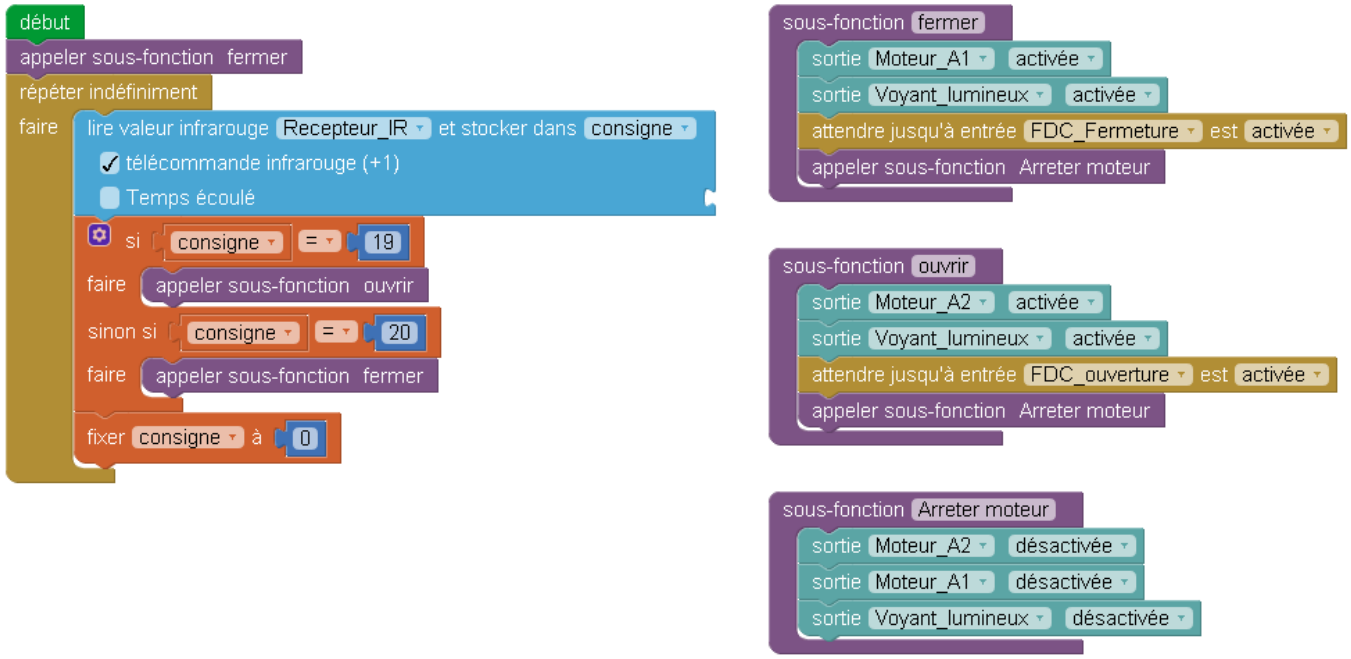
Exercice niveau 3 - A.1 : Contrôle d'ouverture/fermeture avec la télécommande IR

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

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

Correction :

Blocs



Fichier Blockly : PC_N3_A1.xml

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

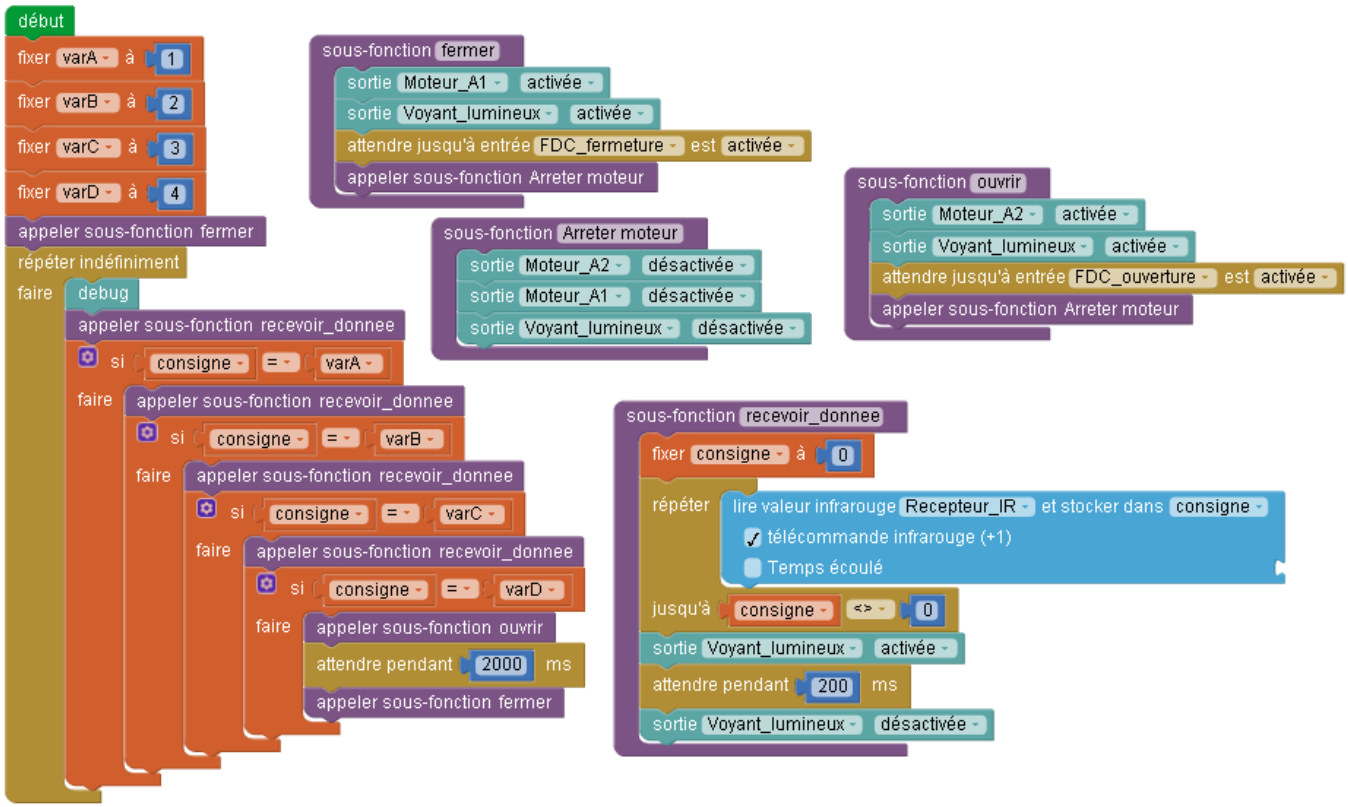
Exercice niveau 3 - A.2 : Ouvrir le portail avec un code

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 infrarouge : télécommande/AutoProg.

Correction :

Blocs



Fichier Blockly : PC_N3_A2.xml

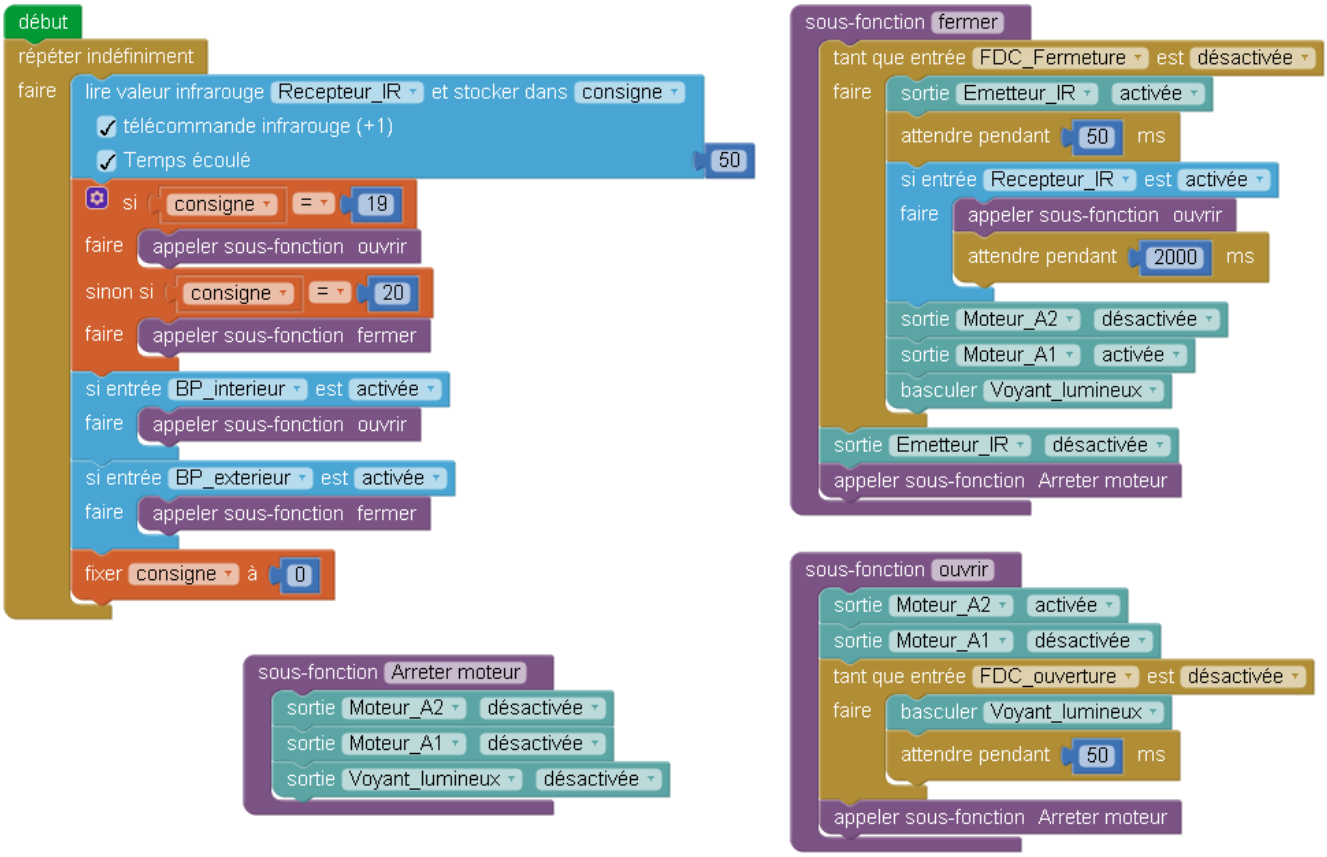
Exercice niveau 3 - A.3 : Télécommande IR + BP + Barrière IR

Objectif : reprendre le programme **PC_N2_A4** et intégrer l'option ouverture par la télécommande.

