

Table des matières



Pourquoi avons-nous besoin de robots ?	p. 58
Robots fischertechnik	p. 60
Acteurs	p. 60
Capteurs	p. 60
Interface ROBO	p. 61
Logiciel ROBO Pro	p. 61
Alimentation électrique	p. 61
Processus d'expérimentation	p. 62
Premières étapes	p. 62
Le premier robot simple	p. 64
Robots roulants intelligents	p. 66
Maquette de base	p. 66
Le viseur de lumière	p. 68
Le viseur de trace	p. 70
Robot avec reconnaissance d'obstacle	p. 71
Viseur de lumière avec reconnaissance d'obstacle	p. 74
Robot avec reconnaissance d'arête	p. 76
Le robot marchant	p. 79
Possibilités d'extension	p. 81
Emetteur manuel à infrarouge	p. 81
ROBO RF ROBO Data Link	p. 81
ROBO I/O-Extension	p. 82
Recherche des pannes	p. 83

Pourquoi avons-nous besoin de robots?



■ Le terme « robot » a été utilisé pour la première fois en 1923 dans le roman « Golem » de Carel Capek. Ce personnage virtuel devait, grâce à ses capacités, effectuer le travail des hommes.

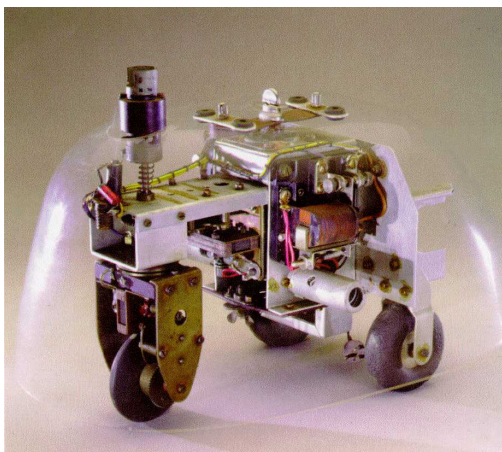
Dans les années 30 et 40 du 20^{ème} siècle, le robot devient plutôt un type d'automate. Les diverses tentatives pour le doter de caractéristiques humaines, comme p.e. une tête avec des lampes clignotantes en guise d'yeux, ne suscitent plus chez nous aujourd'hui qu'un éventuel sourire. Ces machines ne possèdent que très peu de mobilité et encore moins d'intelligence. Le principe de la commande ayant une grande influence sur la robotique, la conception des robots est devenue plus réaliste avec l'apparition des circuits électroniques. La question de « l'intelligence » du robot fait encore aujourd'hui l'objet de recherches et d'études pour de nombreuses sociétés et universités, ainsi que de nombreux instituts.

■ On attendait les premières ébauches de solution grâce à la cybernétique. Le terme « cybernétique » est dérivé du mot grec Kybernetes. Le Kybernetes était le navigateur sur les bateaux à rames grecs. Il devait déterminer la position du bateau et calculer le cap jusqu'à destination.

Il est donc évident que la cybernétique devait rendre le robot « intelligent ». Comment peut-on vraiment se représenter un tel comportement intelligent ?

Nous voulons essayer d'y réfléchir à l'aide d'une expérience mentale. Chacun a déjà une fois observé le comportement d'une mite dans le cercle de lumière d'une lampe. La mite reconnaît la source lumineuse, vole dans sa direction puis fait une embardée tout juste avant de s'écraser contre la lampe. Il est évident que pour avoir ce comportement la mite doit reconnaître la source lumineuse, découvrir un chemin pour s'y rendre et ensuite voler dans sa direction. Ces capacités sont basées sur les modèles de comportement instinctifs et intelligents de l'insecte.

Essayez à présent de transposer ces capacités dans un système technique. Vous devez reconnaître la source lumineuse (capteurs optiques), effectuer un déplacement (actionner les moteurs) et établir un rapport sensé entre la reconnaissance et le déplacement (le programme).



■ Dans les années 50, l'Anglais Walter Grey a mis à exécution l'expérience mentale susmentionnée.

Divers animaux « cybernétiques » possédant un comportement particulier, comme par exemple celui de la mite, furent créés à l'aide de capteurs, de moteurs et de circuits électroniques simples. L'illustration montre une reproduction de la tortue "cybernétique" exposée au musée Smithsonian de Washington.

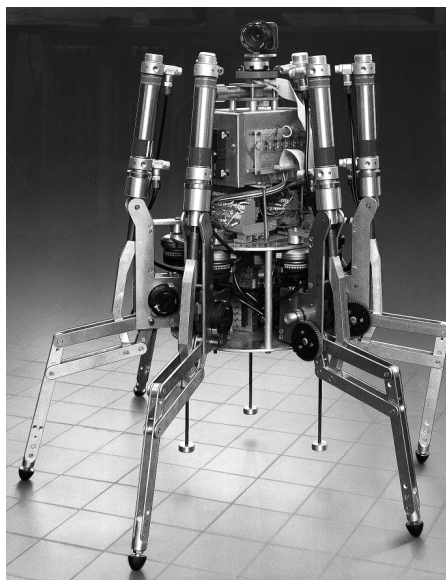
■ En nous basant sur ces réflexions, nous allons également créer pour nos robots des « modèles de comportement » appropriés et nous essaierons de les faire comprendre à nos robots sous la forme de programmes.



Mais, pourquoi avons-nous désormais besoin de robots mobiles ? Essayons une fois d'appliquer le comportement de notre « mite virtuelle » aux instruments techniques. La recherche de la lumière en constitue un exemple simple. Modifiez la source lumineuse en plaçant une bande claire, la ligne directrice, sur le sol et en orientant les capteurs non plus vers l'avant mais vers l'arrière. Un robot mobile peut s'orienter, à titre d'exemple dans un entrepôt, à l'aide de ce type de lignes directrices. Des informations supplémentaires, p.e. sous forme d'un code-barres à certains endroits de la ligne, incitent les robots à accomplir d'autres actions à ces endroits, comme p.e. la prise ou la pose d'une palette. De tels systèmes de robots existent réellement. Dans les grands hôpitaux, l'acheminement des consommables, comme p.e. les draps et les taies d'oreiller, nécessite parfois de longs trajets. Le transport de ces consommables par le personnel soignant est onéreux et partiellement associé à un travail corporel lourd. Ces activités restreignent qui plus est le temps consacré aux soins des patients.

■ Les scientifiques s'intéressent depuis quelques années à une autre forme de déplacement très répandue dans la nature, à savoir la marche et la course. Des robots capables de se déplacer sur des jambes ont été développés. Le robot électropneumatique « Achille » développé à l'Académie royale militaire de Bruxelles est un exemple de robots marchant à six jambes. Equipé d'une caméra sur sa partie supérieure et sur ses six jambes, ce robot doit mécaniquement réagir aux obstacles en hauteur ou en profondeur (objets ou trous).

De telles machines marchantes ont pu être utilisées partout là où les véhicules à roues ou à chenilles n'avaient guère aucune chance, p.e. sur des terrains extrêmement accidentés ou meubles, pour escalader des obstacles, monter des escaliers, franchir des fossés ou accéder à des endroits difficiles ou dangereux dans des centrales nucléaires ou lors d'opérations de sauvetage.



Nous admettons donc que les robots mobiles peuvent tout à fait occuper une place importante dans la société moderne.



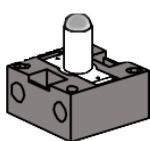
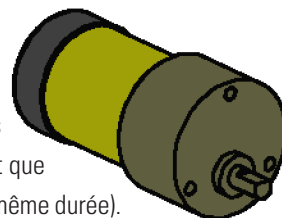
Robots fischertechnik

■ Comment pouvez-vous maintenant construire des robots à partir de notre boîte de construction Fischertechnik? Outre les capteurs (p. ex. les palpeurs) et les acteurs (p. ex. les moteurs), vous avez besoin de nombreuses pièces mécaniques pour construire une maquette. La boîte de construction Fischertechnik ROBO Mobile Set en constitue la base idéale. Les capteurs et acteurs suivants sont inclus dans cette boîte de construction:

Acteurs

Powermotor:

Deux de ces puissants moteurs à courant continu (9 VCC/2, 4 W) avec transmission intégrée et une démultiplication de 50:1 propulsent les maquettes de robots mobiles (ce qui signifie que le moteur tourne 50 fois et que l'arbre qui se dresse au-dessus du moteur ne tourne qu'une seule fois sur la même durée).



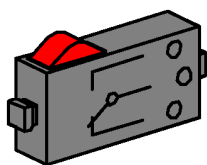
Ampoule lentille:

Cette lampe à incandescence (9 VCC/150 mA) sert à émettre des signaux lumineux simples.

Une lentille qui focalise la lumière émergente est intégrée à l'ampoule. En dirigeant le rayon lumineux sur un capteur de luminosité (transistor photo, voir ci-après), on peut créer une barrière lumineuse qui fait la distinction entre le clair et l'obscur. La lampe peut également être utilisée pour signaler certaines situations ou en tant que lampe clignotante pour émettre des signaux d'avertissement. Dans la boîte de construction, la lampe est utilisée avec 2 transistors photo en tant que capteur spécial pour la reconnaissance de la ligne.

Capteurs

■ Le palpeur est un exemple de capteur numérique. Les grandeurs numériques ne peuvent accepter que 2 situations différentes. Ces situations sont indiquées par les valeurs 0 et 1. Pour le palpeur, « 0 » signifie qu'aucun courant ne circule entre les connexions et « 1 » signifie que le courant circule.

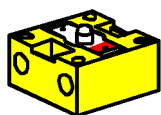
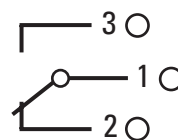


Pour les capteurs, on fait la différence entre les capteurs numériques et les capteurs analogiques.

Le **palpeur** Fischertechnik est conçu comme un commutateur inverseur. C'est la raison pour laquelle il possède 3 connexions. En appuyant sur le bouton rouge, un commutateur qui relie les connexions 1 et 3 l'une à l'autre est mécaniquement activé. Le contact entre les connexions 1 et 2 qui étaient reliées l'une à l'autre à l'état de repos est simultanément interrompu. Les deux positions de sortie possibles peuvent ainsi être consultées:

Fermée à l'état de repos (connexions 1 et 2 occupées)

Ouverte à l'état de repos (connexions 1 et 3 occupées).



Le **transistor** photo peut être utilisé aussi bien comme capteur numérique que comme capteur analogique. Dans le premier cas, il sert à reconnaître les passages distincts du clair à l'obscur, p. ex. d'une ligne signalisée. Cependant, l'intensité des quantités de lumière peut également être différenciée, le transistor photo fonctionne alors comme un capteur analogique. Les valeurs analogiques peuvent se modifier de n'importe quelle manière entre des valeurs extrêmes. Afin que ces grandeurs puissent être traitées par l'ordinateur, elles doivent être converties dans leurs valeurs numériques correspondantes.

Pour le transistor photo, il s'agit en fait d'un composant dit à semi-conducteurs dont les propriétés électriques dépendent de la lumière. Tout le monde connaît les cellules solaires qui permettent d'obtenir

du courant à partir de la lumière du soleil. Nous pouvons considérer le transistor photo comme l'association d'une cellule solaire miniature et d'un transistor. Les impulsions lumineuses (de photons) qui apparaissent sur le phototransistor produisent un courant très faible qui est ensuite amplifié par le transistor.

