

| | |
|---------------------------------------|-----------|
| CHAPITRE N°1. | 1 |
| INTRODUCTION | 1 |
| CHAPITRE N°2. | 2 |
| ASSISTANCE | 2 |
| CHAPITRE N°3. | 3 |
| INSTALLATION | 3 |
| CHAPITRE N°4. | 5 |
| PRISE EN MAIN | 5 |
| CHAPITRE N°5. | 9 |
| L'ENVIRONNEMENT IRIS | 9 |
| EXÉCUTION AU DÉMARRAGE | 11 |
| CHAPITRE N°6. | 12 |
| GÉNÉRALITÉS SUR LES OBJETS | 12 |
| CHAPITRE N°7. | 14 |
| PROCÉDURES | 14 |
| CHAPITRE N°8. | 17 |
| Liste des objets | 17 |
| CHAPITRE N°9. | 19 |
| ACCÈS À L'ÉTAT DES VARIABLES | 19 |
| CHAPITRE N°10. | 20 |
| ORDRES | 20 |
| MODIFICATION DE L'ÉTAT D'UNE VARIABLE | 20 |
| RUN | 20 |
| STOP | 20 |
| INIT | 20 |

| | |
|---|-----------|
| STEP | 20 |
| OPENAOF(NOM D'UN OBJET) | 20 |
| CHAINAOF(NOM D'UN OBJET) | 20 |
| MINIMIZEAOF(TITRE D'UN PUPITRE) | 21 |
| RESTOREAOF(TITRE D'UN PUPITRE) | 21 |
| CLOSEAOF() | 21 |
| ECHANGES ENTRE OBJETS | 21 |
| PARENTPARAM(PARAMÈTRE {+N} {-N}) | 21 |
| SISTERPARAM(IDENTIFICATEUR , PARAMÈTRE) | 21 |
| CHAPITRE N°11. | 22 |
| <hr/> | |
| L'OBJET PUPITRE | 22 |
| CHAPITRE N°12. | 24 |
| <hr/> | |
| LES PROPRIÉTÉS DE L'OBJET PUPITRE - ONGLET « ASPECT » | 24 |
| LES PROPRIÉTÉS DE L'OBJET PUPITRE - ONGLET « BITMAP » | 25 |
| LES PROPRIÉTÉS DE L'OBJET PUPITRE - ONGLET « LIENS » | 25 |
| LES PROPRIÉTÉS DE L'OBJET PUPITRE - ONGLET « OPTIONS » | 25 |
| LES PROPRIÉTÉS DE L'OBJET PUPITRE - ONGLET « ENFANTS » | 25 |
| CHAPITRE N°13. | 26 |
| <hr/> | |
| L'OBJET BPVOYANT | 26 |
| CHAPITRE N°14. | 27 |
| <hr/> | |
| LES PROPRIÉTÉS DE L'OBJET BPVOYANT - ONGLET « ASPECT » | 27 |
| LES PROPRIÉTÉS DE L'OBJET BPVOYANT - ONGLET « LIENS » | 27 |
| LES PROPRIÉTÉS DE L'OBJET BPVOYANT - ONGLET « OPTIONS » | 28 |
| CHAPITRE N°15. | 30 |
| <hr/> | |
| L'OBJET VALDIGI | 30 |
| CHAPITRE N°16. | 31 |
| <hr/> | |
| LES PROPRIÉTÉS DE L'OBJET VALDIGI - ONGLET « ASPECT » | 31 |
| LES PROPRIÉTÉS DE L'OBJET VALDIGI - ONGLET « TEXTES » | 32 |
| LES PROPRIÉTÉS DE L'OBJET VALDIGI - ONGLET « LIENS » | 32 |
| CHAPITRE N°17. | 33 |
| <hr/> | |
| L'OBJET VALANA | 33 |

| | |
|--|-----------|
| CHAPITRE N°18. | 34 |
| LES PROPRIÉTÉS DE L'OBJET VALANA - ONGLET « ASPECT » | 34 |
| LES PROPRIÉTÉS DE L'OBJET VALANA - ONGLET « LIENS » | 34 |
| LES PROPRIÉTÉS DE L'OBJET VALANA - ONGLET « BORNES » | 35 |
| LES PROPRIÉTÉS DE L'OBJET VALANA - ONGLET « GRADUATIONS » | 35 |
| CHAPITRE N°19. | 37 |
| L'OBJET ECRANCLA | 37 |
| CHAPITRE N°20. | 39 |
| LES PROPRIÉTÉS DE L'OBJET ECRANCLA - ONGLET « ASPECT » | 39 |
| LES PROPRIÉTÉS DE L'OBJET ECRANCLA - ONGLET « LIENS » | 39 |
| LES PROPRIÉTÉS DE L'OBJET ECRANCLA - ONGLET « LISTE » | 39 |
| LES PROPRIÉTÉS DE L'OBJET ECRANCLA - ONGLET « OPTIONS » | 40 |
| LES PROPRIÉTÉS DE L'OBJET ECRANCLA - ONGLET « MESSAGES » | 41 |
| CHAPITRE N°21. | 42 |
| L'OBJET ARCHIVE | 42 |
| CHAPITRE N°22. | 43 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « ASPECT » | 43 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « DONNÉES » | 43 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « OPTIONS » | 45 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « TABLES » | 46 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « COURBE » | 47 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « GRADUATIONS » | 48 |
| LES PROPRIÉTÉS DE L'OBJET ARCHIVE - ONGLET « GRILLE » | 49 |
| CHAPITRE N°23. | 50 |
| L'OBJET OBJET | 50 |
| CHAPITRE N°24. | 51 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « ASPECT » | 51 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « LIENS » | 52 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « FORMES » | 53 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « BITMAP » | 53 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « COULEURS » | 54 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « JAUGE » | 55 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « CAPTEUR » | 55 |
| LES PROPRIÉTÉS DE L'OBJET OBJET - ONGLET « OPTIONS » | 56 |

| | |
|---|-----------|
| CHAPITRE N°25. | 57 |
| OBJET OBJET - TECHNIQUES AVANCÉES | 57 |
| CHAPITRE N°26. | 58 |
| OBJET SON | 58 |
| CHAPITRE N°27. | 59 |
| PROPRIÉTÉS DE L'OBJET SON - ONGLET « ASPECT » | 59 |
| PROPRIÉTÉS DE L'OBJET SON - ONGLET « SONS » | 59 |
| CHAPITRE N°28. | 60 |
| OBJET DIAL | 60 |
| CHAPITRE N°29. | 61 |
| PROPRIÉTÉS DE L'OBJET DIAL - ONGLET « ASPECT » | 61 |
| PROPRIÉTÉS DE L'OBJET DIAL - ONGLET « LIENS » | 61 |
| PROPRIÉTÉS DE L'OBJET DIAL - ONGLET « MESSAGES » | 62 |
| CHAPITRE N°30. | 63 |
| OBJET PROG | 63 |
| RÉPARTITION DU TEMPS D'EXÉCUTION | 63 |
| AFFICHAGE | 64 |
| SYNTAXE | 64 |
| DÉCLARATION DES VARIABLES | 64 |
| ÉCRITURE DU PROGRAMME | 65 |
| CONSTANTES | 65 |
| AFFECTATION | 65 |
| CALCULS | 65 |
| TESTS | 66 |
| BOUCLES | 67 |
| ADRESSE D'UNE VARIABLE OU D'UN TABLEAU DE VARIABLES | 67 |
| LISTE DES FONCTIONS | 67 |
| MESSAGES D'ERREURS | 74 |
| CHAPITRE N°31. | 76 |
| PROPRIÉTÉS DE L'OBJET PROG - ONGLET « ASPECT » | 76 |
| COULEURS | 76 |
| TAILLE DE L'OBJET | 76 |
| TEXTE | 76 |
| PROPRIÉTÉS DE L'OBJET PROG - ONGLET « PROGRAMME » | 76 |
| PROGRAMME | 76 |

| | |
|--|------------|
| EXÉCUTER | 76 |
| EXÉCUTION PRIORITAIRE | 76 |
| EXÉCUTION AU DÉMARRAGE | 76 |
| ALLER SUR L'ERREUR | 77 |
| <u>CHAPITRE N°32.</u> | 78 |
| EXEMPLE D'OBJET COMPOSÉ | 78 |
| <u>CHAPITRE N°33.</u> | 81 |
| EXEMPLE D'UTILISATION DE L'OBJET ECRANCLA COMME LISTE À MESSAGES | 81 |
| <u>CHAPITRE N°34.</u> | 87 |
| EXEMPLE D'UTILISATION DE L'OBJET ECRANCLA COMME TERMINAL | 87 |
| <u>CHAPITRE N°35.</u> | 90 |
| EXEMPLE D'APPLICATION COMPOSÉE DE PLUSIEURS PAGES | 90 |
| <u>CHAPITRE N°36.</u> | 92 |
| EXEMPLE D'UTILISATION DE L'OBJET OBJET | 92 |
| <u>CHAPITRE N°37.</u> | 98 |
| EXEMPLES D'UTILISATION DE L'OBJET ARCHIVE | 98 |
| <u>CHAPITRE N°38.</u> | 102 |
| EXEMPLE D'UTILISATION DE L'OBJET PROG | 102 |
| <u>CHAPITRE N°39.</u> | 105 |
| EXEMPLES D'APPLICATION DE SUPERVISION | 105 |
| <u>CHAPITRE N°40.</u> | 110 |
| EXEMPLE D'APPLICATION DE SUPERVISION | 110 |
| <u>CHAPITRE N°41.</u> | 112 |
| EXEMPLE DE SIMULATION D'UNE PARTIE OPÉRATIVE | 112 |

CHAPITRE N°1.

Introduction

Cette notice est destinée aux utilisateurs de la version d'IRIS V3.

IRIS V3 est un progiciel de supervision et de simulation de parties opératives fonctionnant sous WINDOWS (3.1, 3.11, 95 ou NT 4.0).

IRIS V3 est disponible en version 16 bits pour WINDOWS 3.1, 3.11 et 95.

IRIS V3 est disponible en version 32 bits pour WINDOWS 95 et NT 4.0.

IRIS V3 utilise le concept des objets. Les objets sont des entités séparées qui permettent de composer les applications. Les objets sont spécialisés dans un certain domaine : par exemple il existe un objet pour créer des boutons poussoirs. En dehors de cette spécialisation, les objets sont paramétrables et peuvent ainsi à partir d'un type de base (le bouton poussoir par exemple) être adaptés à vos besoins (associer un voyant par exemple ou donner une couleur particulière au bouton poussoir).

Les objets d'IRIS V3 peuvent être exploités à partir de deux environnements différents : l'environnement IRIS V3 ou AUTOMGEN V6.

A partir d'AUTOMGEN les objets se comportent comme les éléments d'aide à la mise au point déjà disponibles (les fenêtres d'état de variables par exemple). Ils permettent de créer des applications de supervision ou de simulation de processus industriels. Le contexte d'exécution est spécifié par AUTOMGEN comme habituellement : cible, programme exécuté, etc ...

A partir de l'environnement IRIS la seule différence est que les objets peuvent fonctionner de façon autonome (sans AUTOMGEN). Le contexte d'exécution est spécifié dans l'environnement IRIS et, est associé à un nom de projet. L'utilisation de l'environnement IRIS est typiquement réservée à la création d'applications de supervision devant fonctionner de façon autonome en dehors d'AUTOMGEN. Les modules de communications utilisés sont les mêmes que pour AUTOMGEN.

CHAPITRE N°2.

Assistance

Notre service d'assistance est ouvert à tous les utilisateurs d'AUTOMGEN et d'IRIS.

Vous pouvez nous contacter :

- par téléphone en composant le (04) 66 54 91 30,
- par fax en composant le (04) 66 54 91 33,
- par courrier : IRAI – 11 rue des Eglantiers - BP 14 - 30110 LA GRAND COMBE,
- par email : contact@irai.com,
- sur internet : <http://www.irai.com>.

CHAPITRE N°3.

Installation

Le logiciel IRIS se compose de :

- 2 disquettes repérées « IRIS V3 »,
- un dongle ou une disquette de protection,
- un manuel de l'utilisateur.

Dongle

Les droits d'utilisation du logiciel IRIS version 3 peuvent se trouver sur un dongle ou une disquette de protection AUTOMGEN. Si vous utilisez simultanément IRIS et AUTOMGEN avec deux dongles distincts alors, le dongle IRIS doit être placé après le dongle AUTOMGEN.

Placez le dongle sur un des ports parallèles de votre PC.

Configuration minimale requise pour la version 16 bits

- compatible PC 386,
- 4 Mo RAM,
- espace libre sur le disque dur 10 Mo,
- carte SVGA 800x600x16 couleurs,
- WINDOWS 3.1 ou 3.11 ou 95 ou 98,
- une souris.

Configuration minimale requise pour la version 32 bits

- compatible PC PENTIUM,
- 8 Mo RAM,
- espace libre sur le disque dur 20 Mo,
- carte SVGA 800x600x16 couleurs,
- WINDOWS 95 ou 98 ou NT 4.0,
- une souris.

Procédure d'installation

Pour installer IRIS V3 placez la disquette repérée « IRIS V3 DISQUE 1/2 » dans un des lecteurs de disquettes de votre ordinateur et lancez l'exécution du programme « IRIAINST.EXE ».

Dans la procédure d'installation, vous pouvez choisir les options à installer :

- « Installer l'environnement IRIS » : pour pouvoir utiliser les objets en dehors d'AUTOMGEN (réalisation d'applications de supervision autonomes),
- « Installer les objets » : obligatoire pour créer des applications avec IRIS,
- « Installer les exemples » : installation des exemples d'applications.

Vous pouvez également préciser le répertoire où doit être installé IRIS V3. Si vous souhaitez utiliser simultanément IRIS et AUTOMGEN sur le même PC, installez IRIS V3 dans le même répertoire que le logiciel AUTOMGEN.

Lancement de l'environnement IRIS

Le lancement de l'environnement IRIS se fera en lançant l'exécution de « IRIS32.EXE ».

CHAPITRE N°4.

Prise en main

Ce chapitre va vous permettre de créer très rapidement votre première application IRIS V3 à partir d'AUTOMGEN. Ce premier exemple va consister à créer un pupitre, à y déposer un objet IRIS (un bouton poussoir) et à lier l'objet aux variables de l'application d'automatisme.

Etape 1

Créez une application minimale avec AUTOMGEN (juste une étape par exemple) et générez le programme exécutable pour cette application. La cible peut être le PC ou un automate.

Etape 2

Lancez l'exécution de l'application AUTOMGEN (option « Charger » du menu « Exécuter »). Passez l'application en mode RUN (option « Exécuter » du menu « Exécuter »).

Etape 3


Choisissez l'option « Objets » de menu « Debug ». Sélectionnez le fichier « PUPITRE.AOF » se trouvant dans le sous-répertoire « AOF » du répertoire où a été installé IRIS V3.

A ce stade l'objet « PUPITRE.AOF » doit apparaître à l'écran sous cette forme :

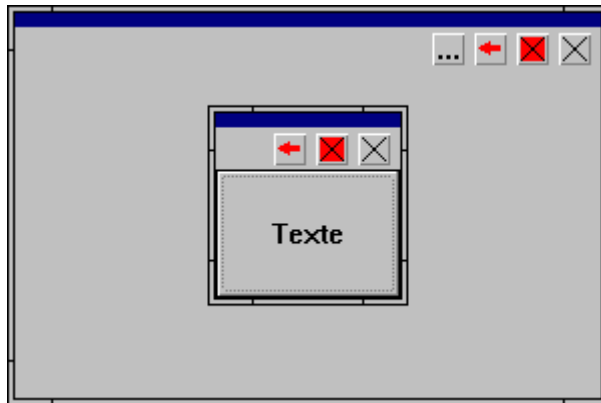


L'objet « PUPITRE.AOF » est un objet particulier qui est le seul capable de contenir d'autres objets.


Etape 4

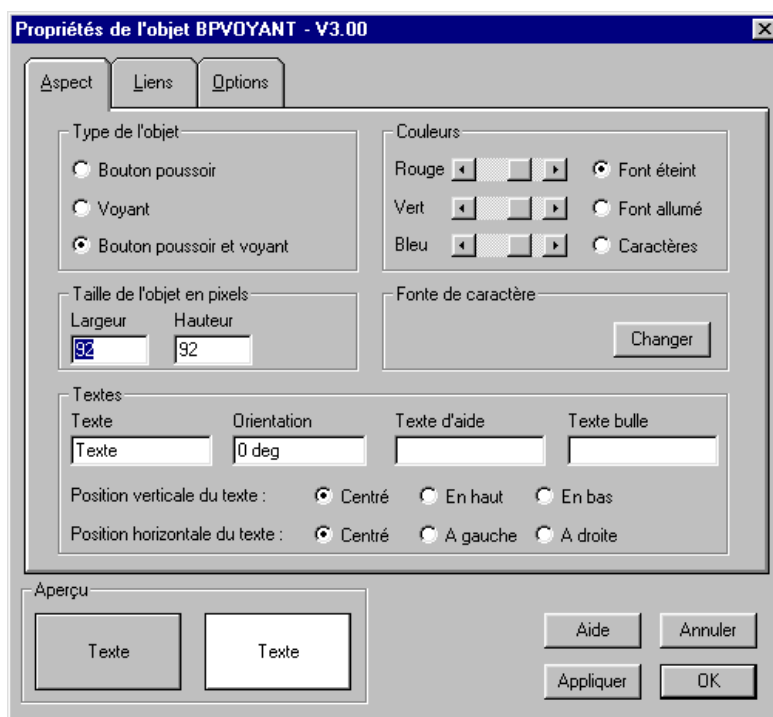
Pour ajouter un bouton poussoir sur le pupitre vous devez maintenant cliquer sur l'icône  du pupitre (accès au menu) et sélectionner l'option « Ajouter un objet ». Sélectionnez l'objet

« BPVOYANT.AOF » présent dans le sous-répertoire « AOF » du répertoire où est installé IRIS V3. L'objet « BPVOYANT » doit alors apparaître sur le pupitre :




Etape 5

Nous allons associer le bouton poussoir à une sortie de l'application d'automatisme, O4 par exemple (cette sortie ne doit pas être affectée dans votre application d'automatisme). Cliquez sur l'icône  du bouton poussoir (pas du pupitre). La boîte de dialogue des propriétés du bouton poussoir s'ouvre :





Cliquez sur l'onglet « Liens » (en haut de la fenêtre de dialogue). Sous la rubrique « Action lorsque le bouton est enfoncé » entrez « O4=1 ». Sous la rubrique « Action lorsque le bouton est relâché » entrez « O4=0 ». Cliquez ensuite sur le bouton poussoir « OK » en bas de la fenêtre de dialogue. Les actions sur le bouton poussoir ont pour effet de piloter la sortie 4 de l'application d'automatisme. Vous pouvez ouvrir une fenêtre « Etat de variable » dans AUTOMGEN pour visualiser l'état de la sortie 4 pendant que vous cliquez puis relâchez le bouton poussoir. Si l'état de la variable n'est pas modifié, c'est certainement que l'application d'automatisme n'est pas en RUN.

Etape 6

Nous allons associer un voyant à l'objet « BPVOYANT », ce voyant sera associé à une entrée de l'application d'automatisme (i2 par exemple). Cliquez de nouveau sur l'icône  du bouton poussoir. Dans l'onglet « Aspect » cliquez sur le bouton radio « Bouton poussoir et voyant ». Cliquez sur l'onglet « Liens » et entrez « I2 » sous la rubrique « Etat du voyant ». Cliquez sur le bouton poussoir « OK » en bas de la fenêtre de dialogue des propriétés. Vous pouvez maintenant modifier l'état de la variable « I2 » (avec une fenêtre « Etat d'une variable » ou en modifiant l'état de l'entrée physique si elle existe).

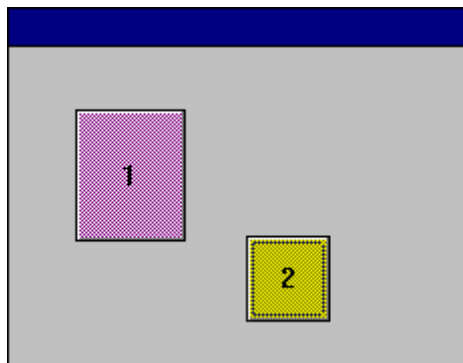
Etape 7

Nous allons dupliquer l'objet « BPVOYANT » présent sur le pupitre pour obtenir un deuxième bouton poussoir dont nous modifierons les propriétés. Cliquez sur le bouton poussoir avec le bouton gauche de la souris en maintenant la touche [SHIFT] enfoncée. Des carrés noirs apparaissent autour de l'objet sélectionné. Cliquez sur l'icône  du pupitre et choisissez l'option « Copier ». Cliquez sur l'icône  du pupitre et choisissez l'option « Coller ». Il y a maintenant deux objets « BPVOYANT » superposés. Saisissez celui du dessus (c'est le seul accessible) par sa barre supérieure et déplacez le en dehors de l'autre bouton poussoir. L'objet ainsi dupliqué possède les mêmes propriétés que le premier. Vous pouvez maintenant paramétrer le deuxième objet pour qu'il soit par exemple lié à la sortie 5 et l'entrée 3.

Vous pouvez également personnaliser l'aspect des boutons poussoirs en utilisant l'onglet aspect pour les deux objets. Vous pouvez aussi modifier la taille des objets en les saisissant par un de leurs bords.

Les trois objets présents à l'écran (le pupitre et les deux boutons poussoirs) sont en mode « Configuration », c'est à dire qu'ils laissent apparaître une mini barre en haut de leur surface, des icônes et une bordure permettant de les dimensionner. Les objets ont un autre mode nommé « Exploitation » dans lequel ils ont un aspect définitif : plus de barre en haut, plus d'icône et plus de bordure pour les dimensionner. Pour basculer un objet d'un mode à l'autre, il suffit de cliquer dessus avec le bouton droit de la souris.

A ce stade vous devez avoir créé un objet qui ressemble à ceci :



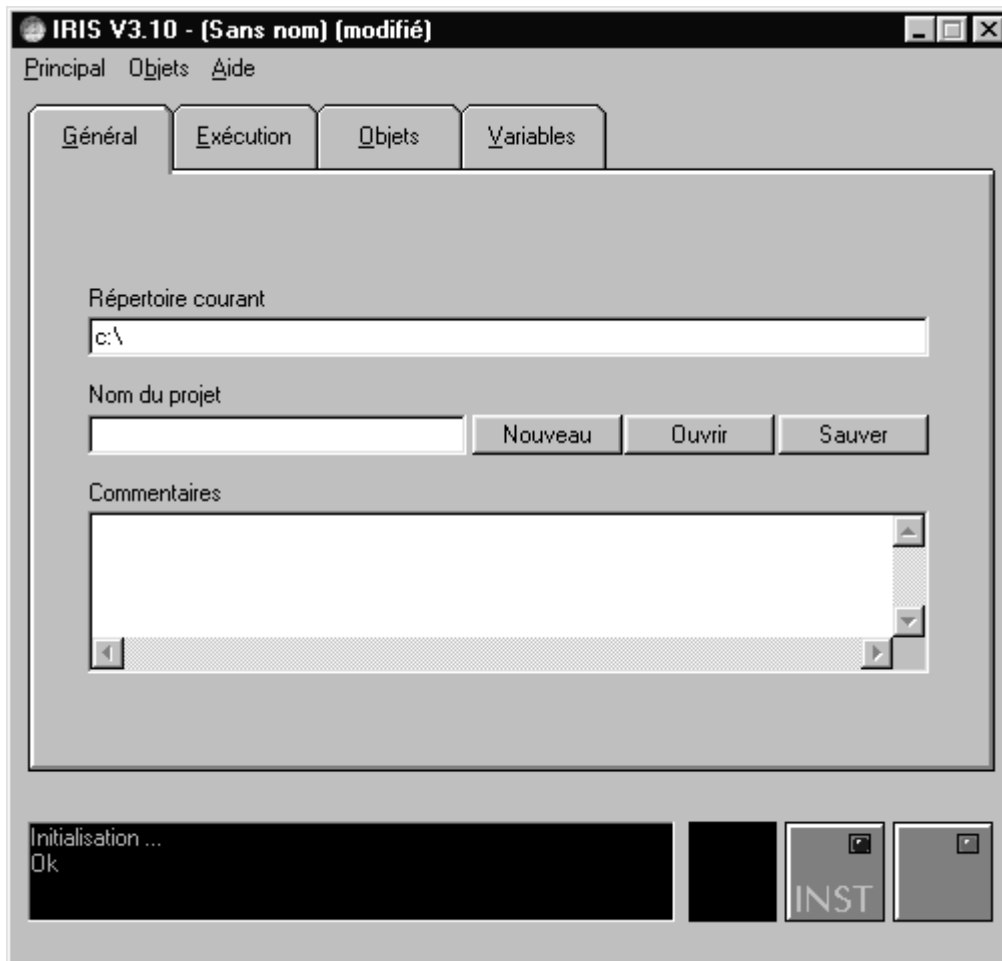
Etape 8

Nous allons pour terminer sauvegarder les objets ainsi créés vers un nouveau fichier « TEST.AOF ». Si des fenêtres « Etat de variables » sont ouvertes fermez les. Dans le menu « Debug » choisissez l'option « Sauver tous les objets ». Tapez dans la zone de saisie du nom du fichier « TEST.AOF » puis cliquez sur « OK ». L'objet « TEST.AOF » est créé. Vous pouvez le relire avec l'option « Objets » du menu « Debug » comme vous l'avez fait avec les objets de base.

CHAPITRE N°5.

L'environnement IRIS

L'environnement IRIS permet de créer des applications de supervision autonomes. L'environnement permet de configurer le contexte d'exécution dans lequel les objets de l'application fonctionneront.



La fenêtre de l'environnement IRIS comporte deux zones principales :

La zone en haut de la fenêtre permet de paramétrer le contexte d'exécution : elle est représentée par plusieurs rubriques organisées sous formes d'onglets.

La zone en bas de la fenêtre contient une fenêtre de messages pour les messages généraux (initialisation de l'environnement, sauvegarde, etc ...), et les messages liés à l'exécution : installation d'un exécuteur, modes de marche, etc... Deux boutons poussoirs sont également disposés dans la zone du bas : un bouton poussoir repéré « Inst » indique si un exécuteur est installé (voyant rouge allumé) et permet d'installer ou de désinstaller. L'environnement IRIS peut être installé lorsque AUTOMGEN fonctionne. Dans ce cas IRIS utilise l'exécuteur

installé par AUTOMGEN. Le deuxième bouton poussoir repéré « Run » indique l'état RUN (voyant bleu allumé) et permet également de piloter l'état RUN/STOP.

Onglet Général - Répertoire courant

Cette rubrique contient le nom du répertoire courant pour l'application de supervision. Les noms de fichiers utilisés pour paramétrer les objets et qui ne comportent pas un nom d'accès complet feront référence à ce répertoire.

Onglet Général - Nom du projet

Cette rubrique contient le nom du projet. L'extension par défaut pour les projets IRIS V3 est « OPF » (Object Project File).

Onglet Général - Nouveau, Ouvrir et Sauver

Ces boutons poussoirs permettent de créer un nouveau projet, d'ouvrir un projet existant ou de sauver le projet courant.

Onglet Général - Commentaires

Cette rubrique contient des lignes de commentaires associées au projet. Elles vous permettent d'écrire des informations relatives au projet.

Onglet Exécution - Nom de l'exécuteur ou du module de dialogue

Cette rubrique contient le nom du module de dialogue ou de l'exécuteur à installer au lancement de l'application. IRIS V3 utilise les mêmes exécuteurs ou module de dialogue que la version 6 d'AUTOMGEN. Un nom d'accès complet doit être précisé. Par exemple : « C:\AUTOMV6\EXECPC32.EXE » ou « C:\AUTOMV6\IFPC32.EXE ».

Onglet Exécution - Paramètres de l'exécuteur ou du module de dialogue

Cette rubrique contient les paramètres de l'exécuteur ou du module de dialogue. Par exemple pour utiliser le port COM2 avec le module de dialogue « IFPC.EXE » cette rubrique devra contenir « -C2 ». Le paramètre « -X » permet d'annuler les opérations de téléchargement et de vérification des applications.

