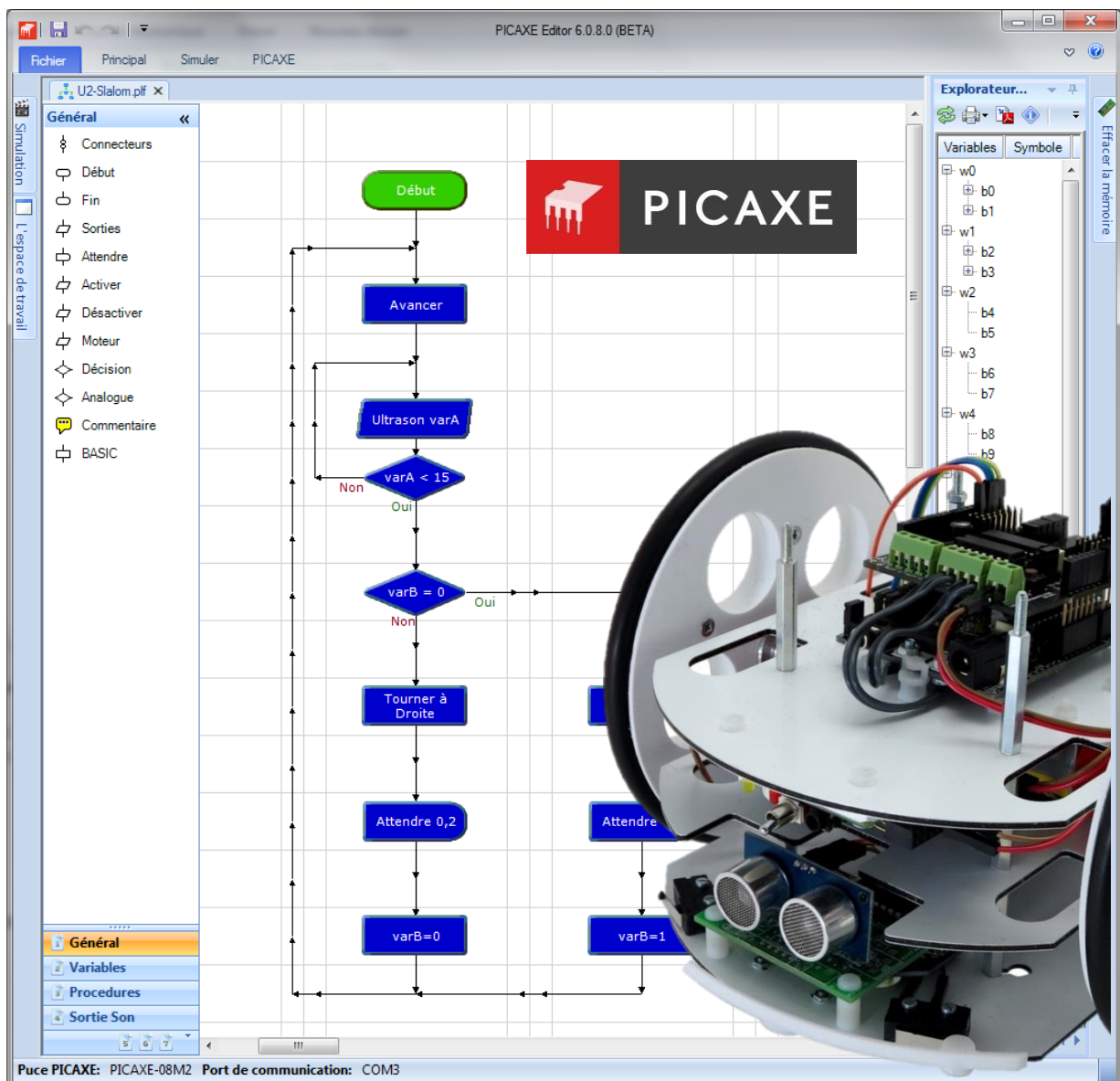


# RoboCoda

## Plateforme robotique modulaire

### Programmation avec l'interface de développement PICAXE Editor 6





PICAXE est une marque de la Sté Revolution Education.

AutoProg est un système développé par la Sté A4, qui utilise des microcontrôleurs PICAXE.



L'ensemble des ressources numériques disponibles autour de nos projets et maquettes sont téléchargeables librement et gratuitement sur [www.a4.fr](http://www.a4.fr)

La duplication de ce dossier est autorisée sans limite de quantité au sein des établissements scolaires, aux seules fins pédagogiques, à la condition que soit cité le nom de l'éditeur : Sté A4.

La copie ou la diffusion par quelque moyen que ce soit à des fins commerciales n'est pas autorisée sans l'accord de la Sté A4.



Edité par la société A4 Technologie  
 Tél. : 01 64 86 41 00 - Fax : 01 64 46 31 19  
[www.a4.fr](http://www.a4.fr)

## SOMMAIRE

<b>1. Introduction.....</b>	<b>2</b>
1.1. Prérequis souhaitables .....	2
1.2. Organisation de ce dossier .....	2
<b>2. Eléments nécessaires .....</b>	<b>3</b>
2.1. Matériels .....	3
2.2. Logiciels.....	3
2.3. Ressources complémentaires .....	3
<b>3. Tableau d'affectation des Entrées/Sorties Arduino.....</b>	<b>4</b>
<b>4. Mise en service des applications.....</b>	<b>5</b>
<b>5. Fiche technique N°1 – F1-Avancer .....</b>	<b>7</b>
<b>6. Fiche technique N°2 – F2-Gauche .....</b>	<b>9</b>
<b>7. Fiche technique N°3 – F3-Accélération .....</b>	<b>11</b>
<b>8. Fiche technique N°4 – F4-Compteur.....</b>	<b>13</b>
<b>9. Fiche technique N°5 – M1-Arret .....</b>	<b>15</b>
<b>10. Fiche technique N°6 – M2-Changement_direction .....</b>	<b>17</b>
<b>11. Fiche technique N°7 – M3-Dégagement .....</b>	<b>19</b>
<b>12. Fiche technique N°8 – DL1-Stop.....</b>	<b>21</b>
<b>13. Fiche technique N°9 – DL2-Suivi_de_ligne .....</b>	<b>23</b>
<b>14. Fiche technique N°10 – DL3-Piste .....</b>	<b>25</b>
<b>15. Fiche technique N°11 – DL4-Perimetre.....</b>	<b>27</b>
<b>16. Fiche technique N°12 – U1-Proximite.....</b>	<b>29</b>
<b>17. Fiche technique N°13 – U2-Slalom .....</b>	<b>31</b>
<b>18. Fiche technique N°14 – U3-Cible .....</b>	<b>33</b>

# 1. Introduction

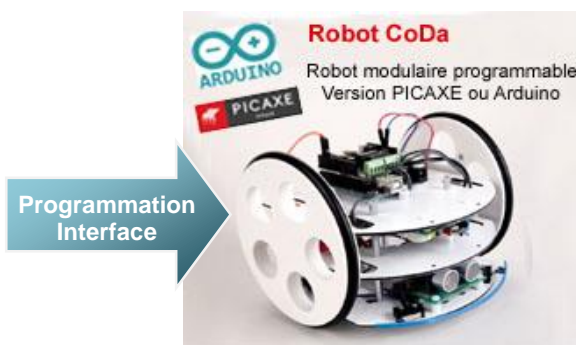
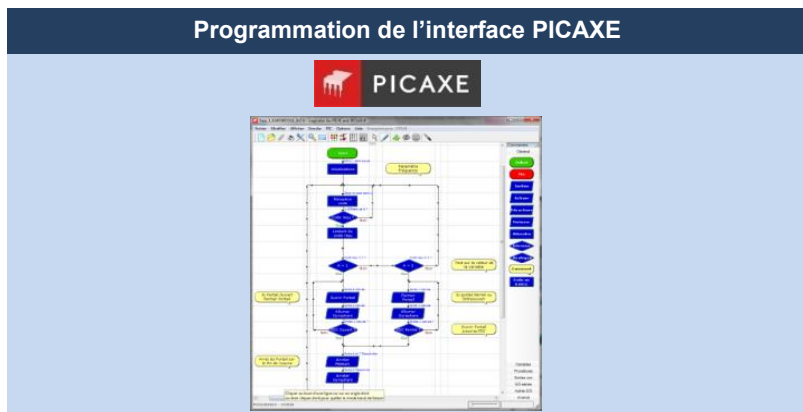
Ce document illustre l'utilisation de la version de base du **robot CoDa** d'A4.

Les applications proposées permettent de programmer et de piloter le robot dans son environnement. Toutes les options y sont traitées ainsi que des fiches pour accompagner l'élève durant tout le projet d'initiation.

Les programmes sont développés avec les IDE\* :

- **PICAXE Editor 6** pour les applications robotique ;

\* IDE : *Environnement de Développement Intégré*



## 1.1. Prérequis souhaitables

La mise en œuvre des applications suppose que l'utilisateur ait des notions de base autour des logiciels et matériels utilisés. Il est utile de maîtriser les outils de programmations de bases avant de s'initier avec les programmes du robot CoDa

## 1.2. Organisation de ce dossier

Les applications se présentent sous forme de fiches classées par ordre de difficulté croissante.

Elles permettent de découvrir la manière d'établir une communication bidirectionnelle entre le smartphone et l'interface de pilotage du robot.

Des propositions de modifications sont suggérées à la fin de chaque fiche.

## 2. Éléments nécessaires

### 2.1. Matériels

- **Robot CoDa** monté avec la carte **PICAXE AXE401**
- **Câble de programmation. USB type B** pour la programmation de la carte Arduino

### 2.2. Logiciels

IDE PICAXE Logicator ou PICAXE Editor6 (<http://www.picaxe.com/Software>) pour la programmation de l'interface AutoProg.

\* IDE : Environnement de Développement Intégré

### 2.3. Ressources complémentaires

Des ressources complémentaires sont disponibles sur [www.a4.fr](http://www.a4.fr)

- Dossier Technique Robot CoDa
- Dossier App Inventor 2 pour prendre en main AppInventor 2
- Guide d'utilisation PICAXE Logicator

D'autres ressources sont également disponibles sur internet. Vous pouvez entrer les mots clés suivants pour les localiser avec un moteur de recherche :

Tuto app inventor 2  
Tuto Picaxe  
Tuto Arduino Uno

Carte moteur Axe 408

Robot CoDa est un robot muni de deux roues activées par deux motoréducteurs.

Il est piloté par une carte programmable Arduino auquel on peut adjoindre différents capteurs.

