

# TABLE DES MATIÈRES

<b>1</b>	<b>Introduction – Commander des maquettes Fischertechnik avec ROBO Pro .....</b>	<b>175</b>
1.1	Installation de ROBO Pro .....	175
1.2	Installation du pilote USB pour ROBO Interface .....	176
1.3	Premiers pas .....	179
<b>2</b>	<b>Petit test du matériel avant la programmation .....</b>	<b>183</b>
2.1	Raccordement de l'Interface au PC ..	183
2.2	Pour que la liaison soit bonne – Mises au point de l'Interface .....	183
2.3	La liaison n'est pas bonne – Il n'y a pas de liaison avec l'Interface !? .....	185
2.4	Tout fonctionne – Test de l'Interface ..	185
<b>3</b>	<b>Niveau 1: Mon premier programme de commande .....</b>	<b>187</b>
3.1	Créer un nouveau programme .....	187
3.2	Les éléments du programme de commande .....	188
3.3	Ajouter, déplacer et modifier un élément de programme .....	188
3.4	Lier les éléments de programme .....	191
3.5	Test du premier programme de commande .....	192
3.6	Autres éléments de programme .....	194
3.6.1	Attente .....	194
3.6.2	Attendre l'entrée .....	195
3.6.3	Compteur d'impulsions .....	195
3.6.4	Boucle de comptage .....	196
3.7	Utilisation en mode Online ou en mode Téléchargement Quelle est la différence ? .....	196
3.8	Trucs et Astuces .....	199
<b>4</b>	<b>Niveau 2: Travailler avec des sous-programmes .....</b>	<b>200</b>
4.1	Mon premier sous-programme .....	201
4.2	Bibliothèque des sous-programmes ..	205
4.2.1	Utiliser la bibliothèque .....	205
4.2.2	Utiliser la bibliothèque personnelle .....	205
4.3	Traiter les symboles des sous-programmes .....	206
<b>5</b>	<b>Niveau 3: Variables, Panneau de commande et Cie .....</b>	<b>208</b>
5.1	Variables et commandes .....	208
5.2	Variables et procédures multiples .....	210
5.3	Panneaux de commande .....	210
5.4	Timer .....	214
5.5	Entrées de commande pour sous-programme .....	216
5.6	Listes (Arrays) .....	219
5.7	Opérateurs .....	220
<b>6</b>	<b>Commander des modules d'extension et des Interfaces multiples .....</b>	<b>224</b>
6.1	Modules d'extension .....	224
6.2	Interfaces multiples .....	224
6.3	Attribution des Interfaces dans les sous-programmes .....	226
6.4	Trucs & Astuces .....	227
6.5	Modifier le numéro de série de l'Interface ou la version du microprogramme .....	227
<b>7</b>	<b>Aperçu des éléments du programme .....</b>	<b>229</b>
7.1	Éléments fondamentaux (Niveau 1) ..	229
7.1.1	Démarrer .....	229
7.1.2	Terminer .....	229
7.1.3	Branchement digital .....	229
7.1.4	Branchement analogique .....	230
7.1.5	Attente .....	231
7.1.6	Sortie Moteur .....	231
7.1.7	Sortie Lampe (Niveau 2) .....	232
7.1.8	Attendre une entrée .....	233
7.1.9	Compteur d'impulsion .....	234
7.1.10	Boucle de comptage .....	235
7.2	Sous-programme I/O (Niveau 2-3) ..	235
7.2.1	Entrée du Sous-programme (Niveau 2) .....	235
7.2.2	Sortie du sous-programme (Niveau 2) .....	235
7.2.3	Entrée de commande du sous-programme (Niveau 3) .....	236
7.2.4	Sortie de Commande du sous-programme (Niveau 3) .....	236
7.3	Variable, Liste, ... (Niveau 3) .....	236
7.3.1	Variable (globale) .....	236
7.3.2	Variable locale .....	238
7.3.3	Constantes .....	238

7.3.4	Variable de Timer.....	238
7.3.5	Liste.....	239
7.4	<i>Commandes (Niveau 3)</i> .....	241
7.4.1	= (Attribuer).....	242
7.4.2	+ ( Plus ).....	242
7.4.3	- ( Moins ) .....	243
7.4.4	Droite.....	243
7.4.5	Gauche.....	243
7.4.6	Stop.....	243
7.4.7	In .....	243
7.4.8	Out .....	243
7.4.9	Texte .....	244
7.4.10	Attacher une valeur.....	244
7.4.11	Supprimer une (des) valeur(s) ..	244
7.4.12	Permuter des valeurs .....	244
7.5	<i>Comparaison, Attente, ... (Niveau 3)</i> .....	244
7.5.1	Branchement (avec entrée de données) .....	245
7.5.2	Comparaison avec une valeur fixe .....	245
7.5.3	Comparaison .....	246
7.5.4	Attente.....	246
7.5.5	Attendre.....	246
7.5.6	Compteur d'impulsions .....	247
7.6	<i>Entrées / Sorties d'Interface, ...</i> .....	247
7.6.1	Entrée digitale .....	247
7.6.2	Entrée analogique.....	248
7.6.3	Entrée IR .....	249
7.6.4	Sortie Moteur.....	250
7.6.5	Sortie Lampe.....	251
7.6.6	Entrée du panneau de commande .....	252
7.6.7	Sortie du panneau de commande .....	252
7.7	<i>Opérateurs</i> .....	253
7.7.1	Opérateurs arithmétiques .....	253
7.7.2	Opérateurs logiques.....	254
<b>8</b>	<b>Aperçu des éléments de commande et des panneaux de commande .....</b>	<b>255</b>
8.1	<i>Affichages</i> .....	255
8.1.1	Appareil de mesure.....	255
8.1.2	Affichage textuel.....	256
8.1.3	Lampe de l'affichage .....	258
8.2	<i>Éléments de commande</i> .....	258
8.2.1	Bouton.....	258
8.2.2	Régulateur à coulisse.....	259
<b>9</b>	<b>Fonctions de dessin.....</b>	<b>261</b>

# 1 Introduction – Commander des maquettes Fischertechnik avec ROBO Pro

Vous vous êtes sans doute déjà demandé comment cela fonctionne lorsque des robots, comme s'ils étaient mus par des mains de fantômes, accomplissent certaines tâches. Mais ce n'est pas seulement le cas des vrais robots ; dans de nombreux autres domaines, nous nous occupons des techniques de commande et d'automatisation, même chez Fischertechnik. Déjà d'ici deux chapitres, nous concevrons ensemble un petit programme de commande pour une porte de garage automatique et nous apprendrons ainsi comment on peut, au moyen du logiciel ROBO Pro pour Windows, résoudre et tester de telles tâches de commande. L'utilisation de ROBO Pro est donc très simple. Les programmes de commande, plus précisément les organigrammes et par la suite les organigrammes de données – comme nous le verrons plus tard –, sont presque exclusivement élaborés sur l'interface utilisateur graphique.

Afin de pouvoir commander votre maquette Fischertechnik via votre ordinateur, vous avez besoin, en plus du logiciel de commande ROBO Pro, d'une Interface, servant d'agent de liaison entre l'ordinateur et la maquette. Les commandes du logiciel sont alors converties de sorte que, par exemple, des moteurs puissent être commandés et des signaux puissent être traités par des capteurs. Chez Fischertechnik, vous trouverez ROBO Interface (N° d'article 93293) et la plus ancienne Intelligent Interface (N° d'article 30402). Ces deux interfaces peuvent être utilisées avec ROBO Pro. Cependant, avec Intelligent Interface, ROBO Pro ne prend en charge que le mode Online. L'ancienne Interface parallèle (N° d'article 30520) n'est plus prise en charge par ROBO Pro.

Encore quelques mots sur l'organisation de ce manuel. Celui-ci se divise en deux parties. La première partie (chapitres 1 à 4) décrit les procédures essentielles à accomplir lors de la programmation avec ROBO Pro. Vous recevrez de nombreuses informations et bases pour la programmation en général et pour l'utilisation du logiciel ROBO Pro.

La seconde partie reprend les chapitres 5 à 7 et vous initie aux fonctions avancées de programmation.

A partir du chapitre 8, vous trouverez essentiellement des outils de référence. Ainsi, lorsque vous vous serez familiarisé avec l'utilisation de ROBO Pro après la première partie et que vous chercherez des informations bien particulières, vous trouverez dans ces derniers chapitres des explications détaillées sur les différents éléments de programmes.

Et maintenant, au travail ! Vous êtes sans doute impatient de connaître les possibilités de programmation que vous offre le logiciel ROBO Pro pour vos maquettes Fischertechnik. Bon amusement !

## 1.1 Installation de ROBO Pro

Conditions requises pour l'installation de ROBO Pro :

- PC compatible IBM avec processeur Pentium d'une fréquence élémentaire d'au moins 600 MHz, 32 Mb RAM et env. 20 Mb de mémoire libre sur le disque dur
- Moniteur et carte graphique d'une résolution d'au moins 1024x768 pixels. Pour les moniteurs à tube cathodique, le taux de rafraîchissement d'image doit s'élever au moins à 85 Hertz afin que l'image ne scintille pas. Sur les écrans plats TFT, l'image scintille quel que soit le taux de rafraîchissement d'image de sorte que le taux de rafraîchissement d'image est insignifiant pour ce type d'écran.

- Microsoft, Windows, Version Windows 95, 98, ME, NT4.0, 2000 ou XP
- Une interface USB libre ou une interface RS232 COM1 à COM4 libre pour le raccordement de ROBO Interface — N° d'article 93293 — ou interface RS232 COM1 à COM4 libre pour le raccordement de l'ancienne Intelligent Interface— N° d'article 30402.

Vous devez tout d'abord allumer votre ordinateur et attendre que le système d'exploitation (Windows) soit totalement chargé. ROBO Interface ne doit être raccordé à votre ordinateur que lorsque l'installation s'est terminée avec succès. Insérez alors le CD d'installation dans votre lecteur CD-ROM. Le programme d'installation sur le CD se lance alors automatiquement.

- Cliquez sur le bouton **Suivant** de la fenêtre d'accueil du programme d'installation.
- La seconde fenêtre **Informations importantes** contient des informations importantes sur l'installation du programme ou sur le programme en lui-même. Cliquez aussi sur le bouton **Suivant**.
- Sur la troisième fenêtre **Contrat de licence**, vous pourrez lire le contrat de licence ROBO Pro. Vous devez accepter ce contrat en cliquant sur **Oui** avant de pouvoir accéder à la fenêtre suivante en cliquant sur **Suivant**.
- Sur la fenêtre suivante **Informations Utilisateur**, vous devez entrer votre nom.
- Sur la fenêtre **Type d'installation**, vous pouvez choisir entre **Installation Express** ou **Installation personnalisée**. Dans l'installation personnalisée, vous pouvez décider de ne pas installer certains des composants de l'installation. Si vous installez une nouvelle version de ROBO Pro sur une ancienne version et avez modifié certains des programmes d'exemple de l'ancienne version, vous pouvez ne pas installer les exemples lors de l'installation personnalisée. Dans le cas contraire, les programmes d'exemple modifiés seront **écrasés sans avertissement** lors de l'installation. Si vous choisissez l'installation personnalisée et cliquez sur **Suivant**, une fenêtre supplémentaire apparaît sur laquelle vous pouvez choisir les composants.
- Dans la fenêtre **Sélection du dossier d'installation**, vous pouvez choisir le dossier ou le chemin de répertoire où vous souhaitez installer ROBO Pro. Normalement, ce chemin est C:\Programme\ROBO Pro. Vous pouvez cependant sélectionner un autre répertoire.
- Lorsque, sur la dernière fenêtre, vous cliquez sur **Achever**, l'installation commence. Dès que l'installation est achevée – ce qui ne dure généralement que quelques secondes –, le programme vous informe que l'installation a été achevée avec succès. En cas de problème, un message d'erreur s'affiche afin de vous aidez à résoudre ce problème.

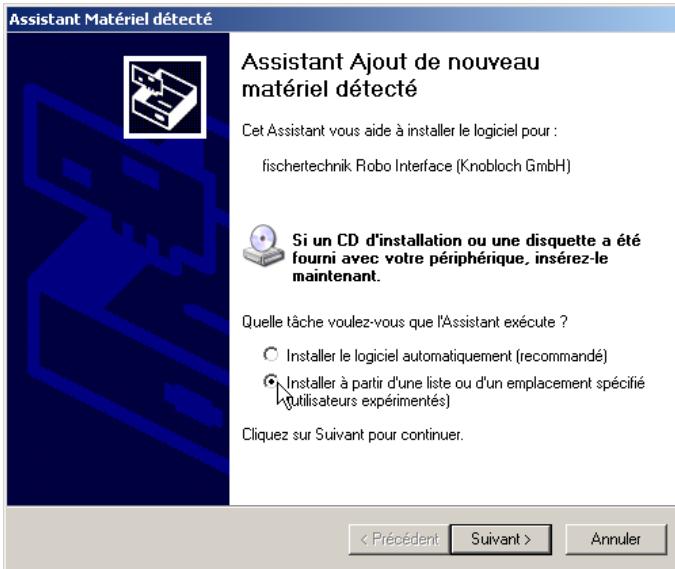
## 1.2 Installation du pilote USB pour ROBO Interface

Cette étape n'est nécessaire que lorsqu'une nouvelle ROBO Interface doit être raccordée à l'interface USB. ROBO Interface peut également être raccordée à l'interface série COM1-COM4. Pour l'ancienne Intelligent Interface, cette étape n'est pas nécessaire car Intelligent Interface ne peut être raccordée que via une interface série. Les anciennes versions de Windows – Windows 95 et Windows NT 4.0 – ne prennent pas en charge l'interface USB. Si vous utilisez Windows 95 et NT 4.0, ROBO Interface ne peut être raccordée que via une interface série. Dans ce cas, un pilote ne doit pas être installé.

**Informations importantes pour l'installation sous Windows 2000 et Windows XP :**

Le pilote USB ne peut être installé que par un utilisateur possédant des droits d'administrateur sur l'ordinateur. Si le programme d'installation vous signale que vous n'avez pas le droit d'installer le pilote USB, vous devez demander à un administrateur système d'installer le pilote ou d'installer ROBO Pro sans ce pilote. Vous pouvez alors raccorder votre Interface à une interface sérieelle un peu plus lente.

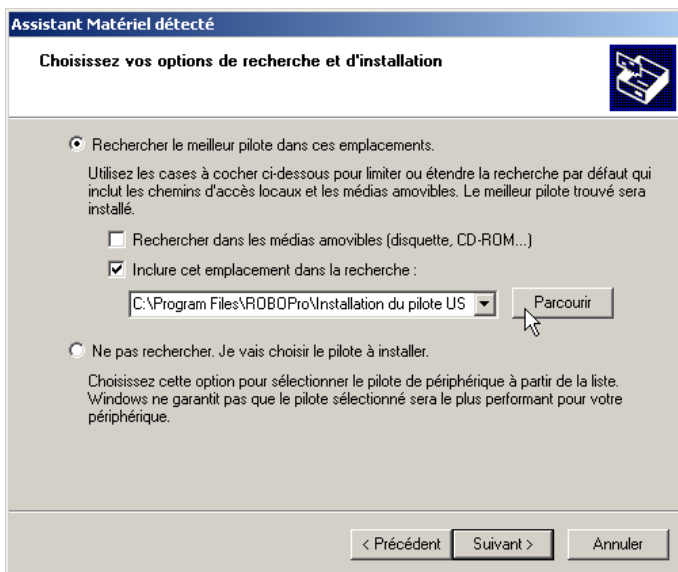
Pour installer le pilote USB, vous devez d'abord raccorder ROBO Interface à l'ordinateur au moyen d'un câble USB et l'alimenter en courant. Windows détecte automatiquement que l'Interface est raccordée et la fenêtre suivante s'affiche :



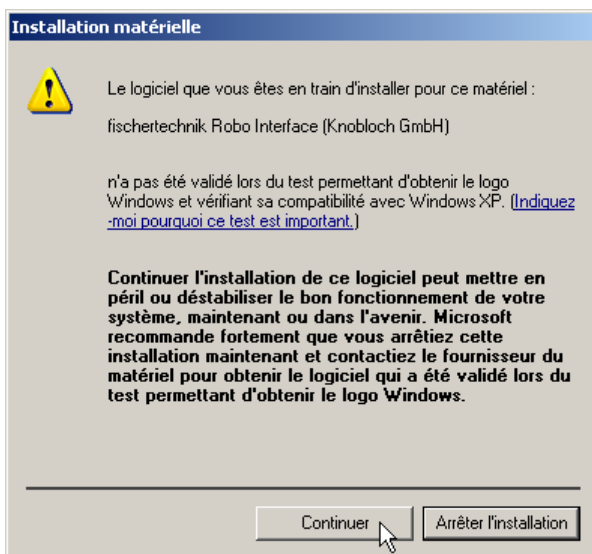
La fenêtre peut légèrement varier en fonction du système d'exploitation !

Vous devez sélectionner **Choisir le logiciel dans une liste ou une source** et cliquer sur **Suivant**.

Sur la fenêtre suivante, désactivez **Parcourir les supports externes** et activez **Parcourir la source suivante**. Appuyez ensuite sur **Parcourir** et choisissez le sous-répertoire **Installation du pilote USB** dans le répertoire où est installé ROBO Pro (le répertoire habituel est C:\ROBOPro):

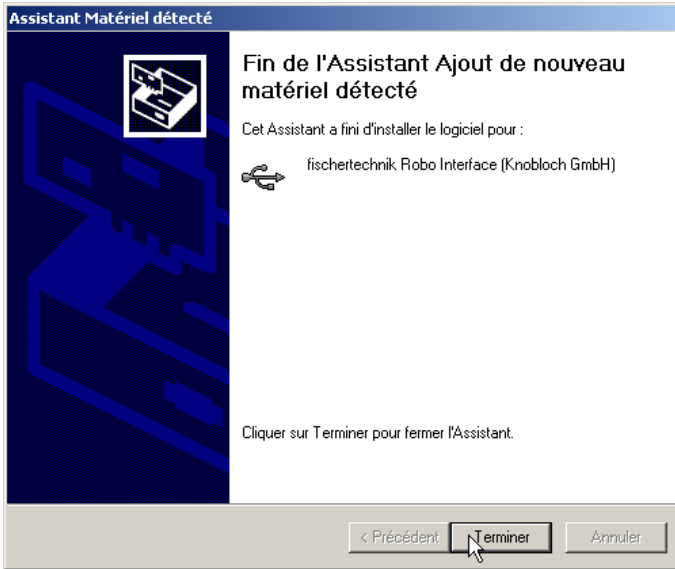


Après avoir appuyé sur **Suivant**, le message suivant peut apparaître sous Windows XP :



Le pilote USB est alors contrôlé par Microsoft. Dès que le contrôle est terminé, le pilote est marqué par Microsoft de sorte que ce message n'apparaisse plus. Afin d'installer le pilote, cliquez sur **Poursuivre l'installation**.

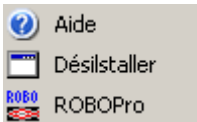
Le message suivant apparaît finalement :



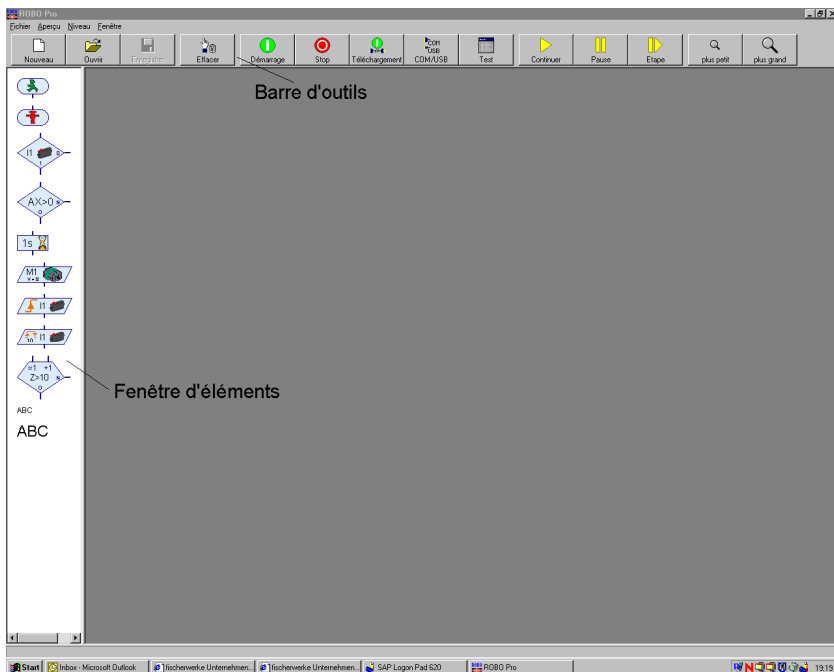
Cliquez sur **Achever** afin de conclure l'installation du pilote USB.

### 1.3 Premiers pas

Impatient ? Démarrez alors le programme ROBO Pro. Pour ce faire, cliquez sur le bouton Démarrer de la barre de tâche et choisissez ensuite **Programmes** ou **Tous les programmes** et **ROBO Pro**. Dans le dossier du menu Démarrer, vous trouverez les entrées suivantes :



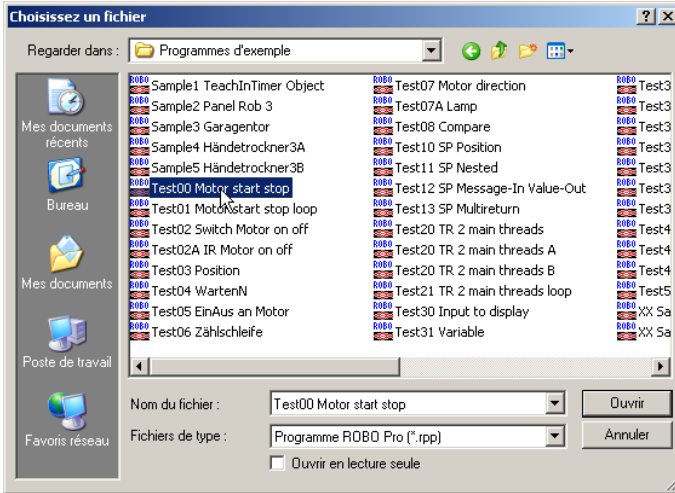
Vous pouvez désinstaller ROBO Pro au moyen de l'entrée Désinstaller. L'entrée Aide ouvre le fichier d'aide de ROBO Pro et l'entrée ROBO Pro lance le programme ROBO Pro. Choisissez maintenant l'entrée **ROBO Pro** pour démarrer le logiciel.



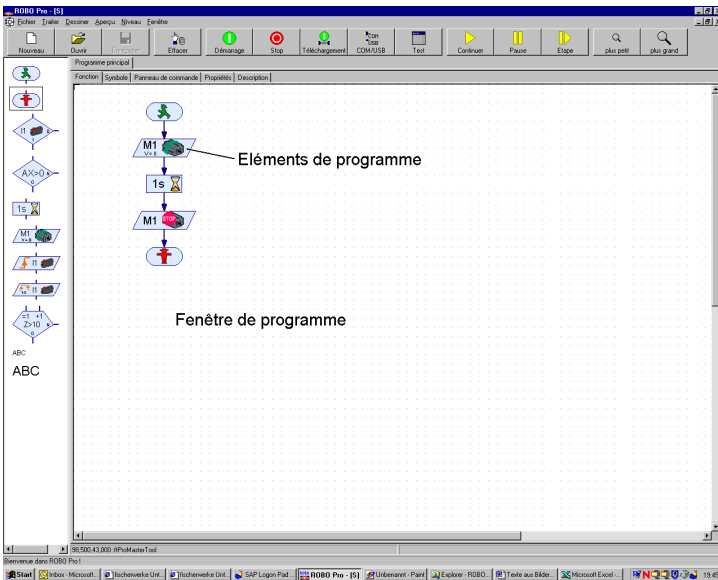
Sur le haut de la fenêtre, vous trouverez une barre de menu et une barre d'outils avec différents boutons de commande, ainsi que, sur le côté gauche, une fenêtre avec des éléments de programme. Si deux fenêtres superposées apparaissent dans la colonne de gauche, ROBO Pro n'est pas réglé sur **Niveau 1**. Pour adapter les fonctionnalités de ROBO Pro aux connaissances croissantes de l'utilisateur, ROBO Pro peut être réglé du Niveau 1, pour les Débutants, au Niveau 5, pour les Experts. Regardez maintenant dans le menu **Niveau** si **Niveau 1 : Débutants** est coché. Si non, sélectionnez Niveau 1.



Vous avez maintenant la possibilité, soit de créer un nouveau fichier de programme, soit d'ouvrir un fichier de programme déjà existant. Nous créerons un nouveau fichier de programme au chapitre 3 lorsque nous concevrons notre premier programme de commande. Pour apprendre à connaître l'interface utilisateur, ouvrons un des programmes d'exemple déjà existant. Pour ce faire, cliquez sur l'entrée **Ouvrir** du menu **Fichier** ou utilisez le bouton **Ouvrir** de la barre d'outils. Les fichiers d'exemple se trouvent dans le répertoire **C:\Programme\ROBO Pro\Programmes Exemples**



Ouvrez le fichier **Test00 Motor start Stopp.rpp** :



Vous pouvez voir ici à quoi ressemble un programme ROBO Pro simple. Les éléments de programme de la fenêtre d'éléments permettent, lors de la programmation dans la fenêtre de

programme, de concevoir les organigrammes des programmes de commande. Les organigrammes conçus peuvent alors être contrôlés et testés avec une Interface Fischertechnik raccordée. Mais allons-y lentement, nous allons apprendre à connaître la programmation pas à pas dans les chapitres suivants ! Après vous être fait une première idée de l'interface utilisateur, fermez le fichier de programme avec la commande **Fermer** du menu **Fichier**. Vous pouvez répondre **Non** lorsque le système vous demandera si vous souhaitez enregistrer le fichier.

## 2 Petit test du matériel avant la programmation

Afin de pouvoir tester les programmes de commande que nous créerons plus tard, l'Interface doit être raccordée à l'ordinateur. Cependant, le logiciel doit être réglé en fonction de l'Interface utilisée (ROBO Interface - N° d'article 93293 - ou l'ancienne Intelligent Interface - N° d'article 30402) et la liaison doit être testée. C'est ce que nous ferons au prochain chapitre.

### 2.1 Raccordement de l'Interface au PC

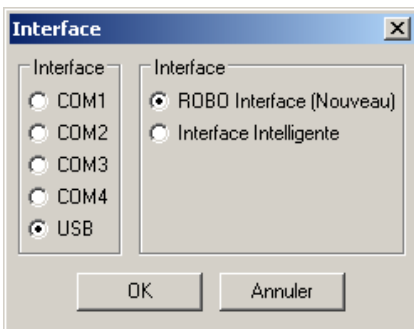
Ceci ne devrait pas poser de gros problèmes. Le câble de liaison fourni avec l'Interface est raccordé à l'Interface et à une des interfaces de l'ordinateur :

- Pour ROBO Interface (N° d'article 93293), une interface USB ou une interface sériele COM1 à COM4 peuvent être utilisées.
- Pour Intelligent Interface (N° d'article 30402), une interface sériele COM1 à COM4 doit être utilisée.

Les raccordements de ces interfaces se trouvent généralement sur la face arrière de l'ordinateur. La position exacte des différents raccordements est décrite avec précision dans les instructions d'utilisation de votre ordinateur, veuillez donc vous y référer. Les raccordements USB peuvent également se trouver sur la face avant de votre ordinateur. N'oubliez pas d'alimenter l'Interface en électricité (bloc d'alimentation ou batterie). Les raccordements particuliers à chaque Interface sont décrits avec précision dans les instructions d'utilisation.

### 2.2 Pour que la liaison soit bonne – Mises au point de l'Interface

Pour que la liaison entre l'ordinateur et l'Interface fonctionne correctement, l'Interface utilisée doit être sélectionnée dans ROBO Pro. Pour ce faire, démarrez ROBO Pro via l'entrée **ROBO Pro** du menu Démarrer, sous **Programmes** ou **Tous les programmes**. Appuyez ensuite sur le bouton **COM/USB** de la barre d'outils. La fenêtre suivante apparaît :

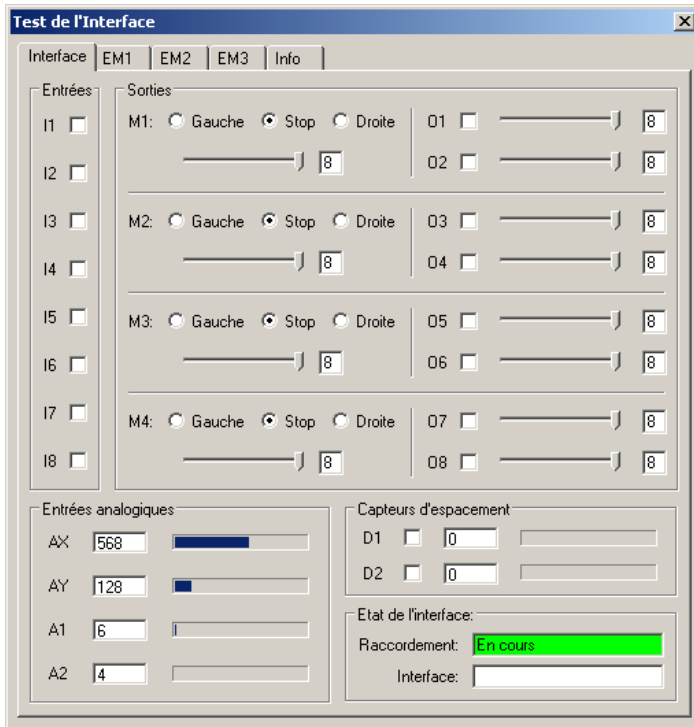


Vous pouvez alors choisir l'interface, mais aussi le type d'interface. Comme nous l'avons déjà mentionné, Intelligent Interface ne prend en charge que les interfaces sérieles COM1-COM4.



Test

Après avoir choisi le réglage correct, fermez la fenêtre en cliquant sur OK. Ouvrez maintenant la fenêtre de test de l'Interface en appuyant sur le bouton **Test** de la barre d'outils :



Cette fenêtre représente les entrées et sorties présentes sur l'Interface. La barre verte en bas à droite indique le statut de la liaison entre l'ordinateur et l'Interface :

- **Liaison avec l'Interface OK** confirme une liaison correcte avec l'Interface
- **Pas de liaison avec l'Interface** signifie que la liaison n'a pas été correctement réglée et que l'ordinateur ne peut pas établir de liaison avec l'Interface. La barre est alors rouge.

Pour pouvoir modifier le réglage de l'Interface ou de la liaison, vous devez fermer la fenêtre de test (avec le X en haut à droite) et sélectionner une autre interface ou un autre type d'interface au moyen du bouton COM/USB de la barre d'outils, comme cela a été décrit plus haut.

Si vous pouvez établir la liaison entre l'ordinateur et l'Interface et que la barre verte apparaît dans la fenêtre **Test**, vous pouvez sans problème passer au prochain chapitre.

Dans le cas contraire, les astuces données dans le prochain sous-chapitre pourront peut-être vous aider.

