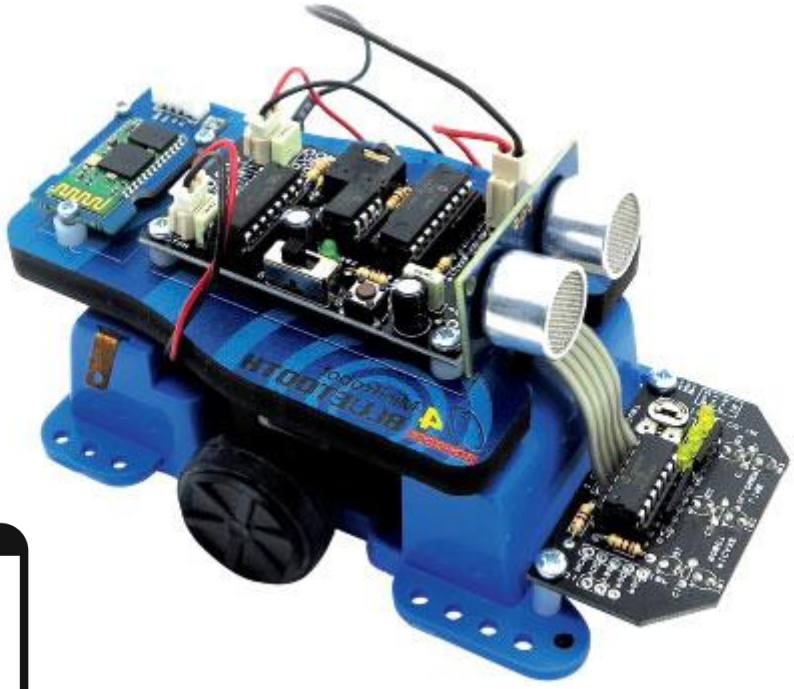
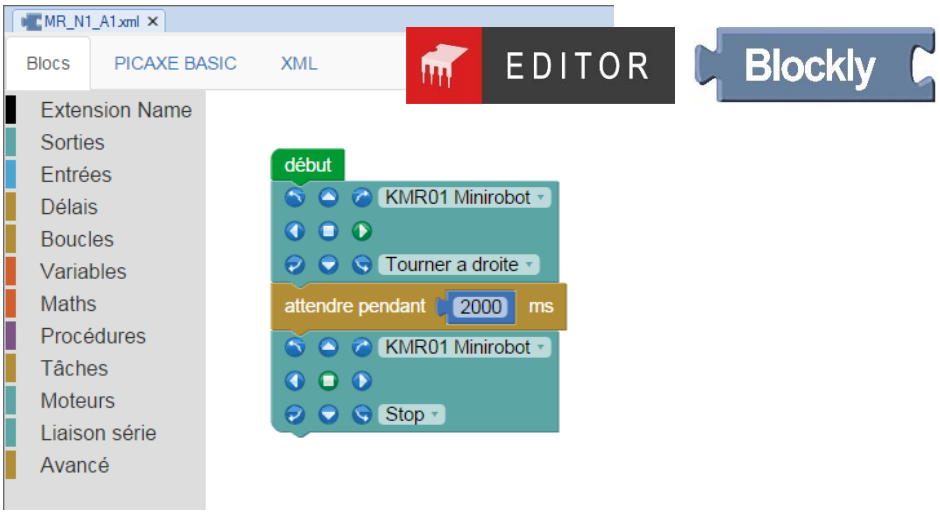
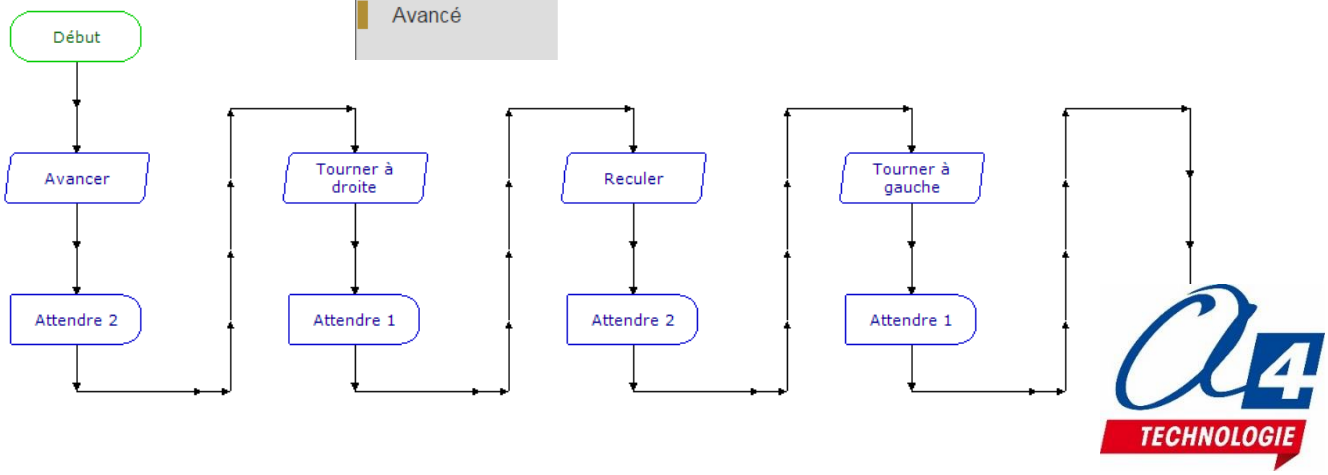
MiniRobot
Robot programmable avec Editor / Blockly
Programmes / Activités / Exercices

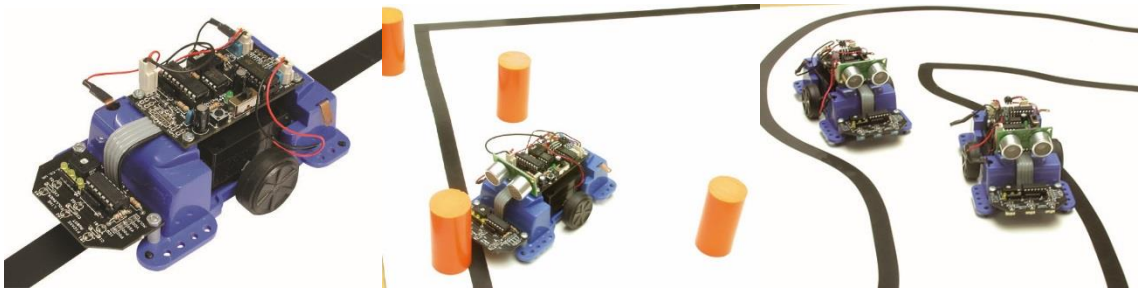






Activités autour du MiniRobot

Activités sur pistes : suivi de ligne, détection d'obstacles, challenge robots aspirateurs …



Impression 3 D – personnalisation du MiniRobot (pince robotisée, coque…)

Ressources disponibles pour le projet MiniRobot

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

MiniRobot

- Fichiers SolidWorks du robot et de ses options.
- Fichier STL pour la coque.

Logiciels Editor 6 et App Inventor

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

Activités / Programmation

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

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

SOMMAIRE

Introduction

Le projet MiniRobot s'adresse aussi bien à des utilisateurs totalement débutants (découverte progressive de la programmation par organigrammes et blocs) qu'à des utilisateurs avertis pour créer des scénarios de programmation élaborés allant jusqu'à l'interaction du robot avec un smartphone.

Le MiniRobot

Le minirobot est un robot de table conçu pour la découverte et l'apprentissage de la programmation.
Son faible encombrement permet de mener facilement des activités robotiques dans un espace restreint se limitant à une feuille format A4 ou A3 à proximité de l'ordinateur utilisé pour le programmer.

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 MiniRobot grâce à ses capteurs et actionneurs. Tous les exemples sont réalisés à partir de Editor / Blockly et les premiers exemples de programmes sont également fournis avec la version logigramme correspondante.

Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour du MiniRobot ont été réalisés sous **Editor 6 / Blockly**. 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 du Minirobot 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 robot 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 robot (version de base).

Niveau 3 :
Exemples d'utilisation des différentes options proposées autour du MiniRobot : Bluetooth, télémètre à ultrasons, détection d'obstacles.

Les fiches exercices

Pour chaque niveau de programmation, nous vous proposons des fiches exercices avec :
- un objectif : ce que doit faire le programme ;
- un fichier de correction qui propose un exemple de programme réalisé sous Picaxe Editor 6 en blocs (extension .xml) et en organigrammes pour le niveau 1 uniquement (extension .plf).

Deux approches :

1. 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.).
2. 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 :
- **MR** : pour MiniRobot
- **N** : niveau de programmation 1-2-3
- **A-B-C** : jeu d'instructions du plus simple au plus avancé

Exemple : MR_N1_C3.xml
Correspond au niveau 1 avec le jeu d'instructions C.

Prérequis

Pour la version de base :

- Installer le logiciel **Picaxe Editor 6** ou **Blockly for Picaxe :** http://www.picaxe.com/Software
- **Câble de programmation** Picaxe USB (Réf : CABLE-USBPICAXE).
- 4 piles AA de 1,5 V.

Pour l'option Bluetooth :

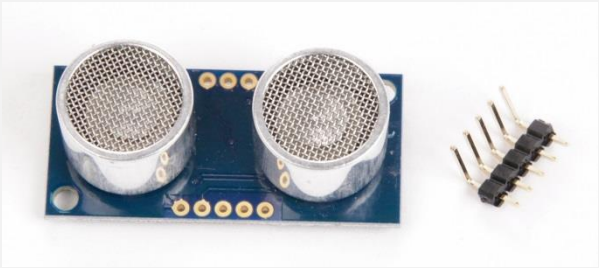
- **Tablette ou smartphone** Android 5 ou + équipés de Bluetooth V3.

- Connexion internet pour accéder à **App Inventor :** http://ai2.appinventor.mit.edu/

- Compte Gmail requis.

Options

Bluetooth :
L'option Bluetooth pour MiniRobot permet de piloter le robot à distance avec une tablette ou un smartphone Android.
Elle est composée d'une platine de fixation qui s'adapte sur le châssis du robot pour fixer un module Bluetooth Grove V3.
La platine permet aussi de fixer une carte Arduino Uno.
Réf : K-MR-BLTH



Télémètre à ultrasons :
Renvoie une valeur de distance entre 0 et 2,5 m. (Valeur du capteur dans le vide, env. 0 à 0,75 cm quand placé sur le robot).
Réf : K-MR-US

Détection d'obstacles :
Un capteur de proximité infrarouge prévient quand un obstacle est détecté à moins de 5 cm.
Réf : K-MR-MSIR

Microrupteurs :
Réf : K-MR-MICRORUP

Télécommande infrarouge
Réf : TELEC-UNIV-IR

Environnement de programmation graphique

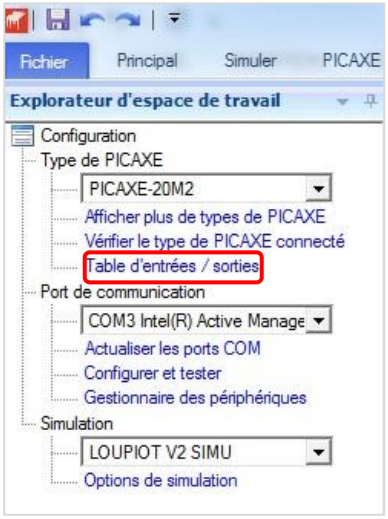
Tous les programmes correspondant aux activités ont été réalisés sous **Editor 6/ Blockly**. En effet, ce logiciel de programmation graphique présente plusieurs avantages :

- 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.

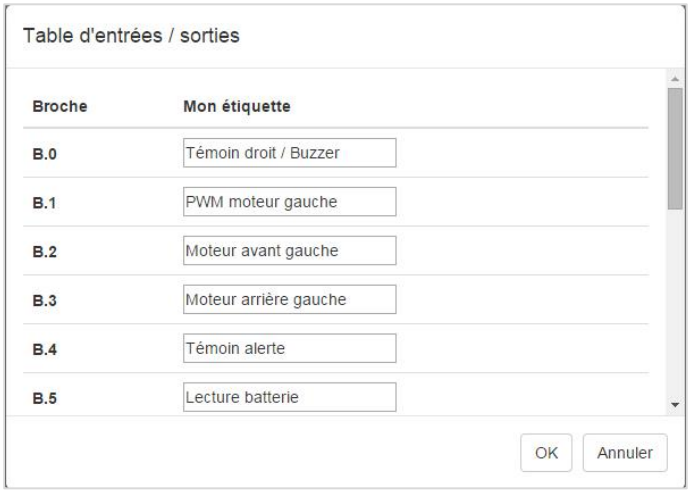
Note : vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

Personnalisation des entrées/ sorties

A partir de Picaxe Editor 6, dans l'explorateur d'espace de travail cliquer sur **Table d'entrées / sorties**.



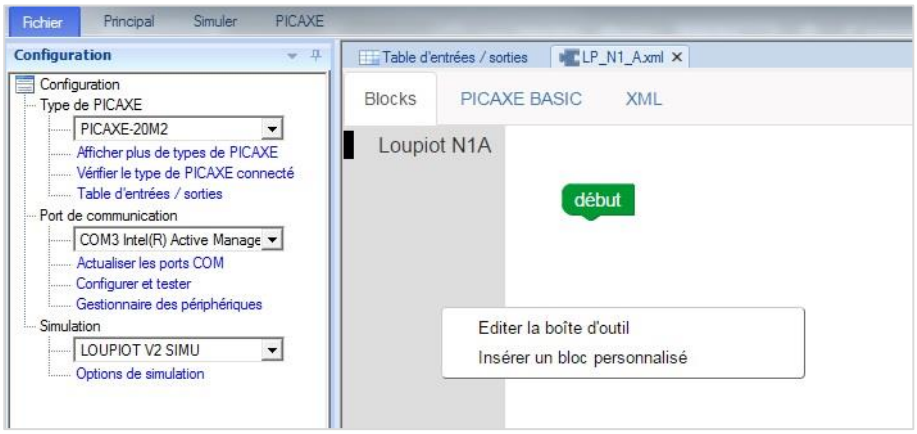
Une fenêtre apparaît à partir de laquelle vous pouvez modifier les noms de toutes les entrées et sorties dans la zone « Mon étiquette ».



Valider en cliquant sur **OK**.

Personnalisation du jeu d'instructions

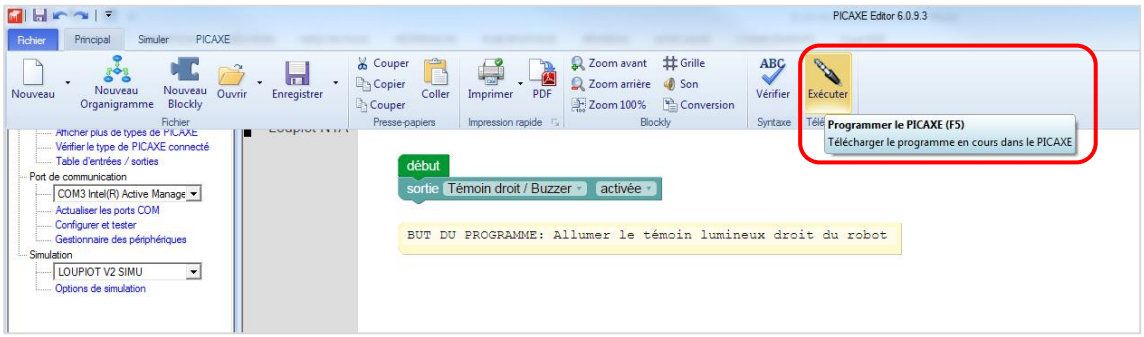
Vous pouvez personnaliser l'affichage du jeu d'instructions pour en limiter la quantité afin de faciliter la l'analyse et la correction des erreurs. Faire un clic droit sur la zone des blocs puis cliquer sur **Editer la boite d'outil**.



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 MiniRobot à l'ordinateur avec le câble de programmation USB et le mettre sous tension. A partir de Editor 6, ouvrir un programme.

