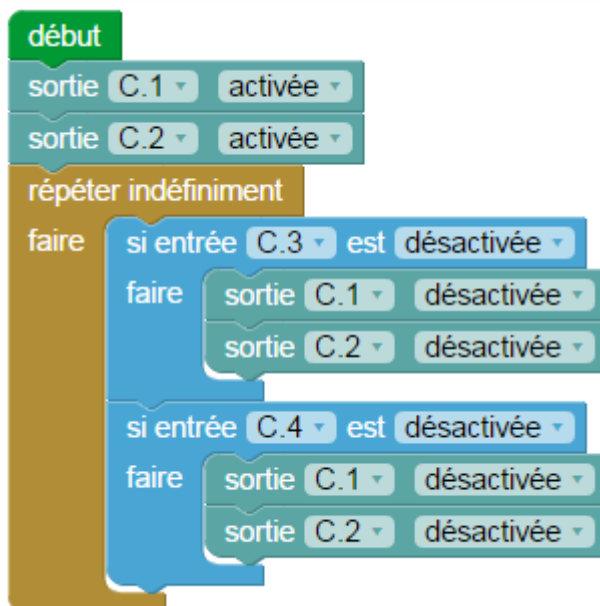


# RoboTribu

## Robots programmables avec Editor 6



# Ressources disponibles pour le projet

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

## Robots Tribu

- Fichiers SolidWorks des robots et des options.

## Logiciel Editor 6

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

## Activités / Programmation

- Fichiers de correction des programmes pour Editor 6.

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



Les documents techniques et pédagogiques signés A4 Technologie sont diffusés librement sous licence Creative Commons BY-NC-SA :

- **BY** : Toujours citer A4 Technologie comme source (paternité).
- **NC** : Aucune utilisation commerciale ne peut être autorisée sans l'accord de A4 Technologie.
- **SA** : La diffusion des documents modifiés ou adaptés doit se faire sous le même régime.

Consulter le site <http://creativecommons.fr/>

# SOMMAIRE

<b>Introduction .....</b>	<b>2</b>
Prérequis .....	2
Editor 6 .....	2
<b>Programmation avec RobotMoto .....</b>	<b>4</b>
Exercice niveau 1 – A1 : Activer un moteur .....	5
Exercice niveau 1 – A2 : Effectuer un virage .....	6
Exercice niveau 1 – A3 : Tourner en rond.....	7
Exercice niveau 1 – A4 : Mouvement répété.....	8
Exercice niveau 1 – A5 : Compteur.....	9
Exercice niveau 1 – A6 : Ligne droite.....	10
Exercice niveau 1 – A7 : Arrêt sur une ligne .....	11
Exercice niveau 1 – A8 : Suivre une ligne.....	12
Exercice niveau 1 – A9 : Suivre une piste.....	13
<b>Programmation avec RobotPilot .....</b>	<b>14</b>
Exercice niveau 1 – B1 : Activer un moteur .....	15
Exercice niveau 1 – B2 : Activer un moteur .....	16
Exercice niveau 1 – B3 : Avancer et s'arrêter au contact d'un obstacle .....	17
Exercice niveau 1 – B4 : Tourner sur un obstacle .....	18
Exercice niveau 1 – B5 : Eviter un obstacle .....	19
<b>Programmation avec RobotServo .....</b>	<b>20</b>
Exercice niveau 1 – C1 : Activer un moteur .....	20
<b>Programmation avec High Power.....</b>	<b>21</b>
Exercice niveau 1 – D1 : Activer un moteur .....	22
Exercice niveau 1 – D2 : Effectuer un virage .....	23
Exercice niveau 1 – D3 : Tourner en rond.....	24
Exercice niveau 1 – D4 : Mouvement répété .....	25
Exercice niveau 1 – D5 : Compteur.....	26
Exercice niveau 1 – D6 : Pare chocs .....	27
Exercice niveau 1 – D7 : Evitement .....	28
Exercice niveau 1 – D8 : Manœuvre d'évitement .....	29
Exercice niveau 1 – D9 : Capteur à ultrasons.....	30

# Introduction

---

RoboTribu est une gamme de robots simples et faciles à réaliser qui permet de se familiariser rapidement avec la programmation des cartes PICAXE.

## Prérequis

- Installer le logiciel **Picaxe Editor 6** : <http://www.picaxe.com/Software>
- **Câble de programmation** Picaxe USB (Réf : CABLE-USBPICAXE).

## Editor 6

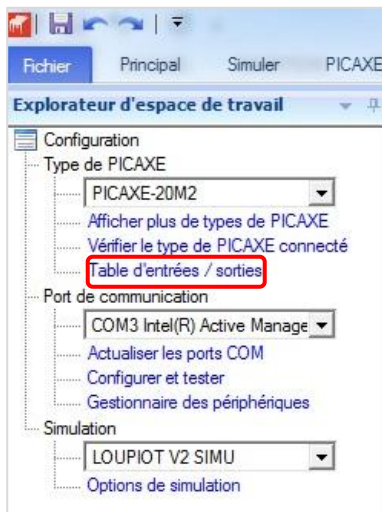
Tous les programmes ont été réalisés sous **Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs **avantages** :

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

## 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 ».

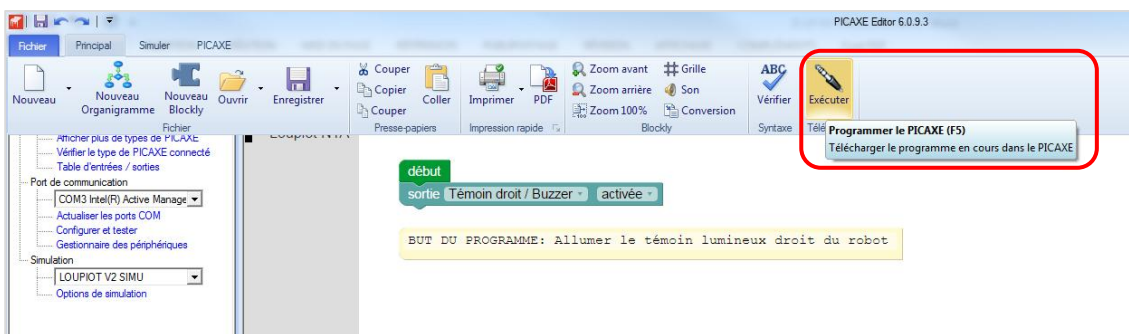
Broche	Mon étiquette
B.0	Témoin droit / Buzzer
B.1	PWM moteur gauche
B.2	Moteur avant gauche
B.3	Moteur arrière gauche
B.4	Témoin alerte
B.5	Lecture batterie

OK Annuler

Valider en cliquant sur **OK**.