## 2.3 La liaison n'est pas bonne – Il n'y a pas de liaison avec l'Interface !?

Si, malgré une interface sériele COM correctement configurée (cf. plus haut), le message **Pas de liaison avec l'Interface** apparaît pour ROBO Interface ou Intelligent Interface, veuillez vérifier les points suivants. Il peut également être utile de demander conseil à quelqu'un qui s'y connaît :

- **Alimentation en courant :**  
L'Interface est-elle correctement alimentée en courant ? Si vous utilisez des batteries ou des accumulateurs pour l'alimentation en courant, il est possible que des accumulateurs vides ne fournissent plus assez de tension. Si la tension de la batterie ne dépasse pas 6 V, le processeur de ROBO Interface ne fonctionne plus. Dans ce cas, la DEL rouge reste allumée ou aucune des DELs ne sont allumées. Si la tension est bonne, certaines DELs sont allumées. Pour l'ancienne Intelligent Interface, il est impossible de constater si la tension est suffisante pour le processeur au moyen des DELs. Si la tension est trop faible, l'accumulateur doit être chargé ou des nouvelles batteries doivent être utilisées ; il est également possible de tester l'Interface avec un bloc d'alimentation.
- **L'interface fonctionne-t-elle ?**  
Vous pouvez le savoir en testant un autre appareil sériel, comme un modem externe, sur cette interface.
- Y a-t-il un conflit avec un autre pilote sur la même interface (p.e. un modem) ? Ce pilote devra éventuellement être désactivé (voir les manuels de Windows et de l'appareil).
- Uniquement pour Windows NT/2000/XP et l'ancienne Intelligent Interface : Si une ancienne Intelligent Interface est déjà reliée à l'ordinateur ou à l'alimentation au démarrage de l'ordinateur, elle basculera malheureusement en mode Téléchargement par Windows NT. Afin de rétablir la liaison avec l'ordinateur, il faut simplement interrompre un court moment l'alimentation en courant de l'Interface. C'est impossible sur les nouvelles ROBO Interface.
- Si la liaison avec l'Interface n'a pas encore pu être établie, celle-ci ou le câble de liaison sont sans doute défectueux. Dans ce cas, adressez-vous au Service Fischertechnik (Adresse: voir menu „?“ / **Info sur**).

## 2.4 Tout fonctionne – Test de l'Interface

Après que la liaison ait été correctement établie, nous pouvons tester l'Interface en elle-même et les maquettes qui y sont raccordées au moyen du test de l'Interface. Comme nous l'avons déjà mentionné, la fenêtre de test montre les différentes entrées et sorties de l'Interface :



Test

- **Entrées digitales I1–I8**  
I1–I8 sont les entrées digitales de l'Interface. Des capteurs y sont raccordés. Les entrées digitales peuvent accepter les statuts 0 et 1 ou Oui et Non. Les types de capteurs pouvant être raccordés aux entrées digitales sont des interrupteurs (mini-boutons), mais aussi des transistors photo (capteurs de lumière) ou des relais à contacts scellés (capteurs magnétiques). Vous pouvez contrôler le bon fonctionnement de ces entrées en raccordant, p.e., un mini-bouton (N° d'article 37783) à l'Interface. (Utilisez les contacts 1 et 3 du bouton). Aussitôt que vous appuyez sur le bouton, l'affichage de I1 se coche. Si le bouton a été raccordé différemment (contacts 1 et 2), l'affichage est directement coché et se décoche lorsque vous appuyez sur le bouton.
- **Sorties Moteur M1–M4**  
M1 – M4 sont les sorties de l'Interface. C'est ici que seront raccordés les „acteurs“. Il peut

s'agit, p.e., de moteurs, d'électroaimants ou de lampes. Les 4 sorties Moteur peuvent être réglées sur 8 niveaux en ce qui concerne la vitesse, mais aussi la direction. Le réglage de la vitesse se fait au moyen d'un régulateur à coulisse. Qui plus est, la vitesse est indiquée de manière chiffrée à côté du régulateur à coulisse. Si vous désirez tester une sortie, raccordez un moteur à une des sorties, p.e. M1.

- **Sorties Lampes O1–O8**

Les sorties Moteur peuvent également être utilisées alternativement comme des couples de sorties particulières. Il est possible de commander des lampes, mais aussi des moteurs ne devant tourner que dans un sens (p.e. des bandes transporteuses). Si vous désirez tester une de ces sorties, raccordez le raccordement d'une lampe à la sortie, p.e. O1. Raccordez l'autre raccordement de la lampe à la douille massive de l'Interface (⊥).

- **Entrées analogiques AX–AY**

Les entrées analogiques AX et AY mesurent la résistance du capteur raccordé. Vous pouvez raccorder des résistances NTC pour la mesure des températures, des potentiomètres, des résistances photo ou des transistors photo.

- **Entrées analogiques A1–A2**

Ces deux entrées mesurent une tension allant de 0 à 10V.

- **Capteurs d'espacement D1–D2**

Les entrées des capteurs d'espacement D1 et D2 permettent le raccordement de capteurs d'espacement particuliers. Les capteurs d'espacement D1 et D2 peuvent être aussi bien des entrées digitales que des entrées analogiques.

- **Modules d'extension EM1–EM3**

Des modules d'extension sont raccordés à l'Interface (jusqu'à trois extensions I/O N° d'article 93294 pour ROBO Interface, mais tout au plus un module d'extension N° d'article 16554 pour Intelligent Interface); ceux-ci peuvent être commandés en basculant sur le module d'extension via le registre situé sur le bord supérieur.

### 3 Niveau 1: Mon premier programme de commande

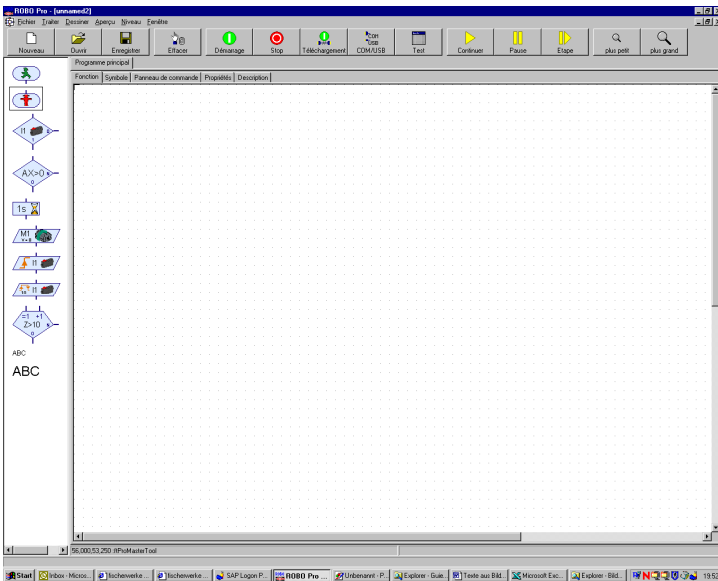
Après avoir testé le matériel, c'est-à-dire l'Interface et les interrupteurs et moteurs y étant raccordés, dans le chapitre précédent, nous allons maintenant nous pencher sur la programmation. Mais que veut vraiment dire „programmation“ ? Imaginez-vous qu'un robot, par exemple, est raccordé à notre Interface. Ce robot est cependant tellement bête qu'il ne peut pas fonctionner tout seul. Par chance, nous sommes un peu plus intelligent que lui. Nous pouvons dire très précisément au robot ce qu'il doit faire. Comment ? Voyons, que s'est-il passé au chapitre précédent lorsque nous avons réglé la sortie Moteur M1 sur „Gauche“ ? Bravo ! Nous avons allumé le moteur. Si ce moteur avait actionné la pince de notre robot, p.e., nous n'aurions rien fait d'autre que de lui dire : "Attrape l'objet !". Nous ne voulons cependant plus déclencher chacun des mouvements de la main, mais bien que le robot le fasse „automatiquement“. Pour ce faire, nous devons conserver en mémoire tous les mouvements à accomplir de manière isolée de sorte que le robot puisse les traiter les uns après les autres ; en d'autres termes, nous devons concevoir un programme qui commandera le robot à notre place. En langage spécialisé, nous appelons cela très logiquement un programme de commande.

#### 3.1 Créer un nouveau programme

Avec le logiciel ROBO Pro, vous disposez d'un merveilleux outil pour développer un tel programme de commande et pour le tester au moyen d'une Interface raccordée. Pas de panique, nous n'allons pas directement programmer un robot. Nous nous contenterons tout d'abord de tâches de commande simples. Pour ce faire, nous devons créer un nouveau programme. Dans la barre d'outils, vous trouverez une entrée „Nouveau“. Si vous cliquez sur cette entrée avec le bouton gauche de la souris, un nouveau programme vide est créé :



Nouveau



Une grande surface blanche sur laquelle vous allez pouvoir saisir votre premier programme apparaît alors. Si deux fenêtres superposées apparaissent sur le bord gauche, veuillez tout d'abord choisir **Niveau 1 : Débutants** dans le menu **Niveau**.

## 3.2 Les éléments du programme de commande

Nous pouvons maintenant nous mettre au travail et concevoir notre premier programme de commande. Nous allons le faire en suivant un exemple concret :

### Description des fonctions:

Imaginez-vous une porte de garage s'ouvrant automatiquement. Peut-être en possédez-vous une à la maison ! Lorsque vous arrivez en voiture devant le garage, une pression sur le bouton de l'émetteur suffit pour que la porte s'ouvre, actionnée par un moteur. Le moteur doit fonctionner suffisamment longtemps pour que la porte du garage soit totalement ouverte.

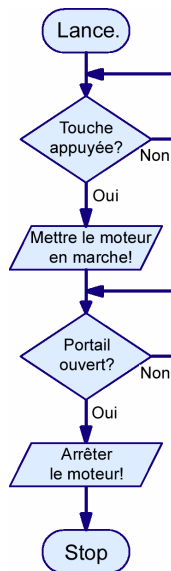
Il est cependant fort compliqué et peu concret de traduire une commande en mots. C'est pourquoi des **organigrammes** sont utilisés pour représenter les différentes actions devant être accomplies les unes après les autres et les conditions devant être remplies pour que ces actions se réalisent. Dans le cas de notre commande, la condition pour que l'action „Allumer le moteur“ s'accomplisse est que le bouton doit être enfoncé. La lecture d'un organigramme de ce type est simple : Il faut toujours suivre les flèches pas à pas ! Ce qui nous donne le mode de fonctionnement de la commande – les différentes étapes ne peuvent être accomplies que dans l'ordre donné par les flèches, jamais autrement. Sinon, aurions-nous besoin de faire tout le travail ? Non ?

Avec l'aide de notre logiciel ROBO Pro, nous pouvons tracer précisément cet organigramme et ainsi créer le **programme de commande** pour le matériel raccordé (Interface, Moteurs, interrupteurs, etc.). Le reste est pris en charge par le logiciel, comme c'est également le cas dans les utilisations industrielles de grande envergure ! Nous pouvons ainsi nous concentrer essentiellement sur la conception de l'organigramme.

Vous composez l'organigramme au moyen d'éléments de programme. Encore un nouveau terme ? Rien de bien grave ! Dans ROBO Pro, les éléments isolés à partir desquels l'organigramme est conçu sont appelés éléments de programme. L'action „Allumer le moteur“ veut simplement dire que l'Interface doit effectivement allumer le moteur y étant raccordé ! Vous trouverez les éléments de programme disponibles dans la fenêtre d'élément sur la gauche.

## 3.3 Ajouter, déplacer et modifier un élément de programme

Il s'agit maintenant de concevoir l'organigramme pour notre commande de porte de garage au moyen des éléments de programme de la fenêtre d'éléments. Tous les éléments de programme disponibles peuvent être pris dans la fenêtre d'éléments et ajoutés à la fenêtre du programme.



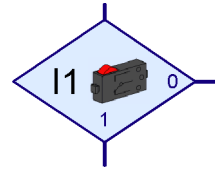


## Ajouter des éléments de programme

Pour ce faire, déplacez la souris sur le symbole de l'élément de programme souhaité et cliquez une fois sur celui-ci avec le bouton gauche de la souris. Placez ensuite la souris à l'endroit souhaité dans la fenêtre du programme (la grande surface blanche) et cliquez à nouveau. Vous pouvez également déplacer l'élément de programme de la fenêtre d'éléments à la fenêtre du programme en gardant le bouton de la souris enfoncé. Un programme commence toujours avec un élément de démarrage. L'élément de démarrage est l'élément arrondi avec le petit bonhomme vert en mouvement. Essayez tout d'abord avec cet élément de programme : Cliquez une fois avec le bouton gauche de la souris sur l'élément de démarrage dans la fenêtre d'élément, déplacez la souris vers le haut de la fenêtre de programme et cliquez à nouveau avec le bouton gauche de la souris.



Il faut ensuite insérer dans l'organigramme un élément qui envoie une requête à une entrée et qui, en fonction du statut de cette entrée, la branche sur l'un ou l'autre chemin du programme. Cliquez dans la fenêtre d'éléments sur l'élément représenté à droite et déplacez ensuite la souris sous l'élément de démarrage inséré précédemment. Si l'entrée supérieure de l'élément de branchement se situe un ou deux points de trame sous la sortie de l'élément de démarrage, une ligne de liaison apparaît dans la fenêtre du programme. Si vous cliquez à nouveau sur le bouton gauche de la souris, l'élément de branchement est inséré et automatiquement relié avec l'élément de démarrage.



## Déplacer des éléments de programme et des groupes

Après avoir été inséré, un élément de programme peut être déplacé à l'endroit souhaité en maintenant le bouton gauche de la souris enfoncé. Si vous désirez déplacer plusieurs éléments en même temps, vous pouvez d'abord étirer un cadre autour de ces éléments en maintenant le bouton gauche de la souris enfoncé. Pour ce faire, vous devez cliquer avec le bouton gauche de la souris dans une zone **vide**, maintenir le bouton enfoncé et étirer un rectangle contenant les éléments souhaités. Les éléments du rectangle seront alors représentés avec un bord rouge. Si vous déplacez alors un des éléments rouges avec le bouton gauche de la souris, tous les éléments rouges seront déplacés. Vous pouvez également marquer en rouge des éléments isolés en cliquant sur les éléments avec le bouton gauche de la souris tout en maintenant la touche de Majuscule enfoncée (il s'agit de la touche Majuscule/minuscule (shift)). Si vous cliquez avec le bouton gauche de la souris dans une zone vide, tous les éléments marqués en rouge redeviendront normaux.

## Copier des éléments de programme et des groupes

Il existe deux solutions pour copier des éléments de programme. Vous pouvez procéder comme pour un déplacement, mais en enfonçant la touche **CTRL** du clavier avant de déplacer l'élément. Les éléments ne sont alors plus déplacés, mais bien copiés. Cette fonction ne vous permet cependant que de copier des éléments au sein même du programme. Si vous désirez copier des éléments d'un programme à un autre, vous pouvez utiliser le **presse-papier** de Windows. Sélectionnez tout d'abord certains éléments comme décrit précédemment pour le déplacement d'éléments. Si vous appuyez ensuite sur les touches **CTRL+C** du clavier ou que vous appelez le point de menu **Traiter / Copier**, tous les éléments sélectionnés seront copiés dans le presse-papier Windows. Vous pouvez alors basculer dans un autre programme et y insérer les éléments en enfonçant les touches **CTRL+V** ou en utilisant la fonction **Traiter / Ajouter**. Vous pouvez également ajouter à plusieurs reprises des éléments copiés une seule fois. Si vous désirez dépla-

cer des éléments d'un programme à un autre programme, vous pouvez, au départ, enfoncer les touches **CTRL+X** ou utiliser la fonction **Traiter / Couper**, plutôt que les touches **CTRL+C** ou la fonction **Traiter / Copier**.

## Effacer des éléments et Fonction Annuler



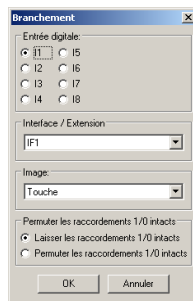
Effacer

Effacer des éléments est également très simple. Vous pouvez supprimer tous les éléments marqués en rouge (voir point précédent) en appuyant sur la touche Effacer (**Delete**) de votre clavier. Vous pouvez également supprimer des éléments isolés avec la fonction d'effacement. Pour ce faire, cliquez sur le bouton représenté ci-contre dans la barre d'outils et ensuite sur l'élément que vous voulez supprimer. Essayez un peu ! Vous pourrez ensuite replacer l'élément supprimé. Afin de remettre en place l'élément effacé, vous pouvez également utiliser la fonction **Annuler** dans le menu **Traiter**. Ce point vous permet de revenir en arrière et de supprimer toutes les modifications apportées au programme.

## Traiter les propriétés des éléments de programme

Si vous cliquez avec le bouton **droit** de la souris sur un élément de programme dans la fenêtre de programme, une fenêtre de dialogue apparaît dans laquelle vous pouvez modifier les propriétés des éléments. La fenêtre de propriétés d'un élément de branchement est représentée à droite.

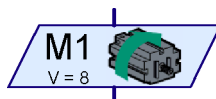
- Les boutons **11** à **18** vous permettent de décider à quelle entrée de l'Interface doit être envoyée la requête.
- La sélection **Interface / Extension** sera expliquée au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Le cadre **Image** vous permet de sélectionner une image pour le capteur raccordé à l'entrée. Les entrées digitales sont le plus souvent utilisées avec des interrupteurs, mais aussi avec des transistors photos ou des relais à contacts scellés.
- Le cadre **Permuter les raccordements 1/0** vous permet de permuter la position des sorties 1 et 0 du branchement. Normalement, la sortie 1 est en bas et la sortie 0 à droite. Il est cependant souvent plus pratique d'avoir la sortie 1 à droite. Cochez **Permuter les raccordements 1/0** ; les raccordements 1 et 0 seront alors permutés dès que vous fermerez la fenêtre en cliquant sur OK.



**Nota Bene:** Si vous raccordez un mini-bouton comme contact de fermeture sur le raccordement 1 et 3, le branchement du programme ira vers la branche 1 lorsque vous appuierez sur l'interrupteur ; dans le cas contraire, il ira vers 0.

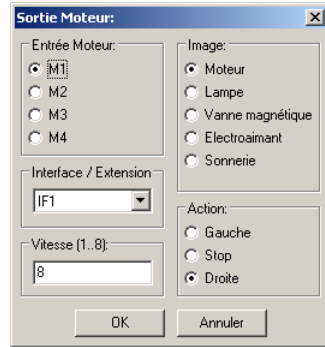
Si vous raccordez un mini-bouton comme contact d'ouverture sur le raccordement 1 et 2, le branchement du programme ira vers la branche 1 lorsque vous n'appuierez pas sur l'interrupteur ; dans le cas contraire, il ira vers 0.

L'élément de programme suivant de notre commande pour notre porte de garage est un élément Moteur. Ajoutez-le dans le programme comme vous l'avez fait pour les deux autres éléments, et ce, sous l'élément de branchement. Si possible, placez l'élément afin qu'il soit relié automatiquement avec l'élément se trouvant au-dessus.



Grâce à l'élément Moteur, vous pouvez allumer ou éteindre aussi bien un moteur qu'une lampe ou un électroaimant. Ouvrez la fenêtre de propriétés de l'élément Moteur en cliquant avec le bouton droit de la souris sur cet élément.

- Les boutons **M1** à **M4** vous permettent de choisir quelle sortie de l'Interface doit être commandée.
- Le champ **Image** vous permet de choisir une image pour représenter le module Fischertechnik raccordé à la sortie.
- La sélection **Interface / Extension** sera expliquée au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Action**, vous pouvez choisir comment la sortie doit être influencée. Vous pouvez démarrer ou arrêter un moteur avec un sens de rotation gauche ou droit. Vous pouvez également allumer ou éteindre une lampe.
- Le champ **Vitesse / Intensité** vous permet de régler avec quelle vitesse le moteur doit tourner ou comment doit briller la lampe. Les valeurs possibles vont de 1 à 8.

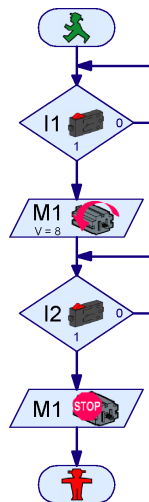
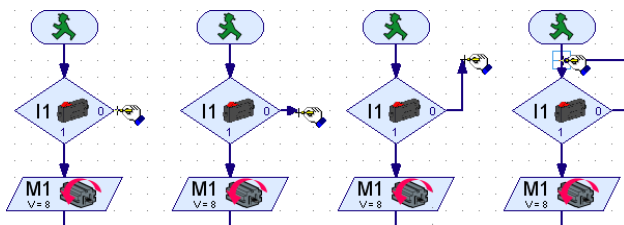


Pour notre organigramme de programme, nous avons besoin de la commande **Moteur M1 gauche avec une vitesse de 8**.

### 3.4 Lier les éléments de programme

Maintenant que vous savez comment ajouter des éléments dans le programme de commande, nous pouvons nous atteler à la tâche d'achever notre programme de commande. Repensez à la description du fonctionnement de la commande de notre porte de garage – ne nous manque-t-il rien ? Evidemment ! Nous avons déjà allumé le moteur en appuyant sur l'interrupteur mais, lorsque la porte est ouverte, le moteur doit s'éteindre automatiquement ! Dans la pratique, nous y parvenons grâce à ce que nous appelons un commutateur de fin. Il s'agit d'un interrupteur positionné sur la porte de garage de sorte qu'il soit actionné au moment où le moteur a totalement ouvert la porte. Et comme pour l'allumage du moteur, ce signal peut être utilisé pour éteindre le moteur. Pour la requête envoyée au commutateur de fin, nous pouvons à nouveau utiliser un élément de branchement.

Ajoutez dans votre programme de commande un élément de branchement envoyant une requête au commutateur de fin sur l'entrée I2. N'oubliez pas de cliquer sur l'élément avec le bouton droit de la souris et de régler l'entrée sur I2. Dès que la porte de garage sera ouverte et aura appuyé sur le commutateur de fin, le moteur doit se couper. On y parvient grâce à un élément Moteur. Utilisez tout d'abord le même élément que pour l'allumage du moteur. Si vous cliquez sur l'élément avec le bouton droit de la souris, vous pouvez modifier le fonctionnement de l'élément et le mettre sur **Arrêter le moteur**. Le programme se termine par un élément de fin. Votre programme devrait alors être plus ou moins semblable à celui représenté ci-contre. Si vous avez à chaque fois placé vos éléments avec un espacement de un ou deux points de trame, la plupart des entrées et des sorties sont déjà reliées entre elles par les flèches de déroulement du programme. La sortie Non (N) des deux branchements n'est cependant pas encore raccordée. Tant que l'interrupteur se situant sur l'entrée I1 n'est pas enfoncé, le programme doit toujours revenir en arrière et envoyer une requête à l'interrupteur. Afin de tracer cette ligne, cliquez avec la souris aux endroits indiqués sur la représentation ci-dessous :



**Nota Bene** : Si une des lignes ne devait pas être correctement reliée au raccordement ou à une autre ligne, un rectangle vert apparaîtrait à la pointe de la flèche. Dans ce cas, vous devez établir la liaison soit en déplaçant la ligne, soit en la supprimant et en la retraçant. Dans le cas contraire, le déroulement du programme ne pourra pas fonctionner à cet endroit.

### Supprimer des lignes de déroulement du programme

Pour supprimer des lignes, vous devez procéder de la même manière que pour supprimer des éléments de programme. Cliquez simplement sur la ligne avec le bouton gauche de la souris de sorte qu'elle soit marquée en rouge. Appuyez ensuite sur la touche Effacer (**Delete**) du clavier pour effacer la ligne. Vous pouvez également sélectionner plusieurs lignes en gardant la touche de majuscule enfoncée (touche de basculement entre les Majuscules et les minuscules) et en cliquant sur les lignes avec les boutons gauches de la souris. Pour marquer plusieurs lignes, vous pouvez également étirer un cadre autour de ces lignes. Vous pouvez alors supprimer toutes les lignes marquées en rouge en une fois en appuyant sur la touche **Delete**.

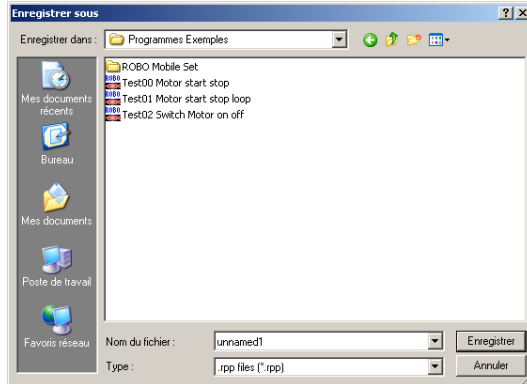
### 3.5 Test du premier programme de commande

Afin de pouvoir tester notre premier programme de commande, vous devez construire une petite maquette. Pour ce faire, il suffit de raccorder un interrupteur sur I1 et I2 et un moteur sur M1 (sur l'Interface).

**Note Bene** : Le raccordement de l'Interface à l'ordinateur et le réglage de l'Interface ont déjà été réalisés au chapitre précédent ; vous pouvez relire les indications à ce sujet dans ce chapitre.

Avant de tester le programme de commande, vous devez enregistrer le fichier du programme sur le disque dur de votre ordinateur. Cliquez avec la souris sur la commande **Enregistrer sous...** du menu **Fichier**. La fenêtre de dialogue suivante apparaît alors :

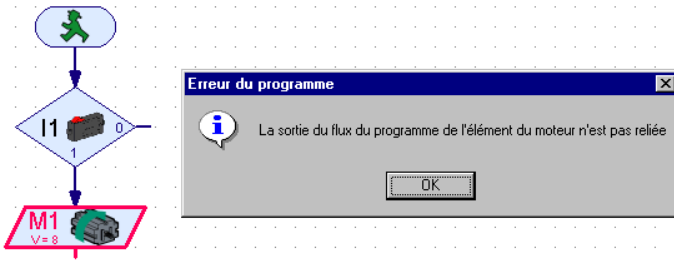
Sous „Enregistrer sous“, choisissez le répertoire dans lequel vous voulez enregistrer le programme. Introduisez un nom non attribué dans le champ „Nom du fichier“, p.e. PORTE-GARAGE et confirmez en cliquant sur „Enregistrer“ avec le bouton gauche de la souris.



Pour tester le programme, appuyez sur le bouton Start représenté ci-contre de la barre d'outils. ROBO Pro teste d'abord si tous les éléments de programme sont bien reliés. Si un élément n'était pas correctement relié ou si un autre problème se posait, l'élément posant problème serait marqué en rouge et un message d'erreur décrivant le type de problème apparaîtrait. Si, p.e., vous avez oublié de raccorder la sortie Non (N) au branchement de programme, le message d'erreur serait le suivant :



Start

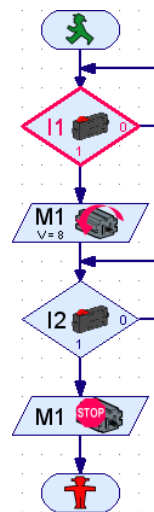


Si un message d'erreur apparaît, vous devez tout d'abord corriger l'erreur signalée. Dans le cas contraire, le programme ne pourra pas démarrer.

**Note Bene** : Vous trouverez une explication détaillée de ce mode d'utilisation et du mode d'utilisation „Téléchargement“ au point 3.7 sur la page 196.

Le premier élément de branchement est marqué en rouge. Cela indique que le déroulement de cet élément de programme attend que l'interrupteur raccordé à I1 (qui doit ouvrir la porte de garage) soit enfoncé. Tant que l'interrupteur n'est pas actionné, le programme se branche sur la sortie Non (N) du branchement du programme et, de là, repart vers le début du branchement. Appuyez maintenant sur l'interrupteur raccordé à l'entrée I1 de l'Interface. La condition nécessaire au transfert est alors remplie et le moteur s'allume. Le déroulement du programme attend alors que le commutateur de fin de l'entrée I2 soit actionné. Dès que le commutateur de fin de l'entrée I2 est actionné, le programme se branche sur la seconde Sortie Moteur et éteint alors le moteur. Le programme parvient finalement à la fin du programme. Un message vous informant que le programme est achevé apparaît alors.

Tout a fonctionné ? Félicitations ! Vous avez conçu et testé votre premier programme de commande. Si cela ne marche pas, ne vous découragez pas ! Vérifiez bien tout à nouveau ; une petite erreur a dû se glisser quelque part. Tous les programmeurs font des erreurs et c'est en faisant des erreurs que l'on apprend le mieux. Alors, courage !!!

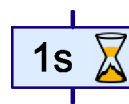


### 3.6 Autres éléments de programme

Si vous essayez votre programme de commande sur une vraie maquette de porte de garage, la porte est maintenant ouverte. Comment la fermer maintenant ? Nous pouvons évidemment redémarrer le moteur en appuyant sur un interrupteur ! Nous voulons cependant tester une autre solution et faire ainsi connaissance avec un nouvel élément de programme. Pour ce faire, réenregistrez le programme sous un nouveau nom (Nous aurons encore besoin de l'organigramme actuel plus tard). Pour ce faire, utilisez le point **Enregistrer sous...** dans le menu **Fichier** et introduisez un nom de fichier non attribué.

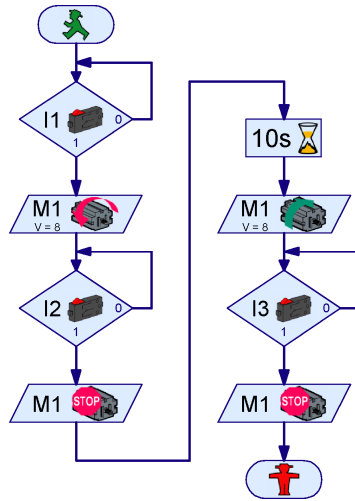
#### 3.6.1 Attente

Avant de pouvoir compléter l'organigramme, vous devez effacer la liaison entre „éteindre le moteur“ et „fin du programme“ et déplacer l'élément de fin vers le bas. Vous pouvez maintenant ajouter de nouveaux éléments de programme entre ces deux éléments. La porte de garage doit se refermer automatiquement après 10 secondes. Pour ce faire, vous pouvez utiliser l'élément de programme **Attente** représenté ci-contre. Vous pouvez régler le temps d'attente dans une large mesure selon votre envie en cliquant, comme d'habitude, sur l'élément avec le bouton droit de la souris. Introduisez alors la durée d'attente souhaitée (10 secondes). Pour fermer la porte de garage, le moteur doit, bien entendu, tourner dans l'autre sens, donc vers la droite. Le moteur se coupe grâce à un autre commutateur de fin I3.

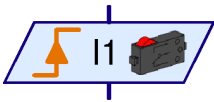




Votre organigramme complet devrait être plus ou moins semblable à celui représenté ici. Pour plus de clarté, les nouveaux éléments de programme ont été déplacés vers la droite. S'il n'y a plus d'erreur dans l'organigramme, vous pouvez, comme toujours, tester la commande étendue de la porte de garage en appuyant sur le bouton **Start**. En actionnant l'interrupteur raccordé sur I1, le moteur s'allume ; en actionnant l'interrupteur sur I2, il se coupe. La porte de garage est alors ouverte. L'élément de programme « Attente de 10 secondes » (le temps d'attente que nous avons programmé) est alors marqué en rouge. Le moteur est alors allumé avec un autre sens de rotation jusqu'à ce que l'interrupteur sur I3 soit actionné. Vous pouvez également essayer de modifier le temps d'attente.



### 3.6.2 Attendre l'entrée



Outre l'élément Attente, il existe deux autres éléments qui attendent un événement avant de poursuivre le déroulement du programme. L'**Elément Attendre l'entrée** représenté à gauche attend jusqu'à ce qu'une entrée de l'Interface ait un statut particulier ou se modifie d'une certaine manière. Il existe 5 variantes de cet élément :

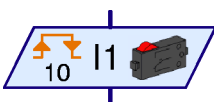
se modifie d'une certaine manière. Il existe 5 variantes de cet élément :

Symbole					
Attendre	Entrée=1 (fermé)	Entrée=0 (ouvert)	Permuter 0-1 (ouvert après fermé)	Permuter 1-0 (fermé après ouvert)	Permutation quelconque (1-0 ou 0-1)
Même fonction mais avec branchement					



Il est également possible d'utiliser une combinaison d'éléments de branchement ; l'opération est cependant plus simple et claire avec l'élément **Attendre l'entrée**.