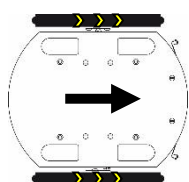
Le module de pilotage dispose de :

- 16 entrées pour des capteurs ;
- 4 sorties spéciales qui s'ajoutent à celles utilisées pour la commande des moteurs.

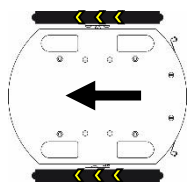
#### Déplacements :

Robot CoDa peut se déplacer dans toutes les directions :

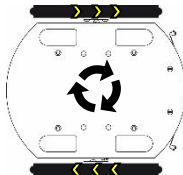
#### MARCHE AVANT



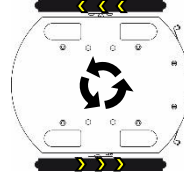
#### MARCHE ARRIERE



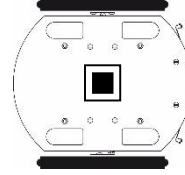
#### TOURNER A DROITE



#### TOURNER A GAUCHE



#### ARRÊT



Il est possible de programmer le robot pour qu'il tourne juste autour d'un seul axe (une seule roue).

Ces instructions sont modélisées par un robot avec une simple flèche en virage. La courbe et le sens de la flèche indiquera la manœuvre qu'effectuera le robot.

### 3. Tableau d'affectation des Entrées/Sorties Arduino

Ci-dessus le tableau résumant la connexion entre les broches de la carte Shield AXE408 et la carte de commande PICAXE AXE401. Ce tableau sert principalement à faire le lien entre le câblage des capteurs/actionneurs et la programmation du microcontrôleur.


N° Broche AXE 408 (Shield moteur)	MODULE CAPTEUR/ACTIONNEURS	N° Broche PICAXE (Carte de commande)
0 (RX)	Broche libre (réservé option Bluetooth)	C.7
1 (TX)	Broche libre (réservé option Bluetooth)	C.6
2	Détecteur de ligne Droit	B.2
3	Détecteur de ligne Central	B.0
4	Détecteur de ligne Gauche	B.1
5	Télémètre à Ultrasons	B.5
6	Broche libre	B.6
7	Broche libre	B.7
8	Direction moteur droit	C.0
9	Puissance moteur droit	C.1
10	Puissance moteur gauche	C.2
11	Direction moteur gauche	C.5
12	Broche libre	C.4
13	Broche libre	C.3
A0	Broche libre	A.0
A1	Broche libre	A.1
A2	Broche libre	A.2
A3	Broche libre	A.3
A4	Microrupteur gauche	B.3
A5	Microrupteur droit	B.4

## 4. Mise en service des applications

Les applications proposées dans les fiches suivantes sont créées avec différents IDE (Environnement de Développement Intégré).


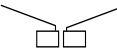


Pour chaque application, un ensemble de fichiers à charger dans l'interface programmable PICAXE est proposé.

Les éléments clés sont mis en évidence et des suggestions de modification sont proposées en vue d'adapter les programmes à un nouveau contexte d'utilisation.





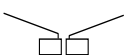
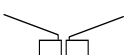
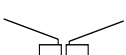







	<b>PICAXE Logicator</b> <a href="http://www.picaxe.com/Software/PICAXE/Logicator-for-PICAXE/">http://www.picaxe.com/Software/PICAXE/Logicator-for-PICAXE/</a> ou <b>PICAXE Editor6</b> <a href="http://www.picaxe.com/Software/PICAXE/PICAXE-Editor-6/">http://www.picaxe.com/Software/PICAXE/PICAXE-Editor-6/</a> Cet IDE est gratuit. Il doit être installé sur un PC.
<b>Matériel associé</b>	<ul style="list-style-type: none"><li>- Interface programmable AutoProg ou autre carte PICAXE compatible.</li><li>- Cordon de programmation PICAXE USB pour transférer les programmes dans l'interface AutoProg.</li></ul>
<b>Fichiers</b>	Fichiers <b>.PLF</b> à ouvrir dans l'IDE PICAXE et à transférer dans l'interface.
<b>Notes</b>	<a href="http://www.a4.fr">www.a4.fr</a> propose de nombreuses ressources autour de PICAXE ; notamment le Guide d'utilisation de Logicator et les dossiers AutoProg Il existe d'autres IDE compatibles PICAXE (par exemple Scratch).

### Liste des symboles utilisés pour les capteurs:

(Voir le chapitre Montage / Assemblage pour la mise en œuvre des capteurs)

Symbole	Description
	Aucun capteur
	Module capteurs microrupteurs (détection de contact avec un obstacle).
	Module capteurs infrarouges (détection de marquage au sol).
	Module capteur à ultrasons (détection d'obstacle à distance).

## Liste des programmes

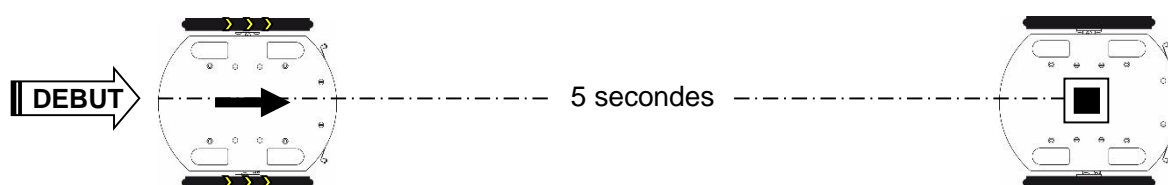
Fiche	Capteur	Description	Fichier programme
N°1		<b>Manœuvre de base :</b> Avancer puis s'arrêter.	<a href="#">F1-AVANCER.plf</a>
N°2		<b>Manœuvre de base :</b> Avancer, puis tourner à gauche.	<a href="#">F2-GAUCHE.plf</a>
N°3		<b>Manœuvre de base :</b> Avancer puis accélérer.	<a href="#">F3-ACCELERATION.plf</a>
N°4		<b>Compter un nombre de séquences :</b> Répéter une figure 5 fois de suite puis s'arrêter.	<a href="#">F4-COMPTEUR.plf</a>
N°5		<b>Utilisation capteurs microrupteurs :</b> S'arrêter au contact d'un obstacle.	<a href="#">M1-ARRET.plf</a>
N°6		<b>Utilisation capteurs microrupteurs :</b> Changer de direction au contact d'un obstacle.	<a href="#">M2-CHANGE_DIRECTION.plf</a>
N°7		<b>Utilisation capteurs microrupteurs :</b> Faire une manœuvre de dégagement au contact d'un obstacle.	<a href="#">M3-DEGAGEMENT.plf</a>
N°8		<b>Utilisation capteurs de détection de marquage au sol :</b> S'arrêter sur un marquage au sol.	<a href="#">DL1-STOP.plf</a>
N°9		<b>Utilisation capteurs de détection de marquage au sol :</b> Suivre une ligne marquée au sol.	<a href="#">DL2-SUIVI LIGNE.plf</a>
N°10		<b>Utilisation capteurs de détection de marquage au sol :</b> Evoluer entre deux lignes marquées au sol.	<a href="#">DL3-PISTE.plf</a>
N°11		<b>Utilisation capteurs de détection de marquage au sol :</b> Evoluer dans un périmètre marqué au sol.	<a href="#">DL4-PERIMETRE.plf</a>
N°12		<b>Utilisation capteur détection d'obstacle :</b> S'arrêter à proximité d'un obstacle.	<a href="#">U1-PROXIMITE.plf</a>
N°13		<b>Utilisation capteur détection d'obstacle :</b> Evoluer en évitant des obstacles.	<a href="#">U2-SLALOM.plf</a>
N°14		<b>Utilisation capteur détection d'obstacle :</b> Localiser un obstacle et se diriger vers lui.	<a href="#">U3-CIBLE.plf</a>



## 5. Fiche technique N°1 – F1-Avancer

But de l'application	Se déplacer en avant pendant 5 secondes puis s'arrêter.
Notions de programmation abordées	Pilotage des moteurs du robot. Prise en main de la conduite du robot CoDa.
Programme associé	Code source Arduino Uno <b>F1-Avancer.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Aucun

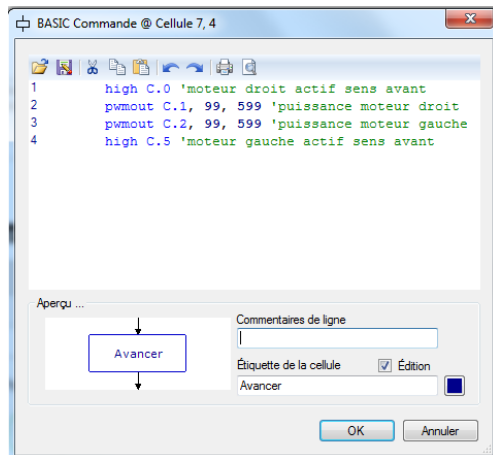
Illustration



Suggestions de modifications	Modifier le programme pour que le robot s'arrête au bout de 10 secondes.
------------------------------	--------------------------------------------------------------------------



Les déplacements du Robot CoDa sont réalisés avec des blocs BASIC pour faciliter la mise en œuvre du programme. Pour piloter un moteur, une consigne de direction et une consigne de vitesse lui sont envoyées.



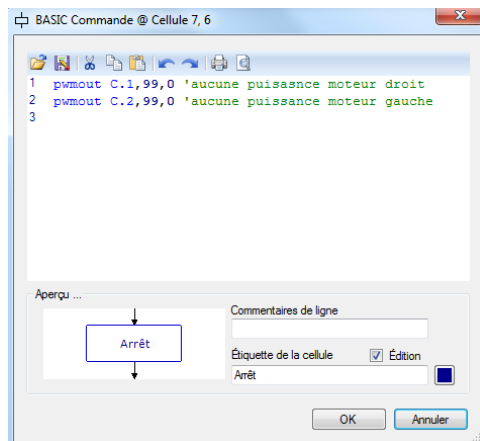
La commande **Avancer** est constitué de 4 instructions :

La sortie **C.0** sert à déterminer le sens de rotation du moteur **droit** (**high** pour avancer, **low** pour reculer).

La sortie **C.1** est la consigne de vitesse du moteur **droit**.

La sortie **C.2** est la consigne de vitesse du moteur **gauche**.