## Procédure de chargement d'un programme

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

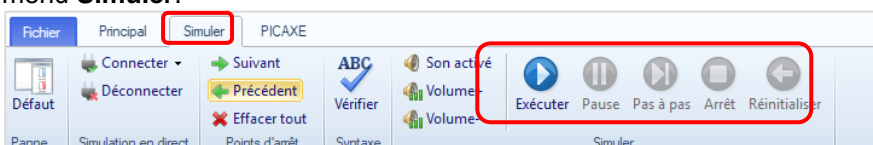


A partir du menu **Principal** ou du menu **PICAXE**, cliquer sur le bouton **Exécuter**. Vous pouvez également utiliser la touche **F5** de votre clavier.

**Note** : un programme téléchargé écrase le précédent.

## Mode simulation

La simulation sur 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 avec RobotMoto

RobotMoto : robot équipé de 2 motoréducteurs PropulsO et d'une carte Picaxe MotoProg avec son option de détection de ligne. 2 capteurs infrarouges pointant vers le sol sous le châssis permettent de détecter une ligne.

La carte MotoProg permet de programmer des mouvements simples (avancer, tourner) pour suivre une ligne ou naviguer sur une piste délimitée par 2 lignes. Ne permet pas la marche arrière.



Nom du fichier	Description
RT_N1-A1	Activer un moteur
RT_N1-A2	Effectuer un virage
RT_N1-A3	Tourner en rond
RT_N1-A4	Mouvement répété
RT_N1-A5	Compteur
RT_N1-A6	Ligne droite
RT_N1-A7	Arrêt sur une ligne
RT_N1-A8	Suivre une ligne
RT_N1-A9	Suivre une piste

## Exercice niveau 1 – A1 : Activer un moteur

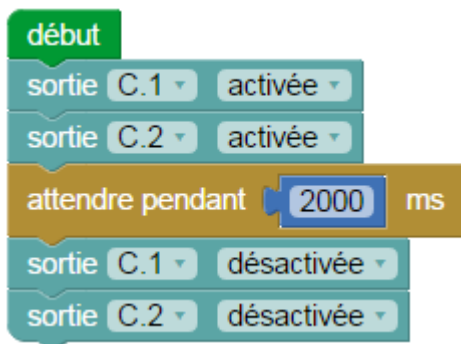
**Objectif :** Avancer 2 secondes puis s'arrêter.

**Notion(s) abordée(s) :** activation d'une sortie et utiliser un temps d'attente.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_A1.xml



**Remarque(s) :**

- C.1 correspond au moteur de droite
- C.2 correspond au moteur de gauche
- Activer une sortie correspond à activer un moteur

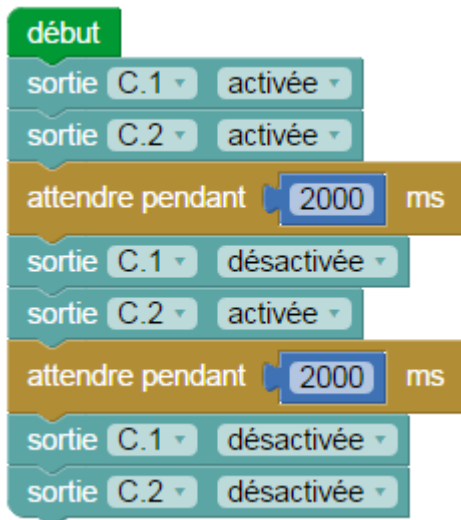
## Exercice niveau 1 – A2 : Effectuer un virage

**Objectif** : Avancer pendant 2 secondes puis tourner pendant 2 secondes et enfin s'arrêter.

**Instruction(s) utilisée(s)** :



**Correction** : RT\_N1\_A2.xml



**Remarque(s)** : N'utiliser qu'un seul moteur pour effectuer un virage



## Exercice niveau 1 – A3 : Tourner en rond

**Objectif :** Tourner à gauche pendant 3 secondes puis tourner à droite pendant 3 secondes puis s'arrêter.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_A3.xml

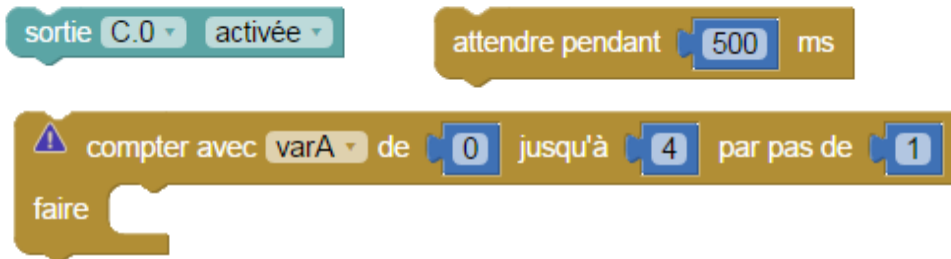


## Exercice niveau 1 – A4 : Mouvement répété

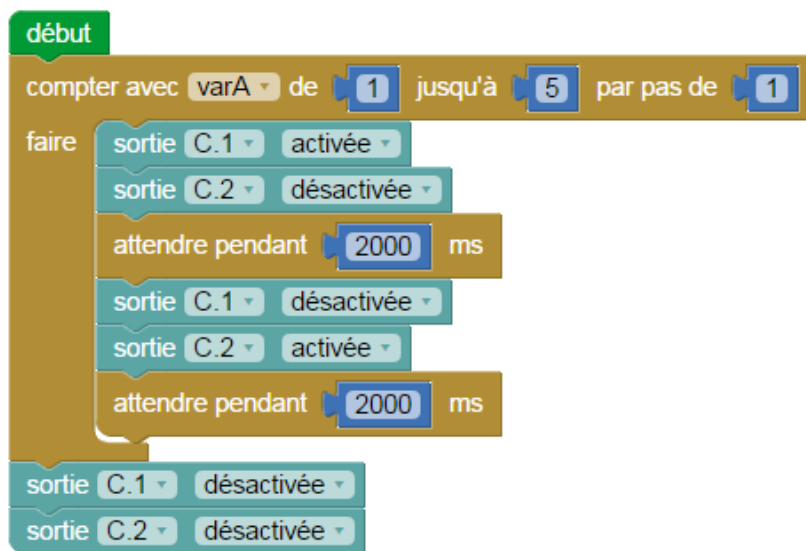
**Objectif** : répéter 5 fois l'action suivante : Tourner dans un sens puis dans l'autre.