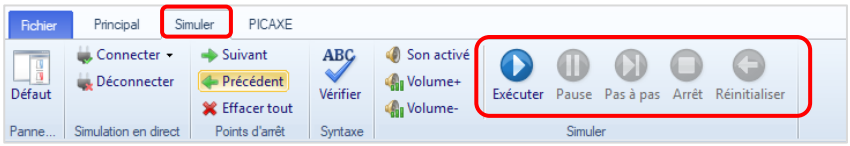


A partir du menu **Principal** ou du menu **PICAXE**, cliquer sur le bouton **Exécuter**. Vous pouvez également utiliser la touche **F5** de votre clavier.

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 le robot.
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 niveau 1 (version de base)

Niveau 1 : découverte progressive des fonctionnalités de base du MiniRobot et maîtrise des principes fondamentaux pour concevoir un programme : séquences, boucles, structures conditionnelles et variables.

Liste des programmes Niveau 1

Nom du fichier	Description	Objectif
Niveau 1 A		
MR_N1_A1	Activer un moteur	- Gestion des moteurs du MiniRobot.
MR_N1_A2	Avancer puis s'arrêter	
MR_N1_A3	Tourner à droite puis à gauche	
MR_N1_A4	Tourner en rond	
MR_N1_A5	Mouvement répété	
MR_N1_A6	Accélération brutale	
MR_N1_A7	Compteur	
Niveau 1 B		
MR_N1_B1	Arrêt	- Gestion du module microrupteurs. - Pilotage du MiniRobot en fonction de l'état d'une entrée.
MR_N1_B2	Changer de direction	
MR_N1_B3	Eviter un obstacle	
MR_N1_B4	Eviter un obstacle (2)	
Niveau 1 C		
MR_N1_C1	Arrêt sur ligne	- Gestion du module suiveur de ligne.
MR_N1_C2	Suivi de ligne	
MR_N1_C3	Piste	
MR_N1_C4	Périmètre	
Niveau 1 D		
MR_N1_D1	Obstacle	- Gestion du module télémètre à ultrasons.
MR_N1_D2	Slalom	
MR_N1_D3	Cible	
Niveau 1 E		
MR_N1_E1	Télécommande	- Pilotage du MiniRobot avec la télécommande infrarouge.
MR_N1_E2	Pilotage du minirobot	
Niveau 1 F		
MR_N1_F1	Avancer et s'arrêter	- Pilotage du MiniRobot en Bluetooth avec des applications sur smartphone.
MR_N1_F2	Pilotage via application Bluetooth	
MR_N1_F2	Pilotage avec accéléromètre	

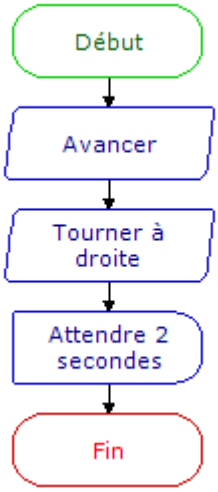
Exercice niveau 1 – A1 : Activer un moteur

Objectif : Activer une fonction de déplacement pendant 2 secondes puis arrêter le robot

Notion(s) abordée(s) : activation d'une sortie, utilisation de l'instruction spéciale pour animer le robot.

Instruction(s) utilisée(s) :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : MR_N1_A1_Organigramme.plf	Fichier Blockly : MR_N1_A1.xml

Remarque(s) :

- Un programme téléchargé écrase le précédent.
- Le programme démarre à partir de l'instruction « **Début** » dès la fin du téléchargement ou dès la mise sous tension du robot.
- Par défaut, toutes les sorties de la carte de pilotage du robot sont désactivées. L'activation d'une sortie reste valide tant qu'une instruction de désactivation n'est pas exécutée, même quand le programme est terminé.

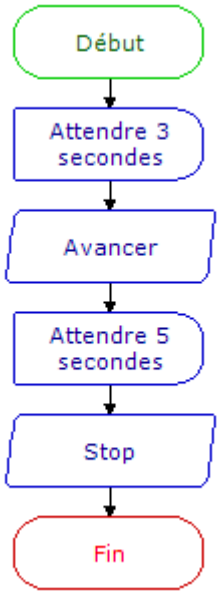
Exercice niveau 1 – A2 : Avancer puis s'arrêter

Objectif : Au bout de 3 secondes, faire avancer le robot pendant 5 secondes puis l'arrêter.

Notion(s) abordée(s) :

Instruction(s) utilisée(s) :

Correction :

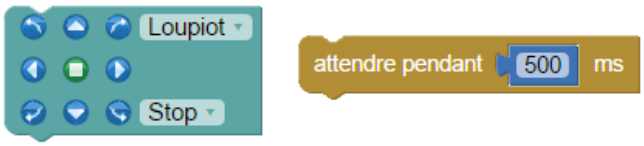
Organigramme	Blocs
	
Fichier organigramme PE6 : MR_N1_A2_Organigramme.plf	Fichier Blockly : MR_N1_A2.xml

Exercice niveau 1 – A3 : Tourner à droite puis à gauche

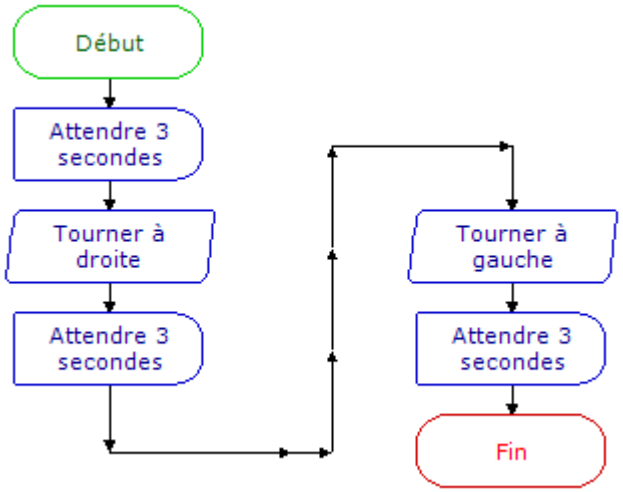
Objectif : Au bout de 3 secondes, faire tourner le robot sur lui-même à droite pendant 3 secondes, puis à gauche pendant 3 secondes, puis l'arrêter.

Notion(s) abordée(s) :

Instruction(s) utilisée(s) :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : MR_N1_A3_Organigramme.plf	Fichier Blockly : MR_N1_A3.xml

Remarque(s) :

- L'instruction « Tourner » inverse le sens des moteurs ce qui a pour conséquence de faire tourner le robot sur lui-même.
- L'instruction « Virer » anime une roue à la fois. Le robot décrit un arc de cercle autour de la roue opposée (voir exemple suivant).

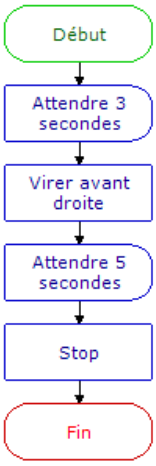
Exercice niveau 1 – A4 : Tourner en rond

Objectif : faire tourner le robot en rond grâce à la fonction « virer » pendant 5 secondes puis l'arrêter.

Notion(s) abordée(s) :

Instruction(s) utilisée(s) :

Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : MR_N1_A4_Organigramme.plf	Fichier Blockly : MR_N1_A4.xml

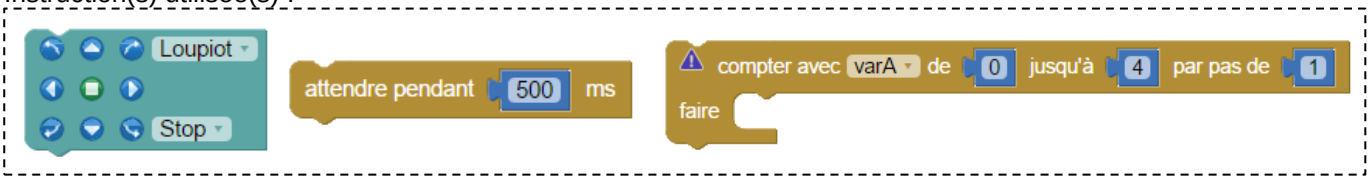
Remarque(s) :

Exercice niveau 1 – A5 : Mouvement répété

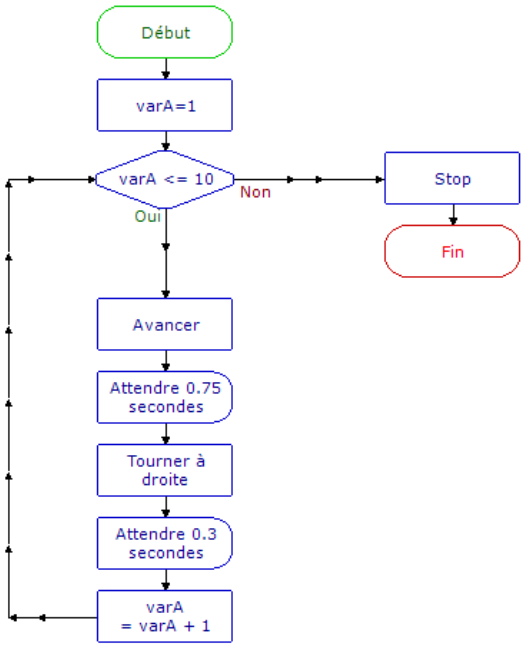
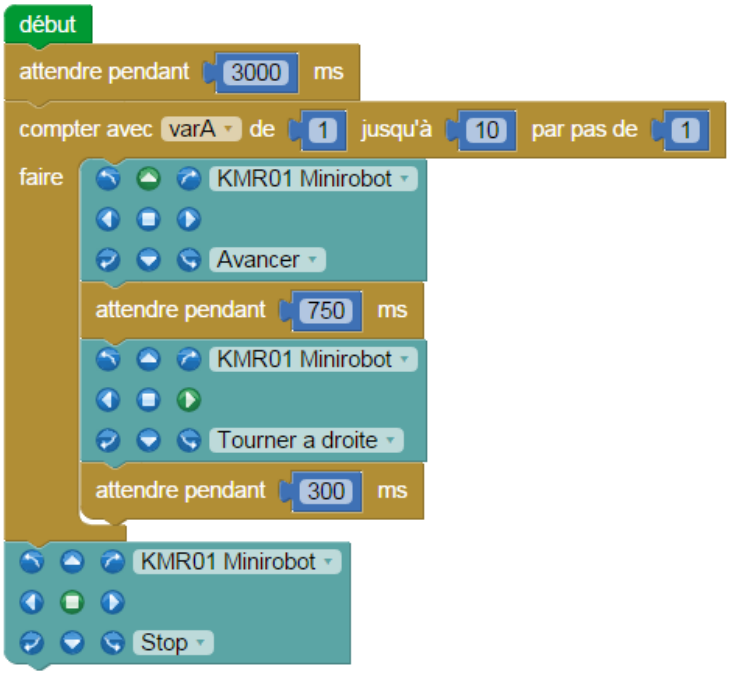
Objectif : répéter 10 fois l'action suivante : avancer puis tourner à droite.

Notion(s) abordée(s) : Comptage avec une variable varA

Instruction(s) utilisée(s) :



Correction :

Organigramme	Blocs
	
Fichier organigramme PE6 : MR_N1_A5_Organigramme.plf	Fichier Blockly : MR_N1_A5.xml

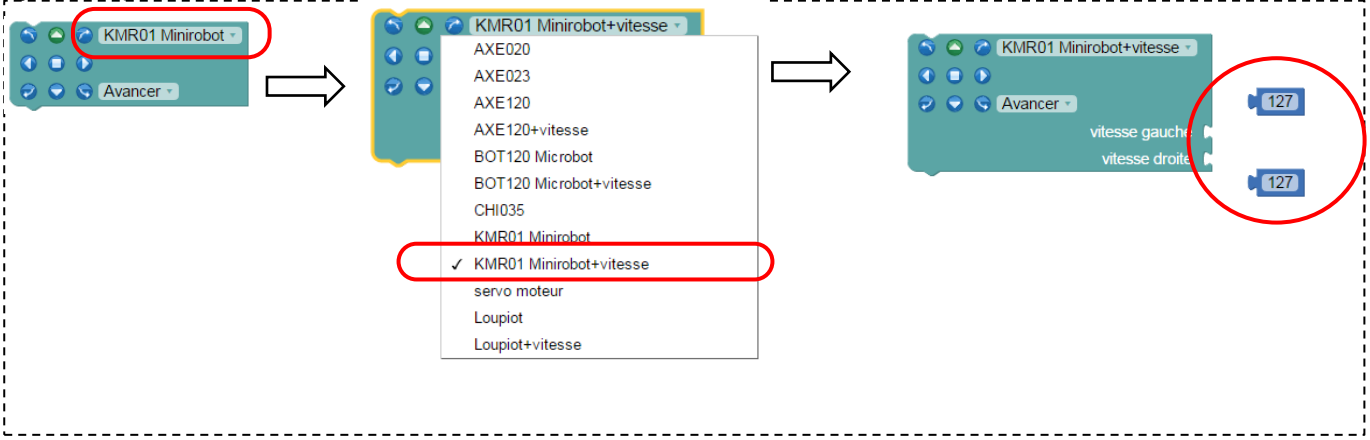
Exercice niveau 1 – A6 : Accélération brutale

Objectif : Au bout de 3 secondes, faire avancer le robot à 50% de sa vitesse pendant 3 secondes, puis à 100% pendant 3 secondes avant de l'arrêter.

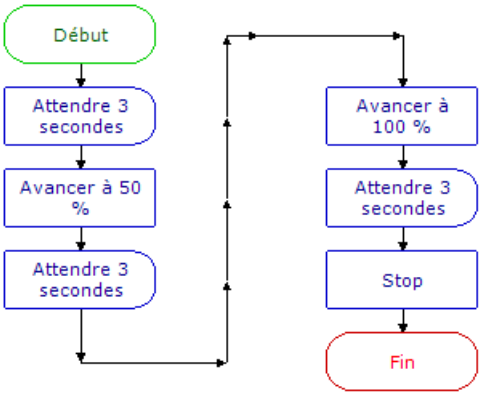
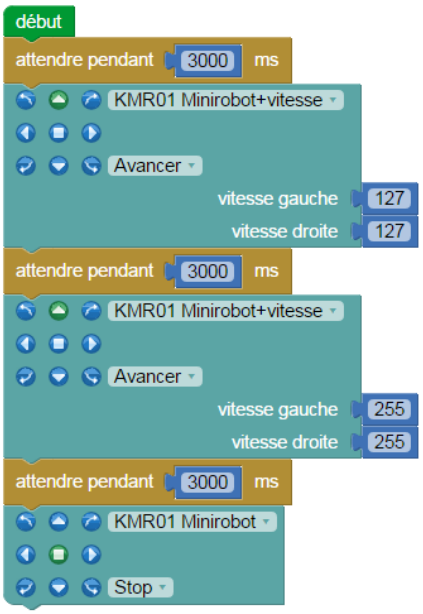
Notion(s) abordée(s) : ajouter le contrôle de la vitesse au bloc de contrôle des moteurs.

Instruction(s) utilisée(s) :

Ajouter l'option vitesse sur le bloc de contrôle des moteurs :



Correction :

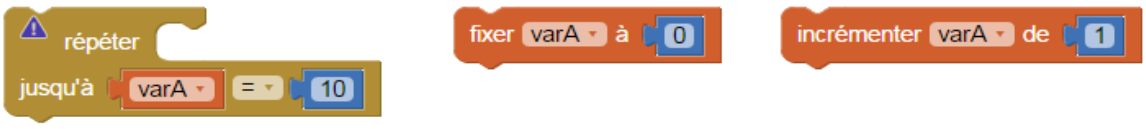
Organigramme	Blocs
	
Fichier organigramme PE6 : MR_N1_A6_Organigramme.plf	Fichier Blockly : MR_N1_A6.xml

Exercice niveau 1 – A7 : Compteur

Objectif : Répéter une figure 6 fois de suite puis s'arrêter à l'aide d'une variable à incrémenter

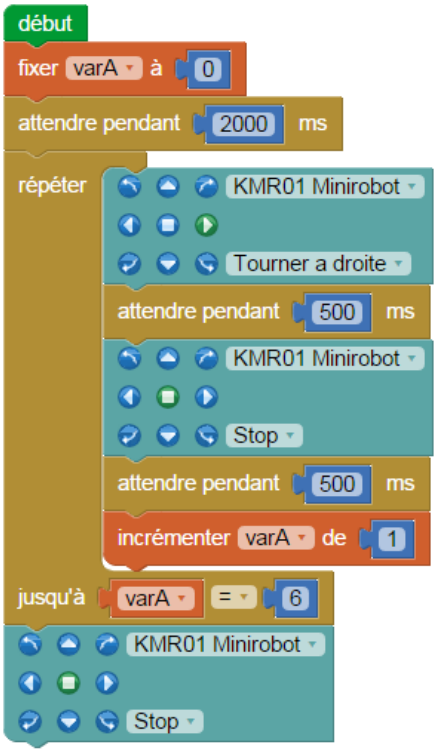
Notion(s) abordée(s) : Définition et test d'une variable.