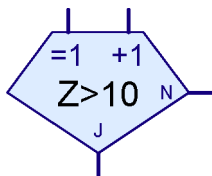
### 3.6.3 Compteur d'impulsions



De nombreuses maquettes de robots Fischertechnik utilisent également des engrenages d'impulsion. Les engrenages actionnent un interrupteur à quatre reprises à chaque rotation. De tels engrenages d'impulsion peuvent allumer un moteur pour un nombre défini de rotations plutôt que pour une période fixe. Pour ce faire, il faut compter le nombre d'impulsions sur une entrée de l'Interface. L'**Elément du compteur d'impulsions** représenté à gauche a été créé à cet effet et attend un nombre à déterminer d'impulsions. Avec

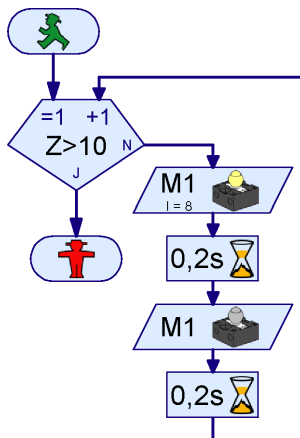
cet élément, vous pouvez déterminer ce qui doit être considéré comme une impulsion (modifications quelconques ou seulement 0-1 ou seulement une permutation 1-0). Les engrenages d'impulsion attendent généralement des modifications quelconques, de sorte que l'on atteigne une résolution de 8 étapes par rotation pour les 4 dents.

### 3.6.4 Boucle de comptage



L'élément Boucle de comptage vous permet de lancer une partie précise d'un programme à plusieurs reprises. Le programme représenté, par exemple, allume et éteint 10 fois une lampe raccordée à **M1**. L'élément boucle de comptage possède un comp-

teur intégré. Lorsque la boucle de comptage va arriver sur l'entrée **=1**, le compteur se positionne sur 1. Par contre, lorsque la boucle de comptage va arriver sur l'entrée **+1**, 1 est ajouté au compteur. Selon que le compteur est plus grand que la valeur que vous avez programmée ou pas, la boucle de comptage se branche sur la sortie Oui (**J**) ou sur Non (**N**). La sortie Oui est donc utilisée lorsque la boucle a été parcourue autant de fois que programmé dans la valeur du compteur. Si la boucle doit effectuer d'autres parcours, la boucle de comptage se branche sur la sortie Non. Comme pour les éléments de branchement, vous pouvez permuter les sorties Oui et Non dans la fenêtre de propriétés.



### 3.7 Utilisation en mode Online ou en mode Téléchargement Quelle est la différence ?



Start

Jusqu'à maintenant, nous avons testé nos programmes de commande en **Mode Online**. Vous pouvez ainsi suivre le déroulement du programme car l'élément actif est toujours marqué en rouge sur l'écran. Le Mode Online est utilisé pour comprendre les programmes ou détecter les erreurs.



Pause

En Mode Online, vous pouvez également arrêter le programme et le remettre en marche en appuyant sur le bouton **Pause**. C'est très pratique pour vérifier un point sur une maquette sans arrêter la totalité du programme. Si vous souhaitez comprendre le déroulement d'un programme, la fonction Pause peut également être très utile.



Etape

Le bouton **Etape** vous permet de lancer le programme étape par étape, élément par élément. A chaque fois que vous appuyez sur le bouton Etape, le programme passe à l'élément de programme suivant. Si vous avez introduit un élément **Attente** ou **Attendre...**, cela peut durer un moment avant que le programme n'arrive sur l'élément suivant.



Téléchargement

Si vous possédez ROBO Interface (pas Intelligent Interface), vous pouvez utiliser le **Mode Téléchargement** plutôt que le mode Online. En Mode Online, les programmes sont lancés par votre ordinateur. Votre ordinateur envoie des commandes comme „Allumer le moteur“ à l'Interface. Il est nécessaire que l'Interface soit reliée à l'ordinateur pendant toute la durée du programme. En Mode Téléchargement, le programme est lancé par l'ordinateur lui-même. Votre ordinateur enregistre le programme dans l'Interface. Lorsque c'est fait, la liaison entre l'ordinateur et l'Interface peut être coupée. L'Interface peut alors lancer le programme de commande indépendamment de l'ordinateur. Le Mode Téléchargement est important, p.e., pour la programmation de robots mobi-



les pour lesquels un câble de liaison entre l'ordinateur et le robot représente un vrai handicap. Cependant, tous les programmes de commande devraient être testés au préalable en Mode Online car les erreurs possibles peuvent alors être mieux détectées. Le programme testé peut alors être transféré par téléchargement sur ROBO Interface. Pour ROBO Interface; le câble gênant peut être remplacé par une liaison radio **ROBO RF Data Link**, N° d'article 93295. La maquette est alors totalement mobile même en Mode Online.

Le Mode Online a cependant également des avantages par rapport au Mode Téléchargement. En comparaison avec l'Interface, un ordinateur a une mémoire de travail beaucoup plus étendue et peut calculer beaucoup plus rapidement. Pour des programmes de grande taille, il s'agit là d'un avantage. Qui plus est, en Mode Online, plusieurs Interfaces peuvent être commandées en parallèle, même un mélange de ROBO Interfaces et d'Intelligent Interfaces.

Aperçu des deux modes d'utilisation :

Mode d'utilisation	Avantage	Inconvénient
Online	<ul style="list-style-type: none"> <li>• Possibilité de suivre le déroulement du programme sur l'écran</li> <li>• Très grande rapidité de déroulement même pour des programmes de grande taille</li> <li>• Commande de plusieurs Interfaces en parallèle</li> <li>• Prise en charge de l'ancienne Intelligent Interface</li> <li>• Possibilité d'utilisation de panneaux de commande</li> <li>• Possibilité de suspendre et de reprendre le déroulement du programme</li> </ul>	<ul style="list-style-type: none"> <li>• Obligation de garder l'ordinateur et l'Interface reliés</li> </ul>
Téléchargement	<ul style="list-style-type: none"> <li>• Possibilité de séparer l'ordinateur et l'Interface après le Téléchargement</li> </ul>	<ul style="list-style-type: none"> <li>• Pas de prise en charge de l'ancienne Intelligent Interface</li> <li>• Impossibilité de suivre le déroulement du programme sur l'écran</li> <li>• Possibilité de commander seulement 3 modules d'extension</li> </ul>



Téléchargement

## Utiliser le mode Téléchargement

Si vous possédez la nouvelle ROBO Interface, vous pouvez transférer la commande de votre porte de garage sur l'Interface en appuyant sur le bouton **Téléchargement** de la barre d'outils. La fenêtre de dialogue représentée à droite apparaîtra alors. ROBO Interface dispose de plusieurs adresses de mémoire pour les programmes : une mémoire **RAM** (Random Access Memory) et deux mémoires **Flash**. Un programme enregistré sur la mémoire RAM sera perdu lorsque l'Interface ne sera plus alimentée en courant ou que les batteries seront vides. Un programme enregistré dans une mémoire Flash restera au contraire enregistré sur l'Interface pendant des années, même si celle-ci n'est pas alimentée en courant. Vous pouvez bien évidemment écraser des programmes enregistrés sur une mémoire Flash. Le Téléchargement sur la mémoire RAM est cependant nettement plus rapide et est donc recommandé pour les tests.



Vous pouvez enregistrer plusieurs programmes dans les deux mémoires Flash, par exemple, deux comportements différents pour un même robot mobile. Vous pouvez alors sélectionner, démarrer ou arrêter les deux programmes avec la touche **Prog** de l'Interface. Lorsque l'option **Lancer le programme après le téléchargement** est activée, le programme se lance directement après le téléchargement. Pendant le déroulement du programme, la DEL verte **Prog 1** (programme chargé dans Flash 1) ou **Prog 2** (programme chargé dans Flash 2) clignote sur ROBO Interface à côté de la touche **Prog**. Si un programme est chargé dans la mémoire RAM, les deux DELs clignent. Pour arrêter le programme, appuyez sur la touche **Prog**. La DEL reste alors allumée. Pour passer du programme 1 au programme 2, maintenez la touche Prog enfoncée jusqu'à ce que la DEL du programme souhaité clignote. Pour lancer le programme, appuyez à nouveau sur cette touche.

L'option **Lancer le programme via la touche de l'interface** prend tout son sens pour les robots mobiles. Si vous ne possédez pas de RF Data Link, vous devez raccorder le câble avant que votre programme ne puisse mettre le robot en mouvement. Dans ce cas, vous devez tout d'abord sélectionner le programme souhaité via la touche Prog de l'Interface et lancer le programme en appuyant à nouveau sur cette touche.

Lorsque la dernière option **Lancer automatiquement au démarrage** est activée, le programme 1 de la mémoire Flash est lancé automatiquement quand l'Interface est alimentée en courant. Vous pouvez alors, p.e., alimenter votre Interface en courant au moyen d'un bloc d'alimentation pourvu d'un minuteur et lancer le programme chaque jour au même moment. Ainsi, l'Interface ne doit pas rester allumée en permanence et le programme ne doit pas à chaque fois être lancé via le bouton Prog après l'allumage.

### Nota Bene :

Lorsqu'un programme est chargé dans la mémoire Flash ou exécuté à parti de cette mémoire, les programmes enregistrés dans la RAM sont perdus car les programmes de la mémoire Flash utilisent également la mémoire RAM.

Vous trouverez une description détaillée des fonctions de ROBO Interface dans les instructions d'utilisation d'Interface.

## 3.8 Trucs et Astuces

### Modifier les lignes de liaison

Lorsque vous déplacez des éléments, ROBO Pro s'efforce d'adapter les lignes de liaison en conséquence. Si une des lignes adaptées devait ne pas vous convenir, vous pouvez légèrement modifier les lignes de liaison en cliquant sur celles-ci avec le bouton gauche de la souris et en déplaçant la ligne en gardant le bouton enfoncé. En fonction de l'endroit où la souris se trouve sur la ligne, un point d'angle ou une arrête de la ligne sera déplacé. Ceci sera indiqué par différents curseurs :



Si la souris se trouve sur une ligne de liaison verticale, vous pouvez déplacer l'ensemble de la ligne verticale en appuyant sur le bouton gauche de la souris.



Si la souris se trouve sur une ligne de liaison horizontale, vous pouvez déplacer l'ensemble de la ligne horizontale en appuyant sur le bouton gauche de la souris.



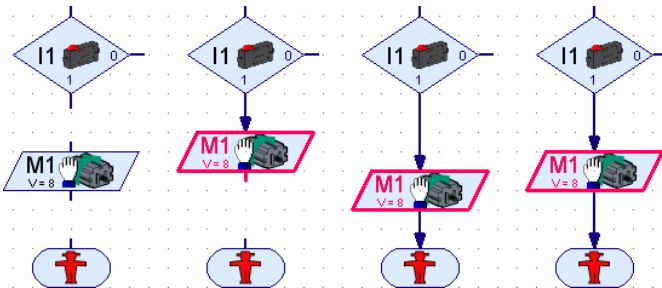
Si la souris se trouve sur une ligne de liaison oblique, un nouveau point sera ajouté sur la ligne de liaison si vous appuyez sur le bouton gauche de la souris. Vous devez garder le bouton droit de la souris enfoncé et ne le lâcher que lorsque la souris se trouve là où le point doit se positionner.



Si la souris se trouve à proximité d'un point d'angle ou d'une extrémité d'une ligne de liaison, vous pouvez déplacer ce point en appuyant sur le bouton gauche de la souris. Vous pouvez alors placer un point d'extrémité sur un autre raccordement d'un élément de programme. Dans ce cas, l'extrémité de la ligne de liaison sera reliée à ce point de raccordement. Dans le cas contraire, le point ne sera pas déplacé.

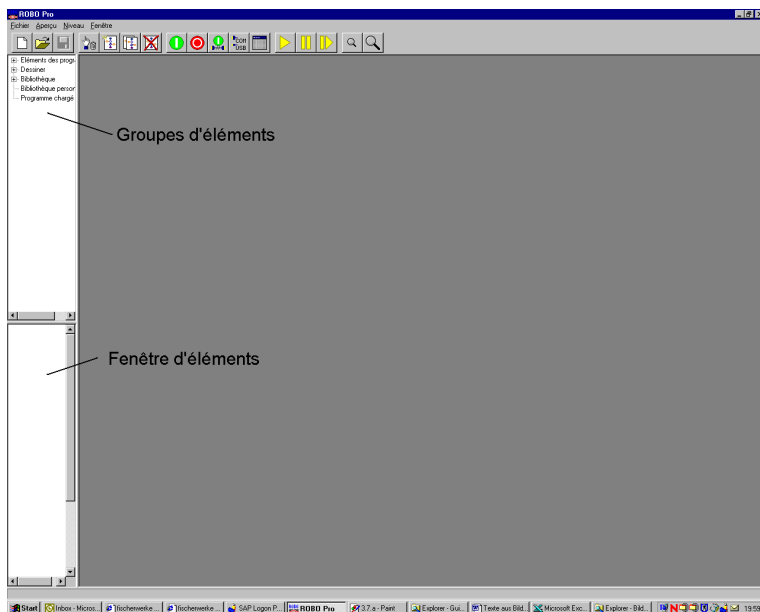
### Modifier les lignes de liaisons une fois

Des lignes de liaisons peuvent également être tracées lors du déplacement d'éléments de programme. Si vous déplacez un élément de programme de sorte que son entrée se trouve un ou deux points de trame sous la sortie d'un autre élément, une ligne de liaison apparaît entre ces deux éléments. Ceci est aussi valable si une sortie est déplacée vers une entrée. L'élément de programme peut ensuite être placé dans sa position définitive ou d'autres liaisons peuvent être tracées pour les entrées et sorties restantes :



## 4 Niveau 2: Travailler avec des sous-programmes

Après avoir créé et testé avec succès votre premier programme de commande, vous êtes prêt pour ROBO Pro Niveau 2. Sélectionnez maintenant l'entrée **Niveau 2 : Sous-programme** dans le menu **Niveau**. Vous remarquerez directement la différence : La fenêtre d'éléments a disparu et vous trouverez à sa place, sur le bord gauche, deux fenêtres superposées :



Pas de panique ! La fenêtre d'éléments est encore là, mais elle est vide. Au Niveau 2, il y a plus d'éléments de programme, de sorte que ce ne serait pas clair si on les rassemblait tous dans une fenêtre. A partir du Niveau 2, les éléments sont donc regroupés en groupes d'éléments. Les éléments sont organisés en groupes comme les fichiers dans les dossiers sur un disque dur. Si vous choisissez un groupe dans la fenêtre supérieure, sur le côté gauche, tous les éléments de ce groupe apparaissent dans la fenêtre inférieure. Les éléments du Niveau 1 se trouvent dans le groupe **Eléments de programme / Eléments du fond**. Etant donné que la fenêtre d'éléments est diminuée de moitié, vous devez utiliser la barre de défilement à droite de la fenêtre d'éléments afin de pouvoir accéder aux éléments inférieurs.

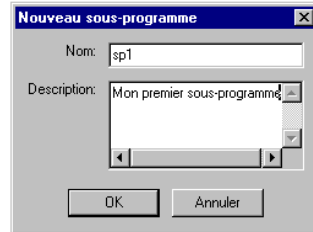
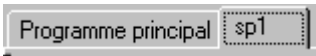
Revenons-en au sujet principal : les sous-programmes ! Les organigrammes que nous avons conçus jusqu'ici ne sont pas encore assez étendus pour que nous puissions perdre une vision globale de ceux-ci, mais vous pouvez certainement vous imaginer que cela peut très facilement être le cas pour les projets de grande envergure, avec des organigrammes étendus. Votre feuille de travail est alors remplie d'éléments, il y a des lignes de liaison partout et, sur l'écran, vous devez sans cesse aller d'un côté à un autre avec la barre de défilement. "Où était donc telle ou telle sortie ?" Bref – On frôle le chaos ! Que faire ? N'y a-t-il pas une solution pour remettre de l'ordre dans tout ça ? Si – Cette solution s'appelle **Sous-programme** !

## 4.1 Mon premier sous-programme

Un sous-programme ressemble fort à un programme comme ceux que vous avez rencontrés jusqu'ici. Afin de pouvoir vous en rendre compte dans les détails, il va falloir créer un nouveau programme et, dans celui-ci, un sous-programme vide. Pour ce faire, cliquez sur **Nouveau** programme et ensuite sur le bouton **Nouveau SP** de la barre d'outils. Une fenêtre apparaît alors dans laquelle vous pouvez introduire un nom et une description pour votre sous-programme.

Le nom ne doit pas être trop long (env. 8-10 lettres) car, dans le cas contraire, le symbole du sous-programme sera très grand. Vous pourrez bien entendu modifier plus tard toutes les saisies que vous réalisez ici.

Dès que vous fermez la fenêtre **Nouveau Sous-programme** avec **OK**, le nouveau sous-programme apparaît dans la barre des sous-programmes :



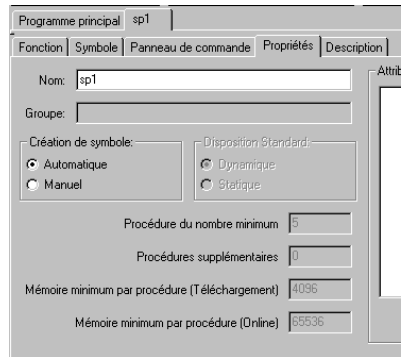
Vous pouvez à tout moment basculer entre le programme principal et le sous-programme en cliquant sur le nom du programme dans la barre des sous-programmes. Etant donné que les deux programmes sont encore vides, il n'y a pas encore de différences.

Nous allons maintenant diviser la commande de notre porte de garage du chapitre précédent (voir point 3.6 *Autres éléments de programme* sur la page 194) en sous-programmes. Ce programme se composait de quatre unités de fonctionnement :

- Attendre jusqu'à ce que l'interrupteur I1 soit actionné
- Ouvrir la porte
- Attendre 10 secondes
- Fermer la porte

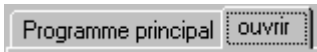
L'ouverture et la fermeture seront désormais divisées en deux sous-programmes. Vous pouvez appeler les deux sous-programmes au départ du programme principal avec un seul symbole. L'attente de l'interrupteur I1 et l'attente de 10 secondes restent dans le programme principal car les deux se composent d'un seul et unique élément. Vous avez déjà créé un nouveau programme avec un sous-programme nommé **Sous-programme 1**. **Ouvrir** et **Fermer** seraient cependant de bien meilleurs noms pour les deux sous-programmes. Vous pouvez renommer le sous-programme déjà créé en sélectionnant tout d'abord le sous-programme 1 dans la barre des sous-programmes, pour autant qu'il ne soit pas déjà sélectionné.

Basculez vers la fenêtre de propriétés du sous-programme via la barre de fonctions en cliquant sur **Propriétés**. Ici, vous pouvez modifier le nom **SP 1** et le renommer **Ouvrir**. La plupart des autres champs ne pourront être modifiés qu'au niveau Avancés ou même au niveau Experts. Le point **Création de symbole** sera également expliqué un peu plus tard.



Si vous cliquez sur **Description** dans la barre de fonctions, vous pouvez modifier la description introduite au préalable, bien que „Mon premier sous-programme“ soit, comme auparavant, très approprié.

Cliquez ensuite sur **Fonction** dans la barre de fonctions afin de pouvoir programmer les fonctions du sous-programme. Vous voyez maintenant la fenêtre de programme dans laquelle vous avez déjà ajouté les éléments de programme de votre premier programme ROBO Pro au chapitre précédent. Veillez à ce que le sous-programme **Ouvrir** soit sélectionné dans la barre des sous-programmes :



Etes-vous prêt à programmer votre premier sous-programme ? Allons-y ! Mais par quoi peut bien commencer un sous-programme ? Bonne question ! Vous avez toujours débuté un programme principal avec un élément de démarrage. Un sous-programme commence avec un élément semblable, l'entrée du sous-programme. Cet élément porte ce nom car le déroulement du programme passe du programme principal au sous-programme via cet élément. Vous ne pouvez pas utiliser d'élément de démarrage ici car il ne s'agit pas de démarrer une nouvelle procédure.




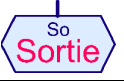
	Élément de démarrage	Démarre une nouvelle procédure indépendante
	Entrée du sous-programme	Le déroulement du programme passe ici du programme principal au sous-programme

Vous trouverez l'entrée du sous-programme dans la fenêtre des groupes d'éléments sous **Sous-programme I/O**. Placez maintenant l'entrée du sous-programme en haut dans la fenêtre de programme du sous-programme **Ouvrir**. Vous pouvez également donner un nom différent à l'élément d'entrée du sous-programme pour remplacer le nom **Entrée**, ce n'est cependant nécessaire que lorsque vous écrirez un sous-programme avec plusieurs entrées.

Eléments des programmes  
- Éléments du fond  
- Sous-programme I/O

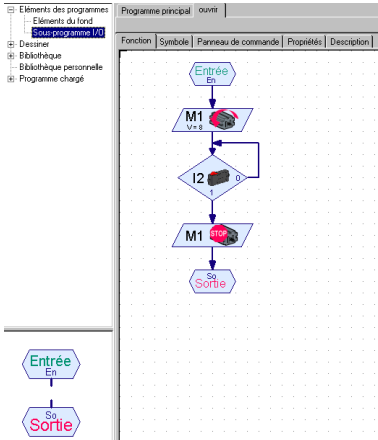
La suite du déroulement du programme dans un sous-programme est identique à la partie Ouverture du programme principal que nous avons jusqu'ici : Allumez le moteur M1 avec un sens de rotation vers la gauche, attendez jusqu'à ce que l'interrupteur sur l'entrée I2 soit éteint et éteignez le moteur.

Pour terminer le sous-programme, utilisez une sortie du sous-programme. La différence entre la sortie du sous-programme et l'élément de fin est la même qu'entre l'entrée du sous-programme et l'élément de démarrage.

	Élément de fin	Termine le déroulement du programme d'une procédure indépendante
	Sortie du sous-programme	Le déroulement du programme repasse du sous-programme au programme principal.

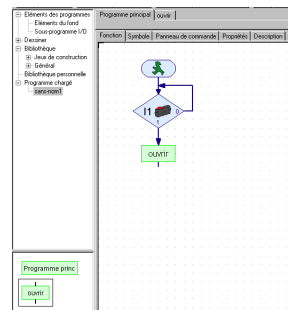


Le sous-programme achevé doit plus ou moins ressembler à ceci :



Veillez à ce que le sous-programme ait bien été introduit sous **Ouvrir** et non sous **Programme principal**. Basculez à nouveau de **Ouvrir** à **Programme principal** dans la barre des sous-programmes. Vous voyez maintenant la fenêtre de programme vide, comme auparavant. Comme d'habitude, ajoutez un élément de démarrage dans le programme principal (pas une entrée du sous-programme !). Ajoutez également dans le programme principal une requête envoyée à l'interrupteur sur I1, qui doit ouvrir la porte de garage.

Vous pouvez maintenant ajouter votre nouveau sous-programme dans le programme principal (ou un autre sous-programme) comme s'il s'agissait d'un élément de programme habituel. Vous le trouverez dans la fenêtre des groupes d'éléments sous **Programmes chargés** et le nom de votre programme. Si vous n'avez pas encore enregistré le fichier, son nom est **sans-nom1**. Si vous avez chargé plusieurs fichiers Programme, vous pouvez également sélectionner des sous-programmes appartenant à d'autres fichiers. Vous pouvez ainsi utiliser très simplement des sous-programmes provenant d'un autre fichier.



Dans le groupe d'éléments **Programmes chargés / sans-nom1**, vous trouverez deux symboles de sous-programmes verts. Le premier porte le nom **Programme principal** et est le symbole du programme principal. Il sera rarement utilisé comme sous-programme; ceci est cependant possible, p.e., si vous commandez tout un parc de machines et que vous avez déjà développé des commandes individuelles pour les machines dans des programmes principaux. Le second symbole porte le nom **Ouvrir** et est le symbole de votre nouveau sous-programme. **Ouvrir** est le nom que vous avez introduit dans la fenêtre des propriétés. Ajoutez maintenant le symbole du sous-

programme dans le programme principal comme vous le faisiez pour les éléments de programme habituels. C'est aussi simple que ça !

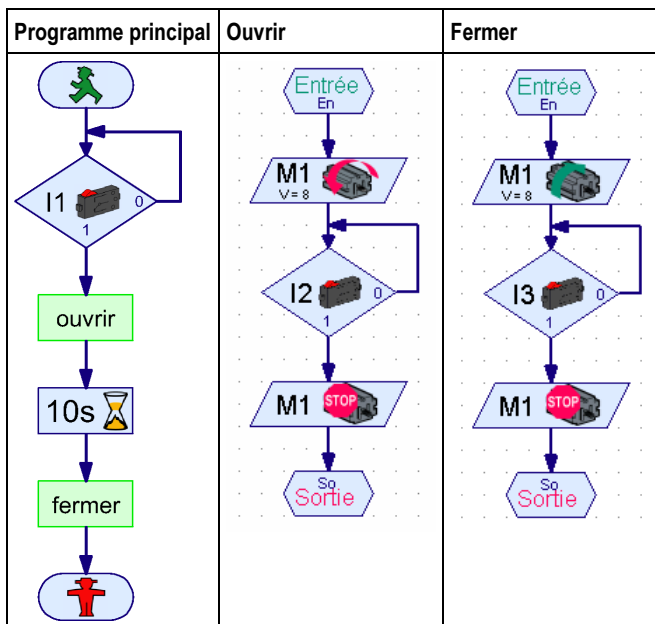


Nouveau  
SP

Vous pouvez déjà terminer votre programme principal avec un élément de fin et le tester. La porte s'ouvre lorsque l'on actionne l'interrupteur I1, mais nous n'avons pas encore programmé la partie Fermeture. Pour ce faire, il vous faut créer un nouveau sous-programme. Cliquez sur le bouton **Nouveau SP** de la barre d'outils et introduisez le nom **Fermer** dans la fenêtre **Nouveau Sous-programme**. Vous ne devez pas introduire de description mais vous pouvez le faire afin de toujours savoir à quoi sert le sous-programme.

Insérez alors dans la fenêtre de programme du sous-programme **Fermer** le programme de fermeture de la porte de garage. Commencez à nouveau avec une entrée du sous-programme. Le moteur **M1** doit maintenant tourner vers la droite. Dès que le commutateur de fin sur **I3** est fermé, le moteur **M1** doit s'arrêter. Le sous-programme se termine à nouveau par une sortie du sous-programme.

Basculez à nouveau avec la barre des sous-programmes vers le programme principal. Si vous avez terminé votre programme principal avec un élément de fin pour pouvoir le tester, l'élément de fin doit maintenant être supprimé. Après l'ouverture de la porte, celle-ci doit rester ouverte 10 secondes avant de se fermer à nouveau. Après une attente de 10 secondes, ajoutez le symbole **Fermer** du sous-programme en allant le chercher dans les groupes d'éléments **Programmes chargés / sans-nom1**. Le programme principal et les deux sous-programmes doivent être plus ou moins semblables à ceci :



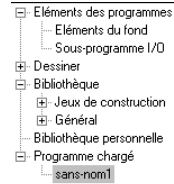
Le programme commence avec l'élément de démarrage du **Programme principal**. Le programme attend alors que l'interrupteur **I1** soit actionné. Pour ce faire, vous pouvez également utiliser l'élément **Attendre une entrée** (voir point 7.1.8 *Attendre une entrée* sur la page 233 ). Après que l'interrupteur **I1** ait été actionné, le programme principal arrive sur l'appel du sous-programme **Ouvrir**. Le déroulement du programme bascule alors vers l'entrée du sous-programme du sous-



programme **Ouvrir**. Le sous-programme **Ouvrir** ouvre la porte du garage et arrive ainsi sur la sortie du sous-programme. A cet endroit, le programme se branche à nouveau sur le programme principal. Après la fin du sous-programme **Ouvrir**, le programme principal attend 10 secondes. Le déroulement du programme bascule alors vers le sous-programme **Fermer**, qui ferme la porte du garage. Après la fin du sous-programme **Fermer**, le programme principal arrive sur l'élément de fin qui termine le programme.

## 4.2 Bibliothèque des sous-programmes

Vous pouvez très bien copier des sous-programmes d'un fichier dans un autre fichier en chargeant les deux fichiers et en ajoutant un sous-programme d'un fichier dans l'autre fichier via le groupe d'éléments **Programmes chargés**. Pour les sous-programmes les plus utilisés, c'est encore plus simple, il suffit d'utiliser la **Bibliothèque**. ROBO Pro contient une bibliothèque de sous-programmes finis que vous pouvez utiliser très simplement. Vous pouvez également créer votre propre bibliothèque dans laquelle vous pourrez enregistrer les sous-programmes que vous utilisez le plus souvent.



### 4.2.1 Utiliser la bibliothèque

La **Bibliothèque** est d'abord divisée en deux groupes principaux. Dans le groupe **Jeux de construction**, vous trouverez les sous-programmes que vous pouvez utiliser pour les maquettes composées de certains jeux de construction. Dans le groupe **Général**, vous trouverez les sous-programmes que vous pouvez utiliser pour tous les types de maquettes. La plupart des sous-programmes du groupe **Général** exigent cependant des techniques du Niveau 3 qui ne seront vues qu'au chapitre suivant.

Pour chaque jeu de construction commandé par ordinateur, p.e. le ROBO Mobile Set, il existe un sous-groupe dans le groupe **Jeux de construction**. Parfois, ce sous-groupe est encore subdivisé par modèles (que vous trouverez dans les instructions de montage du jeu de construction). Si vous sélectionnez le jeu de construction ou une des maquettes, les sous-programmes existant pour cette maquette apparaîtront dans la fenêtre d'éléments.

Si vous passez avec la souris sur un des symboles de sous-programme, une courte description apparaît. Si vous ajoutez un sous-programme dans votre programme, vous pouvez afficher une description précise en sélectionnant le sous-programme dans la barre des sous-programmes et en cliquant sur **Description** dans la barre de fonction :



**Attention:** Lorsque vous ajoutez un sous-programme provenant de la bibliothèque, d'autres sous-programmes utilisés par ce sous-programme sont en partie ajoutés. Vous pouvez supprimer tous les sous-programmes en sélectionnant la fonction **Annuler** du menu **Traiter**.

### 4.2.2 Utiliser la bibliothèque personnelle

Après vous être entraîné un moment avec ROBO Pro, vous avez sans doute quelques sous-programmes que vous utilisez plus souvent. Pour ne pas devoir à chaque fois rechercher et charger les fichiers correspondant, vous pouvez créer votre propre bibliothèque de sous-programmes ; celle-ci fonctionnera comme la bibliothèque prédéfinie. Votre bibliothèque personnelle se compose

d'un ou plusieurs fichiers ROBO Pro enregistrés dans un dossier. Un groupe propre s'affiche dans la sélection des groupes pour chaque fichier de ce dossier.

Dans le menu **Fichier** sous **Liste des bibliothèques personnelles**, vous pouvez choisir dans quel dossier vous souhaitez enregistrer votre bibliothèque personnelle. Le répertoire standard pour votre bibliothèque personnelle est C:\Programme\ROBOPro\Bibliothèque personnelle. Si vous disposez d'un répertoire utilisateur personnel sur votre ordinateur, il est recommandé d'y créer un dossier propre et d'utiliser celui-ci.

**Astuce:** Au début, vous pouvez indiquer le dossier dans lequel vous enregistrez vos programmes ROBO Pro pour la **Liste des bibliothèques personnelles**. Vous avez ainsi accès plus rapidement à tous les sous-programmes de tous les fichiers dans votre dossier de travail.

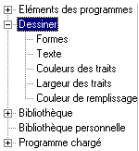
### Organiser ma bibliothèque personnelle

Il n'y a pas de fonctions spéciales pour modifier une bibliothèque dans ROBO Pro. C'est cependant très simple. Si vous souhaitez ajouter ou supprimer des sous-programmes dans un groupe de bibliothèque, vous devez tout d'abord charger le fichier correspondant. Vous trouverez ce fichier dans le répertoire que vous avez choisi sous **Liste des bibliothèques personnelles**. Vous pouvez maintenant, p.e., charger un second fichier ou déplacer un sous-programme du groupe **Programmes chargés** vers le programme principal de la bibliothèque. Dans la bibliothèque, le programme principal n'est pas vraiment un programme, mais seulement une collection de tous les sous-programmes de la bibliothèque. Le programme principal n'est pas affiché dans la fenêtre d'éléments des bibliothèques. Vous pouvez évidemment supprimer des sous-programmes d'une bibliothèque ou les modifier.