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

Correction :

Blocs



Fichier Blockly : PC_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 appuyée par la télécommande pendant un temps de 50 ms.

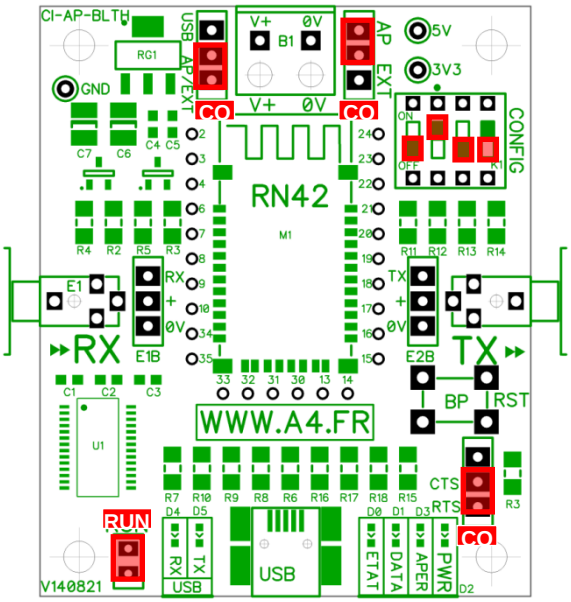
Option : Module Bluetooth

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.



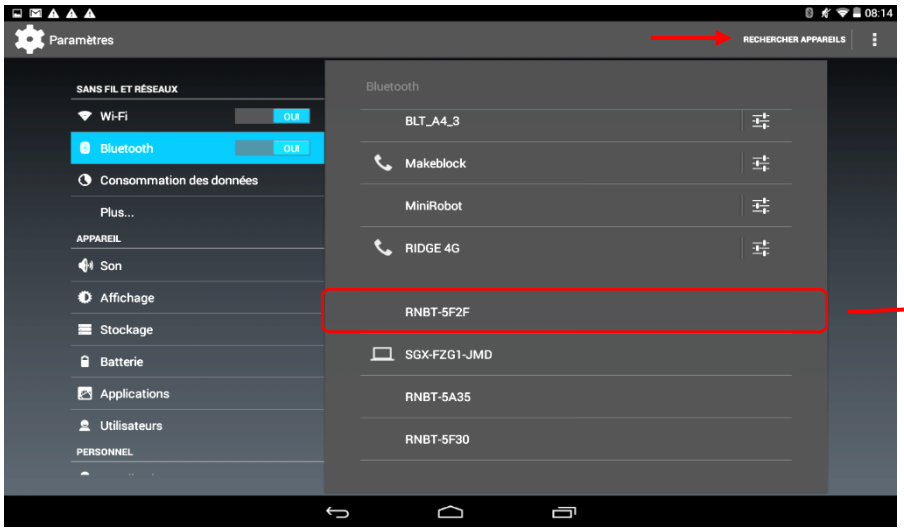
- Le cavalier repéré **RUN** est utilisé lors de la mise au point de programmes avec **Arduino**. Il doit être ôté pour permettre le téléversement du programme puis doit être remis lors de l'utilisation.
- La mise au point de programmes avec **PICAXE** ne nécessite pas d'ôter ce cavalier pour transférer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

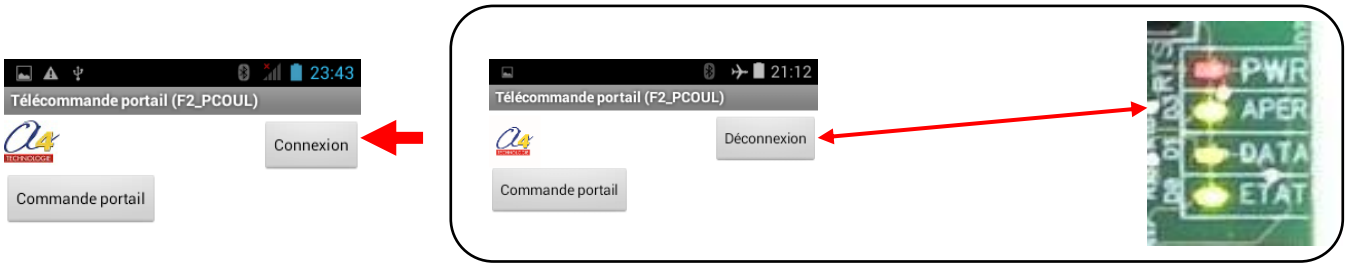
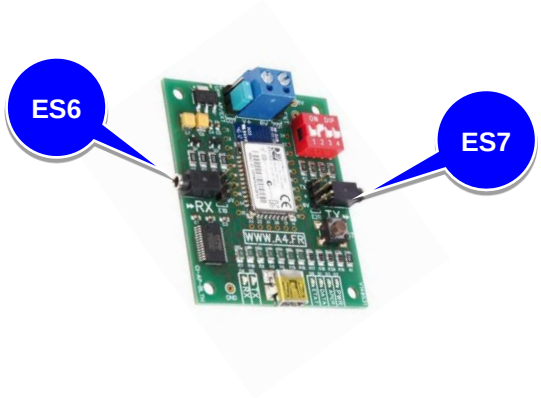


Tableau d'affectation des entrées et sorties du portail coulissant avec option Bluetooth

ES	Modules de communication pour entrées / sorties numériques	Broche	Etiquette Blockly
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	C.5	Recepteur_IR
4	Capteur détection de présence (option)	*C.4*	Detection_PIR*
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**		
3	*(libre)*	*A.3*	
2	*(libre)*	A.2	
1	*(libre)*	A.1	
0	Capteur de courant analogique/numérique (option)	A.0	Capteur_courant
SN	**Modules actionneurs sorties numériques**		
7	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.7	Moteur_A2
6	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.6	Moteur_A1
5	*(libre)*	B.5	
4	*(libre)*	B.4	
3	*(libre)*	B.3	
2	*(libre)*	B.2	
1	Emetteur barrière infrarouge	B.1	Emetteur_IR
0	Module signal LED jaune	B.0	voyant_Lumineux

Câblage du module Bluetooth (K-AP-MBLTH)