Instruction(s) utilisée(s) :



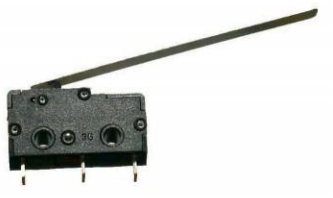
Correction :

Blocs

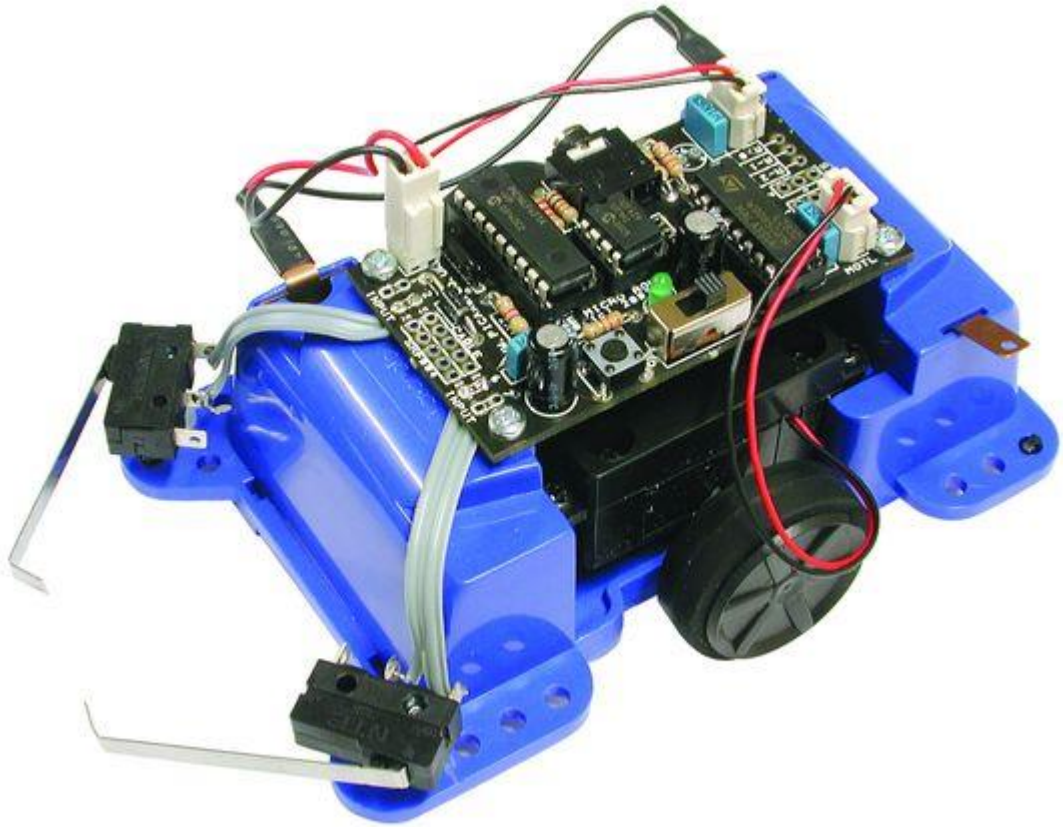


Fichier Blockly : MR_N1_A7.xml

Option : Module microrupteurs



Les microrupteurs fonctionnent comme des interrupteurs, dans le cas du minirobot, ceux-ci sont placés à l'avant afin de détecter des obstacles à l'avant du robot à gauche ou à droite.
Ceux-ci sont programmés comme des entrées qui sont soit activées soit désactivées.

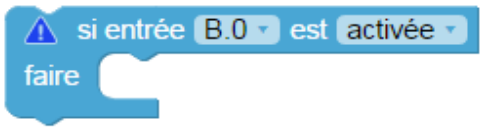


Exercice niveau 1 – B1 : Arrêt

Objectif : Avancer et s'arrêter lorsqu'un microrupteur détecte un obstacle

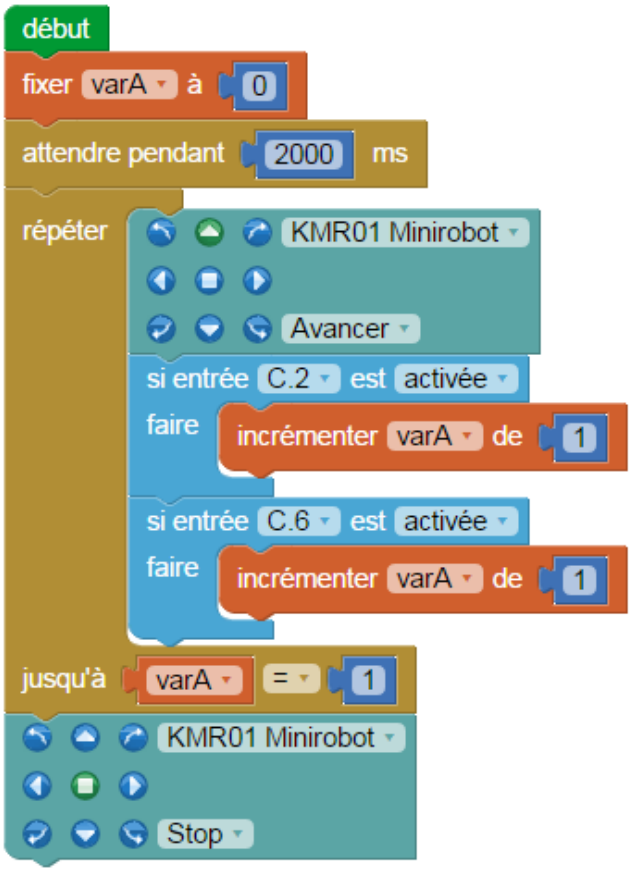
Notion(s) abordée(s) : Test de l'état d'un microrupteur

Instruction(s) utilisée(s) :



Correction :

Blocs

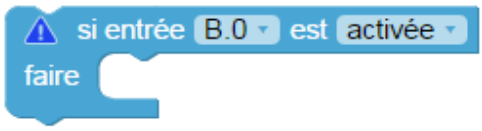


Fichier Blockly : MR_N1_B1.xml

Exercice niveau 1 – B2 : Changer de direction

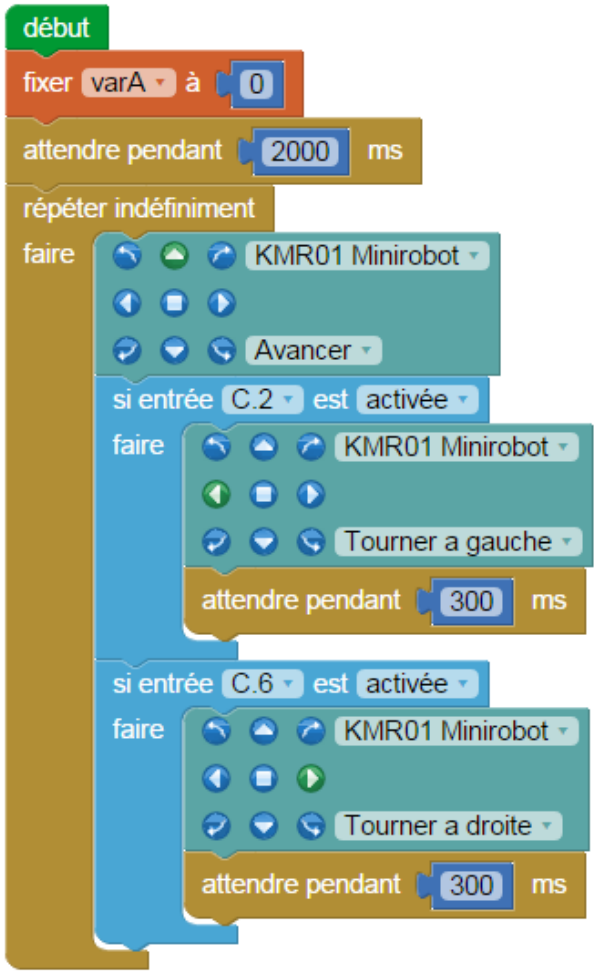
Objectif : Tourner à gauche lorsqu'un obstacle est détecté à droite, aller à droite lorsqu'un obstacle est détecté à gauche, sinon aller tout droit

Instruction(s) utilisée(s) :



Correction :

Blocs

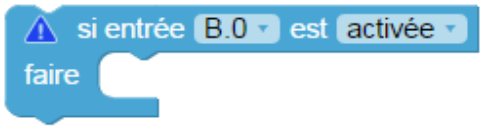


Fichier Blockly : MR_N1_B2.xml

Exercice niveau 1 – B3 : Eviter un obstacle

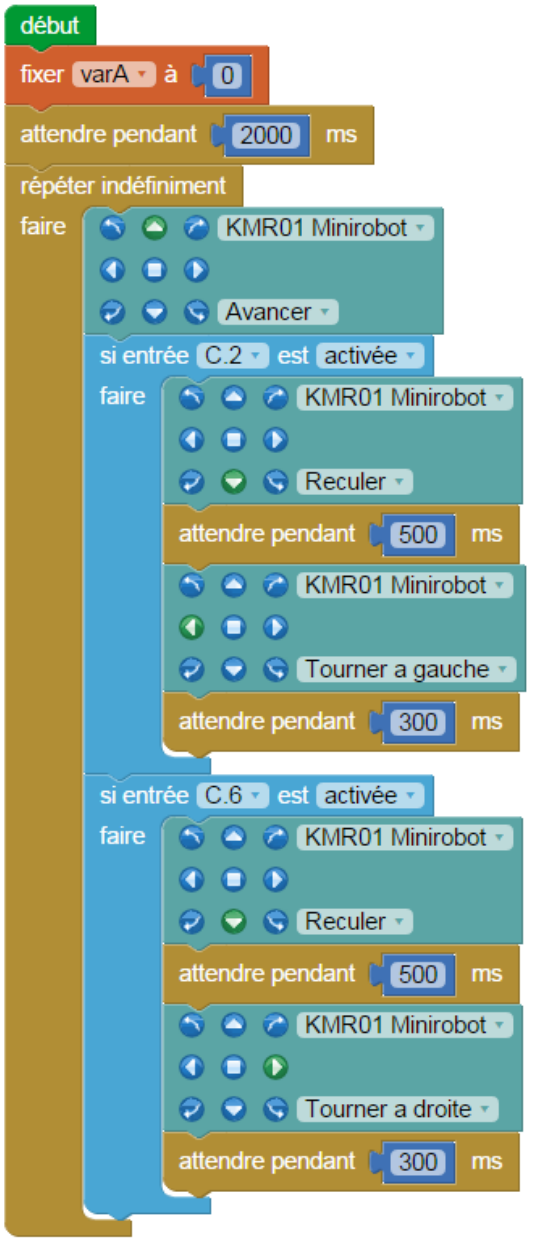
Objectif : Reprendre l'exercice précédent, reculer puis tourner au contact d'un obstacle

Instruction(s) utilisée(s) :



Correction :

Blocs

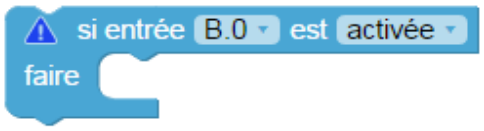


Fichier Blockly : MR_N1_B3.xml

Exercice niveau 1 – B4 : Eviter un obstacle

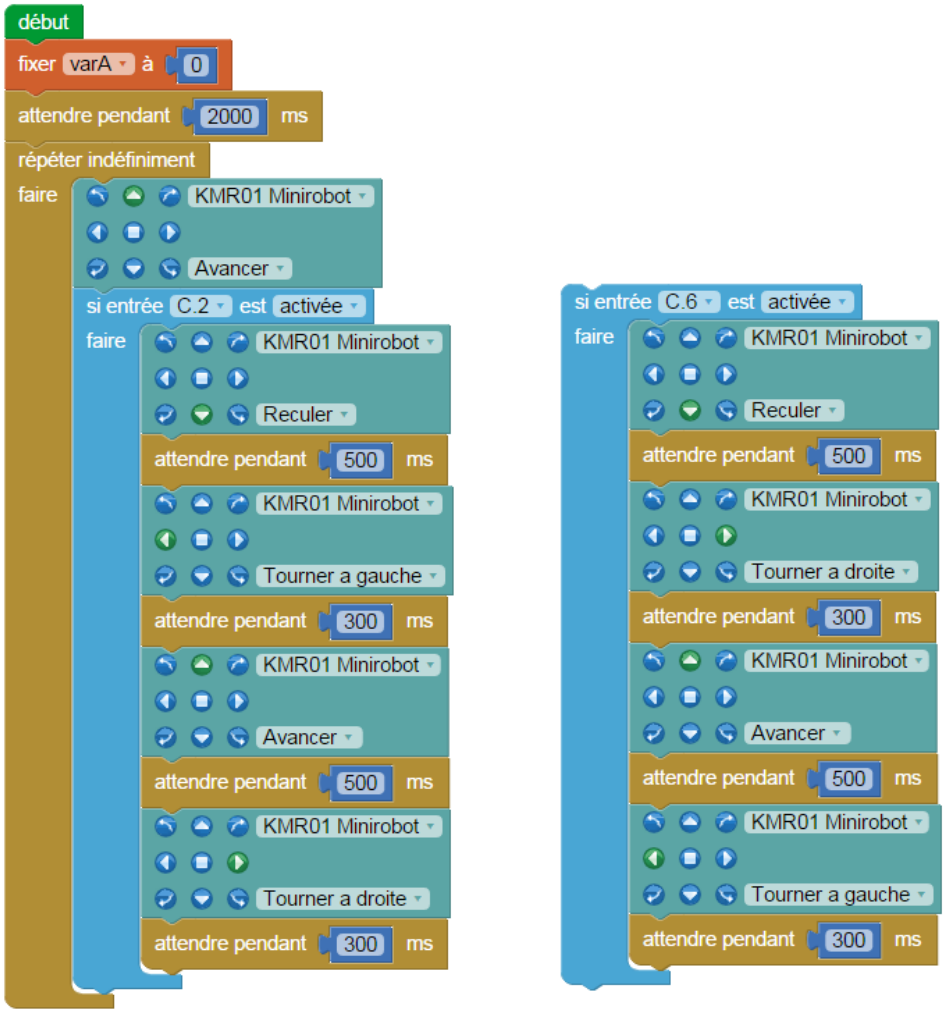
Objectif : Contourner un obstacle en reculant puis en tournant dans une direction puis dans l'autre pour repartir dans le même sens qu'à l'arrivée sur l'obstacle.

Instruction(s) utilisée(s) :



Correction :

Blocs

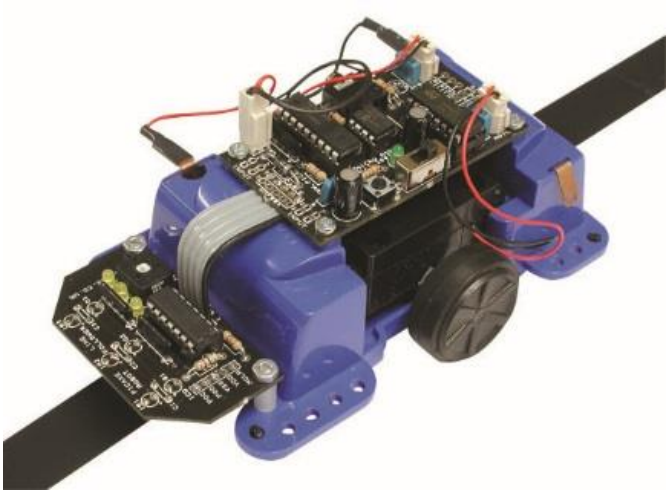


Fichier Blockly : MR_N1_B4.xml

Option : Module suiveur de ligne

Le module suiveur de ligne possède trois couples émetteur / récepteur infrarouge permettant de détecter un trait noir. Ces trois couples sont considérés comme 3 entrées indépendantes.