Onglet Exécution - Nom de l'application

Cette rubrique contient le nom de l'application d'automatisme. Ce nom est utilisé pour éventuellement relire le fichier des symboles de l'application. Si l'exécuteur PC est utilisé le nom de l'application permet également de spécifier le fichier à exécuter.

Onglet Objets

Cet onglet contient la liste des objets de l'application courante. La liste est à sélection multiple. C'est à dire que vous pouvez sélectionner un nombre quelconque d'éléments dans cette liste : pour sélectionner plusieurs éléments cliquez avec le bouton gauche de la souris en maintenant

la touche [CONTROL] enfoncée pour sélectionner ou désélectionner un objet. Les boutons poussoirs « Tout » et « Rien » permettant de sélectionner l'ensemble des objets ou aucun.

Le bouton poussoir « Ouvrir » permet d'ajouter un objet à la liste.

Le bouton poussoir « Sauver » permet de sauvegarder les objets sélectionnés vers un autre objet.

Le bouton poussoir « Effacer » permet de détruire (en mémoire) les objets sélectionnés.

Le bouton poussoir « Propriétés » permet d'accéder aux propriétés des objets sélectionnés.

Le bouton poussoir « Dupliquer » permet de dupliquer les objets sélectionnés.

Le bouton poussoir « Position » permet d'entrer les coordonnées des objets sélectionnés.

La case à cocher « Configuration autorisée » autorise ou interdit l'accès à la configuration des objets.

Onglet Variables

Cette rubrique permet de définir les plages de variables auxquelles les objets accèdent. Cette définition n'est pas obligatoire mais elle accélère les accès aux variables. C'est en fait un système de cache qui optimise les accès aux variables. Son utilisation est effective pour les modules de communication sur liaison série. Chaque ligne de cette rubrique doit contenir un nom de variable suivi d'une longueur. Par exemple :

I0,32

M200,50

Ces deux lignes signifient que l'application IRIS accédera aux entrées I0 à I31 et aux mots M200 à M249.

Menu Objets

Ce menu regroupe deux commandes :

« Montrer cacher » : permet de masquer ou de montrer l'ensemble des objets ouverts.

« Lister la configuration » (version 32 bits seulement) : génère un fichier texte contenant la configuration du ou des objets sélectionnés.

Exécution au démarrage

Au lancement de « IRIS32.EXE » l'option « /EXEC » peut être placée sur la ligne de commande pour installer automatiquement un module de communication ou un exécuteur.

CHAPITRE N°6.

Généralités sur les objets

Les objets d'IRIS V3 permettent de réaliser :


- des applications de supervision,
- des compléments aux outils de visualisation dynamique pour AUTOMGEN,
- des simulateurs de parties opératives.


Les objets sont contenus dans des fichiers portant l'extension « .AOF ». Ces fichiers peuvent contenir un objet ou un groupe d'objets (un nouvel objet composé de plusieurs objets de base).


Chaque objet possède des propriétés qui définissent son aspect et ses fonctionnalités.

Les objets possèdent deux modes : le mode de configuration et le mode d'exploitation.

Dans le mode de configuration l'objet peut être déplacé et redimensionné comme une fenêtre WINDOWS classique : pour le déplacer il faut le saisir par la barre de titre, pour le dimensionner il faut saisir un de ses bords et le déplacer. Dans ce mode il laisse également apparaître des boutons poussoirs en haut à gauche de sa surface.

Le bouton poussoir  détruit l'objet (l'objet en mémoire, pas le fichier qui lui est associé).

Le bouton poussoir  passe l'objet en mode exploitation. Cliquer sur l'objet avec le bouton droit de la souris est équivalent et permet également de passer et de revenir du mode exploitation au mode configuration.

Le bouton poussoir  ouvre la boîte de dialogue permettant de régler les propriétés de l'objet. Cliquer sur l'objet avec le bouton droit de la souris et la touche [CTRL] enfoncée a le même effet et peut être utilisé même si l'objet est en mode exploitation.

En dehors de ces trois boutons poussoirs les objets peuvent en contenir d'autres permettant d'accéder à des fonctions spécifiques.

Exemple d'objet en mode configuration :



En mode exploitation les objets ont leur aspect « définitif ».

Le même objet en mode exploitation :




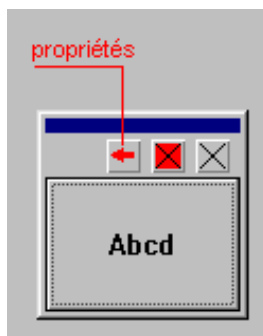
« PUPITRE.AOF » est un objet particulier qui peut recevoir d'autres objets sur sa surface. C'est une enveloppe qui permet de créer des objets composés, formés d'objets de bases solidaires.

CHAPITRE N°7.

Procédures

Modifier les propriétés d'un objet

Pour modifier les propriétés d'un objet il faut cliquer sur l'icône  avec le bouton gauche de la souris ou cliquer sur l'objet avec le bouton droit en maintenant la touche [CTRL] enfoncée.



La définition des propriétés d'un objet est réalisée à l'aide d'une boîte de dialogue spécifique à chaque objet. Ces boîtes de dialogues contiennent des éléments séparés en différentes rubriques matérialisées par des onglets. Pour accéder à une rubrique, il suffit de cliquer sur l'onglet correspondant ou d'utiliser la combinaison de touche [ALT] + la lettre soulignée qui apparaît dans le nom de l'onglet.

Les boîtes de dialogues permettant de modifier les propriétés d'un objet possèdent 5 éléments immuables en bas de leurs surfaces :

- une zone « Aperçu » qui donne une idée approximative de l'aspect de l'objet par rapport aux réglages en cours,
- un bouton poussoir « AIDE » qui donne accès à une aide spécifique à l'objet,
- un bouton poussoir « OK » qui valide les réglages en cours et ferme la boîte de dialogue,
- un bouton poussoir « APPLIQUER » qui valide les réglages en cours sans sortir de la boîte de dialogue, il est ainsi possible d'observer les résultats d'une configuration sans sortir de la boîte de dialogue,
- un bouton poussoir « ANNULER » qui ferme la boîte de dialogue sans valider les modifications, si le bouton poussoir « APPLIQUER » a été utilisé c'est la configuration présente à cet instant qui est remise en place.

La rubrique « ASPECT » des objets contient également de façon standard la taille de l'objet. Ceci permet de définir précisément la taille d'un objet.

Déplacement d'un objet

Pour déplacer un objet il faut le saisir par la barre de tire lorsqu'il est en mode configuration.

Pendant le déplacement d'un objet les flèches du curseur permettent de déplacer l'objet d'un pixel.


Si l'objet appartient à un pupitre il est possible de le déplacer en choisissant l'onglet « Enfants » de la boîte de dialogue de propriétés du pupitre. A partir de l'environnement AUTOMGEN c'est le bouton poussoir « POSITION » de la rubrique « OBJETS » qui permet de modifier la position.

Changement de la taille d'un objet

Pour changer la taille d'un objet deux possibilités :


- saisir un de ses bords et le déplacer (l'objet doit être en mode configuration),
- modifier précisément la taille d'un objet en utilisant les éléments « TAILLE DE L'OBJET » de la rubrique « ASPECT » de la boîte de dialogue des propriétés de l'objet.

Changement de mode

Le changement de mode d'un objet s'effectue en cliquant dessus avec le bouton droit de la souris. En mode configuration l'icône  permet de passer en mode exploitation.


Accès aux propriétés de l'objet

Quatre possibilités :

- cliquez sur l'icône  de l'objet,
- cliquez sur l'objet avec le bouton droit de la souris tout en maintenant la touche [CTRL] enfoncée,
- si la fenêtre appartient à un pupitre, cliquez sur le bouton poussoir « PROPRIETES » de la rubrique « ENFANTS » (sélectionnez l'enfant dans la liste au préalable),
- à partir de l'environnement IRIS : cliquez sur le bouton poussoir « PROPRIETES » de la rubrique « OBJETS » (sélectionnez l'enfant dans la liste au préalable).

Destruction d'un objet (en mémoire)

Trois possibilités :

- cliquez sur l'icône  de l'objet,
- si l'objet appartient à un pupitre, cliquez sur le bouton poussoir « DETRUIRE » de la rubrique « ENFANTS » (sélectionnez l'enfant dans la liste au préalable),
- à partir de l'environnement IRIS : cliquez sur le bouton poussoir « DETRUIRE » de la rubrique « OBJETS » (sélectionnez l'enfant dans la liste au préalable).

Sauvegarder un objet

- depuis AUTOMGEN : utilisez la commande « SAUVER TOUS LES OBJETS » du menu « DEBUG » pour sauvegarder tous les objets existants en mémoire vers un fichier « .AOF ».
- depuis IRIS : le bouton poussoir « SAUVER » de la rubrique « OBJETS » permet de sauvegarder les objets sélectionnés.

Relire un objet

Depuis AUTOMGEN : utilisez la commande « OBJETS » du menu « DEBUG ». Depuis IRIS utilisez le bouton poussoir « OUVRIR » de la rubrique « OBJETS ».

Copier, couper, coller, aligner, dupliquer, déplacer des objets

Ces commandes sont gérées par les objets « PUPITRE ».

L'environnement IRIS permet de dupliquer les objets : bouton poussoir « DUPLIQUER » de la rubrique « OBJETS ».

Interdire la configuration des objets

Depuis AUTOMGEN : c'est la marque placée sur la rubrique « AUTORISER LA CONFIGURATION DES OBJETS » du menu « OPTION » qui détermine si les objets peuvent être configurés. Pour IRIS c'est la case à cocher « CONFIGURATION AUTORISEE » de la rubrique « OBJETS ».

CHAPITRE N°8.

Liste des objets

Les objets de base développés par IRAI sont les suivants :

PUPITRE

Cet objet permet de regrouper et de manipuler d'autres objets.

BPVOYANT

Objet de type bouton poussoir, voyant ou une combinaison des deux : bouton poussoir à voyant intégré. Cet objet est typiquement utilisé pour visualiser ou modifier l'état de variables booléennes.

VALDIGI

Cet objet permet d'afficher et de modifier l'état de variables numériques sous forme digitale.

VALANA

Cet objet permet d'afficher et de modifier l'état de variables numériques sous forme analogique: potentiomètre, cadran, barre de défilement, etc ...

ECRANCLA

Cet objet permet de réaliser différents types de fenêtres d'affichages de message et de terminal (écran + clavier).

ARCHIVE

Cet objet permet de mémoriser l'états de variables dans le temps et d'afficher les valeurs sous forme de table ou de courbe.

OBJET

Objet standard qui possède de très nombreuses propriétés paramétrables. Il peut être utilisé lorsque les autres types d'objets ne conviennent pas, notamment pour créer des parties opératives virtuelles.

SON

Cet objet permet d'émettre des sons (fichier « .WAV » de WINDOWS) en liaison avec des variables de l'application d'automatisme.

DIAL

Objet permettant de créer des boîtes de dialogue en liaison avec l'application d'automatisme.

PROG

Cet objet permet d'effectuer des traitements mathématiques ou algorithmiques en utilisant un langage littéral.

CHAPITRE N°9.

Accès à l'état des variables

Pour accéder à l'état des variables de l'application d'automatisme il suffit de donner leur nom ou le nom d'un symbole.

Un état booléen peut être calculé à partir d'une variable numérique en utilisant une comparaison avec une constante.

La syntaxe / <nom de variable ou symbole> permet de lire l'état complémenté d'une variable booléenne.

Exemples :

M200

I0

/I2

niveau haut

M210=4

M480<2

M260&32

CHAPITRE N°10.

Ordres

Les ordres prédéfinis peuvent être utilisés dans les rubriques « ACTIONS » des objets « BPVOYANT » et « OBJET », et dans la fonction « CMD(...) » de l'objet « PROG ».

Modification de l'état d'une variable

La syntaxe à utiliser est :

<nom de variable ou symbole>=valeur

Pour les variables booléenne la syntaxe <nom de variable ou symbole> = / inverse l'état de la variable.

Exemples :

m200=4

o0=1

o2=0

o10=/

resultat=100

RUN

Valide le mode RUN pour l'application d'automatisme.

STOP

Valide le mode STOP pour l'application d'automatisme.

INIT

Initialise l'application d'automatisme.

STEP

Effectue un pas pour l'application d'automatisme.

OPENAOF(nom d'un objet)

Ouvre un objet AOF, l'objet qui émet cet ordre reste ouvert.

CHAINAOF(nom d'un objet)

Ouvre un objet AOF et ferme l'objet qui émet cet ordre.

MINIMIZEAOF(titre d'un pupitre)

Met en icône un pupitre, le paramètre est le texte de la barre de titre du pupitre.

RESTOREAOF(titre d'un pupitre)

Agrandit un pupitre mis en icône, le paramètre est le texte de la barre de titre du pupitre.

CLOSEAOF()

Ferme l'objet qui émet cet ordre.

Echanges entre objets**PARENTPARAM(paramètre {+n} {-n})**

Permet à un objet enfant d'accéder à un paramètre du pupitre parent. Le paramètre doit être défini dans la rubrique « LIENS : DONNEES POUR LES OBJETS ENFANTS » du pupitre parent. Voir l'objet « PUPITRE ».

SISTERPARAM(identificateur , paramètre)

Utilisé par l'objet OBJET, cette syntaxe permet de lire une valeur propre à un objet. Voir l'objet « OBJET ».

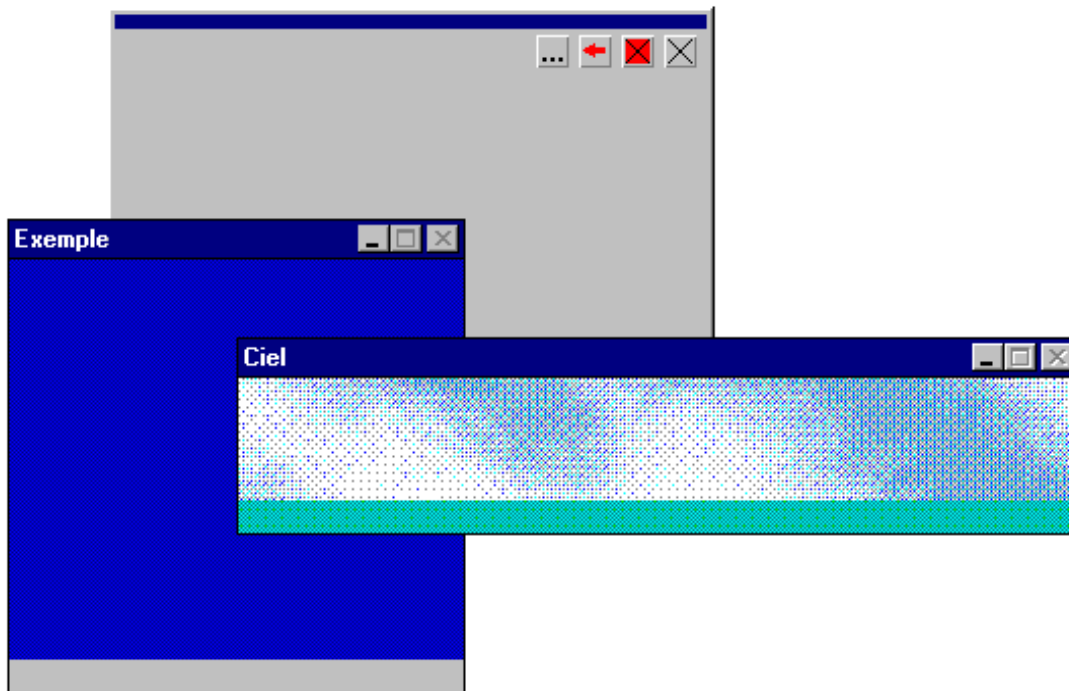
SETPARAM(identificateur , paramètre , valeur)

Permet de modifier le paramètre d'un objet (version 32 bits seulement).


Pour accéder à la liste des paramètres modifiables, cliquez sur le bouton droit de la souris pendant l'édition des zones actions d'un objet BPVOYANT, puis choisissez la commande « Paramètres ».

CHAPITRE N°11.

L'objet PUPITRE



L'objet PUPITRE est utilisé pour grouper d'autres objets, c'est le seul objet qui possède cette caractéristique. Les objets posés sur un pupitre deviennent solidaires et peuvent être sauvegardés ensemble pour constituer un nouvel objet.

L'objet PUPITRE possède un menu accessible en cliquant sur l'icône .

L'objet PUPITRE permet une gestion simplifiée des objets, il est en effet possible d'aligner les objets, de les dupliquer, de les copier, de les coller de façon unitaire ou en groupe (sélection de plusieurs objets). L'objet pupitre permet d'examiner les objets enfants sous la forme d'une liste laissant apparaître pour chaque objet : son nom, sa position, sa taille, son identificateur.

Les objets PUPITRE peuvent être imbriqués (objet PUPITRE dans un objet PUPITRE).

Les objets pupitres peuvent représenter les différentes pages d'une application de supervision.

Ces différentes pages pourront être au choix, visualisées en même temps, fermées ou mises en icône par l'utilisateur ou par des ordres prédéfinis.

Terminologie

Objet enfant ou fenêtre fille : un objet qui appartient à un pupitre.

Objet parent ou fenêtre parent : un pupitre à qui appartient un objet.

Objet frère ou fenêtre sœur : un objet qui appartient au même pupitre.

Autres caractéristiques

L'objet pupitre peut afficher les messages d'aide provenant des enfants qu'il contient.

CHAPITRE N°12.

Les propriétés de l'objet PUPITRE - onglet « Aspect »

Fenêtre

Permet de définir l'aspect de la fenêtre composant le pupitre : présence d'une bordure, d'une barre de titre (dans ce cas un titre peut être précisé), présence des icônes de fermeture et de réduction. La case à cocher « Affichage des messages d'aides » permet de définir une zone de message en bas de la fenêtre, la taille de cette zone est déterminée automatiquement en fonction de la fonte choisie (voir plus loin). Si une telle zone n'est pas définie, alors les messages provenant des enfants seront affichés soit sur le pupitre parent du pupitre, soit sur la fenêtre de l'environnement AUTOMGEN ou IRIS.

Fond du pupitre

Détermine l'aspect du fond du pupitre : coloré (voir plus loin), transparent (accessible seulement si le pupitre est l'enfant d'un autre pupitre), bitmap (le fond est défini par un fichier « .BMP » créé avec PAINTBRUSH par exemple).

Couleurs

Permet de choisir la couleur pour le fond du pupitre (si un fond coloré est sélectionné - voir plus haut), le fond et les caractères de la zone d'affichage des messages d'aides (si cette zone est validée - voir plus haut).

Fonte de caractères pour l'aide

Détermine la fonte utilisée pour afficher les messages d'aide en bas du pupitre.

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Textes

Le texte d'aide et le texte bulle.

Les propriétés de l'objet PUPITRE - onglet « Bitmap »

Bitmap

Si le fond du pupitre contient un bitmap (voir onglet « Aspect ») la zone d'édition doit contenir un nom d'accès complet vers un fichier « .BMP » (les formats 16 couleurs, 256 couleurs et 24 bits sont supportés).

Les boutons poussoirs « PARCOURIR » et « EDITER » permettent respectivement de rechercher un fichier « .BMP » et d'éditer un fichier avec le logiciel PAINTBRUSH de WINDOWS.

Les propriétés de l'objet PUPITRE - onglet « Liens »

Données pour les objets enfants

Cette zone d'édition permet de définir des paramètres auxquels pourront accéder les objets enfants avec le mot clé « PARENTPARAM ». Il faut écrire une définition par ligne. Chaque définition doit respecter la syntaxe suivante : « PARAMETRE=VALEUR ».

Les propriétés de l'objet PUPITRE - onglet « Options »

Grille

Permet de définir une grille (invisible) pour le positionnement des objets. Seule la commande « Déplacer » du menu intégré au pupitre utilise la grille. Les valeurs pour la grille sont exprimées en nombre de pixels. Les valeurs 0 et 1 annulent l'effet de la grille. Cette fonction doit être utilisée pour aligner parfaitement les objets.

Rafraîchissement des enfants

La case à cocher « Continuer à rafraîchir les enfants ... » détermine si les enfants doivent continuer à être mis à jour lorsque le pupitre est mis en icône. Cette option permet, lorsqu'elle n'est pas sélectionnée, d'augmenter les performances du système lorsqu'un pupitre mis en icône ne contient que des éléments visuels.

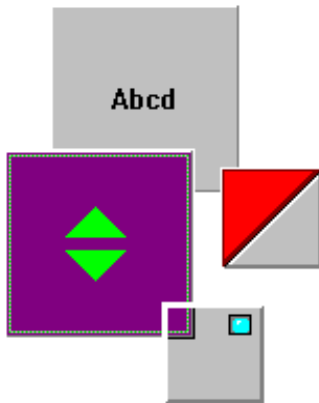
Les propriétés de l'objet PUPITRE - onglet « Enfants »

Enfants

Cette rubrique contient la liste des objets enfants du pupitre. Le bouton poussoir « Propriétés » permet d'ouvrir directement la boîte de dialogue de propriété de l'enfant sélectionné dans la liste. Le bouton poussoir « Détruire » élimine l'objet sélectionné. Les zones d'édition « Positions » permettent de régler la position des objets.

CHAPITRE N°13.

L'objet BPVOYANT



L'objet BPVOYANT est utilisé pour créer des boutons poussoirs, des voyants ou une combinaison des deux : bouton poussoir à voyant intégré.

Deux actions distinctes peuvent être effectuées lorsque le bouton poussoir est enfoncé et lorsqu'il est relâché.

L'aspect de l'objet bouton poussoir peut être simple, un bouton poussoir coloré par exemple, ou plus élaboré : dans ce cas un fichier bitmap sera utilisé pour stocker le dessin du bouton poussoir.

Un texte d'aide et un texte bulle peuvent être associés au bouton poussoir.

Les boutons poussoirs peuvent être bistables (ils restent enclenchés) ou monostables (ils ne restent pas enclenchés).

CHAPITRE N°14.

Les propriétés de l'objet BPVOYANT - onglet « Aspect »

Type de l'objet

Permet de choisir la nature de l'objet : bouton poussoir, voyant ou bouton poussoir intégrant un voyant.

Couleurs

Permet de choisir les couleurs de l'objet. Si l'objet est un bouton poussoir, le réglage « fond éteint » représente la couleur du bouton poussoir. Si l'objet est un voyant ou un bouton poussoir intégrant un voyant le réglage « fond allumé » détermine la couleur du fond lorsque le voyant est allumé et « fond éteint » la couleur lorsque le voyant est éteint. Seule la couleur des caractères est réglable si l'aspect de l'objet est déterminé par un bitmap.

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet. Ceci est nécessaire si l'aspect de l'objet est déterminé par un bitmap.

Fonte de caractères

Permet de choisir la fonte et la taille des caractères. Le fichier de fonte utilisé devra être présent sur le PC ou s'exécute le programme.

Texte

Permet de spécifier le texte affiché sur l'objet, sa position, son orientation ainsi qu'un texte d'aide qui apparaît lorsque le bouton est pressé et un texte bulle qui apparaît lorsque le curseur est placé sur l'objet.

Les propriétés de l'objet BPVOYANT - onglet « Liens »

Action lorsque ...

Permet de définir les actions à réaliser lorsque le bouton est enfoncé et lorsqu'il est relâché.

Une action peut être le forçage de l'état d'une variable, par exemple :

`00=1, m200=4, _depart cycle_=3`

Ou un mot clé prédéfini.

Exemple de configuration pour que l'entrée i10 reflète l'état du bouton poussoir (i10 à 0 si le bouton est relâché, i10 à 1 si le bouton est enfoncé) :

Action lorsque le bouton est enfoncé : i10=1

Action lorsque le bouton est relâché : i10=0

Etat du voyant

Détermine l'état du voyant. Cette rubrique doit contenir le nom d'une variable qui pilotera le voyant : 0 = voyant éteint, 1 = voyant allumé.

Par exemple :

```
b31, o4, _voyant init_, m200=50, m400<8, m500&16
```

Identificateur

Permet de référencer l'objet par rapport aux autres objets.

Condition de désactivation

Permet de désactiver le voyant. Si cette rubrique contient le nom d'une variable, alors cette variable désactive l'objet si elle est vraie.

Les propriétés de l'objet BPVOYANT - onglet « Options »

Type du bouton poussoir

Détermine si le bouton poussoir est bistable (il reste enfoncé) ou monostable.

Clavier

Permet d'associer une touche au bouton poussoir. Si cette touche ou cette combinaison de touche est présente au clavier alors le bouton poussoir sera enfoncé.

Pour préciser le code de la touche différentes syntaxes sont utilisables :

- un simple caractère: par exemple A, Z, 2,
- le caractère \$ suivi de du code de la touche en hexadécimal,
- le nom d'une touche de fonction, par exemple F5.

Pour les combinaisons de touches il faut ajouter au début CTRL+ ou SHIFT+.

Par exemple : CTRL+F4 ou SHIFT+Z.

Bitmap

Permet de spécifier un fichier bitmap qui contiendra le dessin de l'objet. Le fichier bitmap contient les quatre aspects possibles de l'objet : bouton relâché voyant éteint, bouton enfoncé voyant éteint, bouton relâché voyant allumé et bouton enfoncé voyant allumé.

Même si le fichier est un bouton poussoir sans voyant ou un voyant il y a toujours quatre aspects dont seuls deux sont utilisés.

Le fichier bitmap est découpé en quatre horizontalement.

Exemple :



Le sous répertoire d'exemple « EXEMPAOF » contient des fichiers « .BMP » pour créer des boutons poussoirs d'aspect élaboré (boutons poussoirs avec voyants) :

- BPLEDBLEU (bouton poussoir et voyant bleu),
- BPLEDROU (bouton poussoir à voyant rouge),
- BPLEDVER (bouton poussoir à voyant vert),
- BPLEDJEU (bouton poussoir à voyant jaune),
- TRIBLEU (bouton poussoir triangulaire à voyant bleu),
- TRIROUGE (bouton poussoir triangulaire à voyant rouge),
- TRIVERT (bouton poussoir triangulaire à voyant vert),
- TRIJAUNE (bouton poussoir triangulaire à voyant jaune).

Ces fichiers ont été créés pour des boutons poussoirs de dimensions 50 x 50 pixels.

CHAPITRE N°15.

L'objet VALDIGI

+1234
-1.0032
Poids : 003,5678Kg

L'objet VALDIGI est utilisé pour afficher et modifier l'état de variables numériques sous forme digitale.

Un texte peut être affiché avant et après la valeur numérique : explication, désignation, unité de mesure, etc...

Les types de variables suivants peuvent être affichés :

- variables booléennes : entrées, sorties, bits, étapes,
- entiers 16 bits : mots, compteurs, compteur et consigne de temporisation,
- entiers 32 bits : longs,
- réels : flottants.

Les variables numériques entières peuvent être affichées en base 10 ou en base 16.

Lorsque l'objet VALDIGI est posé sur un pupitre son fond peut être rendu transparent pour laisser apparaître le fond du pupitre.

Un texte bulle peut être associé à l'objet.

CHAPITRE N°16.

Les propriétés de l'objet VALDIGI - onglet « Aspect »

Format

Permet de définir le type d'affichage :

- Toujours montrer le signe : affiche le signe '+' pour les valeurs signées positives,
- Valeur signée : définit le mode signé ou non signé pour les entiers 16 ou 32 bits (base 10 uniquement),
- Afficher tous les digits : affiche des 0 au début de la valeur si nécessaire.

Base

Détermine la base d'affichage pour les entiers 16 et 32 bits.

Couleurs

Permet de choisir les couleurs du fond de l'objet (s'il n'est pas transparent) et des caractères.

Fonte de caractères

Permet de choisir la fonte et la taille des caractères. Le fichier de fonte utilisé devra être présent sur le PC ou s'exécute le programme.

Nombre de digits

Définit la longueur de la partie entière et décimale.

Fond

Permet de choisir entre un fond coloré ou transparent (si l'objet est posé sur un pupitre seulement).

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Les propriétés de l'objet VALDIGI - onglet « Textes »

Texte Bulle

Texte affiché dans une bulle lorsque l'utilisateur déplace le curseur sur l'objet.