Si vous avez modifié et enregistré un fichier de la bibliothèque, vous devez sélectionner le point **Actualiser la bibliothèque personnelle** dans le menu **Fichier**. La liste des fichiers de la fenêtres des groupes est alors actualisée.

### 4.3 Traiter les symboles des sous-programmes

Comme vous l'avez vu au chapitre précédent, ROBO Pro crée automatiquement des symboles de sous-programme verts pour vos sous-programmes. Vous pouvez aussi créer vos propres symboles exprimant mieux ce que font vos sous-programmes. Pour ce faire, vous devez basculer de création de symbole automatique à manuelle dans la fenêtre de propriétés du sous-programme. Vous pouvez ensuite passer de **Propriétés** à **Symbole** dans la barre de fonctions et y traiter le symbole du sous-programme. Vous trouverez les fonctions de dessin dans la fenêtre des groupes d'éléments sous **Dessiner**.

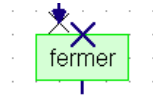


Dans le champ **Signes / Formes**, vous trouverez tous les éléments graphiques de fond usuels comme le rectangle, le cercle, l'ellipse, le polygone et d'autres. Dans le champ **Signes / Texte**, vous trouverez les objets Texte en différentes tailles. Dans les autres groupes du menu Dessiner, vous trouverez les fonctions servant à modifier les couleurs et les propriétés similaires des éléments sélectionnés. L'utilisation détaillée des fonctions de dessin est expliquée au chapitre 9 *Fonctions de dessin* sur la page 261. Regardez également les

fonctions du menu principal sous **Dessiner**.

Vous pouvez également déplacer les raccords du sous-programme, mais vous ne pouvez pas supprimer ou ajouter de nouveaux raccords. Dans le symbole du sous-programme, il y a toujours un raccord pour chaque entrée ou sortie du sous-programme de la fonction du programme. Les éléments de raccordement sont également créés automatiquement lorsque vous sélectionnez la création manuelle de symbole.

Dès que vous quittez la fenêtre de traitement des symboles, tous les appels du sous-programme dans le programme principal ou dans les autres sous-programmes sont adaptés en conséquence. Veuillez noter que, si les raccords étaient déjà raccordés, une légère confusion peut se produire lors de l'appel d'un sous-programme lorsque vous déplacez des raccords dans un sous-programme. Les extrémités des lignes de liaison ne se trouvent parfois plus sur le bon raccordement, ce qui est indiqué par une croix sur l'extrémité de la ligne et sur le raccordement. (Voir image). Généralement, il suffit de cliquer n'importe où sur la ligne de liaison avec le bouton gauche de la souris. La ligne est alors remise en place automatiquement. Dans les sous-programmes contenant de nombreuses liaisons, il se peut que vous deviez encore retravailler la ligne.

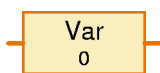


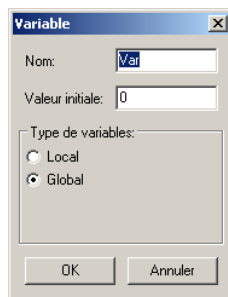
## 5 Niveau 3: Variables, Panneau de commande et Cie

N'oubliez pas de régler ROBO Pro sur **Niveau 3** (ou plus) dans le menu **Niveau** !

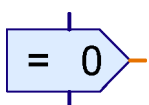
Imaginez-vous que vous découvrez une machine fascinante dans un des recoins inexplorés d'un musée et que vous vouliez absolument la reproduire avec Fischertechnik. En étudiant la machine, vous oubliez cependant le temps et ne remarquez pas que tous les autres visiteurs ont quitté le musée. Ce n'est qu'après la fermeture du musée que vous avez suffisamment étudié la machine pour pouvoir la reproduire. Mais vous allez devoir passer une mauvaise nuit tout seul dans le musée avant de pouvoir vous en aller. Pour que cela n'arrive plus, vous demandez au directeur du musée de programmer un compteur de visiteurs qui compte tous les visiteurs entrant et sortant et allume une lampe d'alarme rouge tant qu'il y a des visiteurs dans le musée. Mais comment faire ? Comment compter quelque chose avec ROBO Pro ? La réponse est... avec des **Variables**.

### 5.1 Variables et commandes

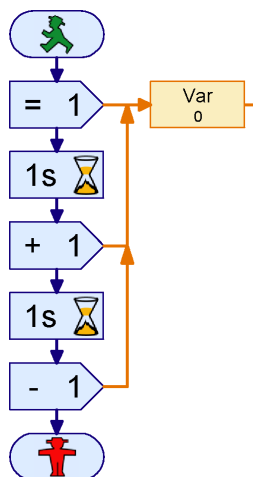
 Une variable est un élément pouvant afficher un chiffre. Introduisez un **Nom** dans la fenêtre de propriétés ; celui-ci doit donner des indications sur le type de chiffre enregistré dans la variable. Dans le champ **Valeur initiale**, vous pouvez indiquer quel chiffre doit être enregistré dans la variable au début du programme. Le réglage **Type de variables** est expliqué au point 7.3.2 *Variable locale* sur la page 238).



Le chiffre enregistré dans la variable peut être modifié en envoyant des ordres à cette variable. Une variable comprend 3 ordres différents : =, + et -. La commande = remplace le chiffre enregistré par un nouveau chiffre. Les commandes + et - ajoutent ou soustraient quelque chose au chiffre enregistré.

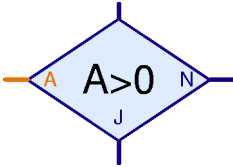
 Vous envoyez les commandes à la variable au moyen d'un **élément de commande**. Comme la plupart des autres éléments de programme, l'élément de commande a, sur sa partie supérieure, une entrée de programme bleue et, sur sa partie inférieure, une sortie de programme bleue. Ce qui est nouveau, c'est qu'il a également un raccordement **orange** sur son côté droit. Il s'agit d'une sortie de commande. A chaque fois que le déroulement passe sur l'élément de commande, il envoie une commande à tous les éléments raccordés via cette sortie. La variable a une entrée de commande adaptée sur son côté gauche. Si vous reliez la sortie de commande à l'entrée de commande, ROBO Pro trace une ligne orange à la place des liaisons bleues habituelles. Les éléments de programme peuvent envoyer des commandes ou des informations par ces lignes orange et peuvent ainsi échanger des informations.

Le programme représenté à droite envoie tout d'abord une commande = 1 à la variable **Var**. Une commande se compose



généralement d'une commande à proprement parlé, comme =, et d'une valeur, comme 1. La commande = 1 met la variable sur 1. Après une seconde, le programme envoie un ordre +1 à la variable. La variable ajoute alors 1 à sa valeur et a alors une valeur 2. Après une nouvelle seconde, le programme envoie la commande -1. La variable a alors à nouveau une valeur 1.

Essayez maintenant de concevoir ce programme simple dans ROBO Pro. Vous trouverez les éléments de commande dans le groupe **Commandes**, les variables dans le groupe **Variables**, **Timer**,... Si vous lancez le programme en Mode Online, vous verrez comment la valeur des variables change.

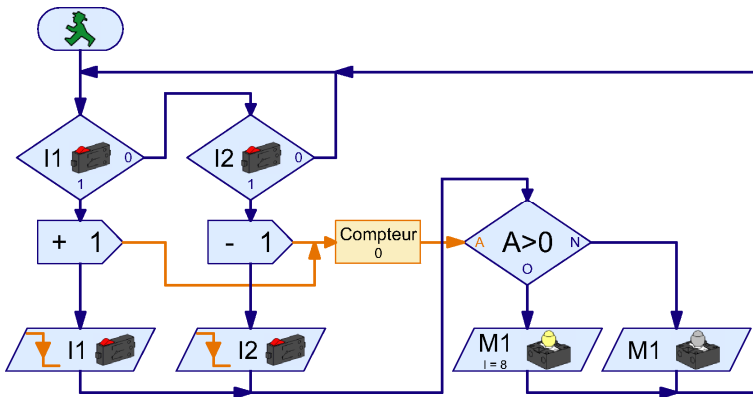


Vous vous dites sans doute : "Je peux voir la valeur des variables, mais qu'est-ce que je peux faire avec ça ?" C'est très simple : La variable a un raccordement orange sur sa droite via lequel vous pouvez envoyer des messages avec la valeur actuelle à tous les éléments raccordés. Certains éléments de ROBO Pro ont une entrée orange à gauche et peuvent être reliés à la sortie des variables. Par exemple, vous trouverez dans le groupe **Branchement**,

**Attente**,... un élément de branchement Oui / Non que n'envoie pas

directement une requête à une entrée, mais peut envoyer une requête à une valeur quelconque, entre autres la valeur d'une variable.

Le compteur de visiteurs du musée peut ainsi être programmé comme suit :



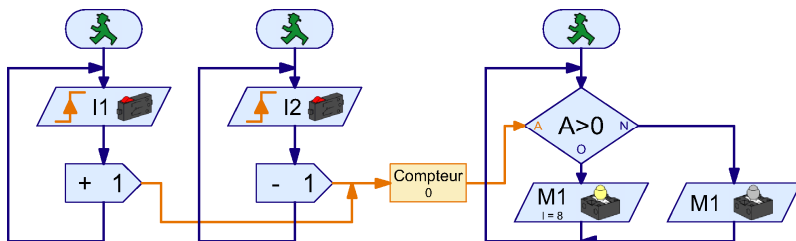
Le croisillon à l'entrée du musée actionne le bouton-poussoir sur I1, le croisillon à la sortie du musée actionne le bouton-poussoir sur I2. Dès que I1 est actionné, le programme envoie une commande + 1 à la variable **Compteur**. Le programme attend ensuite que le bouton-poussoir sur I2 soit à nouveau relâché. Il se passe la même chose avec le bouton-poussoir sur I2 ; dans ce cas, une commande - 1 est envoyée à la variable **Compteur**. A chaque fois que le compteur change, l'état du compteur est contrôlé. Lorsque la variable **Compteur** a une valeur > 0, la lampe d'alarme rouge sur M1 s'allume, dans le cas contraire, elle est éteinte.



Reproduisez maintenant le programme représenté ci-dessus et testez-le. Dès que vous appuyez sur le bouton-poussoir sur I1 et que vous le relâchez, la lampe d'alarme sur M1 s'allume. Si vous actionnez le bouton-poussoir sur I2, elle s'éteint à nouveau. Si vous actionnez I1 à plusieurs reprises, I2 doit être actionné autant de fois pour que la lampe d'alarme s'éteigne. Essayez de voir ce qu'il se passe si 5 visiteurs arrivent, que 2 s'en vont et que 3 arrivent à nouveau. Combien de fois devez-vous alors actionner le bouton-poussoir I2 pour que la lampe d'alarme s'éteigne à nouveau ?

## 5.2 Variables et procédures multiples

Peut-être avez-vous remarqué en testant le compteur de visiteurs qu'un problème survenait lorsque I1 et I2 étaient actionné en même temps. Tant qu'un des boutons-poussoirs est enfoncé, le programme ne peut pas réagir aux autres boutons-poussoirs. Après que les visiteurs soient passés respectivement par les croisillons d'entrée et de sortie au même moment, il y a des problèmes de comptage. Vous pouvez corriger ces erreurs en utilisant plusieurs procédures en parallèle. Jusqu'ici tous les programmes n'avaient qu'un seul élément de démarrage. Vous pouvez cependant utiliser plusieurs éléments de démarrage. Tous les déroulements ayant un élément de démarrage propre seront alors traités en parallèle. Les spécialistes parlent alors de **Procédures parallèles**. Cette technique vous permet de modifier le programme du compteur de visiteurs comme suit :



Des procédures indépendantes sont alors utilisées pour I1 et I2. Lorsque le bouton-poussoir I1 est enfoncé, la procédure pour I2 reste indépendante et peut continuer à surveiller le bouton-poussoir I2. Pour envoyer une requête aux valeurs de comptage et pour allumer et éteindre la lampe d'alarme, une procédure propre est alors utilisée.

Comme vous le voyez, il n'y a pas de problèmes pour accéder à une seule variable au départ de plusieurs procédures. Vous pouvez envoyer des commandes provenant de plusieurs procédures à une variable et utiliser la valeur d'une variable dans plusieurs procédures. Les variables sont donc adaptées aux échanges d'informations entre procédures.



Le directeur du musée est tellement enchanté de votre génial compteur de visiteurs qu'il vous demande de résoudre un autre problème : le musée a organisé une exposition. Etant donné que tous les visiteurs veulent voir la nouvelle exposition, il y a une telle foule que plus personne ne peut rien voir. Le directeur aimerait limiter le nombre de visiteurs de cette exposition à 10. Le directeur a fait placer des croisillons à l'entrée et à la sortie de l'exposition. Le croisillon d'entrée peut être verrouillé électroniquement. Maintenant il n'a plus besoin de d'un programmeur compétent... Ce programmeur, c'est vous !

Essayez de développer le programme décrit avec ROBO Pro. A la base, ce programme fonctionne comme le compteur de visiteurs. Simulez le verrouillage électronique de l'entrée par une lampe rouge sur M1 ; celle-ci doit s'allumer lorsque 10 personnes se trouvent dans l'exposition.

## 5.3 Panneaux de commande

Après avoir résolu le problème de l'exposition, le directeur du musée a encore une nouvelle mission pour vous : Il aimerait savoir combien de visiteurs visitent son musée par jour. Un programme capable de compter n'est plus un problème pour vous, mais comment afficher une valeur ? Bien entendu, vous pourriez lancer le programme en Mode Online et montrer au directeur du musée sur quelle variable il peut contrôler la valeur. Mais pour un profane comme le directeur du musée, c'est encore fort compliqué. Il lui faut quelque chose de plus simple !

Pour de telles situations, vous trouverez des panneaux de commande dans ROBO Pro. Un panneau de commande est une page propre sur laquelle vous pouvez placer des affichages et des boutons de commande. Chargez le programme du compteur de visiteurs et basculez vers **Panneau de commande** dans la barre de fonctions.

Fonction | Symbole | Panneau de commande | Propriétés | Description

Au départ, le panneau de commande est une surface grise vide. Vous pouvez placer sur cette surface des affichages et des éléments de commande que vous trouverez dans la fenêtre des groupes d'éléments sous **Éléments de commande**. Parmi les éléments de commande, vous trouverez des boutons, des régulateurs à coulisse et d'autres choses. Sous Affichages, vous trouverez des affichages textuels, des lampes d'avertissement et des affichages sous forme d'une aiguille rotative.

- + Éléments des programmes
- Éléments de commande
  - Affichage
  - Éléments de commande
- + Dessiner
- + Bibliothèque
- Bibliothèque personnelle
- + Programme chargé

**Attention:** Un panneau de commande fait partie d'un sous-programme. Si vous avez des sous-programmes, veillez à enregistrer le panneau de commande sous **Programme principal** et pas sous un sous-programme ! Plus tard, il sera intéressant d'enregistrer plusieurs panneaux de commande.

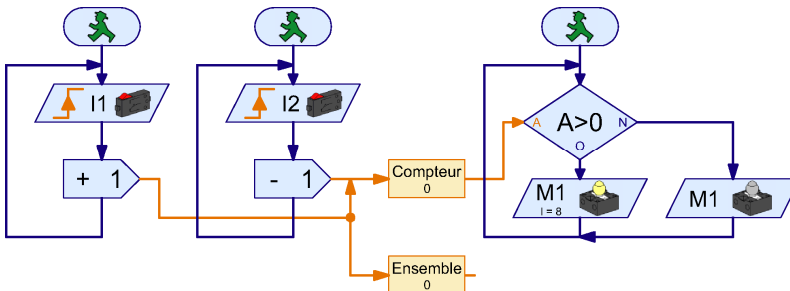
Si vous avez dessiné un panneau de commande et que celui-ci disparaît soudainement par la suite, vous avez sans doute sélectionné un sous-programme dans la barre des sous-programmes. Basculez sur **Programme principal** et le panneau de commande devrait être là.

Var: 0

Pour le compteur de visiteurs, prenez un **Affichage textuel** (peu importe la couleur) dans la fenêtre d'éléments **Éléments de commande / Affichages** et placez-le dans le panneau de commande.

Le nombre de visiteurs du musée doit maintenant être affiché dans cet affichage.

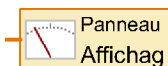
Vous devez cependant ajouter une seconde variable à votre programme ; cette variable doit compter le nombre de visiteurs à l'entrée sans supprimer les visiteurs à la sortie. Pour ce faire, basculez à nouveau sur **Fonction** et ajouter la variable **Général** comme suit :



Comme vous le voyez, un élément de commande peut également être utilisé pour envoyer une commande à deux variables en même temps. La commande - 1 n'est pas envoyée à la variable **Général**, car les commandes ne sont transmises le long des lignes orange que dans le sens des flèches. La commande +1 est quant à elle transmise aux deux variables. Il ne s'agit cependant que d'un exemple. Généralement, il est plus simple et clair d'utiliser un second élément de commande.

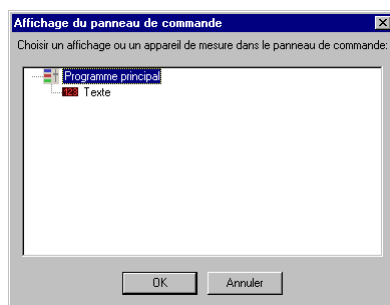


**Astuce** : Lorsque des lignes orange se branchent, il est souvent plus pratique de dessiner les lignes de l'objectif à l'origine. Si, dans l'exemple ci-dessus, vous souhaitez tracer la ligne pour la variable **Général**, cliquez d'abord sur l'entrée de la variable **Général** et reliez ensuite la ligne avec le point de branchement en retournant en arrière. Si, au contraire, vous souhaitez commencer une ligne orange sur un ligne orange existante, vous devez double-cliquer avec le bouton gauche de la souris à l'endroit où vous voulez débiter la nouvelle ligne.



Maintenant vous avez un affichage textuel dans le panneau de commande et une variable que vous aimeriez afficher dans cet affichage. Comment relier les deux ? Etant donné que l'affichage textuel et la variable se trouvent sur deux pages différentes, il serait difficile de relier les deux au moyen d'une ligne. C'est pourquoi il existe un élément spécial transmettant la valeur devant s'afficher sur le panneau de commande à l'affichage correspondant. L'élément représenté ci-dessus **Sortie du panneau de commande** se situe à la fin du groupe **Entrées, Sorties**. Ajoutez donc une sortie du panneau de commande à votre programme à côté de la variable **Général** et reliez le raccordement droit de la variable au raccordement gauche de la **sortie du panneau de commande**.

Etant donné que vous aurez généralement plus d'un affichage dans un panneau de commande, vous devez également signaler à la sortie du panneau de commande vers quel affichage il doit envoyer les valeurs de la variable. Vous pouvez le faire très simplement via la fenêtre de propriété de l'élément. Si vous cliquez avec le bouton droit de la souris sur la sortie du panneau de commande, vous voyez une liste de sélection reprenant tous les affichages déjà insérés dans le panneau de commande. Etant donné que chaque sous-programme peut avoir un panneau de commande propre, les panneaux de commande sont classés par sous-programmes. Dans notre exemple, il n'y a pas de sous-programme, seulement un programme principal. Dans celui-ci, il y a un affichage avec le nom **Texte**. Sélectionner celui-ci et cliquez sur OK.

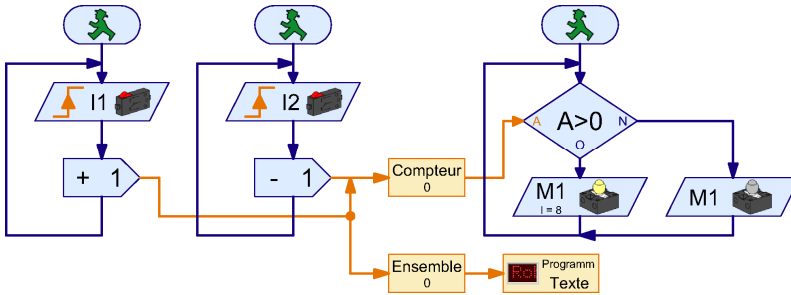


Dès que vous avez raccordé la sortie du panneau de commande avec l'affichage, le symbole change, tout comme l'inscription. La sortie du panneau de commande que nous avons utilisée établit une liaison avec l'affichage textuel nommé **Texte** dans le (sous-)programme **PRINCIPAL**.



Après avoir inséré la sortie du panneau de commande et l'avoir reliée avec l'affichage textuel ; votre programme est semblable à ceci :





Essayez ! Dès que vous aurez lancé le programme en Mode Online, l'affichage affichera dans le panneau de commande le nombre de visiteurs ayant traversé le croisillon de l'entrée.

**Nota Bene :** Si vous voulez utiliser plus d'un affichage dans un panneau de commande, il est important de donner un nom différent à chaque affichage afin de pouvoir les différencier lors de la liaison avec le programme. Pour ce faire, cliquez avec le bouton droit de la souris sur l'affichage dans le panneau de commande. Vous pouvez alors introduire un nom sous le champ **ID / Nom**. Si vous reliez ensuite une sortie du panneau de commande avec l'affichage, ce nom apparaît dans la fenêtre de sélection de la sortie du panneau de commande. Etant donné que, pour le moment, nous n'avons qu'un affichage, le nom n'a pas d'importance et nous conserverons donc le nom **Texte**.

Le programme n'est pas encore tout à fait parfait. Il manque encore un bouton-poussoir pour remettre le compteur à 0. Nous ne devons cependant pas utiliser de bouton-poussoir normal, mais bien un bouton sur le panneau de commande.

### Bouton

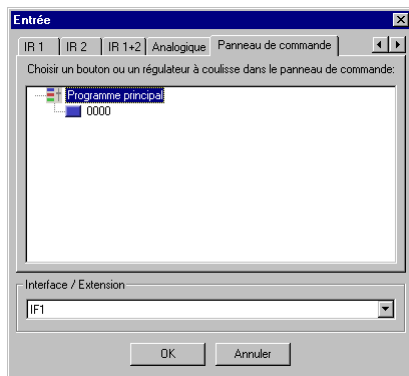
Vous trouverez le bouton de commande dans la fenêtre d'éléments sous le groupe **Éléments de commande / Éléments de commande**. Sélectionnez **Panneau de commande** dans la barre de fonction et ajoutez un bouton sur votre panneau de commande à côté de l'affichage textuel. L'inscription **Bouton** n'est évidemment pas appropriée, elle peut cependant facilement être modifiée via la fenêtre de propriétés. Pour ce faire, cliquez sur le bouton avec le bouton droit de la souris, introduisez, p.e. 0000 dans **Texte de l'inscription** et confirmez avec **OK**.



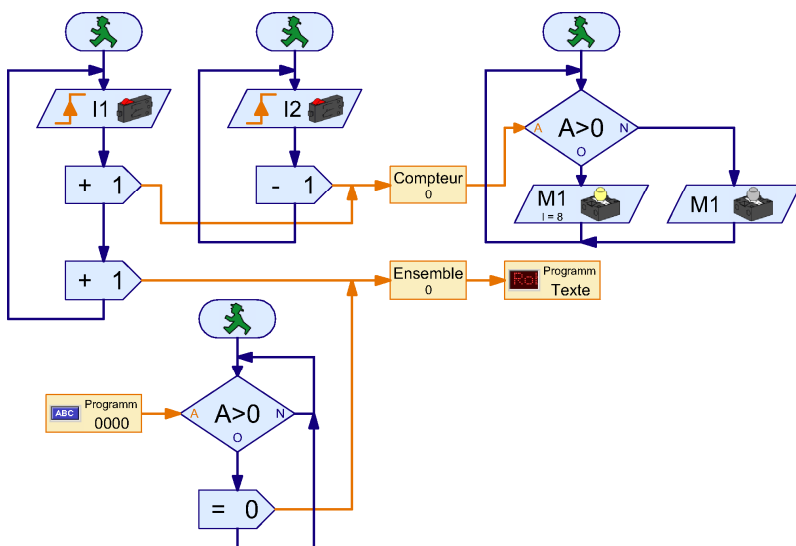
Tout comme pour l'affichage textuel, nous avons besoin d'un élément de programme reliant le bouton au déroulement du programme. Pour ce faire, sélectionnez à nouveau **Fonction** dans la barre de fonctions. Vous trouverez l'élément représenté **Entrée du panneau de commande** dans la fenêtre d'élément dans le groupe **Entrées, Sorties**. Placez cet élément dans le déroulement du programme sous le déroulement actuel.

Maintenant, vous devez encore relier l'entrée du panneau de commande avec le bouton. Pour ce faire, cliquez avec le bouton droit de la souris sur l'élément Entrée du panneau de commande. Les éléments de commande sont, comme les affichages, classés par sous-programmes, car chaque sous-programme peut avoir un panneau de commande propre. Sélectionnez le bouton **0000** et confirmez avec OK.

Vous avez peut-être remarqué qu'il est possible de régler cet élément sur tous les types d'entrées possibles via la barre d'onglets de la fenêtre de propriétés. Ceci ne sera cependant expliqué qu'au prochain chapitre *Entrée de commande pour sous-programmes*.



Vous pouvez envoyer une requête à la valeur envoyée par l'entrée du panneau de commande avec un élément de branchement. Vous avez déjà utilisé cet élément pour envoyer une requête à la variable. Le programme terminé, avec remise à zéro, est semblable à la représentation suivante :



Tant que le bouton **0000** est enfoncé, le compteur général reçoit une commande = 0 mettant le compteur à 0.

## 5.4 Timer

Le directeur du musée ne sait plus quoi faire sans vous et vous nomme son conseiller en informatique. Un tel poste apporte son lot de gloire et d'honneur, mais aussi de travail... Dans le musée, il y a de nombreuses maquettes se mettant en mouvement par une pression sur un bouton. Certains visiteurs appuient cependant relativement longtemps sur les boutons de sorte que les maquettes chauffent et qu'elles doivent souvent être réparées. Le directeur aimerait que les maquettes soient

en mouvement lorsque le bouton est enfoncé, mais seulement pour un maximum de 30 secondes. Lorsque la maquette a fonctionné une fois, une pause de 15 secondes doit être prévue avant que cette maquette ne puisse à nouveau être mise en mouvement.

Hmm, pas de problème... C'est sans doute ce que vous pensez : quelques attentes, quelques branchements de programme et c'est bon... Allons-y calmement ! Après un moment, vous constaterez que ce n'est pas si simple, et ce, pour deux raisons :

- Pendant les 30 secondes, le programme doit envoyer une requête au bouton pour savoir si celui-ci a été relâché avant la fin des 30 secondes. Ce problème peut être résolu en lançant deux procédures parallèles, voir point 5.2 *Variables et procédures multiples* sur la page 210.
- Si un visiteur relâche le bouton après 5 secondes et réappuie sur celui-ci après 15 secondes, l'attente de 30 secondes doit être redémarrée. Une attente de  $5 + 15 = 20$  secondes s'est cependant déjà écoulée et est encore active. Même avec des procédures parallèles, il est impossible de redémarrer une attente. Ce serait éventuellement possible avec deux attentes dans trois procédures que vous pourriez démarrer en alternance, mais on aurait vite des maux de tête à force de réfléchir.

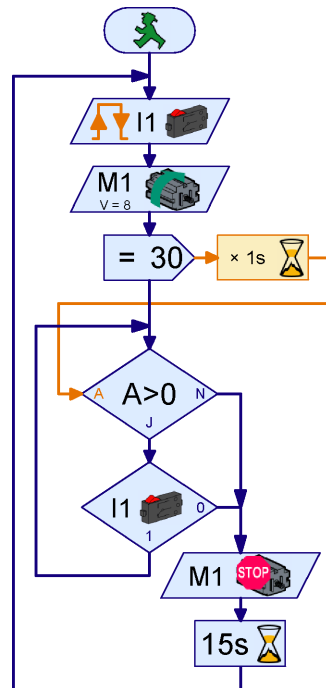


N'y a-t-il pas une solution plus simple ? Et si... en utilisant des **Variables de Timer** ou **Timer**. Un

Timer fonctionne comme une variable normale. Le Timer affiche un chiffre que vous pouvez modifier avec les commandes =, + et -. La différence avec le Timer réside dans le fait qu'il se remet à 0 régulièrement de lui-même. Vous pouvez régler la durée d'une étape de comptage par paliers, entre un millième de seconde et une minute. Avec des Timers, il est possible de résoudre des problèmes de commande du temps beaucoup plus facilement qu'avec des attentes. Voyons comment nous pouvons arranger ça avec un Timer ?

Bien... Dès que le visiteur appuie sur le bouton sur **I1**, la maquette démarre et vous mettez le Timer sur  $30 \times 1$  seconde = 30 secondes avec une commande =. Envoyez-lui alors une requête en boucle pour savoir si la période de 30 secondes est écoulée ou si le bouton sur **I1** a été relâché. Si un des deux critères d'interruption est rempli, arrêtez la maquette et attendez **15** secondes. Tout repart ensuite depuis le début.

Les programmes deviennent peu à peu de plus en plus exigeant. Mais essayez de résoudre ce problème : Développez un programme de même type avec des attentes plutôt que des Timers ! **Attention : Il s'agit d'une mission très importante qui n'a été imaginée que pour les personnes qui aiment placher sur des énigmes ! Tous les autres peuvent passer directement au prochain chapitre.** Il existe deux solutions : vous pouvez utiliser deux attentes démarrées en alternance dans des procédures propres. Etant donné qu'il y a une période de repos de 15 secondes, une des deux attentes se terminera au plus tard après le second passage de sorte qu'elle puisse à nouveau être démarrée. L'autre solution serait de reproduire un Timer avec une variable normale et un élément **Attente** avec une attente courte d'une seconde, p.e.

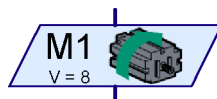


## 5.5 Entrées de commande pour sous-programme

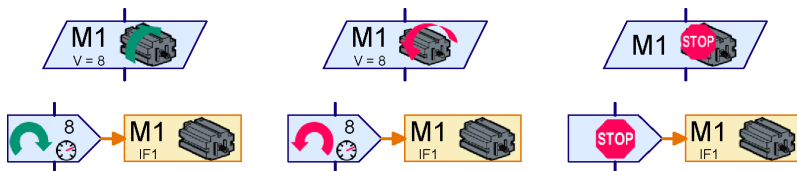
Comme toujours, votre programme fonctionne et Fischertechnik se réjouit car toutes les maquettes du musée vont être équipées avec ROBO Interface. Malheureusement, les caisses du musée sont vides, comme c'est souvent le cas dans les institutions publiques. Le directeur aimerait donc acquérir le moins de ROBO Interface possible. Une ROBO Interface a en effet quatre sorties Moteur et suffisamment d'entrées pour commander quatre maquettes. Etant donné que la plupart des maquettes ne peuvent tourner que dans un sens, vous pouvez commander jusqu'à 8 maquettes via les sorties unipolaires O1 à O8.

Le directeur du musée économise ainsi beaucoup d'argent. Pour ce faire, vous devez maintenant copier le programme 7 fois et adapter les entrées et sorties. Est-ce bien sûr ? Ne serait-il pas possible de faire ça avec des sous-programmes ?

Cela pourrait être possible, mais un problème se poserait alors : Si vous utilisez des requêtes envoyées aux interrupteurs et les éléments Moteur usuels du groupe **Éléments de fond** dans votre sous-programme, chaque appel du sous-programme envoie une requête au même interrupteur et commande le même moteur. Cela vient du fait que, dans un élément de sortie Moteur, p.e., la commande pour le moteur (droite, gauche ou stop) et le numéro de sortie Moteur (M1...M8) forment une unité. Etant donné qu'il n'y a cependant qu'un sous-programme, il s'agit toujours du même programme. Si vous modifiez le numéro de la sortie Moteur dans votre appel du sous-programme, ce numéro sera modifié dans tous les appels existants du sous-programme. Vous devriez alors copier le sous-programme 7 fois, donner un nom différent à chaque sous-programme et adapter manuellement toutes les entrées et sorties.