Remarque :

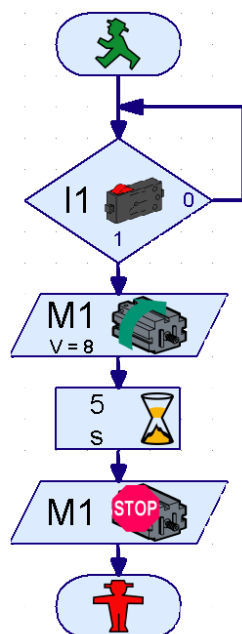
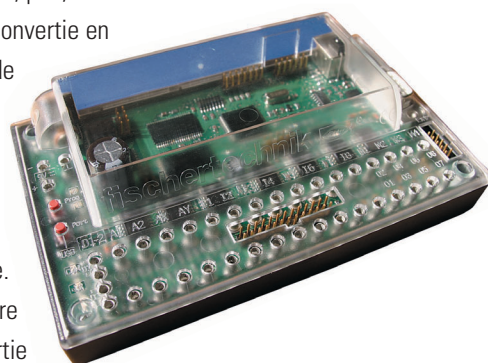
Respectez la polarité correcte lors du raccordement du phototransistor : Marquage rouge = Plus.

Tension admissible : 30 V maxi.

■ Vous pouvez raccorder et évaluer divers capteurs et acteurs sur la ROBO Interface. En complément des 8 entrées numériques, la ROBO Interface dispose de plusieurs entrées analogiques. Ainsi, p.e., une valeur de résistance appliquée sur les entrées AX et AY et comprise entre 0 et 5,5 k Ω est convertie en une valeur numérique comprise entre 0 et 1024. Les valeurs mesurées par un capteur de luminosité, p. ex. un transistor photo, sont de ce fait enregistrées et sont disponibles pour un traitement ultérieur. Des tensions comprises entre 0 et 10 VCC peuvent être mesurées aux entrées analogiques A1 et A2.

La fonction la plus importante de l'Interface consiste à associer les grandeurs d'entrée. L'Interface a, à cet effet, besoin d'un programme. Le programme décide de quelle manière produire, à partir des données d'entrée, les signaux des capteurs, les données de sortie appropriées, les signaux de commande du moteur, etc. Avec la ROBO Interface, vous disposez d'une puissance de calcul suffisante pour concevoir également des programmes ambitieux.

ROBO Interface

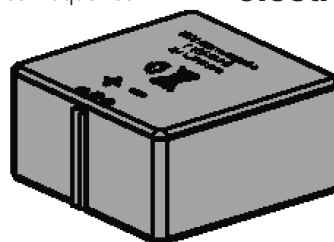


■ Il existe une interface de programmation graphique qui vous permet de développer le plus efficacement possible les programmes nécessaires pour l'Interface. Sous le terme « d'interface de programmation » se cache un logiciel qui vous permet de développer très confortablement nos programmes. Ceci s'effectue à l'aide de symboles graphiques. L'ordinateur de la ROBO Interface ne peut en fait exécuter que les instructions émanant de son jeu dit d'instruction des machines. Il s'agit, pour l'essentiel, de structures simples de contrôle d'utilisation extrêmement difficile pour des débutants. Le logiciel ROBO Pro prévoit en conséquence des éléments graphiques qui peuvent ensuite être traduits dans une langue exécutable par l'Interface.

Logiciel ROBO Pro

■ Le seul élément supplémentaire dont vous avez besoin pour la boîte de construction ROBO Mobile Set est l'Accu Set, Art. n° 34969. Il contient l'accupack en tant qu'alimentation électrique mobile pour nos maquettes de robots et un chargeur spécial pour l'accupack. Il est conseillé de charger immédiatement l'accupack avec le chargeur afin qu'il soit plein lorsque vous voudrez commencer vos expériences.

Alimentation électrique



Processus d'expérimentation

■ Nous avancerons progressivement pour faire notre entrée dans le monde fascinant des robots mobiles. Commencez par un assemblage-test simple afin de vérifier les fonctions de base de l'Interface et de l'analyse sensorielle. Vous construirez ensuite des maquettes simples auxquels des tâches définies sont attribuées et vous vous essayerez ensuite avec des systèmes de plus en plus compliqués. Si, à un moment quelconque, le développement des programmes propres devient trop compliqué et dure trop longtemps, il est possible de télécharger tout simplement les exemples de programmes livrés sur l'Interface et de faire fonctionner les robots avec eux.

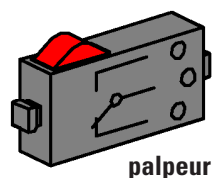
Un chapitre consacré à la recherche des pannes est inclus à la fin pour que vous ne vous désespériez pas en cas de pannes.

Le soin apporté à la construction et à la mise en service de nos robots est un point très important. Suivez exactement les instructions pour le raccordement des composants électriques et vérifiez de préférence deux ou trois fois si tout est correct. Pour les constructions mécaniques, même les créations personnelles, veillez à obtenir une souplesse et un jeu minimal des organes de commande et des fixations. Votre créativité se réserve le droit d'écrire des programmes personnels et de ce fait de définir un nouveau « comportement ». Ceci n'est limité que par la capacité de mémoire et de calcul du matériel. Les exemples qui suivent proposent quelques suggestions à cet effet.

Premières étapes

■ Après les réflexions théoriques, vous voulez maintenant commencer à effectuer quelques expériences. Certains souhaiteraient sans doute commencer immédiatement, et pourquoi pas avec le grand robot marchant. Ceci est bien entendu possible et, en suivant soigneusement les instructions de montage, la construction de la maquette réussit aussi du premier coup.

Cependant que faire lorsque cela ne fonctionne pas ? Dans ce cas, l'origine de la panne doit systématiquement être recherchée. Mais, avant de vous en préoccuper, vérifiez tout d'abord la liaison entre l'ordinateur et l'Interface.

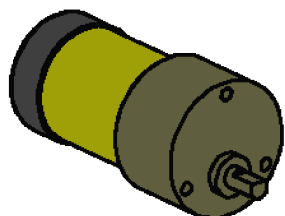


palpeur

Les chapitres 1 et 2 du manuel du logiciel ROBO Pro expliquent comment le logiciel de commande doit être installé sur l'ordinateur et comment l'Interface doit être raccordée. A l'aide du test de l'Interface, testez d'abord les différents capteurs et acteurs.

palpeur

A présent, vous pouvez, p.e. raccorder un palpeur à l'entrée numérique I1 et observer comment l'état de l'entrée se modifie lors de l'activation du palpeur.



Powermotor

Powermotor

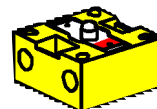
Vérifiez les sorties en reliant un moteur à une sortie Moteur, p. ex. M1. Avec la touche de gauche de la souris, vous pouvez faire tourner le moteur et modifier la vitesse au moyen du régulateur à coulisse.

Fototransistor

Si vous voulez également tester la sortie analogique AX, vous pouvez utiliser un transistor photo comme capteur analogique.

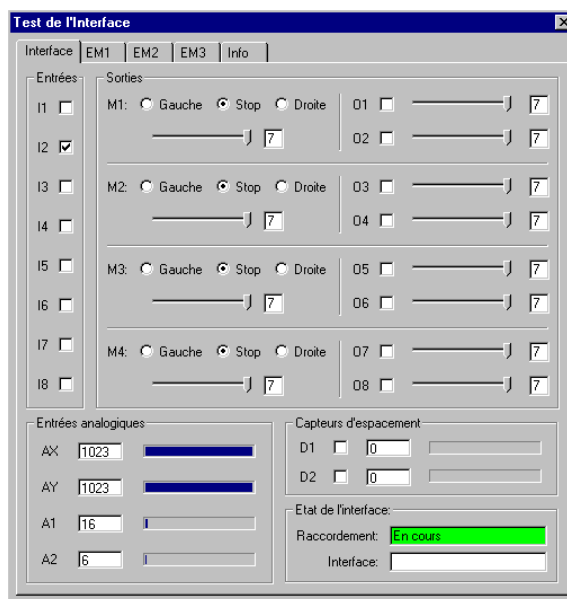
Alors que la polarité des raccordements ne joue aucun rôle pour le moteur et le palpeur (le moteur tourne au pire dans le mauvais sens), le raccordement exact du transistor photo est absolument nécessaire pour un fonctionnement correct.

Reliez le contact du transistor avec le repère rouge au moyen du pin de raccordement rouge, l'autre contact avec le pin vert. Le second pin vert est inséré dans la prise femelle de l'entrée AX la plus proche du bord de l'Interface ; le second pin rouge, dans la fiche femelle la plus éloignée de AX (Attention : si vous raccordez le transistor photo à une entrée numérique I1-I8, le pin rouge doit être inséré dans la prise femelle la plus proche du bord du boîtier).



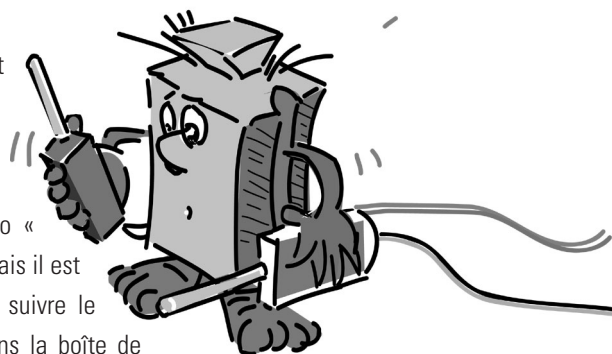
Fototransistor

Vous pouvez maintenant modifier l'éclairement du transistor photo à l'aide d'une lampe de poche et modifier ainsi la déviation de la barre bleue de AX. Si l'aiguille ne bouge pas de sa déviation maximale, vérifiez à nouveau les raccordements du transistor photo. Si, par contre, l'aiguille est aussi sur zéro avec la lampe de poche éteinte, il est possible que l'éclairage de la pièce, à savoir la luminosité ambiante, soit trop intense. La déviation se modifie alors lorsque que vous couvrez le transistor photo.

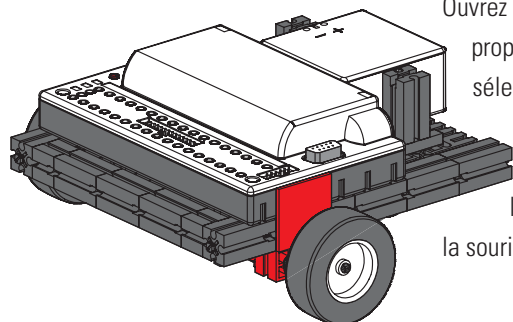


Pour revenir à nouveau brièvement sur la classification des couleurs des pins : lors de l'assemblage, veillez à toujours raccorder un pin rouge au conducteur rouge et un pin vert au conducteur vert. Lorsque la polarité exacte doit être respectée pour le montage, prenez toujours un conducteur rouge comme pôle positif et un conducteur vert comme pôle négatif. Ceci peut paraître quelque peu méticuleux, mais une classification explicite des couleurs facilite considérablement une recherche systématique des pannes.

Nous terminerons ces premiers pas dans le domaine de la robotique avec un programme simple. Le programme expliqué au chapitre 3 du manuel ROBO Pro « Commande d'une porte de garage » n'a certes rien à voir avec les robots mobiles, mais il est parfaitement approprié à l'apprentissage du logiciel ROBO Pro. Afin de pouvoir suivre le programme, il vous suffit de raccorder le moteur et les trois palpeurs inclus dans la boîte de construction ROBO Mobile Set à l'Interface. De plus amples détails sont inclus dans le manuel du logiciel.



Le premier robot simple



■ Après le test de l'Interface et la commande de la porte de garages, vous pourrez enfin mettre les premiers robots en service. Construisez la maquette "Robot simple" avec deux moteurs de commandes, conformément aux instructions de montage. Ceci est très simple et rapide car cette maquette ne contient délibérément que ce qui est nécessaire au pilotage d'un robot. Raccordez les moteurs aux sorties M1 et M2.

Ouvrez le logiciel ROBO Pro et développez un nouveau programme (FICHIER - NOUVEAU). ROBO Pro propose divers niveaux de difficulté avec lesquels vous pouvez travailler. Ils peuvent être sélectionnés dans le menu de ROBO Pro sous NIVEAU. Le niveau 1 suffit pour le moment.