Texte affiché avant et après la valeur

Permet d'afficher des informations à gauche et à droite de la valeur numérique.

Les propriétés de l'objet VALDIGI - onglet « Liens »

Variable ou symbole

Désigne la variable à afficher. Pour accéder au compteur ou à la consigne d'une temporisation il faut utiliser la syntaxe suivante :

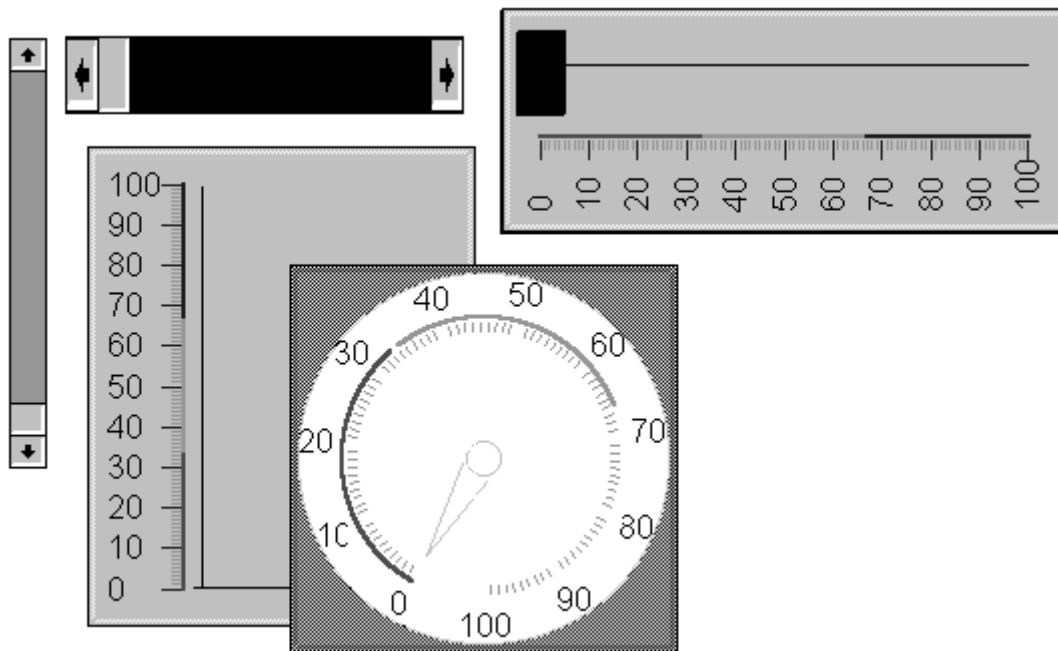
- pour le compteur : COUNT(temporisation), exemple : COUNT(t3),
- pour la consigne : PRED(temporisation), exemple : PRED(t7),

L'état de la variable est modifiable ...

Si cette case est cochée alors l'utilisateur peut modifier l'état de la variable en cliquant sur l'objet.

CHAPITRE N°17.

L'objet VALANA



L'objet VALANA est utilisé pour afficher et modifier l'état de variables numériques sous forme analogique.

Les types de variables suivants peuvent être affichés :

- entiers 16 bits : mots ou compteurs.

Les types d'affichage suivants sont disponibles :

- barre de scrolling de WINDOWS (horizontale ou verticale),
- barre graphe pour valeur signée (horizontal ou vertical),
- barre graphe pour valeur non signée (horizontal ou vertical),
- cadran,
- potentiomètre linéaire (horizontal ou vertical).

Les barres de scrolling, les barres graphes et les potentiomètres linéaires peuvent être utilisés pour piloter l'état d'une variable.

Les barres graphes, les cadrans et les potentiomètres linéaires peuvent recevoir des graduations.

Un texte bulle peut être associés à l'objet.

CHAPITRE N°18.

Les propriétés de l'objet VALANA - onglet « Aspect »

Objets

Permet de définir le type d'affichage.

Orientation

Détermine l'orientation : horizontale ou verticale.

Couleurs

Permet de choisir les couleurs du fond de l'objet (s'il n'est pas transparent) et de l'objet.

Fond

Permet de choisir entre un fond coloré ou transparent (si l'objet est posé sur un pupitre seulement).

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Textes

Le texte bulle.

Les propriétés de l'objet VALANA - onglet « Liens »

Variable ou symbole

Désigne la variable liée à l'objet (un mot ou un compteur).

Les actions de l'utilisateur ...

Détermine si l'état de la variable peut être modifié par l'utilisateur.

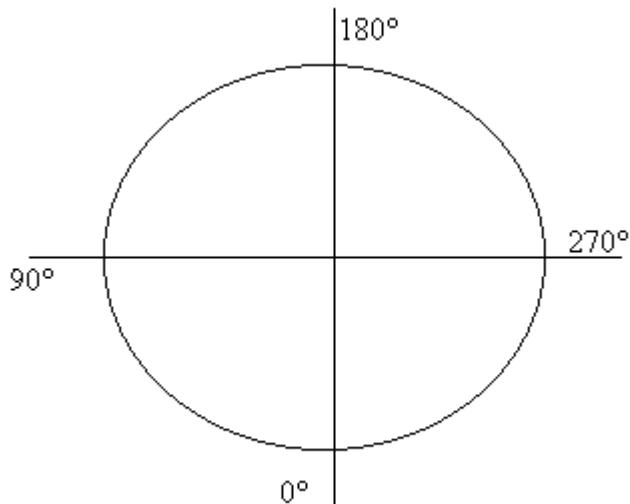
Les propriétés de l'objet VALANA - onglet « Bornes »

Minimum, maximum

Valeurs minimales et maximales.

Angle de départ, angle de fin

Pour l'affichage de type cadran détermine l'angle de départ et l'angle d'arrivée. Les valeurs sont précisées en degrés :



Les propriétés de l'objet VALANA - onglet « Graduations »

Utiliser les graduations

Valide ou invalide l'utilisation des graduations

Valeur de départ, valeur de fin

Valeurs affichées pour les graduations, ces valeurs peuvent être des nombres signés et/ou des nombres à virgule.

Pas des petites graduations, pas des grandes graduations

Pas des graduations (deux niveaux) par rapport aux valeurs de départ et de fin. Ces valeurs peuvent être des nombres à virgule.

Police de caractère

Définit les caractères utilisés pour les graduations.

Zone N°1, zone N°2 et zone N°3

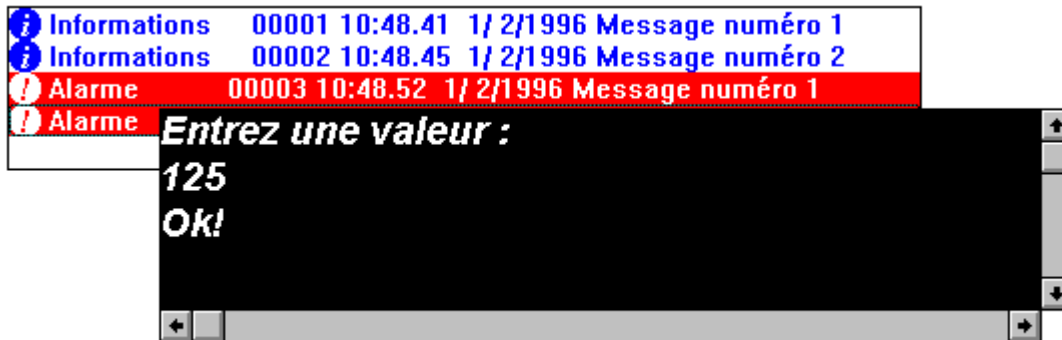
Permet de définir des zones colorées. « Valeur de départ » et « Valeur de fin » définissent chaque zone. La couleur pour chaque zone est spécifiée par trois composantes le rouge, le vert et le bleu comprises entre 0 et 255.

Couleurs

Détermine la couleur des caractères et des graduations. Les couleurs sont ici aussi exprimées par leurs trois composantes : le rouge, le vert et le bleu.

CHAPITRE N°19.

L'objet ECRANCLA



L'objet ECRANCLA est utilisé pour afficher des messages, saisir des textes, afficher des alarmes.

Les types d'objets disponibles pour ECRANCLA sont :

- « Ecran seulement » : l'objet permet d'afficher des messages,
- « Ecran et clavier sans écho » : l'objet permet d'afficher des messages et de recevoir des caractères entrés par l'utilisateur. Les caractères entrés au clavier ne sont pas affichés,
- « Ecran et clavier avec écho » : idem mais les caractères entrés au clavier sont affichés,
- « Boîte à message, liste d'alarmes » : permet d'obtenir une liste de messages structurée avec possibilité de définir des classes de messages (une classe de message définit la couleur des caractères utilisés, un nom de classe et une icône). Les messages peuvent être datés.

Les textes affichés peuvent être envoyés caractère par caractère dans l'objet ECRANCLA ou des messages peuvent être enregistrés dans l'objet et affichés en fournissant leurs numéros.

Liens avec l'application

Le lien entre l'objet ECRANCLA et l'application se fait par des tables de mots.

Pour envoyer des données au type d'objet « Ecran » (avec ou sans clavier) il faut placer les données à partir du deuxième mot de la table de réception puis la longueur des données dans le premier mot de la table (la longueur maximale est 255). Chaque mot contient une donnée.

Les données peuvent être : un caractère ASCII, un numéro de message prédéfini + 8000 hexa, ou une commande spéciale : 100 hexa efface la fenêtre, 200 hexa affiche la date, 300 hexa affiche l'heure, 400 affiche le numéro du message.

Lorsque l'objet a relu les données disponibles dans une table il remet le premier mot à 0 pour signaler que l'opération a été effectuée.

Pour les objets « avec clavier » le principe est le même : le premier mot de la table d'émission contient le nombre de caractères entrés au clavier, les mots suivants contiennent les caractères (un par mot). L'application doit remettre à 0 le premier mot lorsqu'elle a utilisé les données.

Pour l'objet «Boîte à messages, liste d'alarmes » la table d'échange à une longueur fixe de 10 mots. Comme pour le type « Ecran » le premier mot déclenche l'affichage de message. S'il est différent de 0 il désigne un numéro de message à afficher. Seuls des messages enregistrés peuvent être affichés. Le premier mot peut également prendre la valeur ffff hexa pour vider la boîte à messages.

Description des 10 mots utilisés pour les échanges avec le type « Boîte à messages » :

Mot 0 représente le premier mot de la table, Mot 1 le deuxième, etc ...

Mot 0 : numéro de message à afficher ou 0 si pas de message ou ffff hexa pour tout effacer,

Mot 1 : numéro de classe pour le message (voir plus loin les explications sur les classes de messages),

Les mots qui suivent déterminent la date et l'heure qui peuvent être affichées pour chaque message. Une valeur égale à ffff hexa demande à l'objet d'utiliser la date et l'heure courante de l'ordinateur (cela ne concerne pas les millisecondes).

Mot 2 : jour

Mot 3 : mois

Mot 4 : année

Mot 5 : heures

Mot 6 : minutes

Mot 7 : secondes



Mot 8 : millisecondes

Mot 9 : réservé (mettre 0)

Les classes de messages

Les classes de messages permettent de classer les messages en famille qui partageront les caractéristiques suivantes : la couleur du fond, la couleur des caractères et une icône.

Il existe deux classes prédéfinies :

- la classe message d'information : caractères bleus sur fond blanc, icône , elle porte le numéro -1,
- la classe message d'alarme : caractères blancs sur fond rouge, icône , elle porte le numéro -2.

D'autres classes peuvent être définies par l'utilisateur.

Un texte bulle peut être associé à l'objet.

CHAPITRE N°20.

Les propriétés de l'objet ECRANCLA - onglet « Aspect »

Type de l'objet

Permet de définir un des types d'objet. Voir le chapitre précédent.

Couleurs

Permet de choisir les couleurs du fond de l'objet et des caractères.

Fonte

Permet de choisir la fonte de caractères utilisée pour afficher les textes.

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Textes

Le texte bulle.

Les propriétés de l'objet ECRANCLA - onglet « Liens »

Réception, émission

Définissent les premières variables de la table de réception et d'émission. Ces zones peuvent contenir un nom de variable ou un symbole.

Les propriétés de l'objet ECRANCLA - onglet « Liste »

Ces rubriques ne concernent que les objets de type « Boîte à messages ».

Icônes

Cette case à cocher détermine s'il faut afficher une icône avant les messages.

Classes

Cette case à cocher détermine s'il faut afficher le numéro de la classe du message.

Jours, Mois, Années, Heures, Minutes, Secondes, 1/1000 secondes

Ces cases à cocher déterminent s'il faut afficher chacun de ces éléments.

Messages

Cette case à cocher détermine s'il faut afficher un message.

Numéros

Cette case à cocher détermine s'il faut afficher le numéro d'apparition des messages.

Classes de messages

Cette zone d'édition permet de définir de nouvelles classes de messages. Chaque ligne définit une classe. Sur chaque ligne on doit trouver dans l'ordre et séparés par des virgules : la couleur du fond, (trois composantes rouge, vert et bleu) la couleur des caractères (trois composantes rouge, vert et bleu), le nom de la classe, le nom d'un fichier bitmap pour l'icône associé à la classe.

Par exemple :

255,0,0,0,0,0,ALARME,alarme.bmp

Signifie :

Couleur du fond rouge, couleur des caractères noir, nom de la classe ALARME, fichier contenant l'icône : « alarme.bmp ».

Les propriétés de l'objet ECRANCLA - onglet « Options »

Afficher les codes hexadécimaux des caractères

Cette option permet d'afficher le code hexadécimal pour chaque caractère à la place de sa représentation ASCII. Elle est utilisable pour les objets de type « Ecran ... » et permet notamment la mise au point des programmes.

Barre de défilement horizontale, verticale

Montre ou cache les barres de défilements.

Convertir les caractères OEM -> ANSI

Si cette case est cochée, alors les caractères provenant de l'application d'automatisme sont automatiquement convertis du jeu de caractères OEM (MS-DOS) au jeu de caractères ANSI

(WINDOWS). La conversion inverse est appliquée aux caractères allant de l'objet à l'application d'automatisme.

Dupliquer les messages vers ...

Cette rubrique peut recevoir un nom de fichier ou de périphérique (« LPT1 » par exemple pour l'imprimante). Il est possible de spécifier plusieurs fichiers et/ou périphériques en les séparant par une virgule. Les affichages seront automatiquement dupliqués : impression dite « au fil de l'eau ».

Associer un fichier de stockage des messages ...

Permet de définir un fichier qui sera associé à l'objet et utilisé pour le stockage des informations. Si un tel fichier existe alors les messages sont sauvegardés (à concurrence du nombre défini sous la rubrique « nombre de lignes mémorisées », si ce nombre est atteint les données les plus anciennes sont effacées). Lorsque l'objet est ouvert, et si un fichier de stockage existait lors de sa dernière utilisation, alors les données contenues dans le fichier sont transférées dans l'objet.

Ecrire les messages périmés vers ...

Permet de définir un fichier ou un périphérique qui recevra les messages périmés (les messages qui sont éliminés du fichier de stockage pour libérer de la place).

Nombre de lignes mémorisées ...

Détermine la capacité du fichier de stockage des messages en nombre de lignes. La valeur 0 attribue le maximum d'espace mémoire utilisable (pas de limite fixe).

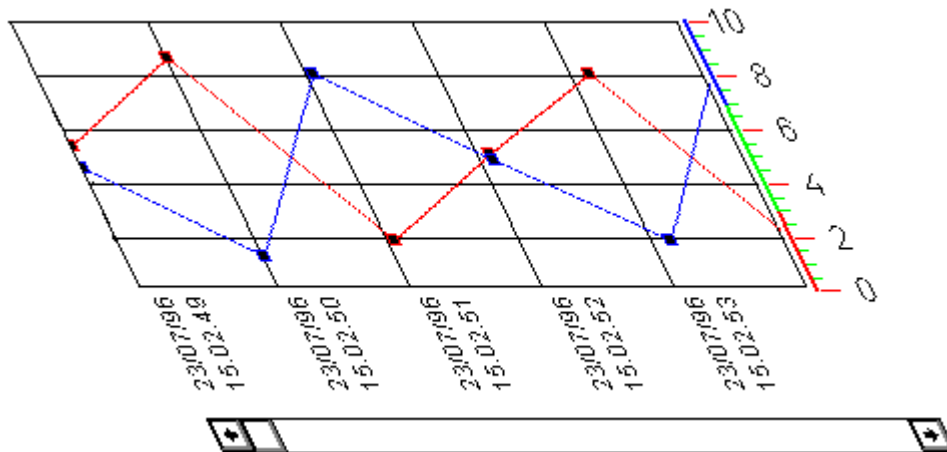
Les propriétés de l'objet ECRANCLA - onglet « Messages »

Messages prédéfinis

Cette boîte d'édition permet de documenter les messages prédéfinies (un par ligne).

CHAPITRE N°21.

L'objet ARCHIVE



L'objet ARCHIVE est utilisé pour enregistrer l'état d'une ou plusieurs variables dans le temps et représenter graphiquement cet état (valeurs numériques ou tracés de courbes).

L'objet ARCHIVE peut lire et archiver les types de variables suivants: mots, compteurs, bits, sorties, entrées, longs, flottants.

La représentation se fait à l'écran sous la forme de table ou sous la forme de tracés de courbes.

Suivant la représentation utilisée (TABLE ou COURBE), des options sont disponibles : nom de colonnes, couleurs, affichage du temps, utilisation de graduations, utilisation d'une grille, police de caractères etc.

Les acquisitions sont réalisées à intervalles réguliers (mode périodique) ou à la demande (mode déclenché).

Les variables sont lues et stockées en mémoire pour constituer une base de données .Cette base de données pourra être sauvegardée dans le FICHER IMAGE et être réutilisée lors d'une prochaine exécution de l'objet ARCHIVE .

Si un FICHER d'ARCHIVAGE est défini, chaque acquisition est enregistrée dans ce fichier au format base de données texte standard avec les nom de colonnes sur la première ligne et les données sur les lignes suivantes séparées par un « ; ».

CHAPITRE N°22.

Les propriétés de l'objet ARCHIVE - onglet « Aspect »

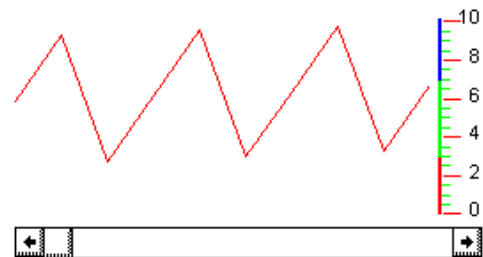
Objets

Permet de définir le type d'affichage.

L'objet peut être représenté sous la forme d'une table (figure 1.1) ou sous la forme de tracés de courbes (figure 1.2).

| Date | Heure d'acquisition | Valeur |
|----------|---------------------|--------|
| 23/07/96 | 16.52.52.443 | -28043 |
| 23/07/96 | 16.52.53.541 | -6059 |
| 23/07/96 | 16.52.54.640 | 16477 |
| 23/07/96 | 16.52.55.738 | -26575 |
| 23/07/96 | 16.52.56.837 | -4091 |
| 23/07/96 | 16.52.57.935 | 18441 |
| 23/07/96 | 16.52.59.034 | -24579 |
| 23/07/96 | 16.53.00.132 | -2067 |

(figure 1.1)



(figure 1.2)

Couleurs

Permet de choisir la couleur de la police de caractère lorsque l'objet est sous la forme d'une table ainsi que la couleur de marquage des valeurs sur les courbes .

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Texte

Le texte bulle associé à l'objet.

Les propriétés de l'objet ARCHIVE - onglet « Données »

Première variable à lire

Permet de sélectionner la première variable qu'il faudra archiver.

Nombre de variables à lire

Signale à l'objet ARCHIVE le nombre de variables consécutives à la « Première variable à lire » qu'il faudra archiver .

Nombre d'enregistrements mémorisés

Permet de dimensionner la base de données en mémoire .

Un enregistrement représente une acquisition de « n » variables (« n » étant le nombre de variables à lire).

Lecture périodique

L'acquisition des variables se fera à intervalle fixe des l'exécution de l'objet ARCHIVE

Lecture déclenchée

Une acquisition des variables sera effectuée dès que le « Mot de contrôle » en aura donné l'ordre .

Période

Permet de fixer le temps qui sépare deux acquisitions. Le temps est au format Jour(s) / Heure(s) /Minute(s) /Seconde(s) /Milliseconde(s) :

J pour les heures

H pour les heures

M pour les minutes

S pour les secondes

MS pour mes millisecondes

Ex : 2J

Ex : 2H10M15S

Contrôle

Permet de définir une variable (un mot) qui contrôlera l'objet ARCHIVE . Dès la valeur prise en compte, son contenu est forcé à 0 par l'objet ARCHIVE.

Valeur

Action

0

Rien

1

Déclenchement d'une acquisition (Lecture déclenchée)

2

Geler les acquisitions

- 3 Reprendre l'archivage (après avoir gelé)
- 4 Vider la base de données en mémoire
- 5 Détruire le fichier d'archivage
- 6 Activer le mode « Suivi des dernières acquisitions »
- 7 Annuler le mode « Suivi des dernières acquisitions »

Les propriétés de l'objet ARCHIVE - onglet « Options »

Utiliser le fichier image

Le fichier image est utilisé :

En fin d'utilisation de l'objet ARCHIVE, pour sauvegarder la base de données présente en mémoire.

Au lancement de l'objet ARCHIVE, pour reconstituer la base de données qui était présente en mémoire lors de la dernière utilisation .

Utiliser le fichier archive

Chaque acquisition est enregistrée dans ce fichier au format base de données standard.

Affichage

Date d'acquisition: Permet d'afficher la date d'acquisition d'un enregistrement.

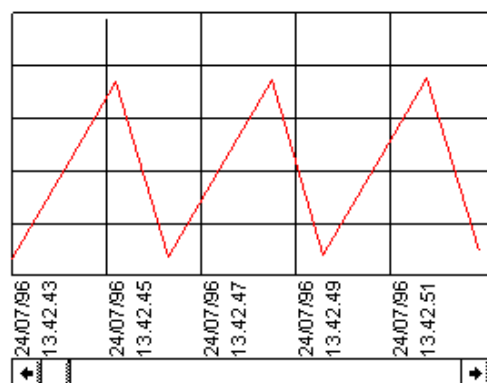
Heure d'acquisition: Permet d'afficher l'heure d'acquisition d'un enregistrement.

Heures, minutes, secondes, millisecondes: Permet de configurer l'affichage de l'heure d'acquisition.

L'affichage du temps se fait en aval de l'affichage des acquisitions pour l'objet TABLE (*figure 3.1*) ou sous la grille lorsqu'elle est définie pour l'objet COURBE (*figure 3.2*)

| Date | Heure d'acquisition | Valeur |
|----------|---------------------|--------|
| 24/07/96 | 13.42.43.112 | 3141 |
| 24/07/96 | 13.42.44.211 | 25217 |
| 24/07/96 | 13.42.45.309 | -18451 |
| 24/07/96 | 13.42.46.408 | 3489 |
| 24/07/96 | 13.42.47.506 | 25525 |
| 24/07/96 | 13.42.48.605 | -17931 |
| 24/07/96 | 13.42.49.703 | 4085 |
| 24/07/96 | 13.42.50.802 | 26149 |
| 24/07/96 | 13.42.51.900 | -17375 |
| 24/07/96 | 13.42.52.999 | 4709 |

(figure 3.1)



(figure 3.2)

Les propriétés de l'objet ARCHIVE - onglet « Tables »

Police de caractères

Permet de sélectionner une police de caractères pour l'affichage du nom des colonnes, du temps et la valeur des acquisitions.

Nom des colonnes

Permet de définir le nom des colonnes pour l'objet TABLE ainsi que le format d'affichage de ces colonnes (*figure 4.1*).

syntaxe : nom, format

format *

pas de format spécifié

h

d

ns

s

nv

v

Affichage

Signé, décimal, visible

Hexadécimal

Décimal

Non signé

Signé

Non visible

Visible

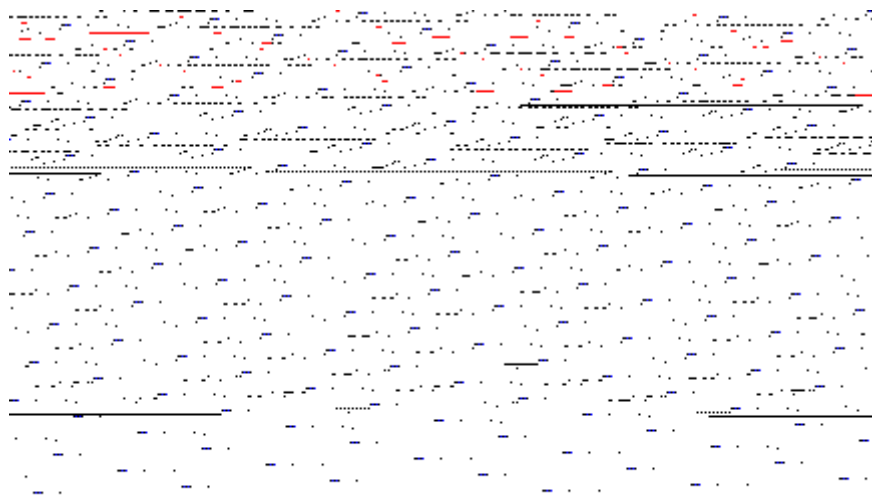
* On peut combiner les différentes options, par exemple:

Format

d,ns,v

Affichage

Décimal sans signe visible



(*figure 4.1*)

Les propriétés de l'objet ARCHIVE - onglet « Courbe »

Valeur minimale, maximale

Permet de sélectionner la valeur minimale et maximale pour l'affichage des courbes.

Seules les valeurs comprises entre la valeur minimale et la valeur maximale seront visibles sur l'écran.

Visualiser

Permet de définir le temps de visualisation .Celui-ci est communiqué à l'objet ARCHIVE suivant le format Jour(s)/Heure(s)/Minute(s)/Seconde(s)/Milliseconde(s) :

J pour les jours

H pour les heures

M pour les minutes

S pour les secondes

MS pour mes millisecondes

Ex : **Visualiser** 2H30M10S

Ex : **Visualiser** 100MS

Marquage des valeurs sur la courbe

Permet de faire une marque sur la courbe pour chaque acquisition (*figure 5.1*).

Affichage du temps

Permet d'afficher la date et l'heure d'acquisition d'une ou plusieurs variables sous la grille si elle est active. On peut définir la couleur et la police de caractères de l'affichage du temps.

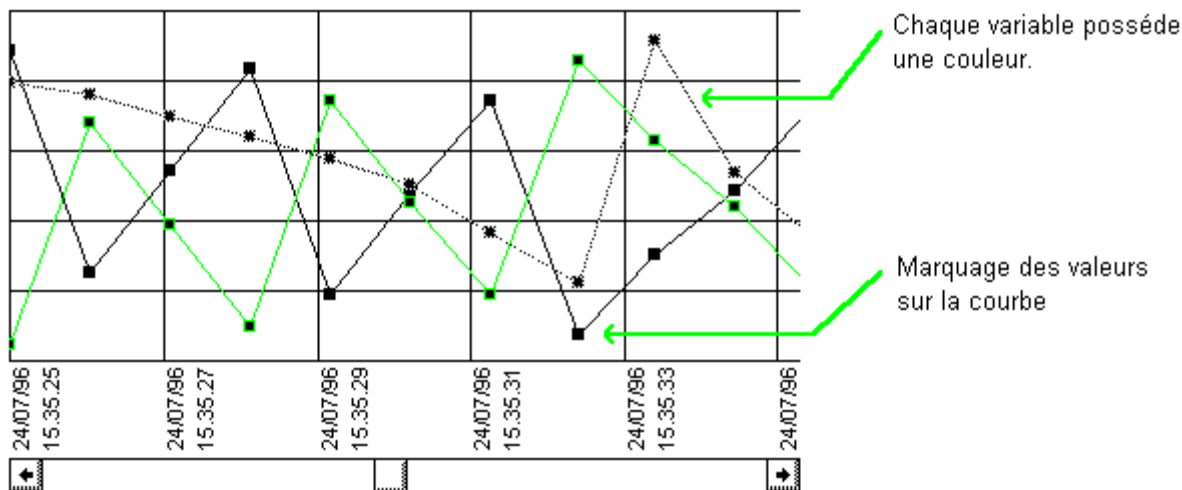
Couleurs des tracés

Permet de définir une couleur pour chaque courbe. La première courbe possède la couleur de la première ligne, la deuxième courbe possède la couleur de la deuxième ligne etc...