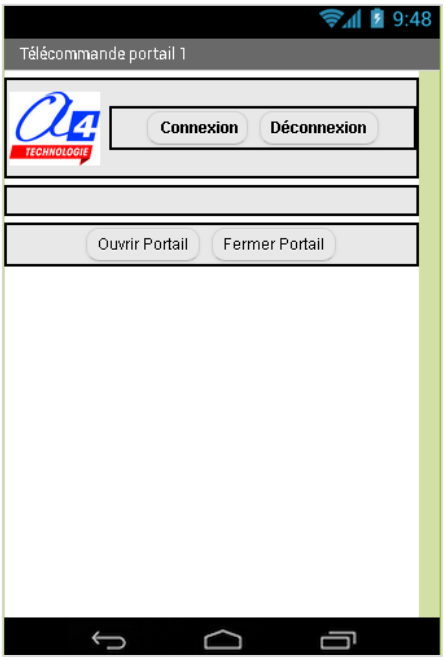


Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth

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

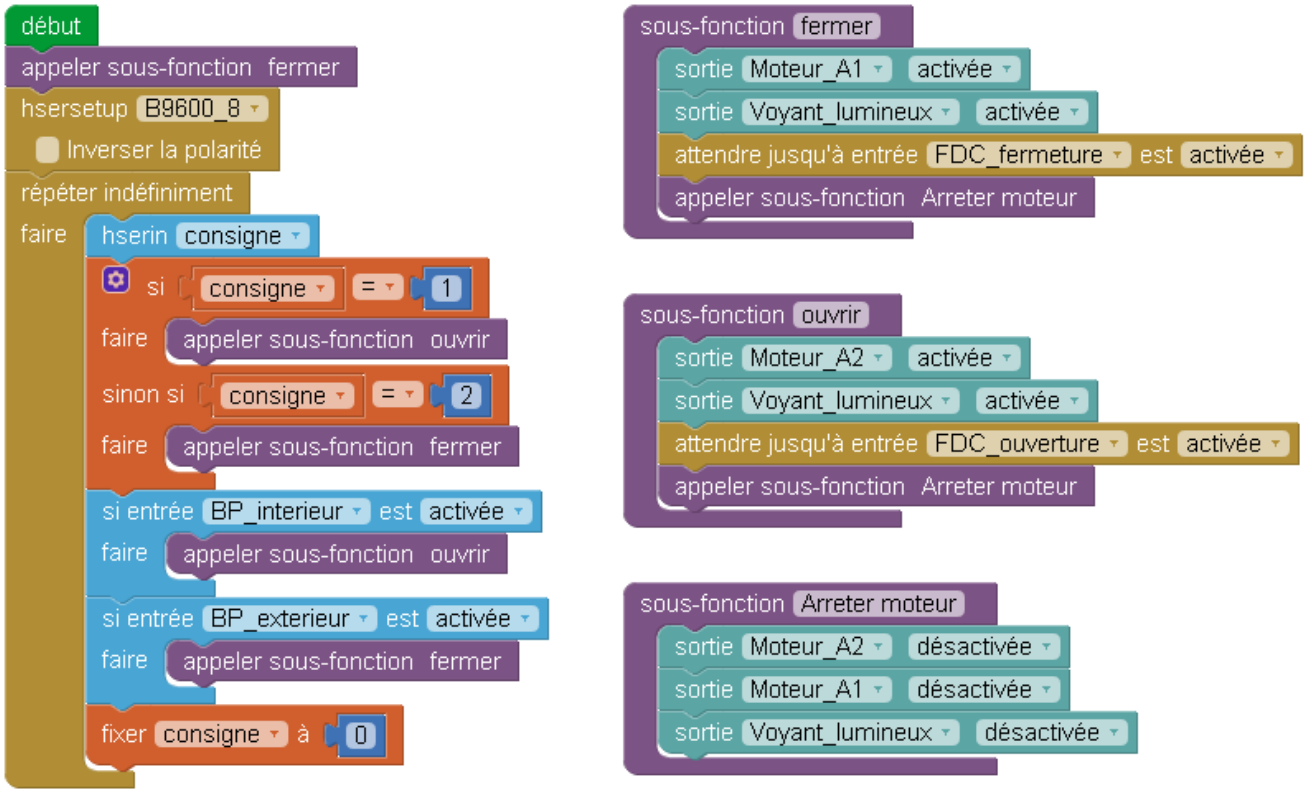
Notion abordée : réception de données Bluetooth envoyées par un Smartphone.

Application Android : Portail_1.apk / **Fichier App Inventor** : Portail_1.aia



Correction :

Blocs



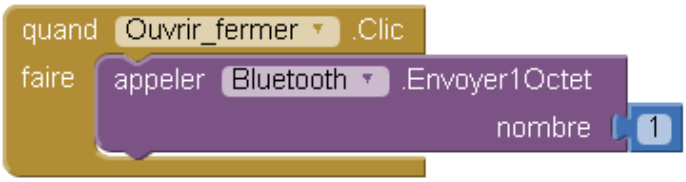
Fichier Blockly : PC_N3_B1.xml

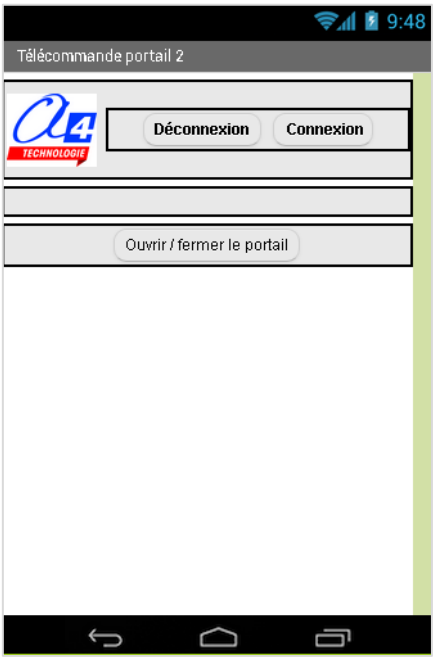
Exercice niveau 3 - B.2 : Contrôle du portail par Smartphone

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.

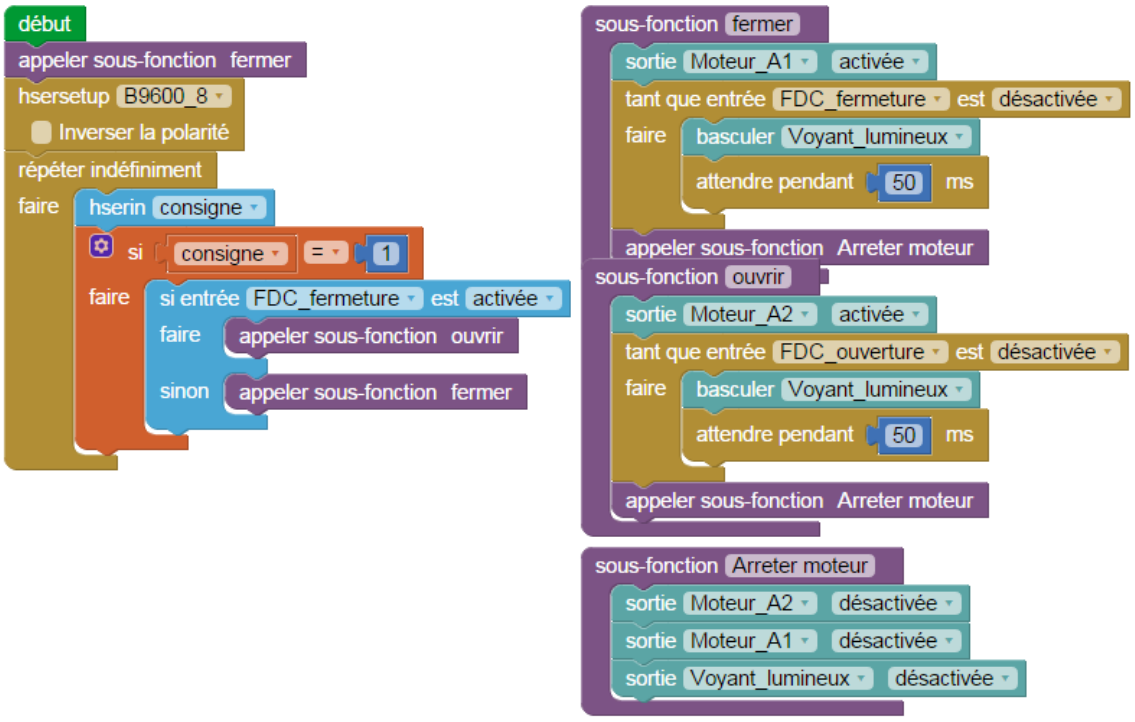
Application Android : Portail_2.apk / **Fichier App Inventor** : Portail_2.aia





Correction :

Blocs



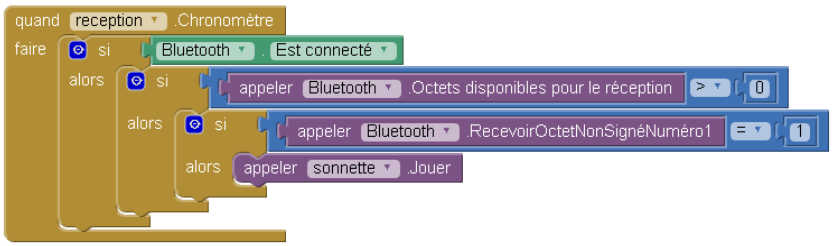
Fichier Blockly : PC_N3_B2.xml

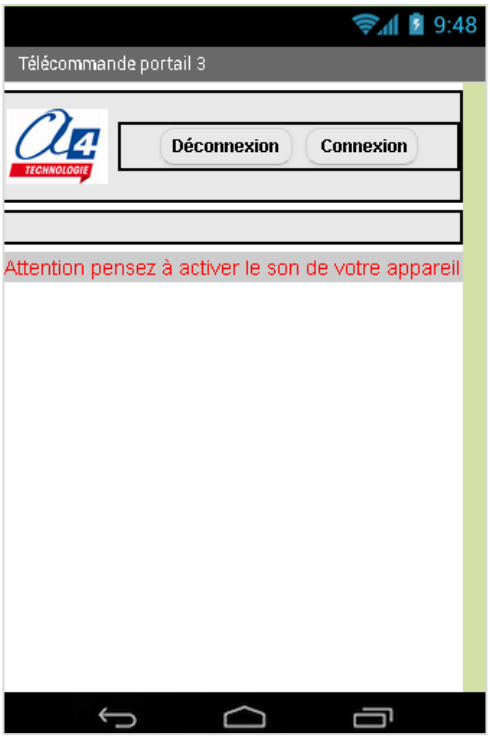
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone

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

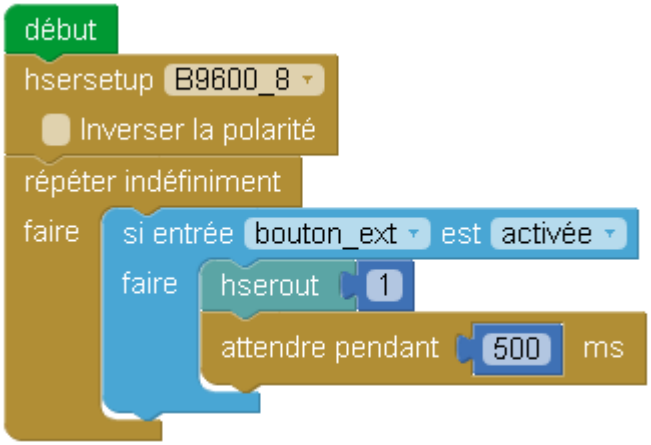
Notion abordée : envoyer des informations à un Smartphone par Bluetooth.