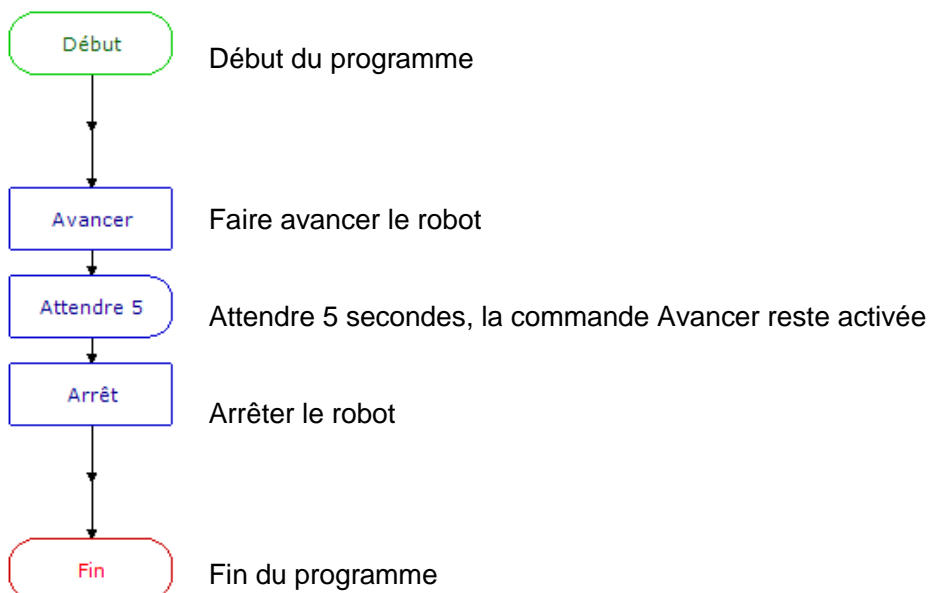
La sortie **C.5** sert à déterminer le sens de rotation du moteur **gauche** (**high** pour avancer, **low** pour reculer).



La commande **Arrêt** est constituée de 2 instructions.

Les consignes de vitesses **C.1** et **C.2** sont toutes deux mises à **0** pour arrêter le moteur.

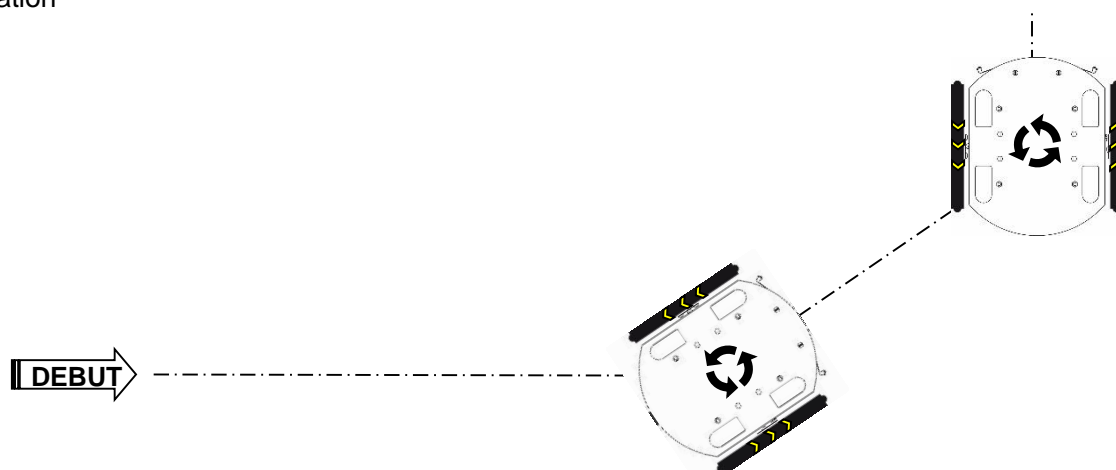
### Organigramme du programme :



## 6. Fiche technique N°2 – F2-Gauche

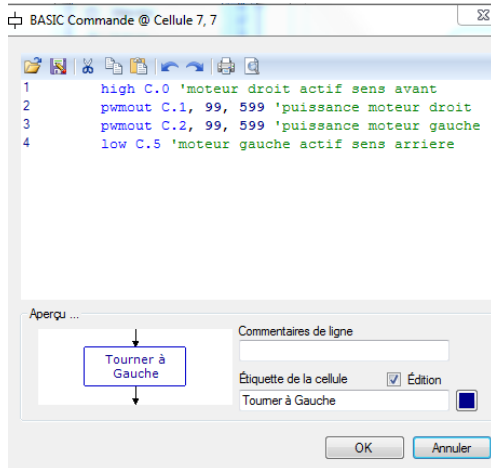
But de l'application	Avancer pendant 4 secondes, tourner à gauche, avancer de nouveau
Notions de programmation abordées	Contrôler la direction du Robot, rebouclage d'un programme
Programme associé	Code source Arduino Uno <b>F2-Gauche.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Aucun

Illustration



Suggestions de modifications	Modifier le programme pour que le robot tourne d'un angle plus faible
------------------------------	-----------------------------------------------------------------------

La première étape du programme est de configurer le bloc BASIC permettant au robot de **tourner à gauche**.



La commande **Tourner à Gauche** est constituée de 4 instructions :

La sortie **C.0** est activée pour faire avancer le moteur droit

La sortie **C.1** est la consigne de vitesse du moteur **droit**.

La sortie **C.2** est la consigne de vitesse du moteur **gauche**

La sortie **C.5** est désactivée pour faire reculer le moteur gauche

Ces 4 instructions font tourner le robot sur lui-même vers la gauche.

### Organigramme du programme :

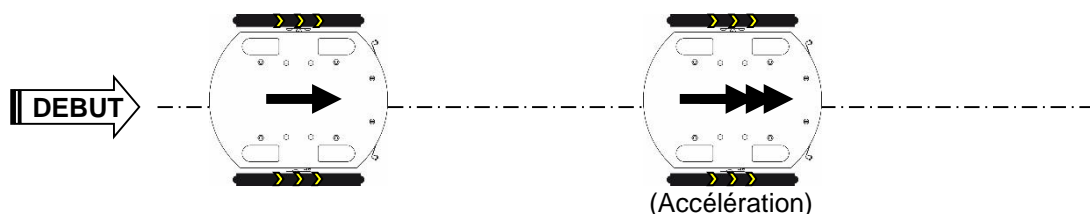
Le but de l'application demande un rebouclage permanent du programme



## 7. Fiche technique N°3 – F3-Accélération

But de l'application	Avancer 1 secondes puis accélérer 1 seconde
Notions de programmation abordées	Modification de la vitesse de déplacement
Programme associé	Code source Arduino Uno <b>F3-Acceleration.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Aucun

Illustration



Suggestions de modifications	Ajouter une variable de vitesse pour chaque moteur. Cette manipulation peut être utile pour faire avancer les deux moteurs à la même vitesse.
------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------

### Commentaire du programme d'automatisme :

La vitesse du RoboCoda est réglable pour chaque moteur (gauche et droit) sur une échelle qui s'étend de 0 à 1023.

Il est possible d'écrire directement la valeur de la vitesse à la place de la variable ou encore définir une vitesse pour chaque moteur dès le début pour faire avancer le robot en ligne droite s'il diverge dans une direction pour la même consigne de vitesse sur chaque moteur.



Le contrôle de la vitesse des moteurs se fait dans les instructions pwmout. Il est possible de régler la vitesse de chaque moteur sur une plage de 0 à 1023.

Le dernier paramètre de l'instruction pwmout correspond à la vitesse du moteur.

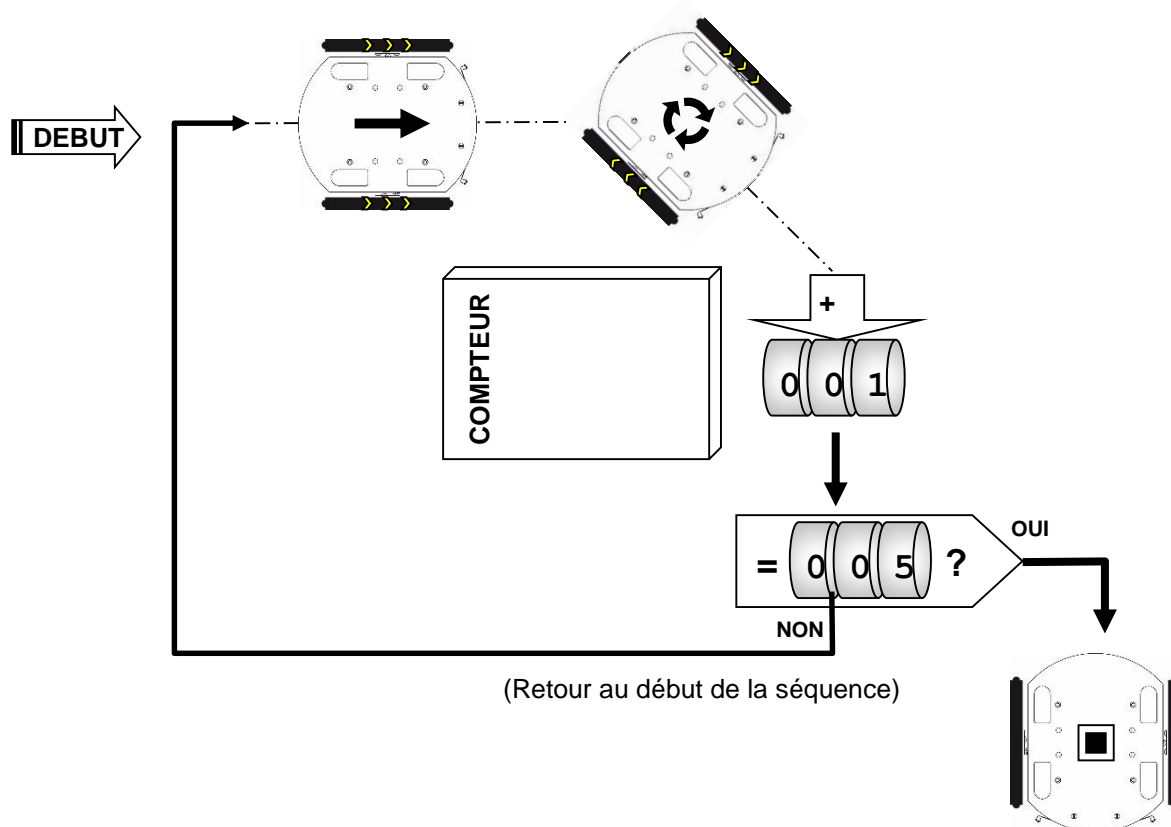
### Organigramme du programme :