Les couleurs sont au format Rouge, Vert, Bleu.

Ex: 255,0,0 tracé rouge

Si une couleur n'est pas définie sur une ligne, la courbe correspondante à cette ligne ne sera pas tracée.



(figure 5.1)

Les propriétés de l'objet ARCHIVE - onglet « Graduations »

Utiliser les graduations

Valide ou invalide l'utilisation des graduations (figure 6.1).

Valeur de départ, valeur de fin

Valeurs affichées pour les graduations, ces valeurs peuvent être des nombres signés et/ou des nombres à virgule.

Pas des petites graduations, pas des grandes graduations

Pas des graduations (deux niveaux) par rapport aux valeurs de départ et de fin. Ces valeurs peuvent être des nombres à virgule.

Police de caractère

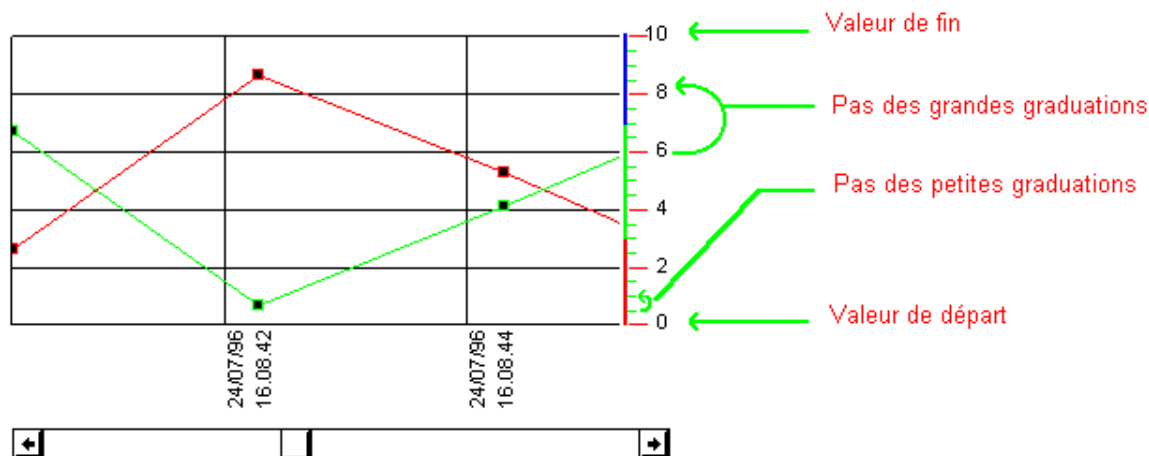
Définit les caractères utilisés pour les graduations.

Zone N°1, zone N°2 et zone N°3

Permet de définir des zones colorées. "Valeur de départ" et "Valeur de fin" définissent chaque zone. La couleur pour chaque zone est définie par trois composantes le rouge, le vert et le bleu comprises entre 0 et 255.

Couleurs

Détermine la couleur des caractères et des graduations. Les couleurs sont ici aussi exprimées par leur trois composante : le rouge, le vert et le bleu.



(figure 6.1)

Les propriétés de l'objet ARCHIVE - onglet « Grille »

Afficher la grille

Valide ou invalide l'affichage de la grille.

Pas pour les ordonnées

Définit le pas vertical de la grille.

Pas pour les abscisses

Définit le pas horizontal de la grille. Le pas est au format Jour(s)/ Heure(s)/ Minute(s)/ Seconde(s)/ Milliseconde(s):

J pour les jours

H pour les heures

M pour les minutes

S pour les secondes

MS pour mes millisecondes

Ex : 1J

Ex : 2H30M15S

Couleur

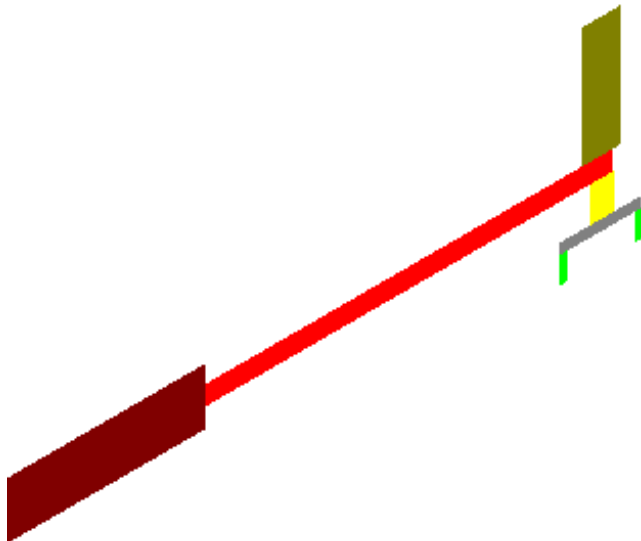
Permet de définir une couleur pour la grille.

La couleur est au format Rouge, Vert, Bleu

Ex : 255,0,0 Tracé rouge

CHAPITRE N°23.

L'objet OBJET



L'objet OBJET est utilisé là où les autres objets sont trop spécialisés. L'objet OBJET est le plus paramétrable de tous les objets. Il peut changer d'aspect, de position et être utilisé pour détecter la position d'autres objets (capteurs).

Les objets OBJET peuvent être liés les uns aux autres. Pour cela chaque objet peut recevoir un identificateur représenté par une valeur numérique. Les objets devant être attachés feront référence à l'identificateur.

Autres caractéristiques

Un texte peut être affiché sur l'objet.

Un texte d'aide et un texte bulle peuvent être associés à l'objet.

CHAPITRE N°24.

Les propriétés de l'objet OBJET - onglet « Aspect »

Type

Permet de définir un des types d'aspect de l'objet :

- « bitmap n aspects » : l'aspect de l'objet est donné par un fichier bitmap qui peut contenir plusieurs aspects (voir l'onglet « Bitmap »).
- « bitmap n couleurs » : l'aspect de l'objet est donné par un fichier bitmap, la couleur est contrôlée par une variable de l'application d'automatisme qui remplace les pixels blancs du bitmap. Les autres pixels du bitmap doivent être noirs. La variable de l'application d'automatisme donne un numéro de couleur, les couleurs sont définies dans l'onglet « Couleurs ».
- « bitmap jauge » : l'objet est une jauge dont la forme est définie par un bitmap. Ce sont les pixels blancs du bitmap qui définissent la forme. Les autres pixels doivent être noirs. Le minimum, le maximum et l'orientation sont réglés dans l'onglet « Jauge ».
- « formes n couleurs » : un rectangle, un rectangle aux bords arrondis ou une ellipse. La couleur est gérée de la même façon que pour le type « bitmap n couleurs ».
- « formes jauges » : l'objet est une jauge rectangulaire. Le principe est le même que pour le type « bitmap jauge ».

Couleurs

Permet de choisir la couleur des caractères du texte affiché sur l'objet.

Fonte de caractères

Détermine la fonte utilisée pour afficher un texte sur l'objet.

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Textes

Le texte d'aide et le texte bulle.

Le texte affiché sur l'objet : la position et l'orientation peuvent être modifiées.

Les propriétés de l'objet OBJET - onglet « Liens »

Objet cliqué, objet non cliqué

Définit les actions à réaliser respectivement lorsque l'utilisateur clique sur l'objet et lorsqu'il arrête de cliquer l'objet.

Une action peut être le forçage de l'état d'une variable, par exemple :

`o0=1, m200=4, _depart cycle_=3`

Ou un mot clé prédéfini.

Exemple de configuration pour que l'entrée i10 reflète l'état cliqué de l'objet (i10 à 0 si l'objet n'est pas cliqué, i10 à 1 si l'objet est cliqué) :

Objet cliqué : `i10=1`

Objet non cliqué : `i10=0`

Lier en permanence avec ...

Cette zone peut recevoir l'identificateur d'un objet frère. Si un tel objet existe alors la position de l'objet est calquée sur celui-ci. L'identificateur d'un objet est une valeur entière comprise entre 1 et 32767. Il est spécifié dans la zone d'édition « Identificateur » de la rubrique « Liens ».

Aspect / Couleur / Remplissage

Cette zone de la boîte de dialogue regroupe 8 zones d'édition qui permettent de définir différents types de comportement de l'objet par rapport aux variables de l'application d'automatisme.

Quel que soit le type de comportement il y aura toujours une position qui selon le type d'objet désignera :

- un aspect contenu dans un bitmap pour le type « bitmap n aspects »,
- un numéro de couleur pour les types « bitmap n couleurs » ou « forme n couleurs »,
- un remplissage pour les types « bitmap jauge » ou « forme jauge ».

La zone « Position » peut contenir un nom de variable numérique (C ou M). Les zones « + Position » et « - Position » peuvent contenir un nom de variables booléennes.

Deux types de fonctionnement sont possibles :

- si les zones « + Position » et « - Position » sont documentées alors les variables booléennes qu'elles contiennent pilotent la position : elles ajoutent ou enlèvent la valeur précisée dans la zone vitesse. Si la zone « Position » est documentée alors la position courante est écrite dans la variable dont elle contient le nom.
- si les zones « + Position » et « - Position » sont vides alors la valeur contenue dans la variable dont le nom est écrit dans la zone « Position » est lue comme position de l'objet.

La position peut varier entre les valeurs définies dans les zones « Mini » et « Maxi ».

Des capteurs peuvent être ajoutés (des noms de variables booléennes) qui seront vrais pour la position minimale et maximale (position égale au minimum ou au maximum).

Déplacement horizontal, déplacement vertical

Ces zones de la boîte de dialogue regroupent chacune 8 zones d'édition qui permettent respectivement de définir la position horizontale et verticale de l'objet. Le principe est identique à celui décrit ci-dessus.

Les propriétés de l'objet OBJET - onglet « Formes »

Formes

Pour le type « Forme n couleurs » cette rubrique permet de sélectionner un rectangle, un rectangle avec des coins arrondis ou une ellipse.

Les propriétés de l'objet OBJET - onglet « Bitmap »

Nom du fichier

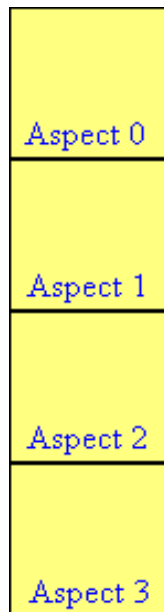
Pour les types « Bitmap n aspects, bitmap n couleurs et bitmap jauge » cette zone d'édition doit contenir un nom d'accès complet vers un fichier « .BMP ». Ces fichiers peuvent être créés avec PAINTBRUSH ou tout autre éditeur graphique capable de créer des fichiers « .BMP ».

Les boutons poussoir « Parcourir » et « Editer » permettent respectivement de rechercher un fichier « .BMP » et d'éditer (lancement de PAINTBRUSH) le fichier « .BMP » dont le nom se trouve dans la zone d'édition.

Nombre d'aspects

Cette zone d'édition doit contenir le nombre d'aspect (d'images) contenus dans un fichier « .BMP ». Cette option est utilisée pour le type « Bitmap n aspects ». Les différents aspects d'un objet doivent être dessinés les uns au dessous des autres. L'aspect le plus haut est le numéro 0.

Exemple de fichier « .BMP » à 4 aspects :



Le bitmap possède des zones transparentes ...

Cette option permet de créer un objet dont certaines parties seront transparentes (c'est le fond du pupitre père qui sera affiché). Les zones transparentes seront définies par des pixels possédant la même couleur, couleur qui sera déterminée par les trois composantes, rouge, verte et bleue. Pour régler ces composantes utilisez les trois barres de défilement. La couleur doit être réglée de façon très précise : exactement la même proportion de rouge, de vert et de bleu que la couleur des pixels des zones transparentes du bitmap.

Les propriétés de l'objet OBJET - onglet « Couleurs »

Couleurs

Cette rubrique est utilisée pour les types « bitmap n couleurs » et « forme n couleurs ». Chaque ligne contient la définition pour une couleur. La syntaxe utilisée pour chaque ligne est :

proportion de rouge (entre 0 et 255), proportion de vert (entre 0 et 255) et proportion de bleu (entre 0 et 255). La première ligne désigne la couleur numéro 0, la deuxième la numéro 1, etc ...

Cette rubrique est aussi utilisée pour les types « bitmap jauge » et « forme jauge ». La première ligne (couleur 0) et la deuxième (couleur 1) détermine les deux couleurs de la jauge (partie active et inactive).

Les propriétés de l'objet OBJET - onglet « Jauge »

Jauge

Cette rubrique est utilisée pour les types « bitmap jauge » et « forme jauge ». Les zones « Valeur minimum » et « Valeur maximum » définissent les bornes pour la variable de pilotage de la jauge.

Orientation de la jauge

Détermine une des quatre directions possibles pour la jauge.

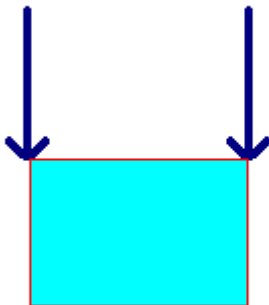
Les propriétés de l'objet OBJET - onglet « Capteur »

L'objet OBJET peut être utilisé comme capteur. Le capteur est associé à une variable booléenne dont le résultat est vrai si le capteur est en contact avec une ou plusieurs couleurs prédéfinies (voir plus bas) ou faux autrement.

Position de la détection

Permet de déterminer le côté de l'objet qui doit détecter. La détection est réalisée sur les deux bords du côté choisi.

Exemple pour une détection au dessus :



Couleurs détectées

Un capteur est capable de détecter jusqu'à trois couleurs différentes. Si une de ces trois couleurs se trouve aux points de test alors la variable booléenne associée au capteur (voir onglet « Liens ») est positionnée à 1, autrement elle est positionnée à 0.

Les trois zones d'édition peuvent contenir une définition de couleur sous la forme de trois valeurs comprises entre 0 et 255 et qui correspondent respectivement aux pourcentages de rouge, de vert et de bleu. Les pourcentages de ces trois couleurs doivent correspondre avec exactitude aux couleurs des objets à détecter pour que le capteur fonctionne.

Les propriétés de l'objet OBJET - onglet « Options »

Touche

Définit une touche qui permet de simuler un clique sur l'objet.

Pour préciser le code de la touche, différentes syntaxes sont utilisables :

- un simple caractère: par exemple A, Z, 2,
- le caractère \$ suivi du code de la touche en hexadécimal,
- le nom d'une touche de fonction, par exemple F5.

Pour les combinaisons de touches il faut ajouter au début « CTRL+ » ou « SHIFT+ »

Par exemple : « CTRL+F4 » ou « SHIFT+Z ».

La touche TAB permet d'accéder à cet objet

Si cette case n'est pas cochée, alors la touche TAB ne permet pas d'activer l'objet.

CHAPITRE N°25.

Objet OBJET - techniques avancées

Lier des objets de façon dynamique

Cette possibilité permet de lier momentanément un objet à un autre. Pour lier un objet à un autre les paramètres « + Position » et « - Position » qui gèrent la position horizontale sont utilisés de façon particulière. Ces deux paramètres doivent contenir le nom d'une variable numérique (M). La variable « + Position » doit contenir la valeur f000 (hexadécimal) et la variable « - Position » l'identificateur de l'objet avec lequel l'attachement doit être effectué. La variable « + Position » est remise à zéro lorsque l'attachement a été réalisé. Pour annuler l'attachement d'un objet il faut placer la valeur f001 (hexadécimal) dans la variable « + Positon ». Voir l'exemple bras au chapitre 41.

Echange de paramètres entre deux objets

Un objet peut accéder aux paramètres d'un objet frère en utilisant le mot clé « SISTERPARAM ».

La syntaxe est :

SISTERPARAM(identificateur de l'objet frère , paramètre)

« paramètre » peut prendre les valeurs suivantes :

| | |
|------------|--|
| STATE | état de l'objet : valeur de Aspect / Couleur / Remplissage |
| -STATE | idem mais valeur négative |
| POSX | position sur l'axe horizontal |
| -POSX | idem mais valeur négative |
| POSY | position sur l'axe des y |
| -POSY | idem mais valeur négative |
| POSX+STATE | position sur l'axe horizontal plus état |
| POSX-STATE | position sur l'axe horizontal moins l'état |
| POSY+STATE | position sur l'axe vertical plus état |
| POSY-STATE | position sur l'axe vertical moins l'état |

CHAPITRE N°26.

Objet SON



L'objet SON est utilisé pour générer des sons.

Les fichiers au format « .WAV » sont utilisés pour stocker les sons. Chaque fichier son est associé à une variable booléenne de l'application d'automatisme. Lorsque la variable est à l'état 1 le son est joué et la variable est remise à 0 (pour signaler que la demande a été prise en compte, pas que le son est terminé).

CHAPITRE N°27.

Propriétés de l'objet SON - onglet « Aspect »

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Propriétés de l'objet SON - onglet « Sons »

Nom des fichiers sons

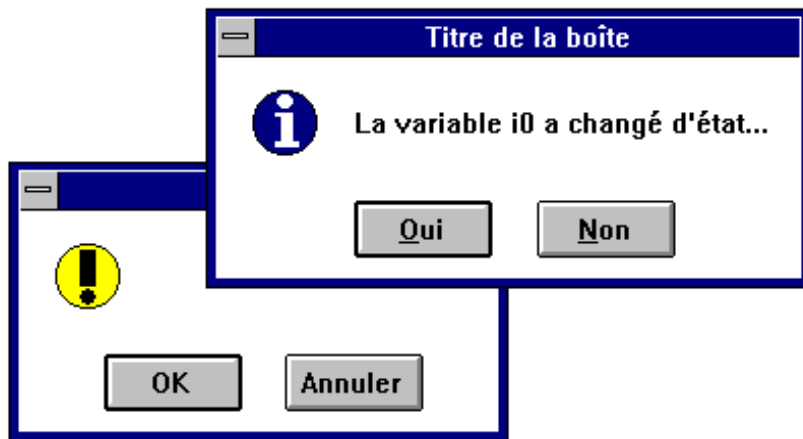
Le nom d'accès complet vers les fichiers « .WAV ».

Variables associées

La variable booléenne associée à chaque son.

CHAPITRE N°28.

Objet DIAL



L'objet DIAL permet, selon l'état d'une variable de l'application d'automatisme, d'afficher un message prédéfini dans une boîte de dialogue, puis d'effectuer des actions différentes suivant la réponse de l'utilisateur.

Cet objet peut surveiller l'état d'une variable booléenne ou numérique.

Si la variable est de type booléenne, on affiche un message prédéfini, alors que si la variable est numérique, on peut afficher différents messages suivant sa valeur.

Par la suite, l'utilisateur peut réagir de différentes manières, et à chaque manière correspond un code de retour particulier que l'on peut utiliser dans le programme.

L'utilisateur a la possibilité de configurer l'apparence graphique de la boîte de dialogue.

CHAPITRE N°29.

Propriétés de l'objet DIAL - onglet « Aspect »

Type de boîte

Permet de choisir les différents contrôles présents dans la boîte de dialogue : un seul bouton OK, deux boutons OK et CANCEL, ou enfin deux boutons OUI et NON.

Icônes

Permet de choisir l'icône qui apparaîtra dans la boîte de dialogue. Il y a quatre icônes différentes, mais bien sur, on peut décider d'en afficher aucune. Il convient de noter aussi qu'à chaque icône est associé un son système particulier. Pour plus de renseignements à ce sujet, se reporter à la rubrique sur l'option BEEP.

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Beep

Permet de spécifier si l'apparition de la boîte de dialogue doit être accompagnée d'un avertissement sonore.

Titre

Permet de spécifier le titre de la boîte de dialogue.

Type de message

Il y a ici deux possibilités. Un message prédéfini est un message présent dans les variables utilisateurs de l'application d'automatisme. L'autre possibilité est de spécifier une liste de messages, et dans ce cas, le message affiché sera fonction de l'état de la variable surveillée.

Propriétés de l'objet DIAL - onglet « Liens »

Nom de la variable

Spécifie le nom de la variable à surveiller. On peut entrer indifféremment des variables booléennes ou numériques.

Par exemple :

m200, i0

Si la variable en question est booléenne, alors le message n°1 dans la liste s'affichera dès que l'état de cette variable passera à 1.

Pour une variable numérique, si l'option de configuration « Liste de message » est cochée, alors la boîte de dialogue s'affichera dès que la valeur sera comprise entre 1 et le nombre de messages mémorisés dans la liste.

Par exemple, si la liste contient 8 messages, alors il ne se passera rien quand la variable prendra des valeurs négatives ou supérieures à 8. Par contre, dès que sa valeur passe dans l'intervalle 1..8, alors le message approprié est affiché...

Si l'option « Message prédéfini » est activée, alors la boîte de dialogue s'affichera avec un message de longueur le contenu de la variable, et situé dans les variables de l'application d'automatisme suivant cette variable.

Par exemple, si m200=4, cela signifie qu'un message de longueur 4 caractères est situé dans les 4 variables suivants m200, c'est à dire m201, m202, m203, m204.

Code de retour d'une boîte de dialogue :

Dans le cas d'une variable booléenne, quelle que soit l'action effectuée par l'utilisateur, son contenu passera à 0. Pour une variable numérique, il existe différents codes de retour :

| | |
|----------------------------|---|
| Appui sur un bouton OK | : La variable prend la valeur 8000 (hexa) |
| Appui sur un bouton CANCEL | : La variable prend la valeur 8001 (hexa) |
| Appui sur un bouton OUI | : La variable prend la valeur 8002 (hexa) |
| Appui sur un bouton NON | : La variable prend la valeur 8003 (hexa) |

Remarque : L'activation d'une boîte de dialogue est basée sur le front montant, c'est à dire passage de 0 à 1 pour une variable booléenne, et passage d'une valeur extérieure à l'intervalle de la liste de messages, à une valeur incluse dans celui-ci, pour une variable numérique.

Identificateur

Permet de référencer l'objet par rapport aux autres objets.

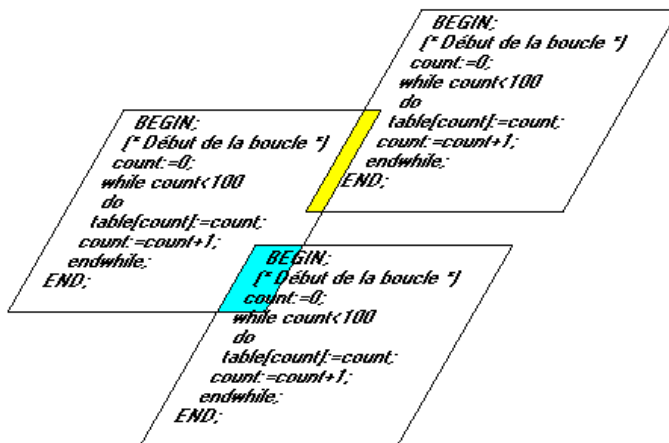
Propriétés de l'objet DIAL - onglet « Messages »

Liste de messages

Entrer dans cette zone les différents messages prédéfinis.

CHAPITRE N°30.

Objet PROG



L'objet PROG est utilisé pour écrire des programmes permettant d'effectuer des traitements numériques et algorithmiques.

Ces programmes peuvent accéder aux variables de l'application d'automatisme en lecture et en écriture.

Répartition du temps d'exécution

Les objets d'IRIS sont exécutés tour à tour. Cette répartition du temps d'exécution est gérée de façon transparente par le gestionnaire d'objet. Deux niveaux de priorité sont possibles pour les objets « PROG » : si la case à cocher « Exécution prioritaire » de l'onglet « Programme » est cochée, alors la totalité du programme s'exécute lorsque l'objet à la main. Dans le cas contraire, seule une ligne s'exécute avant que l'objet ne rende la main. Des exceptions existent à cette règle : les fonctions d'accès aux variables d'automatisme (« READVAR » et « WRITEVAR ») peuvent provoquer un passage de la main, la fonction « YIELD » force un

passage de la main. En mode d'exécution prioritaire, cette fonction doit obligatoirement être utilisée à l'intérieur d'une boucle pour ne pas bloquer l'exécution des autres objets.

Affichage

La surface de l'objet peut être utilisée pour afficher des informations. La fonction « PRINT » permet d'afficher des informations.

Syntaxe

Le caractère « ; » (point virgule) est utilisé comme séparateur. Des commentaires peuvent être écrits entre les chaînes « (* » et « *) ». Il n'y a pas de différence entre les majuscules et les minuscules pour les mots clés et les noms de fonctions, en revanche, pour les noms de variables la différenciation est faite.

Déclaration des variables

Les variables utilisées dans un programme doivent être déclarées avant le programme entre les mots clés « BEGINVAR; » et « ENDVAR; ».

Les types de variables utilisables sont :

| | |
|--------|--------------------------|
| INT | entier 16 bits signé |
| UINT | entier 16 bits non signé |
| LONG | entier 32 bits signé |
| ULONG | entier 32 bits non signé |
| STRING | chaîne de caractères |
| FLOAT | flottant |

La syntaxe générale d'une déclaration est :

<type> <nom de variable>;

La syntaxe générale pour déclarer un tableau de variables est :

<type> <nom de variable> [<longueur>];

Par exemple :

```
BEGINVAR;  
INT compteur; (* un entier 16 bits signé *)  
STRING chaine; (* une chaîne *)  
(* un tableau de 100 entiers 32 bits non signés *)  
ULONG table[100];  
ENDVAR;
```

Ecriture du programme

Le programme doit être écrit entre les mots clés « BEGIN; » et « END; ».

Exemple :

```
BEGIN;  
print "Bonjour !";  
END;
```

Constantes

- entier 16 bits : un nombre décimal compris entre -32768, et 32727 ou "\$" suivi d'un nombre hexadécimal compris entre 0 et FFFF. Exemple : 12, -4, \$abcd
 - entier 32 bits : un nombre décimal compris entre -2147483648 et 214743648 suivi de "L" ou "\$" suivi d'un nombre hexadécimal compris entre 0 et FFFFFFFF suivi de "L". Exemple: 10000L, -200000L, \$12345678L
 - chaîne de caractères : un caractère guillemet suivi des caractères de la chaîne suivi d'un caractère guillemet. Des caractères de contrôles peuvent être insérés dans une chaîne. « \n » remplace un caractère LF (code ASCII 10), « \r » un caractère CR (code ASCII 13). Exemple : "Abcdef", "" (chaîne nulle), "Suite\r\n"
- flottant : un nombre décimal suivi du caractère "R", le caractère "." sert de délimiteur entre la partie entière et la partie décimale. Exemple : 3.14r, -100.4r

Affectation

La chaîne « := » marque une affectation.

Exemple :

```
compteur:=4;  
var:="ABCDEF";
```

Calculs

Les opérateurs de calculs sont évalués de la gauche vers la droite. Des parenthèses peuvent être utilisées pour spécifier une priorité de calcul.

Liste des opérateurs de calcul :

- + addition (concaténation pour les chaînes de caractères)
- - soustraction
- * multiplication
- / division
- << décalage à gauche
- >> décalage à droite
- ^ élévation à une puissance
- AND "et" binaire
- OR "ou" binaire
- XOR "ou exclusif" binaire

Exemples :

```
resultat:=var1*(var2+var3);  
resultat:=resultat<<2;
```

Tests

Syntaxe :

```
IF <condition> THEN ... ENDIF;
```

ou

```
IF <condition> THEN ... ELSE ... ENDIF;
```

Exemple :

```
IF (count<100) AND (count>10)  
    THEN  
    count:=count+1;  
    ELSE  
    count:=0;  
    ENDIF;
```


Boucles

Syntaxe :

```
WHILE <condition> DO ... ENDWHILE;
```

Exemple :

```
count:=0;
WHILE count<1000
    DO
        table[count]:=table[count+1];
        count:=count+1;
    ENDWHILE;
```

Adresse d'une variable ou d'un tableau de variables

La syntaxe &nom de variable ou &nom d'un tableau de variables retourne l'adresse d'une variable ou d'un tableau de variables. Cette syntaxe est nécessaire pour certaines fonctions.