Un tableau vide s'affiche ainsi que, sur le bord gauche de l'écran, la fenêtre d'élément via laquelle vous pourrez sélectionner les divers éléments du programme avec la touche gauche de la souris et les placer sur la surface de travail. Modifiez les propriétés avec la touche droite de la souris.

Exercice 1 (Niveau 1) :

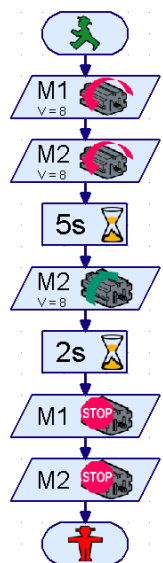
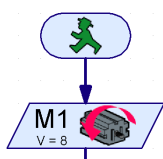
Votre "robot simple" doit aller tout droit pendant 5 secondes, puis tourner en rond pendant 2 secondes et ensuite s'immobiliser.



Astuces :

Nous programmerons les premiers robots ensemble, pas à pas :

- Commencez avec l'élément vert représentant un bonhomme en marche. Il symbolise le démarrage du programme.
- Sélectionnez ensuite le symbole du moteur dans la fenêtre d'éléments et positionnez-le sous l'élément de démarrage afin que la ligne de liaison soit automatiquement marquée. Dans la fenêtre d'éléments, réglez la sortie Moteur « M1 », ainsi que le sens de rotation « à gauche » et validez avec OK.
- Placez, de la même manière, sous ce symbole un autre élément Moteur et mettez ainsi en marche le moteur 2.
- Pour attendre pendant un certain temps, utilisez l'élément Attente, placez-le sous le second élément Moteur et réglez l'attente sur 5 secondes.
- Faites ensuite tourner le moteur M2 dans l'autre direction (vers la droite), attendez ensuite 2 secondes et coupez pour finir les deux moteurs. Notre programme se termine avec l'élément de fin, l'élément rouge représentant un bonhomme à l'arrêt. L'illustration montre le déroulement du programme terminé.



Si vous n'êtes pas certain que tout est correct, comparez votre programme avec l'exemple de programme fourni. Pour cela, le programme personnel doit préalablement être sauvegardé et le fichier Robot simple 1.rpp téléchargé à partir du répertoire d'exemples de ROBO Pro (paramètre standard C:\Programme\ROBO Pro\Programmes Exemples\ROBO Mobile Set).

Si tout est correct, le programme est chargé dans l'Interface par téléchargement. Après avoir appuyé sur le bouton Téléchargement, une fenêtre de dialogue s'ouvre. Là, indiquez que le programme doit être chargé dans la mémoire FLASH 1 et doit démarrer immédiatement après le téléchargement.

Notre maquette démarre tout de suite après le téléchargement, tourne ensuite brièvement et s'immobilise. Si vous voulez redémarrer le programme, appuyez brièvement sur la touche Prog de l'Interface. La DEL Prog1 clignote alors de nouveau tant que le programme fonctionne. Elle est ensuite allumée en continu. Le programme dans la mémoire FLASH de l'Interface reste d'ailleurs sauvegardé même en cas de coupure de l'alimentation électrique de l'Interface. Vérifiez ce détail en retirant une fiche de l'accupack. Rebranchez ensuite la fiche, sélectionnez le programme sauvegardé en appuyant sur la touche Prog jusqu'à ce que la DEL Prog 1 s'allume. Appuyez à nouveau sur la touche et lancez ainsi le programme.

Le robot ne fait pas encore grand chose, n'est-ce pas ? Elargissons donc quelque peu l'exercice.



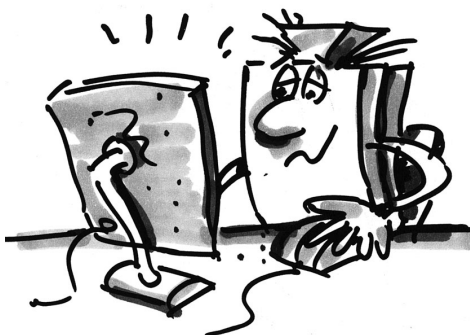
Exercice 2 (Niveau 1) :

Afin que votre robot ne s'immobilise pas déjà après 7 secondes, vous voulez maintenant lui apprendre à danser.

- **Faites-le avancer tout droit, à gauche, à droite, en arrière pendant une durée variable, et ce, à différentes vitesses.**
- **Le processus doit être répété jusqu'à la fin du programme avec la touche Prog sur l'Interface.**

Astuces :

- Inversez à nouveau la polarité des moteurs de manière à ce que le robot se déplace dans les directions souhaitées.
- Vous pouvez régler la vitesse des moteurs entre 1 et 8, dans la fenêtre de propriétés de chaque élément Moteur. Si M1 et M2 tournent à des vitesses différentes dans la même direction, le robot se déplace en formant une courbe.
- Afin de répéter continuellement le programme, tirez une ligne de liaison depuis la sortie du dernier élément de programme jusqu'à la ligne qui rentre dans le premier élément.
- Vous trouverez un exemple terminé sous [Robot simple 2.rpp](#).



Félicitations ! Vous avez construit et programmé vous-même votre premier robot personnel. Il n'est certes pas encore particulièrement intelligent, puisqu'il ne reconnaît aucun obstacle et tombe de la table lorsque vous ne faites pas attention, mais ceci va changer au cours des autres expériences.

Robots roulants intelligents

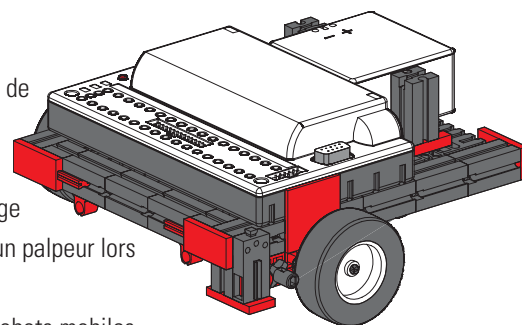
■ Pour pouvoir reconnaître leur environnement, les robots ont besoin de capteurs. Quelques variantes de robots mobiles sur lesquels l'utilisation de divers capteurs a été testée sont présentées à l'aide des propositions de maquettes suivantes. Il importe à cet effet d'associer aussi bien les états internes du robot, p. ex. la mesure de la distance parcourue par les engrenages à impulsions, que les signaux provenant de l'extérieur, p. ex. la recherche de la lumière ou de traces. Des tâches précises sont à cet effet assignées à chaque maquette. Elles doivent servir de suggestions et vous permettre de vous familiariser avec le matériel. Les programmes relatifs aux diverses tâches se trouvent dans le répertoire ROBO Pro sous \Programmes Exemples\ROBO Mobile Set\.

Mais trouvez aussi quelques tâches personnelles à assigner aux maquettes. Lorsque vous aurez terminé avec les exemples suivants, vous aurez sans doute encore plein d'autres idées.

Maquette de base

■ Par rapport à notre premier « robot simple », la maquette de base est plus stable et plus robuste. Il contient en outre 2 capteurs de mesure de distance se composant chacun d'un interrupteur et d'un engrenage d'impulsions. L'engrenage d'impulsions est relié à l'arbre du moteur et actionne 4 fois un palpeur lors de chaque rotation du moteur.

Cette maquette sert de base pour les autres maquettes de robots mobiles.



Construisez la maquette de base selon les instructions de montage. Procédez très soigneusement pour le montage. Lorsque toute la partie mécanique est terminée, vérifiez la souplesse des moteurs en reliant directement pendant une courte durée chaque moteur à l'accumulateur sans passer par l'Interface.



Exercice 1 (Niveau 1) :

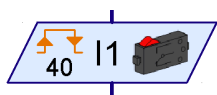
- Programmez l'Interface de manière à ce que la maquette se déplace tout droit pendant 40 impulsions. Pour mesurer les impulsions, utilisez le interrupteur sur l'entrée I1.
- Mesurez la distance parcourue par la maquette et calculez la distance parcourue par impulsion.
- Répétez l'essai 3 fois et inscrivez dans un tableau les variations des valeurs.

Astuces :

- Démarrez tout d'abord les deux moteurs (sens de rotation à gauche).
- Pour compter les impulsions au niveau de I1, utilisez l'élément de programme

Compteur d'impulsions.

- Comptez les deux types d'impulsion (0-1 en appuyant sur l'interrupteur, 1-0 en le relâchant). Vous pouvez paramétrer ceci dans la fenêtre de propriétés sous Type d'impulsion. Augmentez ainsi la précision de la mesure de la distance.
- Coupez ensuite les moteurs et terminez le programme.
- Vous trouverez le programme terminé sous Maquette de base1.rpp.



**Résultat :**

	Nombre d'impulsions	Distance parcourue	Distance/Impulsion
Essai 1	40		
Essai 2	40		
Essai 3	40		

Vous pouvez observer en gros que la maquette parcourt environ une distance d'un centimètre par impulsion.

Vous savez aussi quel sens de rotation, vous devez paramétrer pour les différentes maquettes afin que la maquette avance dans une certaine direction. Inscrivez ces informations dans le tableau suivant afin de ne pas avoir à y réfléchir à chaque modification du sens de rotation. Si vous câblez les maquettes exactement comme indiqué dans les instructions de montage, le sens de rotation à gauche signifie pour chaque moteur que la roue tourne vers l'avant. C'est ainsi que les moteurs sont programmés dans tous les programmes d'exemple.

**Complétez le tableau :**

Sens de marche de la maquette	Sens de rotation M1	Sens de rotation M2
Vers l'avant	A gauche	A gauche
Vers l'arrière		
A gauche		
A droite		
Arrêt		

Afin de ne pas devoir placer deux éléments Moteur sur l'écran lors de chaque changement de direction, vous pouvez créer pour chaque sens de marche un sous-programme qui assume cette tâche. Ceci simplifie considérablement la programmation. Le chapitre 4 du manuel du logiciel ROBO Pro explique comment créer des sous-programmes. Dès que vous aurez lu ce chapitre, vous pourrez vous risquer à faire l'exercice suivant. Passez maintenant au **niveau 2** dans ROBO Pro.

**Exercice 2 (Niveau 2):**

- Créez un sous-programme pour chaque sens de marche.
- Programmez la maquette de manière à ce qu'elle parcoure un carré d'un mètre de côté.
- Quelle est la fidélité de reproduction ?

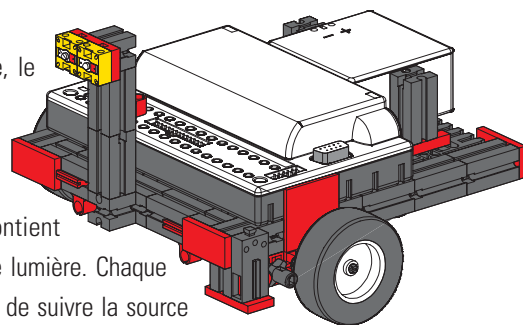


Astuces :

- Créez tout d'abord un sous-programme **Vers l'avant**. Vous pouvez créer les autres sous-programmes en copiant celui-ci. Il ne vous restera alors qu'à ajuster les sens de rotation des moteurs.
- Utilisez, pour tourner à gauche ou à droite, une vitesse réduite. Ceci augmente la précision.
- Pour compter les impulsions, utilisez à nouveau l'élément **Compteur d'impulsions** et l'interrupteur sur l'entrée I1.
- Chargez tout d'abord le programme dans la mémoire vive pour faire un essai jusqu'à ce que vous sachiez de combien d'impulsions vous avez besoin pour effectuer une rotation de 90°. Premièrement, le chargement dans la mémoire vive est plus rapide que le chargement dans la mémoire FLASH et deuxièmement la mémoire FLASH n'a qu'une durée de vie « limitée » d'environ 100.000 téléchargements.
- Le programme terminé s'appelle Maquette de base2.rpp.