## 8. Fiche technique N°4 – F4-Compteur

But de l'application	Répéter une figure 5 fois de suite puis s'arrêter.
Notions de programmation abordées	Répéter une séquence plusieurs fois. Définir une variable de comptage interne au programme.
Programme associé	Code source Arduino Uno <b>F4-Compteur.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Aucun

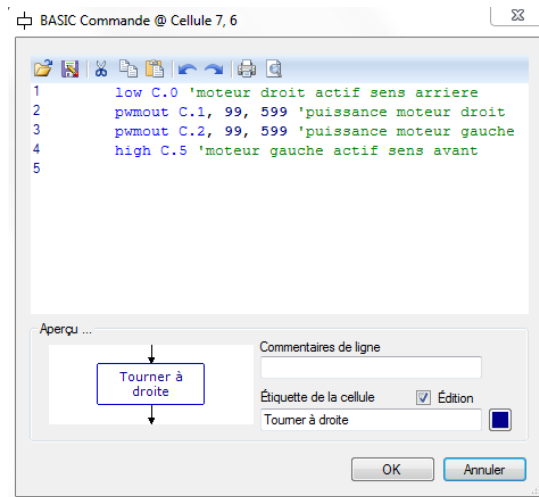
Illustration



Suggestions de modifications	Utilisation d'un autre type de boucle, modification du nombre d'itérations.
------------------------------	-----------------------------------------------------------------------------



La première étape du programme est de configurer le bloc BASIC permettant au robot de **tourner à droite**.



La commande **Tourner à droite** est constitué de 4 instructions :

La sortie **C.0** est désactivée pour faire reculer le moteur droit

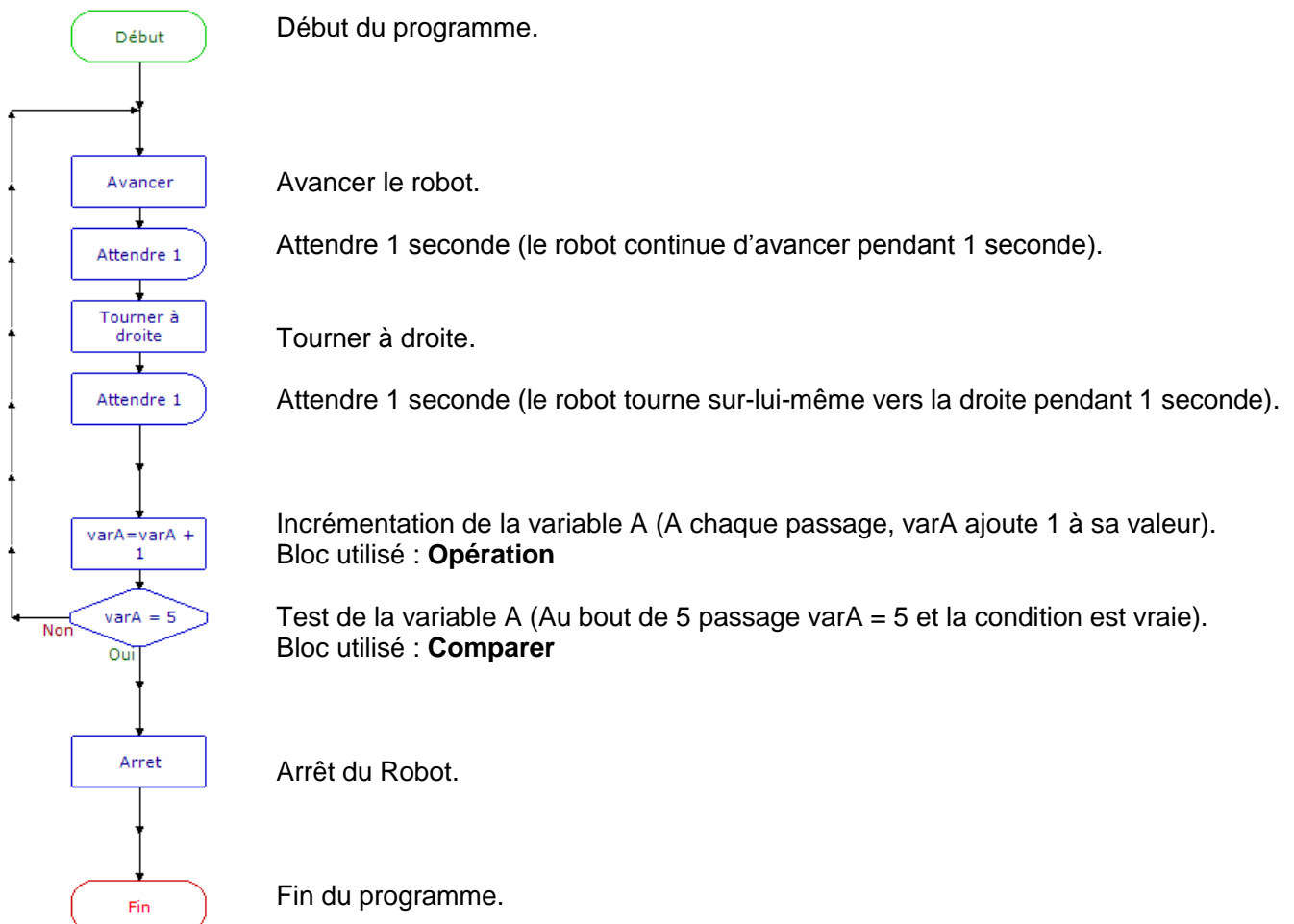
La sortie **C.1** est la consigne de vitesse du moteur **droit**.

La sortie **C.2** est la consigne de vitesse du moteur **gauche**

La sortie **C.5** est activée pour faire avancer le moteur gauche


Ces 4 instructions font tourner le robot sur lui-même vers la droite.

### Organigramme du programme :

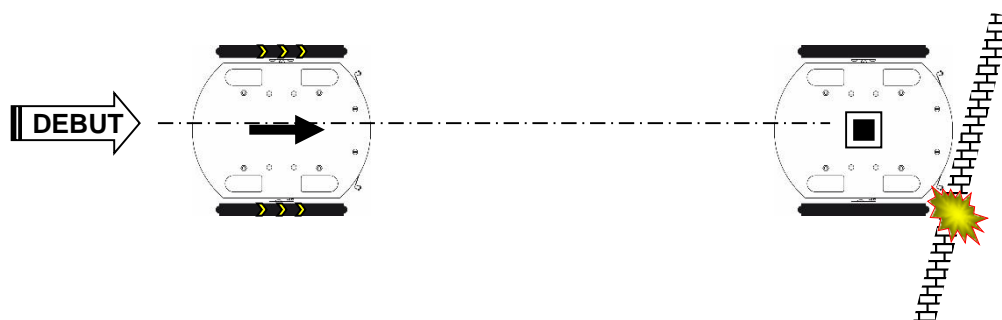




## 9. Fiche technique N°5 – M1-Arret

But de l'application	<b>S'arrêter au contact d'un obstacle.</b>
Notions de programmation abordées	Test de l'état de chaque capteur microrupteurs droit et gauche.
Programme associé	Code source Arduino Uno <b>M1-Arret.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Microrupteurs gauche et droit 

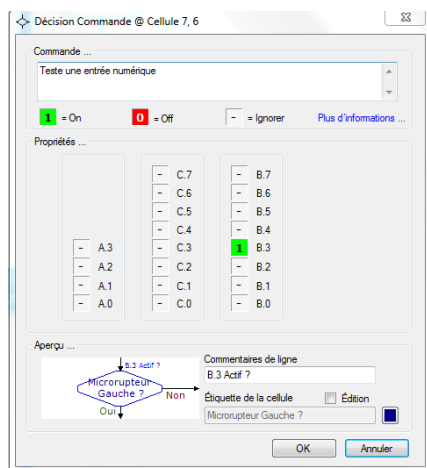
Illustration



Suggestions de modifications	
------------------------------	--



L'utilisation des capteurs microrupteurs se paramètre avec la commande **Décision**  **Décision**

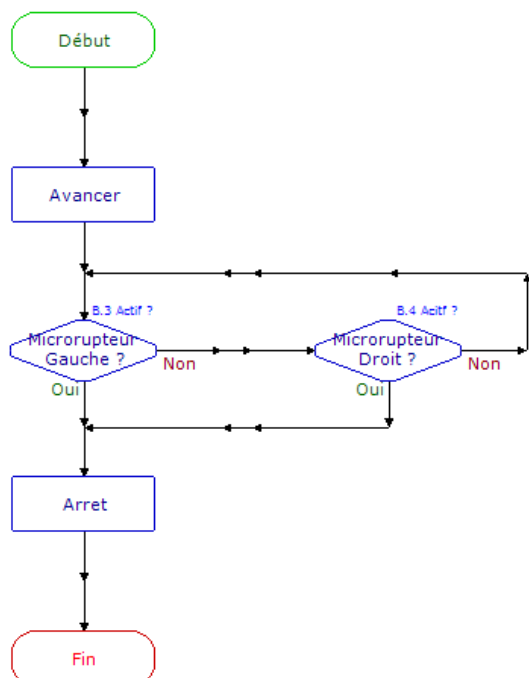


Configuration des blocs **Décision** :

**B.3** correspond à l'entrée où est connecté le microrupteur droit.

Si l'entrée **B.3** est active, alors le test sera **VRAI**

### Organigramme du programme :



Début du programme

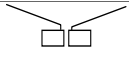
Avancer le Robot

Test sur les deux microrupteurs.  
Si un des deux microrupteurs est activé, alors le programme se poursuit.  
Si non le programme reboucle jusqu'à ce qu'un des microrupteurs est activé.