Liste des fonctions

Pour les exemples proposés dans ce qui suit on admettra que :

vint est une variable de type INT, vlong une variable de type LONG, vuint une variable de type UINT, vulong une variable de type ULONG, vfloat une variable de type FLOAT et vstring une variable de type STRING.

PRINT

Fonction d'affichage. Les données à afficher sont écrites à la suite et séparées par des virgules.

Exemple :

```
print "Le résultat est :",vint/12,"\n";
```

NOT

Complément. Cette fonction peut être utilisée avec le test if pour compléter un résultat.

Exemple :

```
if not(1<2) then ...
```

ABS

Valeur absolue.

Exemple :

```
print abs(0-4); (* affiche 4 *)
```

VAL

Retourne la valeur d'une chaîne de caractères exprimée sous la forme d'un nombre décimal.

Exemple :

```
vlong=val("-123456"); (* vlong contiendra -123456 *)
```

HVAL

Retourne la valeur d'une chaîne de caractères exprimée sous la forme d'un nombre hexadécimal.

Exemple :

```
vuint=hval("abcd"); (* vuint contiendra abcd hexa *)
```

ASC

Retourne le code ASCII du premier caractère d'une chaîne.

Exemple :

```
vuint :=asc("ABCD"); (* vuint contiendra 65 : code ascii de 'A' *)
```

CHR

Retourne une chaîne composée de 1 caractère dont le code ASCII est passé en paramètre.

Exemple :

```
vstring:=chr(65); (*vstring contiendra la chaîne "A" *)
```

STRING

Retourne une chaîne composée de n caractères. Le premier argument est le nombre de caractères, le deuxième le caractère.

Exemple :

```
vstring:=string(100," ");  
(* vstring contiendra une chaîne composée de 100 espaces *)
```

STR

Convertit une valeur numérique entière en une chaîne de caractères représentant cette valeur en décimal.

Exemple :

```
vstring:=str(100); (* vstring contiendra la chaîne "100" *)
```

HEX

Convertit une valeur numérique entière en une chaîne de caractères représentant cette valeur en hexadécimal.

Exemple :

```
vstring:=hex(100); (* vstring contiendra la chaîne "64" *)
```

LEFT

Retourne la partie gauche d'une chaîne de caractères. Le premier argument est la chaîne de caractères, le deuxième le nombre de caractères à extraire.

Exemple :

```
vstring:=left("abcdef",2); (* vstring contiendra "ab" *)
```

RIGHT

Retourne la partie droite d'une chaîne de caractères. Le premier argument est la chaîne de caractères, le deuxième le nombre de caractères à extraire.

Exemple :

```
vstring:=right("abcdef",2); (* vstring contiendra "ef" *)
```

MID

Retourne une partie d'une chaîne de caractères. Le premier argument est la chaîne de caractères, le deuxième la position où doit commencer l'extraction, le troisième le nombre de caractères à extraire.

Exemple :

```
vstring:=mid("abcdef",1,2); (* vstring contiendra "bc" *)
```

LEN

Retourne la longueur d'une chaîne de caractères.

Exemple :

```
vuint:=len("123"); (* vuint contiendra 3 *)
```

COS

Retourne le cosinus d'une valeur réelle exprimée en radian.

Exemple :

```
vfloat:=cos(3.14r); (* vfloat contiendra le cosinus de 3.14 *)
```

SIN

Retourne le sinus d'une valeur réelle exprimée en radian.

Exemple :

```
vfloat:=sin(3.14r); (* vfloat contiendra le sinus de 3.14 *)
```

TAN

Retourne la tangente d'une valeur réelle exprimée en radian.

Exemple :

```
vfloat:=tan(3.14r); (* vfloat contiendra la tangente de 3.14 *)
```

ATN

Retourne l'arc tangente d'une valeur réelle.

Exemple :

```
vfloat:=atn(0.5r); (* vfloat contiendra l'arc tangente de 0.5 *)
```

EXP

Retourne l'exponentielle d'une valeur réelle.

Exemple :

```
vfloat:=exp(1r); (* vfloat contiendra l'exponentielle de 1 *)
```

LOG

Retourne le logarithme d'une valeur réelle.

Exemple :

```
vfloat:=log(1r); (* vfloat contiendra le logarithme de 1 *)
```

LOG10

Retourne le logarithme base 10 d'une valeur réelle.

Exemple :

```
vfloat:=log10(1r);
```

```
(* vfloat contiendra le logarithme base 10 de 1 *)
```

SQRT

Retourne la racine carrée d'une valeur réelle.

Exemple :

```
vfloat:=sqrt(2); (* vfloat contiendra la racine carrée de 2 *)
```

DATE

Retourne une chaîne de caractères représentant la date.

Exemple :

```
print "La date est :",date(),"\n";
```

TIME

Retourne une chaîne de caractères représentant l'heure.

Exemple :

```
print "L'heure est :",time(),"\n";
```

RND

Retourne un nombre aléatoire.

Exemple :

```
print rnd();
```

OPEN

Ouvre un fichier. Le premier argument est le nom du fichier, le deuxième le mode d'accès qui peut être : « r+b » ouverture en lecture/écriture, « w+b » ouverture en écriture (si le fichier existe il est détruit). La fonction retourne un long qui identifiera le fichier. Si l'ouverture échoue, la valeur retournée est 0.

Exemple :

```
vulong:=open("nouveau","w+b");
```

CLOSE

Ferme un fichier. L'argument est l'identificateur de fichier retourné par la fonction OPEN.

Exemple :

```
close(vulong);
```

WRITE

Ecrit des données dans un fichier. Le premier argument est l'identificateur de fichier retourné par la fonction OPEN. Le deuxième argument est l'adresse d'une variable, le troisième le nombre d'octets à écrire. La fonction retourne le nombre d'octets effectivement écrits.

Exemple :

```
vuint:=write(vulong,&buff,5);
```

READ

Lit des données dans un fichier. Le premier argument est l'identificateur de fichier retourné par la fonction OPEN. Le deuxième argument est l'adresse d'une variable, le troisième le nombre d'octets à lire. La fonction retourne le nombre d'octets effectivement lus.

Exemple :

```
vuint:=read(vulong,&buff,5);
```

SEEK

Déplace un pointeur de fichier. Le premier argument est l'identificateur de fichier retourné par la fonction OPEN, le deuxième la position.

Exemple :

```
seek(vulong,01);
```

GOTO

Effectue un saut au label passé en argument. L'argument est une chaîne de caractères.

Exemple :

```
goto "fin"  
...  
fin;
```

CALL

Effectue un saut à un sous-programme. L'argument est une chaîne de caractères contenant le label du sous-programme.

Exemple :

```
BEGIN;  
(* programme principal *)  
call "sp"  
END;  
BEGIN;  
(* sous programme *)  
sp:
```

```
print "Dans le sous-programme\n";  
return;  
END;
```

RETURN

Marque la fin d'un sous-programme.

READVAR

Lit une ou plusieurs variables de l'application d'automatisme. Le premier argument est le nom de la variable d'automatisme (nom de variable ou de symbole). Le deuxième argument est l'adresse d'une variable ou d'un tableau de variables 32 bits (longs ou flottants). Le troisième argument est le nombre de variables à lire. Si la fonction est exécutée sans erreur, la valeur retournée est 0.

Exemple :

```
readvar("i0",&buff,16); (* lecture de 16 entrées à partir de i0 *)
```

WRITEVAR

Ecrit une ou plusieurs variables de l'application d'automatisme. Le premier argument est le nom de la variable d'automatisme (nom de variable ou de symbole). Le deuxième argument est l'adresse d'une variable ou d'un tableau de variables 32 bits (longs ou flottants). Le troisième argument est le nombre de variables à écrire. Si la fonction est exécutée sans erreur, la valeur retournée est 0.

Exemple :

```
writevar("o0",&buff,16);  
(* écriture de 16 sorties à partir de o0 *)
```

CMD

Exécute une commande. L'argument est une chaîne qui spécifie la commande à exécuter. Cette fonction permet d'utiliser les commandes prédéfinies d'IRIS. Pour plus de détails consultez la notice d'utilisation au chapitre 10. La valeur retournée est 0 si la commande a été exécutée sans erreur.

Exemple :

```
cmd("run");
```

YIELD

Passes la main. Cette fonction permet de ne pas monopoliser l'exécution lorsque l'objet est exécuté en mode prioritaire.

Exemple :

```
WHILE 1
    DO
        ...
        yield();
ENDWHILE;
```

DLL

Appelle une DLL. Le premier argument est le nom du fichier DLL. Le deuxième est le nom de la fonction. Le troisième est un pointeur sur une variable 32 bits qui recevra le code de retour de la fonction. Les autres arguments sont passés à la fonction.

Exemple :

```
dll "user", "messagebeep", &vulong, -1;
```

Messages d'erreurs

| | |
|---|---|
| « séparateur ';' manquant » | il manque un point virgule |
| « erreur de syntaxe » | une erreur de syntaxe a été détectée |
| « variable définie plusieurs fois » | une variable est définie plusieurs fois |
| « pas assez de mémoire » | l'exécution du programme a saturé la mémoire disponible |
| « variable non définie » | une variable utilisée dans le programme n'est pas définie |
| « constante trop grande » | une constante est trop grande |
| « programme trop complexe » | une expression est trop complexe, il faut la décomposer |
| « type de variable ou de constante incompatible » | une variable ou une constante n'est pas du type attendu |
| « ')' manquant » | il manque une parenthèse fermante |
| « ENDIF manquant » | il manque le mot-clé ENDIF |
| « 'ENDWHILE' manquant » | il manque le mot-clé ENDWHILE |
| « label introuvable » | un label de saut ou de sous-programme est introuvable |
| « ']' manquant » | il manque un crochet fermant |
| « numéro d'élément hors borne » | un élément de tableau hors limite a été utilisé |
| « trop de 'CALL' imbriqués » | trop de sous-programme imbriqués ont été utilisés |

« 'RETURN' trouvé sans 'CALL' » RETURN trouvé en dehors d'un sous-programme
« taille de variable trop petite » la taille d'une variable est insuffisante
« fichier DLL introuvable » le fichier DLL est introuvable
« fonction introuvable dans DLL » la fonction est introuvable dans le fichier DLL
« division par zéro » une division par 0 s'est produite
« erreur mathématique » une fonction mathématique a provoqué une erreur

CHAPITRE N°31.

Propriétés de l'objet PROG - onglet « Aspect »

Couleurs

Permet de choisir la couleur du fond et des caractères de l'objet.

Taille de l'objet

Détermine les dimensions de l'objet en nombre de points. La modification de ces valeurs permet de régler très précisément la taille de l'objet.

Texte

Permet de spécifier un texte bulle qui apparaît lorsque le curseur est placé sur l'objet.

Propriétés de l'objet PROG - onglet « Programme »

Programme

Cette zone d'édition contient le programme.

Exécuter

Si cette case est cochée, alors le programme est exécuté.

Exécution prioritaire

Si cette case est cochée, alors le programme s'exécute plus rapidement.

Exécution au démarrage

Si cette case est cochée, alors le programme s'exécute à l'ouverture de l'objet. Cette option permet de sauvegarder un objet avec l'option « Exécuter » non cochée tout en demandant une exécution de l'objet au chargement.

Aller sur l'erreur

Si une erreur a été détectée pendant l'exécution du programme, alors ce bouton poussoir permet de placer le curseur à l'endroit qui a provoqué l'erreur.

CHAPITRE N°32.

Exemple d'objet composé

Cet exemple va vous permettre de comprendre comment créer un objet « Clavier décimal » composé de touches : « 0 » à « 9 » plus une touche [ENTER] pour valider.

Ouvrez un objet « PUPITRE.AOF », puis à partir du menu du pupitre ouvrez un objet « BPVOYANT.AOF ». Nous allons paramétrer cet objet BPVOYANT puis nous le dupliquerons pour obtenir les autres touches. Ensuite nous retoucherons les propriétés des touches dupliquées pour les personnaliser : texte affiché sur la touche et action. Nous aurons ainsi un clavier dont l'aspect des touches sera homogène.

Le lien avec l'application sera réalisé par l'intermédiaire d'un mot. Lorsqu'une touche sera enfoncée elle écrira son code (0 à 9 ou 13 pour la touche de validation) dans ce mot.

Pour spécifier ce mot nous pourrions donner son numéro sous la rubrique action des propriétés de chacun des objets. L'inconvénient est que lorsque nous réutiliserons l'objet « Clavier décimal » et si nous voulons utiliser un autre mot il faudra modifier les propriétés des 11 objets BPVOYANT.

Pour contourner ce problème nous allons utiliser la possibilité donnée aux objets enfants d'accéder à un paramètre défini dans les propriétés d'un pupitre parent. L'onglet « Liens » de la fenêtre des propriétés du pupitre permet de définir le paramètre. Ecrivez dans la zone d'édition sur une seule ligne : « CLAVIER=M200 ». Cette ligne signifie que le paramètre clavier est égal à M200.

Les touches du clavier feront référence au paramètre « CLAVIER » et non directement au mot M200. Ainsi pour changer le mot utilisé il suffira de modifier la définition du paramètre dans les propriétés du pupitre.

Revenons à l'aspect de notre clavier ...

Pour que l'aspect du clavier soit satisfaisant nous allons définir une grille pour aligner les touches. Dans la fenêtre des propriétés du pupitre et l'onglet « Options » écrivez la valeur « 10 » sous les deux rubriques « Grilles ». Ainsi la fonction déplacer du menu du pupitre utilisera une grille de 10 pixels. Nous allons également fixer les dimensions de la première touche. Nous pourrions directement modifier les dimensions de la touche en la saisissant par un de ses bords, cependant pour plus de précision nous allons modifier directement les dimensions sous la rubrique « Taille de l'objet en pixel » de l'onglet « Aspect » de la fenêtre des propriétés de l'objet BPVOYANT. Entrez par exemple « 30 » pour la largeur et la hauteur.

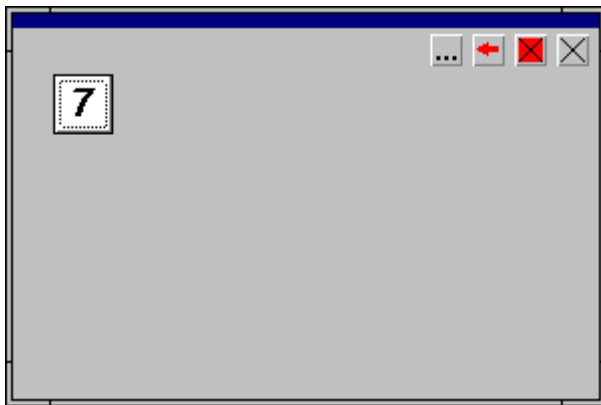
A ce stade, vous pouvez également personnaliser le style de la touche : la couleur, la fonte utilisée pour le marquage, etc ...

Nous placerons cette première touche en haut et à gauche du clavier (c'est un choix arbitraire). Le clavier que nous allons créer ressemblera au pavé numérique d'un clavier d'ordinateur. Nous allons donc marquer cette touche avec le texte « 7 » sous la rubrique « Texte » de l'onglet « Aspect ».

Nous allons également paramétrer l'aspect fonctionnel de la touche : sous la rubrique « Action lorsque le bouton est enfoncé » de l'onglet « Liens » nous allons écrire: « PARENTPARAM(CLAVIER)=7 ». Ce qui signifie que lorsque le bouton poussoir sera enfoncé le mot désigné par le paramètre « CLAVIER » du pupitre parent recevra la valeur 7. Effacez ce qui se trouve sous la rubrique « Action lorsque le bouton est relâché ».

Nous pouvons également affecter à l'objet BPVOYANT une touche du clavier de l'ordinateur. Ainsi il sera possible d'utiliser le clavier avec la souris ou le clavier de l'ordinateur. Pour affecter une touche à l'objet BPVOYANT utilisez la rubrique « Touche » de l'onglet « Options ». Entrez par exemple « 7 » pour associer la touche « 7 » du clavier de l'ordinateur à l'objet.

Placez ensuite la touche « 7 » en haut et à gauche du clavier, comme ceci :



Pour déplacer cette touche sélectionnez d'abord l'objet (touche [SHIFT] enfoncée puis cliquez avec le bouton gauche de la souris sur l'objet), puis utilisez la fonction « Déplacer » du menu du pupitre. Cette fonction est la seule qui utilise la grille contrairement au déplacement en saisissant la barre des objets enfants.

Pour créer les autres touches, dupliquez la touche existante :

- sélectionnez la première touche,
- choisissez « Copier » dans le menu du pupitre, puis « Coller »,
- déplacez la touche ainsi collée,
- paramétrez la nouvelle touche : (texte, liens et touche du clavier de l'ordinateur).

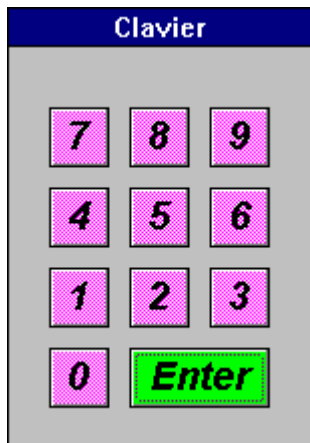
Lorsque vous avez terminé la rangée du haut (touches « 7 », « 8 » et « 9 ») vous pouvez sélectionner ensemble ces trois touches et les dupliquer.

Vous pouvez créer une touche de validation (plus large pour remplir la surface du clavier).

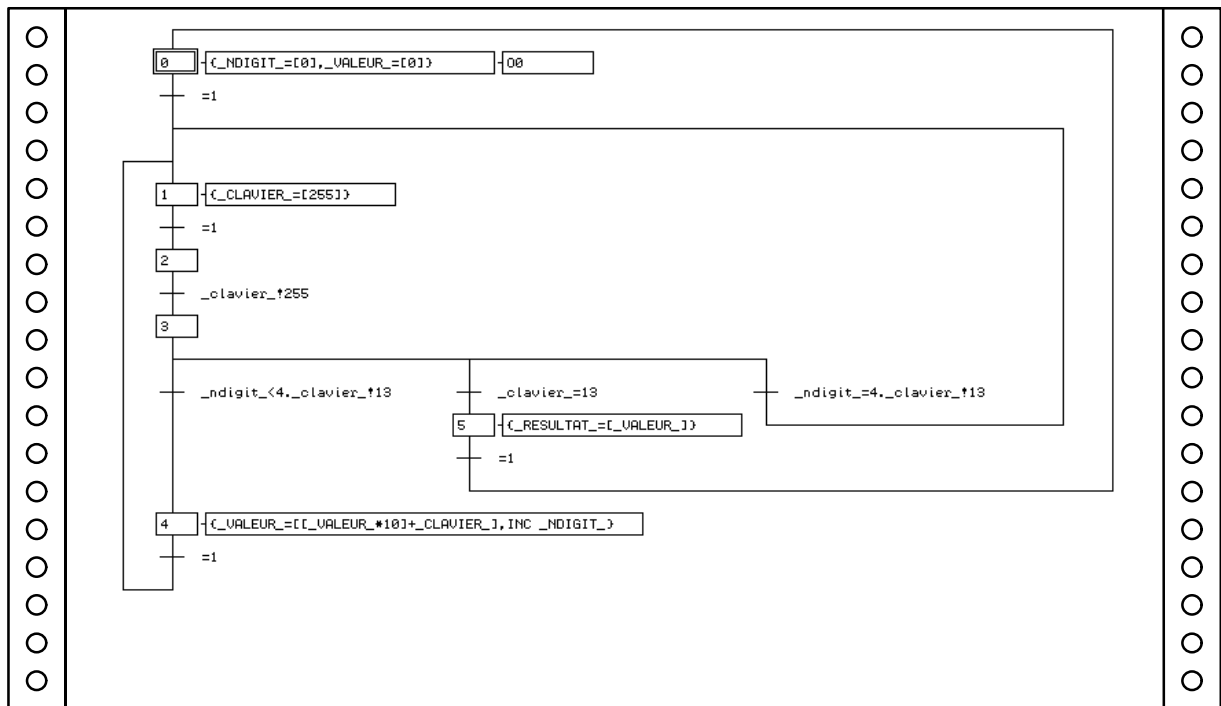
Pour terminer, redimensionnez le pupitre, passez les objets en mode « Exploitation » et sauvegardez l'objet sous le nom « CLAVIER.AOF » par exemple.

L'objet « CLAVIER.AOF » et une l'application AUTOMGEN « CLAVIER.GR7 » (test de l'objet CLAVIER) se trouvent dans le sous répertoire « EXEMPAOF » du répertoire où a été installé IRIS V3.

Le résultat final doit ressembler à ceci :



☞ « <répertoire où est installé IRIS>\EXEMPAOF\CLAVIER.AOF »



☞ « <répertoire où est installé IRIS>\EXEMPAOF\CLAVIER.GR7 »

CHAPITRE N°33.

Exemple d'utilisation de l'objet ECRANCLA comme liste à messages

Cet exemple illustre l'utilisation de l'objet ECRANCLA.

Cahier des charges :

- l'objet doit afficher quatre messages différents suivant l'état de quatre entrées (i0 à i3),
 - pour l'entrée 0 : un message d'information « Départ de cycle »,
 - pour l'entrée 1 : un message d'information « Fin de cycle »,
 - pour l'entrée 2 : un message d'erreur « Erreur 1 »,
 - pour l'entrée 3 : un message d'erreur « Erreur 2 ».
- les messages doivent être affichés à l'apparition du front montant des entrées,
- un historique de 50 messages sera conservé dans l'objet et sauvegardé sur disque,
- les messages seront dupliqués sur une imprimante connectée sur « LPT1 : »,
- un bouton poussoir doit permettre d'effacer les messages.

Solution :

L'application d'automatisme va gérer les tests de fronts sur les entrées. Le nombre de fronts sera mémorisés dans des mots puis les demandes d'affichage des messages seront transmises à l'objet. Cette structure assure qu'aucune information ne sera perdue notamment si plusieurs entrées deviennent actives simultanément.

Les mots m200 à m209 seront utilisés pour dialoguer entre l'application d'automatisme et l'objet ECRANCLA.

Programme d'automatisme réalisé avec AUTOMGEN :

```
Exemple d'utilisation de l'objet ECRANCLA

#$m200=0,0,-1,-1,-1,-1,-1,-1,0
#$m300=0,0,0,0

↓

if(#i0)
  then
  inc m300
  endif
if(#i1)
  then
  inc m301
  endif
if(#i2)
  then
  inc m302
  endif
if(#i3)
  then
  inc m303
  endif
if(m300>0,m200=0)
  then
  m200=[1]           ; message 1
  m201=[-1]         ; classe information
  dec m300
  endif
if(m301>0,m200=0)
  then
  m200=[2]           ; message 2
  m201=[-1]         ; classe information
  dec m301
  endif
if(m302>0,m200=0)
  then
  m200=[3]           ; message 3
  m201=[-2]         ; classe alarme
  dec m302
  endif
if(m303>0,m200=0)
  then
  m200=[4]           ; message 4
  m201=[-2]         ; classe alarme
  dec m303
  endif
```

☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXECLA.GR7 »

Les mots m300 à m303 cumulent le nombre de fronts montants pour chaque entrée. Le mot m200 est le numéro de message à afficher, le mot m201 est la classe de messages.

Les valeurs -1 et -2 sont des classes prédéfinies pour les messages d'information et les alarmes.

Les mots m202 à m209 sont prédisposés à la valeur -1 pour afficher la date et l'heure courante du PC.

Nous allons utiliser trois objets : un objet PUPITRE qui contiendra, un objet ECRANCLA et un objet BPVOYANT pour l'effacement des messages.

Paramétrage de l'objet ECRANCLA :

Aspect Liens Ljste Options Messages

Type de l'objet

- Ecran seulement
- Ecran et clavier avec echo
- Ecran et clavier sans echo
- Boîte à message, liste d'alarmes

Textes

Texte bulle

Fonte de caractères

Couleurs

Rouge Fond

Vert

Bleu Caractères

Taille de l'objet en pixels

| | |
|----------------------------------|----------------------------------|
| Largeur | Hauteur |
| <input type="text" value="470"/> | <input type="text" value="130"/> |

Aspect Liens Ljste Options Messages

Options

- Afficher les codes hexa. des caractères
- Base de défilement horizontal
- Base de défilement vertical
- Convertir les caractères OEM -> ANSI
- Dupliquer les messages vers (nom de fichier ou de périphérique) :
- Associer un fichier de stockage des messages sous le nom :
- Ecrire les messages périmés vers :
- Nombre de lignes mémorisées (0=pas de limite)

Aspect Liens Ljste Options Messages

Messages prédéfinis

| | |
|----|-----------------|
| 1 | Départ de cycle |
| 2 | Fin de cycle |
| 3 | Erreur 1 |
| 4 | Erreur 2 |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |
| 13 | |

Paramétrage de l'objet BPVOYANT :

The image shows two screenshots of the BPVOYANT configuration interface. The top screenshot is the 'Options' tab, and the bottom screenshot is the 'Liens' tab.

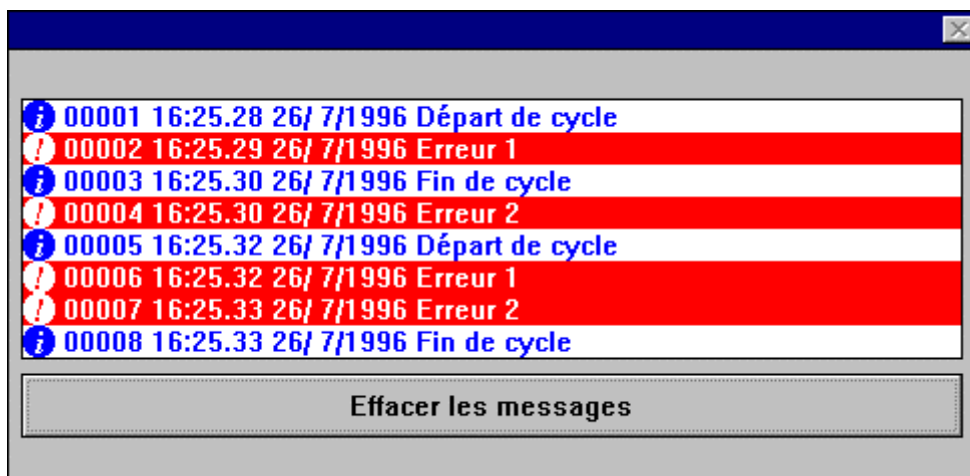
Options Tab:

- Type de l'objet:**
 - Bouton poussoir
 - Voyant
 - Bouton poussoir et voyant
- Couleurs:**
 - Rouge:
 - Vert:
 - Bleu:
 - Fond éteint:
 - Fond allumé:
 - Caractères:
- Taille de l'objet en pixels:**
 - Largeur: 470
 - Hauteur: 32
- Fonte de caractères:**
- Textes:**
 - Texte: Effacer les messa
 - Orientation: 0 deg
 - Texte d'aide:
 - Texte bulle:
 - Position verticale du texte: centré en haut en bas
 - Position horizontale du texte: centré à gauche à droite

Liens Tab:

- Action lorsque le bouton est enfoncé:
- Action lorsque le bouton est relâché:
- Etat du voyant: i0
- Identificateur: 0

Le résultat final peut avoir l'aspect suivant :



« <répertoire où est installé IRIS>\EXEMPAOF\EXECLA.AOF »

Variante :

L'appui sur le bouton poussoir « Effacer les messages » provoque l'ouverture d'une boîte de dialogue « Voulez vous effacer les messages » avec un choix OUI ou NON.