**Le viseur de lumière**

■ Après votre étude approfondie de la maquette de base, le robot doit apprendre à réagir aux signaux de l'environnement. Tout comme la mite dans notre expérience mentale du premier chapitre, il doit reconnaître une source lumineuse et la suivre. La boîte de construction contient 2 transistors photo que nous utilisons comme détecteur de lumière. Chaque capteur agit à cet effet sur un moteur afin de lui permettre de suivre la source lumineuse. Le programme est composé de deux parties. La première contient la recherche d'une source lumineuse et la seconde indique comment suivre et viser cette source lumineuse. Pour cela, des sous-programmes sont à nouveau utilisés. Le sous-programme Recherche de la lumière s'active après le démarrage. Ce sous-programme n'est quitté que lorsqu'une source lumineuse a été trouvée. Le programme principal essaie alors de diriger le robot vers la source lumineuse. Lorsque la direction du robot dévie beaucoup de la ligne idéale, l'un des capteurs n'est plus illuminé par la source lumineuse. Le robot corrige alors son sens de marche afin que les deux capteurs puissent à nouveau reconnaître la source lumineuse.



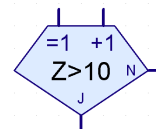
Construisez tout d'abord la maquette Viseur de lumière selon les indications des instructions de montage.

**Exercice 1 (Niveau 2) :**

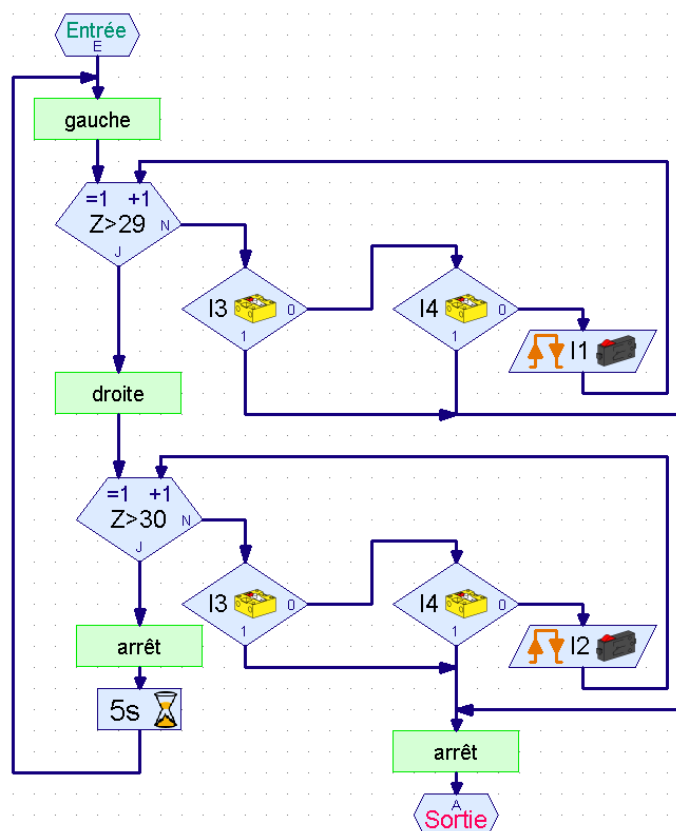
- **Programmez tout d'abord la fonction « Recherche de la lumière ».** Le robot doit pour ce faire effectuer une lente rotation de 360° minimum. Si une lumière est trouvée pendant la recherche, le robot s'arrête. Sinon, il effectue une nouvelle rotation de 360° dans l'autre direction. S'il ne trouve toujours pas de source lumineuse, il doit attendre pendant 5 secondes puis commencer une nouvelle recherche.
- **Si la recherche de la lumière est couronnée de succès, le robot doit viser la source lumineuse.** Si la source lumineuse se déplace vers la gauche ou vers la droite, le robot doit suivre les mouvements de la lumière. S'il perd le contact, le programme doit à nouveau être démarré avec la recherche de la lumière. Essayez d'attirer le robot avec une lampe de poche et d'effectuer un parcours comportant un obstacle.

Astuces :

- Utilisez, pour les différents sens de marche, les sous-programmes que vous avez déjà créés pour la maquette de base. Dès que le programme Maquette de base2.rpp s'ouvrira, vous trouverez le programme Maquette de base2 dans la **fenêtre des groupes d'éléments** de ROBO Pro sous le **programme chargé** et en dessous les sous-programmes contenus dans le programme Maquette de base2. Vous pouvez simplement ajouter ces sous-programmes dans votre nouveau programme.
- Pour le sous-programme « Recherche de la lumière », utilisez l'élément **Boucle de comptage**. (Reportez-vous au manuel ROBO Pro pour la description de l'élément).
- Dans la boucle entre la sortie « N » et la sortie « +1 », envoyez une requête aux transistors photo et comptez une impulsion sur le l'interrupteur I1. La boucle est parcourue jusqu'à ce que le robot ait trouvé une lumière ou ait effectué une rotation de 360°. Déterminez simplement combien de fois il doit parcourir la boucle pour effectuer une rotation complète et paramétrez la valeur « Z » correspondante dans l'élément Boucle de comptage.
- Programmez ensuite une seconde boucle exactement comme pour la recherche mais avec un sens de rotation inversé.
- Si le robot trouve une lumière, il s'arrête et quitte le sous-programme.
- Voici le sous-programme Recherche de lumière complet :



Boucle de comptage

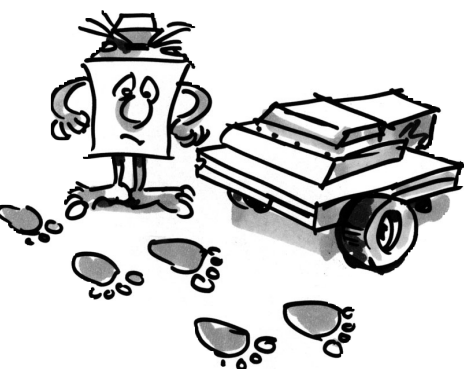


- Interrogez à nouveau les transistors photo dans le programme principal et actionnez les moteurs en fonction du transistor photo qui reconnaît la lumière :

Lumière à I3 et I4	Vers l'avant
Lumière uniquement à I3	Courbe vers la droite
Lumière uniquement à I4	Courbe vers la gauche
Aucune lumière reconnue	Arrêt, retour au sous-programme Recherche de la lumière

- créez la courbe vers la droite et vers la gauche avec des vitesses différentes de M1 et de M2 avec un sens de rotation identique. Il en résulte un style de marche très harmonieux.

- Voici à quoi ressemble maintenant le programme principal :
- Vous trouverez le programme terminé sous Recherche de la lumière.rpp.
- Utilisez une lampe de poche comme source de lumière. Essayez de ne pas focaliser le rayon lumineux trop étroitement afin que les deux photocapteurs soient illuminés par la source lumineuse. Veillez à ce que, dans les pièces très claires, votre lampe de poche ne perde pas de focalisation en raison des autres sources lumineuses, p.e. la lumière solaire entrant par une grande fenêtre. Dans ces conditions, le robot passe devant votre lampe et se dirige vers la lumière plus claire.



Le viseur de trace

■ Rechercher et suivre sont des propriétés essentielles des créatures intelligentes. Avec le viseur de lumière, vous avez créé et programmé un robot qui réagit aux signaux directs de son objectif.

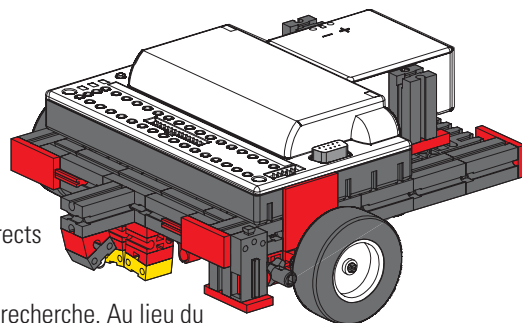
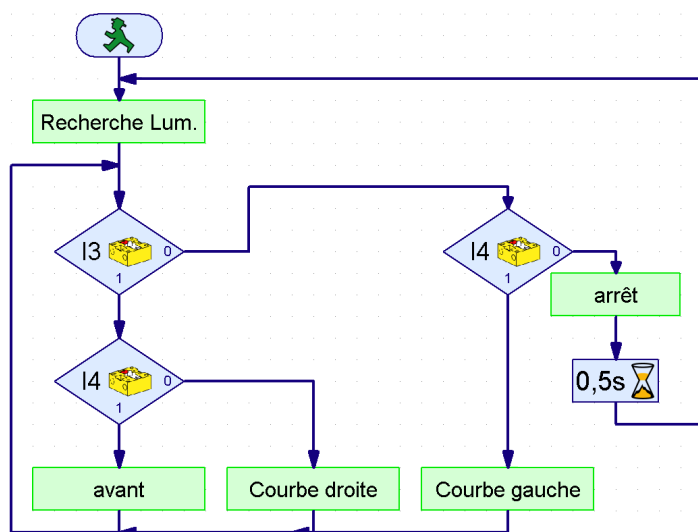
Avec le viseur de trace, nous utilisons un autre principe de recherche. Au lieu du parcours exact jusqu'à la source lumineuse, nous traçons une ligne noire qu'un robot doit suivre. Cet exercice peut être facilement résolu avec les transistors photo. La lumière réfléchie du tracé est mesurée et les moteurs sont ensuite corrigés. Afin que ceci fonctionne également correctement, la ligne est éclairée avec la lampe. Veillez à ce que les photocapteurs ne soient pas éblouis par la lumière diffusée par la lampe en raison d'une disposition défavorable. Dans ce contexte, la focalisation de la lumière de la lentille optique de la lampe à incandescence s'avère particulièrement avantageuse.

Construisez maintenant la maquette Viseur de trace selon les instructions de montage.



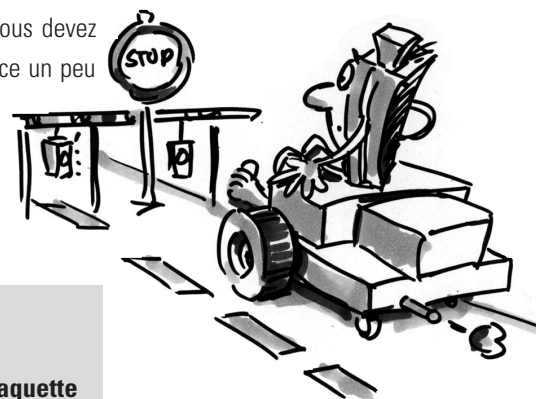
Exercice 1 :

- Créez tout d'abord un sous-programme avec lequel la trace sera recherchée. Le robot doit pour cela tourner en rond une fois.
- S'il ne trouve pas de trace, il doit avancer tout droit et effectuer une nouvelle recherche. Des requêtes sont envoyées aux transistors photo pour la reconnaissance de la trace.
- Lorsque le robot a trouvé une trace, il doit la suivre.
- Si la trace est finie ou si le robot la perd, p.e. en raison d'un grand changement de direction, la recherche doit être recommencée.



Astuces :

- Il faut attendre un court instant (env. une seconde) après l'allumage de la lampe avant d'envoyer une requête aux transistors photo. Dans le cas contraire, le transistor photo reconnaît « l'obscurité », à savoir une trace là où il n'y en a pas, parce que l'interrogation a lieu avant que la lampe soit réellement claire.
- Comme trace, utilisez un ruban isolant d'env. 20 mm de large ou dessinez avec un feutre une trace noire de cette largeur sur une feuille de papier blanc. Les courbes ne doivent pas être trop étroites, sinon le robot perd trop souvent la trace.
- Vérifiez d'abord avec le test de l'Interface si votre trace est correctement reconnue par les transistors photo. N'oubliez pas d'allumer la lampe.
- Réglez la lampe de manière à ce que les deux transistors photo produisent la valeur de 1 sur un fond clair, même lorsque les moteurs M1 et M2 sont en marche. Si votre accumulateur est un peu faible, la lampe devient un peu plus foncée lors du démarrage des moteurs. Si elle n'est pas correctement réglée, il est possible qu'un transistor photo signale de l'« obscurité » bien qu'il n'ait trouvé aucune trace.
- La recherche d'une trace fonctionne de la même façon que la recherche de la lumière. Vous devez seulement ajuster la recherche afin qu'en cas d'échec de la recherche la maquette avance un peu vers l'avant après avoir accompli un tour complet avant de continuer la recherche.
- N'oubliez pas que la maquette doit avancer tout droit pour suivre la trace lorsque les deux transistors photo produisent la valeur « Obscur » (=0).
- Vous trouverez le programme fini sous Viseur de trace.rpp.