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

**Instruction(s) utilisée(s)** :



**Correction** : RT\_N1\_A4.xml



## Exercice niveau 1 – A5 : Compteur

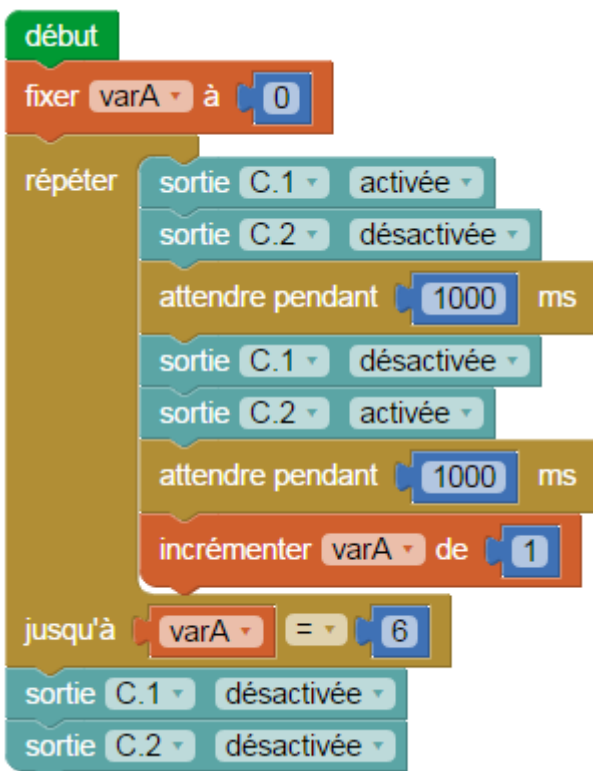
**Objectif :** Répéter une figure 6 fois de suite puis s'arrêter à l'aide d'une variable à incrémenter.

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

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_A5.xml

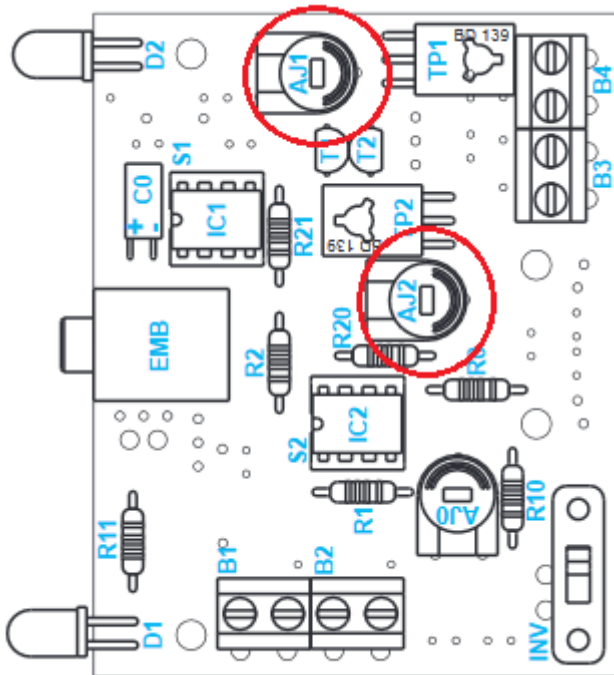


## Exercice niveau 1 – A6 : Ligne droite

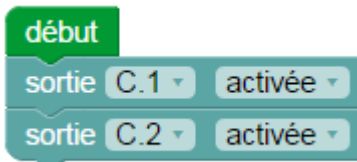
**Objectif :** Avancer en continu et régler le robot pour qu'il se dirige en ligne droite.

Pour cela, tourner les potentiomètres entourés en rouge.

Les potentiomètres permettent de régler la puissance envoyée dans un moteur. Le potentiomètre AJ1 permet de régler le moteur de droite, et AJ2 le moteur de gauche.



**Correction :** RT\_N1\_A6.xml



**Remarque(s) :** ceci est un programme permettant de faire fonctionner les exercices suivants.

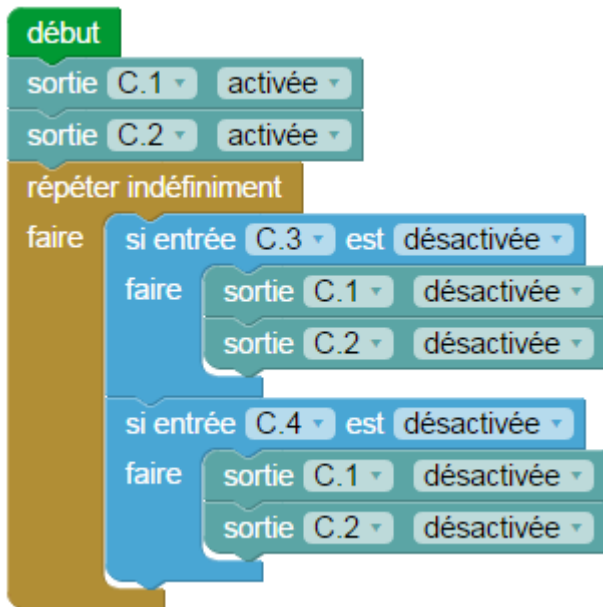


## Exercice niveau 1 – A8 : Suivre une ligne

**Objectif :** Suivre une ligne noire sur une surface.

**Notion(s) abordée(s) :** Utilisation du suiveur de ligne.

**Correction :** RT\_N1\_A8.xml



**Remarque(s) :**

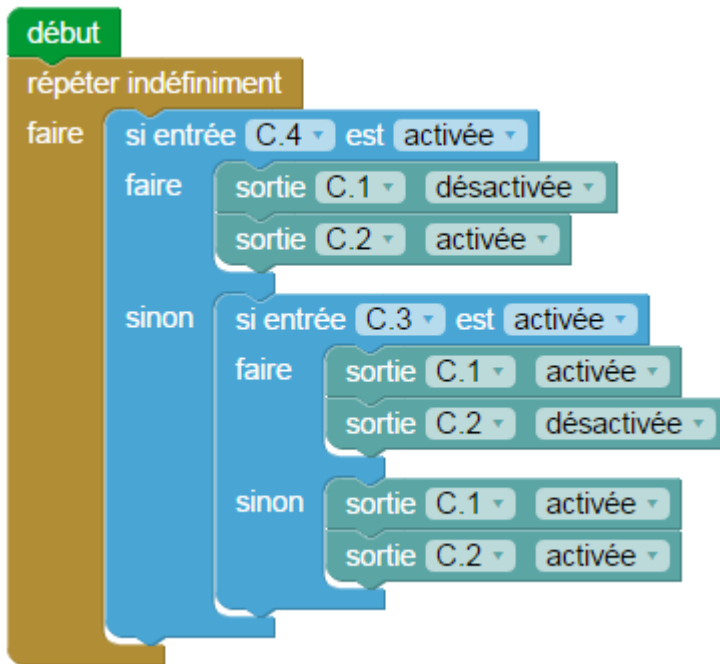
- Le suiveur de ligne se compose de 2 capteurs
- C.3 correspond au capteur de droite
- C.4 correspond au capteur de gauche

