

Séquence 1 : Découvrir les fonctionnalités du robot

Afin de vérifier si la plate forme robotisée répond au cahier des charges, vous allez réaliser quelques programmes afin de valider son fonctionnement.

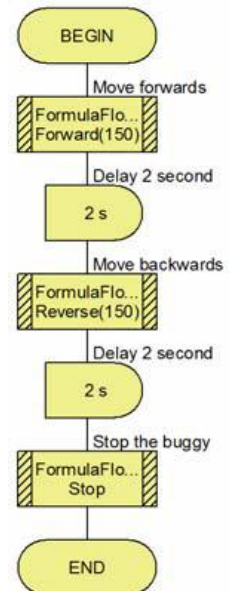
1. Marche avant - marche arrière.

Ce programme utilise les macros intégrés «Forward», «Reverse» et "Stop" pour contrôler le mouvement du robot Formula Flowcode.

- ❖ Créer ce programme sous Flowcode puis le télécharger sur le robot. Une fois qu'il a été téléchargé, vous devrez appuyer sur l'un des interrupteurs à l'avant du robot pour démarrer le programme (option dans la fenêtre "Propriétés" du Formula Flowcode).

Q1. Le robot dévie légèrement vers la gauche ou la droite, expliquer pourquoi.

Q2. Que faudrait-il rajouter sur les roues afin de palier ce problème ?



La macro "SetMotors" vous permet de spécifier une valeur différente pour chaque moteur et vous permettra de compenser ce problème.

2. Tourner autour

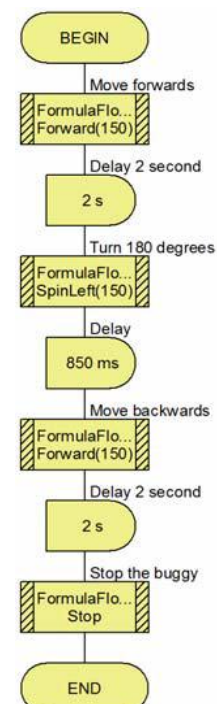
Pour faire tourner le robot, il faut utiliser "SpinLeft" (ou "SpinRight"). Pour connaître le temps nécessaire à faire un demi-tour de 180° par exemple, il faut faire des essais pour déterminer la durée de la temporisation pendant laquelle le robot tourne.

- ❖ Créer ce programme sous Flowcode puis le télécharger sur le robot. Vous aurez besoin d'expérimenter avec des valeurs de temporisation différentes (ou modifier "SpinLeft" macro) afin de trouver les valeurs qui conviennent le mieux au fonctionnement.

Ce programme est un exemple de « contrôle en boucle ouverte ».

Ce programme fonctionne (une fois que vous avez trouvé la valeur du retard correct), mais pas en toutes circonstances.

Q3. Placer le robot sur une autre surface, la rotation est-elle toujours de 180°, justifier pourquoi.



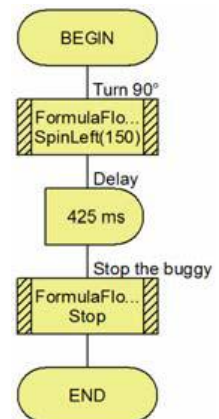
3. Créer sa macro

Vous pouvez également construire et utiliser votre propres "Macros". C'est une fonction qui permet à vos programmes de devenir modulaire.

Dans le programme précédent, nous avons travaillé sur un temps de retard qui a permis au robot de tourner sur 180 °.

Dans ce programme, vous allez l'utiliser pour créer une macro qui fait le tour du robot de 90 °. Vous allez ensuite utiliser cette macro pour recréer le virage à 180 ° du précédent programme.

- ❖ Tout d'abord, ouvrez le programme précédent, puis sélectionnez "Enregistrer sous" dans le menu "Fichier". Pour créer une nouvelle macro, sélectionnez "Macro ... Nouveau ..." à partir du menu. Cela fera apparaître une fenêtre où vous pouvez taper le nom de la macro, une description, et quelques autres informations. Pour l'instant, tout simplement tapez "Turn90" que le nom et puis cliquez sur "OK". Vous allez voir un organigramme vierge.
- ❖ Ajouter le "SpinLeft" et l'icône de retard à cette nouvelle macro. Rappelez-vous que le temps de retard sera d'environ la moitié de celle du retard utilisé pour tourner de 180 °. Revenir ensuite au programme « principal » organigramme (menu "Fenêtre ... Principal"). Dans l'organigramme principal, supprimer l'ancienne macro "SpinLeft" et les macros de retard. A leur place, ajouter la nouvelle macro "Turn90". Téléchargez ce nouveau programme dans le robot et confirmer que cela fonctionne comme précédemment.



Programmer des trajectoires prédéfinies

Traduire un comportement à l'aide d'un algorithme : Boucle ouverte



- ❖ Tracer l'une de ces trois trajectoires au sol. Dans la mesure du raisonnable, vous choisirez de manière arbitraire les dimensions de la forme.

Q4. Programmer les trajectoires suivantes en utilisant les sous programmes adéquats. Vous devrez définir le protocole expérimental que vous avez mis en place.