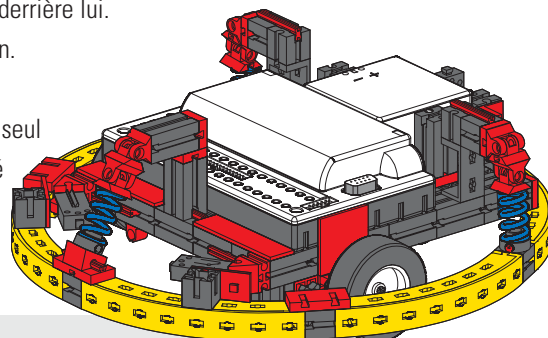
**Exercice 2 :**

- **Créez une trace avec différentes courbes étroites. Dans quel rayon la maquette arrive-t-elle encore à aller droit ?**
- **Faites l'expérience en corrigeant la trace avec différentes vitesses de M1 et M2. Quelle combinaison produit le meilleur résultat ?**
- **Créez une trace avec un tracé rond. Essayez d'optimiser les vitesses afin que le robot parcoure le cercle aussi rapidement que possible. Cet exercice convient en particulier pour un concours avec plusieurs robots.**

■ Tous les robots construits jusqu'à présent peuvent parcourir une certaine distance et suivre une source lumineuse ou une trace. Cependant, que se passe-t-il lorsqu'un obstacle se trouve sur le chemin ? Et bien, soit l'obstacle est poussé de côté, soit le robot bute vainement contre lui jusqu'à ce que l'accumulateur soit vide. Il serait bien entendu plus intelligent que le robot reconnaisse et évite l'obstacle. Pour ce faire, le robot est muni d'une tige de butée circulaire mobile avec trois palpeurs. Avec cette tige de butée, il peut détecter si un obstacle se trouve à gauche, à droite ou derrière lui. La manière dont il doit réagir alors n'est ensuite qu'une question de programmation.

Construisez tout d'abord la maquette « Robot avec reconnaissance d'obstacle ». Un seul interrupteur (I1) est nécessaire pour mesurer la distance. L'interrupteur I2 est enlevé de la maquette de base et utilisé pour la reconnaissance d'obstacle.

Robot avec reconnaissance d'obstacle





Exercice 1 (Niveau 2) :

- Le robot doit tout d'abord avancer tout droit. S'il bute à gauche contre un obstacle (E4), il doit revenir un peu en arrière puis faire une embardée vers la droite.
- S'il bute à droite contre un obstacle (E3), il doit revenir un peu en arrière puis faire une embardée vers la gauche.

Astuces :

- La reconnaissance d'obstacle en marche arrière n'est pas prise en compte pour le moment.
- Le programme principal envoie des requêtes aux interrupteurs. En fonction de l'interrupteur activé, la maquette fait une embardée vers la gauche ou vers la droite. Ceci se produit à chaque fois dans un sous-programme.
- Le nombre d'impulsions lors d'une rotation vers la droite doit être différent de celui de la rotation vers la gauche (p. ex. 3 impulsions vers la droite, 5 impulsions vers la gauche). Dans le cas contraire, la maquette risque d'aller dans un coin et de ne plus en sortir parce qu'elle tourne toujours autant vers la gauche que vers la droite.
- Le programme terminé s'appelle Obstacle1.rpp.

Il y a deux choses que le détecteur d'obstacles ne peut pas encore faire : En marche arrière, il ne reconnaît encore aucun obstacle. Lorsque les obstacles se trouvent juste devant lui, il ne les remarque pas non plus. Il pourrait cependant reconnaître ces deux choses. Si I5 est activé pendant la marche arrière, un obstacle se trouve derrière la maquette. Si I3 et I4 sont simultanément activés pendant la marche avant, un obstacle se trouve juste devant la maquette. Dans ce cas, le robot pourrait tout de suite tourner sur 90°. Les possibilités auxquelles le robot doit maintenant réagir sont les suivantes :

Obstacle	Palpeur	Réaction
à droite	Uniquement I3	Faire une embardée vers la gauche (rotation d'env. 30°)
à gauche	Uniquement I4	Faire une embardée vers la droite (env. 45°)
devant	I3 et I4	Faire une embardée vers la gauche (env. 90°)
derrière	I5	Uniquement interrogé en marche arrière. S'arrêter puis faire une embardée comme prévu

Quelques nouveaux éléments du programme, comme p. ex. les opérateurs (p. ex. ET, OU, = du ROBO Pro Niveau 3) peuvent vous être très utiles pour résoudre cet exercice avec facilité. Le niveau 3 permet également d'échanger des données entre divers éléments via les flèches orange. Passez, en conséquence, à ce niveau dans le logiciel. Prenez ensuite le manuel ROBO Pro et lisez attentivement le chapitre 5. Vous êtes alors prêt pour l'exercice suivant.



Exercice 2 (Niveau 3) :

- Modifiez le programme Obstacle afin que la maquette réagisse selon les indications du tableau ci-dessus.
- Utilisez pour ce faire les possibilités du Niveau 3 de ROBO Pro.

Astuces :

- Des requêtes sont envoyées aux différentes combinaisons d'interrupteurs dans le sous-programme « Interrogation obstacle » à l'aide des opérateurs. Le sous-programme dispose d'une sortie pour chaque possibilité.

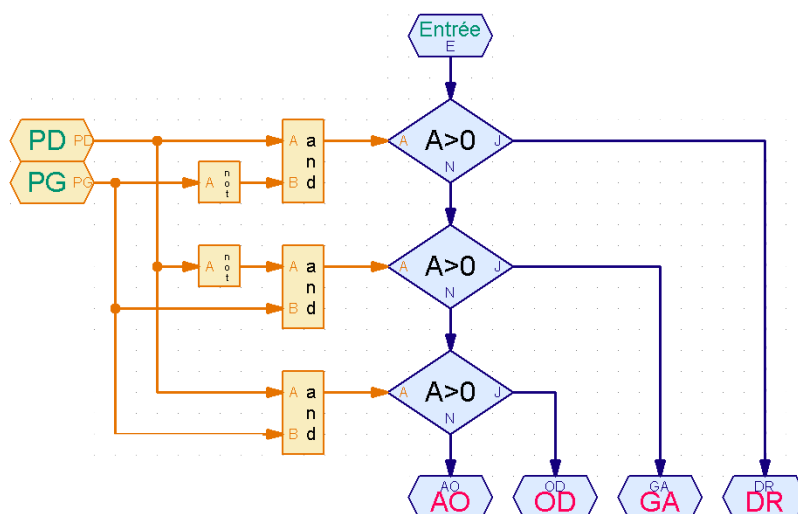


Sortie A0 = aucun obstacle

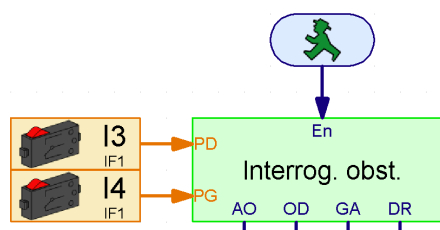
Sortie OD = obstacle devant

Sortie GA = obstacle à gauche

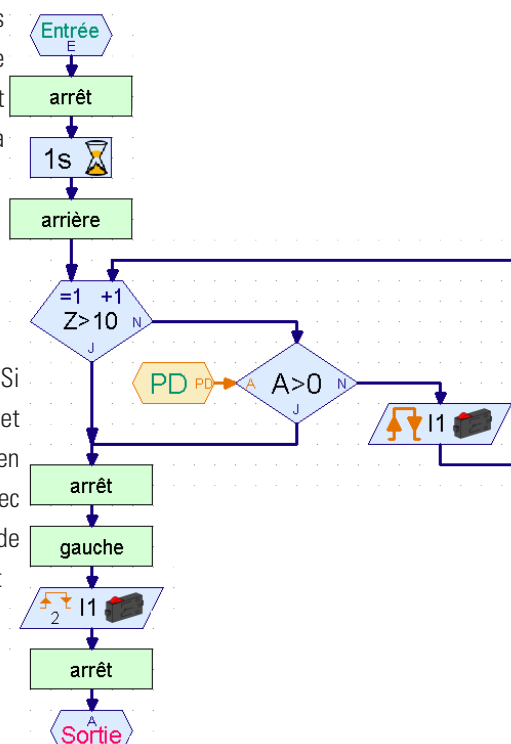
Sortie DR = obstacle à droite



- Pour reconnaître immédiatement le palpeur à interroger, placez les éléments orange de l'interrupteur dans le programme principal et associez-les en saisissant les données dans le sous-programme.



- Une requête est envoyée à L5 pendant la marche arrière dans les différents sous-programmes d'évitement. La maquette recule jusqu'à ce que le nombre d'impulsions défini soit atteint ou que I5 soit pressé. I5 est remis dans le programme principal afin savoir immédiatement dans à quel sous-programme une requête doit être envoyée.
- Vous trouverez le programme complet sous `Obstacle2.rpp`.



Viseur de lumière avec reconnaissance d'obstacle



■ Les possibilités offertes par le ROBO Mobile Set sont loin d'être épuisées.

C'est pourquoi les deux fonctions Recherche de la lumière et Reconnaissance d'obstacle doivent à présent être combinées. D'un point de vue scientifique, le robot dispose alors de deux modes de comportement. Cependant, dans la mesure où les deux types de comportement peuvent être simultanément actifs, des priorités différentes leur sont attribuées. Le robot est normalement réglé sur la recherche de la lumière. S'il reconnaît un obstacle, donc un danger pour lui, le comportement Evitement d'un obstacle s'active. Lorsque tout est dans la zone verte, le robot peut reprendre la recherche de la source lumineuse.

Lorsque les concepteurs professionnels de logiciels se mettent à une tâche aussi ambitieuse, ils ne programment pas à la légère, mais ils emploient une certaine stratégie pour développer le programme.

L'une de ces méthodes s'appelle le « projet Top-Down ». Pour ce type de procédure, tout le système est défini du haut vers le bas, sans se préoccuper de tous les détails, du moins au début. Nous emploierons également cette méthode pour résoudre ce problème.

Exercice 1 (Niveau 3) :

Apprenez les comportements suivants au robot :

- Recherchez une source lumineuse.
- Dès que vous l'avez trouvée, suivez-là.
- Si un obstacle se trouve sur son chemin, évitez-le.
- Recherchez ensuite à nouveau une source lumineuse

Pour le résoudre, utilisez les éléments du programme de ROBO Pro Niveau 3.

Résolvez l'exercice « du haut vers le bas » selon la méthode Top-Down.

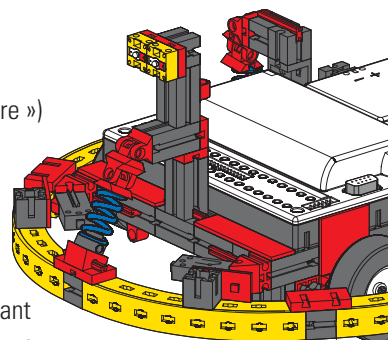


Astuces :

Divisez l'exercice en trois parties :

- Demandez si le robot voit une source lumineuse (sous-programme « Lumière »)
- Demandez s'il bute contre un obstacle (sous-programme « Obstacle »)
- En fonction de ces résultats, indiquez au robot ce qu'il doit faire (sous-programme « Avancer »)

Pour les sous-programmes « Lumière » et « Obstacle », réfléchissez maintenant aux différentes situations que le robot peut percevoir. Attribuez, à chaque situation, une valeur numérique que vous sauvegarderez dans une variable à l'aide d'un élément de commande. Chaque situation engendre alors une réaction qui est exécutée dans le sous-programme « Avancer ».