## Exercice niveau 1 – A9 : Suivre une piste

**Objectif :** S'arrêter sur une ligne.

**Notion(s) abordée(s) :** Utilisation de capteurs.

**Correction :** RT\_N1\_A9.xml



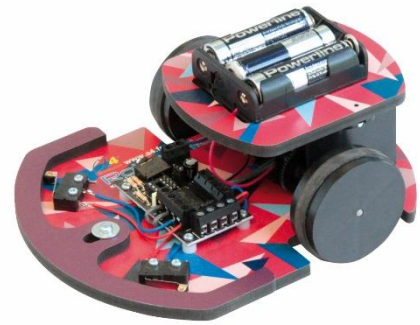
**Remarque(s) :**

- Le suiveur de ligne se compose de 2 capteurs
- C.3 correspond au capteur de droite
- C.4 correspond au capteur de gauche

# Programmation avec RobotPilot

RobotPilot : robot équipé de 2 motoréducteurs PropulsO, d'un pare-chocs / détecteur d'obstacles et d'une carte Picaxe MotoPilot. 2 microinterrupteurs connectés sur la même entrée de la carte permettent de détecter un obstacle (contacts gauche et droit indifférenciés).

Permet de programmer des mouvements simples (avancer, reculer, tourner) et effectuer des manœuvres d'évitement au contact d'un obstacle.



Nom du fichier	Description
RT_N1-B1	Activer un moteur
RT_N1-B2	Activer un moteur
RT_N1-B3	Avancer et s'arrêter au contact d'un obstacle
RT_N1-B4	Tourner sur un obstacle
RT_N1-B5	Eviter un obstacle



## Exercice niveau 1 – B1 : Activer un moteur

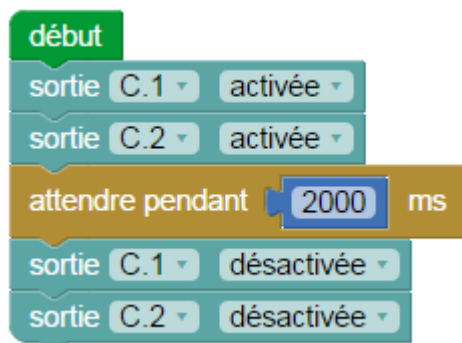
**Objectif** : Avancer 2 secondes puis s'arrêter.

**Notion(s) abordée(s)** : activation d'une sortie et utiliser un temps d'attente.

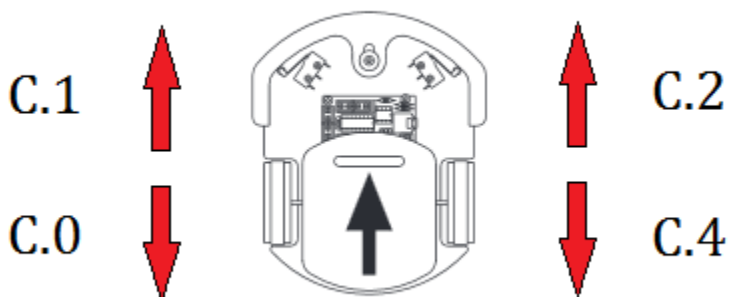
**Instruction(s) utilisée(s)** :



**Correction** : RT\_N1\_B1.xml



**Remarque(s)** : Voici les différentes sorties permettant de contrôler les roues du robot

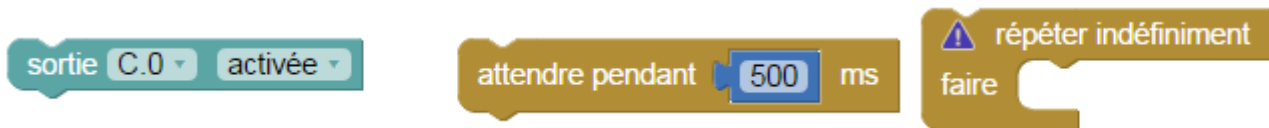


## Exercice niveau 1 – B2 : Activer un moteur

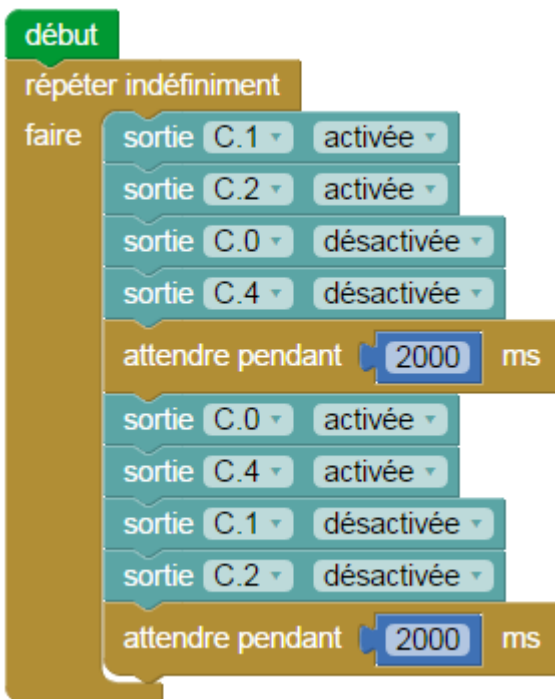
**Objectif :** Avancer 2 secondes puis reculer 2 secondes indéfiniment.

**Notion(s) abordée(s) :** Boucle infinie.

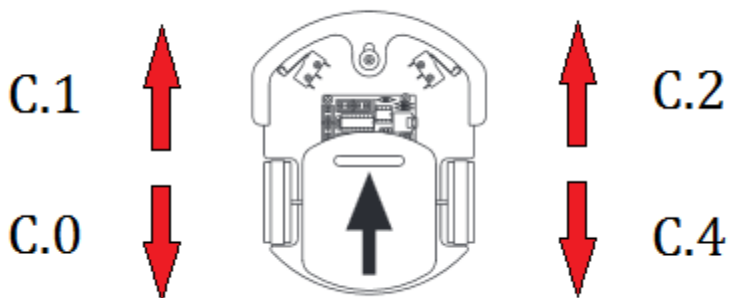
**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_B2.xml