Modifier la rubrique « Action lorsque le bouton est relâché » en « U100=1 ». Le bit U100 servira de drapeau et sera traité dans l'application d'automatisme. Ajouter dans le pupitre un objet DIAL et paramétrez le de la façon suivante :

Aspect Liens Messages

Type de boîte

- Boîte OK
- Boîte OK CANCEL
- Boîte OUI NON

Messsages

- Message prédéfini
- Liste de messages

Taille de l'objet en pixels

92 122

Largeur Hauteur

Beep

- oui
- non

Titre

Icône

-
-
-
-
- aucune

Aspect Liens Messages

Nom de la variable

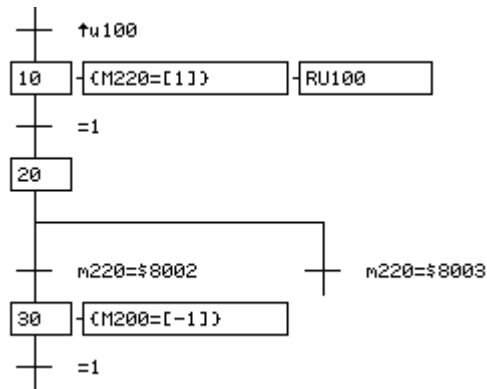
M220

Identificateur

0



Le morceau de programme suivant est ajouté à l'application d'automatisme :



L'exemple modifié se nomme « EXCLA2.GR7 » et « EXCLA2.AOF » et se trouve dans le sous répertoire « EXEMPAOF » du répertoire où est installé IRIS.

CHAPITRE N°34.

Exemple d'utilisation de l'objet ECRANCLA comme terminal

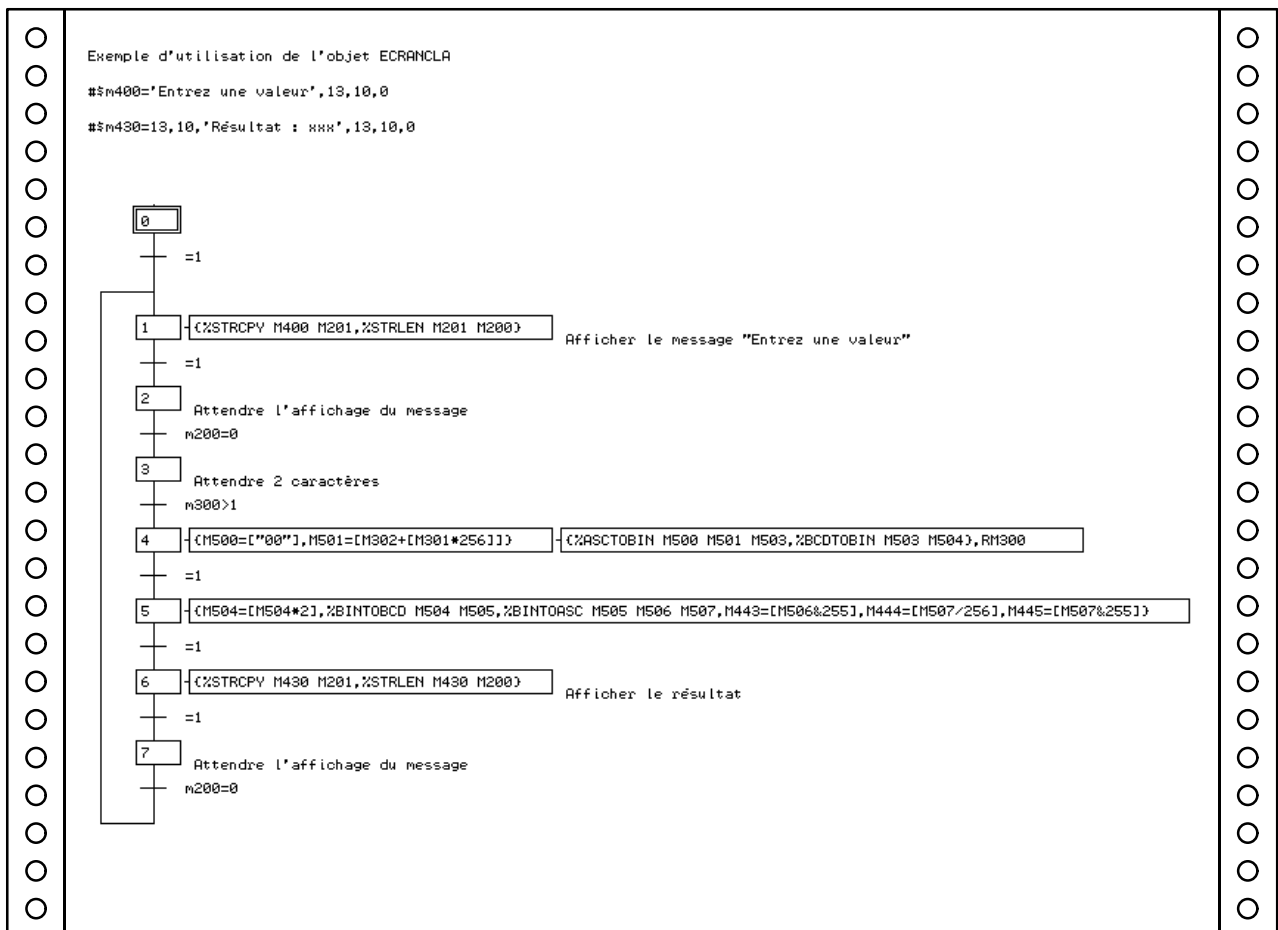
Cahier des charges :

Afficher un message « Entrez une valeur », attendre une valeur décimale tapée au clavier (deux caractères) puis afficher cette valeur multipliée par deux derrière le texte « Résultat : ».

Solution :

Un objet ECRANCLA va être utilisé en mode terminal. Les échanges de données se feront au travers de deux zones de mots commençant respectivement en m200 et en m300.

Application d'automatisme réalisée avec AUTOMGEN :



☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXTER.GR7 »

Configuration de l'objet ECRANCLA :

The image displays two screenshots of a configuration window for the 'ECRANCLA' object. The window has a tabbed interface with tabs for 'Aspect', 'Liens', 'Ljste', 'Options', and 'Messages'.
The top screenshot shows the 'Options' tab. It is divided into several sections:

- Type de l'objet**: A list of radio buttons with the following options: 'Ecran seulement', 'Ecran et clavier avec echo' (selected), 'Ecran et clavier sans echo', and 'Boîte à message, liste d'alarmes'.
- Textes**: A text input field labeled 'Texte bulle'.
- Fontes de caractères**: A section with a 'Changer' button.
- Couleurs**: Three color selection controls (Rouge, Vert, Bleu) with 'Fond' and 'Caractères' radio buttons.
- Taille de l'objet en pixels**: Two input fields for 'Largeur' (347) and 'Hauteur' (179).

The bottom screenshot shows the 'Liens' tab. It contains a list of links with two entries:

- Réception**: M200
- Emission**: M300

L'objet ECRANCLA :

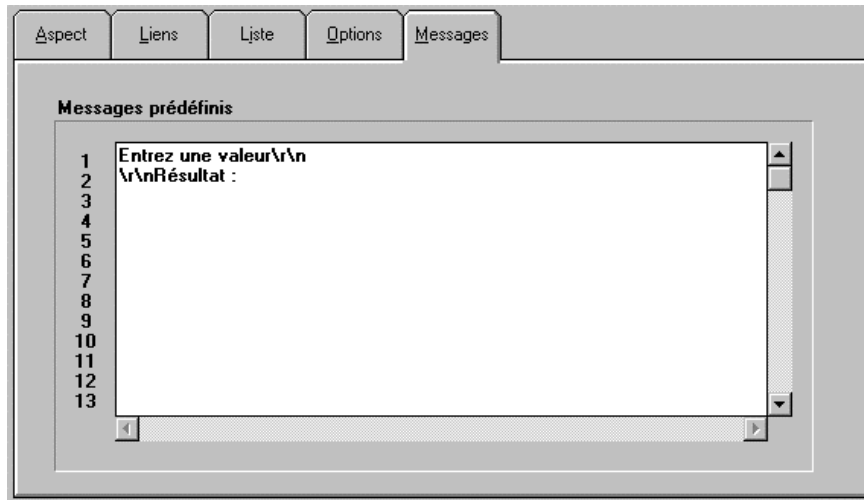


« <répertoire où est installé IRIS>\EXEMPAOF\EXTER.AOF »

Variante :

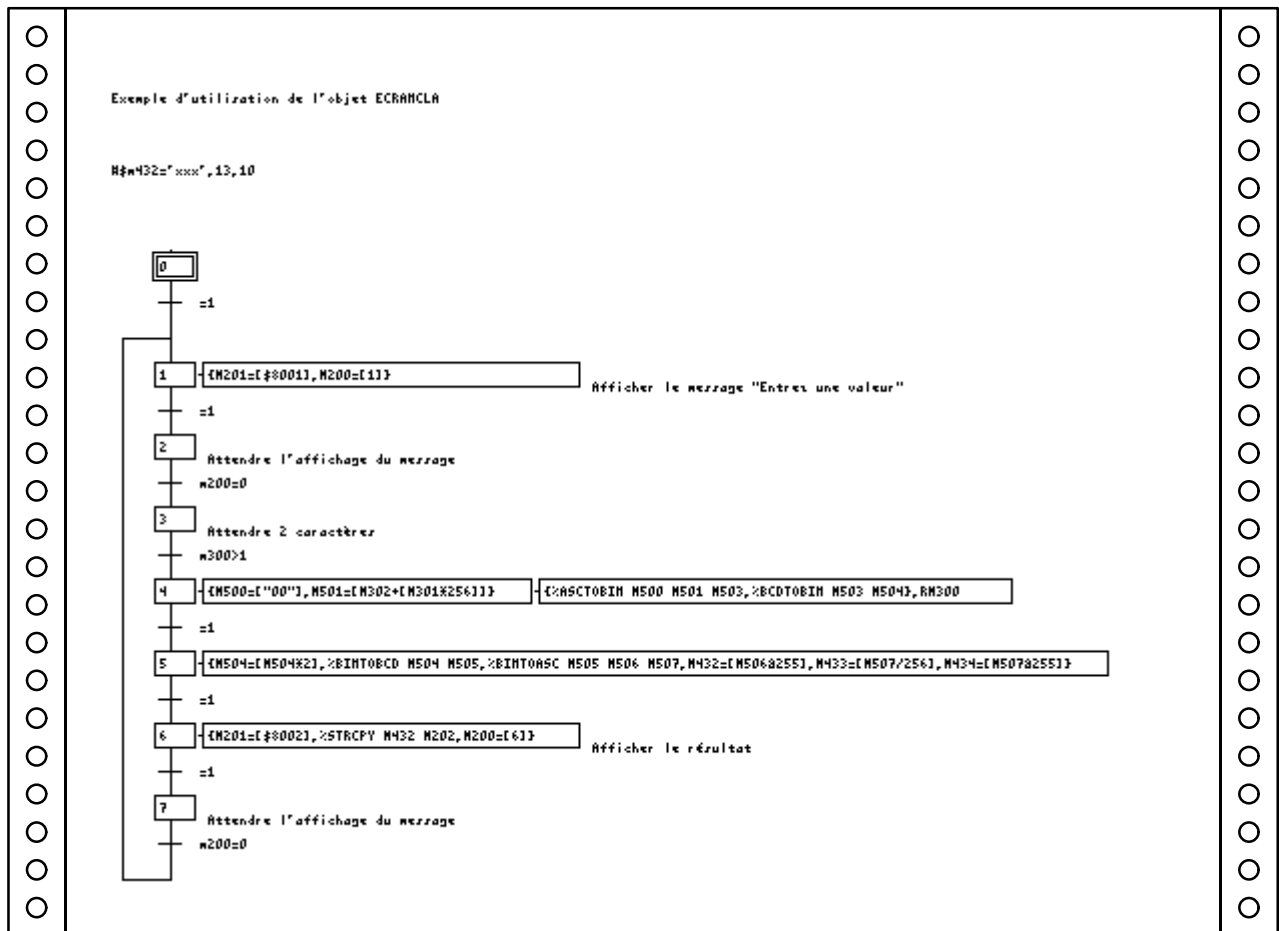
Les messages affichés sont stockés dans l'objet et non plus dans l'application d'automatisme.

Les messages sont écrits dans l'onglet message de l'objet ECRANCLA :



L'objet ECRANCLA modifié est stocké dans « EXTER2.AOF ».

L'application d'automatisme est modifiée de la façon suivante :



« <répertoire où est installé IRIS>\EXEMPAOF\EXTER2.GR7 »

CHAPITRE N°35.

Exemple d'application composée de plusieurs pages

Cet exemple va vous permettre de comprendre comment créer une application composée de plusieurs éléments : dans ce cas particulier un menu qui permet d'accéder à deux pages différentes.

Le menu sera composé d'un pupitre et de deux boutons poussoirs repérés « Application 1 » et « Application 2 ». Ce nouvel objet sera nommé « MENU.AOF ».

Les applications seront nommées « APPLI1.AOF » et « APPLI2.AOF ». Elles contiendront un objet « VALANA » et un bouton poussoir pour retourner au menu.

Pour effectuer le chaînage entre les objets nous allons utiliser le mot clé « CHAINAOF(nom d'un objet) ».

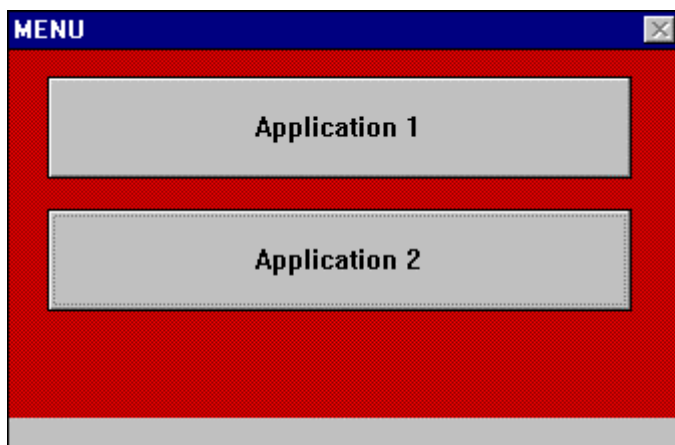
Commençons par créer l'objet menu : ouvrez un objet PUPITRE, puis insérez sur celui-ci deux objets BPVOYANT. Associez respectivement aux deux objets BPVOYANT un marquage « Application 1 » et « Application 2 » et une action (dans la rubrique « Action lorsque le bouton est relâché ») « CHAINAOF(appli1.aof) » et « CHAINAOF(appli2.aof) ».

Attention : ne cliquez pas sur les boutons poussoirs avant d'avoir sauvegardé l'objet, sinon il pourrait être définitivement perdu.

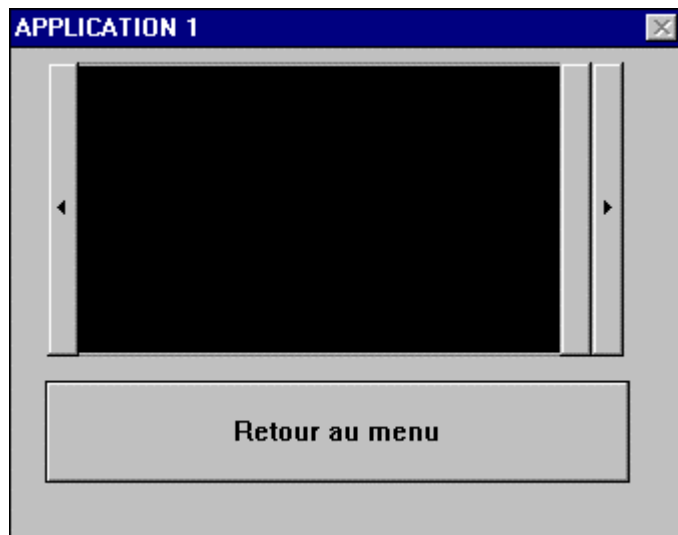
Sauvegardez l'objet « MENU.AOF ».

Pour créer les objets « APPLI1.AOF » et « APPLI2.AOF » vous pouvez, soit fermer l'objet MENU et les recréer entièrement, soit modifier l'objet MENU : suppression d'un des boutons poussoirs, ajout d'un objet VALANA, modification des propriétés du bouton poussoir restant (marquage « Retour au menu » et action « CHAINAOF(menu.aof) »).

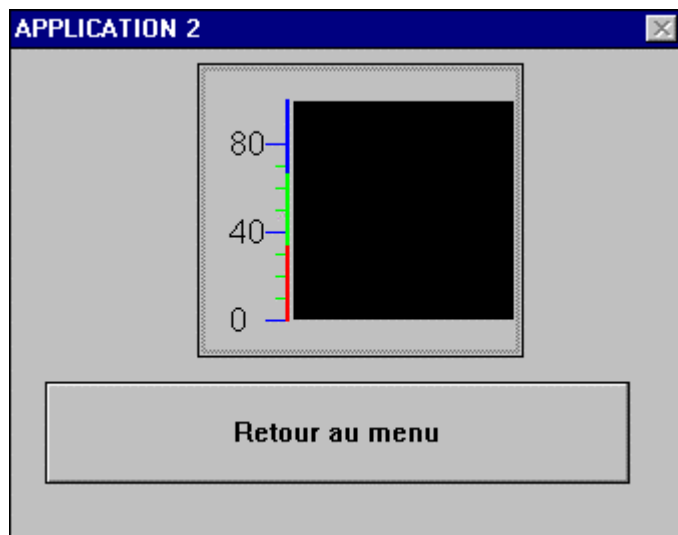
Ces trois objets se trouvent dans le sous-répertoire « EXEMPAOF » du répertoire où a été installé IRIS V3.



☞ « <répertoire où est installé IRIS>\EXEMPAOF\MENU.AOF »



« <répertoire où est installé IRIS>\EXEMPAOF\APPLI1.AOF »



« <répertoire où est installé IRIS>\EXEMPAOF\APPLI2.AOF »

CHAPITRE N°36.

Exemple d'utilisation de l'objet OBJET

Simulation d'un vérin.

Cahier des charges :

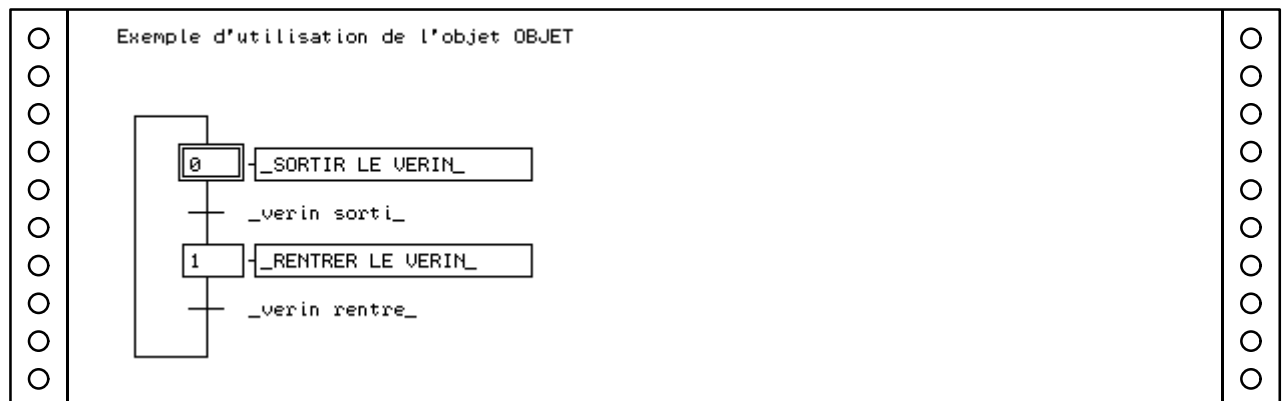
- vérin piloté par deux sorties o0 (sortir le vérin) et o1 (rentrer le vérin),
- deux entrées de fin de course i0 (vérin rentré) et i1 (vérin sorti).

Trois objets seront utilisés :

- un objet PUPITRE servant de support,
- un objet OBJET servant de corps de vérin,
- un objet OBJET servant de tige de vérin.

Solution :

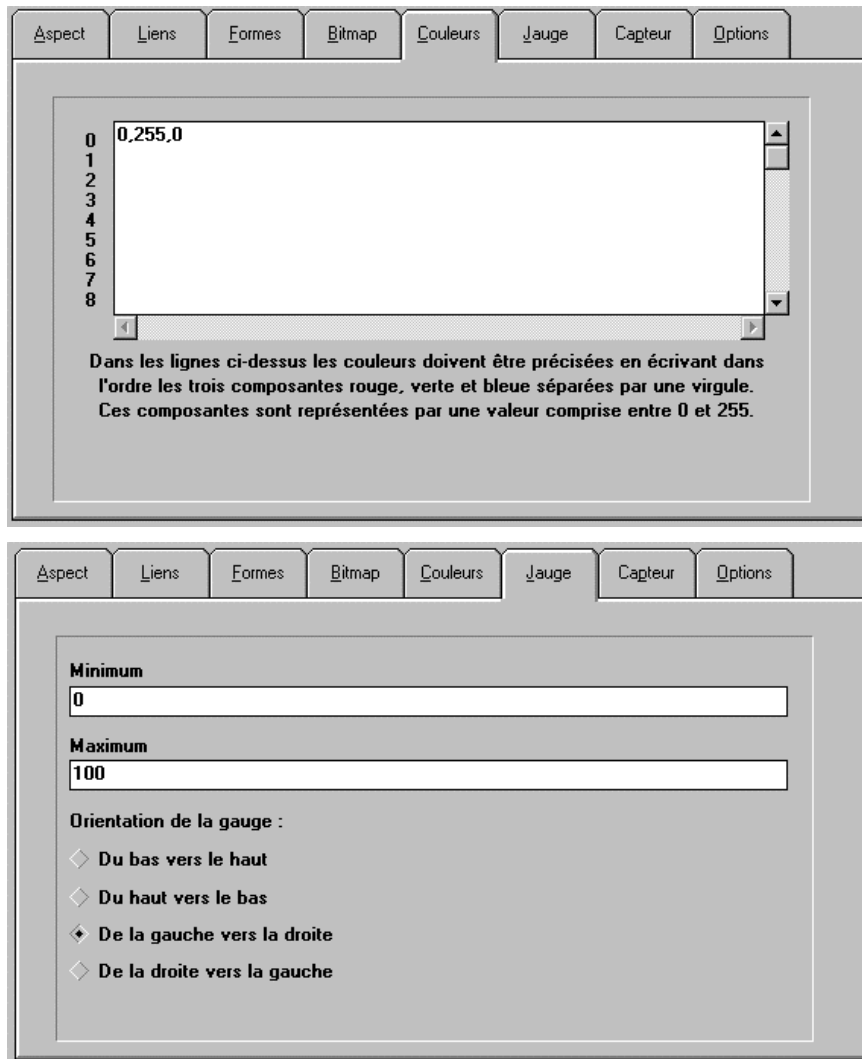
Le programme d'automatisme effectue simplement un mouvement de va et vient :



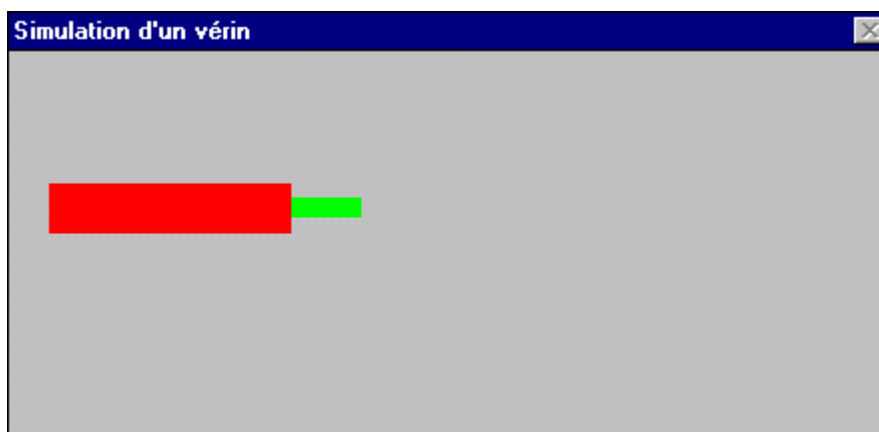
📁 « <répertoire où est installé IRIS>\EXEMPAOF\EXOBJ.GR7 »

Le corps du vérin est un objet OBJET qui reste statique, seul son aspect est configuré :

La tige du vérin est un objet OBJET configuré de la façon suivante :



Le résultat est le suivant :

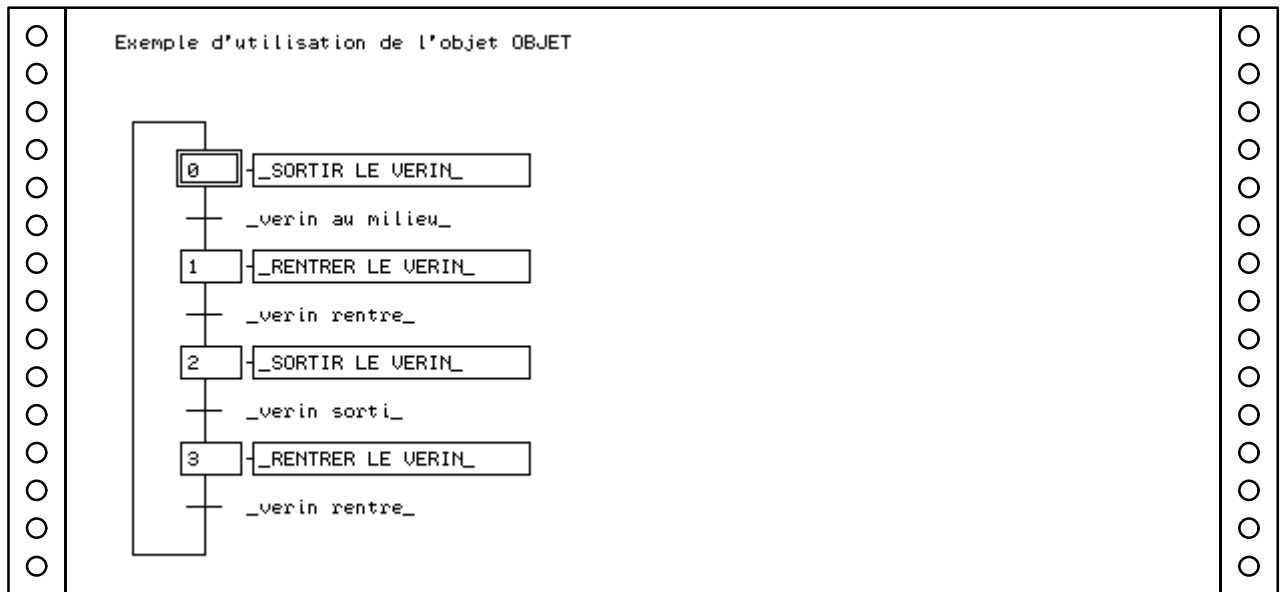


☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXOBJ.AOF »

Variante :

Une position intermédiaire doit être ajoutée sur le vérin. Pour cela nous allons utiliser deux objets supplémentaires : une pièce fixée sur la tige du vérin qui actionnera un capteur et un capteur.