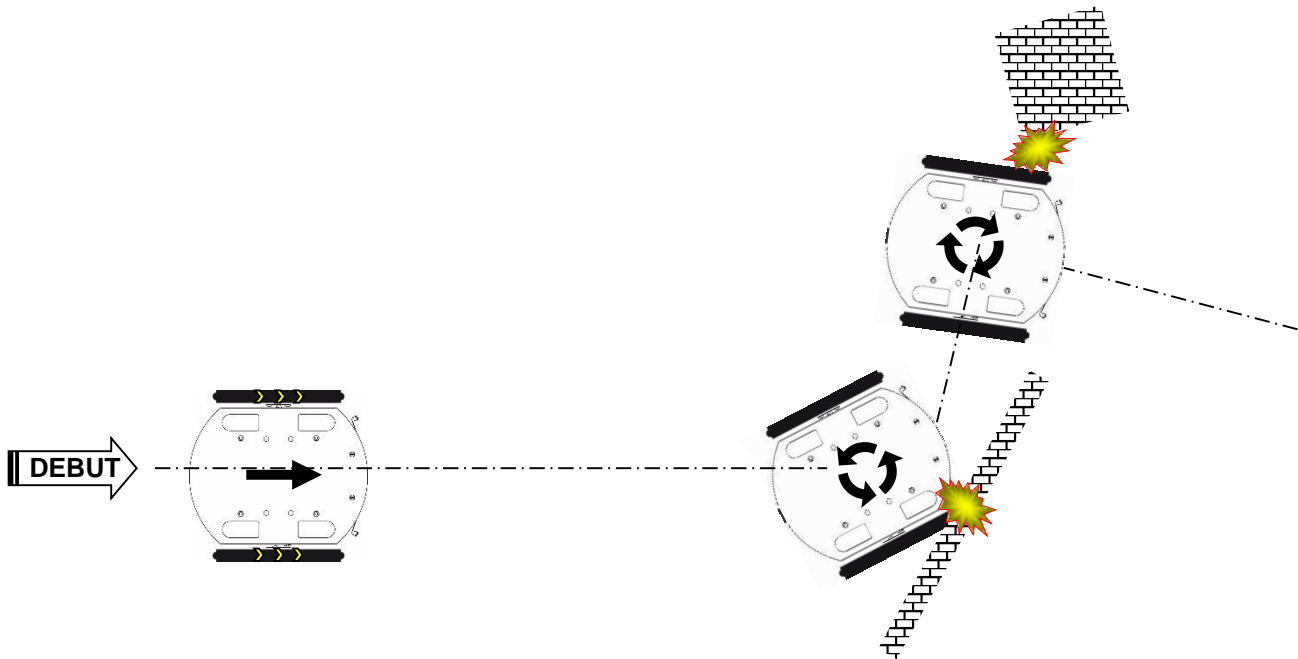
Arrêt du Robot

Fin du programme

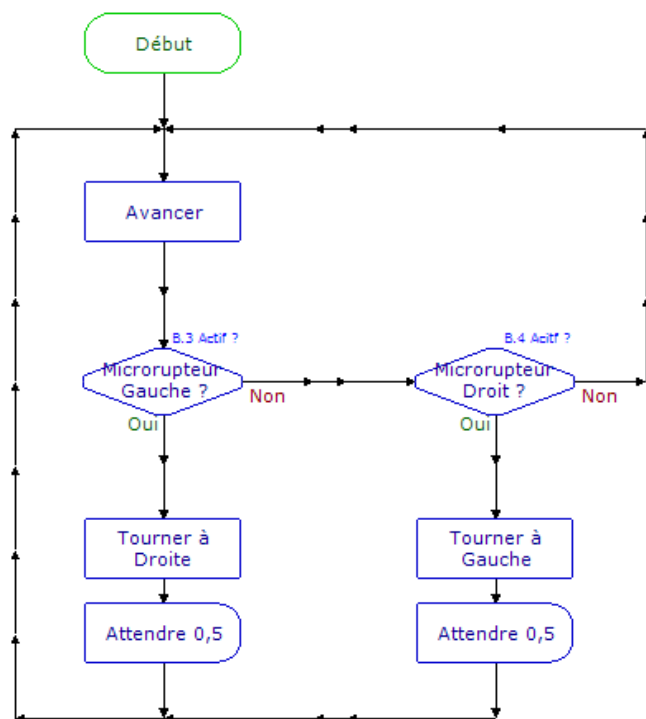
## 10. Fiche technique N°6 – M2-Changement\_direction

But de l'application	Changer de direction dès la rencontre d'un obstacle
Notions de programmation abordées	Test de l'état de chaque capteur microrupteurs droit et gauche puis exécute une instruction en fonction du microrupteur activé.
Programme associé	Code source Arduino Uno <b>M2-Changement_direction.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Microrupteurs gauche et droit 

Illustration



Suggestions de modifications

Organigramme du programme :

Début du programme

**Avancer** le Robot

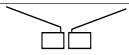
Test sur les deux microrupteurs.  
Si un des deux microrupteurs est activé, alors le programme se poursuit.  
Si non le programme reboucle jusqu'à ce qu'un des microrupteurs est activé.

**Tourner à Gauche** ou **Tourner à Droite** selon le microrupteur activé

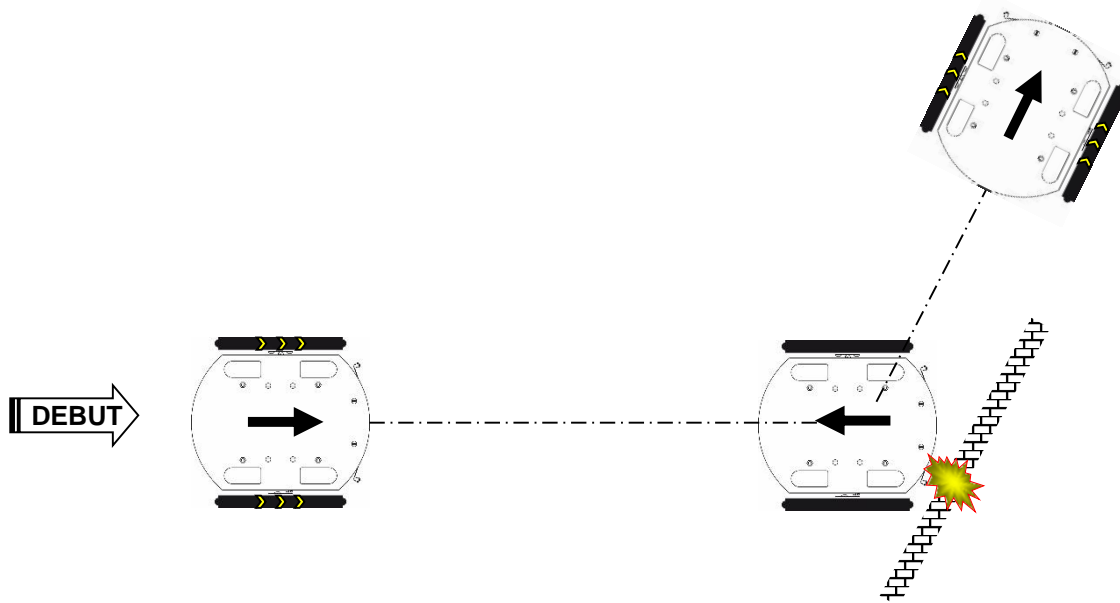
Tourner pendant 0.5 seconde

Rebouclage du programme

## 11. Fiche technique N°7 – M3-Dégagement

But de l'application	Eviter un obstacle détecté par un des capteurs microrupteurs en reculant dans un premier temps pour se dégager de l'obstacle puis en changeant de direction.
Notions de programmation abordées	Test de l'état de chaque capteur microrupteurs droit et gauche puis exécute une instruction en fonction du microrupteur activé.
Programme associé	Code source Arduino Uno <b>M3-Degagement.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Microrupteurs gauche et droit 

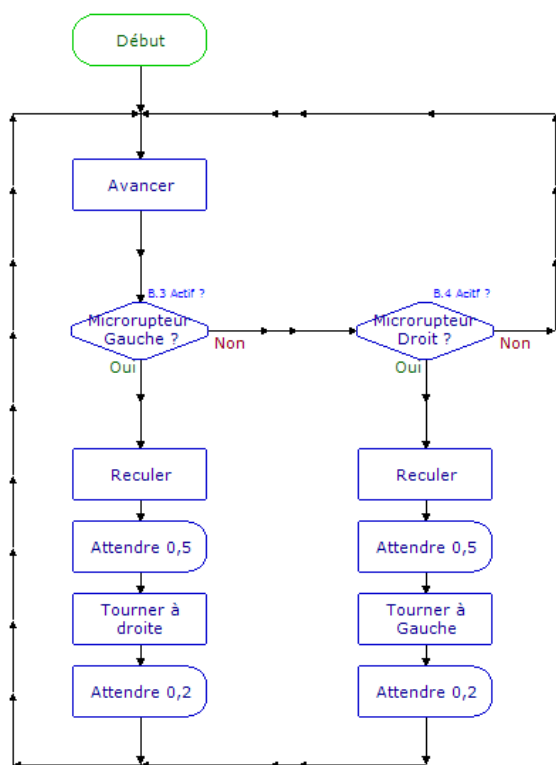
Illustration



Suggestions de modifications



### Organigramme du programme :

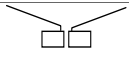


**Reculer** pendant 0.5 seconde

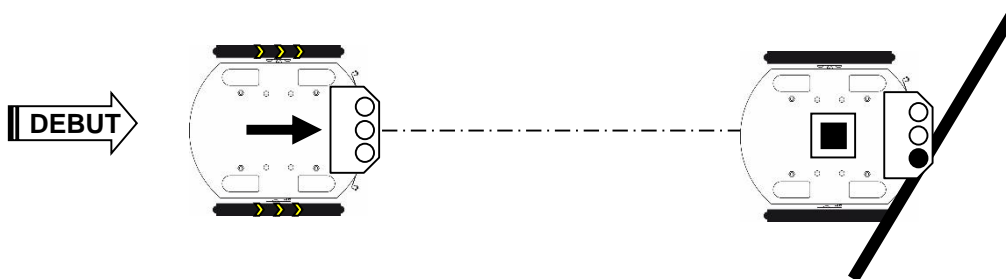
Tourner pendant 0.2 seconde

Rebouclage du programme

## 12. Fiche technique N°8 – DL1-Stop

But de l'application	Avancer en ligne droite et s'arrêter au croisement d'un marquage au sol.
Notions de programmation abordées	Test de l'état de chaque capteur détection de ligne.
Programme associé	Code source Arduino Uno <b>DL1-Stop.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	Microrupteurs gauche et droit 

Illustration



Suggestions de modifications	Arrêter le robot lorsque deux capteurs ou plus sont activés.
------------------------------	--------------------------------------------------------------



L'utilisation des capteurs microrupteurs se paramètre avec la commande **Décision**.

Configuration des blocs **Décision** :

**B.1** correspond à l'entrée où est connecté le détecteur gauche.

Si l'entrée **B.1** est active, alors le test sera **VRAI et le détecteur détecte une ligne noire**.