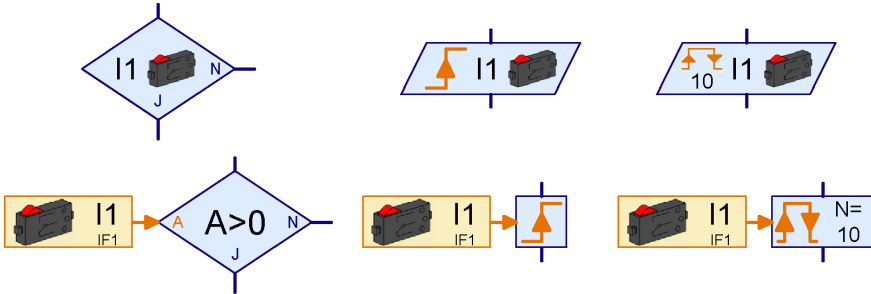


Mais il est possible de résoudre ce problème beaucoup plus facilement. Pour ce faire, il faut séparer les commandes des symboles Moteur. Il est alors possible de positionner les commande (gauche, droite, stop) dans le sous-programme et les éléments Moteur dans le programme principal. Vous envoyez alors une commande gauche, droite ou stop au programme principal via un élément de commande – avec lequel vous avez déjà fait connaissance pour les variables – dans le déroulement du sous-programme ; vous pourrez alors transmettre les commandes aux différents moteurs. Pour le moteur, il existe un élément Moteur représentant seulement un moteur sans indiquer ce que le moteur doit faire. Cet élément a une entrée de commande sur laquelle vous pouvez envoyer des commandes. Les éléments du groupe Éléments de fond peuvent être remplacés comme suit par un élément de commande et un élément Moteur :



Dans la ligne supérieure, vous voyez à chaque fois un élément Moteur du groupe **Éléments de fond**. La seconde ligne représente la combinaison d'un élément de commande du groupe **Commandes** et un élément Moteur du groupe **Entrées, Sorties**, ayant exactement le même effet. Les éléments supérieurs ne sont en fait que des raccourcis ou des simplifications des combinaisons de la ligne inférieure. Les deux envoient au moteur **M1** une commande droite, gauche ou stop.

Ceci est également valable pour les requêtes envoyées aux interrupteurs :



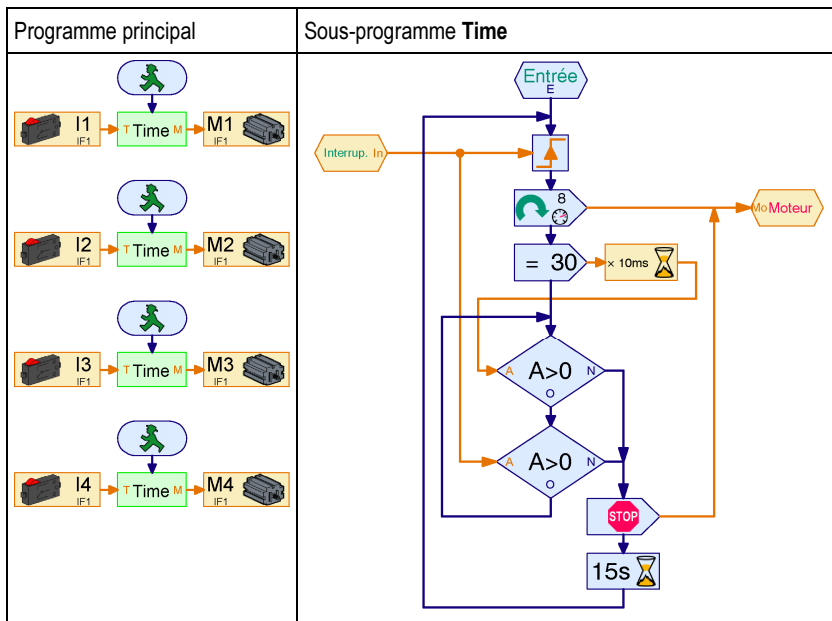
Dans la ligne supérieure, vous voyez à nouveau des éléments du groupe **Éléments de fond**. La rangée inférieure reprend une combinaison d'une entrée digitale et d'un élément du groupe **Branchement, Attente,...** Vous trouverez l'élément orange **Entrée digitale** dans le groupe **Entrées, Sorties**, comme l'élément Moteur.

Cette astuce vous permet de séparer la logique du déroulement d'un programme des entrées et des sorties. Il vous manque cependant encore quelque chose. Si les éléments Moteur et Interrupteur doivent se trouver dans le programme principal et les commandes, dans un sous-programme, les éléments Moteur et Interrupteur doivent pouvoir être reliés avec le sous-programme via un chemin. Les éléments de raccordement nécessaires se trouvent dans le groupe **Sous-programme I/O**.

**Entrée En** Vous pouvez envoyer des commandes extérieures à un sous-programme via une entrée de commande de sous-programme. L'élément Entrée digitale (Interrupteur) envoie sa nouvelle valeur via la ligne orange lorsque le statut de l'entrée change (avec une „commande =“). Vous pouvez donner un nom à l'entrée dans la fenêtre de dialogue.

**So Sortie** La sortie de commande de sous-programme vous permet d'envoyer des commandes à partir d'un sous-programme. Vous pouvez ainsi envoyer les commandes gauche, droite, stop à un moteur à partir d'un sous-programme. Vous pouvez également donner un nom à cet élément dans la fenêtre de dialogue.

Vous disposez maintenant de toutes les données pour votre **Timer pour plusieurs maquettes avec sous-programmes** :



Le **Sous-programme Time** est quasiment identique au programme du point précédent. Les éléments **Attendre une Entrée digitale I1** au début et dans la boucle sont seulement remplacés par des éléments **Attendre...** avec raccordements de données pour les lignes oranges provenant du groupe **Branchement, Attente,...** Les deux sont reliés à l'entrée de commande du sous-programme **Interrupteur**. Les deux éléments Moteur au début et à la fin du programme ont été remplacés par des éléments de commande. Les deux envoient leur commande à la sortie de commande du sous-programme **Moteur**.

Le sous-programme **Time** est appelé à quatre reprises dans le **Programme principal**. L'entrée de commande du sous-programme **Interrupteur** a généré automatiquement le raccordement orange **T** sur le côté gauche du symbole vert du sous-programme. Le raccordement **M** sur le côté droit est généré par la sortie de commande du sous-programme **Moteur**. Le raccordement **T** du symbole de sous-programme est toujours relié avec un des interrupteurs **I1** à **I4**. Un des moteurs **M1** à **M4** est alors raccordé au raccordement **M**. De cette façon, chaque appel du sous-programme **Time** envoie une requête à un autre interrupteur et commande un autre moteur !

Essayez de reproduire le sous-programme et le programme principal représentés ci-dessus et testez-les. Vous devez concevoir le sous-programme en premier lieu car celui-ci ne pourra pas être ajouté au programme principal dans le cas contraire. Si vous avez des problèmes avec le sous-programme, relisez le chapitre 4 **Niveau 2 : Travailler avec des sous-programmes** sur la page 200.

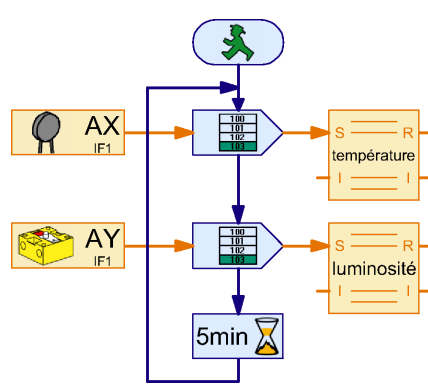
## 5.6 Listes (Arrays)

Après que toutes les maquettes du musée aient été équipées avec votre système économique de commande, le problème suivant du directeur du musée ne se fait pas attendre : Une des pièces dans laquelle se trouve d'antiques pièces de collections de très grande valeur a connu ces derniers temps des variations de température pouvant endommager les oeuvres. Vous supposez bien que ce problème est dû au rayonnement du soleil. Pour pouvoir contrôler cette relation, vous devez fabriquer un appareil enregistrant la luminosité et la température. ROBO Interface dispose de plusieurs entrées analogiques et vous savez déjà comment il est possible d'enregistrer des valeurs au moyen de variables. Tout ceci ne devrait pas poser problème... Pour enregistrer deux valeurs toutes les 5 minutes pendant 12 heures, 288 variables sont nécessaires ! Votre programme serait alors énorme et peu clair. Est-il possible de simplifier tout cela avec des sous-programmes ? Oui, mais il y a une bien meilleure façon d'y arriver : L'élément **Liste** (les programmeurs appellent ça un „Array“).

Dans une liste, il n'est pas seulement possible d'enregistrer une valeur, mais bien toute une liste de valeurs. Au début, la liste est généralement vide. Si vous envoyez une commande **Attacher** à l'entrée de données en haut à gauche avec la description S, la valeur donnée dans cet élément de commande est attachée à la fin de la liste. La longueur maximale de la liste peut être réglée entre 1 et 32767 via la fenêtre de propriétés de l'élément **Liste**. Le programme d'enregistrement de la température et de la luminosité est ainsi très simple :

Le capteur de température est raccordé à l'entrée analogique **AX** et le capteur de luminosité, à l'entrée analogique **AY**. Le programme lit les deux valeurs en boucle toutes les 5 minutes et les ajoute à une liste via la commande **Attacher**.

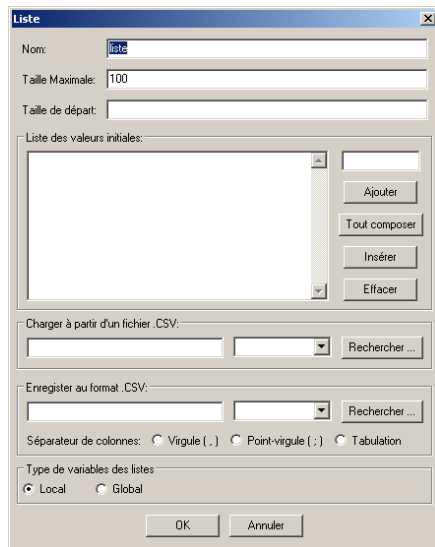
**Nota Bene:** Lors de l'ajout de l'élément de commande, vous devez activer l'option **Entrée de données pour la valeur de commande** dans la fenêtre de propriétés. Une entrée de données apparaît alors sur la gauche de l'élément de commande ; vous pouvez alors y raccorder une entrée analogique.



Pour tester ce programme, il est utile de réduire le temps de la boucle de 5 minutes à une seconde.

Vous vous demandez sans doute comment vous pourrez lire les valeurs enregistrées dans la liste. Il y a deux possibilités : vous pouvez lire les valeurs comme pour les autres variables et les exploiter dans votre programme. Etant donné que la liste comprend plusieurs éléments, sélectionnez tout d'abord le numéro de l'élément que vous voulez lire sur l'entrée de données de gauche avec la description **I**. La valeur de cet élément est alors envoyée à la sortie de données **R**, sur la droite.

ROBO Pro peut également enregistrer toutes les données de la liste dans un fichier sur votre ordinateur ; vous pourrez alors exploiter ces données, p.e., dans Excel. Etant donné que, dans le cas présent, vous ne voulez que voir et comparer les luminosités et températures, cette solution est très pratique. ROBO Pro enregistre les valeurs dans un **fichier .CSV** (comma separated values = valeurs séparées par des virgules). Les fichiers .CSV sont des fichiers textes comprenant une ou plusieurs colonnes avec à chaque fois une rangée de données. Vous pouvez également enregistrer plusieurs rangées de mesure comme la température et la luminosité dans différentes colonnes d'un fichier .CSV. Les colonnes sont séparées par des virgules. Dans les pays dans lesquels 0,5 est écrit avec une virgule et non pas avec un point (p.e. l'Allemagne), le signe de séparation pour les colonnes est souvent un **point-virgule (;)**. Si vous avez des problèmes pour échanger des fichiers .CSV entre ROBO Pro et, p.e., Microsoft Excel, vous pouvez modifier le **séparateur de colonnes** dans la fenêtre de propriétés de la liste.



Vous pouvez régler le nom du fichier .CSV et la colonne dans laquelle le contenu d'une liste doit être enregistré dans la fenêtre de propriétés de la liste sous **Enregistrer un fichier .CSV**. Les données sont enregistrées lorsque le programme s'achève en Mode Online ou lorsque vous sélectionnez le point **Enregistrer des fichiers .CSV** dans le menu **Fichier** pendant que le programme est en cours (Online ou Téléchargement). En Mode Téléchargement, vous pouvez séparer ROBO Interface de votre ordinateur pour l'inscription des données et la raccorder à nouveau pour l'enregistrement.

Après avoir lancé le programme ci-dessus en Mode Online, vous pouvez ouvrir le fichier .CSV créé par ROBO Pro avec les données dans Microsoft Excel ou dans un autre programme de tableur. Si vous ne possédez pas de programme de tableur, vous pouvez également utiliser le Windows Editor (Notepad.exe) que vous trouverez le plus souvent dans le menu de démarrage de Windows sous Accessoires.

Tant que le programme est lancé en Mode Online, vous pouvez également voir les données dans une liste en cliquant avec le bouton droit de la souris sur l'élément Liste.

## 5.7 Opérateurs

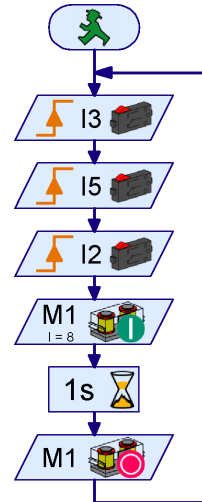
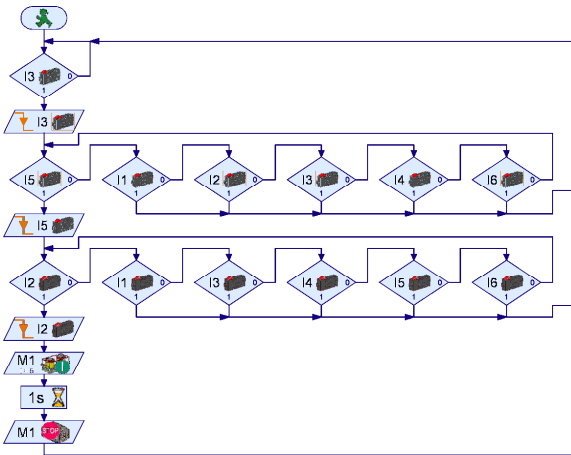
Votre programme d'inscription de la luminosité et de la température a bien fonctionné, mais vous constatez que la température dans la salle d'exposition n'a rien à voir avec le soleil. Comme vous le constatez, certains visiteurs ont confondu la commande de l'air climatisé de la salle d'exposition avec la commande d'une maquette et ont sans arrêt appuyé sur celle-ci. Ce n'est donc pas un miracle si la température de la salle d'exposition s'est détraqué !

Mais vous pouvez résoudre ce problème simplement, avec une serrure à combinaison électronique. La serrure à combinaison doit posséder un clavier avec des touches de 1 à 6. Si trois chiffres



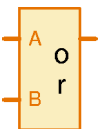
sont encodés correctement les uns à la suite des autres, la serrure à combinaison doit libérer le couvercle de la commande de l'air climatisé via un aimant.

A première vue, une serrure à combinaison de ce type est simple : le programme attend simplement que les bonnes touches sont enfoncées dans le bon ordre. Vous voyez un programme de ce type pour la combinaison 3-5-2 sur la droite. Lorsqu'on y regarde de plus près, cette solution présente cependant un problème : Il est possible de forcer cette serrure très facilement en appuyant 3 fois de suite sur toutes les touches de 1 à 6. En effet, en procédant ainsi, vous êtes sûr d'appuyer toujours sur la bonne touche. Comme le disait si bien Albert Einstein : „Pourquoi faire simple quand on peut faire compliqué“. Le programme doit non seulement savoir si les bonnes touches sont enfoncées, mais aussi si une mauvaise touche est enfoncée. Le programme ressemble à ceci :

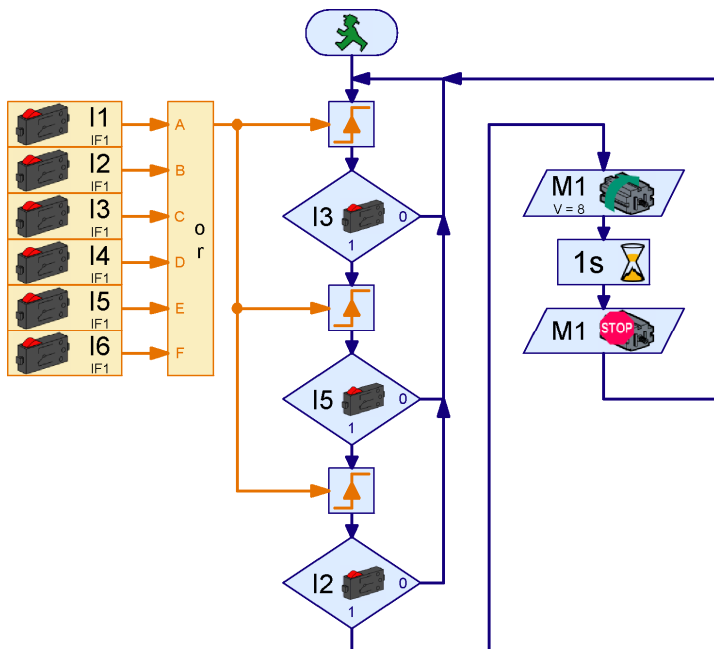


Ce programme n'ouvre la serrure que si les touches 3 5 2 sont enfoncées sans qu'une autre touche ne soit enfoncée entre-temps. Si, par exemple, la touche 3 est enfoncée, le programme attend d'abord que la touche soit relâchée. Si une autre touche que la touche 5 est

ensuite enfoncée, le programme recommence au début. Le programme fonctionne donc correctement, mais ce n'est ni simple, ni clair. Il est également très important de modifier le code. Mais pas de panique, c'est également très simple grâce aux **Opérateurs**. Il y a différents types d'opérateurs. Vous les trouverez dans le groupe **Opérateurs** sous **Éléments de programme**. Pour la serrure à combinaison, nous avons tout d'abord besoin d'un **Opérateur Ou**.



On peut raccorder plusieurs signaux aux entrées de l'**Opérateur Ou** (anglais **or**). L'opérateur donne toujours 1 à la sortie lorsqu'au moins une des entrées est 1 (ou est plus grande que 0). Lorsque l'on raccorde plusieurs boutons aux entrées **Opérateur Ou**, la sortie de l'opérateur est toujours 1 lorsqu'au moins un des boutons est actionné. Le nombre d'entrées peut être réglé jusqu'à 26 dans la fenêtre de propriétés de l'opérateur. On peut donc raccorder les 6 touches à un opérateur. Vous vous demandez sans doute comment on peut simplifier la serrure à combinaison ? C'est très simple... Avec l'opérateur, vous pouvez à chaque étape attendre qu'une touche soit enfoncée. Vous pouvez ensuite vérifier si c'est la bonne touche. Vous n'avez donc plus besoin de 7 éléments de programme par chiffre, mais bien de 2.



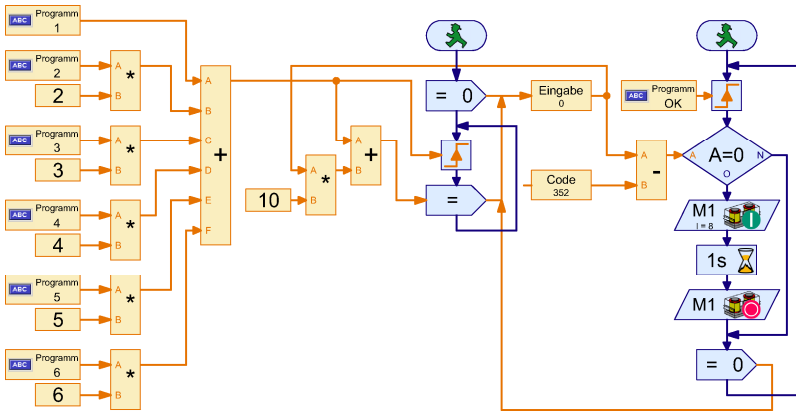
Les boutons sur les entrées I1 à I6 sont rassemblés par un Opérateur Ou à 6 entrées. Si au moins un des boutons est actionné, l'opérateur Ou envoie une valeur initiale 1, ou 0. Avec un élément **Attendre...**, le programme attend qu'un bouton soit enfoncé. On teste ensuite s'il s'agit du bon bouton. Si oui, on attend à nouveau qu'une touche soit enfoncée. Si une mauvaise touche a été enfoncée, le programme reprend au début.



Modifiez le programme ci-dessus de sorte qu'il utilise des éléments de commande du panneau de commande plutôt que des boutons. Pour ce faire, tracez tout d'abord un panneau de commande avec 6 boutons portant les inscriptions 1 à 6. Modifiez ensuite les entrées digitales dans la fenêtre de propriétés. Vous devez remplacer les branchements avec des entrées de données par des entrées du panneau de commande.

La serrure à combinaison fonctionne maintenant parfaitement, mais ce n'est toujours pas très facile de modifier le code (3 5 2). Pour ce faire, il faut changer les entrées en trois éléments de branchement. Pour la commande de l'air climatisé du musée, il n'est pas nécessaire de modifier le code régulièrement ; cependant, si vous utilisez la serrure à code pour une alarme, il est préférable de modifier le code régulièrement. Ce serait évidemment plus simple si l'on pouvait enregistrer le code dans une variable. On pourrait alors modifier le code automatiquement. Si, par exemple, une alarme silencieuse se déclenche dans l'installation, il serait possible de remplacer le code normal par un code d'alarme défini.

Afin de pouvoir comparer le chiffre introduit avec la variable du code, les chiffres introduits doivent être enregistrés dans une variable. Au départ, la variable avec la valeur initiale doit avoir une valeur 0. Si vous appuyez sur la touche 3, le variable doit avoir la valeur 3. Après avoir appuyé sur la touche 5, la valeur 35 et, finalement, après avoir appuyé sur la touche 2, la valeur 352.



La serrure à combinaison avec variable de code suit deux procédures. Dans la procédure de gauche, un nombre est attribué à chaque touche avec quelques Opérateurs Multiplication et un Opérateur Plus. La touche 1, le nombre 1 ; la touche 2, le nombre 2 ; et ainsi de suite. Les touches envoient une valeur 1 ou 0 et, lorsque l'on multiplie cette valeur par un nombre fixe X, on obtient une valeur 0 ou X. Etant donné que les valeurs sont 0 lorsque les touches ne sont pas enfoncées, il est possible d'additionner toutes les valeurs et d'obtenir ainsi un nombre comme valeur de touche. Dès qu'une touche est enfoncée, la variable initiale est attribuée 10 fois à la valeur précédente additionnée à la valeur de la touche enfoncée. La multiplication par 10 déplace la valeur précédente de la valeur initiale d'une dizaine (35 donne, par exemple, 350).



La procédure de droite attend que la touche OK du panneau de commande soit enfoncée après l'introduction du code. La variable de code Code ayant la valeur 352 lorsque le code est bien introduit est alors comparée avec la variable initiale. Si les deux ont une valeur semblable, l'aimant d'ouverture se détache, sinon non. La variable initiale est finalement remise à 0. Les variables Initial et Code sont comparées en comparant leur différence par rapport à 0 Vous auriez également pu utiliser un élément de comparaison.

Si vous enfoncez simultanément deux touches, les valeurs de ces deux touches s'additionnent. Par exemple, si vous enfoncez les touches 3 et 6 simultanément, on obtient la valeur 9. Vous pouvez ainsi fabriquer une serrure ultrasécurisée pour laquelle il faut parfois appuyer sur plusieurs touches simultanément. Imaginez sur quelles touches et dans quel ordre vous devez appuyer pour ouvrir la serrure avec un code de 495. Veillez à ce que l'élément **Attendre...** continue le programme lorsque la valeur s'élève, pas seulement lorsqu'elle passe de 0 à 1.



La serrure à combinaison fonctionne-t-elle aussi pour des codes à deux ou à quatre chiffres ? Si oui, jusqu'à combien de chiffres fonctionne-t-elle et pourquoi ? Comment cela se passe-t-il avec les autres programmes de serrure à combinaison ?



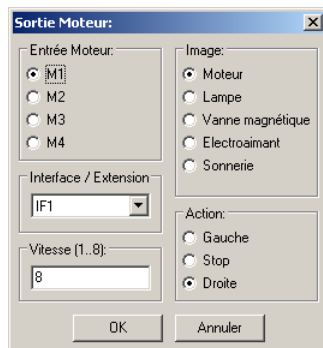
## 6 Commander des modules d'extension et des Interfaces multiples

Avec ROBO Interface ou Intelligent Interface, vous pouvez commander des maquettes relativement compliquées. Cependant, pour certains d'entre vous, il faut aller encore plus loin. Si vous n'êtes pas satisfait avec 8 interrupteurs, 4 moteurs et 4 entrées analogiques, vous pouvez étendre votre ROBO Interface grâce à 1, 2 ou même 3 **ROBO I/O-Extensions** ou étendre votre Intelligent Interface avec un **Module d'extension**. Et ceux à qui cela ne suffit pas peuvent commander plusieurs Interfaces (chacune, s'il le faut, raccordée à des modules d'extension) au départ d'un programme en mode Online. En mode Online, le nombre d'entrées et de sorties n'est plus limité que pas les capacités de votre ordinateur.

### 6.1 Modules d'extension

Peut-être avez-vous déjà remarqué dans les fenêtres de propriétés des éléments d'entrée et de sortie la liste de sélection du champ **Interface / Extension**. Vous pouvez y choisir sur quelle Interface ou sur quel Module d'Extension se trouve l'entrée ou la sortie utilisée. Si vous n'avez pas procédé à un autre réglage (voir point suivant), la liste comprend les entrées suivantes :

- **IF1:** Il s'agit de l'Interface en elle-même (donc pas de module d'extension)
- **EM1..EM3:** Il s'agit des modules d'extension 1 à 3. Si vous utilisez une Intelligent Interface, vous pouvez seulement utiliser un module d'extension (**EM1**). Les modules d'extension **EM2** et **EM3** sont cependant également affichés lorsque vous travaillez avec une Intelligent Interface, car les programmes ne doivent pas différer selon que vous utilisiez ROBO Interface ou Intelligent Interface.



Il est donc très facile d'utiliser des modules d'extension comme **ROBO I/O Extensions**. Pour ce faire, vous devez simplement sélectionner le module souhaité (Interface ou Extension) pour les entrées et sorties.

### 6.2 Interfaces multiples

Si vous désirez commander plusieurs Interfaces avec un programme, vous ne pouvez le faire qu'en Mode Online. Vous pouvez, p.e., raccorder une ROBO Interface à COM1, et une seconde à USB. Les deux peuvent être équipées avec 1, 2 ou même 3 ROBO I/O-Extensions. Une autre possibilité de combinaison serait, p.e., 2 ROBO Interfaces sur USB (toutes les deux avec un module I/O-Extensions) et une Intelligent Interface sur COM1 (également avec un module d'extension). Afin de pouvoir définir quelle Interface doit être utilisée dans la fenêtre de propriétés des entrées et sorties, vous devez adapter l'attribution de l'Interface.

Tant que vous ne modifiez pas les réglages, vous trouverez les entrées **IF1**, **EM1**, **EM2** et **EM3** dans les listes de sélection des entrées et sorties **Interface / Extension**. Vous pouvez également modifier ou compléter la liste, et ce, pour plusieurs raisons :

- Pour avoir un meilleur aperçu, vous souhaitez donner un nom autre que IF1 ou EM1 aux modules ; ce nom reprend la partie de votre machine ou de votre robot commandée par le module.
- Vous aimeriez permuter deux modules d'extension (par exemple, EM1 et EM2) car ce serait plus facile pour raccorder le câble, mais ne voulez pas modifier votre programme.
- Vous aimeriez lancer un programme conçu pour ROBO Interface avec plus d'un module d'extension avec plusieurs Intelligent Interfaces.
- Vous aimeriez utiliser plus d'une ROBO Interface ou plus d'une Intelligent Interface dans votre programme.

Vous pouvez le faire très facilement en modifiant l'**Attribution de l'Interface** dans la fenêtre de propriétés du **Programme principal** :

Programme principal

Fonction | Symbole | Panneau de commande | Propriétés | Description

Nom: Programme principal

Groupe:

Création de symbole:
   
 Automatique
   
 Manuel

Disposition Standard:
   
 Dynamique
   
 Statique

Procédure du nombre minimum: 5

Procédures supplémentaires: 0

Mémoire minimum par procédure (Téléchargement): 4096

Mémoire minimum par procédure (Online): 85536

Attribution de l'Interface

IF1 = Interface principal  
EM1 = Extension I/O 1  
EM2 = Extension I/O 2  
EM3 = Extension I/O 3

Nouveau Traiter Effacer

Vous voyez ici quel module (Interface ou Extension) a reçu les noms **IF1** à **EM3**. Vous pouvez ajouter une nouvelle Interface avec le bouton **Nouveau**. Si vous voulez modifier une entrée de la liste, sélectionnez-la et cliquez sur **Traiter**. Dans les deux cas, la fenêtre suivante apparaît :

- Dans le champ **Nom**, vous pouvez modifier le nom du module. Le nom ne doit pas être trop long car la place du nom de l'Interface est très petite dans les symboles graphiques. Si vous modifiez le nom, vous devez également modifier le nom du module dans tous les éléments d'entrée et de sortie utilisant ce nom.
- Dans le champ **Extension**, vous pouvez définir si le nom se rapporte à une Interface ou à un des modules d'extension 1 à 3.
- Dans le champ **Interface**, vous pouvez sélectionner à quelle interface l'Interface est raccordée. Si vous introduisez **Sélection de l'utilisateur**, l'interface que vous avez sélectionnée sous **COM/USB** dans la barre d'outils est utilisée. C'est le plus simple tant que vous souhaitez utiliser une Interface avec plusieurs modules d'extension, mais pas plusieurs Interfaces, car ainsi une autre personne peut utiliser votre programme sans effectuer de modifications.

Interface

Nom: EM1

Extension: Extension I/O 1

Interface
   
 COM1
   
 COM2
   
 COM3
   
 COM4
   
 Choix de l'utilisateur

Interface
   
 ROBO Interface (Nouveau)
   
 Interface Intelligente
   
 Automatique

Liste des interfaces USB:

Indiquer un Interface:
   
 Au sujet du numéro de série
   
 Au sujet du rang

OK Annuler



Si vous utilisez plusieurs Interfaces, définissez ici l'interface à laquelle chaque Interface doit être raccordée.

- Dans le champ **Interface**, vous pouvez choisir si vous souhaitez utiliser ROBO Interface ou Intelligent Interface. Si l'Interface est reliée à une interface sérielle, le programme peut automatiquement reconnaître de quelle Interface il s'agit (Sélection **Automatique**).
- La partie droite de la fenêtre est seulement importante si vous avez raccordé plusieurs ROBO Interfaces à une interface USB. Si vous cliquez sur **USB** sous **Interface**, vous pouvez sélectionner une des Interfaces sous **Liste des Interfaces USB**.

**Attention** : Si vous désirez utiliser plusieurs Interfaces sur le bus USB, vous devez d'abord attribuer un numéro de série à chaque Interface. Normalement, toutes les Interfaces sont fournies avec le même numéro de série pour éviter tout problème en cas d'échange d'Interfaces. Le système d'exploitation Windows ne reconnaît cependant que des Interfaces avec des numéros de série différents. Vous en apprendrez plus sur ce sujet dans le point 6.5 *Modifier le numéro de série de l'Interface ou la version du microprogramme* sur la page 227.