L'application d'automatisme est maintenant la suivante :

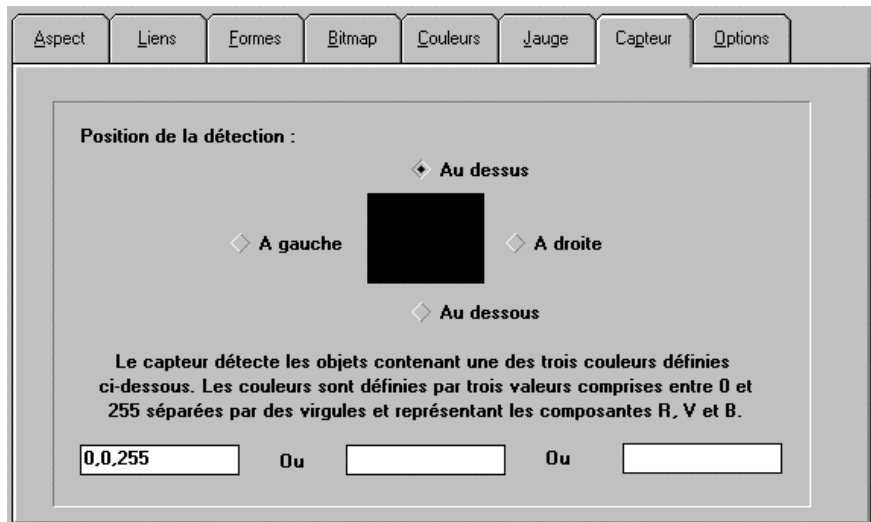


☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXOBJ2.GR7 »

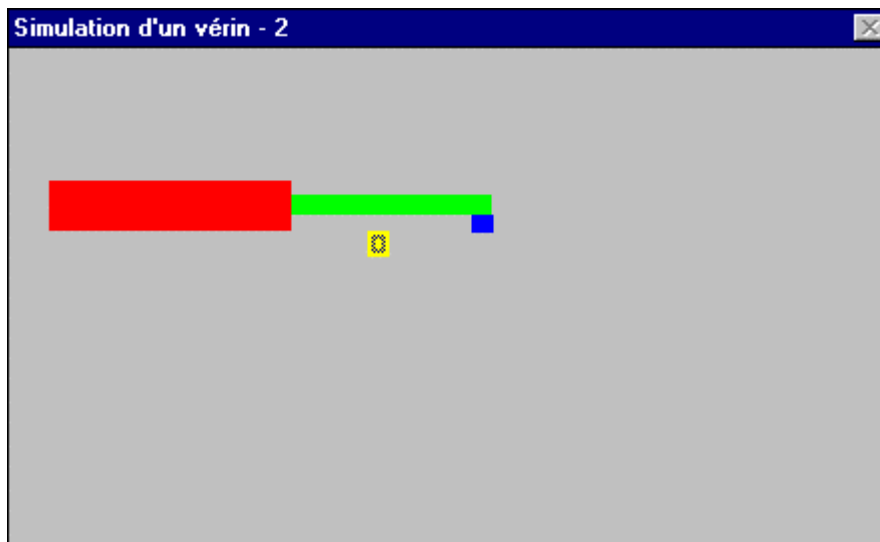
Pour lier la pièce actionnant le capteur à la tige du vérin il faut associer à la tige du vérin un identificateur : sous la rubrique « Identificateur » de l'onglet « Liens » écrire « 100 ». Pour lier la pièce à la tige, écrire sous la rubrique « Déplacement horizontal, Position » de l'onglet « Liens » : « SISTERPARAM(100,STATE) ». Ceci a pour effet de lier la pièce avec l'état de la tige du vérin.

L'objet utilisé comme capteur est paramétré comme suit :

| Aspect | | Liens | | Formes | | Bitmap | | Couleurs | | Jauge | | Capteur | | Options | |
|--|------------|------------------|-----------|---|------------|------------|-----------|---|--|-------|--|---------|--|---------|--|
| Liens | | | | | | | | | | | | | | | |
| Action | | | | Déplacement horizontal (axe des x) | | | | | | | | | | | |
| Objet cliqué | | Objet non cliqué | | Position | + Position | - Position | Capt.min | | | | | | | | |
| | | | | | | | | | | | | | | | |
| Lier en permanence avec (identificateur) | | | | Mini | Maxi | Vitesse | Capt. max | | | | | | | | |
| 0 | | | | 0 | 200 | 5 | | | | | | | | | |
| Aspect / Couleur / Remplissage | | | | | | | | Déplacement vertical (axe des y) | | | | | | | |
| Position | + Position | - Position | Capt.min | Position | + Position | - Position | Capt.min | | | | | | | | |
| | | | | | | | | | | | | | | | |
| Mini | Maxi | Vitesse | Capt. max | Mini | Maxi | Vitesse | Capt. max | | | | | | | | |
| 0 | 200 | 5 | | 0 | 200 | 5 | | | | | | | | | |
| Identificateur | | | | Capteur | | | | | | | | | | | |
| 0 | | | | I2 | | | | | | | | | | | |



Le résultat est le suivant :

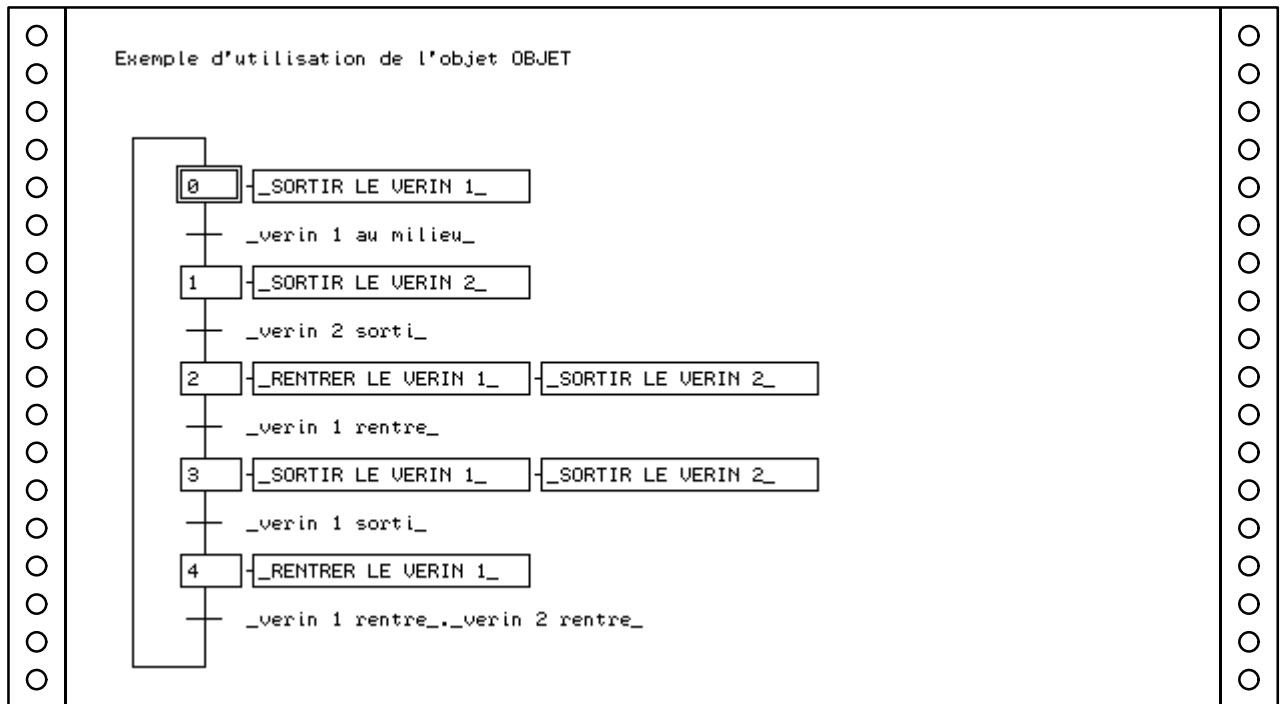


☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXOBJ2.AOF »

Deuxième variante :

Un vérin vertical fixé sur la tige du vérin horizontal est ajouté. Ce vérin est actionné par une seule sortie (O2=1 pour sortir le vérin, O2=0 pour le rentrer). Deux fins de course sont associés i3 et i4.

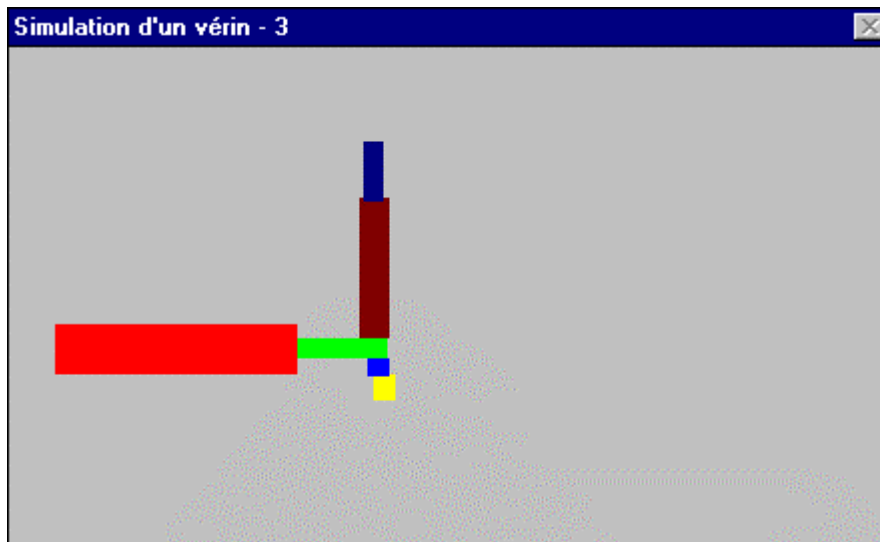
La nouvelle application d'automatisme est :



« <répertoire où est installé IRIS>\EXEMPAOF\EXOBJ3.GR7 »

Deux objets OBJET sont ajoutés : un pour le corps du vérin et un pour la tige.

Le résultat est le suivant :



« <répertoire où est installé IRIS>\EXEMPAOF\EXOBJ3.AOF »

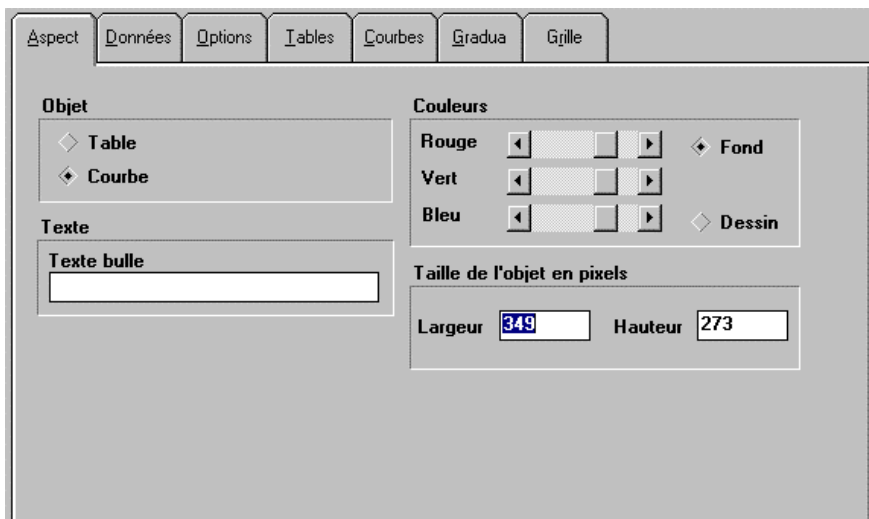
CHAPITRE N°37.

Exemples d'utilisation de l'objet ARCHIVE

Cahier des charges :

- archiver l'état de 3 mots de l'application d'automatisme (m31 à m33) toutes les secondes,
- l'état des 4 mots sera affiché sur une courbe laissant apparaître 10 secondes d'acquisition,
- 1000 valeurs seront mémorisées dans l'objet,
- les acquisitions seront archivées dans un fichier « data.txt » au format texte.

Solution :



Aspect Données Options Tables Courbes Gradua Grille

Objet

Table
 Courbe

Texte

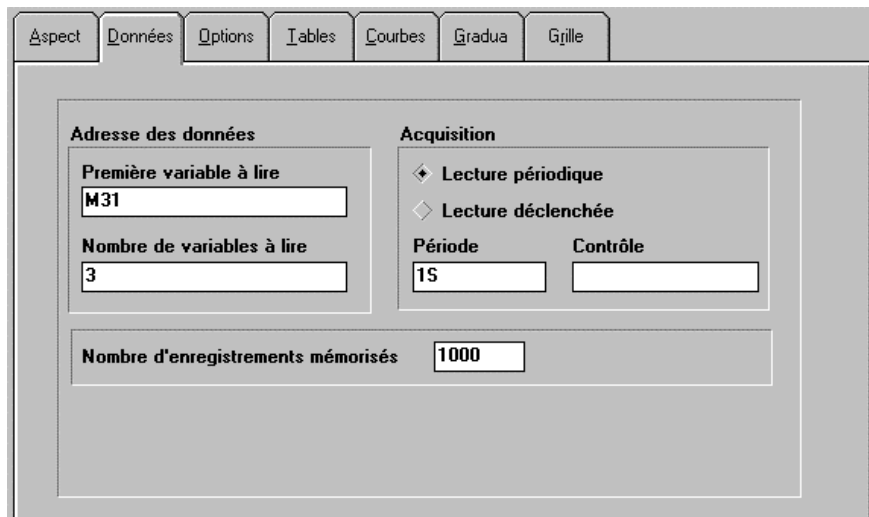
Texte bulle

Couleurs

Rouge Fond
Vert Dessin
Bleu

Taille de l'objet en pixels

Largeur Hauteur



Aspect Données Options Tables Courbes Gradua Grille

Adresse des données

Première variable à lire
Nombre de variables à lire

Acquisition

Lecture périodique
 Lecture déclenchée

Période Contrôle

Nombre d'enregistrements mémorisés

Aspect Données Options Tables Courbes Gradua Grille

Utiliser le fichier image

Utiliser le fichier d'archivage

Affichage

Afficher la date d'acquisition
 Afficher l'heure d'acquisition
 Affichage automatique des dernières valeurs

Heures
 Minutes
 Secondes
 Millisecondes

Aspect Données Options Tables Courbes Gradua Grille

Echelle

Valeur minimale
Valeur maximale

Paramètres de visualisation

Visualiser
 Marquage des valeurs sur la courbe

Affichage du temps

Couleur Police

Couleurs des tracés

| | |
|----|---------|
| 1 | 255,0,0 |
| 2 | 0,255,0 |
| 3 | 0,0,255 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |

Aspect Données Options Tables Courbes Gradua Grille

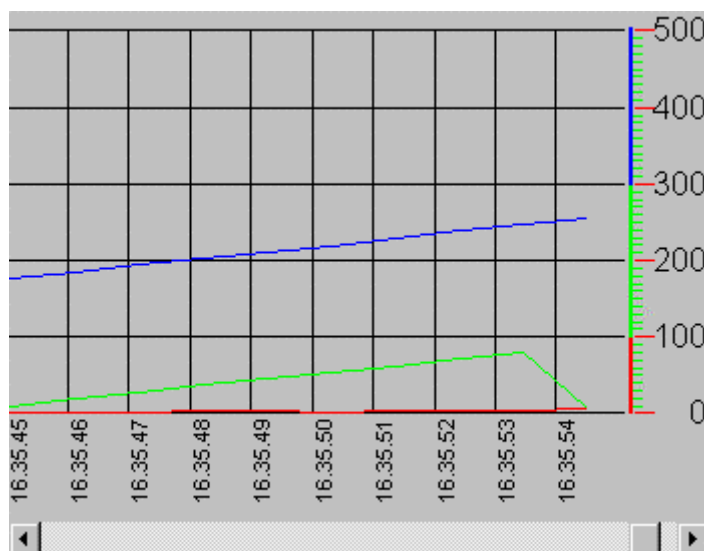
Utiliser des graduations

Valeur de départ Pas des grandes graduations Police de caractère
Valeur de fin Pas des petites graduations

| Zone No1 | Zone No2 | Zone No3 | Couleurs |
|---|---|---|--|
| Valeur de départ <input type="text" value="0"/> | Valeur de départ <input type="text" value="100"/> | Valeur de départ <input type="text" value="300"/> | Caractères <input type="text" value="0,0,0"/> |
| Valeur de fin <input type="text" value="100"/> | Valeur de fin <input type="text" value="300"/> | Valeur de fin <input type="text" value="500"/> | Grandes graduations <input type="text" value="255,0,0"/> |
| Couleur <input type="text" value="255,0,0"/> | Couleur <input type="text" value="0,255,0"/> | Couleur <input type="text" value="0,0,255"/> | Petites graduations <input type="text" value="0,255,0"/> |



Le résultat est le suivant :

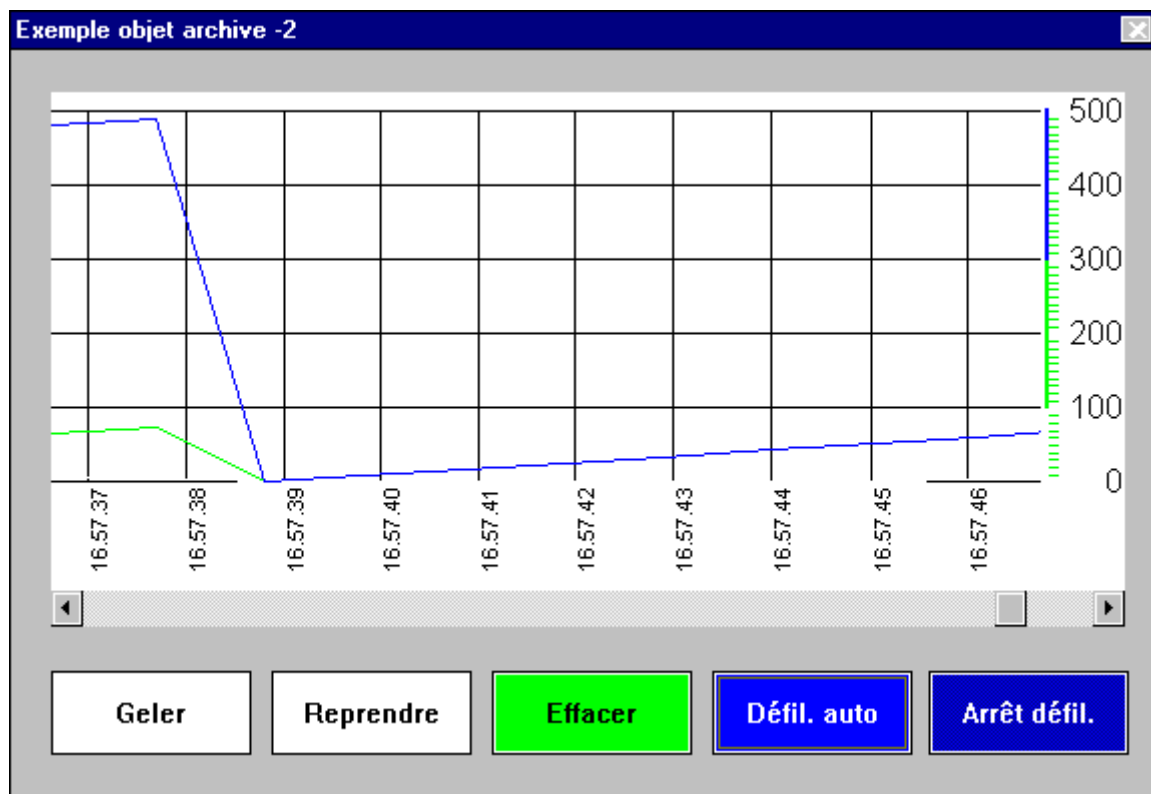


☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXARCHI.AOF »

Variante : l'objet ARCHIVE est inséré sur un pupitre est des boutons poussoirs permettent de geler les acquisitions, d'arrêter ou de valider l'affichage des dernières valeurs et d'effacer la base de données.

Le mot de contrôle (rubrique « Contrôle » de l'onglet « Données » de l'objet ARCHIVE) est utilisé pour effectuer ces différentes fonctions.

Le résultat est le suivant :



« <répertoire où est installé IRIS>\EXEMPAOF\EXARCHI2.AOF »

CHAPITRE N°38.

Exemple d'utilisation de l'objet PROG

Cahier des charges :

- l'appui sur un bouton poussoir doit provoquer l'inversion de l'état des sorties O0 à O99.

Solution :

Trois objets sont utilisés : un objet PUPITRE contenant un objet BPVOYANT et un objet PROG. Le lien entre l'appui sur le bouton poussoir et le déclenchement d'un traitement dans l'objet PROG sera effectué par une variable de l'application d'automatisme : M200.

Paramétrage de l'objet BPVOYANT :

The image shows a software configuration window for the 'BPVOYANT' object. At the top, there are three tabs: 'Aspect', 'Liens', and 'Options'. The 'Liens' tab is selected. Below the tabs, the title 'Liens' is displayed. The configuration area contains five fields:

- Action lorsque le bouton est enfoncé**: An empty text input field.
- Action lorsque le bouton est relâché**: A text input field containing the value 'M200=1'.
- Etat du voyant**: A text input field containing the value 'i0'.
- Identificateur**: A text input field containing the value '0'.

Le programme contenu dans l'objet PROG est le suivant :

```
BEGINVAR;
ULONG table[100];
UINT index;
ULONG decl;
INT result;
ENDVAR;

BEGIN;
(* lire l'état de m200 *)
result:=readvar("m200",&decl,1);
if result=0 (* pas d'erreur *)
then
  if decl=1
  then
    (* on arrive ici si m200=1 : demande d'inversion des sorties *)
    (* lecture des sorties 00 à 099 *)
    readvar("00",&table,100);
    index:=0;
    while index<100
    do
      table[index]:=1-(table[index]);
      index:=index+1;
    endwhile;
    writevar("00",&table,100);
    decl:=0;
    (* raz demande *)
    writevar("m200",&decl,1);
  endif;
endif;

END;
```

L'objet est sauvegardé sous le nom :

☞ « <répertoire où est installé IRIS>\EXEMPAOF\EXPROG.AOF »

Variante :

L'objet PROG n'est pas ouvert au démarrage de l'application, c'est l'objet BPVOYANT qui l'ouvre. L'objet PROG se ferme automatiquement en fin de traitement.

Paramétrage de l'objet BPVOYANT :

Aspect Liens Options

Liens

Action lorsque le bouton est enfoncé

Action lorsque le bouton est relâché
openaof(invsor.aof)

Etat du voyant
i0

Identificateur
0

Le pupitre et le bouton poussoir sont sauvegardés sous le nom :

« <répertoire où est installé IRIS>\EXEMPAOF\EXPROG2.AOF »

Le programme contenu dans l'objet PROG est le suivant :

```
○ BEGINVAR;
○ ULONG table[100];
○ UINT index;
○ ENDEVAR;
○
○ BEGIN;
○ (* lecture des sorties 00 à 099 *)
○ readvar("00",&table,100);
○ index:=0;
○ while index<100
○ do
○ table[index]:=1-(table[index]);
○ index:=index+1;
○ endwhile;
○ writevar("00",&table,100);
○ (* ferme l'objet courant *)
○ cmd("closeaof()");
○ END;
```

L'objet PROG est sauvegardé sous le nom :

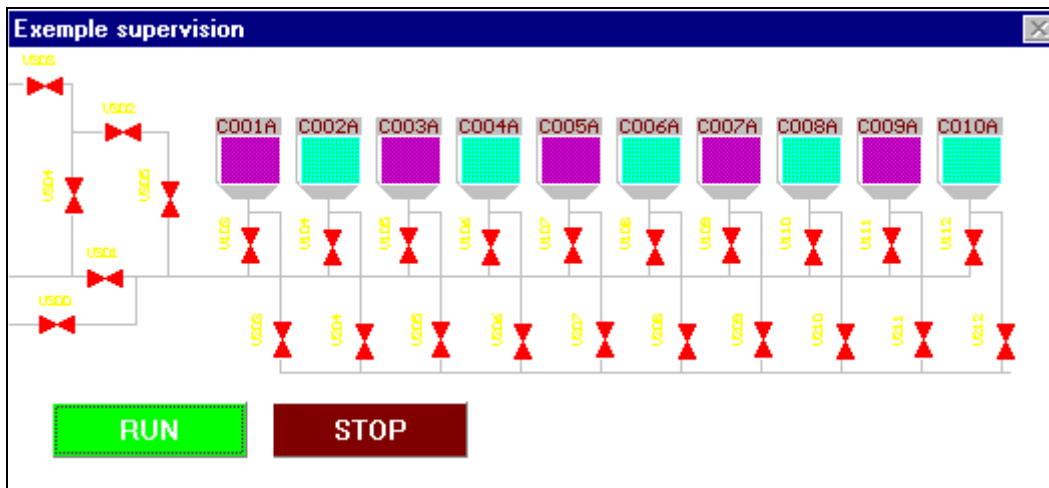
« <répertoire où est installé IRIS>\EXEMPAOF\INVSOR.AOF »

CHAPITRE N°39.

Exemples d'application de supervision

L'exemple suivant illustre la création d'une application de supervision. L'application de supervision affichera l'état de vannes et de niveaux de cuves. L'action de l'utilisateur sur les vannes aura pour effet d'inverser l'état de la vanne (ouverte ou fermée). L'état RUN/STOP de l'application d'automatisme sera également affiché et deux boutons poussoirs permettront de passer en RUN ou en STOP.

Le résultat est le suivant :



« <répertoire où est installé IRIS>\EXEMPAOF\EXSUP.AOF »

Des objets OBJET seront utilisés pour représenter les vannes. Un fichier bitmap est créé pour représenter les vannes : états ouvert (couleur verte) et état fermée (couleur rouge) :



Les objets OBJET sont paramétrés de la façon suivante :

| | | | | | | | |
|--------|-------|--------|--------|----------|-------|---------|---------|
| Aspect | Liens | Formes | Bitmap | Couleurs | Jauge | Capteur | Options |
|--------|-------|--------|--------|----------|-------|---------|---------|

| | | | | | |
|---|---|-----------------------------------|--|-----------------------------------|----------------------------------|
| Type <input checked="" type="checkbox"/> Bitmap n aspects <input type="checkbox"/> Forme n couleurs <input type="checkbox"/> Bitmap n couleurs <input type="checkbox"/> Forme jauge <input type="checkbox"/> Bitmap jauge | Couleurs Rouge <input type="text"/> <input type="text"/> <input type="text"/> Vert <input type="text"/> <input type="text"/> <input type="text"/> Bleu <input type="text"/> <input type="text"/> <input type="text"/> | | | | |
| Taille de l'objet en pixels Largeur <input type="text" value="9"/> Hauteur <input type="text" value="18"/> | Fonte de caractères <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="button" value="Changer"/> | | | | |
| Textes <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Texte <input type="text"/></td> <td style="width: 25%;">Orientation <input type="text" value="0 deg"/></td> <td style="width: 25%;">Texte d'aide <input type="text"/></td> <td style="width: 25%;">Texte bulle <input type="text"/></td> </tr> </table> Position verticale du texte : <input checked="" type="checkbox"/> centré <input type="checkbox"/> en haut <input type="checkbox"/> en bas Position horizontale du texte : <input checked="" type="checkbox"/> centré <input type="checkbox"/> à gauche <input type="checkbox"/> à droite | | Texte <input type="text"/> | Orientation <input type="text" value="0 deg"/> | Texte d'aide <input type="text"/> | Texte bulle <input type="text"/> |
| Texte <input type="text"/> | Orientation <input type="text" value="0 deg"/> | Texte d'aide <input type="text"/> | Texte bulle <input type="text"/> | | |

| | | | | | | | |
|--------|-------|--------|--------|----------|-------|---------|---------|
| Aspect | Liens | Formes | Bitmap | Couleurs | Jauge | Capteur | Options |
|--------|-------|--------|--------|----------|-------|---------|---------|

| | | | | | | | |
|--|--|----------------------------------|--|---|--|---|--|
| Liens | | | | | | | |
| Action | | Objet cliqué | | Objet non cliqué | | Déplacement horizontal (axe des x) | |
| <input style="width: 100%;" type="text" value="03=/"/> | | <input type="text"/> | | <input type="text"/> | | Position + Position - Position Capt.min <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | |
| Lier en permanence avec (identificateur) | | Mini | | Maxi | | Vitesse | |
| <input style="width: 100%;" type="text" value="0"/> | | <input type="text" value="0"/> | | <input type="text" value="200"/> | | <input type="text" value="5"/> | |
| Aspect / Couleur / Remplissage | | | | Déplacement vertical (axe des y) | | | |
| Position | | + Position | | - Position | | Capt.min | |
| <input type="text" value="03"/> | | <input type="text"/> | | <input type="text"/> | | <input type="text"/> | |
| Mini | | Maxi | | Vitesse | | Capt. max | |
| <input type="text" value="0"/> | | <input type="text" value="200"/> | | <input type="text" value="5"/> | | <input type="text"/> | |
| Identificateur | | | | Capteur | | | |
| <input style="width: 100%;" type="text" value="0"/> | | | | <input type="text"/> | | | |

| | | | | | | | |
|--------|-------|--------|--------|----------|-------|---------|---------|
| Aspect | Liens | Formes | Bitmap | Couleurs | Jauge | Capteur | Options |
|--------|-------|--------|--------|----------|-------|---------|---------|

| | | | |
|--|--|--|---------------------------------------|
| Nom du fichier | <input type="text" value="C:\AUTOM500\EXEMPA0F\MINI\V.BMP"/> | | |
| Nombre d'aspect(s) | <input type="text" value="2"/> | <input type="button" value="Parcourir"/> | <input type="button" value="Editer"/> |
| <input type="checkbox"/> Le bitmap possède des zones transparentes définies par la couleur : | | | |
| Rouge : | <input type="text" value="0"/> | <input type="text"/> | <input type="text"/> |
| Vert : | <input type="text" value="0"/> | <input type="text"/> | <input type="text"/> |
| Bleu : | <input type="text" value="0"/> | <input type="text"/> | <input type="text"/> |

Des objets VALANA sont utilisés pour les niveaux de cuves. Ils sont paramétrés de la façon suivante :

Aspect | Liens | **Bornes** | Graduations

Objet

- Barre de scrolling
- Barre graphe signé
- Barre graphe non signé
- Aiguille
- Potentiomètre

Couleurs

Rouge Fond

Vert Dessin

Bleu

Taille de l'objet en pixels

Largeur Hauteur

Orientation

Horizontale Verticale

Fond

Coloré Transparent

Texte

Texte bulle

Aspect | Liens | **Bornes** | Graduations

Variable ou symbole

Les actions de l'utilisateur sur l'objet modifient l'état de la variable

Aspect | Liens | **Bornes** | Graduations

Bornes

| Minimum | Maximum | Angle de départ | Angle de fin |
|--------------------------------|----------------------------------|---------------------------------|----------------------------------|
| <input type="text" value="0"/> | <input type="text" value="100"/> | <input type="text" value="30"/> | <input type="text" value="360"/> |

Des objets BPVOYANT sont utilisés pour les boutons poussoirs RUN et STOP. Ils sont configurés de la façon suivante :

Bouton poussoir RUN:

The screenshot shows the configuration window for the 'Bouton poussoir RUN' object. It has three tabs: 'Aspect', 'Liens', and 'Options'. The 'Options' tab is active.