Application Android : Portail_3.apk / **Fichier App Inventor** : Portail_3.aia





Correction :

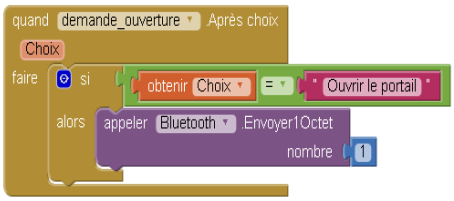
Blocs

Fichier Blockly : PC_N3_B3.xml

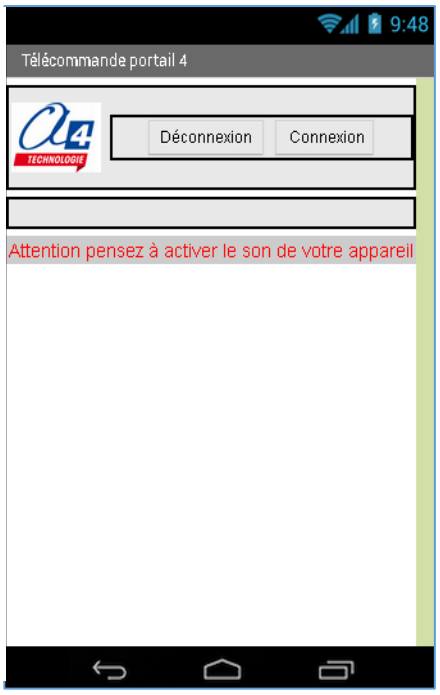
Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone

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.

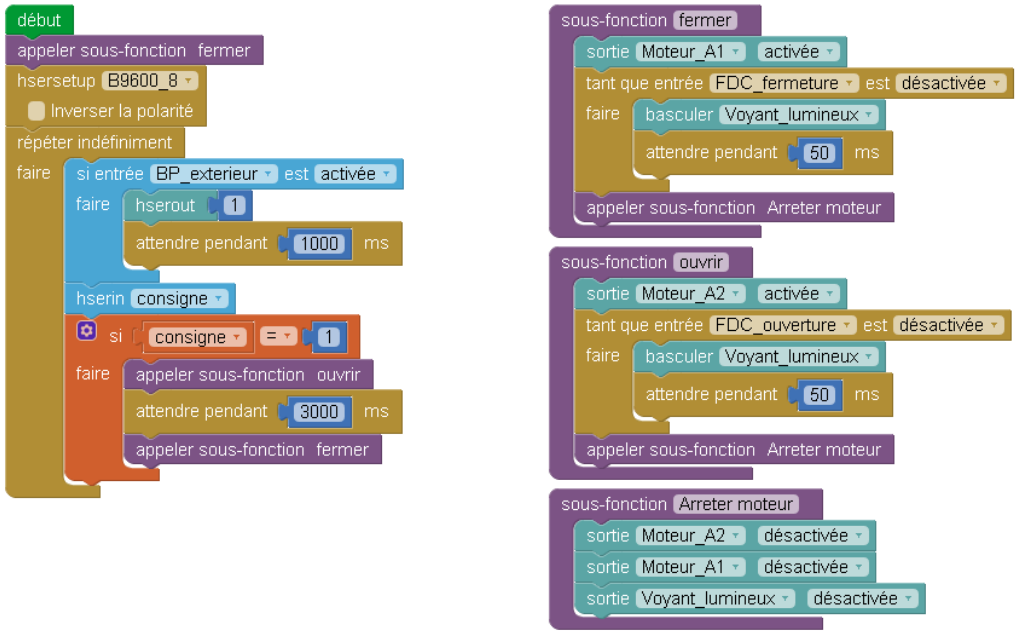
Application Android : Portail_4.apk / **App Inventor** : Portail_4.aia





Correction :

Blocs



Fichier Blockly : PC_N3_B4.xml

Option : Module capteur de courant

Ce module permet de sécuriser le fonctionnement d'un automatisme animé par un moteur à courant continu. En effet, lorsqu'un événement anormal se produit (ex : blocage du portail), la consommation de courant du moteur augmente. La détection de la surintensité au-delà d'un seuil permet de déclencher l'arrêt du moteur et ainsi mettre le système en sécurité.

Le module capteur de courant peut fonctionner selon deux modes.

Mode analogique :
Le cavalier S1 est en position ANA. La sortie MESURE renvoie une tension proportionnelle au courant circulant dans le module. Cette valeur est lue sur une entrée analogique du boîtier.

Mode numérique (tout ou rien) :
Le cavalier S1 est en position NUM. Le potentiomètre SEUIL permet de régler un seuil de courant au-delà duquel la sortie MESURE basculera. Si le courant circulant dans le module dépasse le seuil, la LED témoin et la sortie MESURE s'activent, sinon elles sont inactives.

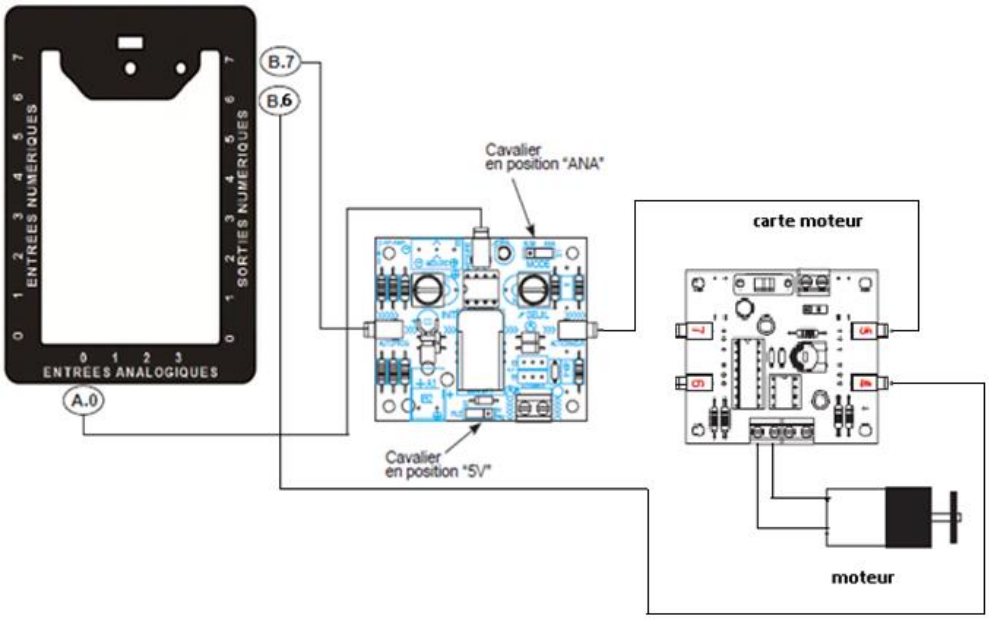
Le potentiomètre INIT permet d'adapter la plage de mesure du module en fonction du contexte d'utilisation (consommation plus ou moins élevé de courant selon matériel utilisé).
La mesure de courant se fait au travers de de la connectique jack (AUTOPROG / ACTIONNEUR) ou bien directement via le bornier à vis ACTIONNEUR.

Mesure via connectique jack :
Le courant mesuré correspond au courant total qui circule dans le montage (courant consommé par l'interface AutoProg, les modules et le moteur).

Mesure via bornier à vis :
Le courant mesuré correspond à celui circulant dans l'élément connecté au bornier (moteur).

Dans la suite des exercices, le capteur de courant est utilisé en mode Analogique et la mesure est faite via la connectique jack. Il s'agit de détecter les surintensités dans le moteur ; on considère que la consommation de l'interface et des modules est négligeable par rapport à celle du moteur. Ce mode de d'utilisation facilite le câblage.
L'interface AutoProg est alimenté par un bloc d'alimentation externe afin de garantir une tension constante de 5V dans l'ensemble du montage.

Mise en service du module



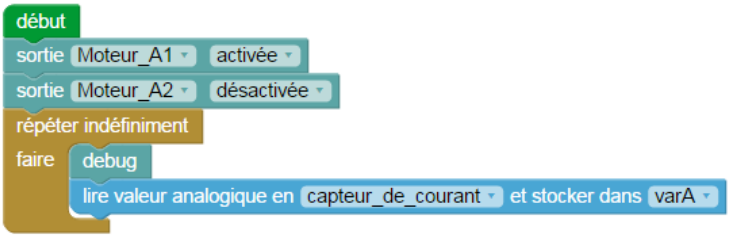
Réglage du potentiomètre d'initialisation

Il s'agit de régler la tension de sortie du module de telle sorte qu'elle soit maximale lorsque le moteur et bloquée (consommation maximum) afin de travailler sur plage de mesure adaptée au moteur du portail.
En fonctionnement à vide (débrayé du portail) le moteur consomme jusqu'à 250 mA à vitesse moyenne. Si le moteur est bloqué, sa consommation monte au-delà de 500 mA.

Le réglage s'effectue en débrayant le moteur de la barrière (dérailler la barrière pour libérer le moteur de toute contrainte mécanique).

1) Positionner l'interrupteur du module moteur sur OFF puis mettre le cavalier du moteur de Vext sur Vint pour que le moteur soit alimenté directement par l'interface.
   Positionner le cavalier S1 du module Capteur de courant sur ANA et le cavalier S2 sur la position 5V (Schéma)

Relier les embases jack A1 et A2 du module moteur respectivement sur les sorties B.6 et B.7 de l'interface AutoProg. Relier l'embase jack MESURE du module capteur de courant l'entrée analogique A.0 de l'interface AutoProg puis charger le programme de test **Test_Capteur_Courant.xml**
Ce programme permet de visualiser en direct à l'écran une valeur (varA) qui est proportionnelle à celle du courant consommé par le moteur.