### Organigramme du programme :

Début du programme

Avancer le Robot


Test sur les trois détecteurs.  
Si un des détecteurs est activé, alors le programme se poursuit.  
Si non le programme reboucle jusqu'à ce qu'un des détecteurs est activé.

Arrêt du Robot

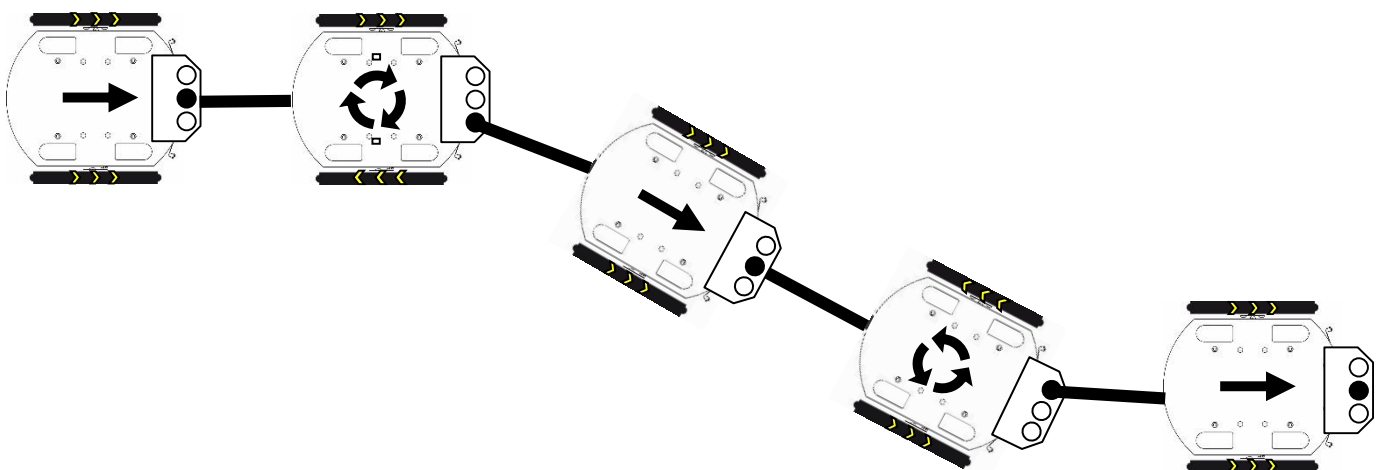
Fin du programme



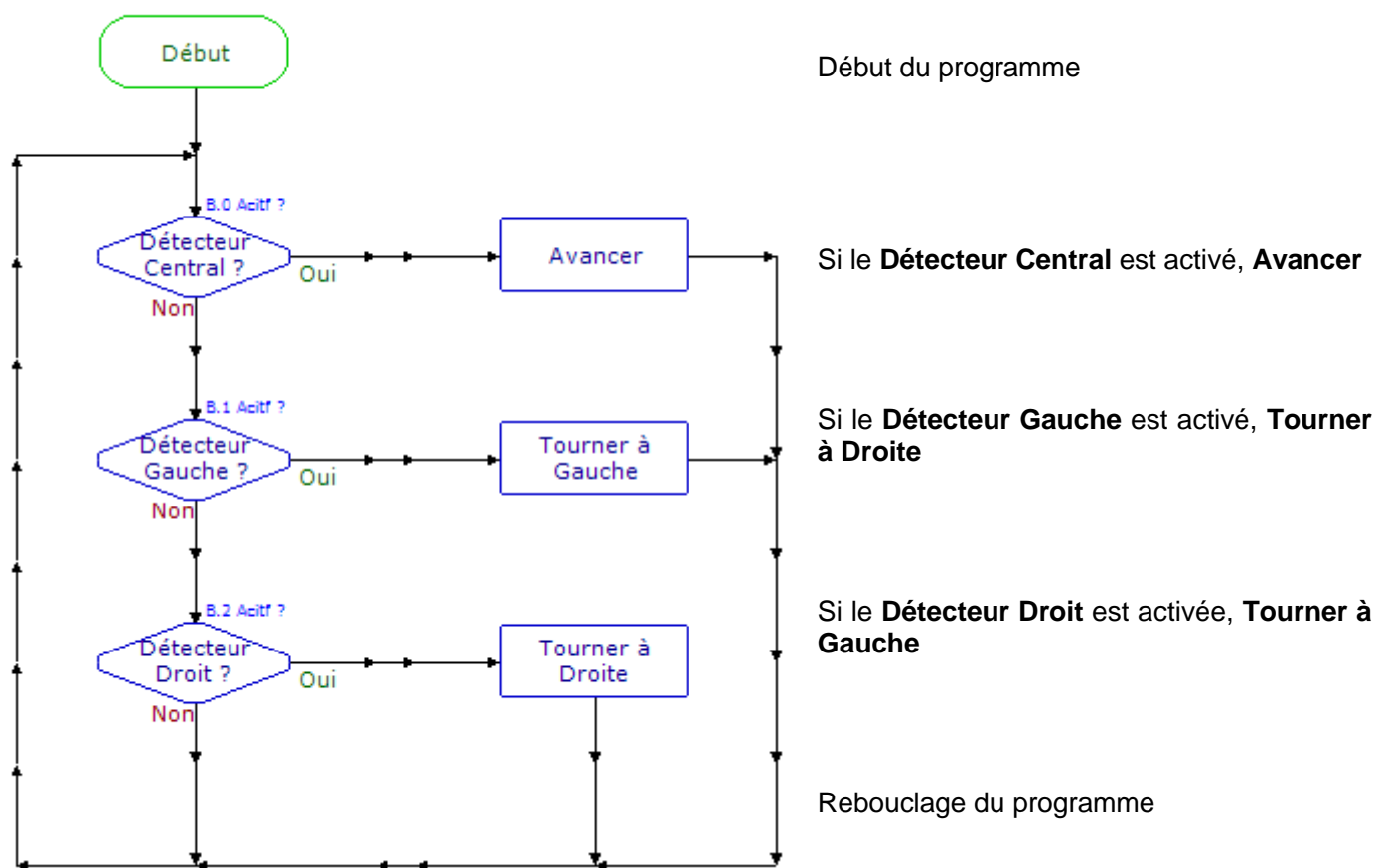
### 13. Fiche technique N°9 – DL2-Suivi\_de\_ligne

But de l'application	Suivre une ligne marquée au sol
Notions de programmation abordées	Condition sur chaque capteur détection de ligne.
Programme associé	Code source Arduino Uno <b>DL2-Suivi_de_ligne.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	 Module capteurs infrarouges (détection de marquage au sol).


#### Illustration



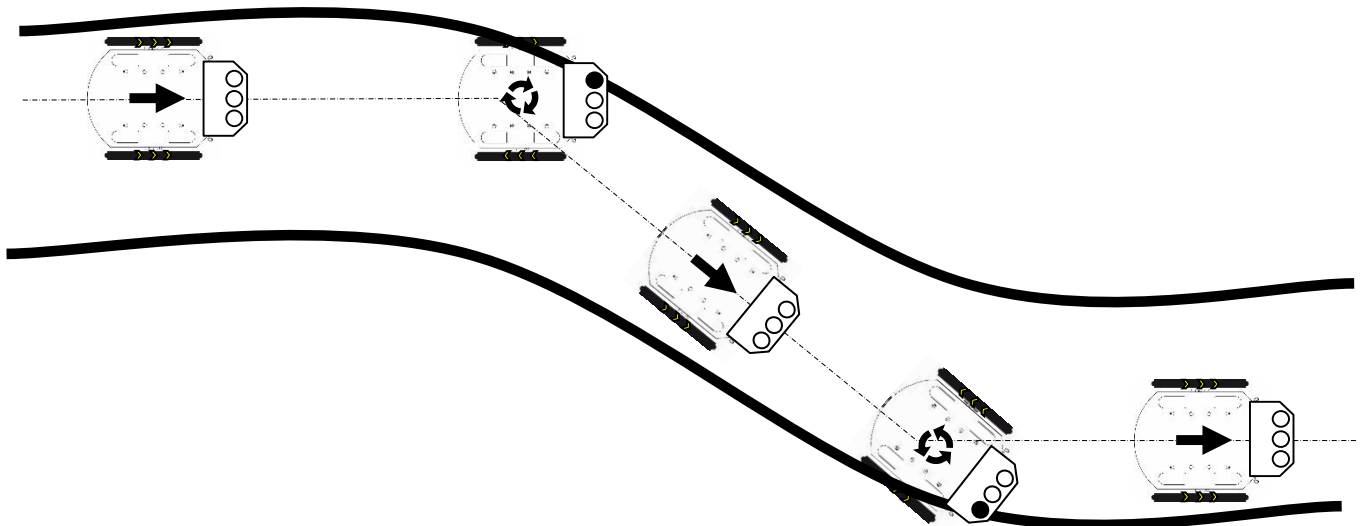
Suggestions de modifications	Créer de nouvelle sous-fonction qui remplacera les sous-fonctions droite() et gauche() et qui permettent de rendre le robot plus fluide dans ses déplacements (n'activer qu'un seul moteur lors d'un virage à droite ou à gauche)
------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Organigramme du programme :

## 14. Fiche technique N°10 – DL3-Piste

But de l'application	Evoluer sur une piste délimitée par des marquages au sol
Notions de programmation abordées	Condition sur chaque capteur détection de ligne.
Programme associé	Code source Arduino Uno <b>DL3-Piste.plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	 Module capteurs infrarouges (détection de marquage au sol).