- Type de l'objet:** Three radio buttons are present: 'Bouton poussoir', 'Voyant', and 'Bouton poussoir et voyant' (which is selected).
- Couleurs:** Three color selection fields (Rouge, Vert, Bleu) with color swatches and arrows. To their right are three radio buttons: 'Fond éteint', 'Fond allumé', and 'Caractères'.
- Taille de l'objet en pixels:** Two input fields: 'Largeur' (value: 100) and 'Hauteur' (value: 30).
- Fonte de caractères:** A 'Changer' button.
- Textes:** Four input fields: 'Texte' (value: RUN), 'Orientation' (value: 0 deg), 'Texte d'aide', and 'Texte bulle'.
- Position verticale du texte:** Three radio buttons: 'centré' (selected), 'en haut', and 'en bas'.
- Position horizontale du texte:** Three radio buttons: 'centré' (selected), 'à gauche', and 'à droite'.

The screenshot shows the 'Liens' tab of the configuration window for the 'Bouton poussoir RUN' object.

- Liens:** A section containing five input fields:
 - 'Action lorsque le bouton est enfoncé' (empty)
 - 'Action lorsque le bouton est relâché' (value: run)
 - 'Etat du voyant' (value: B11)
 - 'Identificateur' (value: 0)

Bouton poussoir STOP:

The screenshot shows the configuration window for the 'Bouton poussoir STOP' object. It has three tabs: 'Aspect', 'Liens', and 'Options'. The 'Options' tab is active.

- Type de l'objet:** Three radio buttons are present: 'Bouton poussoir', 'Voyant', and 'Bouton poussoir et voyant' (which is selected).
- Couleurs:** Three color selection fields (Rouge, Vert, Bleu) with color swatches and arrows. To their right are three radio buttons: 'Fond éteint', 'Fond allumé', and 'Caractères'.
- Taille de l'objet en pixels:** Two input fields: 'Largeur' (value: 100) and 'Hauteur' (value: 30).
- Fonte de caractères:** A 'Changer' button.
- Textes:** Four input fields: 'Texte' (value: STOP), 'Orientation' (value: 0 deg), 'Texte d'aide', and 'Texte bulle'.
- Position verticale du texte:** Three radio buttons: 'centré' (selected), 'en haut', and 'en bas'.
- Position horizontale du texte:** Three radio buttons: 'centré' (selected), 'à gauche', and 'à droite'.

Aspect Liens Options

Liens

Action lorsque le bouton est enfoncé

Action lorsque le bouton est relâché

stop

Etat du voyant

/B11

Identificateur

0

CHAPITRE N°40.

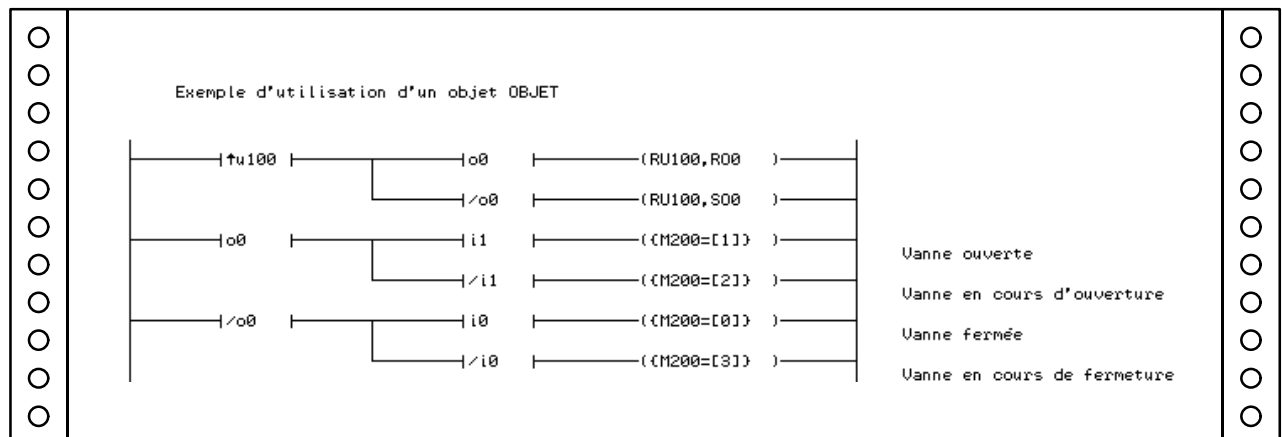
Exemple d'application de supervision

Cet exemple illustre une utilisation plus évoluée d'un objet OBJET. L'application affiche l'état d'une vanne qui peut être :

- vanne ouverte (ouverture commandée et capteur vanne ouverte vrai) : couleur verte,
- vanne fermée (fermeture commandée et capteur vanne fermée vrai) : couleur rouge,
- vanne en cours d'ouverture (ouverture commandée et capteur vanne ouverte faux) : couleur bleue,
- vanne en cours de fermeture (fermeture commandée et capteur vanne fermée faux) : couleur violette.

L'utilisateur peut inverser l'état de la vanne en cliquant dessus.

L'application d'automatisme gère l'état de la vanne :



« <répertoire où est installé IRIS>\EXEMPAOF\EXSUP2.GR7 »



« <répertoire où est installé IRIS>\EXEMPAOF\EXSUP2.AOF »

L'objet OBJET est paramétré de la façon suivante :

| | | | | | | | |
|--|-------|---|--------|---|-------|----------------------------------|---------|
| Aspect | Liens | Formes | Bitmap | Couleurs | Jauge | Capteur | Options |
| Type | | Couleurs | | | | | |
| <input checked="" type="checkbox"/> Bitmap n aspects <input type="checkbox"/> Bitmap n couleurs <input type="checkbox"/> Bitmap jauge | | <input type="checkbox"/> Forme n couleurs <input type="checkbox"/> Forme jauge | | Rouge <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | | | |
| | | | | Vert <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | | | |
| | | | | Bleu <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | | | |
| Taille de l'objet en pixels | | | | Fonte de caractères | | | |
| Largeur <input type="text" value="9"/> | | Hauteur <input type="text" value="18"/> | | <input type="button" value="Changer"/> | | | |
| Textes | | | | | | | |
| Texte <input type="text"/> | | Orientation <input type="text" value="0 deg"/> | | Texte d'aide <input type="text"/> | | Texte bulle <input type="text"/> | |
| Position verticale du texte : <input checked="" type="checkbox"/> centré <input type="checkbox"/> en haut <input type="checkbox"/> en bas | | | | | | | |
| Position horizontale du texte : <input checked="" type="checkbox"/> centré <input type="checkbox"/> à gauche <input type="checkbox"/> à droite | | | | | | | |

| | | | | | | | |
|---|-------|--|--------|---|-------|--|---------|
| Aspect | Liens | Formes | Bitmap | Couleurs | Jauge | Capteur | Options |
| Liens | | | | | | | |
| Action | | | | Déplacement horizontal (axe des x) | | | |
| Objet cliqué <input type="text" value="U100=1"/> | | Objet non cliqué <input type="text"/> | | Position <input type="text"/> | | + Position <input type="text"/> | |
| | | | | - Position <input type="text"/> | | Capt.min <input type="text"/> | |
| Lien en permanence avec (identificateur) <input type="text" value="0"/> | | | | Mini <input type="text" value="0"/> | | Maxi <input type="text" value="200"/> | |
| | | | | Vitesse <input type="text" value="5"/> | | Capt. max <input type="text"/> | |
| Aspect / Couleur / Remplissage | | | | Déplacement vertical (axe des y) | | | |
| Position <input type="text" value="M200"/> | | + Position <input type="text"/> | | Position <input type="text"/> | | + Position <input type="text"/> | |
| | | - Position <input type="text"/> | | - Position <input type="text"/> | | Position <input type="text"/> | |
| | | Capt.min <input type="text"/> | | | | Capt.min <input type="text"/> | |
| Mini <input type="text" value="0"/> | | Maxi <input type="text" value="200"/> | | Mini <input type="text" value="0"/> | | Maxi <input type="text" value="200"/> | |
| | | Vitesse <input type="text" value="5"/> | | | | Vitesse <input type="text" value="5"/> | |
| | | Capt. max <input type="text"/> | | | | Capt. max <input type="text"/> | |
| Identificateur <input type="text" value="0"/> | | | | Capteur <input type="text"/> | | | |

| | | | | | | | |
|---|-------|---|--------|---------------------------------------|-------|---------|---------|
| Aspect | Liens | Formes | Bitmap | Couleurs | Jauge | Capteur | Options |
| Bitmap | | | | | | | |
| Nom du fichier <input type="text" value="C:\AUTOM500\EXEMPAOF\MINIIVV4S.BMP"/> | | | | | | | |
| Nombre d'aspect(s) <input type="text" value="4"/> | | <input type="button" value="Parcourir"/> | | <input type="button" value="Editer"/> | | | |
| <input checked="" type="checkbox"/> Le bitmap possède des zones transparentes définies par la couleur : | | | | | | | |
| Rouge : <input type="text" value="0"/> | | <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | | | | | |
| Vert : <input type="text" value="0"/> | | <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | | | | | |
| Bleu : <input type="text" value="0"/> | | <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> | | | | | |

CHAPITRE N°41.

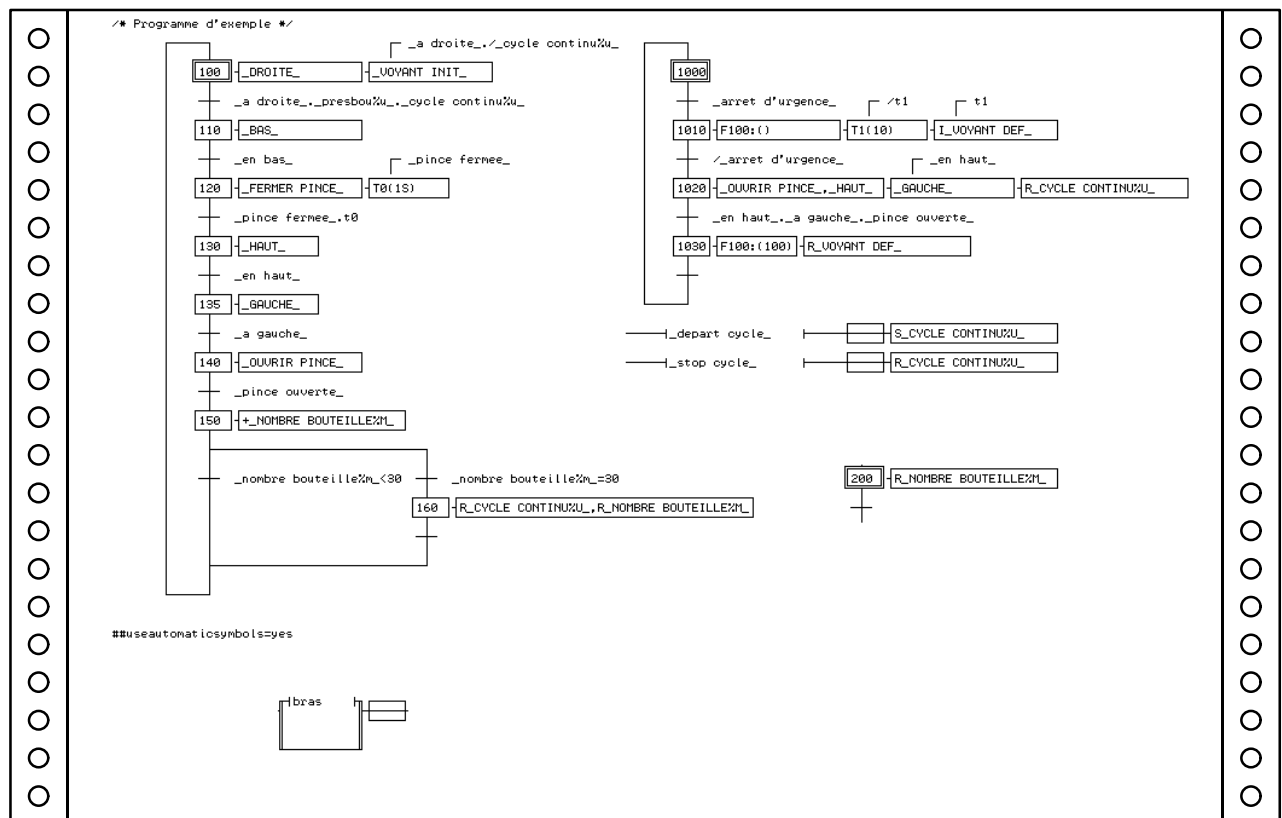
Exemple de simulation d'une partie opérative

Exemple BRAS : simulation d'un bras manipulateur

L'objet « BRAS.AOF » et le programme AUTOMGEN « BRAS.GR7 » contenus dans le sous répertoire « EXEMPAOF » du répertoire où est installé IRIS V3 illustre la simulation d'une partie opérative.

Pour essayer cet exemple fixez le répertoire courant d'AUTOMGEN vers le sous-répertoire « EXEMPAOF », compilez le folio « BRAS.GR7 » puis en exécution ouvrez l'objet « BRAS.AOF ».

Le programme d'automatisme de l'exemple BRAS réalisé avec AUTOMGEN :



« <répertoire où est installé IRIS>\EXEMPAOF\BRAS.GR7 »

La gestion du déplacement de la bouteille est réalisée en langage littéral dans le bloc fonctionnel BRAS.LIB :

```

#begin_st

_tempo%m:=_tempo%m+1;
if _tempo%m=200
  then
    _presbou%u:=0;
    _tempo%m:=0;
    if _saisbout%u_
      then
        (* Bouteille lachée par la pince *)
        if /_pince fermee_ then
          _saisbout%u:=0; _posboutcmd%m:=16#f001;
          endif;
        else
          (* Présence bouteille sur le tapis 1 *)
          if _posboutx%m=-220 and _posbouty%m=0
            then
              _presbou%u:=1;
            else
              if _posboutx%m>>-219 (* Zone du tapis 1 *)
                then
                  if _posbouty%m<<0 then (* Chute sur le tapis 1 *)
                    _posbouty%m:=_posbouty%m+10; endif;
                  if _posbouty%m>>0 then _posbouty%m:=0; endif;
                  if _posbouty%m=0 (* Avance sur le tapis 1 *)
                    then
                      _posboutx%m:=_posboutx%m-5;
                    endif;
                  else
                    if _posboutx%m<<-399 (* Zone du tapis 2 *)
                      then
                        if _posbouty%m<<-90 then (* Chute sur tapis 2 *)
                          _posbouty%m:=_posbouty%m+10; endif;
                        if _posbouty%m>>-90 then _posbouty%m:=-90;
                        endif;
                        if _posbouty%m=-90 (* Avance sur le tapis 2 *)
                          then
                            _posboutx%m:=_posboutx%m-10;
                          endif;
                        else
                          (* chute au milieu *)
                          _posbouty%m:=_posbouty%m+10;
                        endif;
                      endif;
                    endif;

```

```

    endif;

    (* Disparition de la bouteille *)
    if _posboutx%m_<<-600 or _posbouty%m_>>200
    then
        _posboutx%m_:=0;
        _posbouty%m_:=0;
    endif;

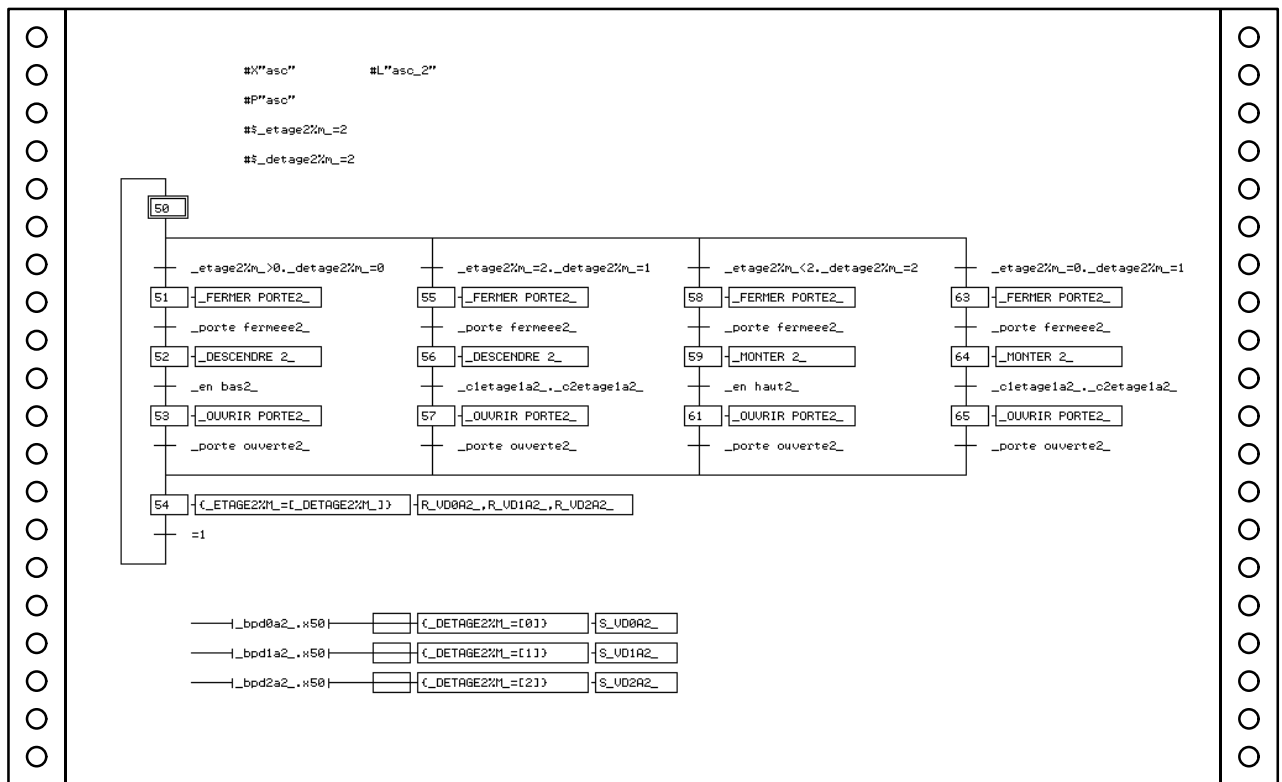
    (* Saisie de la bouteille par le bras *)
    if _pince fermee_._en bas_._a droite_._posboutx%m_=-220._posbouty%m_=0
    then
        _saisbout%u_:=1;
        _posboutcmd%m_:=16#f000;
        _posboutid%m_:=4000;
    endif;
endif;

endif;

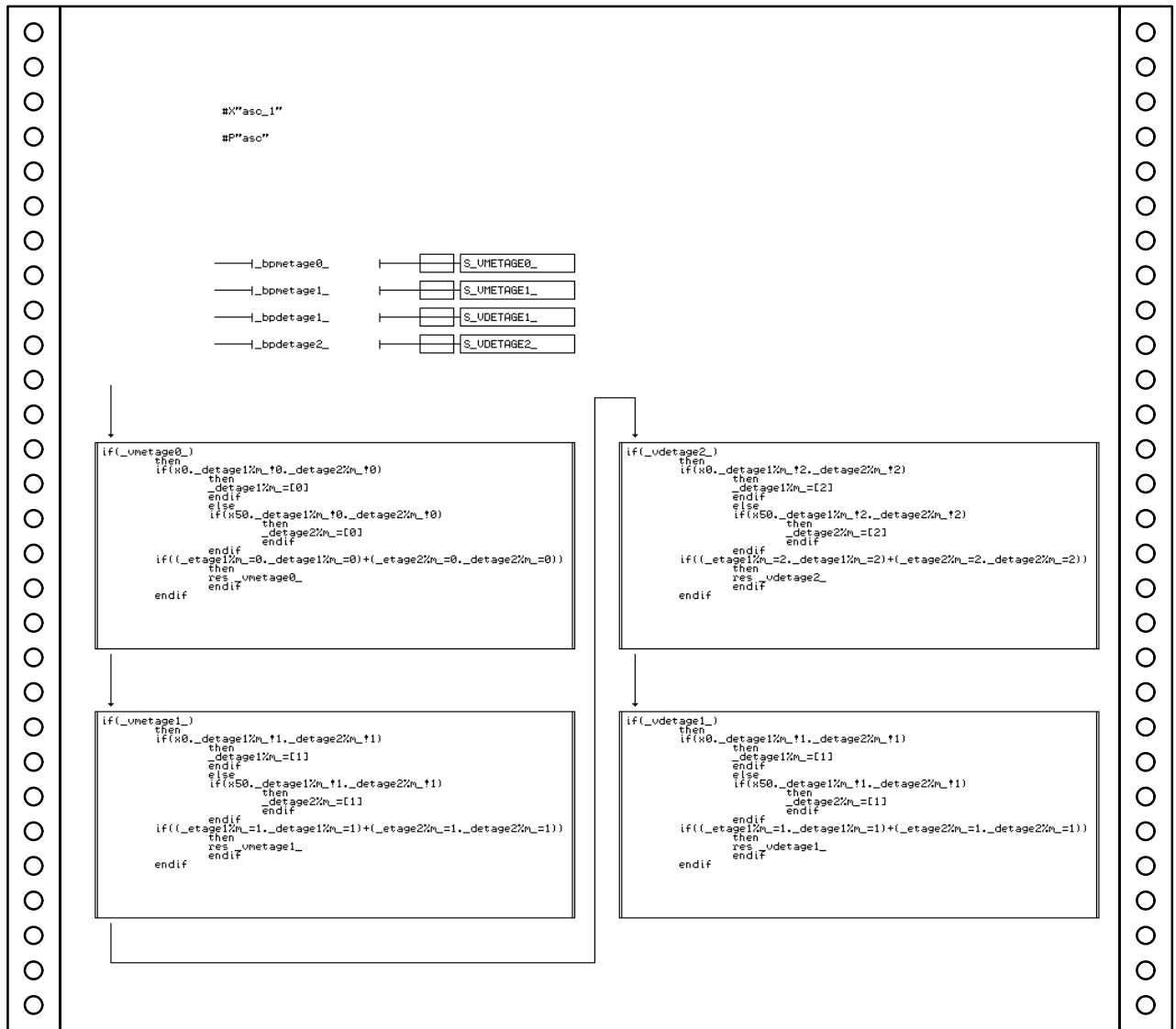
#end_st

```

☞ « <répertoire où est installé IRIS>\EXEMPAOF\BRAS.LIB »

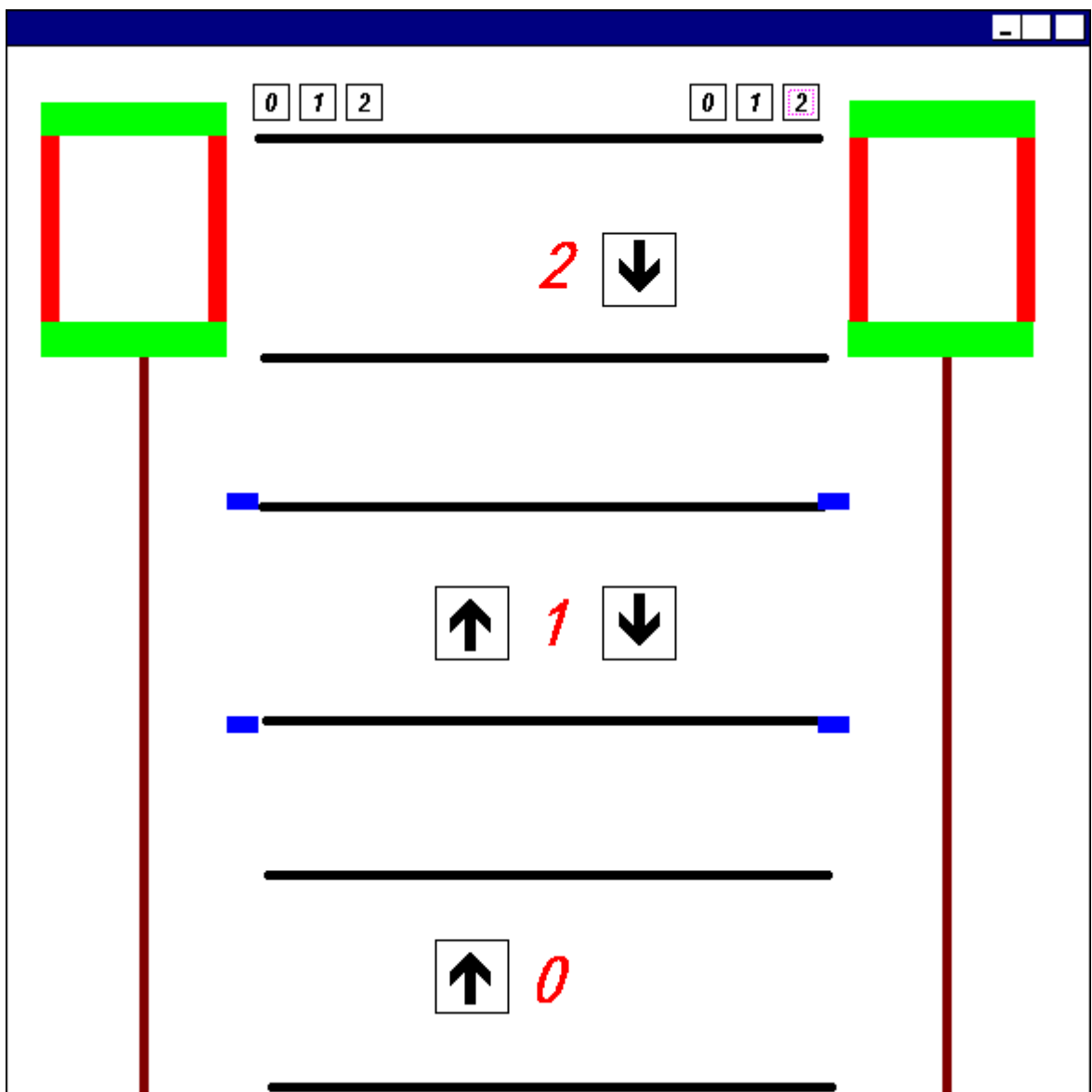


« <répertoire où est installé IRIS>\EXEMPAOF\ASC_1.GR7 »



« <répertoire où est installé IRIS>\EXEMPAOF\ASC_2.GR7 »

L'application de simulation de la partie opérative :



☞ « <répertoire où est installé IRIS>\EXEMPAOF\ASC.AOF »