Sous-programme Lumière :

N°	Situation	Etat des capteurs	Réaction
0	Aucune source lumineuse présente	I6=0; I7=0	Rechercher la lumière
1	Source lumineuse juste devant le robot	I6=1; I7=1	Avancer tout droit
2	Source lumineuse à gauche du robot	I7=1	Tourner sur la gauche
3	Source lumineuse à droite du robot	I6=1	Tourner sur la droite

Sous-programme Obstacle :

N°	Situation	Etat des capteurs	Réaction
4	Obstacle directement devant le robot	I3=1; I4=1	Faire une embardée de 90°
5	Obstacle à droite du robot	I3=1	Faire une embardée sur la gauche
6	Obstacle à gauche du robot	I4=1	Faire une embardée sur la droite

Il ne vous reste maintenant plus qu'à représenter ces connaissances dans ROBO Pro avec les éléments du programme.

Sous-programme Lumière :

PD= transistor photo de droite
PG= transistor photo de gauche

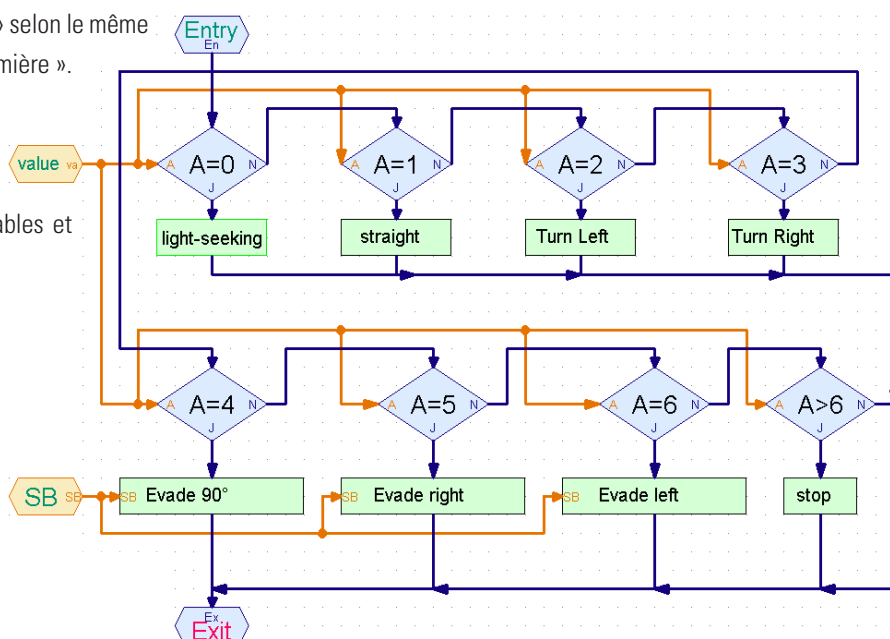
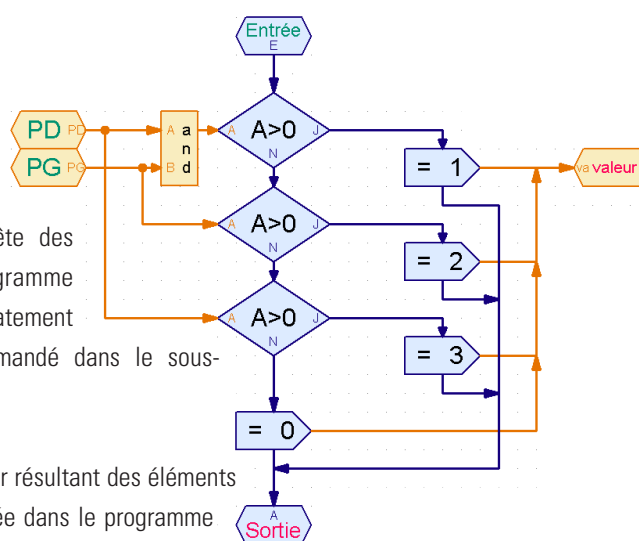
Remettez les éléments de requête des transistors photo dans le programme principal afin de pouvoir immédiatement reconnaître ce qui doit être demandé dans le sous-programme.

La variable qui sauvegarde la valeur résultant des éléments de commande est également placée dans le programme principale. Elle est utilisée dans plusieurs sous-programmes. Associez-la au sous-programme via une sortie de données.

Créez le sous-programme « Obstacle » selon le même principe que le sous-programme « Lumière ».

Dans le sous-programme « Avancer », demandez, au moyen des éléments de branchement, la valeur actuelle des variables et programmez la réaction appropriée :

PD= Interrupteur derrière



Pour finir, vous devez encore créer les sous-programmes qui sont utilisés dans ce sous-programme. Mais, une minute ! Il existe déjà presque tous. Vous pouvez p.e. copier le sous-programme Recherche de la lumière dans le programme de la maquette Viseur de lumière. Si vous ne savez plus comment faire, relisez le chapitre 4 du manuel ROBO Pro.

Mais attention :

Pour la maquette Viseur de lumière, les transistors photo I3 et I4 étaient raccordés. Mais maintenant ils se trouvent sur I6 et I7. En outre, pour le comptage des impulsions, l'interrupteur I1 a été interrogé lors d'une rotation vers la gauche et l'interrupteur I12 pour une rotation vers la droite. Maintenant, il ne reste plus que I1 pour compter les impulsions, ce qui en fait fonctionne tout aussi bien. Vous devez donc ajuster le sous-programme Recherche de la lumière après l'avoir copié. La requête envoyée aux interrupteurs étant cachée dans le sous-programme, il est facile de pas la voir. Ceci ne se produit plus si vous placez les entrées dans le programme principal et que vous les associez avec le sous-programme en saisissant les données. Mais, pour le viseur de lumière, vous ne connaissiez pas encore cette possibilité.

Les sous-programmes d'évitement existent déjà également, notamment pour la maquette Reconnaissance d'obstacle. L'interrupteur I5 qui est aussi interrogé pour la marche arrière est sorti.

Vous pouvez regarder le programme fini sous [Obstacle-Lumière.rpp](#).

Le programme principal a l'air très clair et très simple au premier regard. Et pourtant, il y a de quoi s'arracher les cheveux dans les sous-programmes. Mais, en procédant progressivement avec la méthode Top-Down, vous auriez pu aussi résoudre un problème complexe.

Au fait... Si vous avez un ami qui possède aussi une boîte de construction ROBO Mobile Set, vous pouvez pousser l'expérimentation encore plus loin. Une source lumineuse est simplement installée sur chacun des deux robots. Les robots se cherchent alors mutuellement.



Robot avec reconnaissance d'arête

■ Après avoir vu dans le dernier exemple comment procéder lors de la programmation d'un problème complexe, vous pouvez maintenant vous consacrer à un autre comportement très important d'un robot mobile. Il doit notamment apprendre à ne pas tomber d'une table. Buter contre un obstacle ne fait, dans la plupart des cas, aucun effet sur le robot. Cependant, tomber d'une table d'une hauteur de presque un mètre pourrait l'endommager d'une manière ou d'une autre, bien que les modules Fischertechnik soient très solides. C'est pour cette raison que le robot est muni de capteurs lui permettant de reconnaître les arêtes. Ces détecteurs d'arêtes sont composés d'un interrupteur qui est actionné par un engrenage mobile à paliers. Cet engrenage peut aussi se déplacer vers le haut et vers le bas. Dès que l'engrenage dépasse du bord de la table, il s'abaisse, l'interrupteur n'est plus actionné, le programme reconnaît que la maquette se trouve dans une abîme et réagit en conséquence. Le robot possède au total 4 détecteurs d'arêtes afin de pouvoir palper les abîmes sur les deux côtés en marche avant, comme en marche arrière. En revanche, cette maquette n'a pas de capteur d'impulsions pour mesurer la distance. La distance parcourue est commandée au moyen de la durée de fonctionnement des moteurs. Construisez tout d'abord la maquette selon les instructions de montage.

Vérifiez avec précision que les détecteurs d'arêtes se déclenchent correctement :

- lorsque la maquette s'approche du bord de la table et que l'interrupteur est à nouveau pressé,
- lorsque la roue se trouve à nouveau sur la table.

Vous devez éventuellement ajuster l'un ou l'autre des interrupteurs vers le haut ou vers le bas.



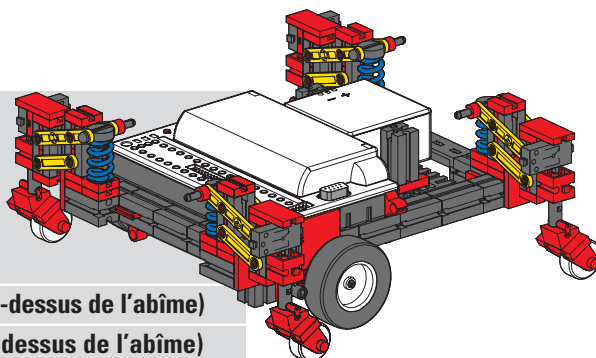
Exercice 1 (Niveau 3) :

- Réfléchissez tout d'abord à la façon dont le robot doit réagir lorsqu'il arrive dans une abîme.
- En réfléchissant davantage, vous vous rendrez compte qu'il existe de nombreuses possibilités de combinaisons dans lesquelles les capteurs pourraient se trouver dans une abîme. Il est possible qu'un des 4 détecteurs se déclenche ainsi que 2 ou 3, voire les 4 interrupteurs simultanément.
- Comment le robot doit-il à chaque fois réagir ?

Astuces :

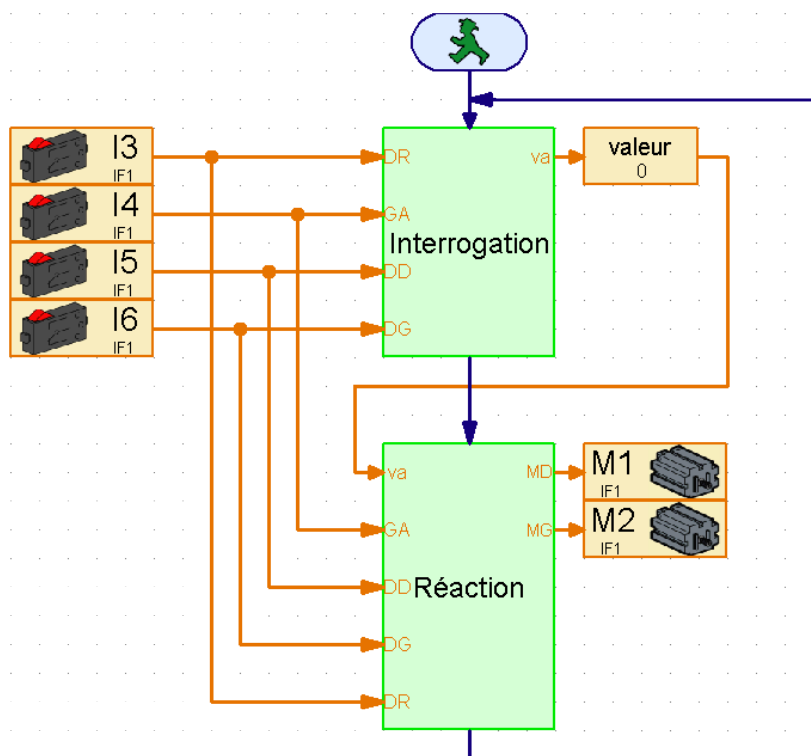
Vous trouverez la solution dans le tableau suivant. Les capteurs qui se trouvent au-dessus de l'abîme (palpeur=0) sont cochés. Un nombre est attribué à chaque combinaison. Le chiffre correspondant est attribué à chaque possibilité dans le programme qui sera créé ultérieurement. Le robot réagit à la situation présente à l'aide du nombre. Mais nous y reviendrons plus en détail. Réfléchissez tout d'abord uniquement à la façon dont se tient le robot afin que la combinaison correspondante se produise et qu'il réagisse correctement.