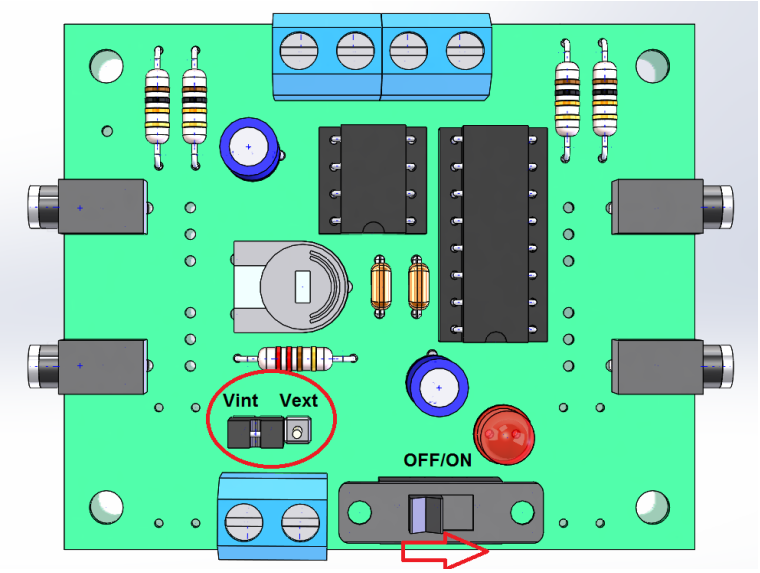
2) Positionner l'interrupteur du module moteur sur ON : le moteur doit tourner
   Ajuster le potentiomètre F-MOTA du module moteur à une position intermédiaire pour fixer la vitesse de rotation du moteur.

3) Ajuster le potentiomètre MESURE du module capteur de courant jusqu'à avoir une valeur de varA comprise entre 0 et 5.

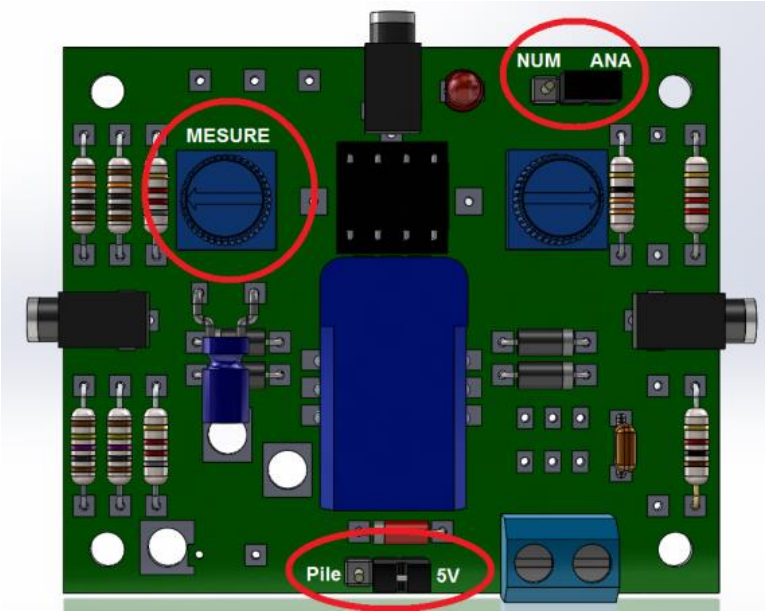
Symbole	Décimal
varA	2

4) Bloquer le moteur manuellement en retenant la roue dentée d'entrainement du portail. La valeur de varA doit augmenter de façon significative (augmentation brutale du courant consommé par le moteur).

   Cette procédure permet de vérifier la bonne configuration du montage et de visualiser la valeur du courant circulant dans le moteur en vue de déterminer le seuil au-delà duquel on considère que celui-ci à consommation excessive (blocage).



Module pilotage moteurs (REF : K-AP-MMOT-M)



Module Capteur de courant (REF : K-AP-MAMP-M)

Tableau d'affectation des entrées et sorties avec capteur de courant

ES	Modules de communication pour entrées / sorties numériques	Broche	Etiquette Blockly
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	C.5	Recepteur_IR
4	Capteur détection de présence (option)	*C.4*	Detection_PIR*
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**		
3	*(libre)*	A.3	
2	*(libre)*	A.2	
1	*(libre)*	A.1	
0	**Capteur de courant analogique/numérique**	**A.0**	**Capteur_courant**
SN	**Modules actionneurs sorties numériques**		
7	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.7	Moteur_A2
6	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.6	Moteur_A1
5	*(libre)*	B.5	
4	*(libre)*	B.4	
3	*(libre)*	B.3	
2	*(libre)*	B.2	
1	Emetteur barrière infrarouge	B.1	Emetteur_IR
0	Module signal LED jaune	B.0	voyant_Lumineux

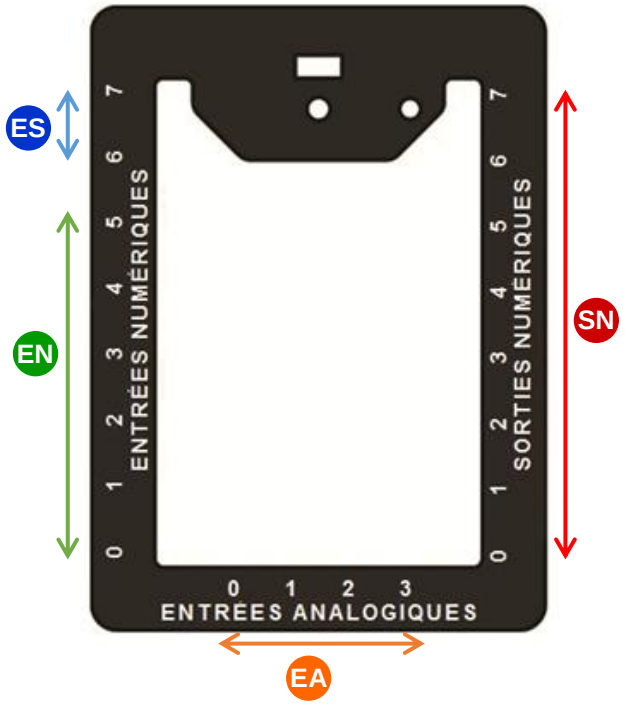
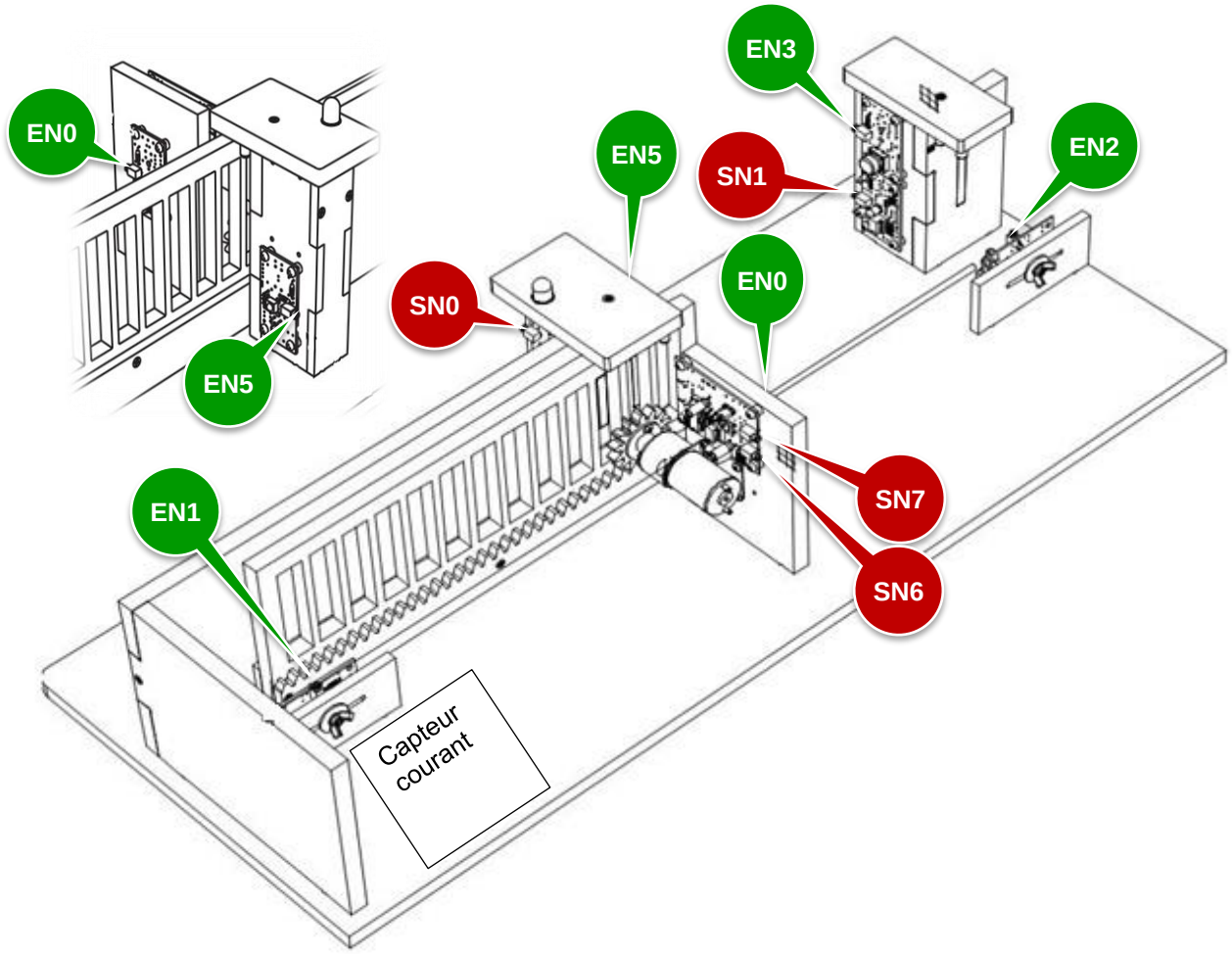
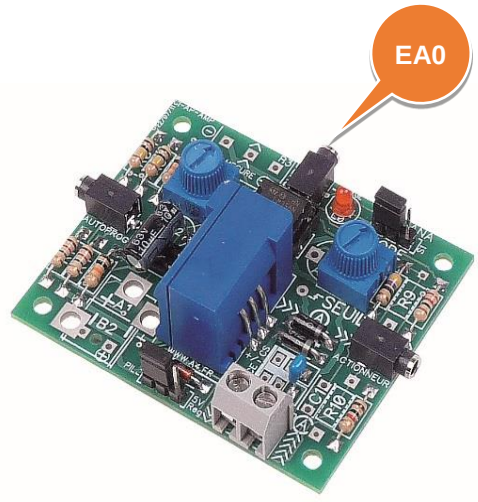