Le capteurs sont caractérisés par les entrées suivantes :
Gauche : Entrée 6
Centre : Entrée 1
Droite : Entrée 2

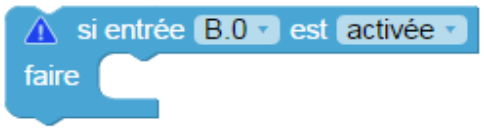


Exercice niveau 1 – C1 : Arrêt sur ligne

Objectif : Avancer en ligne droite et s'arrêter au croisement d'un marquage au sol

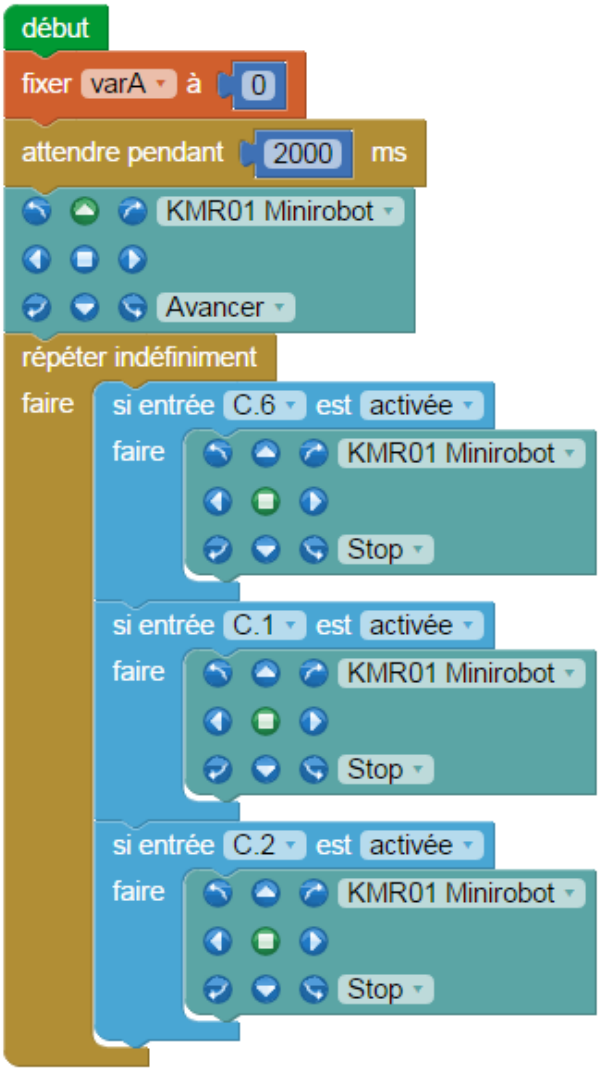
Notion(s) abordée(s) : Utilisation du module suiveur de ligne

Instruction(s) utilisée(s) :



Correction :

Blocs



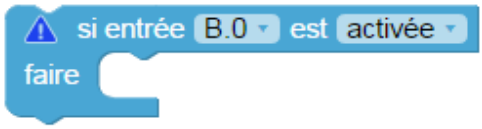
Fichier Blockly : MR_N1_C1.xml

Exercice niveau 1 – C2 : Suivi de ligne

Objectif : Suivre une ligne marquée au sol à l'aide du module suiveur de ligne. Tourner à gauche lors d'une détection sur le capteur de gauche et à droite lors d'une détection sur le capteur de droite.

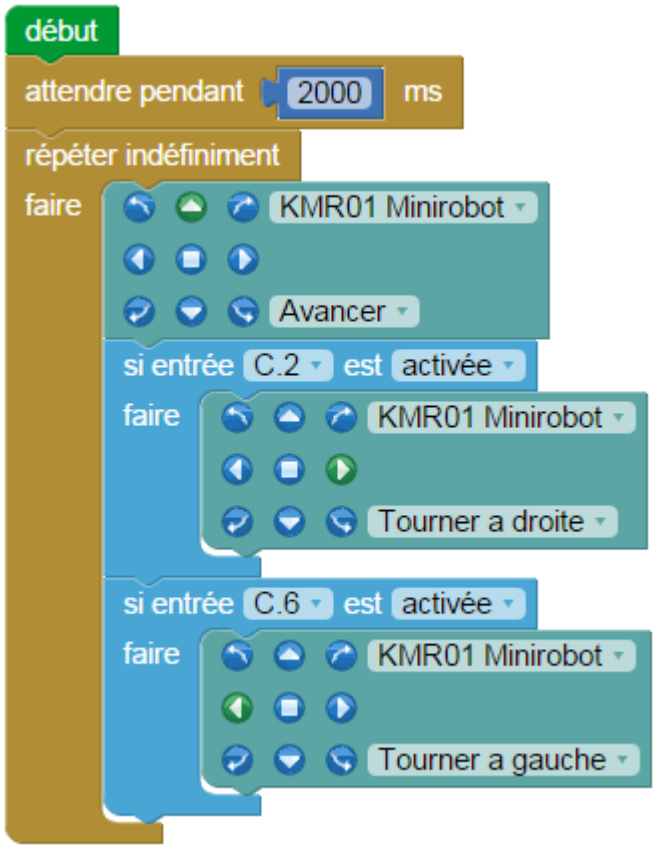
Notion(s) abordée(s) : Utilisation du module suiveur de ligne

Instruction(s) utilisée(s) :



Correction :

Blocs



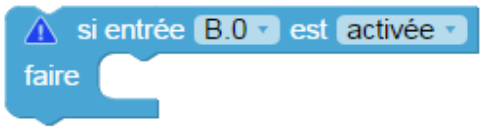
Fichier Blockly : MR_N1_C2.xml

Exercice niveau 1 – C3 : Piste

Objectif : Evoluer dans une piste délimitée par des marquages au sol.

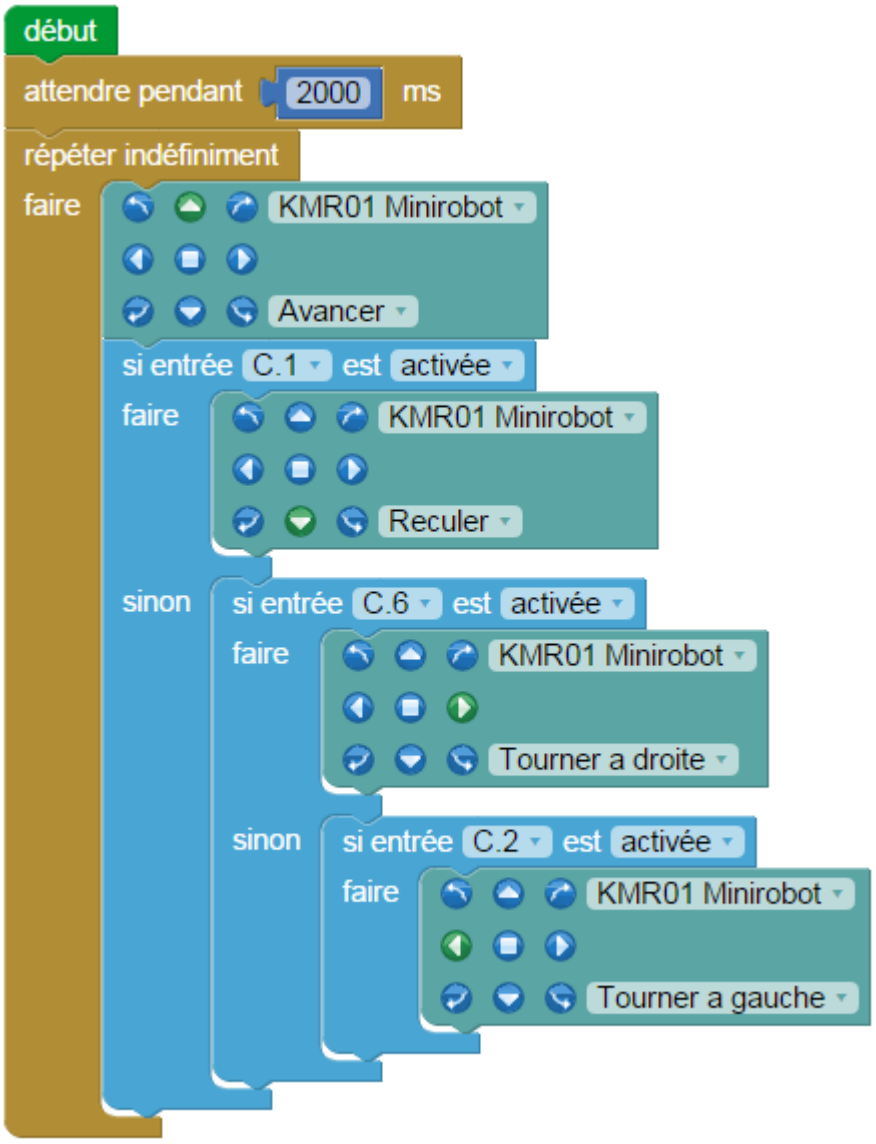
Notion(s) abordée(s) : Utilisation du module suiveur de ligne

Instruction(s) utilisée(s) :



Correction :

Blocs

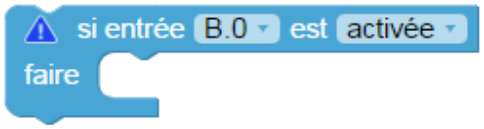


Fichier Blockly : MR_N1_C3.xml

Exercice niveau 1 – C4 : Périmètre

Objectif : Evoluer dans une piste délimitée par des marquages au sol

Instruction(s) utilisée(s) :



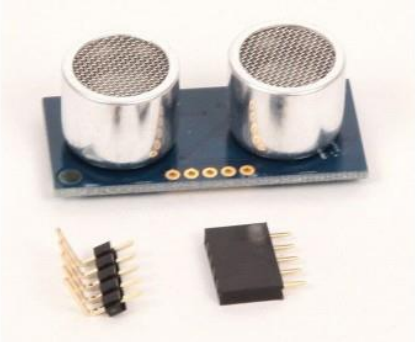
Correction :

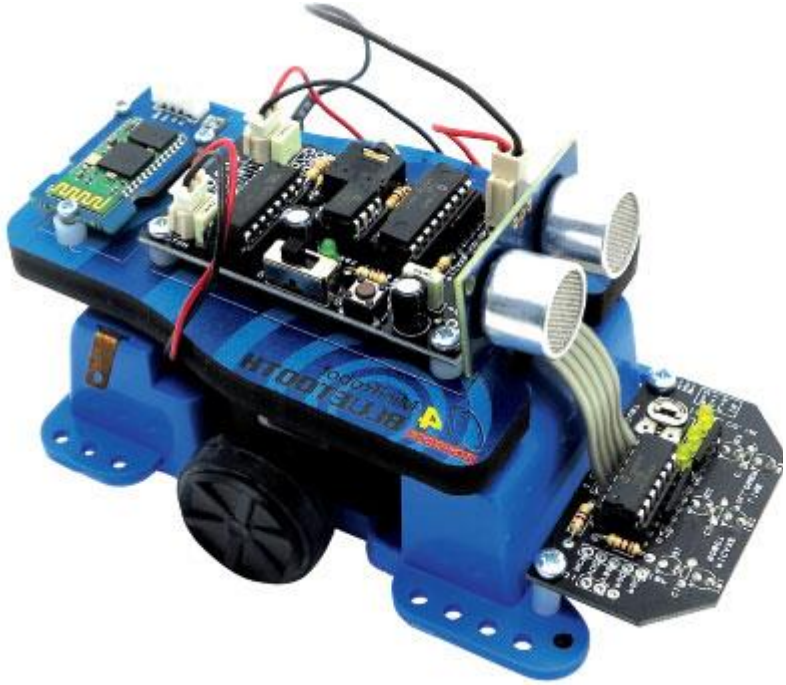
Blocs

Fichier Blockly : MR_N1_C4.xml

Option : Module télémètre à ultrasons

Le module télémètre à ultrasons permet de mesurer une distance entre 2 et 255 cm.
Il fonctionne en deux parties, l'émetteur et le récepteur. Le module calcule le temps mit par un ultrason pour aller de l'émetteur vers le récepteur.
Ce module prend en compte une sortie (Le trigger) et une entrée (Le récepteur).
Une fonction spécifique permet de récupérer facilement cette distance.





Exercice niveau 1 – D1 : S'arrêter devant un obstacle

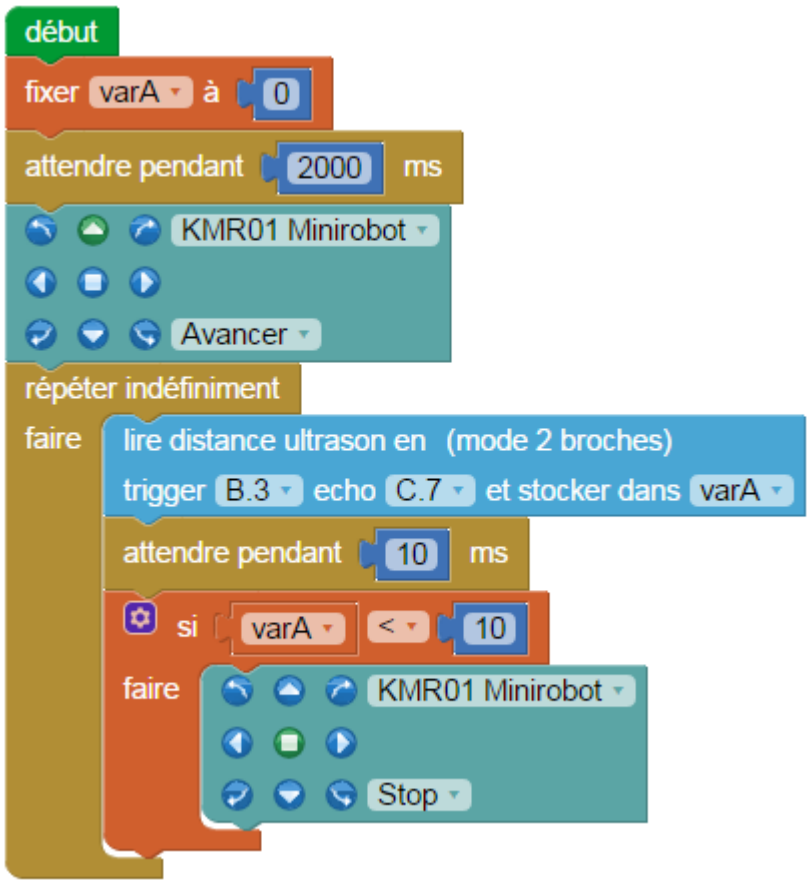
Objectif : Avancer puis s'arrêter à 10 cm d'un obstacle.