Appeler votre enseignant pour valider chaque forme

Q5. La trajectoire est-elle bien suivie ? Expliquer à quoi sont dus ces erreurs ?

Optimisation 1 de votre programme

Les capteurs de distance sont des capteurs infrarouges qui mesurent la distance avec un obstacle. Ils retournent une valeur représentative de la distance à l'obstacle.

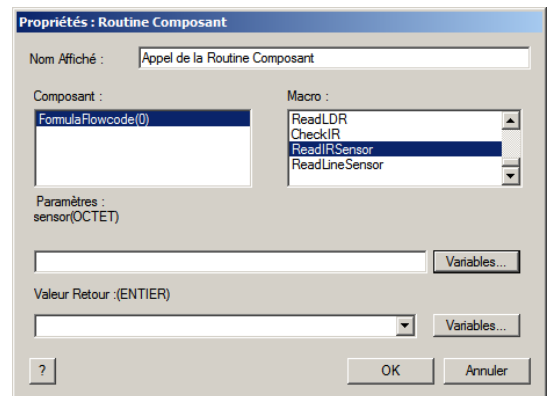
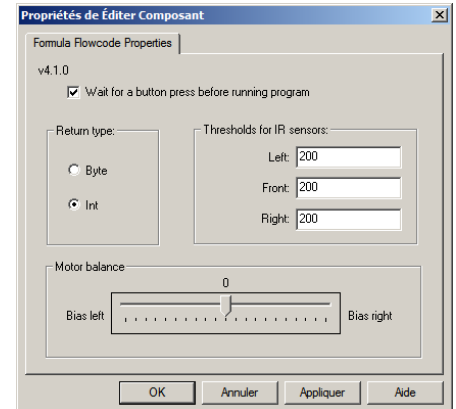
La fonction « ReadIRSensor (sensor) » retourne une valeur sur 8 bit (octet) (0 - 255) ou sur 10 bit (int)(0 - 1023) selon la propriété de la variable de retour choisie.

En affichant les propriétés étendues du composant formula flowcode, vous pourrez choisir entre les deux formats.

Si la variable d'entrée sensor est égale à 'L' ou 'l' ou 0 c'est alors le capteur de gauche qui est lu.

Si la variable d'entrée sensor est égale à 'F' ou 'f' ou 1 c'est alors le capteur de devant qui est lu.

Si la variable d'entrée sensor est égale à 'R' ou 'r' ou 2 c'est alors le capteur de droite qui est lu.



Q6. Réaliser un programme qui li la valeur de distance sur 8 bits et l'affiche sur les leds.

Appeler votre enseignant pour valider le programme

Q7. Utiliser ensuite ce programme pour tracer la caractéristique du capteur sous Excel. Vous mettez sur l'axe des abscisses la distance en cm (jusqu'à 15cm) et en ordonnée la valeur retournée par le capteur en décimal.

Appeler votre enseignant pour valider le programme

Traduire un comportement à l'aide d'un algorithme : Boucle fermée

Q8. Maintenant vous allez utiliser ces capteurs pour optimiser votre programme réalisé en boucle ouverte (voir paragraphe précédent).

Afin de ne pas être « collé » au mur, le robot devra être à une distance de 2cm du mur.

Remarque : une autre fonction CheckIR (sensor) retourne une variable booléenne représentant le résultat de la comparaison entre la valeur lue par le capteur et une valeur de seuil prédéfinie dans la fenêtre des propriétés étendues (voir Threshold Properties)

Si la variable d'entrée sensor est égale à 'L' ou 'l' ou 0 c'est alors le capteur de gauche qui est lu.

Si la variable d'entrée sensor est égale à 'F' ou 'f' ou 1 c'est alors le capteur de devant qui est lu.

Si la variable d'entrée sensor est égale à 'R' ou 'r' ou 2 c'est alors le capteur de droite qui est lu.

Si la valeur lue par le capteur est inférieure au seuil (Pas d'obstacle suffisamment près) la fonction retourne 0 sinon elle retourne 1

Il n'est pas obligatoire d'utiliser cette fonction.

Appeler votre enseignant pour valider le programme

Q9. Le programme est-il optimisé ? Quels sont les erreurs qui subsistent ?

Q10. Réaliser un rapport (Word ou PowerPoint) contenant les programmes commentés. Les résultats de simulations commentés de chaque programme. Ces résultats seront ensuite présentés à l'ensemble de la classe.

Optimisation 2 de votre programme

On désire rajouter un capteur de vitesse sur notre robot. A chaque tour de roue, une impulsion sera envoyée au microcontrôleur.

On connaît :

- Le temps entre deux impulsions, correspondant à un tour de roue,
- Le diamètre de la roue.

Q11. Proposer un algorithme ou organigramme permettant de donner la vitesse du robot en fonction de ces caractéristiques.

Q12. Tester et commenter le programme comptvitesse.

Q13. Modifier ce programme afin d'afficher la vitesse sur la première ligne (à la place du temps mis pour faire un tour)