Schéma de câblage





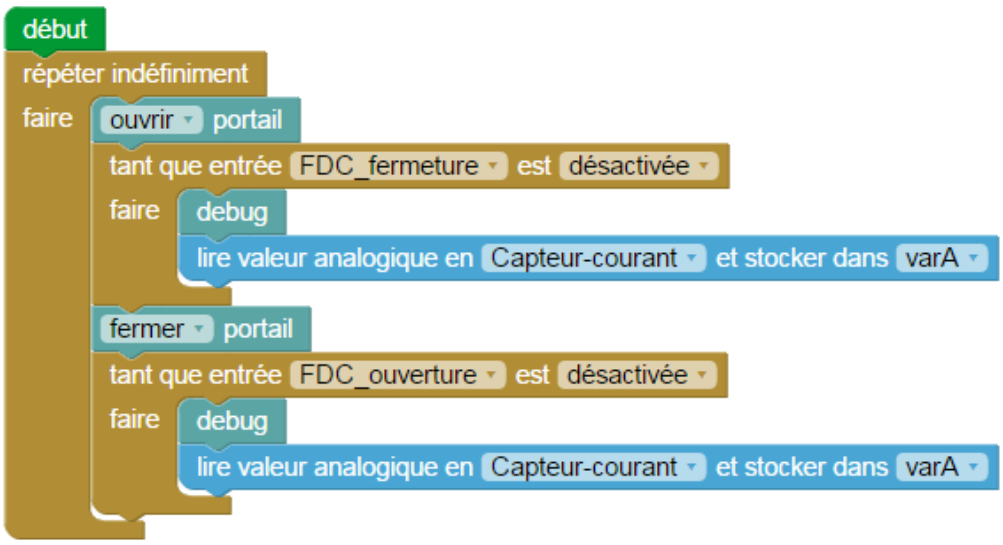
Exercice niveau 3 – C.1 : Utilisation du capteur de courant

Objectif : le portail avance et change de sens à chaque passage sur un capteur fin de course, lire la valeur du capteur de courant en continu et déterminer le seuil (Valeur maximale de varA lorsqu'on bloque la barrière).

Notion(s) abordée(s) : lire une entrée analogique.

Correction :

Blocs

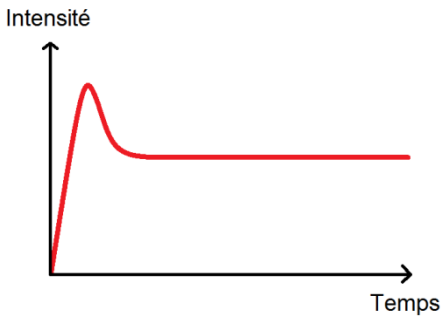


Fichier Blockly : PC_N3_C1.xml

Remarque : Lorsqu'un effort est apporté au moteur, sa demande de courant augmente. Plus la résistance opposante au moteur sera grande, plus il va consommer de courant.

Lorsque d'un démarrage du moteur, il utilise pendant un bref instant plus de courant qu'en fonctionnement normal. Visualiser ce pic d'intensité sur varA à chaque changement de sens et considérer cette valeur comme le seuil bas, cela car le moteur ne doit pas s'arrêter au démarrage à cause de cette surintensité.

Voici une représentation de la surintensité de démarrage du moteur :



Celui-ci après le démarrage va se remettre à la valeur d'intensité normale de fonctionnement.

Il est préférable pour l'exercice suivant de mettre un seuil plus petit que le seuil maximum
Par exemple si lorsque vous bloquez la barrière vous trouvez un seuil max avec varA = 80, nous vous conseillons pour la suite de fixer ce seuil à 70 par exemple.

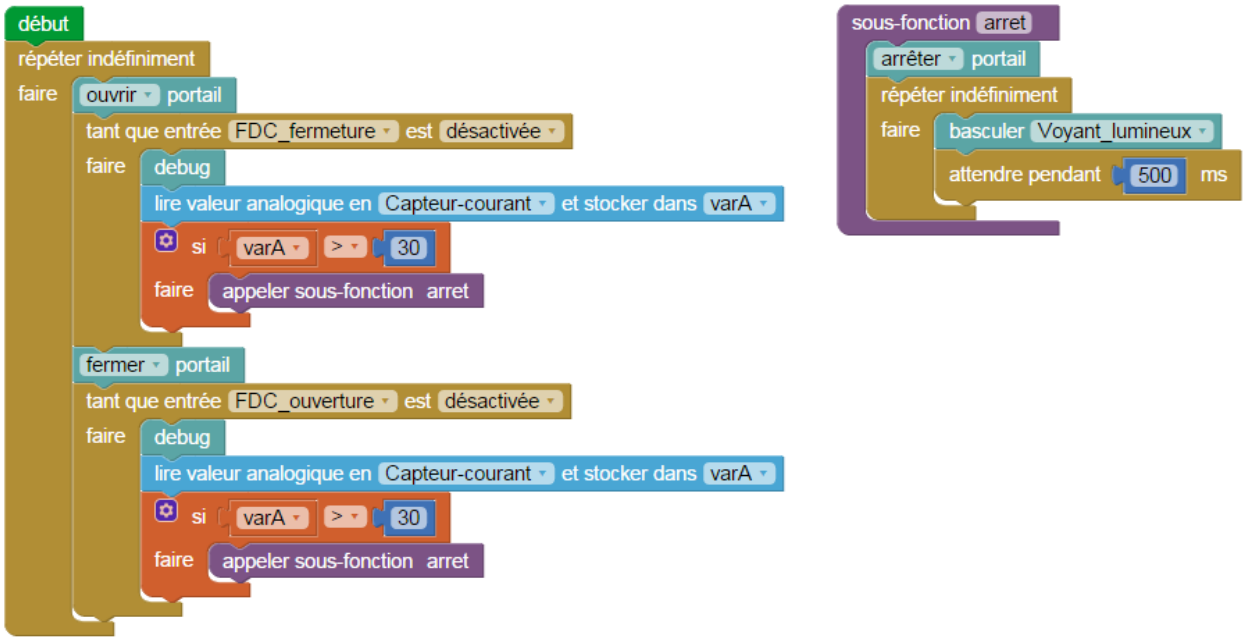
Exercice niveau 3 – C.2 : Capteur de courant et variable

Objectif : à partir de la valeur seuil établie dans l'exercice précédent, arrêter le moteur et faire clignoter le voyant lumineux lorsque le capteur de courant détecte un blocage (Donc lorsque varA atteint le seuil établi en fin d'exercice N3_C1).

Notion abordée : lire et interpréter une donnée d'une entrée analogique.

Correction :

Blocs

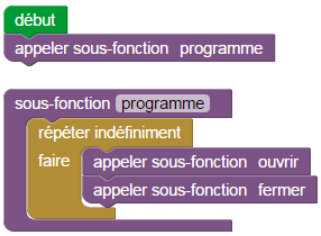


Fichier Blockly : PC_N3_C2.xml

Remarque : La valeur de seuil sélectionnée ici est 30, mais elle correspondra à celle que vous aurez établie dans votre programme.