Notion(s) abordée(s) : Utilisation du capteur à ultrasons

Instruction(s) utilisée(s) :

Correction :

Blocs



Fichier Blockly : MR_N1_D1.xml

Exercice niveau 1 – D2 : Faire un slalom

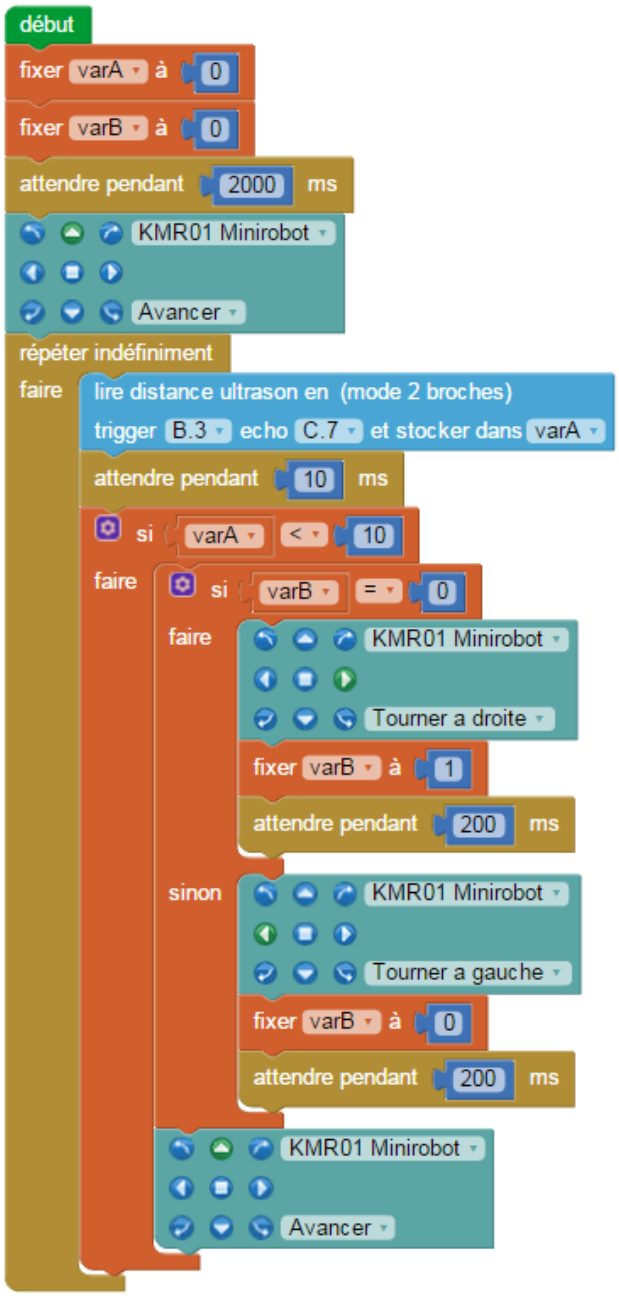
Objectif : Alterner un changement de direction à gauche puis à droite à l'approche d'obstacles

Notion(s) abordée(s) : Utilisation du capteur à ultrasons

Instruction(s) utilisée(s) :

Correction :

Blocs



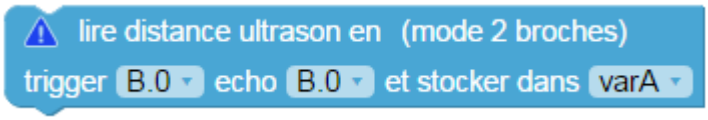
Fichier Blockly : MR_N1_D2.xml

Exercice niveau 1 – D3 : Détecter une cible

Objectif : Balayer une zone pour détecter une cible à une distance inférieure à 20 cm et se diriger vers elle

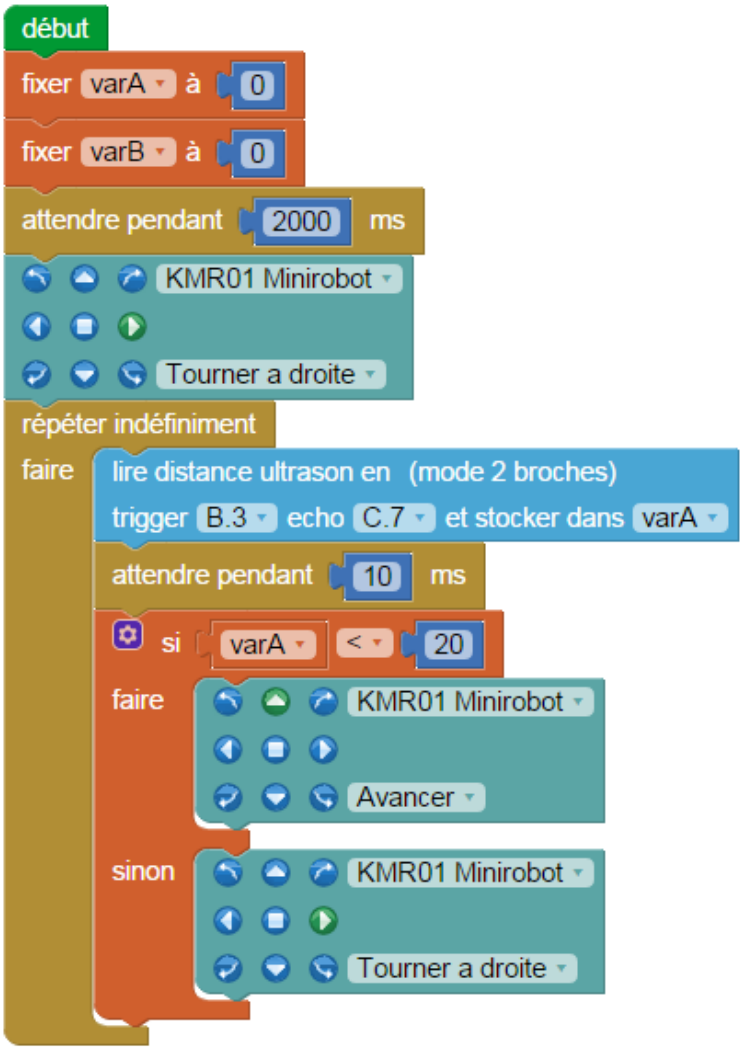
Notion(s) abordée(s) : Utilisation du capteur à ultrasons

Instruction(s) utilisée(s) :



Correction :

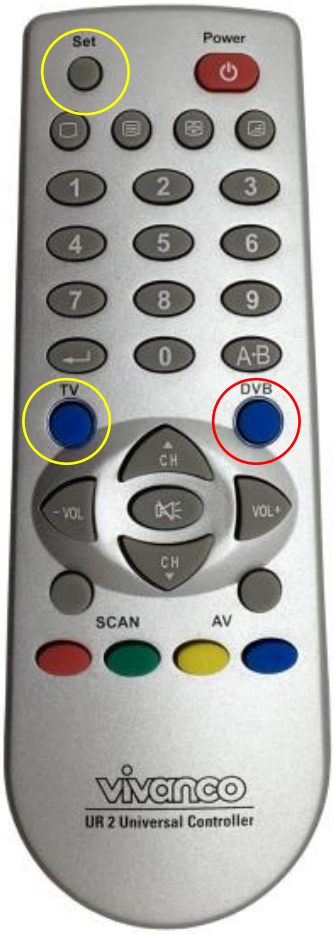
Blocs



Fichier Blockly : MR_N1_D3.xml

Option : Module télécommande infrarouge

Le module télécommande infrarouge permet de piloter le mini robot à l'aide d'envoi de données via la télécommande vers un récepteur infrarouge



Procédure de mise en service pour PICAXE

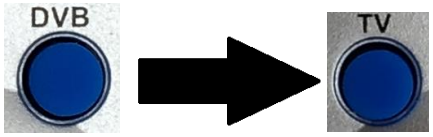
1. Insérer 2 piles AAA dans le logement au dos de la télécommande.

2. Appuyer simultanément sur 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.

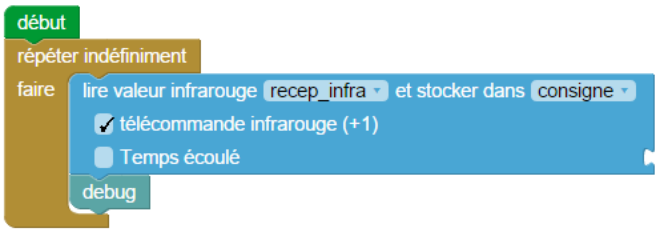
ATTENTION !
La touche **DVB** risque de changer le mode. Appuyer sur **TV** pour revenir dans le bon mode.



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.



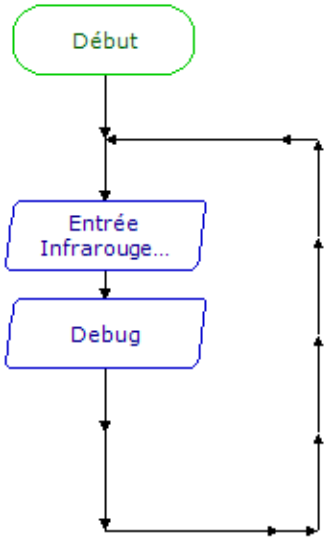
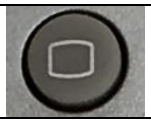
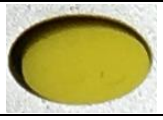


Tableau de correspondance des touches

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.

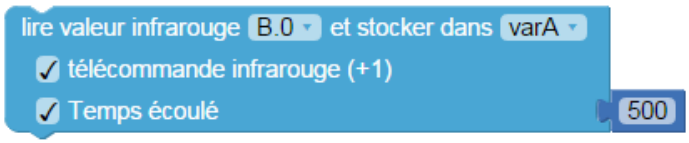
Touche					
Code émis standard	**9**	**0**	**1**	**2**	**3**
Compatibilité activée	**10**	**1**	**2**	**3**	**4**
Touche					
Code émis standard	**4**	**5**	**6**	**7**	**8**
Compatibilité activée	**5**	**6**	**7**	**8**	**9**
Touche					
Code émis standard	**12**	**37**	**16**	**37**	**56**
Compatibilité 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 1 – E1 : Avancer / s'arrêter avec la télécommande IR

Objectif : Avancer sur l'appui du bouton 1, s'arrêter sur l'appui du bouton 2

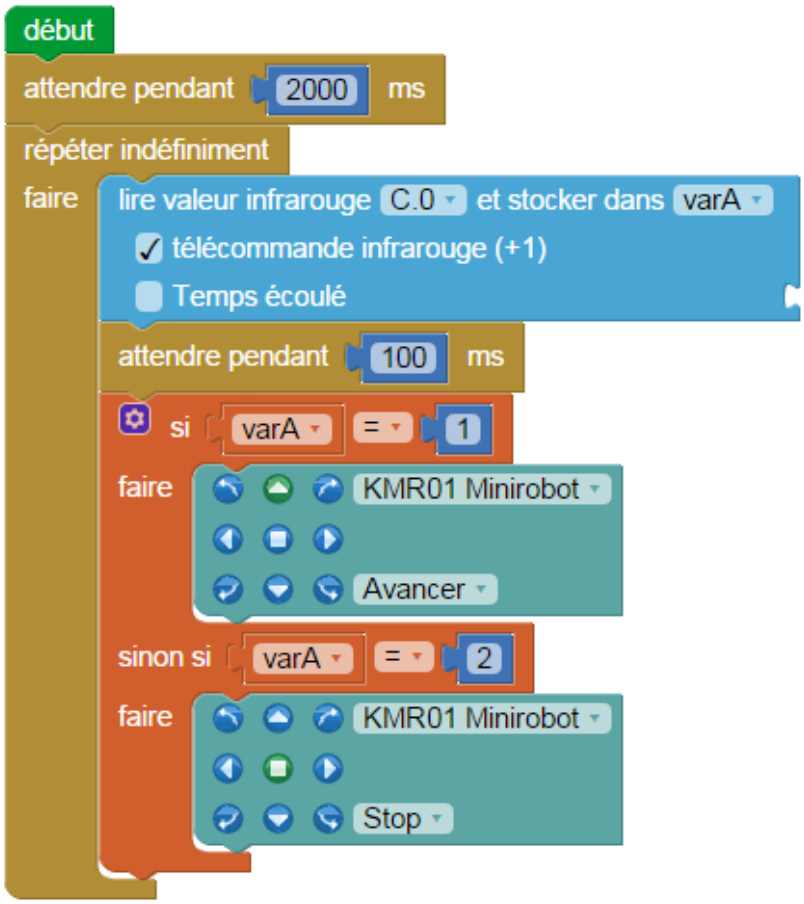
Notion(s) abordée(s) : Utilisation de la télécommande infrarouge

Instruction(s) utilisée(s) :



Correction :

Blocs



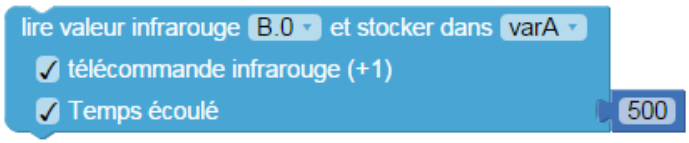
Fichier Blockly : MR_N1_E1.xml

Exercice niveau 1 – E2 : Piloter le MiniRobot avec la télécommande IR

Objectif : Diriger le robot à l'aide des flèches et l'arrêter grâce à un autre bouton.

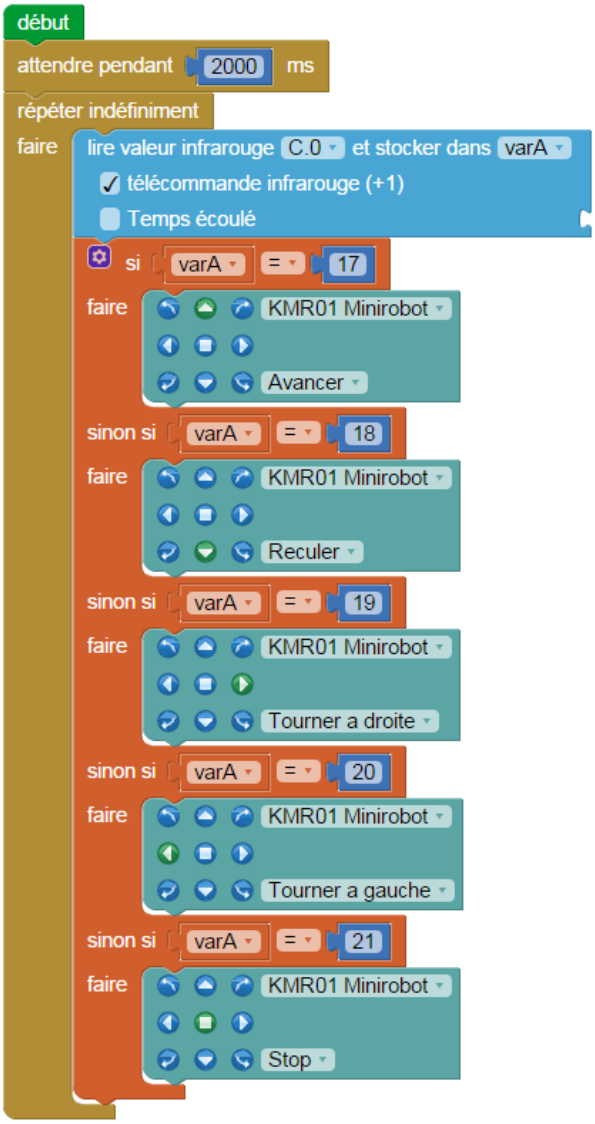
Notion(s) abordée(s) : Utilisation de la télécommande infrarouge

Instruction(s) utilisée(s) :



Correction :

Blocs



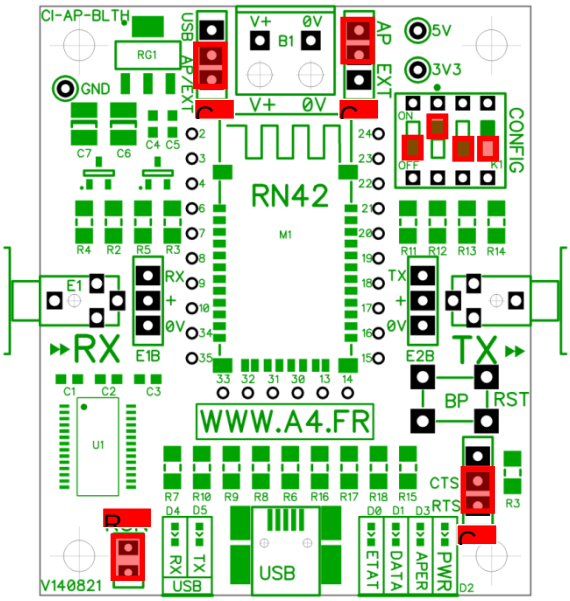
Fichier Blockly : MR_N1_E2.xml

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.