**Remarque(s) :** Voici les différentes sorties permettant de contrôler les roues du robot :

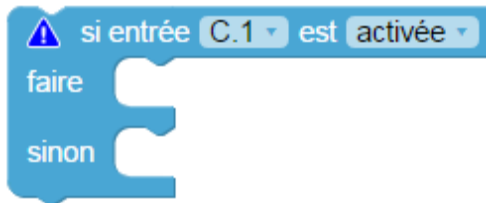


## Exercice niveau 1 – B3 : Avancer et s'arrêter au contact d'un obstacle

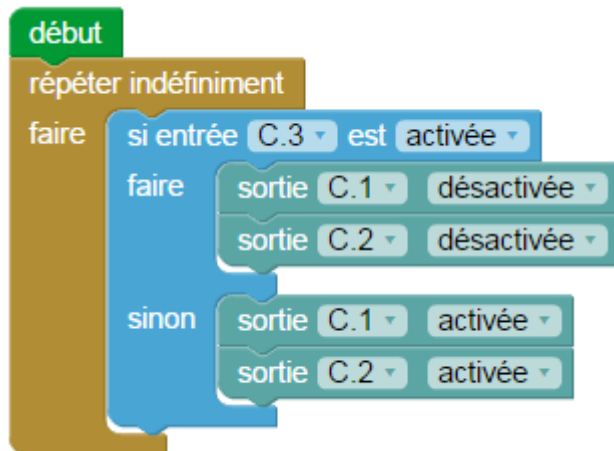
**Objectif :** S'arrêter lorsque qu'un obstacle est détecté par le pare-chocs.

**Notion(s) abordée(s) :** Utiliser une condition dépendant de l'état d'une entrée.

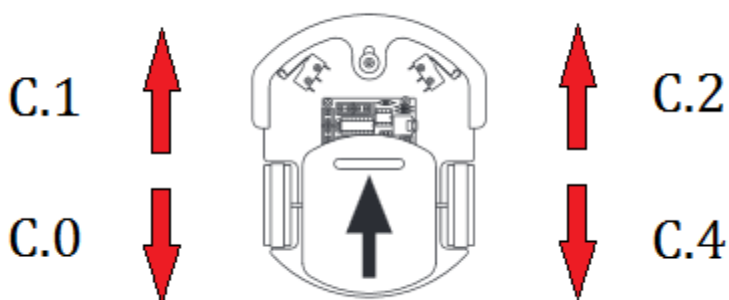
**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_B3.xml



**Remarque(s) :** Voici les différentes sorties permettant de contrôler les roues du robot

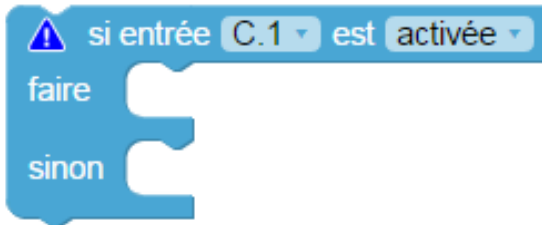


## Exercice niveau 1 – B4 : Tourner sur un obstacle

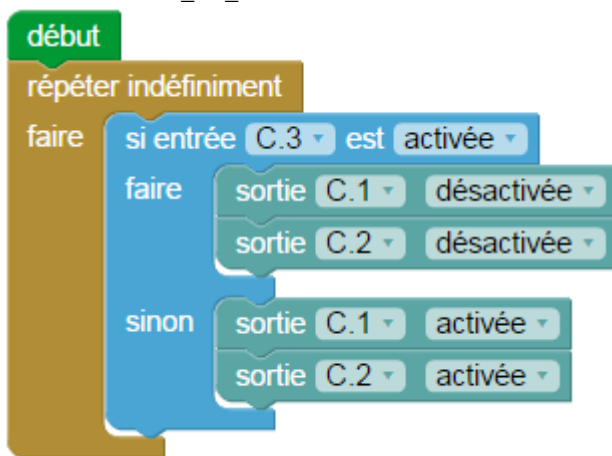
**Objectif :** Tourner à droite lorsqu'un obstacle est détecté par le pare-chocs.

**Notion(s) abordée(s) :** Utiliser une condition dépendant de l'état d'une entrée.

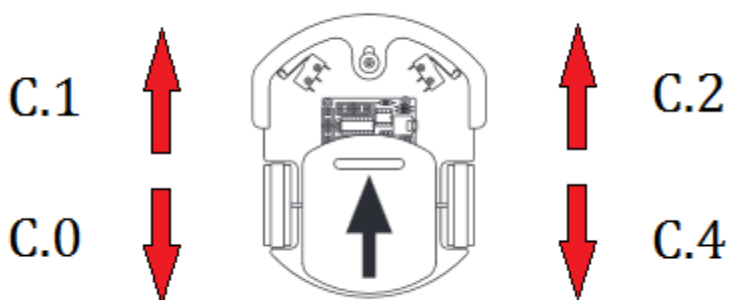
**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_B4.xml



**Remarque(s) :** Voici les différentes sorties permettant de contrôler les roues du robot

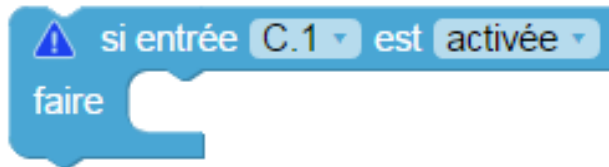


## Exercice niveau 1 – B5 : Eviter un obstacle

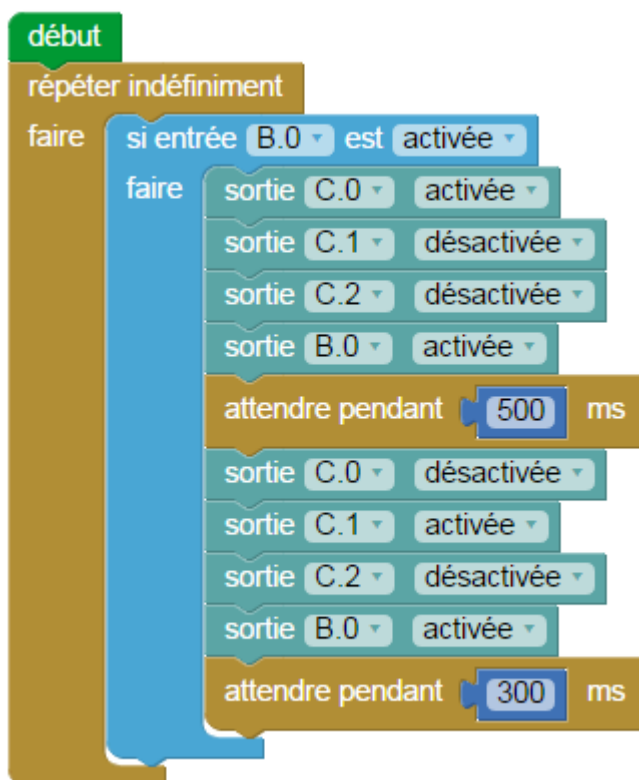
**Objectif :** Lorsqu'un obstacle est détecté par le pare-chocs, reculer puis tourner à droite puis avancer.