Exercice niveau 3 – C.3 : Sécurité et réactivation

Objectif : Après un arrêt, relancer le programme (ouverture et fermeture) sur l'appui d'un bouton poussoir. Pour le programme, il est préférable d'utiliser une sous fonction « programme » qui comprendra la boucle de répétition infinie :

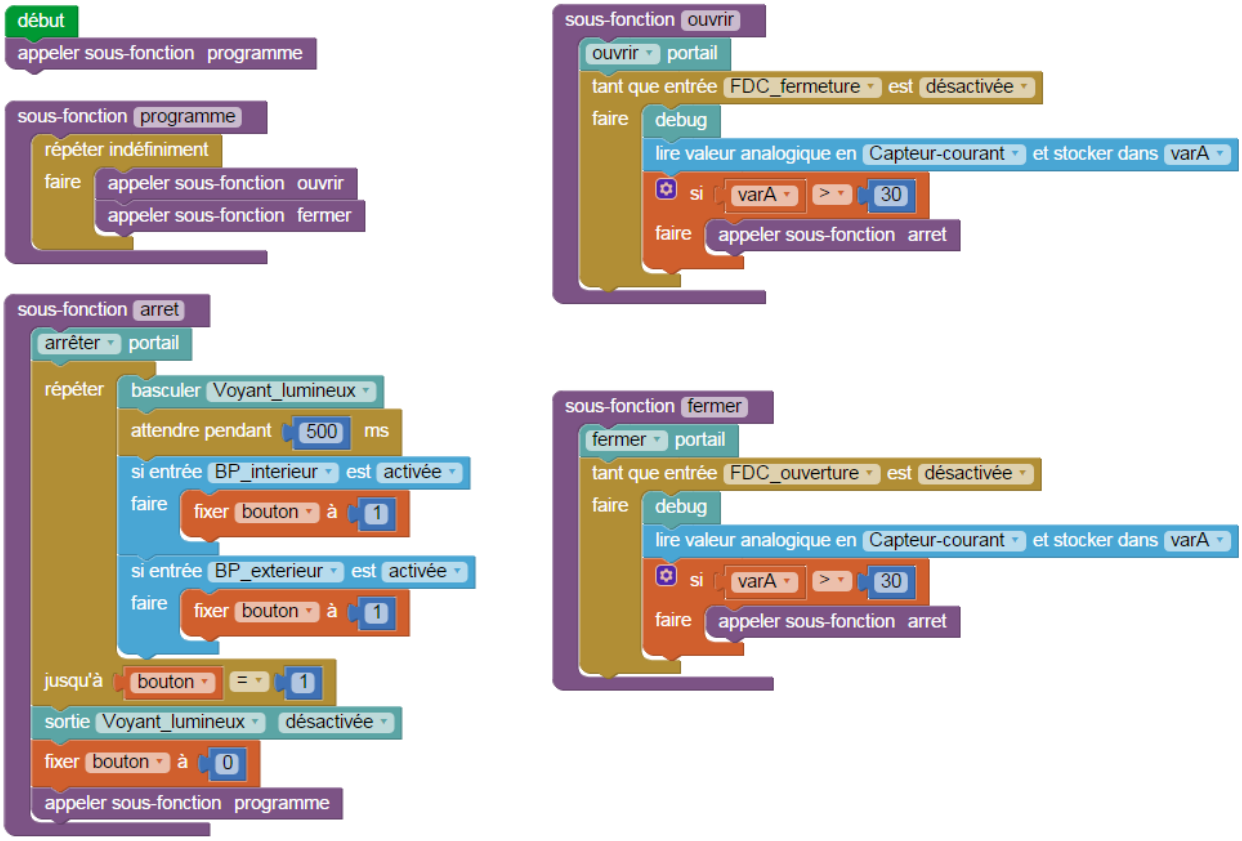


Cela afin de pouvoir rappeler le programme depuis la sous-fonction d'arrêt.

Notion abordée : lire et interpréter une donnée d'une entrée analogique.

Correction :

Blocs



Fichier Blockly : PC_N3_C3.xml

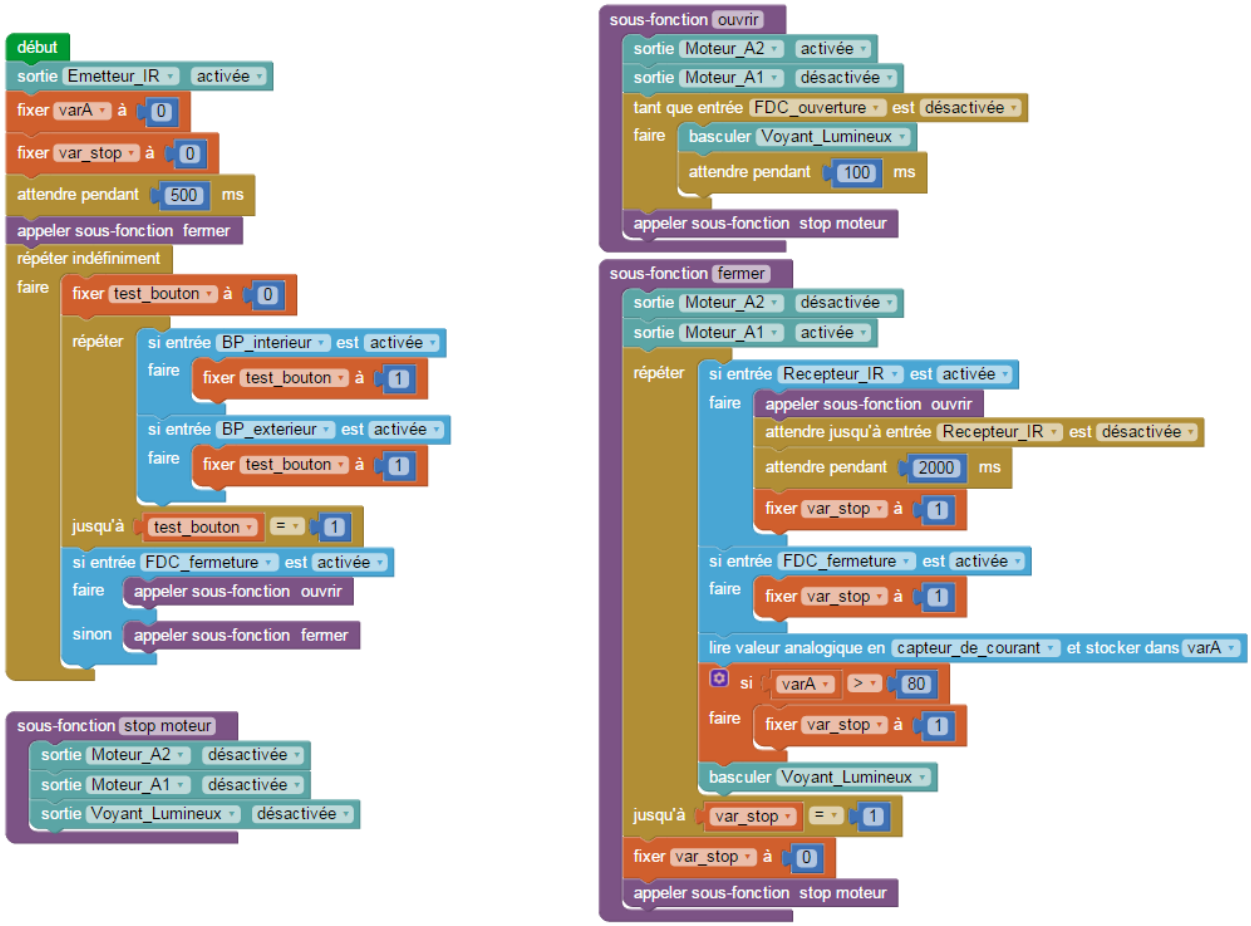
Exercice niveau 3 – C.4 : Sécurité du portail par contrôle du capteur de courant

Objectif : reprendre le programme **PC_N2_A4** et y ajouter le capteur de courant.

Notion abordée : gérer une séquence d'instruction complète.

Correction :

Blocs



Fichier Blockly : PC_N3_C4.xml

Option : Module capteur PIR

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

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

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

Tableau d'affectation des entrées et sorties avec capteur PIR

ES	Modules de communication pour entrées / sorties numériques	Broche	Etiquette Blockly
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	C.5	Recepteur_IR
4	**Capteur détection de présence**	**C.4**	**Detection_PIR***
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**		
3	*(libre)*	A.3	
2	*(libre)*	A.2	
1	*(libre)*	A.1	
0	*Capteur de courant analogique/numérique (option)*	*A.0*	*Capteur_courant*
SN	**Modules actionneurs sorties numériques**		
7	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.7	Moteur_A2
6	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.6	Moteur_A1
5	*(libre)*	B.5	
4	*(libre)*	B.4	
3	*(libre)*	B.3	
2	*(libre)*	B.2	
1	Emetteur barrière infrarouge	B.1	Emetteur_IR
0	Module signal LED jaune	B.0	voyant_Lumineux