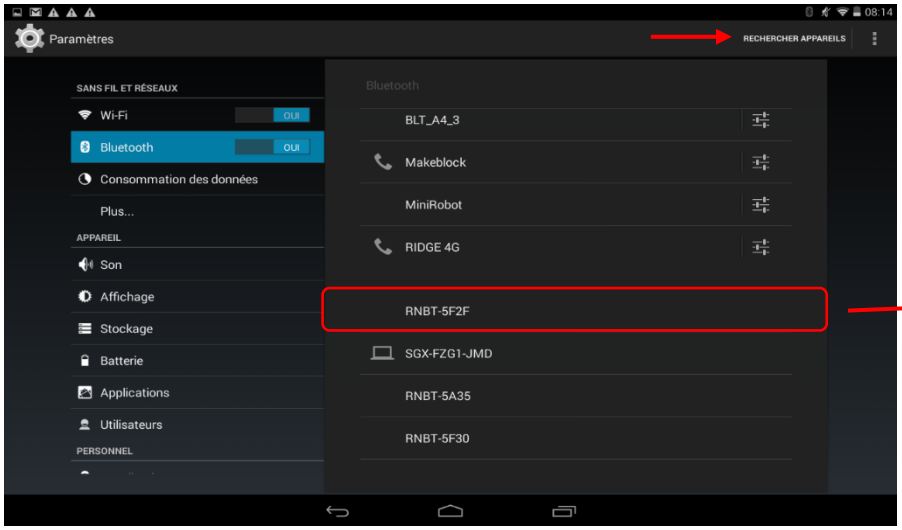
Le cavalier repéré **RUN** est utilisé lors de la mise au point de programmes avec **Arduino**. Il doit être ôté pour permettre le téléversement du programme puis doit être remis lors de l'utilisation.
La mise au point de programmes avec **PICAXE** ne nécessite pas d'ôter ce cavalier pour transférer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

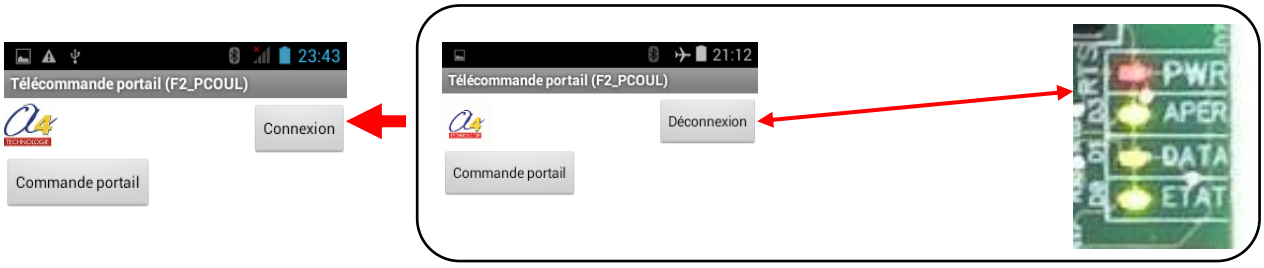
PWR indique que le module est sous tension.
APER indique que le module est associé avec un matériel Bluetooth.
DATA indique qu'il y a un flux de données entre le module et l'appareil avec lequel il est connecté.
ETAT indique que le module est opérationnel. L'affichage clignotant indique qu'il n'est pas 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.



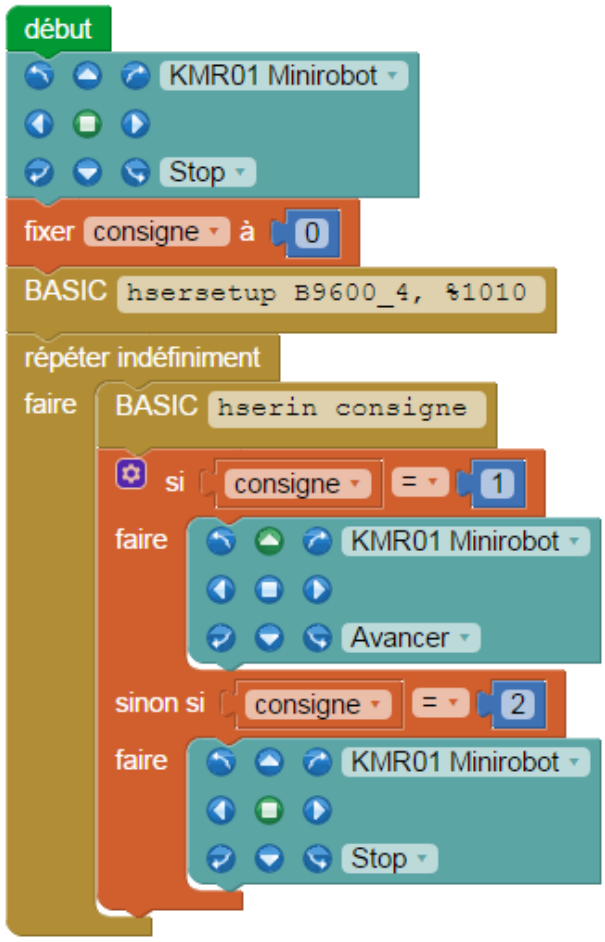
Exercice niveau 1 – F1 : Avancer et s'arrêter

Objectif : Faire s'avancer ou s'arrêter le robot à l'aide de données envoyées via Bluetooth.

Notion(s) abordée(s) : Utilisation du module Bluetooth.

Correction :

Blocs



Fichier Blockly : MR_N1_F1.xml

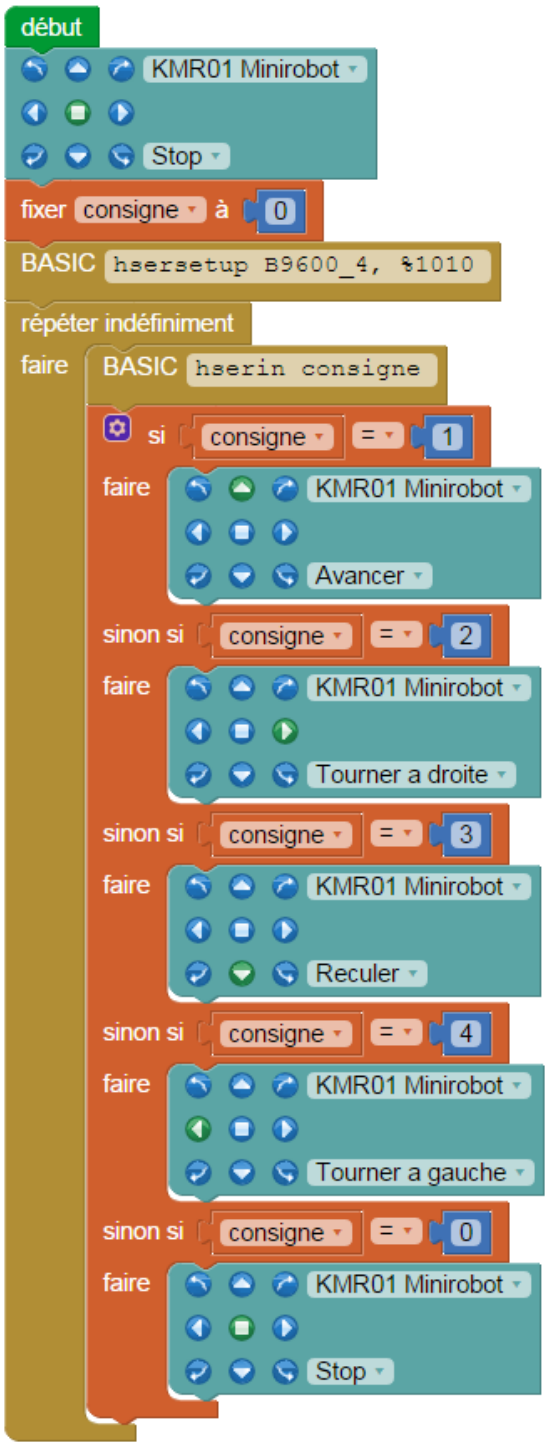
Exercice niveau 1 – F2 : Pilotage via application Bluetooth

Objectif : Contrôler le robot à l'aide de données envoyées via Bluetooth.

Notion(s) abordée(s) : Utilisation du module Bluetooth.

Correction :

Blocs



Fichier Blockly : MR_N1_F2.xml

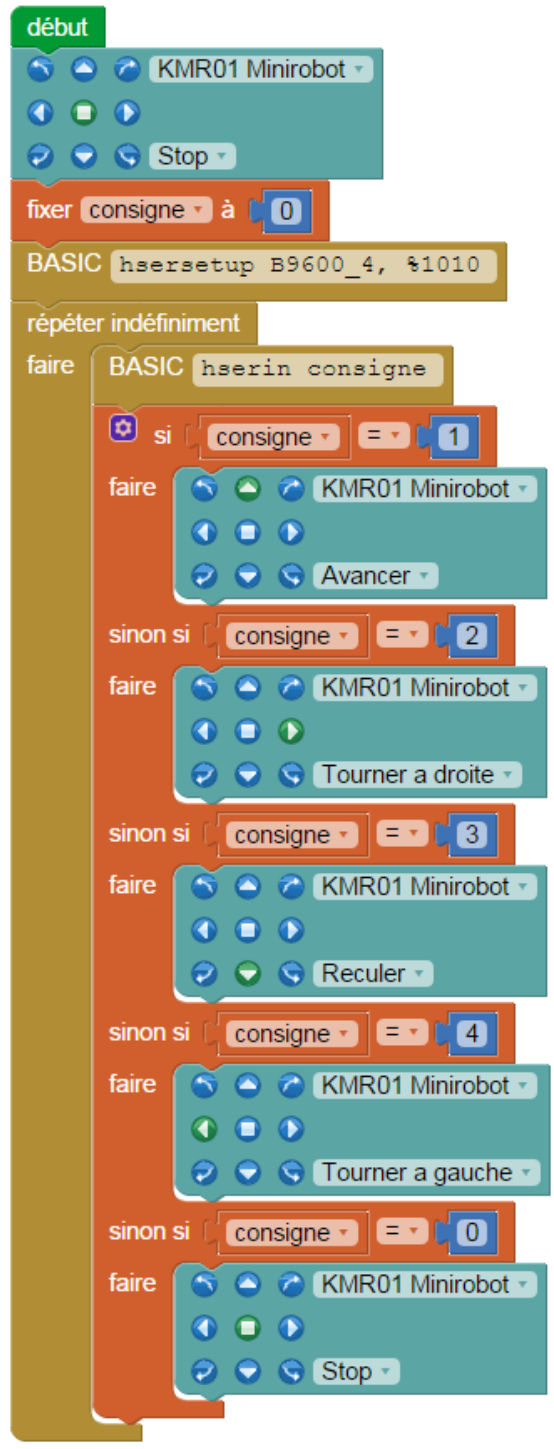
Exercice niveau 1 – F3 : Pilotage via application Bluetooth et accéléromètre

Objectif : Contrôler le robot à l'aide de données envoyées via Bluetooth en fonction de la position de l'appareil (exemple : pencher l'appareil vers l'avant fait avancer le minirobot)

Notion(s) abordée(s) : Utilisation du module Bluetooth

Correction :

Blocs



Fichier Blockly : MR_N1_F3.xml

Programmation niveau 2 (version de base)

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 robot.

Liste des programmes du niveau 2

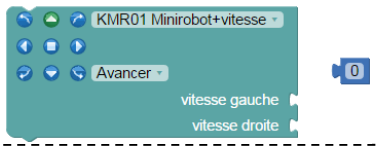
Nom du fichier	Description	Objectif
Niveau 2 A		
MR_N2_A1	Ligne droite	Régler le déplacement en ligne droite.
MR_N2_A2	Eviter	Eviter des obstacles. Compter le nombre de contacts avec les obstacles et s'arrêter dès que 3 obstacles ont été touchés.
MR_N2_A3	Poursuite	Poursuivre un deuxième MiniRobot.
MR_N2_A4	Multi-modes	Sélectionner des modes de fonctionnement différents avec la télécommande.

Exercice niveau 2 – A1 : Ligne droite

Objectif : Changer la vitesse des moteurs afin que celui-ci avance en ligne droite.

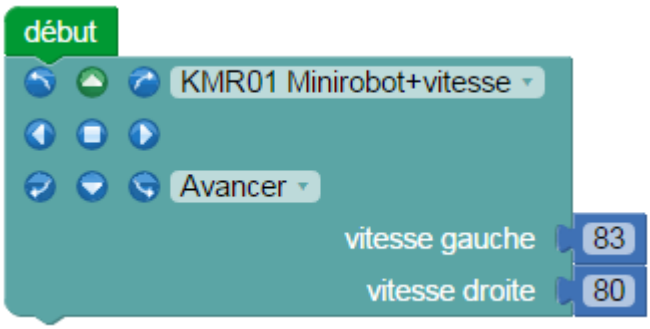
Notion(s) abordée(s) : Vitesse des moteurs

Instruction(s) utilisée(s) :



Correction :

Blocs



Fichier Blockly : MR_N2_A1.xml

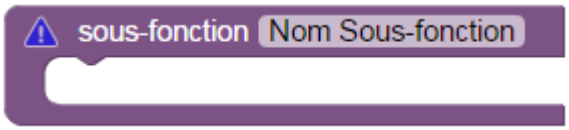
Remarque : les moteurs ont des caractéristiques très proches mais pas strictement identiques.
Malgré un paramétrage de la vitesse avec des valeurs identiques, les moteurs ne tourneront pas forcément à la même vitesse.
Il convient, si nécessaire, d'observer les déviations de MiniRobot lorsque celui-ci est sensé avancer en ligne droite et de compenser le déséquilibre des caractéristiques des moteurs en ajustant les paramètres de vitesse.

Exercice niveau 2 – A2 : Eviter

Objectif : Eviter les obstacles détectés par les microrupteurs. Lorsque 5 obstacles sont détectés via les microrupteurs, arrêter le robot. Utiliser des sous-fonctions pour éviter les obstacles

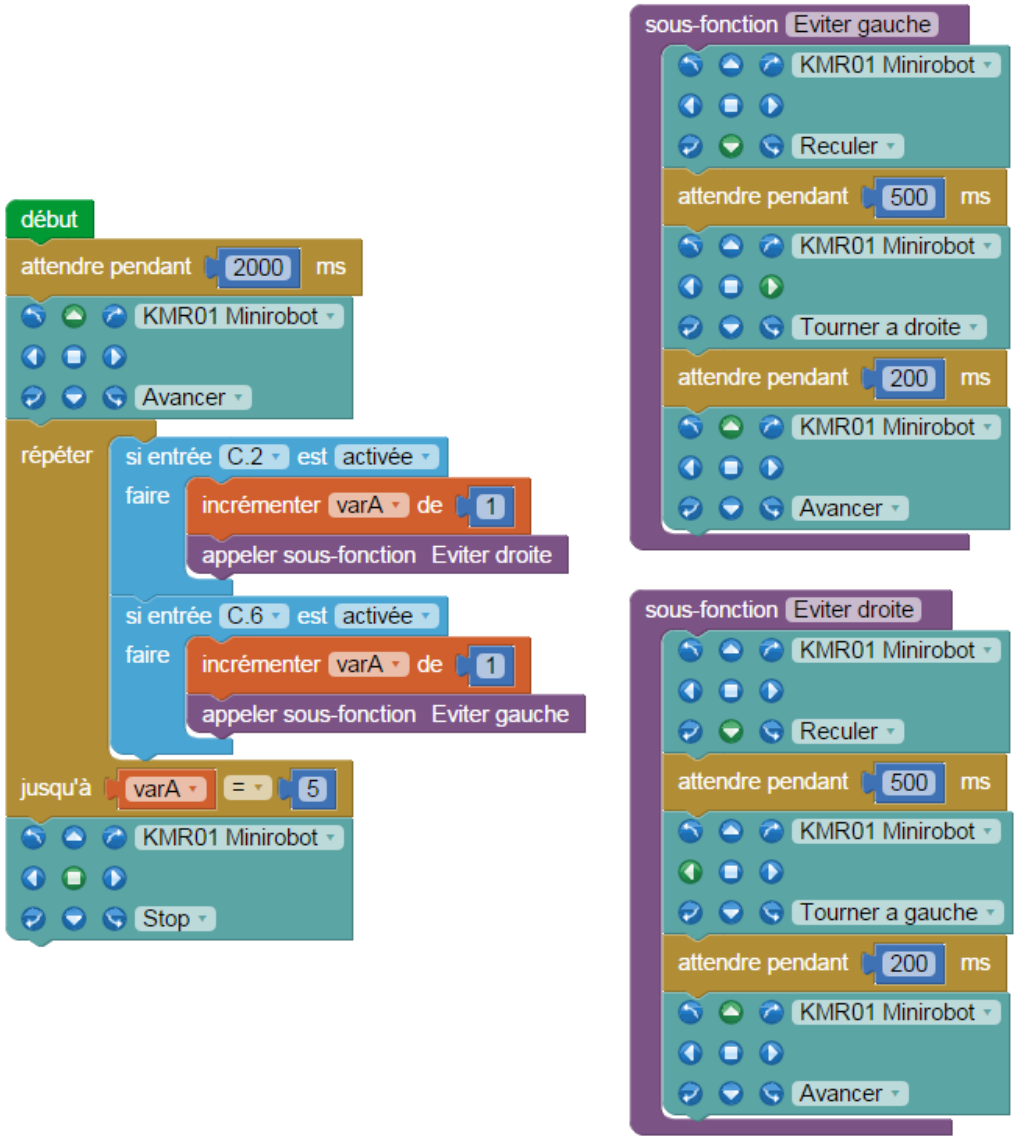
Notion(s) abordée(s) : Sous fonctions

Instruction(s) utilisée(s) :



Correction :

Blocs



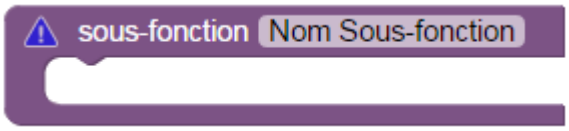
Fichier Blockly : MR_N2_A2.xml

Exercice niveau 2 – A3 : Poursuite

Objectif : Poursuivre un deuxième MiniRobot.
Le premier MiniRobot équipé du module de détection de marquage au sol suit une ligne. Un deuxième MiniRobot équipé uniquement du module à ultrasons le poursuit.