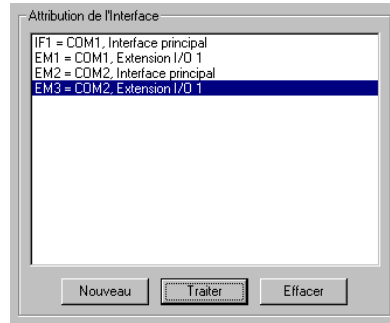
- Dans le champ **Indiquer une Interface**, vous pouvez choisir comment le programme doit indiquer l'Interface choisie. Il y a deux possibilités : Si vous sélectionnez **Numéro de série**, le programme enregistre le numéro de série de l'Interface. Même si vous raccordez et enlevez d'autres Interfaces au bus USB, le programme peut retrouver l'Interface choisie à tout moment au moyen du numéro de série. Cependant, il existe un désavantage : le programme ne fonctionnera plus qu'avec une Interface ayant un numéro de série identique. Si vous désirez utiliser une Interface avec un autre numéro de série, vous devrez modifier l'attribution de l'Interface ou le numéro de série. Pour éviter les problèmes de numéro de série, il y a une autre possibilité : **Rang**. Si vous sélectionnez ce point, le programme enregistre le rang dans la liste plutôt que le numéro de série. Ceci peut provoquer un certain désordre lorsque vous ajoutez ou supprimez une Interface sur le bus USB ; le programme fonctionne cependant de la même façon avec l'une ou l'autre Interface.

### 6.3 Attribution des Interfaces dans les sous-programmes

Normalement, toutes les attributions des Interfaces de votre programme se font dans la fenêtre de propriétés du Programme principal. Vous pouvez cependant également introduire des attributions d'Interfaces dans des sous-programmes. Dans le sous-programme, vous pouvez alors utiliser les attributions d'Interface du sous-programme et du programme principal. Lorsque deux attributions ont le même nom, l'attribution du sous-programme est prépondérante. Vous pouvez ainsi définir si vous voulez que IF1 accède à l'Interface principale dans le programme principal et que, dans un sous-programme particulier, IF1 se rapporte à un module d'extension. C'est très pratique lorsque vous souhaitez commander tout un parc de machines dans lequel chaque machine est commandée par une Interface propre. Vous pouvez ainsi développer les commandes des différentes machines dans des programmes indépendants ; chaque programme principal accède alors à IF1. Vous pouvez ensuite regrouper tous les programmes principaux des machines en tant que sous-programmes dans un programme global. Dans le programme global, vous devrez alors seulement modifier les attributions des Interfaces, mais pas les noms de chaque entrée et sortie.

## 6.4 Trucs & Astuces

Si vous désirez lancer un programme – conçu pour ROBO Interface avec 3 modules d'extension – sur 2 Intelligent Interfaces avec à chaque fois un module d'extension, vous pouvez utiliser l'attribution d'Interface représentée ici. Les modules d'extension 2 et 3 sont alors remplacés par une autre Intelligent Interface avec un module d'extension sur COM2.

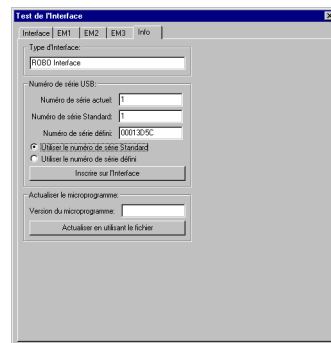


## 6.5 Modifier le numéro de série de l'Interface ou la version du microprogramme

Généralement, toutes les ROBO Interfaces et ROBO I/O-Extensions sont livrées avec le même numéro de série. Tant que vous n'utilisez qu'une Interface sur votre ordinateur, cette solution est très pratique, car, ainsi, toutes les Interfaces sont semblables pour l'ordinateur et il n'y a pas de problème lorsque vous changez d'Interface. Cependant, si vous souhaitez utiliser plus d'une Interface sur un ordinateur via une interface USB, vous devrez tout d'abord modifier le numéro de série de l'Interface, afin que l'ordinateur différencie les Interfaces et puissent les appeler. Si vous envoyez une requête à l'Interface via plusieurs interfaces sérielles, ceci n'est pas nécessaire.

Pour modifier le numéro de série, suivez les étapes suivantes :

- Raccordez l'Interface **individuelle** au bus USB de l'ordinateur.
- Appuyez sur le bouton **COM/USB** de la barre d'outils et sélectionnez l'interface USB.
- Ouvrez la fenêtre du test de l'Interface via le bouton **Test** de la barre d'outils et basculez vers l'onglet **Info** :
- Dans le champ **Type d'Interface** s'affiche le type d'Interface, par exemple **ROBO Interface** ou **ROBO I/O Extension**.
- Dans le champ **Numéro de Série USB**, vous pouvez introduire le numéro de série que l'Interface utilisera au démarrage. Chaque Interface contient deux numéros de série incorporés, un **Numéro de série standard** – 1 tant que vous n'introduisez pas un autre numéro – et un **Numéro de série défini** que vous ne pouvez pas modifier et qui est différent pour chaque Interface. La façon la plus simple d'utiliser plusieurs Interfaces sur le bus USB est de régler le bouton de sélection de chaque Interface sur **Utiliser le Numéro de série défini**. Chaque Interface aura donc avec certitude un numéro de série propre, interchangeable. Si vous utilisez beaucoup d'Interfaces pour une maquette, il peut être peu pratique d'indiquer la totalité des numéros de série. Dans ce cas, il est plus simple de régler le numéro de série standard de vos Interfaces, p.e., sur 1, 2, 3,... et d'utiliser ce numéro. Après avoir modifié ou sélectionné le numéro de série, vous devez encore appuyer sur le bouton **Inscrire sur l'Interface**. Après



avoir modifié le numéro de série, vous devez débrancher l'alimentation en courant et la rebrancher.

**Attention** : Si le numéro de série est modifié, il se peut que le pilote doive être réinstallé ; pour ce faire sous Windows NT, 2000 et XP, vous devez disposer de droits Administrateur. Si vous modifiez le numéro de série mais ne pouvez pas réinstaller le pilote car vous n'avez pas de droits Administrateur, vous ne pourrez plus avoir accès à l'Interface via l'interface USB. Dans ce cas, vous pouvez débrancher l'Interface de l'alimentation en courant et gardez le bouton **Port** enfoncé lorsque vous reconnectez l'alimentation en courant. L'Interface démarre alors avec le numéro de série 1 et est reconnue par le pilote déjà installé. Le numéro de série n'est cependant pas modifié définitivement ; en effet, au prochain démarrage, le numéro de série précédent sera utilisé si vous n'appuyez pas à nouveau sur le bouton Port. Pour modifier définitivement le numéro de série, procédez comme nous l'avons vu plus haut.

- Dans le champ **Actualiser le microprogramme**, vous pouvez enfin actualiser le programme de commande interne de votre ROBO Interface dans le cas où Fischertechnik proposerait une nouvelle version du microprogramme de l'Interface.



## 7 Aperçu des éléments du programme

Tous les éléments de programme disponibles dans ROBO Pro sont organisés dans la suite en Groupes d'éléments et décrits dans l'ordre dans lequel ils apparaissent dans la fenêtre d'éléments.

### 7.1 Eléments fondamentaux (Niveau 1)

#### 7.1.1 Démarrer



Dans un programme, une procédure commence toujours par un élément de démarrage. Si cet élément de programme ne se trouve pas au début du programme, la procédure ne peut pas être traitée. Si un programme comprend plusieurs procédures, chacune de ces procédures doit commencer par un élément de démarrage. Les différentes procédures seront alors lancées simultanément.

Les propriétés de l'élément de démarrage ne peuvent pas être modifiées. C'est pourquoi, contrairement à la plupart des autres éléments, **aucune** fenêtre de propriétés ne s'ouvre lorsque vous cliquez sur l'élément avec le bouton droit de la souris.

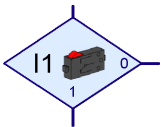
#### 7.1.2 Terminer



Pour terminer une procédure, il faut relier la sortie du dernier élément avec l'élément de fin. Une procédure peut également se terminer à plusieurs endroits avec cet élément. Il est également possible de relier les sorties de différents éléments avec un seul élément de fin. Une procédure peut cependant également avoir la forme d'une boucle sans fin et ne pas contenir d'élément de fin.

Les propriétés de l'élément de fin ne peuvent pas être modifiées. C'est pourquoi, contrairement à la plupart des autres éléments, **aucune** fenêtre de propriétés ne s'ouvre lorsque vous cliquez sur l'élément avec le bouton droit de la souris.

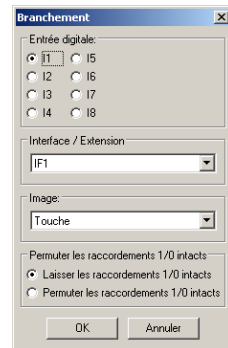
#### 7.1.3 Branchement digital



Ce branchement vous permet de diriger le déroulement du programme dans deux directions indépendamment du statut des entrées digitales **I1** à **I8**. Si, par exemple, un interrupteur est fermé sur l'entrée digitale (=1), le programme se branche sur la sortie **1**. Si, au

contraire, l'interrupteur est ouvert (=0), le programme se branche sur la sortie **0**.

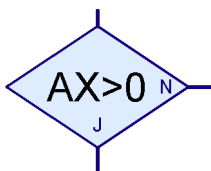
Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :



- Les boutons I1 à I8 vous permettent de choisir à quelle entrée de l'Interface doit être envoyée la requête.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une entrée de l'Interface ou une entrée du module d'extension ou d'une deuxième Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.

- Dans le champ **Image**, vous pouvez sélectionner une image pour le capteur raccordé à l'entrée. Le plus souvent, des interrupteurs sont raccordés aux entrées analogiques, mais aussi des transistors photo et des relais à contacts scellés.
- Dans le champ **Permuter les raccordements 1/0**, vous pouvez permuter les positions des sorties 1 et 0 du branchement. Normalement, la sortie 1 se situe en-dessous et la sortie 0, à droite. Il peut être pratique d'avoir la sortie 1 à droite. Appuyez sur **Permuter les raccordements 1/0**, les deux raccordements seront alors inversés dès que vous cliquerez sur OK pour fermer la fenêtre.

### 7.1.4 Branchement analogique



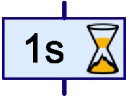
Outre les entrées analogiques, ROBO Interface possède 6 entrées analogiques, 2 entrées de résistance AX et AY, 2 entrées de tension A1 et A2 et deux entrées pour des capteurs d'espacement D1 et

D2. Avec ces branchements, vous pouvez comparer la valeur d'une entrée analogique avec un nombre fixe et, selon que la comparaison est bonne ou non, la brancher sur une sortie Oui (J) ou Non (N).

Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :

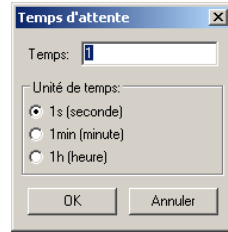
- Dans le champ **Entrée analogique**, vous pouvez choisir l'entrée à laquelle une requête sera envoyée. Toutes les entrées analogiques fournissent une valeur entre 0 et 1023. Vous trouverez de plus amples informations sur les différentes entrées analogiques au point 7.6.2 *Entrée analogique* sur la page 248.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous désirez utiliser une entrée de l'Interface ou une entrée d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Condition**, vous pouvez sélectionner un opérateur de comparaison comme plus petit (<) ou plus grand (>) et introduire la valeur de comparaison. La valeur de comparaison doit être comprise entre 0 et 1023. Si vous lancez un programme avec un branchement pour des entrées analogiques en Mode Online, la valeur analogique actuelle s'affiche.
- Dans le champ **Permuter les raccordements J/N**, vous pouvez permuter les positions des sorties 1 et 0 du branchement. Normalement, la sortie Oui (J) se trouve en bas et la sortie Non (N), à droite. Il peut être pratique d'avoir une sortie Oui à droite. Appuyez sur **Permuter les raccordements J/N** ; les raccordement J et N seront alors inversés dès que vous fermez la fenêtre en cliquant sur OK.

### 7.1.5 Attente



L'élément **Attente** vous permet de retarder le déroulement d'une procédure jusqu'à un moment donné.

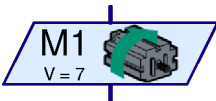
Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche. Vous pouvez alors introduire l'attente en secondes, minutes ou heures. L'intervalle de temps pour l'attente va de 1 milliseconde (un millième de seconde) à 500 heures (presque 3 semaines). Cependant, la précision de la mesure du temps varie en fonction de la longueur de l'attente.



La liste suivante donne la précision de la mesure pour différentes attentes :

Attente	Précision
Jusqu'à 30 secondes	1/1000 de seconde
Jusqu'à 5 minutes	1/100 de seconde
Jusqu'à 50 minutes	1/10 de seconde
Jusqu'à 8,3 heures	1 seconde
Jusqu'à 83 heures	10 secondes
Jusqu'à 500 heures	1 minute

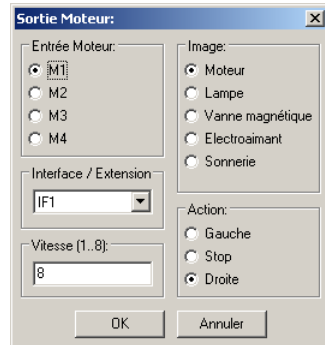
### 7.1.6 Sortie Moteur



L'élément de programme **Sortie Moteur** permet de connecter une des sorties bipolaires M1-M4 de l'Interface. Les sorties de l'Interface peuvent aussi bien être utilisées pour des moteurs que pour des lampes ou des électroaimants. Pour les moteurs, la vitesse et le sens de rotation peuvent être définis.

Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :

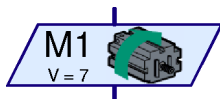
- Dans le champ **Sortie Moteur**, vous pouvez régler laquelle des quatre sorties Moteur **M1** à **M4** doit être utilisée.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une sortie de l'Interface ou une sortie d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Image**, vous pouvez sélectionner une image qui représentera l'élément Fischertechnik raccordé à la sortie.
- Dans le champ **Action**, vous pouvez régler comment la sortie doit être influencée. Vous pouvez faire tourner un moteur vers la gauche ou vers la droite, ou encore le stopper. Si vous



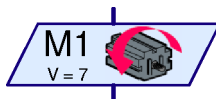
raccordez une lampe à une sortie Moteur (voir Astuce sous Sortie Lampe), vous pouvez allumer ou éteindre celle-ci.

- Vous pouvez finalement définir une **Vitesse** ou une **Intensité** entre 1 et 8. 8 est la plus grande vitesse, luminosité ou force magnétique, et 1, la plus petite. Pour l'arrêt ou l'extinction, vous ne devez évidemment pas définir de vitesse.

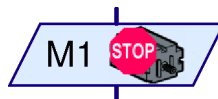
Voici les symboles de quelques actions et images :



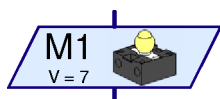
Moteur à droite



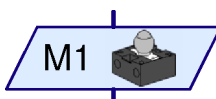
Moteur à gauche



Arrêt du Moteur



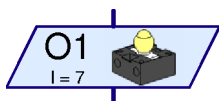
Allumer la Lampe



Eteindre la lampe

**Astuce** : Parfois, un moteur ne peut être actionné que dans un sens, p.e. une bande transporteuse. Dans ce cas, vous pouvez utiliser une Sortie Lampe pour ce moteur ; vous utiliserez alors un raccordement de moins.

### 7.1.7 Sortie Lampe (Niveau 2)

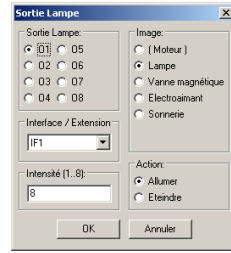


L'élément de programme **Sortie Lampe** vous permet de connecter une des sorties unipolaires O1-O8 de l'Interface. Les sorties de l'Interface peuvent soit être utilisées par paire comme Sorties Moteur (voir plus haut), soit être utilisées de manière isolée comme Sorties Lampe O1-O8. Contrairement aux Sorties Moteur, les Sorties Lampe n'ont qu'un pin de raccordement. Vous pouvez donc commander 8 lampes ou vannes magnétiques isolées. Raccordez alors l'autre raccordement de la lampe à la douille massique de l'Interface (⊥).

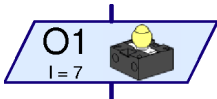
**Astuce**: Lorsque vous ne voulez raccorder que quatre lampes, vous pouvez également utiliser une sortie Moteur pour des lampes. C'est fort pratique car ainsi les deux raccordements de la lampe peuvent être directement raccordés à la Sortie de l'Interface et ne doivent pas être relié à la douille massique comme pôle négatif.

Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :

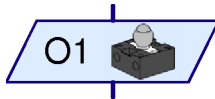
- Dans le champ **Sortie Lampe**, vous pouvez choisir laquelle des huit sorties **O1** à **O8** vous voulez utiliser.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une sortie de l'Interface ou une sortie d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Image**, vous pouvez sélectionner une image qui représentera l'élément Fischertechnik raccordé à la sortie.
- Dans le champ **Action**, vous pouvez régler comment la sortie doit être influencée. Vous pouvez allumer ou éteindre une lampe.
- Finalement, vous pouvez aussi régler l'**Intensité** entre 1 et 8. 8 est la plus grande intensité ou force magnétique, et 1, la plus petite. Lorsque vous éteignez la lampe, vous ne devez évidemment pas régler l'intensité.



Voici les symboles des différentes actions de l'image **Lampe** :

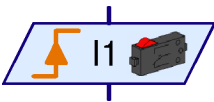


Allumer la lampe



Eteindre la lampe

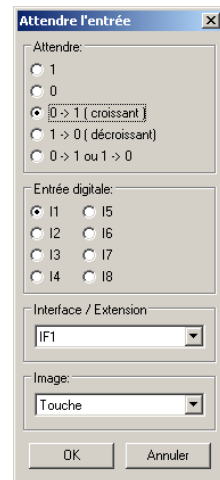
### 7.1.8 Attendre une entrée



L'élément **Attendre une entrée** attend qu'une entrée de l'Interface ait un statut particulier ou se modifie d'une certaine manière.

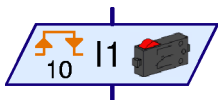
Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :

- Dans le champ **Attendre...**, vous pouvez sélectionner quel type de modification ou de statut doit être attendu. Si vous choisissez **1** ou **0**, l'élément attend que l'entrée soit fermée (1) ou ouverte (0). Si vous choisissez **0 -> 1** ou **1 -> 0**, l'élément attend que le statut de l'entrée **change** d'ouvert à fermé (0->1) ou de fermé à ouvert (1->0). Dans le dernier cas, l'élément attend que l'entrée se modifie, peu importe si elle passe du statut ouvert à fermé ou le contraire. Au point 3.6 *Autres éléments de programme* sur la page 194, vous comprendrez mieux comment vous pouvez relier cet élément à l'élément Branchement.
- Dans le champ **Entrée digitale**, vous pouvez définir à laquelle des 8 entrées digitales **I1** à **I8** doit être envoyée la requête.



- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une entrée de l'Interface ou une entrée d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Image**, vous pouvez choisir une image pour le capteur raccordé à l'entrée. Le plus souvent, on raccorde des interrupteurs aux entrées digitales, mais on peut également raccorder des transistors photo ou des relais à contacts scellés.

### 7.1.9 Compteur d'impulsion



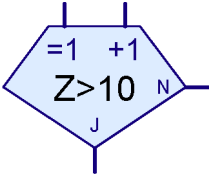
De nombreuses maquettes Fischertechnik utilisent des engrenages d'impulsion. Ces engrenages actionnent un interrupteur quatre fois par rotation. Des tels engrenages d'impulsion vous permettent d'allumer un moteur pour un certain nombre de rotations plutôt que pour une certaine durée. Pour ce faire, il faut compter le nombre d'impulsions sur une entrée de l'Interface. A cet effet, vous pouvez utiliser l'élément **Compteur d'impulsions** qui attend un nombre définissable d'impulsions.



Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :

- Dans le champ **Type d'impulsion**, vous pouvez choisir quel type d'impulsion doit être comptabilisé. Si vous choisissez **0 -> 1** (croissant), l'élément attend que le statut de l'entrée passe de ouvert à fermé (0->1) aussi souvent que vous l'avez défini dans le champ **Nombre d'impulsions**. Avec **1 -> 0** (décroissant), l'élément attend que le statut de l'entrée passe de fermé à ouvert aussi souvent que vous l'avez défini dans le champ **Nombre d'impulsions**. Avec les engrenages d'impulsions, on utilise souvent la troisième solution : Ici, l'élément compte aussi bien les modifications **0 -> 1** que les modifications **1 -> 0**, de sorte que 8 impulsions sont comptées par rotation de l'engrenage.
- Dans le champ **Entrée digitale**, vous pouvez définir à laquelle des 8 entrées digitales **11 à 18** doit être envoyée la requête.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une entrée de l'Interface ou une entrée d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Image**, vous pouvez choisir une image pour le capteur raccordé à l'entrée. Le plus souvent, on raccorde des interrupteurs aux entrées digitales, mais on peut également raccorder des transistors photo ou des relais à contacts scellés.

### 7.1.10 Boucle de comptage

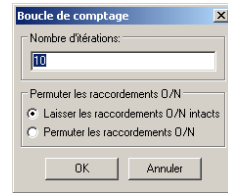


L'élément **Boucle de Comptage** vous permet de faire se dérouler une certaine partie de programme à plusieurs reprises. L'élément Boucle de Comptage contient un chiffre incorporé. Lorsque la boucle de comptage passe sur l'entrée **=1**, le compteur se met sur 1. Lorsque la boucle de comptage passe sur l'entrée **+1**, on ajoute 1 au compteur. Selon que la valeur du compteur est plus grande ou plus petite que la valeur que vous avez définie, la boucle de comptage se branche sur la sortie Oui (**J**) ou Non (**N**). Vous trouverez un exemple au point 3.6.4

*Boucle de comptage sur la page 196.*

Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche :

- Dans le champ **Nombre de passages de la boucle**, vous introduisez combien de fois la boucle de comptage doit passer sur la sortie Non (**N**) avant que la sortie Oui (**J**) ne soit activée. La valeur introduite doit être positive.
- Si vous cliquez sur **Permuter les raccordements J/N**, les raccordement **J** et **N** seront inversés dès que vous fermerez la fenêtre en cliquant sur OK. En fonction de l'endroit où se trouve le raccordement **J** et le raccordement **N**, la partie de programme répétée se trouvera à droite ou à gauche de la boucle de comptage.



## 7.2 Sous-programme I/O (Niveau 2-3)

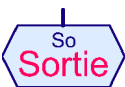
Dans ce groupe d'éléments, vous trouverez les éléments de programme que vous n'utilisez que dans des sous-programmes.

### 7.2.1 Entrée du Sous-programme (Niveau 2)



Un sous-programme peut comporter une ou plusieurs entrées du sous-programme. Via cette entrée, le programme principal ou le sous-programme d'ordre supérieur fait passer le déroulement du programme sur le sous-programme. Un pin de raccordement est ajouté sur la partie supérieure du symbole vert du sous-programme inséré dans le programme d'ordre supérieur pour chaque entrée du sous-programme. Les raccordements du symbole suivent le même ordre (de gauche à droite) que les entrées du sous-programme dans le plan de fonctionnement du sous-programme. Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche. Vous pouvez y donner un nom à l'entrée ; ce nom sera alors affiché sur le symbole. Vous pouvez en lire plus sur le sujet des sous-programmes au chapitre 4 *Au secours, je m'y perds – Travailler avec des sous-programmes* sur la page 200.

### 7.2.2 Sortie du sous-programme (Niveau 2)



Un sous-programme peut avoir une ou plusieurs sorties du sous-programme. Via cette sortie, le sous-programme fait repasser le déroulement du programme au programme principal ou au sous-programme d'ordre supérieur. Un pin de raccordement est ajouté sur la partie inférieure du symbole vert du sous-programme inséré dans le programme d'ordre supérieur pour chaque sortie du sous-programme. Les raccordements du symbole suivent le même ordre (de gauche à droite) que les sorties du sous-

programme dans le plan de fonctionnement du sous-programme. Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche. Vous pouvez y donner un nom à la sortie ; ce nom sera alors affiché sur le symbole. Vous pouvez en lire plus sur le sujet des sous-programmes au chapitre 4 *Niveau 2 : Travailler avec des sous-programmes* sur la page 200.

### 7.2.3 Entrée de commande du sous-programme (Niveau 3)



Cet élément vous permet de relier des sous-programmes avec des éléments d'entrée, comme des interrupteurs dans le programme principal ou dans un sous-programme de rang supérieur, ou de les alimenter avec des valeurs provenant d'éléments de variables, comme des coordonnées. Un pin de raccordement est ajouté sur le côté gauche du symbole vert du sous-programme inséré dans le programme d'ordre supérieur pour chaque entrée de commande du sous-programme. Les raccordements du symbole suivent le même ordre (de haut en bas) que les entrées de commande du sous-programme dans le sous-programme. Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche. Vous pouvez y donner un nom à l'entrée ; ce nom sera alors affiché sur l'élément. L'utilisation de cet élément est expliqué en détail au point 5.5 *Entrées de commande pour sous-programmes* sur la page 216.

### 7.2.4 Sortie de Commande du sous-programme (Niveau 3)

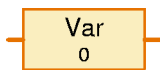


Cet élément de programme vous permet d'envoyer des commandes, p.e. gauche, droite, stop, à des moteurs ou à d'autres éléments de sortie se trouvant dans le programme principal ou dans un sous-programme de rang supérieur. Un pin de raccordement est ajouté sur le côté droit du symbole vert du sous-programme inséré dans le programme d'ordre supérieur pour chaque sortie de commande du sous-programme. Les raccordements du symbole suivent le même ordre (de haut en bas) que les sorties de commande du sous-programme dans le sous-programme. Si vous cliquez avec le bouton droit de la souris sur l'élément, la fenêtre de propriétés s'affiche. Vous pouvez y donner un nom à la sortie ; ce nom sera alors affiché sur l'élément. L'utilisation de cet élément est expliqué en détail au point 5.5 *Entrées de commande pour sous-programmes* sur la page 216.

## 7.3 Variable, Liste, ... (Niveau 3)

Les éléments de programme de ce groupe peuvent enregistrer une ou plusieurs valeurs. Il est ainsi possible de développer des programmes ayant une mémoire.

### 7.3.1 Variable (globale)



Une variable peut enregistrer une valeur chiffrée entre -32767 et 32767. La valeur de la variable est fixée en raccordant un élément de commande = à l'entrée de commande sur le côté gauche. (Voir point 7.4.1 = *(Attribuer)* sur la page 242 ). Il est également possible de donner une valeur initiale à la variable via la fenêtre de propriétés ; la variable conservera cette valeur initiale jusqu'à ce qu'elle reçoive la première commande de modification de la valeur.

ROBO Pro enregistre une seule variable pour tous les éléments de variable ayant le **même nom** et dont le **Type de Variable** est **Global**. Toutes les variables globales portant le même nom sont identiques et ont toujours la même valeur, même lorsqu'elles apparaissent dans différents sous-programmes. Si un de ces éléments de variable est modifié par une commande, tous les autres



éléments de variable portant le même nom seront modifiés. Il existe aussi des variables locales (voir chapitre suivant) pour lesquelles le système est différent.

Outre la commande =, une variable comprend également les commandes + et -. Si une variable reçoit, p.e., la commande + 5, elle ajoute la valeur 5 à la valeur de départ. Avec la commande -, la valeur transmise avec la commande est soustraite de la valeur de départ de la variable.

### Attention :

Si la valeur de la variable dépasse les limites de valeur de la variable suite à une commande + ou -, 65536 est ajouté ou retiré de la valeur de la variable, de sorte que la valeur soit à nouveau valable. Etant donné que ce comportement n'est généralement pas recommandé, il faut veiller à ce que cela n'arrive pas.

A chaque fois que la valeur de la variable change, elle envoie une commande = avec la nouvelle valeur à tous les éléments raccordés à la sortie de commande de la variable. Si vous voulez observer la valeur d'une variable, vous pouvez raccorder un affichage de panneau de commande à la sortie de la variable. (Voir point 7.6.6 *Entrée du panneau de commande* sur la page 252).

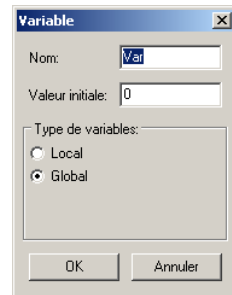
Voici un aperçu de toutes les commandes pouvant être traitées par l'élément de variable :

Commande	Valeur	Action
=	-32767 à 32767	Met la valeur de la variable à la valeur transmise avec la commande
+	-32767 à 32767	Ajoute la valeur transmise avec la commande à la valeur actuelle
-	-32767 à 32767	Soustrait la valeur transmise avec la commande de la valeur actuelle

Les limites de valeur allant de -32767 à 32767 viennent du fait que l'ordinateur compte en système binaire et non décimal comme nous. Dans le système binaire, 32767 est un nombre lisse, comme 9999 dans le système décimal. Mais nous ne devons pas nous occuper de cela car l'ordinateur transforme tous les chiffres du système binaire en chiffres du système décimal. On ne remarque donc cette particularité que dans les valeurs maximales des variables et lorsqu'il y a un dépassement de capacité lors du calcul.

### Fenêtre de propriétés des Variables

- Dans le champ **Nom**, vous pouvez donner un nom à la variable.
- Dans le champ **Valeur Initiale**, vous pouvez introduire la valeur initiale de la variable. La variable conserve cette valeur tant qu'elle ne reçoit pas de nouvelle valeur via une commande =, + ou -.
- Le point **Type de variable** n'est intéressant que pour les variables des sous-programmes et sera précisément expliqué au point suivant „Variable Locale“. Pour les variables dans les programmes principaux, les deux réglages ont le même effet.



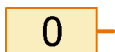
### 7.3.2 Variable locale



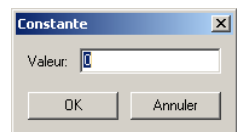
Tous les éléments de variables **globaux** portant le même nom utilise une seule et même variable et ont toujours la même valeur. C'est sans doute ce que vous attendiez et c'est bien entendu ce qui est pratique avec ce genre d'éléments. Cependant, si vous utilisez des variables dans des sous-programmes, cela peut vous poser de gros problèmes. Si votre programme comprend plusieurs procédures en parallèle, un sous-programme peut être lancé plusieurs fois sur une période. Dans une telle situation, c'est généralement le chaos lorsque le sous-programme utilise la même variable dans toutes ses procédures. C'est pour cette raison que nous avons créé les **Variables locales**. Une variable locale se comporte presque comme une variable globale, mais il y a une différence : la variable locale n'est valable que pour le sous-programme dans lequel elle est définie. Même lorsque deux variables locales dans des sous-programmes différents ont le même nom, ce sont des variables différentes, indépendantes. Même lorsqu'un sous-programme est lancé plusieurs fois en parallèle par plusieurs procédures, le sous-programme a, pour chaque procédure, une valeur indépendante pour les variables locales. Les variables locales n'existent que tant que le sous-programme dans lequel elles sont définies tourne. La valeur initiale des variables locales n'est donc pas attribuée au lancement du programme, mais bien à chaque démarrage du sous-programme en question. Etant donné qu'un sous-programme est généralement censé être identique à chaque appel, il est bien plus pratique de fixer la valeur initiale des variables à chaque appel. Les variables locales ne se „souviennent“ pas des appels précédents du même sous-programme.

Dans le programme principal, les variables locales et globales se comportent de la même façon car le programme global et le programme principal sont lancés au même moment. Les variables locales sont cependant plus efficaces dans le déroulement du programme. Les éléments de listes doivent cependant être plutôt définis de manière globale car la mémoire des variables globales est plus vaste que celle des variables locales.

#### 7.3.3 Constantes



Une constante a une valeur, comme une variable, mais cette valeur ne peut pas être modifiée par le programme. Vous pouvez relier une constante avec une entrée de données d'un symbole de sous-programme lorsque le sous-programme doit toujours utiliser la même valeur. Les constantes sont également très pratiques pour les calculs avec des opérateurs. Vous trouverez un exemple à la fin du point 5.7 *Opérateurs* à la page 220.



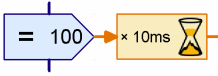
#### 7.3.4 Variable de Timer



Une Variable de Timer se comporte essentiellement comme une variable. La différence entre variables normales et statiques existe aussi chez les Variable de Timer. La seule différence vient du fait qu'une Variable de Timer remet la valeur enregistrée à 0 avec une fréquence déterminée. Dès que la valeur du Timer est 0, la Variable du Timer s'arrête. Lorsque la valeur du Timer devient négative, p.e. avec une commande négative, la valeur revient à 0 à l'étape suivante.