**Notion(s) abordée(s) :** Utiliser une condition dépendant de l'état d'une entrée.

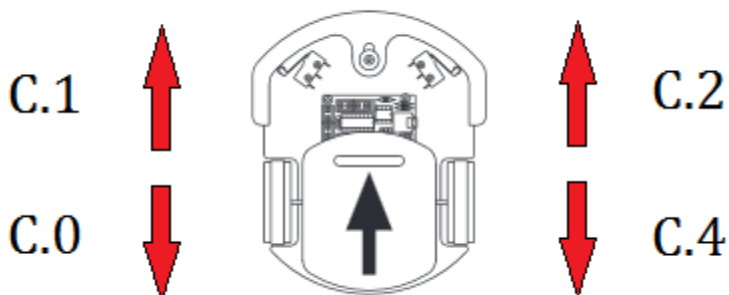
**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_B5.xml



**Remarque(s) :** Voici les différentes sorties permettant de contrôler les roues du robot

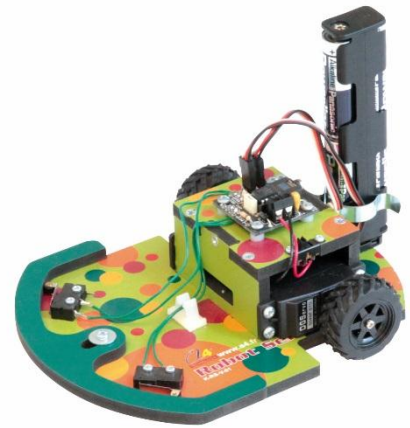


# Programmation avec RobotServo

RobotServo : robot équipé de 2 servomoteurs à rotation continue, d'un pare-chocs / détecteur d'obstacles et d'une carte Picaxe ServoPilot.

2 microrupteurs indépendants permettent de détecter le contact avec un obstacle à gauche ou à droite.

Permet de programmer des mouvements simples (avancer, reculer, tourner) pour effectuer des manœuvres d'évitement.



Les servomoteurs sont pilotables via des fonctions spécifiques sur Blockly.

Le servomoteur est paramétrable avec des valeurs allant de 0 à un maximum.

La valeur 0 donne pour consigne au servomoteur d'aller à la vitesse maximum dans un sens, tandis que la valeur max lui donne pour consigne d'aller, encore à la vitesse maximum, mais dans l'autre sens.

Lorsqu'on se rapproche de la moitié de la valeur max, le servomoteur va ralentir, et vers la moitié, il va s'arrêter.

Nom du fichier	Description
RT_N1-C1	Activer un moteur

## Exercice niveau 1 – C1 : Activer un moteur

**Objectif :** Avancer 2 secondes puis s'arrêter.

**Notion(s) abordée(s) :** activation d'une sortie et utiliser un temps d'attente.

**Instruction(s) utilisée(s) :**

initialiser servo C.0 à

150

positionner servo C.0 à

150

**Correction :** RT\_N1\_C1.xml

début

initialiser servo C.1 à

150

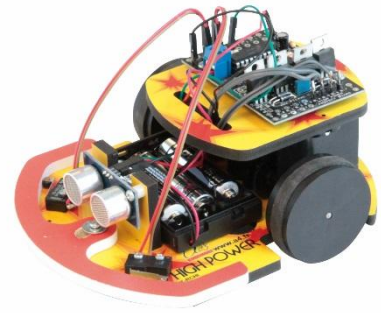
positionner servo C.1 à

200

# Programmation avec High Power

High Power : robot programmable, simple et facile à réaliser, qui permet de se familiariser rapidement avec la programmation de la carte PICAXE High Power.

Il est possible de décliner ce robot dans différentes versions : des options détection de ligne et ultrasons vous sont proposées en kit. Vous pouvez ainsi mettre en place des scénarios des plus simples aux plus élaborés.



Nom du fichier	Description
RT_N1-D1	Activer un moteur
RT_N1-D2	Effectuer un virage
RT_N1-D3	Tourner en rond
RT_N1-D4	Mouvement répété
RT_N1-D5	Compteur
RT_N1-D6	Pare-chocs
RT_N1-D7	Evitement
RT_N1-D8	Manœuvre d'évitement
RT_N1-D9	Capteur à ultrasons

## Exercice niveau 1 – D1 : Activer un moteur

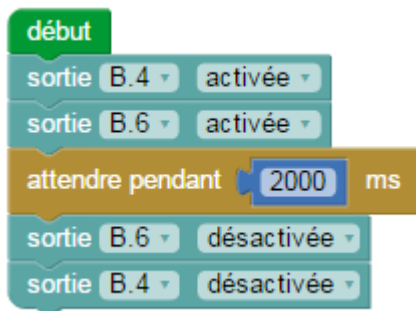
**Objectif :** Avancer 2 secondes puis s'arrêter.

**Notion(s) abordée(s) :** activation d'une sortie et utiliser un temps d'attente.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_D1.xml



**Remarque(s) :**

- B.4 fait avancer la roue de droite
- B.5 fait reculer la roue de droite
- B.6 fait avancer la roue de gauche
- B.7 fait reculer la roue de gauche
- Activer une sortie correspond à activer un moteur



## Exercice niveau 1 – D2 : Effectuer un virage

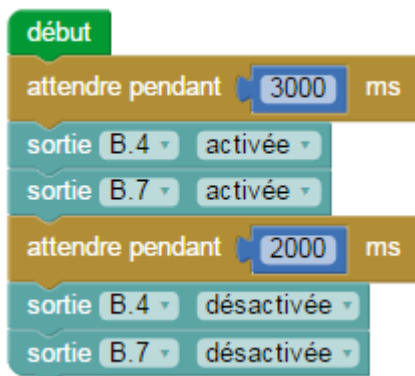
**Objectif :** Tourner 2 secondes puis s'arrêter.