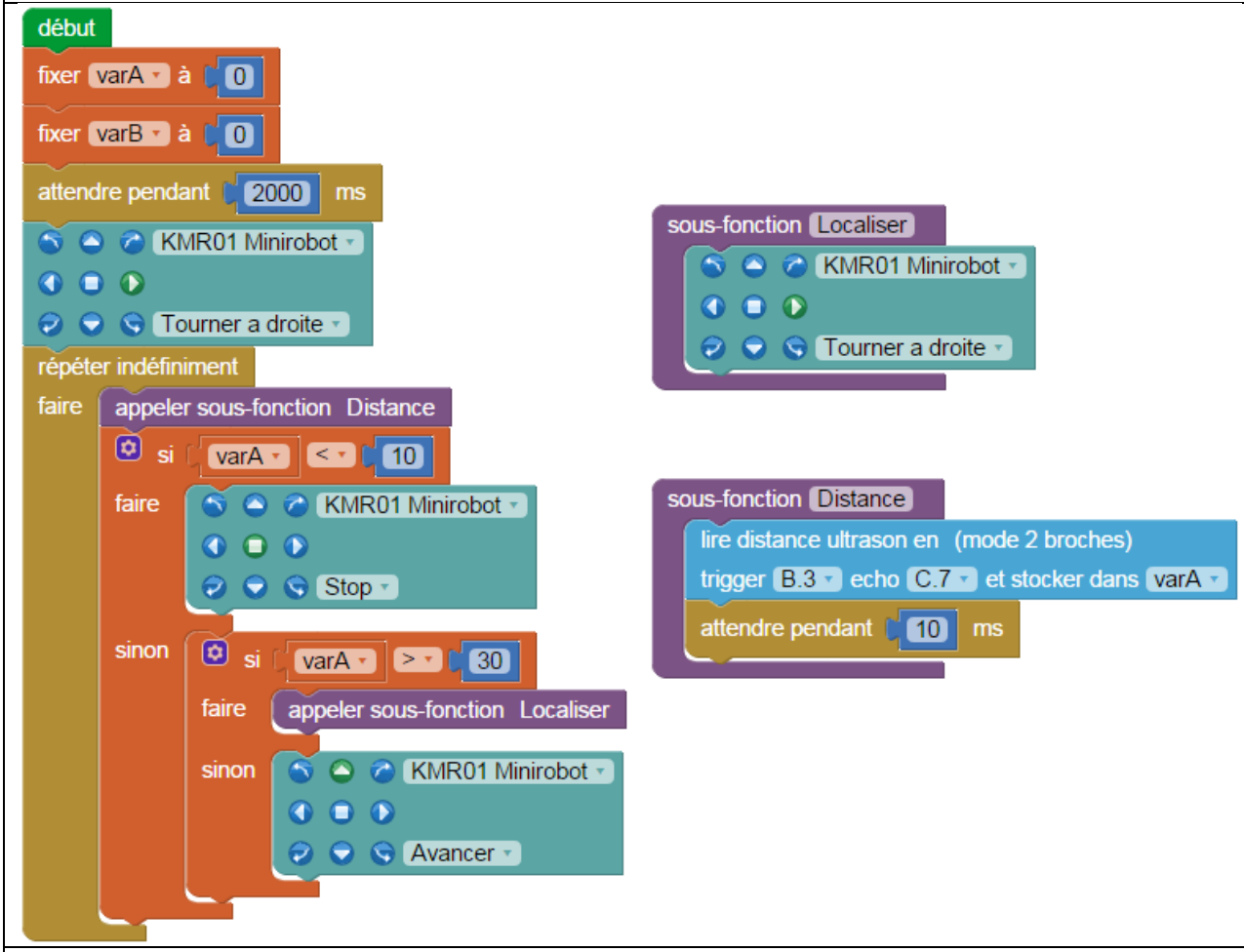
Notion(s) abordée(s) : Sous fonctions

Instruction(s) utilisée(s) :



Correction :

Blocs



Fichier Blockly : MR_N2_A3.xml

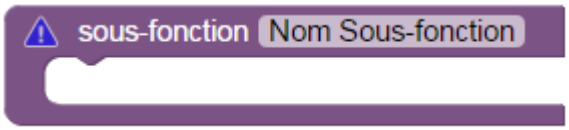
Note : le programme est prévu pour anticiper des virages vers la droite de MiniRobot à poursuivre (sens des aiguilles d'une montre).
On peut adapter le programme pour que la poursuite se fasse avec un MiniRobot qui évolue dans le sens inverse des aiguilles d'une montre en remplaçant l'instruction « Tourner à droite » du sous-programme « Localise » par une instruction « Tourner à gauche ».

Exercice niveau 2 – A4 : Multi-modes

Objectif : Lors d'un appui sur le bouton 1 de la télécommande, le robot doit suivre une ligne, lors de l'appui sur le bouton 2, le robot doit évoluer entre deux lignes. Utiliser des sous-fonctions

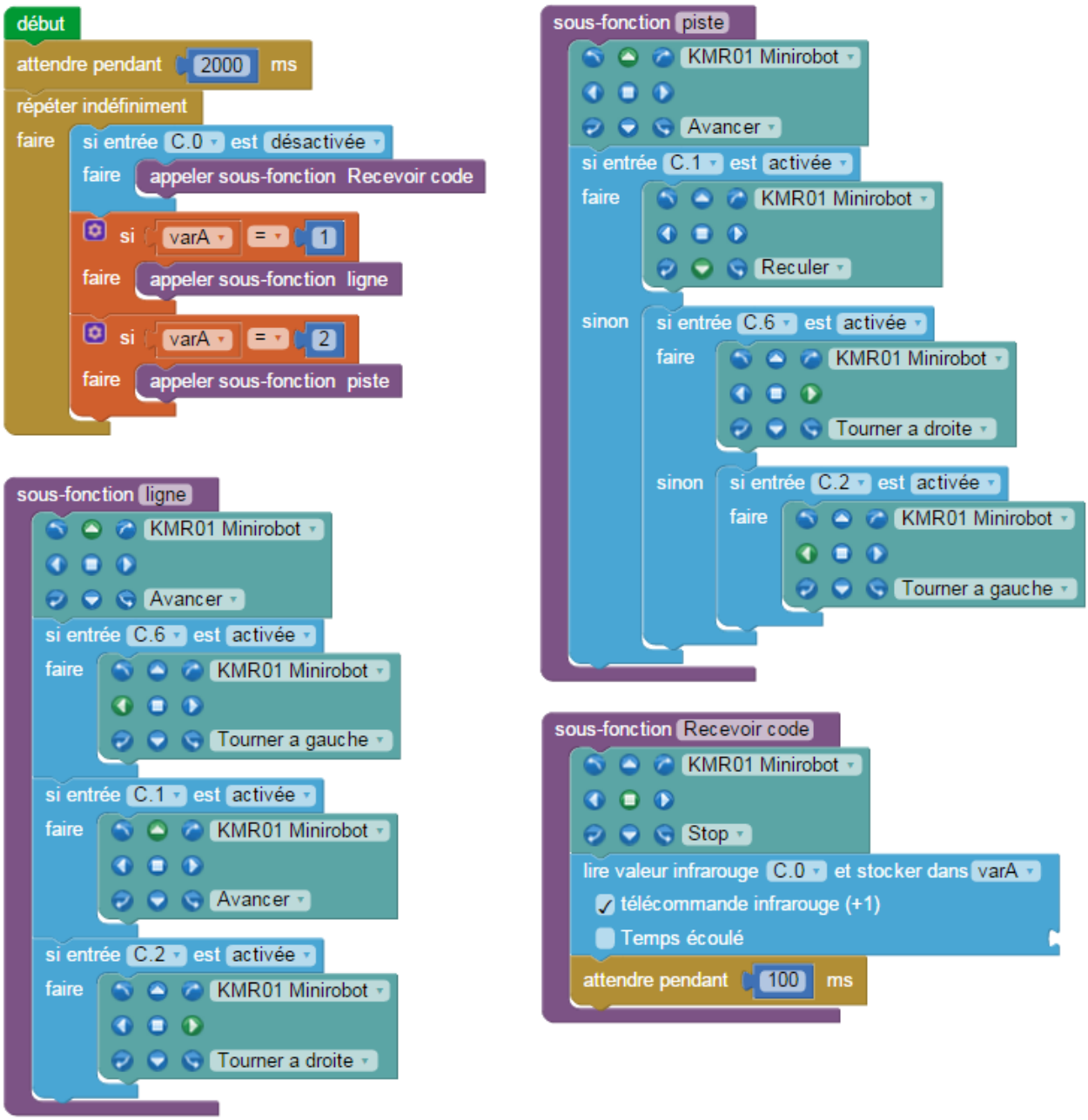
Notion(s) abordée(s) : Sous fonctions

Instruction(s) utilisée(s) :



Correction :

Blocs



Fichier Blockly : MR_N2_A4.xml

Programmation niveau 3

Liste des programmes du niveau 3

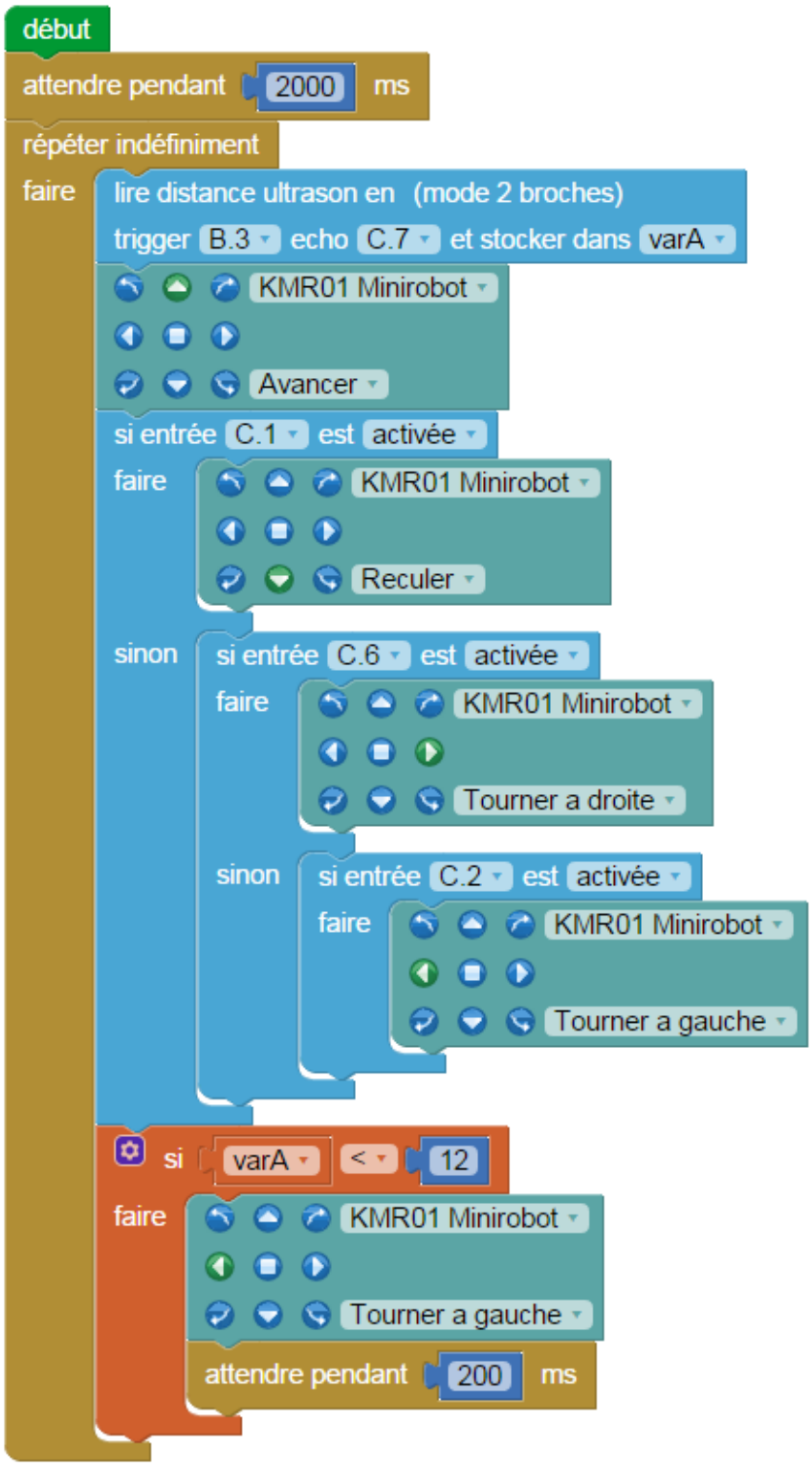
Nom du fichier	Description	Objectif
Niveau 3 A		
MR_N3_A1	Grand Prix	
MR_N3_A2	Epingle à cheveux	
MR_N3_A3	Ejecter des plots	
MR_N3_A4	Labyrinthe	

Exercice niveau 3 – A1 : Grand prix

Objectif : Le robot évolue entre les 2 lignes qui délimitent la piste. S'il détecte la présence d'un autre robot à moins de 12 cm, il tourne à gauche afin d'éviter la collision. Il poursuivra sa route sur la piste dès lors qu'un des bords de la piste sera détecté.

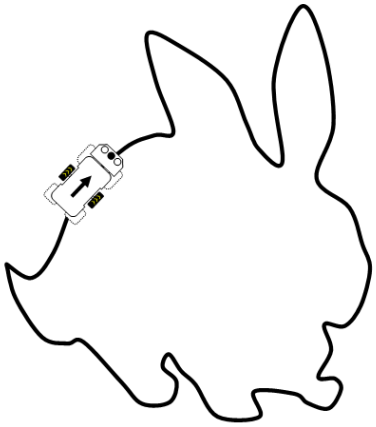
Correction :

Blocs



Fichier Blockly : MR_N3_A1.xml

Exercice niveau 3 – A2 : Epingles à cheveux

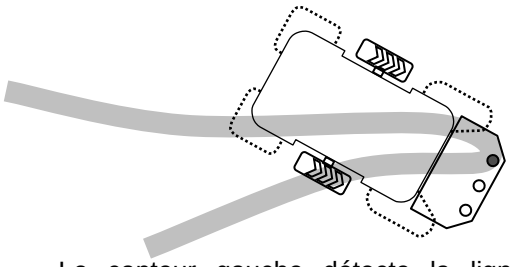


Objectif : Permettre au minirobot d'effectuer des virages en épingles à cheveux

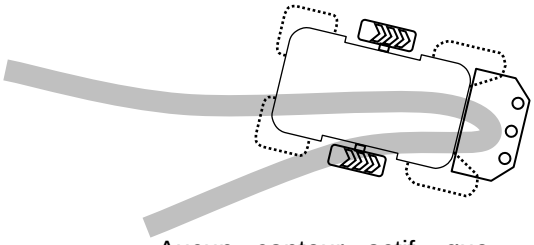
Traitement du cas particulier d'une épingle à cheveux à droite :

La manière classique de traiter le suivi d'une ligne qui tourne à droite consiste à aller tout droit lorsque le capteur central est actif et de tourner à droite dès que le capteur droit devient actif afin de repositionner le capteur central sur la ligne.