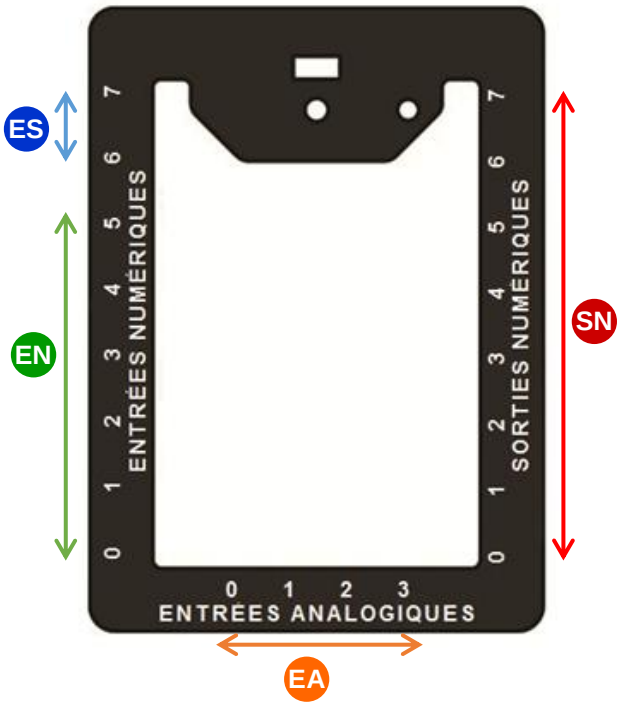
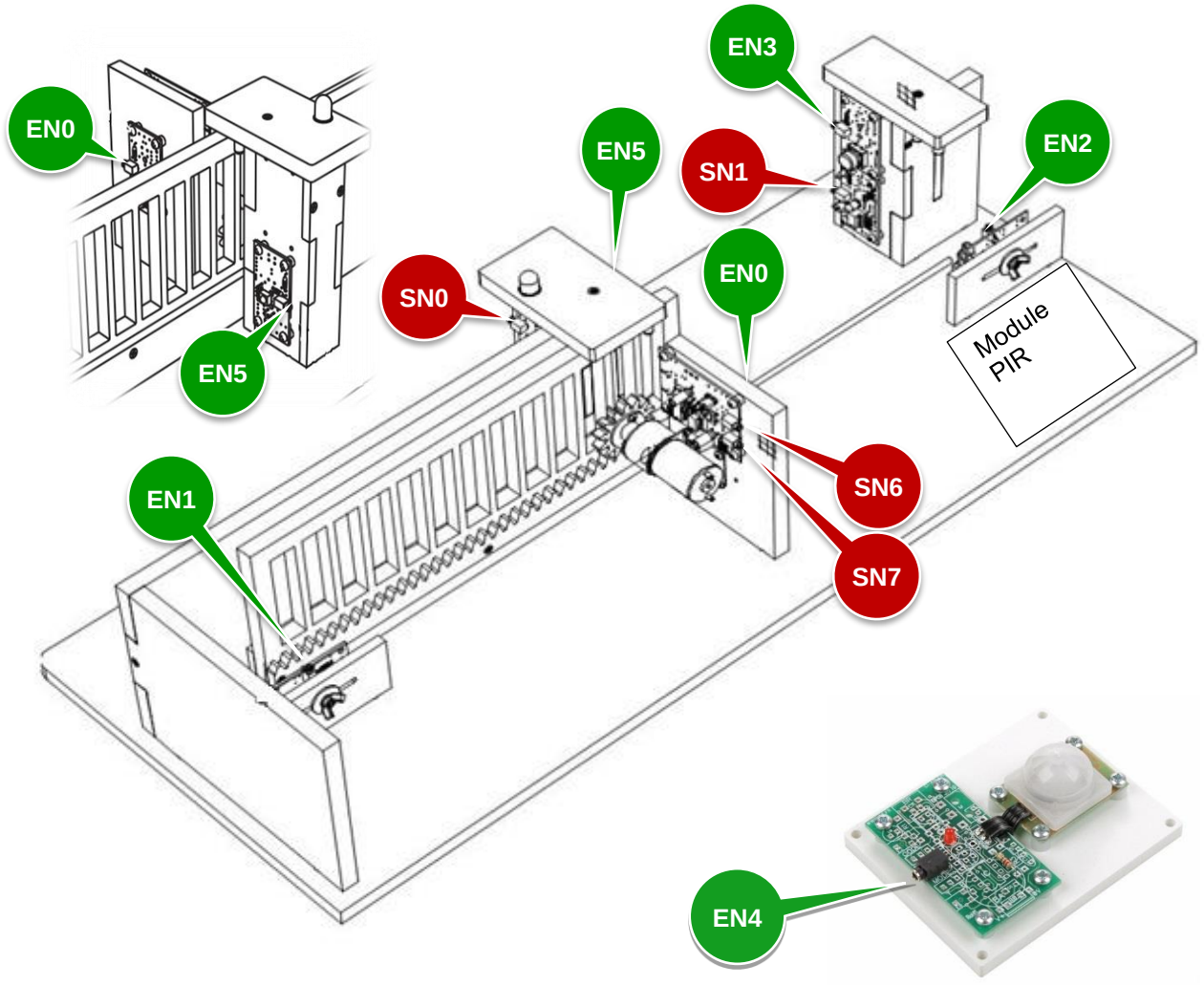


Schéma de câblage



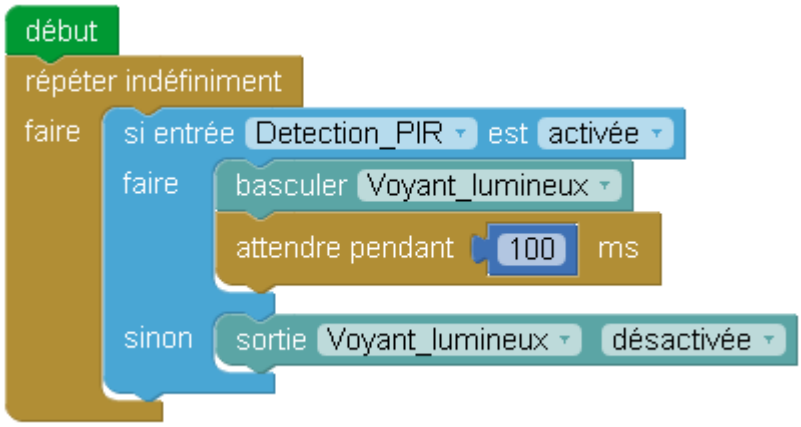
Exercice niveau 3 – D.1 : Utilisation du capteur PIR

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

Notion abordée : utilisation du capteur de présence PIR.

Correction :

Blocs



Fichier Blockly : PC_N3_D1.xml

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

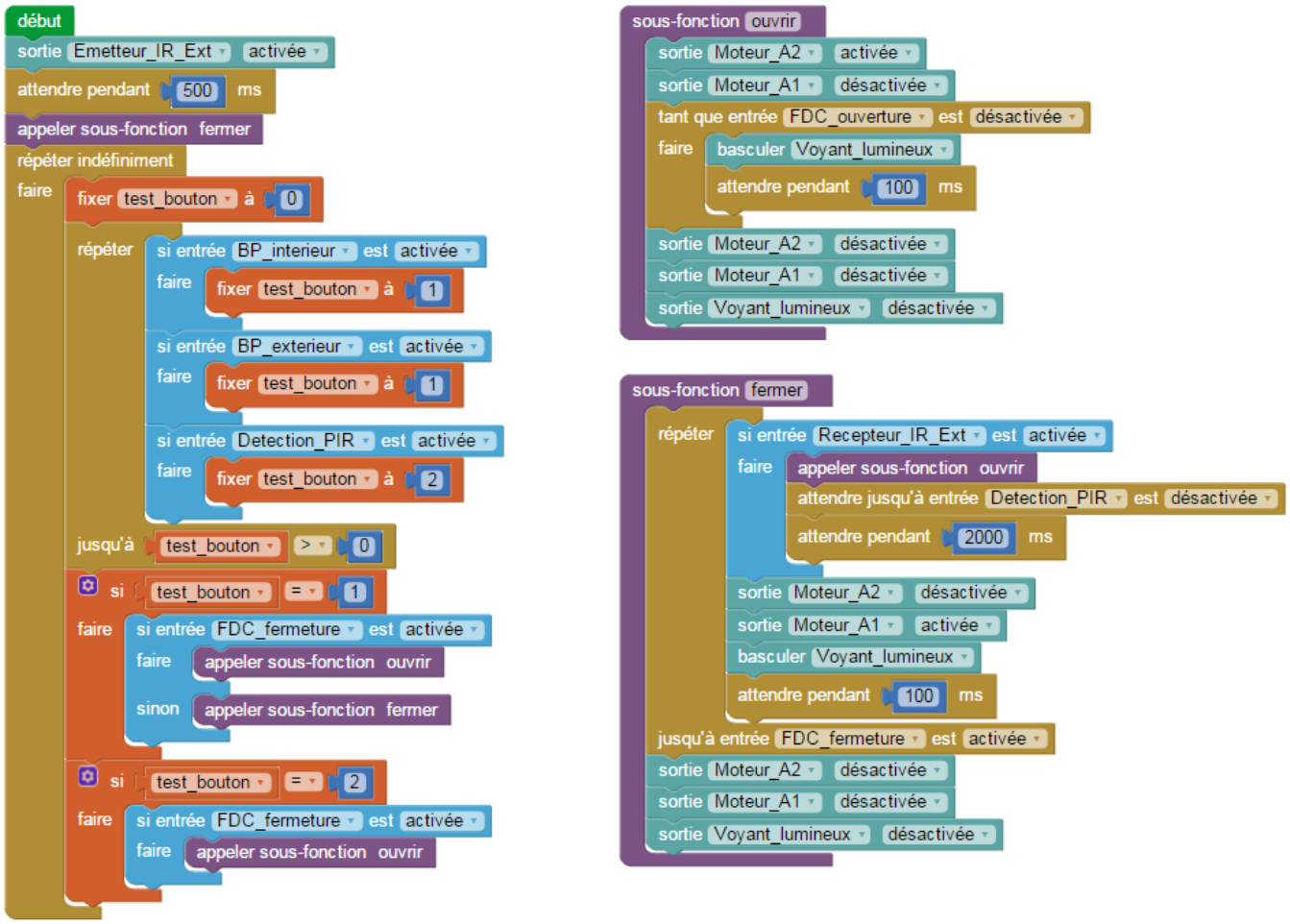
Exercice niveau 3 – D.2 : Ouverture contrôlée à l'aide du PIR

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

Notion abordée :

Correction :

Blocs



Fichier Blockly : PC_N3_D2.xml

www.a4.fr

Concepteur et fabricant de matériels pédagogiques