N°	Devant à droite (I3)	Devant à gauche (I4)	Derrière à droite (I5)	Derrière à gauche (I6)	Réactions
0					Avancer (aucun capteur au-dessus de l'abîme)
1	●	●	●	●	Arrêter (les 4 capteurs au-dessus de l'abîme)
2	●	●	●		Tourner légèrement vers la droite
3	●	●		●	Tourner légèrement vers la gauche
4	●		●	●	Tourner légèrement vers la gauche
5		●	●	●	Tourner légèrement vers la droite
6	●	●			Reculer puis tourner vers la droite
7	●		●		Tourner légèrement vers la gauche
8	●			●	Tourner légèrement vers la gauche
9		●	●		Tourner légèrement vers la droite
10		●		●	Tourner légèrement vers la droite
11			●	●	Avancer légèrement
12	●				Reculer puis tourner vers la gauche
13		●			Reculer puis tourner vers la droite
14			●		Avancer légèrement
15				●	Avancer légèrement



Assez violent, n'est-ce pas ? Mais n'ayez crainte, il existe pour cette maquette un programme complet qui utilise tous les avantages de ROBO Pro. Il s'appelle Arêtes.rpp.

Les éléments les plus importants se trouvent dans le programme principal afin que vous puissiez comprendre l'ensemble du processus. Les requêtes complexes envoyées aux interrupteurs et la commande des moteurs sont cachées dans les sous-programmes. Voici tout d'abord le programme principal :



Le déroulement commence par l'interrogation des 4 interrupteurs. Voyez, à l'extrémité gauche, les interrupteurs auxquels une requête est envoyée. Ils sont associés au sous-programme via les entrées des données. Le sous-programme « Interrogation » détermine les interrupteurs qui sont pressés et donne la valeur indiquée dans le tableau. Des variables de même nom que vous pouvez reconnaître dans le programme principal sont attribuées à cette valeur. La valeur des variables est transmise au sous-programme « Réaction » qui commande ensuite les deux moteurs en fonction de cette valeur. Les interrupteurs sont également insérés dans le sous-programme « Réaction », car les capteurs d'arêtes sont aussi interrogés pendant que la maquette fait une embardée.

Vous pourriez maintenant modifier sans problèmes l'affectation des palpeurs sur l'Interface, ainsi que les sorties Moteurs sans devoir faire une recherche dans tous les sous-programmes dans lesquels un élément

d'entrée ou un symbole de moteur s'est caché. On ne rencontre qu'une seule fois chaque entrée et chaque sortie.

Vous pouvez avant tout utiliser cette technique de programmation lorsqu'un sous-programme doit être utilisé pour de nombreuses maquettes différentes et lorsque vous ne connaissez pas exactement les entrées et les sorties qu'il convient d'utiliser pour cela dans l'Interface.

Si votre curiosité est maintenant excitée, parcourez simplement les sous-programmes et essayez de les comprendre. Le principe de programmation est similaire à celui de la maquette « Viseur de lumière avec reconnaissance d'obstacle ».



Exercice 2 (Niveau 3) :

Chargez le programme sur l'Interface et faites avancer le robot sur une table.

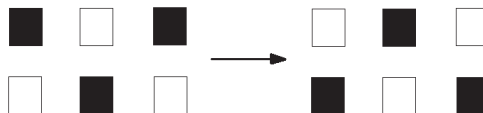
- La maquette réagit-elle toujours correctement ?
- Devrait-elle se comporter différemment avec certaines combinaisons de touches ?
- Optimisez, si nécessaire, le programme.

■ Après nous être occupés dans les détails des robots roulants, passons à une autre forme de déplacement que nous pouvons utiliser pour les robots mobiles, à savoir la marche.

L'allure des insectes convient à la perfection comme maquette de « créature mécanique à six pattes ». Pour marcher sur trois pattes, trois des six pattes se soulèvent simultanément du sol, la patte avant et la patte arrière d'un même côté en même temps que la patte du milieu de l'autre côté :

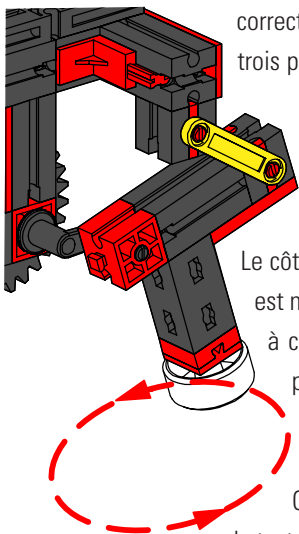
Marche sur trois pattes

Les pattes qui restent au sol (représentées en noir) forment un trépied stable afin que la maquette soit toujours debout en toute sécurité et ne se reverse pas en marchant.



Les pattes du robot marchant Fischertechnik sont conçues de manière à constituer un quadrilatère articulé. La forme de construction du quadrilatère articulé utilisé ici s'appelle un « mécanisme à manivelle et bielle oscillante ». Propulsés par une manivelle, les éléments mobiles de l'organe de commande effectuent des mouvements oscillants. Les écartements entre les diverses articulations et la position de la base (à savoir l'extrémité inférieure de la patte) sont choisis de manière à ce que la base décrive un mouvement elliptique lorsque la manivelle de commande tourne. Il en résulte un mouvement qui ressemble à un pas effectué en marchant.

Les 6 manivelles qui commandent les pattes doivent être exactement ajustées selon les indications des instructions de montage. Les trois pattes qui reposent en même temps sur le sol ont la même position de manivelle. Les manivelles des 3 pattes qui sont alors soulevées tournent sur 180°. La position correcte des manivelles l'une par rapport à l'autre garantit que la maquette peut marcher sur trois pattes à la cadence correcte.



Les contre-écrous avec lesquels les roues dentées sont fixées sur les axes doivent être correctement serrés afin que les manivelles ne se déplacent pas pendant la marche.

Le côté droit et le côté gauche de la maquette sont chacun commandés par un moteur (ceci est nécessaire pour le fonctionnement des manivelles). C'est pourquoi il convient de veiller à ce que la patte centrale d'un côté soit toujours dans la même position que les deux pattes extérieures de l'autre côté. Cette synchronisation est commandée par ordinateur via les palpeurs I1 et I2.

Construisez tout d'abord la maquette selon les instructions de montage. Vérifiez, avec le test de l'Interface, que tous les interrupteurs et moteurs sont correctement raccordés. Sens de rotation des moteurs : sens de rotation à gauche = vers l'avant.



Exercice 1 (Niveau 1) :

Apprenez au robot à marcher.

- Programmez la maquette afin qu'elle marche tout droit sur trois pattes.
- Utilisez les interrupteurs I1 et I2 pour synchroniser les pattes gauches et droites.
- Veillez pour cela à ce que les deux pattes extérieures d'un côté et que la patte centrale de l'autre côté soient toujours dans la même position.

Le robot marchant



Astuces :

- Mettez tout d'abord les pattes du côté gauche et du côté droit dans leur position initiale. Mettez les deux moteurs en marche (sens de rotation vers la gauche).
- Le processus ne doit se poursuivre que lorsque les deux interrupteurs I1 et I2 ne sont pas pressés (cette interrogation est nécessaire dès que la maquette veut effectuer le second pas).
- Laissez les moteurs tourner jusqu'à ce que l'interrupteur approprié (I1 pour M1 et I2 pour M2) soit à nouveau pressé. Il est important à ce effet que la maquette ne commence le pas suivant que lorsque les deux interrupteurs sont pressés. Les pattes sont alors dans la position correcte l'une par rapport à l'autre. La condition préalable est que les manivelles qui commandent les pattes soient correctement ajustées, comme indiqué dans les instructions de montage.
- Le processus peut maintenant recommencer du début et le robot fait le deuxième pas. La maquette marche alors tout droit jusqu'à ce que le programme s'arrête.
- Vous trouverez le programme complet sous [Robot marchant1.rpp](#).

Tout comme pour la maquette roulante de base, vous pouvez faire marcher la maquette vers la gauche, vers la droite ou en arrière en modifiant les sens de rotation des moteurs. Pour compter les pas, vous pouvez utiliser I1 ou I2.

**Exercice 2 (Niveau 2) :**

- Programmez votre maquette afin qu'il fasse 10 pas vers l'avant, 3 pas vers la gauche, 3 pas vers la droite et à nouveau 10 pas en arrière.
- Créez un sous-programme pour chaque direction.
- Utilisez l'élément Boucle de comptage pour compter les pas.

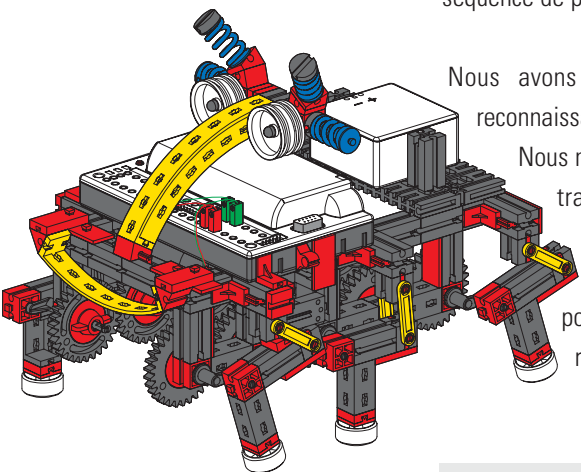
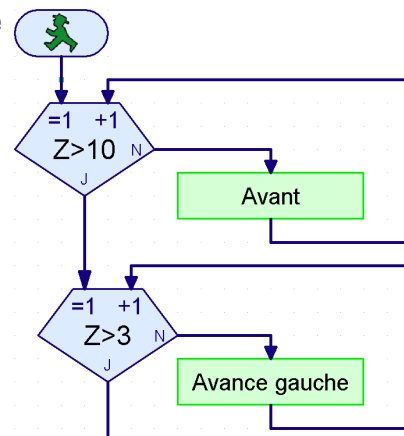
Astuces :

- Copiez simplement le programme Robot marchant1.rpp dans un sous-programme.
- Copiez ce sous-programme autant de fois que nécessaire pour les différents sens de marche. Modifiez les sens de rotation des moteurs dans chaque sous-programme afin que la maquette se déplace dans la direction souhaitée.
- Utilisez l'élément Boucle de comptage pour compter le nombre de pas pour chaque sens de rotation. La maquette fait un pas à chaque déroulement d'un sous-programme. Si le programme parcourt la boucle 10 fois avec le sous-programme, la maquette fait 10 pas.

Vous pouvez ainsi apprendre à votre robot marchant à effectuer une séquence de pas quelconque ([Robot marchant2.rpp](#)).

Nous avons déjà traité en détail le sujet de la reconnaissance d'obstacle pour les robots roulants.

Nous ne voulons pas y revenir ici. Mais, essayez de transposer ce comportement au robot marchant. Les capteurs nécessaires sont inclus dans la boîte de construction. vous pouvez, pour la programmation, prendre le robot roulant comme modèle. Bonne chance !



■ La ROBO Interface offre davantage de fonctionnalités que celle dont les robots mobiles disposaient jusqu'à présent. Des composants supplémentaires qui ne sont pas inclus dans la livraison de la boîte de construction sont disponibles. Cependant, comme ils sont extrêmement intéressants pour les robots, nous voulons vous les présenter brièvement ici.