**Notion(s) abordée(s) :** activation d'une sortie et utiliser un temps d'attente.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_D2.xml



**Remarque(s) :** pour tourner, il suffit d'activer un moteur dans un sens et l'autre moteur dans l'autre sens

## Exercice niveau 1 – D3 : Tourner en rond

**Objectif :** Tourner à gauche pendant 3 secondes puis à droite pendant 3 secondes et enfin s'arrêter.

**Notion(s) abordée(s) :** activation d'une sortie et utiliser un temps d'attente.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_D3.xml

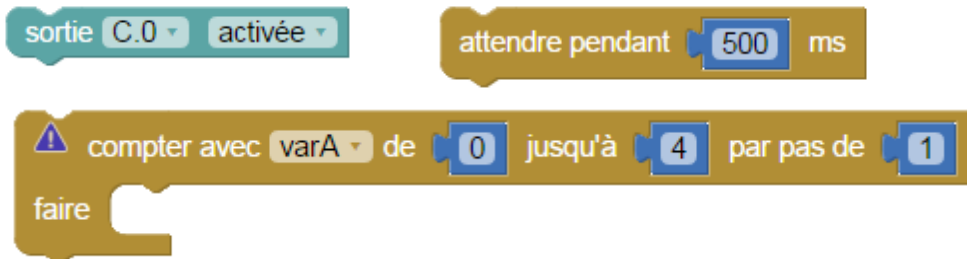


## Exercice niveau 1 – D4 : Mouvement répété

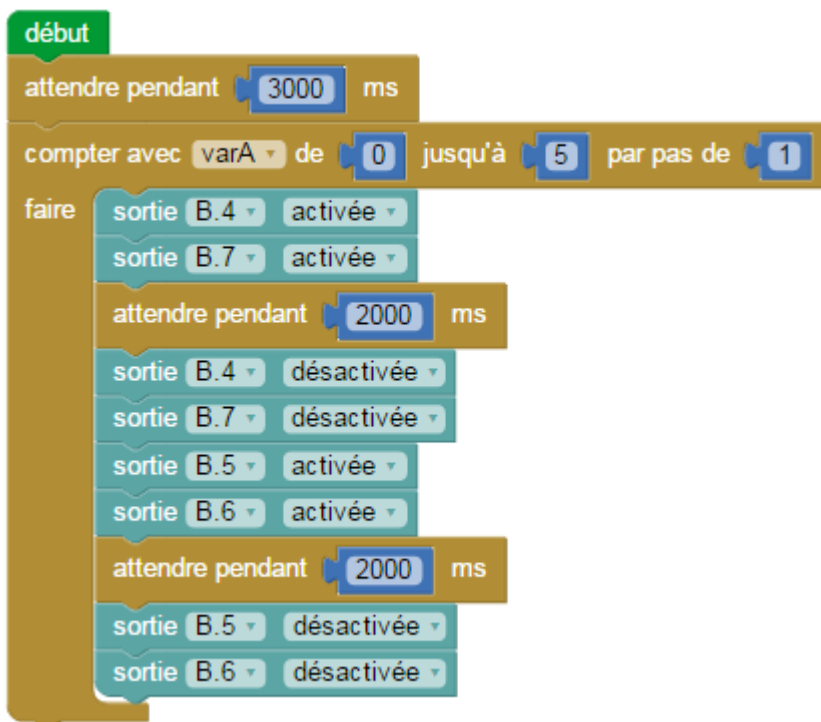
**Objectif** : répéter 5 fois l'action suivante : Tourner dans un sens puis dans l'autre.

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

**Instruction(s) utilisée(s)** :



**Correction** : RT\_N1\_D4.xml

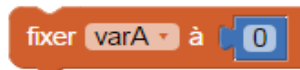


## Exercice niveau 1 – D5 : Compteur

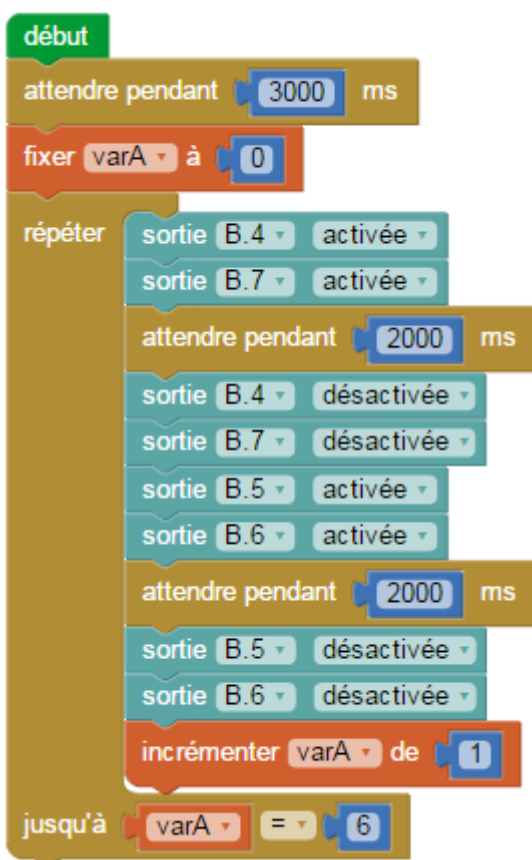
**Objectif :** Répéter une figure 6 fois de suite puis s'arrêter à l'aide d'une variable à incrémenter.

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

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_D5.xml

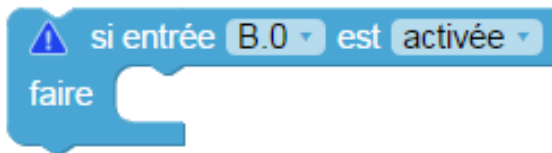


## Exercice niveau 1 – D6 : Pare chocs

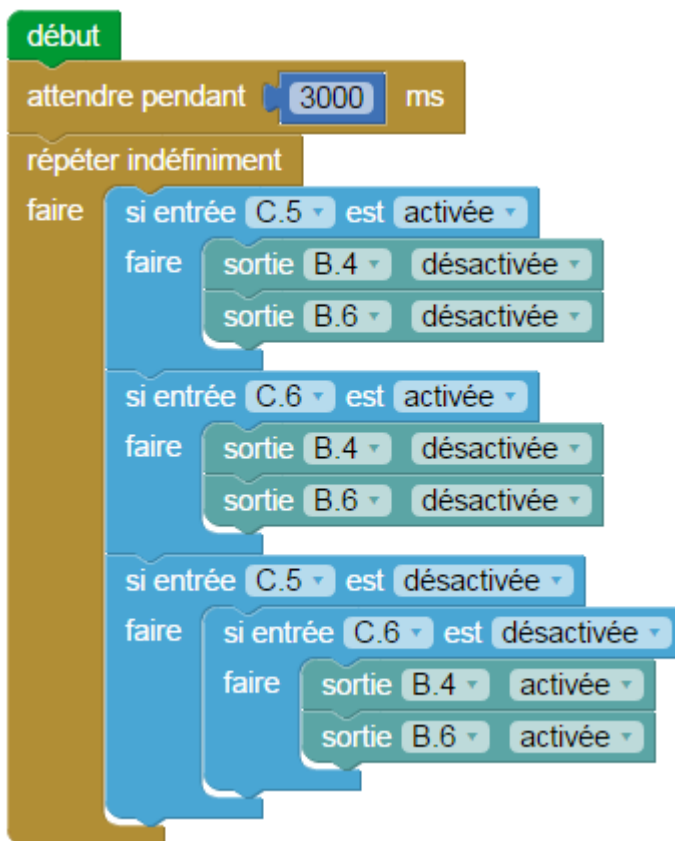
**Objectif :** Avancer et s'arrêter au contact d'un obstacle.