La vitesse à laquelle la variable de Timer décompte peut être définie entre 1/1000 de seconde par palier et 1 minutes par palier dans la fenêtre de propriétés. Vous devez garder à l'esprit que la précision du Timer dépend des paliers définis. Si, p.e., vous définissez votre Timer sur 1 x 10 secondes, la période de 10 secondes suivantes peut se débiter juste après (p.e. déjà après une

seconde) ou seulement après 10 secondes. Le Timer est donc toujours aussi précis que la période définie. Il vaut donc mieux choisir des périodes de courte durée, comme  $10 \times 1$  seconde ou  $100 \times 0,1$  seconde plutôt que  $1 \times 10$  secondes. Il ne faut choisir une période de une minute que lorsque le programme doit attendre au moins une heure. Une minute en plus ou en moins n'est alors pas importante.



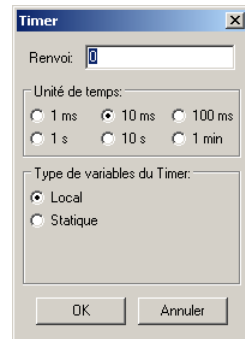
Le nombre de période devant être décomptée est généralement attribué à la variable de Timer par une commande = d'un élément de commande. L'exemple ci-contre montre le décompte de 100 périodes de 10 ms chacune. Ce qui correspond à une durée de  $1000 \text{ ms} = 1 \text{ seconde}$ . La précision est alors de 10 ms.

Grâce aux variables de Timer, il est possible de résoudre des problèmes compliqués de mesure du temps ou de renvoi. Lorsqu'un robot doit, p.e., interrompre ses recherches après 20 secondes, vous pouvez définir une variable de Timer sur  $20 \times 1$  seconde (ou  $200 \times 0,1$  seconde) au début de la recherche et demander régulièrement au programme de recherche si la valeur du Timer est toujours supérieure à 0. Vous pouvez également remettre le Timer à sa valeur initiale en cours de recherche.

Si vous souhaitez mesurer le temps, vous devez définir la variable de Timer sur la plus grande valeur positive possible au début (30000 ou 32767), afin qu'il reste beaucoup de temps à la valeur de Timer 0. Si vous voulez savoir combien de temps s'est déjà écoulé, il vous faut alors soustraire la valeur actuelle du Timer de la valeur initiale.

### Fenêtre de propriétés des Variables de Timer

- Dans le champ **Renvoi**, vous pouvez définir la valeur initiale de la Variable de Timer. Généralement, cette valeur est 0 et l'on règle la valeur de la Variable de Timer au moment voulu avec une commande =. Lorsque le Timer doit se déclencher au démarrage du programme ou d'un sous-programme, la valeur correspondante peut être introduite ici.
- Dans le champ **Unité de temps**, vous pouvez choisir la taille des périodes utilisées par la variable de Timer pour décompter.
- Dans le champ **Type de Variable de Timer**, vous pouvez décider si le Timer est une variable globale ou locale (voir point 7.3.2 *Variable Locale* sur la page 238).






### 7.3.5 Liste



L'élément **Liste** correspond à une variable dans laquelle non pas une mais plusieurs valeurs peuvent être enregistrées. Le nombre maximal de valeurs pouvant être enregistrées dans une liste est déterminé dans la fenêtre de propriétés de l'élément.

Vous pouvez attacher ou supprimer des valeurs à la fin de la liste. De plus, vous pouvez modifier ou lire une valeur quelconque de la liste ou permuter une valeur quelconque de la liste avec la première valeur de la liste. Il est impossible d'ajouter ou de supprimer directement une valeur au milieu ou au début de la liste. Il est cependant possible de concevoir un programme adapté pouvant réaliser ces fonctions.

Les fonctions suivantes d'une liste sont utilisées en envoyant des commandes à l'élément raccordé à l'entrée **S** (pour Ecrire). Les commandes suivantes peuvent être envoyées à l'entrée **S** :

Commande	Valeur	Action
Attacher 	-32767 à 32767	Attache la valeur transmise avec la commande à la fin de la liste. La liste s'agrandit d'un élément. Lorsque la liste atteint sa taille maximale, la commande est ignorée.
Supprimer 	0 à 32767	Supprime le nombre d'éléments donnés à la fin de la liste. La valeur transmise avec la commande est le nombre d'éléments à supprimer. Lorsque le nombre est plus grand que le nombre d'éléments de la liste, tous les éléments de la liste sont supprimés. Si le nombre est 0 ou est négatif, la commande est ignorée.
Permuter 	0 à 32767	Permute l'élément donné avec le premier élément de la liste. La valeur transmise avec la commande est le numéro de l'élément devant être permuté.

L'entrée **I** (pour Index) vous permet de sélectionner un élément précis de la liste. Pour ce faire, une commande = est envoyée à l'entrée **I** avec le numéro de l'élément souhaité. Le premier élément a donc le numéro 0. Il est possible d'attribuer une nouvelle valeur à l'élément sélectionné via la sortie **I** en envoyant une commande = avec la nouvelle valeur à l'entrée **S**.

La sortie **R** (pour Retour) peut envoyer une requête à l'élément sélectionné via l'entrée **I**. Si l'entrée **I** ou la valeur de l'entrée sélectionnée par l'entrée **I** est modifiée, la liste envoie la valeur actuelle de l'entrée sélectionnée à l'élément raccordé à la sortie **R**.

La sortie **I** permet de voir si l'index enregistré sur l'entrée **I** est valable. Si **N** est le nombre d'éléments, la valeur enregistrée sur l'entrée **I** doit être comprise entre 0 et **N**-1. Si c'est le cas, la sortie **I** envoie une commande = d'une valeur **N** à tous les éléments raccordés ; dans le cas contraire, la valeur est 0.

## Fenêtre de propriétés des Listes

- Dans le champ **Taille Maximale**, vous pouvez déterminer le nombre maximal d'éléments de la liste. Cette taille ne peut pas être dépassée par les commandes **Attacher**.
- Dans le champ **Taille Initiale**, vous pouvez définir le nombre d'éléments de la liste au démarrage.
- Dans le champ **Liste des Valeurs Initiales**, vous pouvez fixer les valeurs initiales de la liste. Vous pouvez retravailler la liste au moyen des boutons situés à droite de celle-ci.
- Dans le champ **Charger à partir d'un fichier .CSV**, vous pouvez sélectionner un fichier .CSV compatible avec Excel et dans lequel la liste ira chercher ses valeurs. Dans le champ de sélection du milieu, vous pouvez sélectionner la colonne du fichier .CSV devant être utilisée. Le fichier est directement chargé et affiché dans le champ **Liste des Valeurs Initiales**. Si vous lancez le programme ou réalisez un Téléchargement, ROBO Pro essaie à nouveau de charger les valeurs actuelles dans le fichier. Si cela ne fonctionne pas, les valeurs enregistrées dans le champ Liste des Valeurs Initiales sont utilisées.
- Dans le champ **Enregistrer au format .CSV**, vous pouvez entrer un fichier dans lequel le contenu de la liste devra être enregistré à la fin du programme. Ceci ne fonctionne cependant qu'en Mode Online et pour des listes statiques (voir point suivant). Le contenu de la liste est inscrit dans la colonne sélectionnée. Dans le champ **Séparateur de colonnes**, vous pouvez choisir si les colonnes de la liste doivent être séparées par des virgules ou des points-virgules. Dans les pays où 0.5 est écrit avec un point, le séparateur de colonnes sera généralement une virgule. Etant donné que, en Allemagne, 0,5 est écrit avec une virgule, on utilisera généralement un point-virgule comme séparateur de colonnes. Si vous avez des problèmes pour importer un fichier .CSV ROBO Pro dans Microsoft Excel, p.e., essayez d'utiliser un autre séparateur de colonnes.
- Dans le champ **Type des Variables de Liste**, vous pouvez définir si la liste est une variable locale ou globale (voir point 7.3.2 *Variables locales* à la page 238). Pour les grandes listes (Taille maximale supérieure à 100 éléments), le type **Global** est recommandé car l'espace mémoire est plus vaste pour les variables globales que pour les variables locales.

## 7.4 Commandes (Niveau 3)

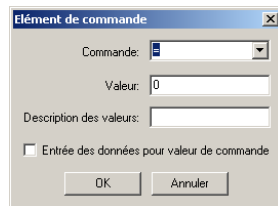
Tous les éléments de programme de ce groupe sont des éléments de commande. Selon leur utilisation, ils peuvent également être appelés Eléments de communication. Lorsqu'un élément de commande est lancé (c'est-à-dire quand le déroulement du programme passe dans l'entrée bleue de l'élément située sur le dessus de celui-ci), l'élément de commande envoie une commande ou une communication à l'élément raccordé à sa sortie gauche. Il existe plusieurs com-



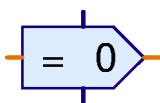
mandes comme droite, gauche, stop, ayant des effets différents sur l'élément raccordé. Généralement, les éléments raccordés ne comprennent que peu de commandes. Pour chaque élément de programme, il existe une liste reprenant les commandes qu'ils comprennent et les effets de ces commandes. La plupart des commandes sont accompagnées d'une valeur. Une commande **Droite** est par exemple accompagnée d'une vitesse entre 1 et 8. Une commande **Stop** n'aura pas de valeur.

### Fenêtre de propriétés des Eléments de commande

- Dans le champ **Commande**, vous pouvez sélectionner la commande souhaitée dans une liste reprenant toutes les commandes possibles.
- Dans le champ **Valeur**, introduisez la valeur chiffrée devant être transmise avec la commande. Ce champ reste vide si aucune valeur ne doit être transmise.
- Dans le champ **Description de la valeur**, vous pouvez introduire un petit texte indicatif qui sera affiché avec la valeur dans l'élément de commande (p.e. X= ou T=). L'indication doit expliquer de quel type de valeur il s'agit. Elle ne sert cependant que de commentaire et n'a pas de fonction.
- Dans le champ **Entrée de Données pour Valeur de Commande**, vous pouvez déterminer si l'élément de commande doit avoir une entrée de données orange sur son côté gauche pour les valeurs à transmettre. La valeur peut être introduite directement dans l'élément de commande pour tous les éléments de commande ou être lue via une entrée de données sur le côté gauche de l'élément de commande. Il est donc possible de commander un moteur avec une vitesse variable dans un circuit de réglage.



#### 7.4.1 = (Attribuer)

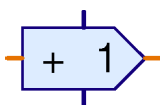


La commande = attribue une valeur au récepteur. Généralement, cette commande est utilisée pour attribuer une valeur aux variables, variables de Timer, éléments de listes ou sorties du panneau de commande.

La commande = n'est cependant pas envoyée que par des éléments de commande, mais aussi par tous les éléments de programme ayant des sorties de données. Tous les éléments envoient des commandes = lorsque la valeur d'une sortie change. Un élément d'entrée digitale envoie par exemple une commande = 1 lorsque l'interrupteur raccordé à l'entrée est fermé et une commande = 0 lorsque l'interrupteur est ouvert. Aucun élément de commande n'est cependant utilisé. Des éléments de commande = sont pour ainsi dire incorporés dans tous les éléments de programme ayant des sorties de données.

Toutes les entrées de données des éléments de programme ROBO Pro peuvent au minimum traiter la commande =. La commande = est donc la commande la plus utilisée de ROBO Pro.

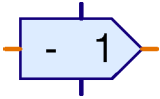
#### 7.4.2 + ( Plus )



La commande + est envoyée à des variables ou à des variables de Timer pour augmenter la valeur de la variable. La commande + permet de transmettre une valeur quelconque qui sera ajoutée à la variable. Etant donné que la valeur transmise avec la commande peut également être négative, cette commande peut également diminuer la valeur de la variable. Voir point 7.3.1

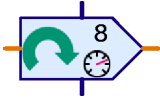
*Variable* sur la page 236 et point 7.3.4 *Variable de Timer* sur la page 238.

### 7.4.3 – ( Moins )



La commande **–** est utilisée comme la commande **+** décrite ci-dessus. La seule différence est que cette commande soustrait la valeur transmise de la valeur de la variable.

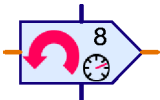
### 7.4.4 Droite



La commande **Droite** est envoyée à un élément Sortie Moteur pour faire tourner le moteur vers la droite. Voir point 7.6.4 *Sortie Moteur* sur la page 250.

La valeur est une vitesse comprise entre 1 et 8.

### 7.4.5 Gauche



La commande **Gauche** est envoyée à l'élément Sortie Moteur pour faire tourner le moteur vers la gauche. Voir point *Sortie Moteur* sur la page 250.

La valeur est une vitesse comprise entre 1 et 8.

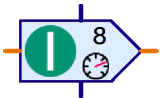
### 7.4.6 Stop



La commande **Stop** est envoyée à l'élément Sortie Moteur pour arrêter le moteur. Voir point 7.6.4 *Sortie Moteur* sur la page 250.

Aucune valeur n'est transmise avec la commande **Stop**.

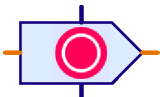
### 7.4.7 In



La commande **In** est envoyée à un élément Sortie Lampe pour allumer la lampe. Voir point 7.6.5 *Sortie Lampes* sur la page 251. Une commande **In** peut également être envoyée à des éléments Sortie Moteur, elle correspond à la commande **Droite**. Pour les moteurs, il vaut mieux utiliser la commande **Droite** car le sens de rotation est ainsi directement reconnaissable.

La valeur de la luminosité ou de l'intensité va de 1 à 8.

### 7.4.8 Out



La commande **Out** est envoyée à un élément Sortie Lampe pour éteindre la lampe. Voir point 7.6.5 *Sortie Lampes* sur la page 251. Une commande **Out** peut également être envoyée à des éléments Sortie Moteur, elle correspond à la commande **Stop**.

Aucune valeur n'est transmise avec la commande **Out**.

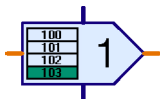
### 7.4.9 Texte



La commande **Texte** est une commande particulière car la commande envoyée à l'élément raccordé n'est pas un chiffre, mais bien un texte quelconque. Il n'y a cependant qu'un élément de programme pouvant traiter la commande Texte : l'affichage textuel d'un panneau de commande. Vous trouvez de plus amples informations au point 8.1.2

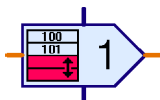
*Affichage textuel* sur la page 256.

### 7.4.10 Attacher une valeur



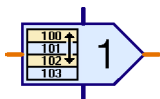
La commande **Attacher** est une commande spéciale pour les éléments Liste. Voir point 7.3.5 *Liste* sur la page 239. Cette commande transmet une valeur qui est ensuite attachée à la fin de la liste. Si la liste est déjà pleine, la commande est ignorée.

### 7.4.11 Supprimer une (des) valeur(s)



La commande **Supprimer** est une commande spéciale pour les éléments Liste. Voir point 7.3.5 *Liste* sur la page 239. Cette commande permet d'effacer un certain nombre d'éléments à la fin d'une liste. Le nombre voulu est transmis avec la commande sous la forme d'une valeur. Si la valeur transmise est plus grande que le nombre d'élément de la liste, tous les éléments de la liste seront supprimés. Pour effacer totalement une liste, il est possible d'envoyer une commande **Supprimer** avec la valeur maximale possible de 32767 à un élément de liste.

### 7.4.12 Permuter des valeurs



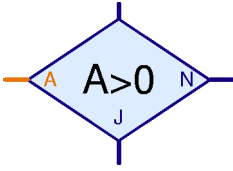
La commande **Permuter** est une commande spéciale pour les éléments Liste. Voir point 7.3.5 *Liste* sur la page 239. Cette commande permet de permuter un élément quelconque de la liste avec le premier élément de cette liste. Le numéro de l'élément avec lequel le premier élément doit être permuté est transmis avec la commande sous la forme d'une valeur. **Attention** : le premier élément d'une liste porte le numéro 0. Lorsque la valeur transmise n'est pas un numéro d'élément valable, la commande est ignorée par l'élément Liste.

## 7.5 Comparaison, Attente, ... (Niveau 3)

Les éléments de programme de ce groupe servent tous au branchement du programme et au renvoi du déroulement du programme.

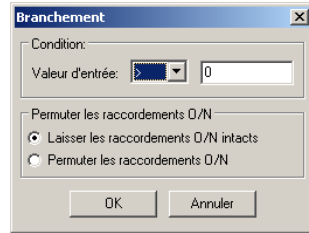


## 7.5.1 Branchement (avec entrée de données)



Ce branchement de programme possède une entrée de données **A** sur la gauche de l'élément. Celle-ci lit une valeur provenant le plus souvent d'un élément d'entrée (voir point 7.6.1 à

7.6.6 à partir de la page 247). L'entrée de données **A** peut également être reliée à des sorties de données de variables, variables de Timer ou opérateurs (voir point 7.7 *Opérateurs* sur la page 253). La valeur sur l'entrée de données **A** est comparée par l'élément avec une valeur fixe, pouvant être définie librement. Selon que la comparaison est bonne ou non, l'élément se branche sur la sortie **J** ou **N**.

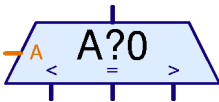


### Fenêtre de propriétés des Branchements

- Dans le champ **Condition**, introduisez dans le champ sur la gauche la valeur avec laquelle la valeur de l'entrée **A** doit être comparée. Toutes les opérations de comparaison habituelles sont disponibles pour la comparaison.
- Si vous sélectionnez **Permuter les raccordements J/N**, les sorties **J** et **N** seront inversées dès que vous fermerez la fenêtre de propriétés en appuyant sur **OK**. Pour remettre les raccordements **J/N** à leur position initiale, vous pouvez les permuter à nouveau.

La comparaison la plus utilisée est **A > 0**. Cela signifie que le déroulement du programme se branche sur la sortie **J** lorsque la valeur donnée sur l'entrée de données **A** est plus grande que 0. Vous pouvez ainsi exploiter les entrées digitales fournissant 1 ou 0. Il peut également être utile d'exploiter les variables de Timer et de nombreuses autres valeurs avec la comparaison **A > 0**.

## 7.5.2 Comparaison avec une valeur fixe

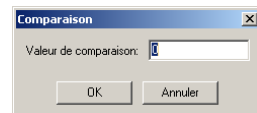


L'élément de programme **Comparaison avec une valeur fixe** vous permet de comparer la valeur de l'entrée de données **A** avec une valeur fixe pouvant être définie librement. Selon que la valeur donnée sur l'entrée de données **A** est plus grande, plus petite ou identique à la valeur fixe, cet élément de comparaison se branche sur la sortie de

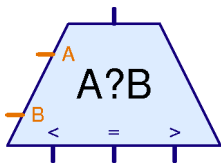
droite, de gauche ou centrale. Généralement, une variable ou une liste est raccordée à l'entrée de données **A** de la sortie. L'élément de comparaison peut être remplacé par deux éléments de branchement. Il est cependant plus aisé dans la plupart des cas de n'utiliser qu'un élément.

### Fenêtre de propriétés des Comparaisons

- Dans le champ **Valeur de comparaison**, vous pouvez introduire la valeur constante avec laquelle la valeur de l'entrée **A** doit être comparée.



### 7.5.3 Comparaison

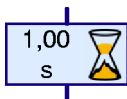


L'élément de programme **Comparaison** permet de comparer les valeurs des deux entrées de données A et B. Selon que **A** est plus petit, plus grand ou identique à **B**, l'élément se branche sur la sortie de gauche, de droite ou centrale. L'utilisation la plus courante de cet élément est la comparaison d'une valeur théorique avec une valeur réelle. Selon la position de la valeur théorique par rapport à la valeur réelle, le moteur peut, p.e., tourner vers la gauche ou vers la droite et

s'arrêter.

L'élément de programme **Comparaison** ne peut pas être réglé et n'a donc pas de fenêtre de propriétés.

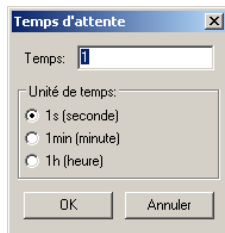
### 7.5.4 Attente



Cet élément permet de programmer une **Attente** dans le déroulement du programme. L'attente commence lorsque le tour de cet élément arrive dans le déroulement du programme. Dès que l'attente donnée est terminée, le déroulement continue. Voir également le point 3.6.1 *Attente* sur la page 22.

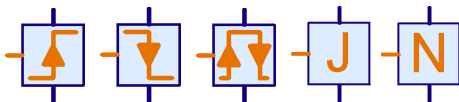
#### Fenêtre de propriétés des Attentes :

- Dans le champ **Temps**, vous pouvez introduire l'attente. Vous pouvez également utiliser des chiffres avec des virgules comme 1, 2, 3.
- Dans le champ **Unité de temps**, vous pouvez sélectionner une unité de temps entre seconde, minute ou heure. Contrairement aux variables de Timer, l'unité de temps n'a pas d'influence sur la précision de l'attente. Une attente de 60 secondes et une attente d'1 minute seront identiques.



En mode Experts (Niveau 5), une fenêtre de propriétés élargie apparaît ; celle-ci est semblable à la fenêtre de propriétés des variables de Timer.

### 7.5.5 Attendre...

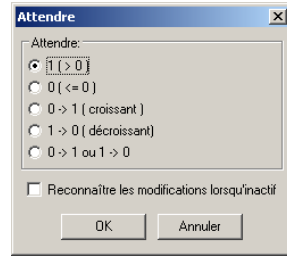


L'élément de programme **Attendre...** retarde le déroulement du programme jusqu'à ce que l'entrée de données de l'élément change ou passe à un statut déterminé. Il existe 5 versions de cet

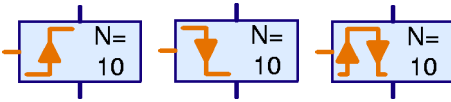
élément : L'élément de gauche attend jusqu'à ce que la valeur de l'entrée augmente. Toutes les augmentations sont alors prises en compte, comme un passage de 2 à 3, et pas seulement les modifications de 0 à 1. Le second élément attend que la valeur de l'entrée diminue et l'élément du milieu attend une modification, peu importe qu'il s'agisse d'une augmentation ou d'une diminution. Le troisième élément est souvent utilisé pour les engrenages d'impulsions. Le quatrième et le cinquième élément n'attendent pas une modification, mais bien que l'entrée ait le statut Oui (>0) ou Non (<=0). Lorsque l'on se trouve déjà en présence du statut voulu, l'élément n'attend pas. Les trois premiers éléments attendent par contre toujours une modification de l'entrée.

## Fenêtre de propriétés des Attentes de modification

- Dans le champ **Type de modification**, vous pouvez choisir entre les 5 types de fonctions décrites ci-dessus.
- Lorsque **Reconnaître les modifications lorsqu'inactif** est coché, l'élément reconnaît également les modifications qui se sont passées lorsque ce n'était pas le tour de l'élément dans le déroulement du programme. Dans ce cas, l'élément enregistre la dernière valeur connue. Lorsque l'élément est à nouveau lancé, le déroulement du programme se poursuit directement lorsque la valeur s'est modifiée de la façon voulue entre-temps. De cette façon, il est presque impossible de rater une modification parce que le programme était en train de faire autre chose.



## 7.5.6 Compteur d'impulsions

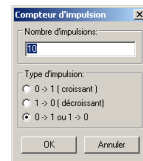


Cet élément de programme attend un certain nombre d'impulsions sur le côté gauche de l'entrée de données avant de continuer le déroulement du programme. Ce qui est très pratique pour des tâches

simples de positionnement avec des engrenages d'impulsion. Pour des positionnements plus compliqués, p.e. avec une valeur variable, des sous-programmes avec des variables doivent être utilisés.

### Fenêtre de propriétés des Compteurs d'Impulsion

- Dans le champ **Nombre d'Impulsions**, introduisez le nombre d'impulsions qu'il faut attendre avant de poursuivre le déroulement du programme.
- Dans le champ Type d'Impulsions, vous pouvez choisir entre trois types d'impulsion : **0-1**, **1-0** ou quelconque.



Pour cet élément, il est impossible de reconnaître des modifications survenues lorsque l'élément était inactif, comme c'était le cas avec l'élément **Attendre...** simple.

## 7.6 Entrées / Sorties d'Interface,...

Ce groupe d'éléments de programme contient tous les éléments d'entrées et de sorties. Vous avez appris comment les utiliser au chapitre 5 Niveau 3 : *Variables, Panneau de Commande & Cie* à la page 208.

### 7.6.1 Entrée digitale

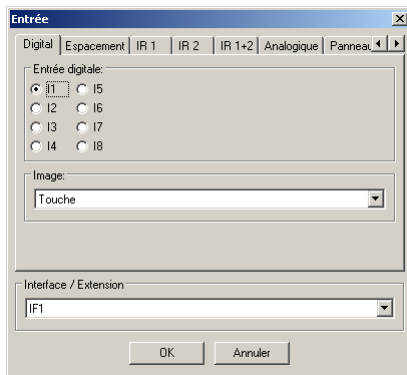


L'élément **Entrée digitale** vous permet d'envoyer une requête à la valeur d'une des entrées digitales I1 à I8 de l'Interface. Si les deux douilles appartenant à l'entrée sont reliées à l'Interface, l'élément Entrée digitale a une valeur de 1 sur son raccordement orange ; dans le cas contraire, la valeur

est 0.

## Fenêtres de propriétés des Entrées digitales

- Dans le champ **Entrée digitale**, vous pouvez sélectionner quelle entrée de l'Interface doit être utilisée. Les entrées des modules d'extension peuvent être sélectionnées dans le champ **Interface / Extension**.
- Dans le champ **Image**, vous pouvez sélectionner l'image du capteur raccordé à l'entrée. Dans la plupart des cas, il s'agira d'un **Interrupteur**. Un **relais à contacts scellés** est un interrupteur réagissant aux champs magnétiques. Il est également possible de raccorder un **transistor photo** à une entrée digitale même s'il s'agit en fait d'un capteur analogique. Raccordé à une entrée digitale, vous pouvez également utiliser le transistor photo avec une ampoule lentille comme barrière lumineuse pouvant être interrompue (=0) ou fermée (=1). Si, par contre, vous raccordez le transistor photo à une *Entrée analogique*, vous pouvez différencier de nombreux niveaux de clarté ou d'obscurité.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une entrée de l'Interface ou une entrée d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.



Plus précisément, il n'y a qu'un seul élément de programme par type d'entrées. Les onglets dans le haut de la fenêtre de propriétés vous permettent de changer de type d'entrée à tout moment. Ce qui est particulièrement pratique pour basculer entre Interrupteur, Entrées IR et Entrées de panneau de commande.

## 7.6.2 Entrée analogique



L'élément **Entrée analogique** vous permet d'envoyer une requête à la valeur d'une des entrées analogiques. Contrairement aux entrées digitales ne pouvant fournir que des valeurs 0 ou 1, les entrées analogiques peuvent faire la différence entre des niveaux très différents. Toutes les entrées analogiques fournissent une valeur initiale comprise entre 0 et 1023. Dans ROBO Interface, il y a cependant différents types d'entrées analogiques mesurant des tailles physiques différentes. Il existe des entrées analogiques pour les mesures de résistance, pour les mesures de tension et pour un capteur spécial de mesure de la distance :

Entrée	Type d'entrée	Zone de mesure
A1, A2	Entrées de tension	0-10,23 V
AX, AY	Entrées de résistance	0-5,5 kΩ
D1, D2	Entrées de capteur de distance	env. 0-50 cm
AV	Tension d'alimentation	0-10 V

Les capteurs Fischertechnik habituels (Résistance NTC, Transistor photo et résistance photo) transforme la taille de mesure (température ou intensité de la lumière) en résistance. Vous devez

donc raccorder ces capteurs aux entrées **AX** et **AY**. Les entrées de tension **A1** et **A2** ont été conçues pour tous les capteurs ayant une tension entre 0 et 10 V.

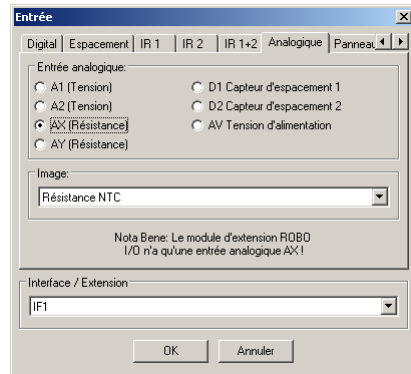
ROBO Interface n'est pas pourvu d'une douille pour l'entrée **AV**. Celle-ci est toujours raccordée à la tension d'alimentation de l'Interface. De cette façon, vous pouvez par exemple surveiller la tension de votre accumulateur et éteindre votre maquette avant que la batterie ne soit vide.

Des capteurs Fischertechnik spéciaux peuvent être raccordés aux entrées des capteurs de distance **D1** et **D2** ; ceux-ci peuvent par exemple mesurer la distance par rapport à un obstacle.

Intelligent Interface n'a que deux entrées analogiques, EX et EY. Ces entrées correspondent aux entrées AX et AY de ROBO Interface. Les autres entrées analogiques ne peuvent pas être utilisées avec Intelligent Interface !

### Fenêtre de propriétés des Entrées analogiques

- Dans le champ **Entrée analogique**, vous pouvez sélectionner l'entrée analogique souhaitée conformément au tableau ci-dessus.
- Dans le champ **Image**, vous pouvez sélectionner une image pour le capteur raccordé à l'entrée.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une entrée de l'Interface ou une entrée d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.



Dans la fenêtre de propriétés des entrées analogiques, on remarque à nouveau que ROBO Pro n'utilise qu'un seul élément pour toutes les entrées ; dans la fenêtre de propriétés de cet élément, vous pouvez passer à tous les types d'entrée grâce aux onglets. Par souci de simplification, plusieurs éléments d'entrée sont déjà disponibles dans la fenêtre d'élément.

### 7.6.3 Entrée IR

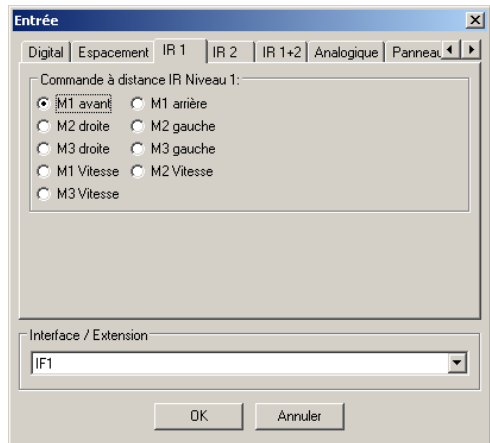


Un récepteur infrarouge est incorporé à ROBO Interface pour l'émetteur manuel du **IR Control Set** de Fischertechnik – N° d'article 30344. L'émetteur manuel infrarouge n'est pas seulement pratique pour les commandes à distance, mais aussi comme clavier pour vos maquettes. **IR**

**Control Set** dispose de deux récepteurs entre lesquels vous pouvez basculer avec les touches **1** et **2** de l'émetteur manuel. Sur ROBO Interface, vous pouvez donc occuper chaque touche de l'émetteur manuel deux fois. Vous pouvez basculer entre les deux occupations avec les touches de basculement **1** et **2**. Vous pouvez également utiliser les touches **1** et **2** comme des touches normales.

Dans la fenêtre de propriétés de l'entrée IR, vous pouvez basculer entre **IR 1**, **IR 2** et **IR 1+2** au moyen de la barre d'onglets. Si vous choisissez **IR 1**, l'élément d'entrée IR ne fournit qu'un 1 lorsque le bouton correspondant sur l'émetteur est enfoncé et que l'émetteur a été réglé au préalable sur l'occupation via la touche 1. Si vous choisissez **IR 2**, l'émetteur doit par contre être réglé sur l'occupation 2 avec la touche 2.

Par contre, si vous choisissez **IR 1+2**, peu importe comment est réglé l'émetteur manuel. Vous pouvez alors utiliser les touches 1))) et 2))) comme des entrées.