Dans le cas particulier d'une épingle à cheveux, ce type de programmation fait qu'il arrive un moment ou aucun capteur ne détecte la ligne ou bien que pendant le virage à droite le capteur gauche détecte la ligne. MiniRobot risque alors de partir dans la nature !

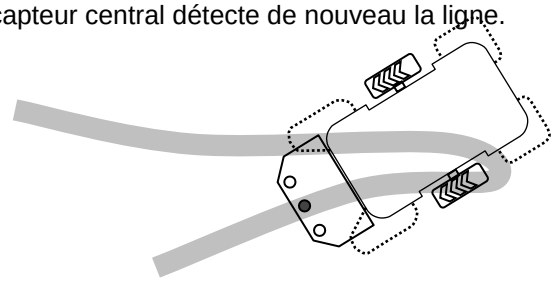


Le capteur gauche détecte la ligne pendant un virage à droite. Si l'on tourne à gauche dans l'espoir de raccrocher la ligne, plus aucun capteur ne la détecte et il risque de partir dans la nature !

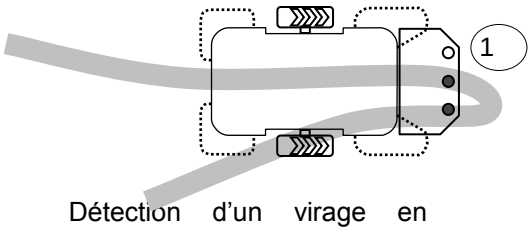


Aucun capteur actif, que doit faire MiniRobot ?

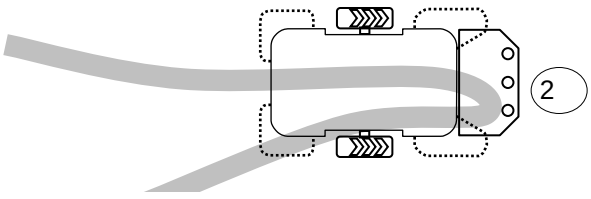
Pour réagir à cette situation particulière, on peut par exemple continuer à tourner à droite jusqu'à ce que le capteur central détecte de nouveau la ligne.



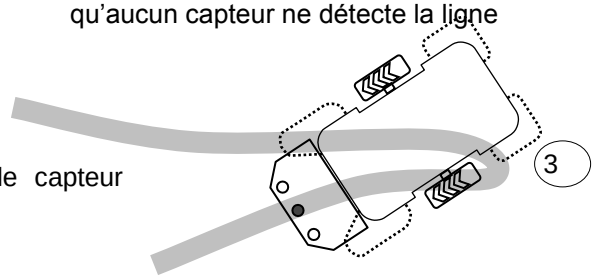
On peut anticiper cette situation particulière en partant du principe que s'il y a un virage brusque à droite, le capteur droit est activé alors même que le capteur central détecte la ligne.



Détection d'un virage en épingle à cheveux à droite.



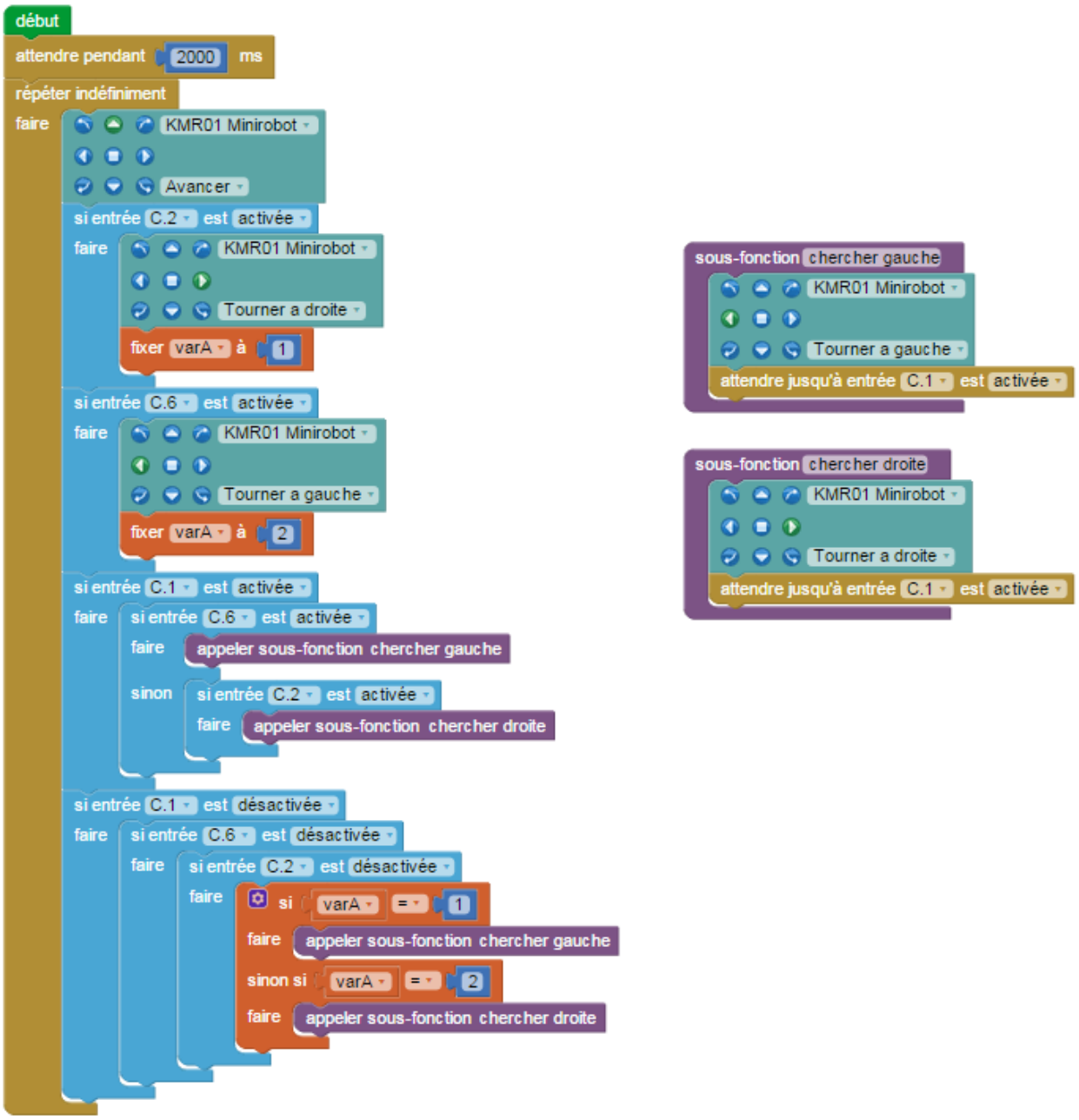
MiniRobot avance jusqu'à ce qu'aucun capteur ne détecte la ligne



MiniRobot tourne jusqu'à ce que le capteur central détecte de nouveau la ligne.

Correction :

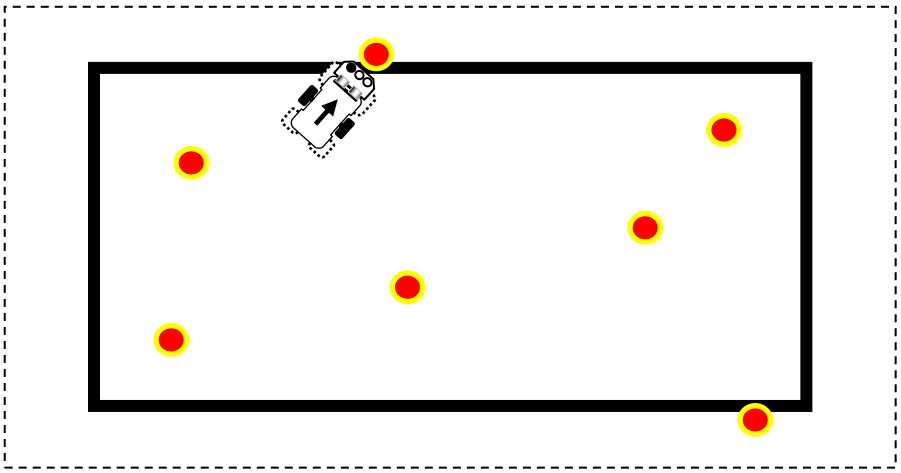
Blocs



Fichier Blockly : MR_N3_A2.xml

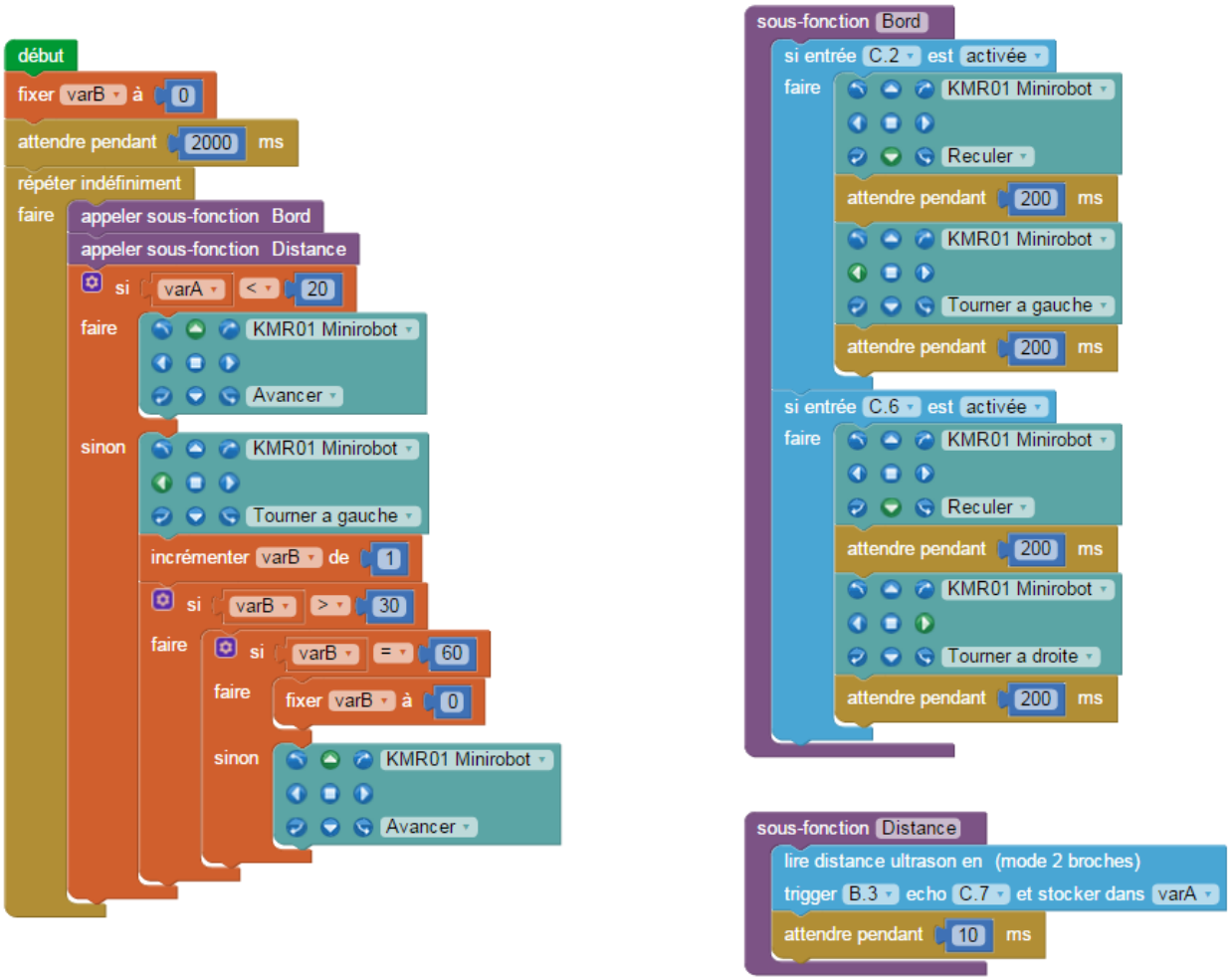
Exercice niveau 3 – A3 : Ejecter des plots

Objectif : Le robot cherche des objets sur un terrain rectangulaire à bords noirs et doit les pousser hors de la zone.
Le robot ne doit pas sortir de la zone



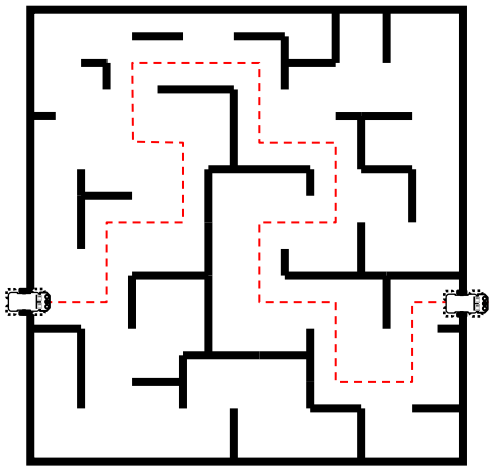
Correction :

Blocs



Fichier Blockly : MR_N3_A3.xml

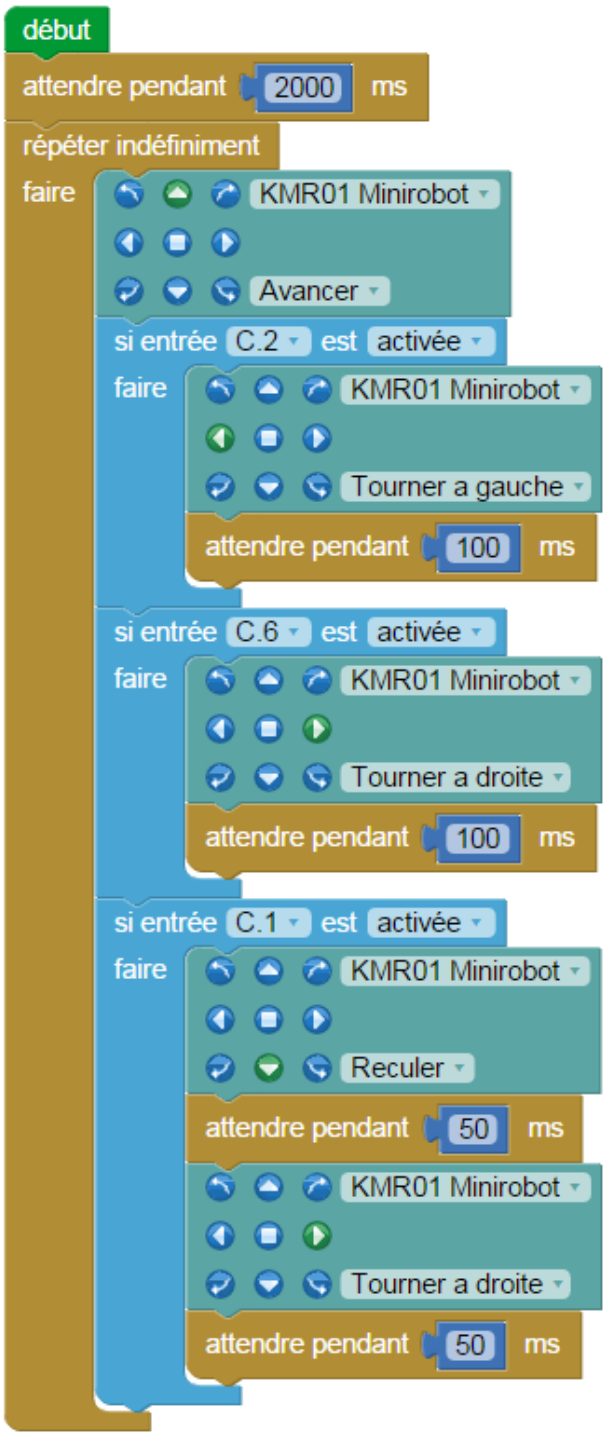
Exercice niveau 3 – A4 : Labyrinthe



Objectif : Le robot doit sortir d'un labyrinthe.

Correction :

Blocs



Fichier Blockly : MR_N3_A4.xml

Remarque : Dans ce programme, le robot cherchera toujours à aller à droite afin de trouver la sortie à coup sûr après un certain temps.

www.a4.fr