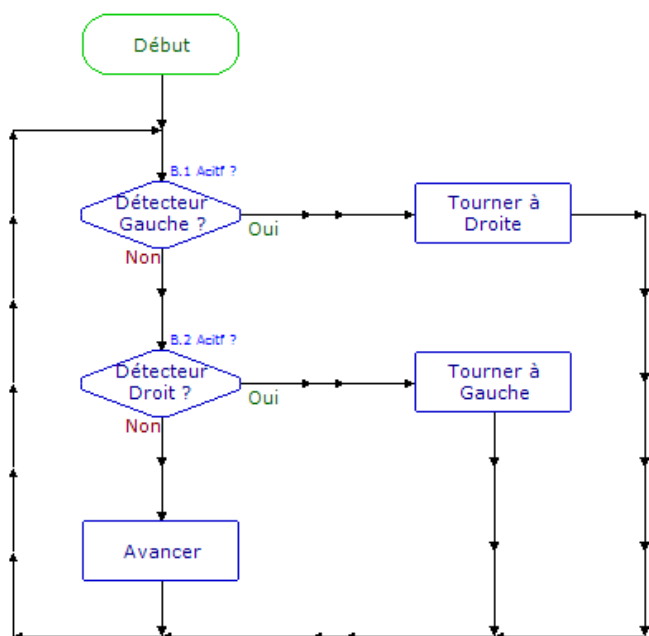
Illustration



Suggestions de modifications	Modifier la structure de la condition du détecteur central dans le cas où le robot détecte une ligne noire sur le détecteur central droit devant lui.
------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------



### Organigramme du programme :




Début du programme

Si le détecteur gauche est activé, **Tourner à Droite**

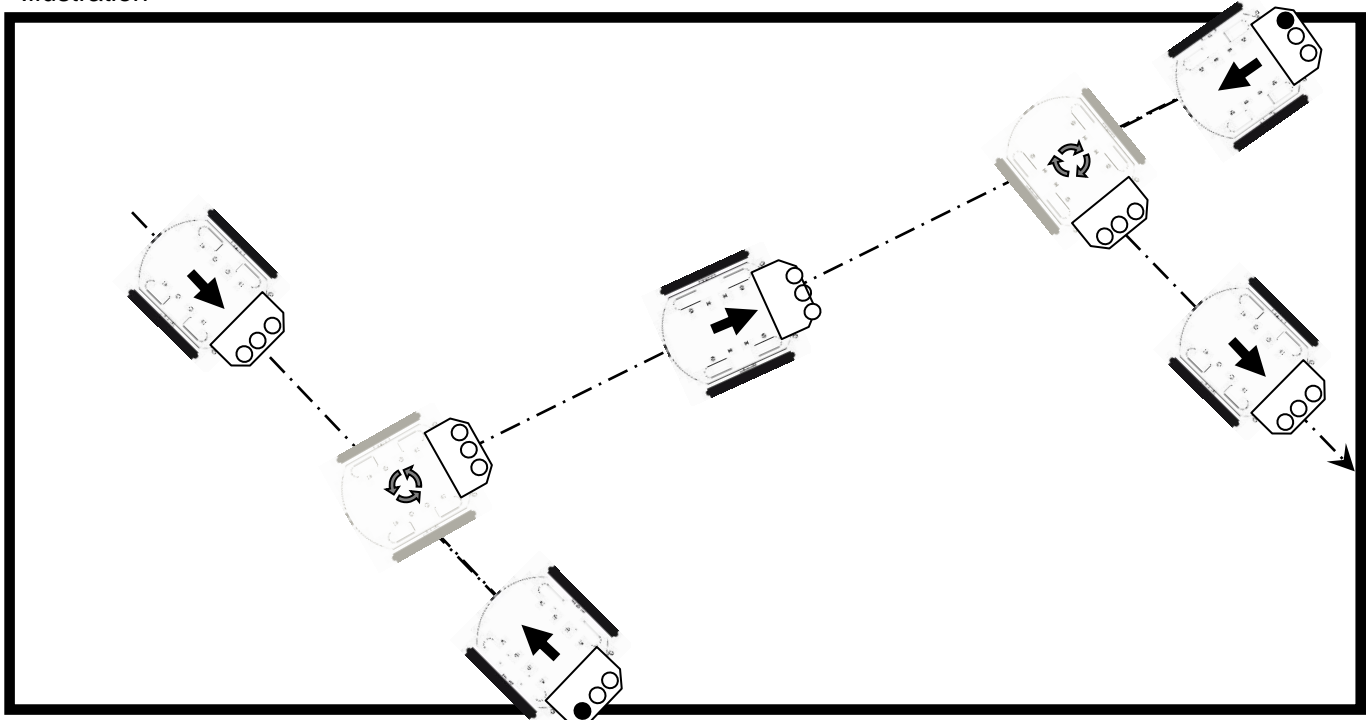
Si le détecteur droit est activé, **Tourner à Gauche**

Si aucun des détecteurs n'est activé, le Robot avance

## 15. Fiche technique N°11 – DL4-Perimetre

But de l'application	Evoluer à l'intérieur d'un périmètre rectangulaire délimité par un marquage au sol
Notions de programmation abordées	Combiner deux manœuvres successives en utilisant un temps d'attente
Programme associé	Code source Arduino Uno <b>DL4-Perimetre .plf</b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	 Module capteurs infrarouges (détection de marquage au sol).

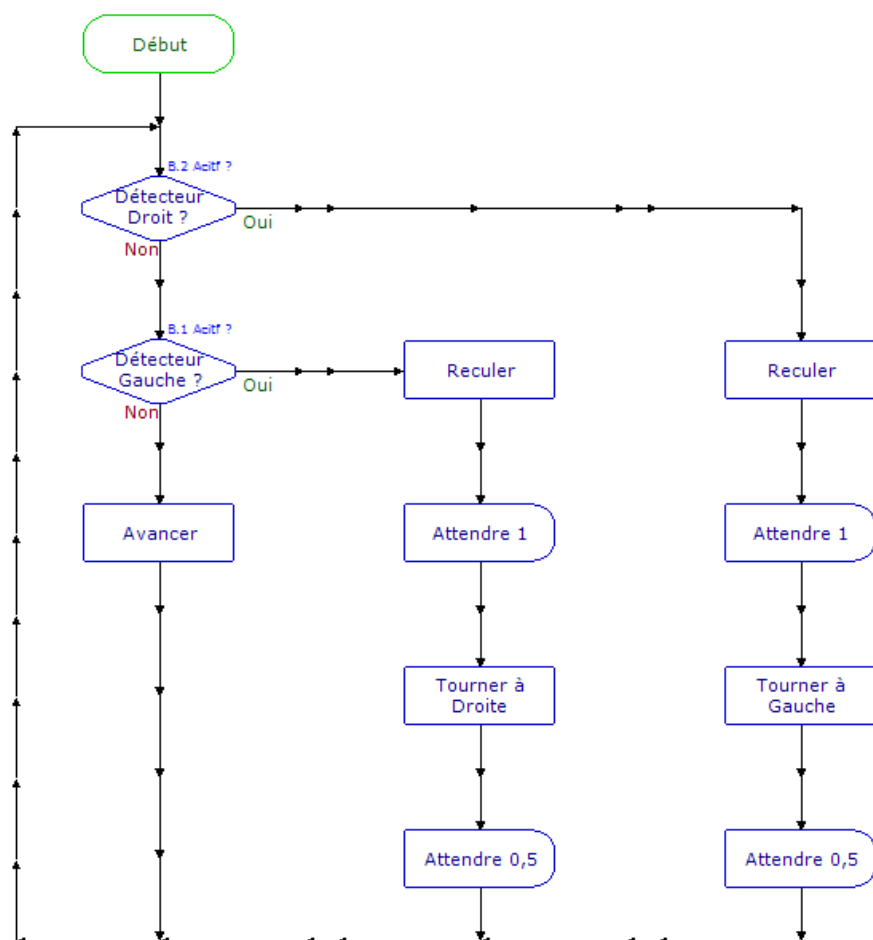
Illustration




Suggestions de modifications	Modifier la structure de la condition du détecteur central dans le cas où le robot détecte une ligne noire sur le détecteur central droit devant lui.
------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------



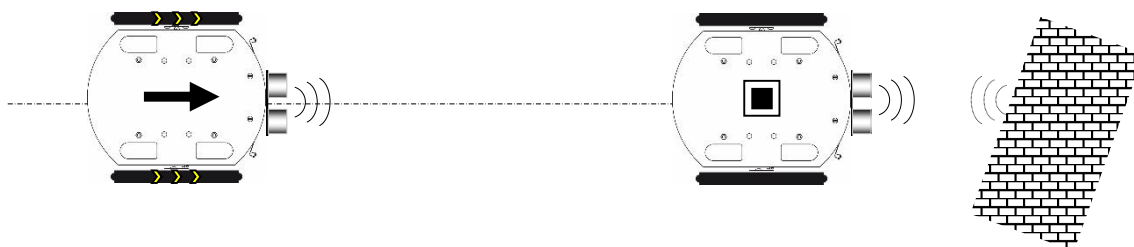
### Organigramme du programme :



## 16. Fiche technique N°12 – U1-Proximate

But de l'application	Evoluer en ligne droite et s'arrêter à l'approche d'un obstacle
Notions de programmation abordées	Utilisation du module à ultrasons pour détecter un obstacle à distance
Programme associé	Code source Arduino Uno <b><i>U1-Proximate .plf</i></b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	 Module capteur à ultrasons (détection d'obstacle à distance).

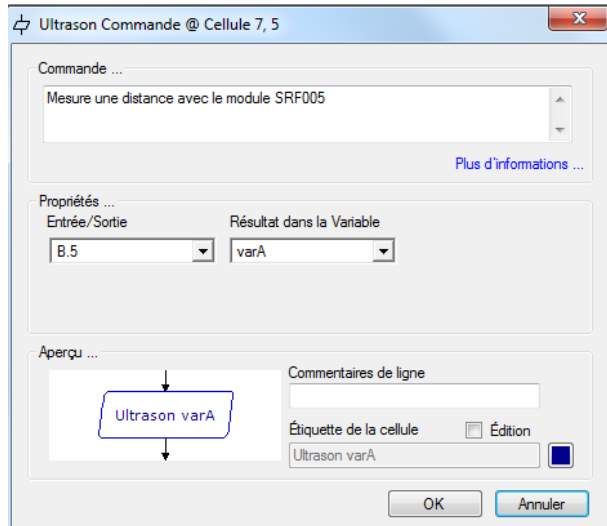
Illustration



Suggestions de modifications	Modifier la distance maximum d'arrêt du robot lors de la détection d'un obstacle
------------------------------	----------------------------------------------------------------------------------



L'utilisation du Télémètre à Ultrasons se configure avec la commande Ultrason :  Ultrason