Dans l'élément de programme, la sélection est indiquée par un 1 ou un 2 blanc en bas à droite sur le symbole de l'émetteur manuel. Pour **IR 1+2**, aucun chiffre n'est affiché dans l'élément de programme.

## 7.6.4 Sortie Moteur



L'élément Sortie Moteur vous permet de commander une des 4 sorties Moteur bipolaires de ROBO Interface ou Intelligent Interface. Une sortie Moteur utilise toujours 2 raccordements de l'Interface alors qu'une sortie Lampe n'en utilise qu'un. Vous en saurez plus sur la différence entre Sortie Moteur et Sortie Lampe aux points 7.1.6 *Sortie Moteur* sur la page 231 et 7.1.7 *Sortie Lampe*.

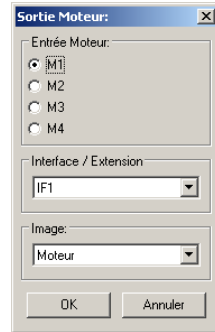
Une commande peut être envoyée à une Sortie Moteur via un élément de commande afin que la sortie soit connectée. Un élément Moteur peut traiter les commandes suivantes :

Une commande peut être envoyée à une Sortie Moteur via un élément de commande afin que la sortie soit connectée. Un élément Moteur peut traiter les commandes suivantes :

Commande	Valeur	Action
Droite	1 à 8	Le moteur tourne vers la droite avec une vitesse 1 à 8
Gauche	1 à 8	Le moteur tourne vers la gauche avec une vitesse 1 à 8
Stop	Aucune	Le moteur s'arrête
In	1 à 8	Comme Droite
Out	Aucune	Comme Stop
=	-8 à 8	Valeur -1 à -8: Le moteur tourner vers la gauche Valeur 1 à 8: Le moteur tourne vers la droite Valeur 0: Le moteur s'arrête

## Fenêtre de propriétés des Éléments Moteur

- Dans le champ **Sortie Moteur**, vous pouvez choisir quels raccordements de sorties de l'Interface doivent être utilisés. Sélectionnez les sorties des modules d'extension dans **Interface / Extension**.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une sortie de l'Interface ou une sortie d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Image**, choisissez l'image de l'utilisateur raccordé à la sortie. Dans la plupart des cas, il s'agira d'un **moteur**. Vous pouvez cependant également raccorder un **électroaimant**, une **valve magnétique** ou une **lampe** à une sortie Moteur.



## 7.6.5 Sortie Lampe



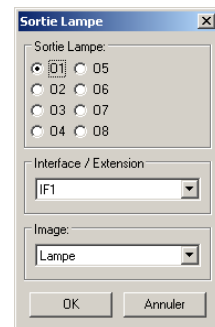
L'élément **Sortie Lampe** vous permet de commander une des 8 sorties Lampe unipolaires O1-O8 d'une ROBO Interface ou Intelligent Interface. Une sortie Lampe utilise toujours un seul raccordement de sortie de l'Interface. L'autre raccordement de l'utilisateur est relié à une douille masculine. Vous pouvez allumer ou éteindre un utilisateur raccordé de la sorte, mais pas changer sa polarité. Vous en saurez plus sur la différence entre *Sortie Moteur* et *Sortie Lampe* aux points 7.1.6 *Sortie Moteur* sur la page 231 et 7.1.7 *Sortie Lampe*.

Une commande doit être envoyée à un élément Lampe via un élément de commande pour connecter la sortie. Un élément Lampe peut traiter les commandes suivantes :

Commande	Valeur	Action
In	1 à 8	La lampe s'allume avec une intensité de 1 à 8
Out	Aucune	La lampe s'éteint
=	0 à 8	Valeur 1 à 8: La lampe s'allume Valeur 0: La lampe s'éteint

## Fenêtre de propriétés des Éléments de sortie Lampe

- Dans le champ **Sortie Lampe**, vous pouvez choisir quels raccordements de sorties de l'Interface doivent être utilisés. Sélectionnez les sorties des modules d'extension dans **Interface / Extension**.
- Dans le champ **Interface / Extension**, vous pouvez choisir si vous souhaitez utiliser une sortie de l'Interface ou une sortie d'un module d'extension ou d'une seconde Interface. Vous trouverez plus d'informations sur ce sujet au chapitre 6 *Commander des modules d'extension et des interfaces multiples* sur la page 224.
- Dans le champ **Image**, choisissez l'image de l'utilisateur raccordé à la sortie. Dans la plupart des cas, il s'agira d'un **Lampe**. Vous pouvez cependant également raccorder un **électroaimant**, une



**valve magnétique** ou un **moteur** à une sortie Lampe. Un moteur raccordé à une sortie Lampe ne peut cependant tourner que dans un sens.

## 7.6.6 Entrée du panneau de commande



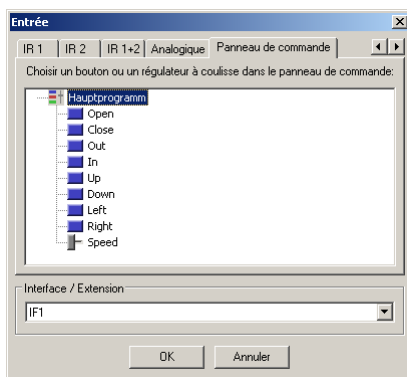
ROBO Pro vous offre la possibilité de dessiner un panneau de commande propre pour votre maquette. Vous en saurez plus sur ce sujet au chapitre 8 *Aperçu des éléments de commande et des panneaux de commande* sur la page 255. De cette manière, vous pouvez commander vos maquettes confortablement installé devant votre ordinateur. Des boutons, des régulateurs à coulisse et des éléments de saisie sont à votre disposition dans le panneau de commande. Une requête peut être envoyé à ces éléments peut connaître leur statut dans le programme via l'élément **Entrée du Panneau de Commande**. Les boutons fournissent une valeur de 0 ou 1. Les régulateurs à coulisse fournissent une valeur dans un champ définissable (normalement jusqu'à 100).

Les panneaux de commande ne peuvent être utilisés qu'en Mode Online. Vous en saurez plus au point 3.7 *Utilisation en mode Online ou en mode Téléchargement Quelle est la différence ?* sur la page 196.

### Fenêtre de propriétés des Entrées du Panneau de Commande

Chaque programme principal ou sous-programme peut avoir un panneau de commande. Les éléments de commande sont listés sous le nom du programme. Si vous n'avez pas encore enregistré d'élément de commande, aucun élément n'apparaît dans la liste. Vous devez donc tout d'abord dessiner le panneau de commande avant de pouvoir relier une entrée du panneau de commande avec un élément de commande.

Le choix dans le champ **Interface / Extension** est ignoré pour les entrées du panneau de commande car il ne s'agit pas vraiment d'entrées d'un module de l'Interface.



## 7.6.7 Sortie du panneau de commande



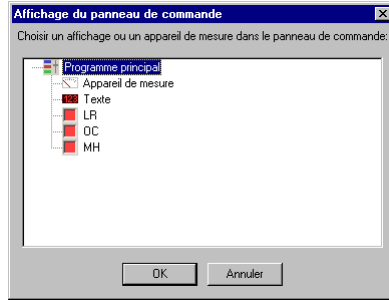
ROBO Pro vous offre la possibilité de dessiner un panneau de commande propre pour votre maquette. Vous en saurez plus sur ce sujet au chapitre 8 *Aperçu des éléments de commande et des panneaux de commande* sur la page 255. Outre les boutons et autres éléments de saisie pour la commande de votre maquette, vous pouvez ajouter des éléments d'affichage dans votre panneau de commande. Les éléments d'affichage vous permettent, par exemple, d'afficher les coordonnées des axes d'un robot ou l'état d'un commutateur de fin. Les valeurs affichées peuvent être modifiées en ajoutant un élément **Sortie du panneau de commande** dans votre programme et en envoyant une commande = à cet élément, p.e. en y raccordant une variable, une entrée analogique ou un élément de commande.



Les panneaux de commande ne peuvent être utilisés qu'en Mode Online. Vous en saurez plus au point 3.7 *Utilisation en mode Online ou en mode Téléchargement Quelle est la différence ?* sur la page 196.

### Fenêtre de propriétés des Affichages du panneau de commande

Chaque programme principal ou sous-programme peut avoir un panneau de commande. Les affichages du panneau de commande sont listés sous le nom du programme. Si vous n'avez pas encore enregistré d'élément de commande, aucun élément n'apparaît dans la liste. Vous devez donc tout d'abord dessiner le panneau de commande avant de pouvoir relier une entrée du panneau de commande avec un élément de commande.



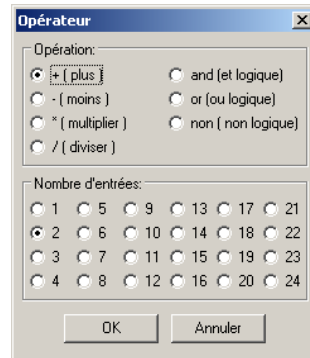
## 7.7 Opérateurs

Les éléments de programme de ce groupe sont des Opérateurs. Les opérateurs ont une ou plusieurs entrées de données orange. Les valeurs des entrées de données sont reliées à une valeur par l'opérateur et transmises à la sortie de l'opérateur par une commande =.

### Fenêtre de propriétés des Opérateurs

Tous les opérateurs utilisent la même fenêtre de propriétés. Vous pouvez transformer un opérateur en un autre opérateur via cette fenêtre de propriétés.

- Dans le champ **Opération**, vous pouvez choisir comment l'opérateur doit relier ses entrées. Les différentes fonctions sont décrites dans les deux points suivants.
- Dans le champ **Nombre d'entrées**, vous pouvez le nombre d'entrées que doit avoir l'opérateur.



#### 7.7.1 Opérateurs arithmétiques

ROBO Pro vous propose les quatre opérations élémentaires comme Opérateurs. Les symboles, pour deux entrées, sont ceux-ci :

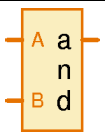
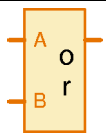
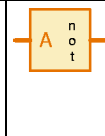
Plus	Moins	Multiplié	Divisé	Moins
$A + B$	$A - B$	$A * B$	$A / B$	$- A$

Lorsque l'opérateur **Moins** a plus de deux entrées, toutes les valeurs des autres entrées sont soustraites de la valeur de l'entrée A. Lorsque l'opérateur Moins n'a qu'une entrée, l'opérateur inverse le signe de la valeur d'entrée.

Lorsque l'opérateur **Divisé** a plus de deux entrées, la valeur de l'entrée A est divisée par les valeurs de toutes les autres entrées.

## 7.7.2 Opérateurs logiques

Pour la liaison d'entrées digitales, ROBO Pro met trois opérateurs logiques à votre disposition :

		
Et	Ou	Non
$A > 0$ et $B > 0$	$A > 0$ ou $B > 0$	$A \leq 0$

Les opérateurs logiques interprètent une valeur plus grande que 0 à l'entrée comme étant **oui** ou **vraie** et une valeur plus petite ou égale à 0 comme étant **non** ou **fausse**. Les entrées digitales fournissent une valeur de 0 ou 1 de sorte que 0 sera interprété comme **fausse** et 1 comme **vraie**.

L'opérateur **Et** envoie une commande = avec une valeur 1 aux éléments raccordés à la sortie lorsque la valeur est vraie, donc  $> 0$ , sur toutes les entrées. Dans le cas contraire, l'élément envoie une commande = avec une valeur 0.

L'opérateur **Ou** envoie une commande = avec une valeur 1 aux éléments raccordés sur la sortie lorsque la valeur d'au moins une des entrées est vraie, donc  $> 0$ . Dans le cas contraire, l'élément envoie une commande = avec une valeur 0.

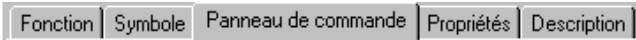
L'opérateur **Non** envoie une commande = avec une valeur 1 aux éléments raccordés sur la sortie lorsque la valeur à l'entrée est fausse, donc  $\leq 0$ . Dans le cas contraire, l'élément envoie une commande = avec une valeur 0.

Le fonctionnement des opérateurs logiques peut également être reproduit avec plusieurs éléments de branchement. Il est cependant plus clair de relier plusieurs entrées par des opérateurs.

## 8 Aperçu des éléments de commande et des panneaux de commande

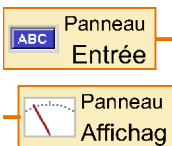
Des panneaux de commande propres peuvent être définis dans ROBO Pro. Via ces panneaux de commande, vous pouvez commander confortablement des maquettes complexes. Etant donné que les panneaux de commande s'affiche sur l'écran de votre ordinateur, ceux-ci ne peuvent être utilisés qu'en Mode Online. Voir point 3.7 *Utilisation en mode Online ou en mode Téléchargement* *Quelle est la différence ?* sur la page 196.

Pour créer un panneau de commande, sélectionnez **Panneau de commande** dans la barre de fonctions :



Vous pouvez ajouter des éléments de commande dans la surface grise vide. Un panneau de commande appartient toujours au programme principal ou au sous-programme dans lequel vous vous trouviez au moment de la création de ce panneau de commande. Il est donc important de sélectionner le bon sous-programme dans la barre des sous-programmes avant de concevoir le panneau de commande. Le plus souvent, le panneau de commande est créé dans le **Programme principal**.

Les panneaux de commande contiennent des affichages et des éléments de commande. Les affichages vous permettent d'afficher des valeurs de variables ou des messages textuels. Les éléments de commande fonctionnent quant à eux comme des boutons supplémentaires ou des entrées analogiques.



Chaque élément de commande que vous insérez dans le panneau de commande correspond dans le programme à un élément **Entrée du panneau de commande** (pour les éléments de commande) ou à une **Sortie du panneau de commande** (pour les affichages). Vous pouvez ainsi établir un lien entre votre programme et votre panneau de commande via ces éléments de programme. Vous trouverez ceux-ci dans le groupe d'éléments **Entrées, Sorties**. Selon le type d'élément de commande avec lequel vous reliez ces éléments de programme, un symbole différent sera affiché. Il n'y a que deux éléments dans la liste d'éléments : un pour les affichages et une pour les éléments de commande.

### 8.1 Affichages

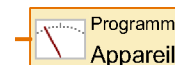
Les affichages sont utilisés comme des sorties de l'Interface. La valeur d'un affichage peut être déterminée au moyen d'une commande =.

#### 8.1.1 Appareil de mesure



L'**appareil de mesure** est inspiré d'un instrument à aiguille analogique. Il est souvent utilisé pour afficher la valeur des entrées analogiques ; vous pouvez cependant l'utiliser pour des variables ou des éléments de programme.

L'appareil de mesure est commandé au départ du programme par une sortie du panneau de commande. Vous trouverez la **Sortie du panneau de commande** dans le groupe d'éléments



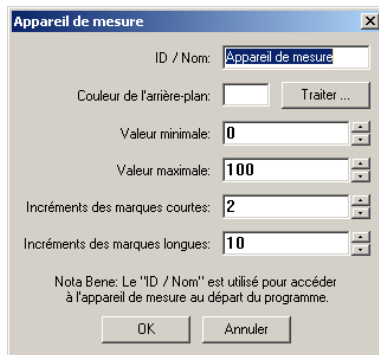
## Entrées, Sorties.

Vous déterminez la valeur de l'appareil de mesure en envoyant une commande = à la sortie du panneau de commande appropriée du programme. Presque tous les éléments de programme avec des sorties analogiques envoient une commande = lorsque leur valeur change. Vous pouvez par exemple relier directement des entrées analogiques ou des variables avec la sortie du panneau de commande.



### Fenêtre de propriété des Appareils de mesure

- Dans le champ **ID / Nom**, vous devez tout d'abord introduire un nom pour l'appareil de mesure. Le nom est important pour que vous puissiez différencier plusieurs appareils de mesure dans votre programme.
- Dans le champ **Couleur de l'arrière-plan**, vous pouvez choisir une autre couleur que le blanc.
- Dans le champ **Valeur Minimale** et **Valeur Maximale**, introduisez les valeurs correspondant aux extrémités gauche et droite de l'échelle de l'indicateur. Si une des valeurs est plus petite que 0 et l'autre, plus grande que 0, le trait 0 sera plus long que les autres.
- L'échelle se compose de traits longs et de traits courts. Vous pouvez déterminer la longueur des traits longs et des courts sous **Ecart Marques courtes / longues**. Si les deux ont la même valeur, seules des marques longues seront visibles.



## 8.1.2 Affichage textuel



Dans un affichage textuel, vous pouvez afficher des valeurs chiffrées, du texte ou les deux.

L'affichage textuel est commandé par le programme via une sortie du panneau de commande. Vous trouvez la **Sortie du panneau de commande** dans le groupe d'éléments **Entrées, Sorties**.

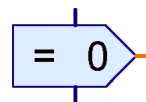


Dès que vous avez relié la Sortie du panneau de commande avec l'affichage textuel au moyen de la fenêtre de propriétés, le symbole se modifie et affiche le nom du panneau de commande (p.e. Principal) et l'affichage (p.e. Texte).



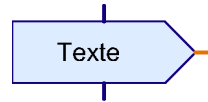
Vous pouvez afficher le texte dans l'affichage de deux façons :

- Vous pouvez afficher le contenu de l'affichage en envoyant une commande = à la Sortie du panneau de commande appropriée. C'est très pratique lorsque vous souhaitez utiliser l'affichage pour afficher la valeur d'une variable ou d'autres éléments de programme car la plupart des éléments de programme envoient automatiquement une commande = via leur sorties de données lorsque leur valeur change. La commande = n'écrase que les 6 derniers signes de l'affichage. Vous pouvez remplir le reste de l'affichage avec du texte. De cette façon, vous pouvez ajouter un texte indicatif à la valeur de l'affichage. Lorsque l'affichage contient plusieurs lignes, vous pouvez également positionner le texte indicatif sur



une ligne propre. Dans les affichages à plusieurs lignes, seuls les 6 derniers signes de la dernière ligne sont écrasés par la commande =.

- La commande Texte vous permet de déterminer le contenu de l'affichage selon vos envies. La Commande Texte est un élément de commande spécial pouvant envoyer non pas un chiffre, mais bien tout un texte via sa sortie. Comme les autres éléments de commande habituels, l'élément de commande **Texte** peut également avoir une entrée de données. Vous pouvez alors incorporer la valeur chiffrée de l'entrée de données dans le texte. Lorsque vous envoyez plusieurs commandes **Texte** à un élément d'affichage, les textes se rattachent les uns aux autres. Vous pouvez donc mélanger texte et chiffres selon vos envies.



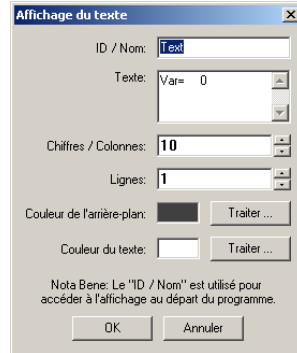
### Signes de commande des Commandes Texte

Vous pouvez utiliser les signes suivants dans vos éléments de commande **Texte** pour atteindre des effets particuliers :

Signe de commande	Effet
#####	Affiche la valeur de l'entrée de données comme un nombre à 5 chiffres + un signe
##.##	Affiche la valeur de l'entrée de données comme un nombre à deux décimales séparées par un <b>point</b> .
##,##	Affiche la valeur de l'entrée de données comme un nombre à deux décimales séparées par une <b>virgule</b> .
\c	Efface l'affichage et place un point d'insertion au début de l'affichage

### Fenêtre de propriétés des Affichages textuels

- Dans le champ **ID / Nom**, vous devez tout d'abord donner un nom à l'affichage. Le nom est important afin de pouvoir différencier plusieurs affichages dans votre programme.
- Dans le champ **Texte**, introduisez le contenu de l'affichage. Ce contenu est conservé jusqu'à ce que vous envoyez une commande à l'affichage via le programme. Lorsque vous envoyez une commande = à l'affichage, seuls les 6 derniers signes de l'affichage sont écrasés. Le début du texte reste identique de sorte que vous pouvez, devant un chiffre, ajouter une indication disant de quoi il s'agit. Dans l'exemple représenté, le texte **Var=** est conservé. L'affichage a 10 signes, donc 10-6=4 signes sont conservés.
- Dans le champ **Chiffres / Colonnes** et dans le champ **Lignes**, vous pouvez définir combien de signes peuvent être inscrit dans l'affichage. Dans les affichages à plusieurs lignes, vous pouvez afficher dans une ligne à part une indication comme **Var=** ou **Visiteurs**.
- Dans le champ **Couleur de l'arrière-plan** et **Couleur du texte**, vous pouvez modifier le design de l'affichage. Cliquez sur **Traiter...** pour sélectionner une couleur ou définir une couleur personnalisée.



### 8.1.3 Lampe de l'affichage



La **Lampe de l'affichage** est le type d'affichage le plus simple. Elle fonctionne comme un élément Lampe Fischertechnik raccordé à une sortie de l'Interface.

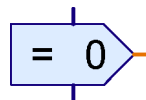
La lampe de l'affichage est commandée par le programme via une sortie du panneau de commande. La **Sortie du panneau de commande** se trouve dans le groupe d'éléments **Entrées, Sorties**.



Dès que vous avez relié la sortie du panneau de commande avec la lampe de l'affichage au moyen de la fenêtre de propriétés, le symbole se modifie et le nom du panneau de commande (p.e. Principal) et de la lampe apparaissent.

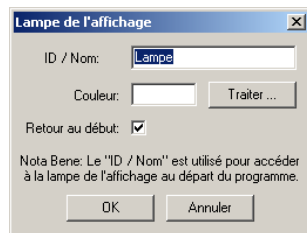


Vous pouvez allumer ou éteindre la lampe en envoyant une commande **In** ou **Out** à la sortie du panneau de commande correspondante de votre programme, comme vous le faisiez également lorsque vous utilisez les véritables sorties lampes. Vous pouvez également allumer ou éteindre la lampe de l'affichage avec une commande =. Si la valeur est supérieure à 0, la lampe s'allume. Si la valeur est inférieure ou égale à 0, la lampe s'éteint.



#### Fenêtre de propriétés des Lampes de l'affichage

- Dans le champ **ID / Nom**, vous devez tout d'abord entrer un nom pour la lampe de l'affichage. Le nom est important car il vous permet de distinguer différentes lampes de l'affichage dans votre programme.
- Dans le champ **Couleur**, vous pouvez modifier la couleur de la lampe de l'affichage. Cliquez pour ce faire sur le bouton **Traiter**.
- Si **Retour au début** est coché, la lampe de l'affichage est allumée jusqu'à ce que l'élément de programme adéquat envoie la première commande. Dans le cas contraire, la lampe de l'affichage est éteinte au départ.



## 8.2 Eléments de commande

Les éléments de commande sont utilisés comme des entrées de l'Interface.

### 8.2.1 Bouton

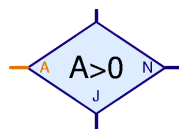


L'**Élément de commande Bouton** peut être utilisé comme un interrupteur Fischertechnik raccordé à une des entrées de l'Interface.

Le programme envoie une requête au bouton via une **Entrée du panneau de commande**. Vous trouverez l'**Entrée du panneau de commande** dans le groupe d'éléments **Entrées, Sorties**.

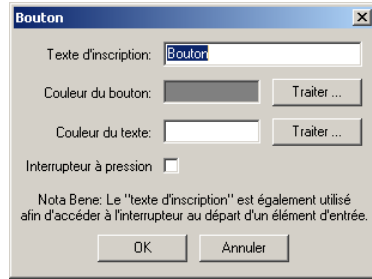


Vous pouvez raccorder l'entrée du panneau de commande correspondant au bouton à tous les éléments de programme possédant une entrée de données, p.e. à un **Branchement**, comme vous le faisiez pour une entrée digitale de l'Interface. Lorsque le bouton est actionné, il fournit une valeur 1, dans le cas contraire, la valeur est 0.



## Fenêtre de propriétés des Boutons

- Dans le champ **Texte d'Inscription**, vous pouvez introduire une inscription pour le bouton. Il s'agira également du nom grâce auquel le programme aura accès au bouton. Sur les boutons, il n'y a pas de champ ID / Nom supplémentaire comme pour les autres éléments de commande.
- Dans le champ **Couleur du bouton** et **Couleur du texte**, vous pouvez modifier le design du bouton, Pour ce faire, cliquez sur **Traiter...**
- Si la case **Interrupteur à pression** est cochée, le bouton ne se comporte pas comme un bouton, mais bien comme un interrupteur. A la première pression sur le bouton, celui-ci s'enfonce et reste enfoncé jusqu'à la seconde pression. Dans le cas contraire, le bouton se comporte vraiment comme un bouton et ressort directement lorsqu'il est relâché.



## 8.2.2 Régulateur à coulisse

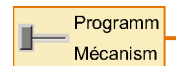


Vous pouvez utiliser le régulateur comme un potentiomètre raccordé à une entrée analogique de l'Interface. Contrairement au bouton, le régulateur peut non seulement avoir des valeurs 0 et 1, mais aussi un très grand nombre d'autres valeurs, comme une entrée analogique. Le champ de valeurs peut être réglé dans la fenêtre de propriété. Le régulateur peut, par exemple, être utilisé pour régler la vitesse du moteur entre 1 et 8.

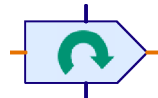
Le programme envoie une requête au régulateur est via une entrée du panneau de commande. Vous trouverez l'**Entrée du panneau de commande** dans le groupe d'élément **Entrées, Sorties**.



Dès que vous avez relié l'entrée du panneau de commande avec le régulateur au moyen de la fenêtre de propriétés, le symbole change et le nom du panneau de commande (p.e. Principal) et du régulateur apparaissent.

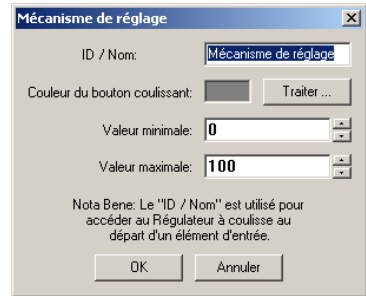


Vous pouvez raccorder l'entrée du panneau de commande appartenant au régulateur à tous les éléments de programme possédant une entrée de données, comme c'est le cas pour les entrées analogiques de l'Interface. Très souvent, le régulateur est raccordé à une élément de commande possédant une entrée de données, de sorte que le régulateur commande la vitesse d'un moteur, par exemple.



### Fenêtre de propriétés des Régulateurs

- Dans le champ **ID / Nom**, vous devez tout d'abord entrer un nom pour le régulateur. Le nom est important car il vous permet de distinguer différents régulateurs dans votre programme.
- Dans le champ **Couleur de la coulisse**, vous pouvez modifier la couleur de la coulisse. Pour ce faire, cliquez sur **Traiter**.
- Dans les champs **Valeur Minimale** et **Valeur Maximale**, introduisez un champ de valeur pour le régulateur. Si vous voulez utiliser le régulateur pour commander la vitesse d'un moteur, cette gamme de valeurs doit aller de 1 à 8.

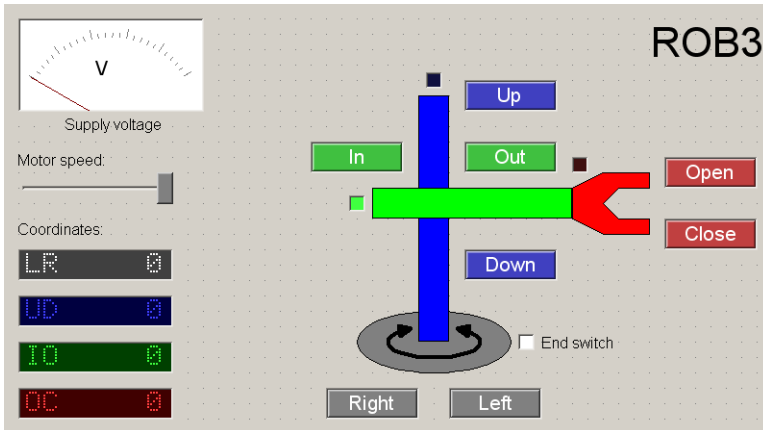




## 9 Fonctions de dessin

ROBO Pro dispose des fonctions de dessin habituelles. Vous les trouverez dans la fenêtre des groupes d'éléments sous **Dessiner**. Dans le sous-groupe **Formes**, vous trouverez des outils de dessin pour différentes formes géométriques de base. Le sous-groupe **Texte** contient les outils de texte pour différentes tailles de police. Les autres sous-groupes contiennent des fonctions pour modifier la couleur et l'épaisseur des traits.

Les fonctions de dessin vous permettent d'illustrer vos panneaux de commande et vos programmes afin que leurs différentes fonctions apparaissent plus clairement. Vous trouverez, par exemple, ci-dessous un panneau de commande pour un robot, dessiné au moyen de ces fonctions :



Les boutons, affichages de coordonnées et lampes des commutateurs de fin sont à chaque fois indiqués dans la même couleur que les axes eux-mêmes dans la représentation schématique du robot. Le panneau de commande est ainsi beaucoup plus clair.

L'utilisation des fonctions de dessin ne devrait pas poser de gros problèmes. Seuls quelques points sont donc expliqués ci-dessous ; il s'agit des points qui pourraient ne pas être clairs à première vue.

- Les objets graphiques comme les rectangles ou les cercles **ne sont pas**, comme dans de nombreux autres programmes, étirés en gardant le bouton de la souris enfoncé, mais bien au moyen de deux clics sur le bouton de la souris : un dans le coin supérieur gauche et l'autre dans le coin inférieur droit.
- Le texte n'est pas traité dans une fenêtre de dialogue, mais directement dans la zone de travail. Lorsque vous ajoutez un objet Texte, celui-ci apparaît tout d'abord sous la forme d'un cadre bleu clair. Vous pouvez alors simplement taper votre texte et il apparaît directement dans la zone de travail. Vous pouvez également ajouter du texte provenant du Presse-papier avec les touches CTRL+V.
- Après avoir dessiné un objet, vous pouvez le retravailler en déplaçant les petites manettes bleues. Certaines manettes vous permettent également de tourner ou de déformer l'objet. Un rectangle a deux manettes sur son bord supérieur gauche. Si vous déplacez les deux manettes les plus grandes, vous pouvez arrondir les angles du rectangle. Vous pouvez achever les modifications en cliquant sur le bouton droit de la souris ou avec la touche **ESC** de votre clavier.

- Si vous voulez retravailler l'objet ultérieurement, sélectionnez la fonction **Traiter** du menu **Dessiner**. Si vous cliquez ensuite sur un objet, les manettes bleu clair apparaissent à nouveau.
- De nombreux objets ont deux modes de dessin et de travail ou plus. Vous pouvez basculer entre les différents modes avec la touche Tabulations de votre clavier pendant que vous dessinez ou travaillez le dessin. Pour un cercle, vous pouvez par exemple choisir si vous devez donner deux points de la circonférence ou bien le point central et un point de la circonférence. Pour le polygone, vous pouvez basculer entre un travail au moyen de point et des fonctions comme "tourner". Pour les objets Texte, vous pouvez choisir le travail du texte, ainsi que la modification de la taille du texte ou l'angle d'inclinaison.
- Dans le menu Dessiner, vous trouverez la fonction **Objet à l'avant-plan / à l'arrière-plan**. Ces fonctions vous permettent de faire glisser tous les éléments sélectionnés (marqués en rouge) vers l'avant ou vers l'arrière, de sorte que les autres objets soient recouverts ou recouverts ces objets.
- Avec la fonction **Snap de la trame** du menu **Dessiner**, vous permet d'afficher ou non la trame du dessin. Veillez cependant à ce que la trame soit affichée lorsque vous travaillez dans votre programme car tous les éléments sont adaptés à la trame.
- Pour les objets Texte, vous pouvez modifier l'orientation du texte en appuyant sur la touche CTRL et une touche de 1 à 9 du pavé numérique. Ceci est seulement possible lorsque la lampe Num-Lock du clavier est allumée. Si ce n'est pas le cas, vous devez d'abord enfoncer la touche Num-Lock.