■ La ROBO Interface contient une diode de réception à infrarouge pour l'émetteur manuel du IR Control Set Art. n° 30344. Dans le logiciel ROBO Pro, vous pouvez ainsi interroger les touches de l'émetteur manuel comme des entrées numériques et ainsi démarrer et arrêter les moteurs.

Nous avons programmé, en guise d'exemple de programmation, une commande de fenêtre pour le robot marchant. Vous pouvez commander la maquette pour qu'elle se déplace vers l'avant, vers l'arrière, vers la gauche et vers la droite à l'aide des 4 flèches ovales. Vous devez juste charger préalablement le programme IR-Robot marchant.rpp sur l'Interface.

Le programme Mobile-Teach-IR.rpp est un autre programme génial en liaison avec la commande à distance. Avec ce programme Teach-In, vous pouvez télécommander l'un des robots roulants, p.e. le robot simple ou la maquette de base. La maquette retient alors la distance parcourue et peut la reproduire à volonté. La distance sauvegardée est toutefois effacée lorsque le programme s'arrête.

Un tel programme est possible grâce à l'élément de programme « Liste » dans ROBO Pro. On peut sauvegarder dans cet éléments de nombreuses valeurs et les appeler (reportez-vous également au manuel ROBO Pro). Le programme lui-même est certes assez complexe, mais son application est très simple :

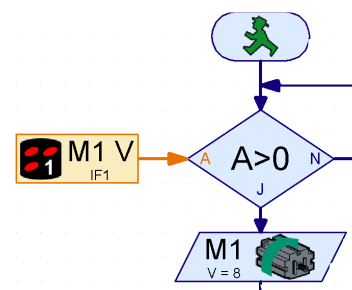
1. Chargez et démarrez le programme Mobile-Teach-In.rpp dans la mémoire Flash de la ROBO Interface.
2. Appuyez sur la touche de commande à distance **M1** ▶ / ▶▶. Le « processus d'apprentissage » démarre.
3. Avec les flèches ovales, orientez la maquette dans la direction souhaitée.
4. Appuyez sur la touche **M2** ▶ / ▶▶. La distance parcourue est sauvegardée.
5. Appuyez sur la touche **M3** ▶ / ▶▶. La distance sauvegardée est parcourue.

Grâce à cette application, programmer des robots devient un jeu d'enfant. Vous ne devez pas oublier que la distance sauvegardée est effacée dès que le programme est arrêté avec le palpeur Prog de l'Interface.

■ L'interface radio ROBO RF Data Link Art. N° 93295 remplace le câble d'Interface entre l'ordinateur et l'Interface par une transmission hertzienne des données. Ceci est très astucieux. Premièrement, vous ne devez pas brancher et débrancher le câble à chaque fois que vous chargez un programme sur l'Interface. Deuxièmement, vous pouvez exécuter le programme sans câble en mode Online et trouver ainsi plus facilement les erreurs qu'en mode de téléchargement. Et, enfin, les robots mobiles peuvent être commandés en mode Online via le panneau de commande de ROBO Pro, de la même façon qu'avec la commande à distance IR Online via l'écran. Contrairement à la commande à distance, on peut également visualiser sur l'écran les données que l'Interface fournit, p.e. les valeurs des variables ou des entrées analogiques, la tension d'alimentation fournie par l'accumulateur, la vitesse des moteurs.

Possibilités d'extension

Emetteur manuel à infrarouge



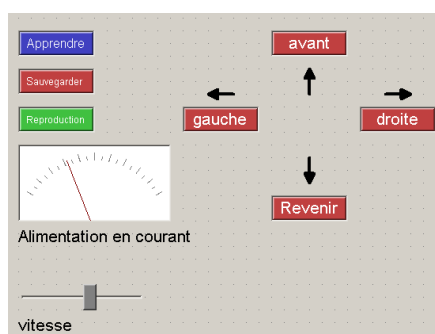
ROBO RF Data Link



Nous avons, comme exemple, transformé le programme Teach-In et commandé le robot roulant via un panneau de commande. Le programme s'appelle Mobile-Teach-RF.rpp. Vous pouvez, bien entendu, l'essayer avec le câble d'Interface. Mais ceci est assez inconfortable. Le rayon d'action de la maquette est limité, le câble s'emmêle et le robot ne tourne plus correctement. Il est certain que vous sortirez ensuite pour acheter la RF Data Link.

Chargez le programme Mobile-Teach-RF.rpp.

Passez dans la barre de fonctions du programme principal sur le panneau de commande. Démarrez ensuite le programme en mode Online. Vous pouvez maintenant commander et programmer la maquette via les boutons du panneau de commande.



1. Appuyez sur la touche « Apprendre ». Le « processus d'apprentissage » démarre.
2. Orientez la maquette dans la direction souhaitée avec les flèches.
3. Appuyez sur la touche « Sauvegarder ». La distance parcourue est sauvegardée.
4. Appuyez sur la touche « Reproduction ». La distance sauvegardée est parcourue.

Le trajet sauvegardé est également effacé lorsque le programme s'achève.

Vous en apprendrez davantage sur la création des panneaux de commande dans le manuel ROBO Pro.

ROBO I/O-Extension

■ Si vous construisez une maquette avec tellement de capteurs et de moteurs de manière que les sorties et les entrées de la ROBO Interface ne suffisent plus, vous pouvez connecter une ROBO I/O-Extension Art. N° 93294 à l'Interface. Vous disposez ainsi de 8 entrées numériques, 4 sorties Moteurs et d'une entrée de résistance analogique supplémentaires. Vous pouvez raccorder un second module sur cette extension I/O et, sur ce second module, vous pouvez encore raccorder un troisième module qui sont tous commandés par une ROBO Interface. Vous disposerez alors d'un total de 16 sorties Moteurs, 32 entrées numériques, 5 entrées de résistance analogiques, 2 entrées de tension analogiques et 2 entrées pour les capteurs de distance.

Si vous n'en avez toujours pas assez, vous pouvez aussi commander sur l'ordinateur plusieurs Interfaces en mode Online, p.e. une interface série COM, une interface USB ou 2 interfaces sur le bus USB et chacun avec jusqu'à 3 ROBO I/O-Extensions ! Cela peut facilement faire tourner la tête. Le chapitre 6 du manuel ROBO Pro explique également comment tout cela fonctionne.

Recherche des pannes

■ Faire des expériences est amusant. Et il est encore plus amusant que tout fonctionne, ce qui est la plupart du temps le cas, mais malheureusement pas toujours.

Ce n'est que lorsqu'une maquette ne fonctionne pas que l'on découvre si l'on a correctement compris le mécanisme et que l'on peut tout de suite trouver la panne.

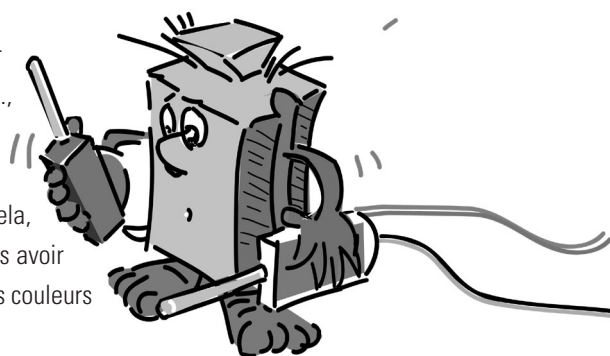
Pour les pannes mécaniques, on arrive encore à voir (montage incorrect) ou à sentir (dureté) quelque chose. Mais quand des problèmes électriques s'y ajoutent, cela devient plus difficile.

Pour rechercher les pannes, les professionnels utilisent toute une série d'instruments de mesure divers, comme p.e. un tensiomètre ou un oscillographe. Tout le monde ne dispose pas de tels appareils. Nous voulons en conséquence essayer de détecter et de réparer une panne par des moyens simples.

Câblage

Avant de commencer nos expériences, vous devez tout d'abord achever quelques composants de la boîte de construction Fischertechnik. Il faut, p.e., raccorder les connecteurs livrés aux diverses sections de câble.

Le câble doit tout d'abord être coupé. Mesurez les longueurs indiquées et découpez les sections. Chaque câble doit être mesuré après la finition. Pour cela, vous avez besoin de l'accumulateur et de la lampe. Si la lampe s'allume après avoir été reliée à l'accumulateur, le câble convient. Vérifiez aussi si l'affectation des couleurs est correcte, pin rouge avec câble rouge, pin vert avec câble vert.



Test de l'Interface

Si le programme (même celui fourni) ne fonctionne pas avec votre maquette, démarrez le test de l'Interface. Ce programme d'aide vous permet de tester séparément les entrées et les sorties. Les capteurs fonctionnent-ils ? Les moteurs tournent-ils dans le sens correct ? Les moteurs de tous vos robots mobiles sont raccordés de manière à ce que, dans le sens de rotation vers la gauche, la roue ou la patte se déplace vers l'avant. Si là aussi tout est correct, recherchez la cause mécanique.

Mauvais contacts

Les mauvais contacts sont des pannes fâcheuses. Les pins de raccordement peuvent, d'une part, être mobiles dans les douilles. Si tel est le cas, les ressorts de contact des connecteurs doivent être un peu élargis à l'aide d'un petit tournevis. Attention, un trop grand élargissement peut provoquer la rupture des contacts ou rendre difficile l'introduction dans la prise.

Les jonctions par serrage relâchées au point de fixation des connecteurs peuvent être une autre cause de mauvais contacts. Resserrer-les avec précaution ! Il faut vérifier en même temps si aucun mince fil de cuivre n'est cassé.

Courts-circuits

Il peut aussi arriver que l'on provoque un court-circuit si un câble est mal posé. Rien ne fonctionne alors plus comme cela devrait le faire. L'accupack contient un fusible qui coupe le courant en cas de surintensité de courant ou de température trop élevée. Les sorties de l'Interface sont également fermées en cas de surchauffe.

Un court-circuit peut aussi se produire si la petite vis de la prise électrique avec laquelle le câble est fixé n'est pas serrée suffisamment sur la prise électrique. Elle dépasse alors du bord de la prise. Si l'on branche deux connecteurs dans deux douilles situées côte à côte sur l'Interface et que les petits vis sont

en contact, un court-circuit se produit. C'est pourquoi les petites vis doivent toujours être bien serrées et les connecteurs branchés de manière à ce que les petites vis ne soient pas en contact.

Alimentation électrique

En cas de raté inexpliqué pendant le fonctionnement, la cause peut pratiquement toujours être attribuée à un accumulateur vide. La tension chute brièvement lors de la mise sous tension d'une charge (moteur allumé), ce qui déclenche une remise à zéro du processeur de l'Interface. La ROBO Interface indique, par l'allumage de la DEL rouge, que la tension de l'alimentation électrique est trop basse. L'accumulateur doit alors être rechargé.

Erreur de programmation

Si des pannes que l'on ne peut pas expliquer surviennent pour les programmes décrits, il convient, par mesure de précaution, d'exécuter un programme aussi identique que possible aux programmes fournis afin de pouvoir exclure des défauts électriques ou mécaniques. En mode Online, il est possible de suivre le déroulement du programme sur l'écran. Si le programme s'arrête à un certain endroit, la cause peut alors être recherchée ici, p.e. sélection incorrecte de l'entrée ou du moteur, valeur incorrecte interrogée lors d'un branchement ou inversion des raccordements O/N.

Si tout ceci ne sert à rien, il ne vous reste qu'à prendre contact avec le service après-vente Fischertechnik (email : info@fischertechnik.de).

Vous pouvez aussi visiter notre site Internet, sous www.fischertechnik.de. Vous y trouverez un forum, un chat, un marché, une galerie et vous pourrez gratuitement devenir membre du fan club Fischertechnik.

Nous vous souhaitons de vous amuser longtemps avec le ROBO Mobile Set avec de nombreux effets « ha ! ha ».