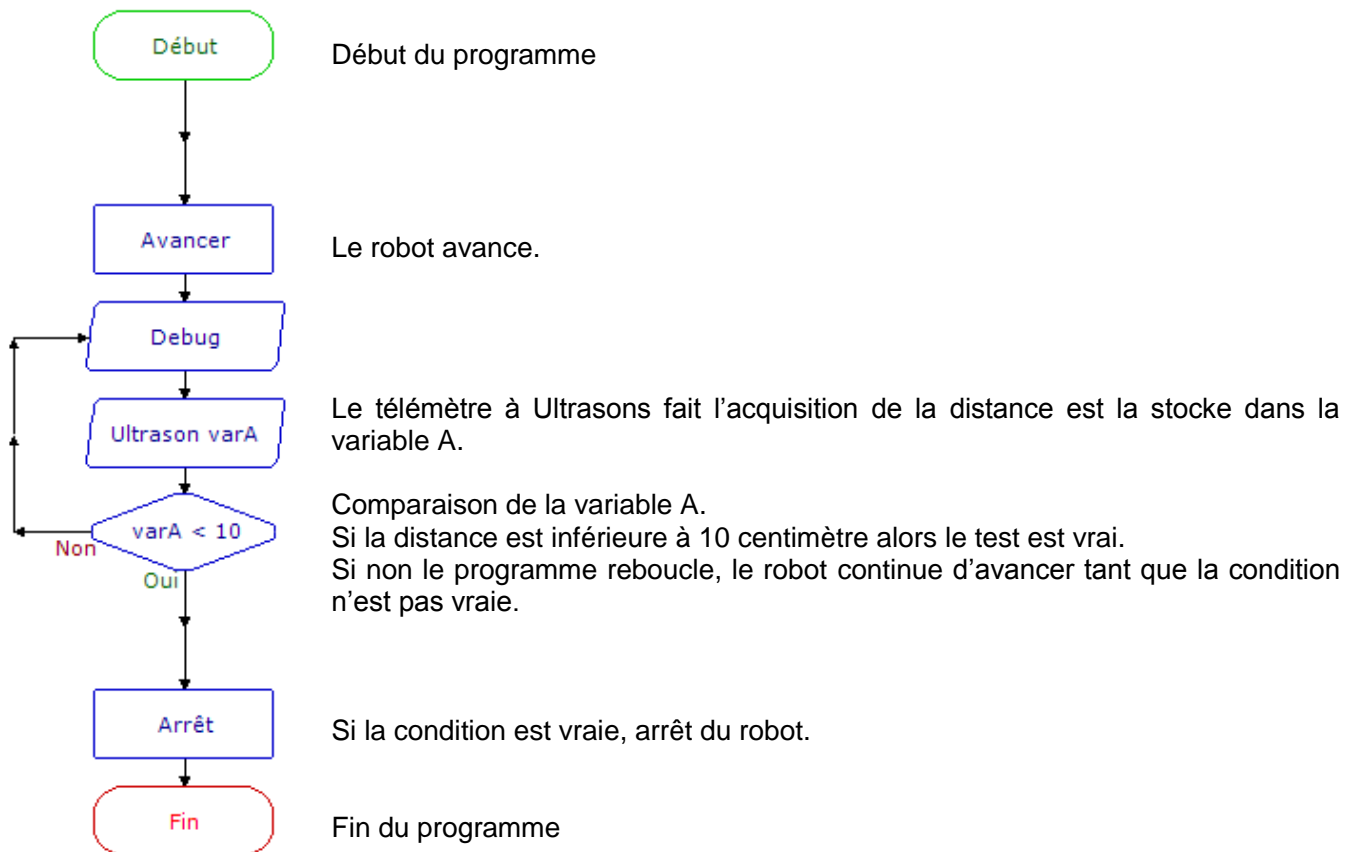


Configuration du bloc Ultrason

**B.5** correspond à l'entrée où est connecté le Télémètre à Ultrasons.


**VarA** correspond à la variable dans laquelle est stockée l'acquisition de la distance. La distance est directement exprimée en centimètre. La variable A s'étend de 0 à 255. Il est possible de mesurer une distance allant de 0 à 255 centimètres.

### Organigramme du programme :

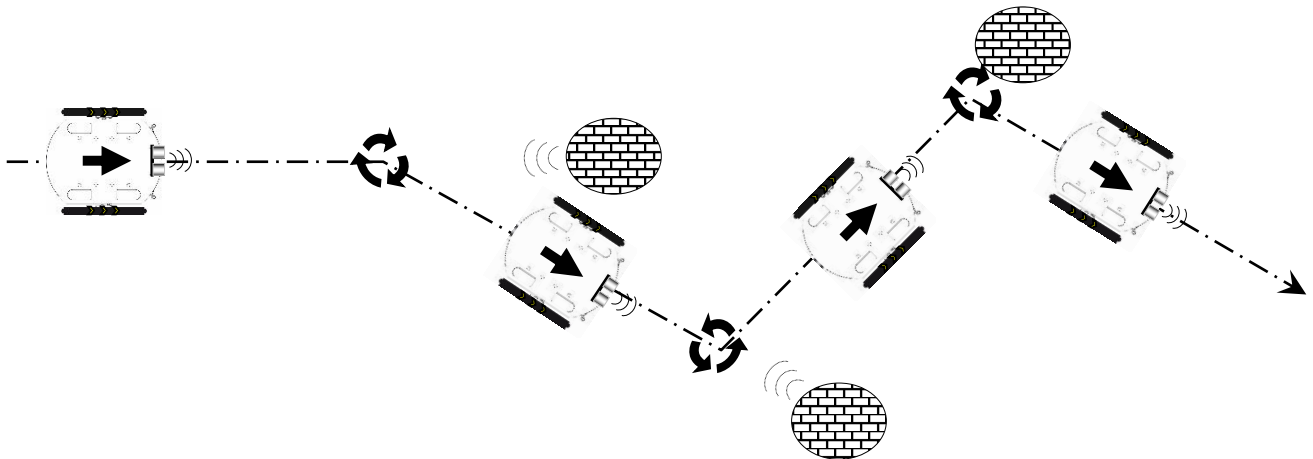




## 17. Fiche technique N°13 – U2-Slalom

But de l'application	Evoluer en ligne droite et alterner un changement de direction à droite puis à gauche à l'approche d'un obstacle
Notions de programmation abordées	Utilisation d'une variable dans une condition
Programme associé	Code source Arduino Uno <b><i>U2-Slalom.plf</i></b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	 Module capteur à ultrasons (détection d'obstacle à distance).

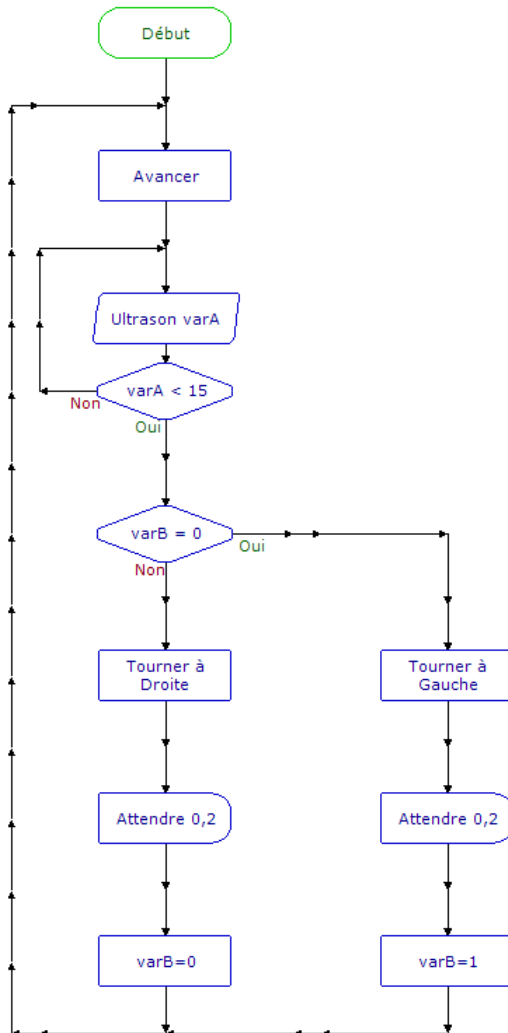
### Illustration



Suggestion de modification	Modifier la distance maximum d'arrêt du robot lors de la détection d'un obstacle
----------------------------	----------------------------------------------------------------------------------



Organigramme du programme :



Avancer.

Acquisition de la distance.

Test de la variable A.


Test sur la variable B (test sur la dernière direction prise).

**Tourner à Gauche ou Tourner à Droite.**

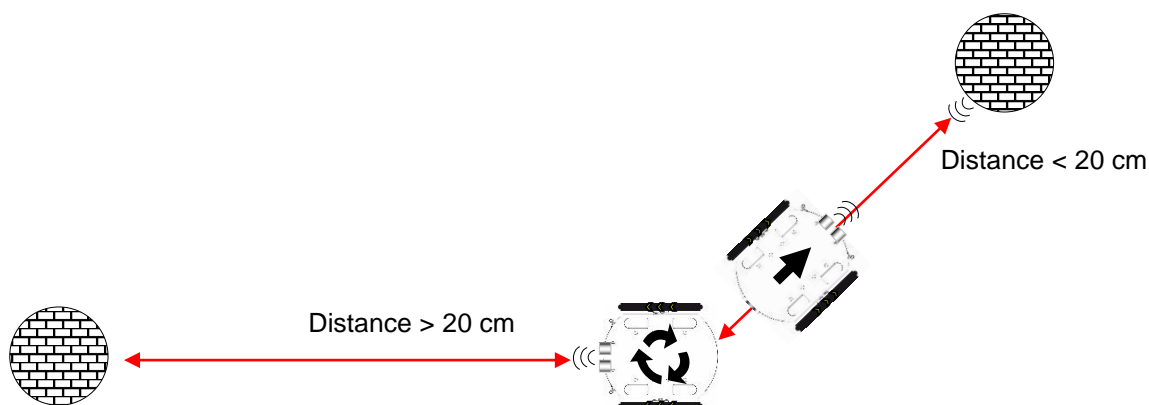
Tourner pendant 0.2 seconde

Changement de la variable B pour qu'à la prochaine détection, le robot prenne la direction opposée.

## 18. Fiche technique N°14 – U3-Cible

But de l'application	Balayer une zone pour détecter la présence d'une cible située à une distance inférieure à 20 cm et se diriger vers elle
Notions de programmation abordées	Evoluer en ligne droite et alterner un changement de direction à droite puis à gauche à l'approche d'un obstacle
Programme associé	Code source Arduino Uno <b><i>U3-Cible.plf</i></b> (Fichiers téléchargeables sur <a href="http://www.a4.fr">www.a4.fr</a> )
Capteur mis en jeu	 Module capteur à ultrasons (détection d'obstacle à distance).

### Illustration



Suggestion de modification	Permettre au robot de s'arrêter lorsqu'il est à proximité de l'obstacle
----------------------------	-------------------------------------------------------------------------

Organigramme du programme :