**Notion(s) abordée(s) :** Utilisation de l'instruction conditionnelle si.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_D6.xml

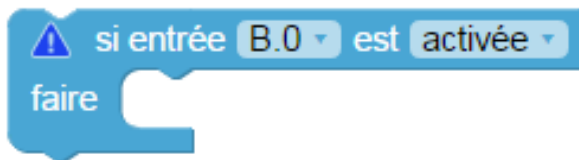


## Exercice niveau 1 – D7 : Evitement

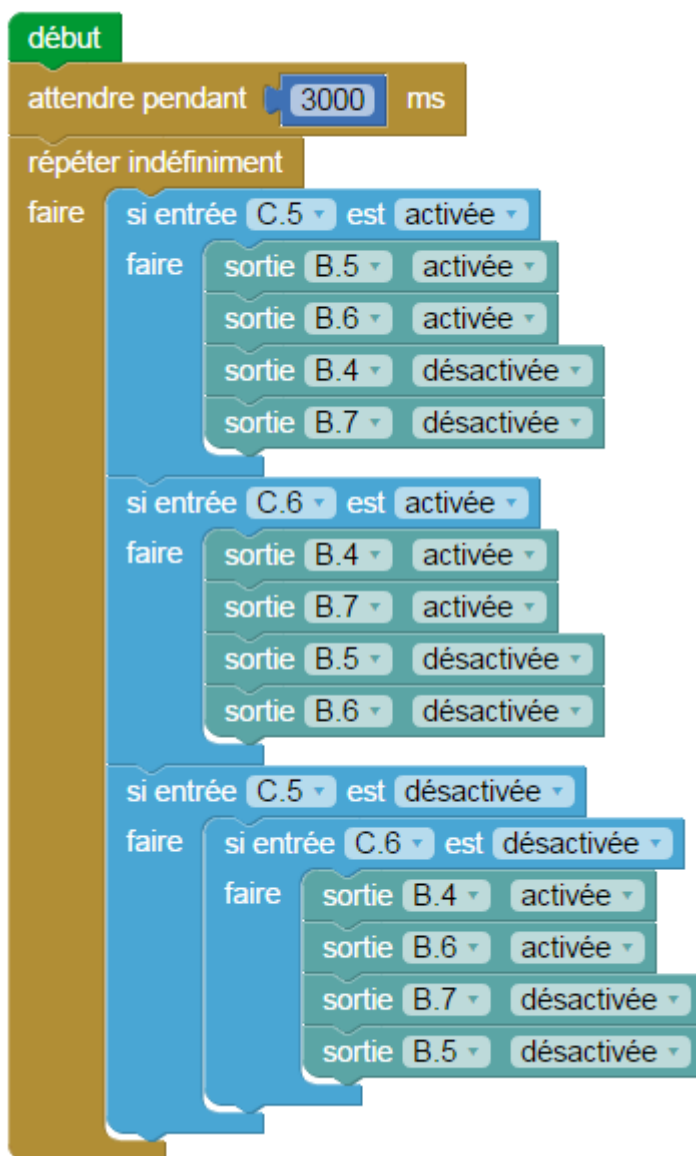
**Objectif :** Avancer et tourner au contact d'un obstacle.

**Notion(s) abordée(s) :** Utilisation de l'instruction conditionnelle si.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_D7.xml

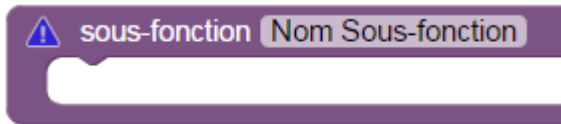


## Exercice niveau 1 – D8 : Manœuvre d'évitement

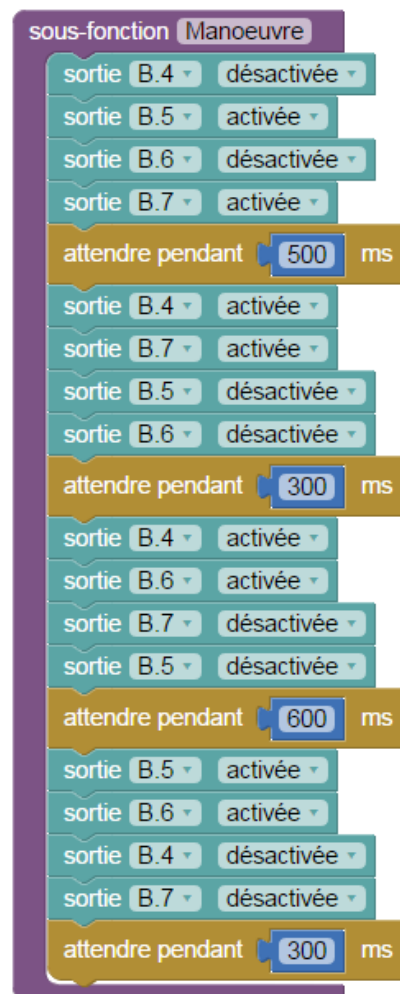
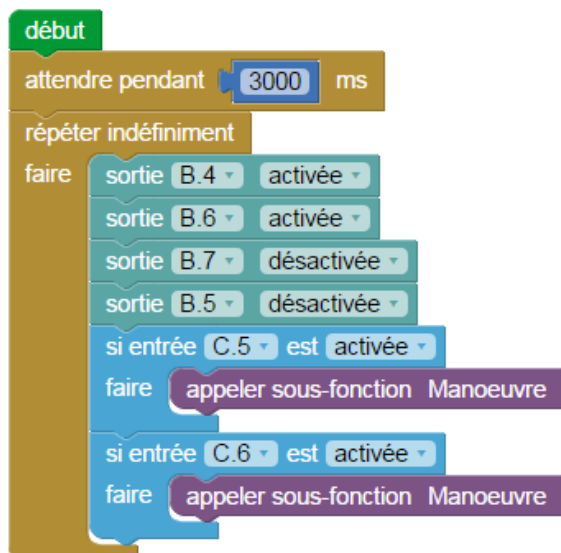
**Objectif :** Programmer une manœuvre d'évitement dans une sous fonction (Exemple : Reculer, tourner à gauche, avancer et tourner à droite.)

**Notion(s) abordée(s) :** Utilisation de sous fonctions.

**Instruction(s) utilisée(s) :**



**Correction :** RT\_N1\_A8.xml



## Exercice niveau 1 – D9 : Capteur à ultrasons

**Objectif :** Arrêter le robot à 8 cm ou moins d'un obstacle.

**Notion(s) abordée(s) :** Utilisation d'un capteur à ultrasons.

**Instruction(s) utilisée(s) :**

lire distance ultrason en B.0 et stocker dans varA

**Correction :** RT\_N1\_D9.xml

