Terminaux opérateurs

Dialog 80 et Dialog 640
Guide d'utilisation

SERAD SA
271, route des crêtes
44440 TEILLE – France
☎ +33 (0)2 40 97 24 54
📠 +33 (0)2 40 97 27 04
💻 http://www.serad.fr
✉ info@serad.fr

SOMMAIRE

1- INTRODUCTION

1-1- Présentation générale

Les terminaux opérateurs dialog 80 et dialog 640 servent à superviser et à contrôler une installation ou une machine commandée par un périphérique quelconque (Commande numérique, automate, variateur intelligent, etc…). Chaque terminal est équipé d'une liaison série de type RS232 et en option de l'un des types de liaisons suivantes : RS422, RS485 ou CANBus. Le terminal utilise l'une de ses liaisons pour communiquer avec un périphérique.

La gamme « terminal intelligent » est la composition des terminaux opérateurs dialog 80 ou dialog 640 associé avec un système d'exploitation et le logiciel DWIN.

Le logiciel DWIN permet de programmer simplement et rapidement sur PC les pages écran des dialog 80 et 640 et leurs enchaînements. La définition des pages écran permet à l'utilisateur d'adjoindre des objets de type textes figés ou dynamiques, des champs d'affichage de variables numériques ou alphanumériques et des champs de saisies numériques ou alphanumériques. Le dialog 640 possède en outre des objets graphiques tels que les lignes horizontales ou verticales et les cadres. Le transfert du programme se fait par une liaison série RS232.

Les terminaux peuvent communiquer avec un périphérique au moyen des protocoles Modbus ou CANOpen. L'échange et le mode d'échange (lecture ou écriture) des variables est directement paramétrable par le logiciel DWIN.

1-2- Présentation du dialog 80

Ecran

- Afficheur LCD 4 lignes de 20 caractères avec rétroéclairage par leds
- Fenêtre d'affichage 74×23 mm
- Affichage normal, clignotant
- Jeux de caractères ASCII

Clavier

- 28 touches à effet tactile
- 4 touches de fonctions dynamiques
- 6 touches de fonctions relégendables
- Touches de contrôle et de défilement
- Touche d'aide et d'affichage des alarmes
- Pavé numérique et alphanumérique
- 8 leds de visualisation d'état
- Buzzer

Performances

- Processeur 16 bits
- 512Ko de mémoire flash
- 128Ko de mémoire ram sauvegardée
- Un port de communication série RS232
- Un port de communication série optionnel RS422 ou RS485

- Bus de terrain optionnel CANBUS

Caractéristiques techniques

- Alimentation 24Vdc
- Consommation 4W
- Température de service 0 à 45°C
- Température de stockage –20 à 70°C
- Indice de protection face avant IP65

1-3- Présentation du dialog 640

Ecran

- Afficheur LCD monochrome avec rétroéclairage fluorescent
- Fenêtre d'affichage 122×66 mm
- Affichage normal, inversé, clignotant
- Jeux de caractères ASCII
- Résolution 240×128 pixels en mode graphique
- 4 tailles de caractères en mode texte pouvant être utilisées simultanément :
  - 3×4 mm 16 lignes de 40 caractères
  - 4×7 mm 9 lignes de 30 caractères
  - 5×8 mm 8 lignes de 26 caractères
  - 7×10 mm 6 lignes de 17 caractères

Clavier

- 33 touches à effet tactile
- 6 touches de fonctions dynamiques
- 6 touches de fonctions relégendables
- Touches de contrôle et de défilement
- Touche d'aide et d'affichage des alarmes
- Pavé numérique et alphanumérique
- 8 leds de visualisation d'état
- Buzzer

Performances

- Processeur 16 bits
- 512Ko de mémoire flash
- 128Ko de mémoire ram sauvegardée
- Un port de communication série RS232
- Un port de communication série optionnel RS422 ou RS485
- Bus de terrain optionnel CANBUS

Caractéristiques techniques

- Alimentation 24Vdc

- Consommation 6W
- Température de service 0 à 45°C
- Température de stockage –20 à 70°C
- Indice de protection face avant IP65

1-4- Présentation du logiciel DWIN

Le logiciel DWIN est définit par un éditeur plein page et de plusieurs outils pour une mise en œuvre conviviale. Cet éditeur plein page permet d'observer sur l'écran le résultat de ce qu'affichera le terminal. Les caractéristiques du logiciel sont les suivantes :

- Gestion de 1 à 4 langues
- Gestion de 200 pages écran pour le dialog 640 avec insertion de 50 objets maxi par page
- Gestion de 600 pages écran pour le dialog 80 avec insertion de 10 objets maxi par page
- Choix de 4 polices de caractères, mode normal ou inversé pour le dialog 640
- Définition et nombre d'objets :
    - 1000 textes figés, lignes verticales, lignes horizontales, cadres
    - 1000 textes dynamiques
    - 1000 champs d'affichage de variable numérique ou alphanumérique
    - 1000 champs de saisie de variable numérique ou alphanumérique
- Affichages et saisies au format :
    - décimal
    - hexadécimal
    - binaire
    - ascii
- Gestion de page d'aide avec indicateur à led
- Gestion de page alarme avec indicateur à led :
- 320 alarmes paramétrables maximum
- Paramétrage de chaque alarme : avec acquittement ou non, page d'aide associée
- Navigation sur les différents champs de saisie d'une page avec les touches fléchées
- Déclaration des touches de fonction au niveau local ou global
- Touches de fonctions programmables :
    - fonction poussoir
    - fonction interrupteur ...
- Indicateurs touches de fonctions à leds paramétrables :
    - allumées
    - éteintes
    - clignotantes
- Gestion des boîtes de dialogue intégrées :
    - boîte confirmation

- ⇨ boîte code d'accès

- Gestion de 10 codes d'accès paramétrables
- Gestion de 3000 textes figés pour le dialog 80 et 1500 pour le dialog 640
- Gestion de 3000 textes dynamiques pour le dialog 80 et 1500 pour le dialog 640
- Gestion de 5000 variables générales pour l'échange avec les périphériques :
  - ⇨ type word ou dword
  - ⇨ format numérique ou alphanumérique
- Gestion de 10000 variables recettes (exemple : 200 recettes de 50 éléments)

2- INSTALLATION / MISE EN OEUVRE

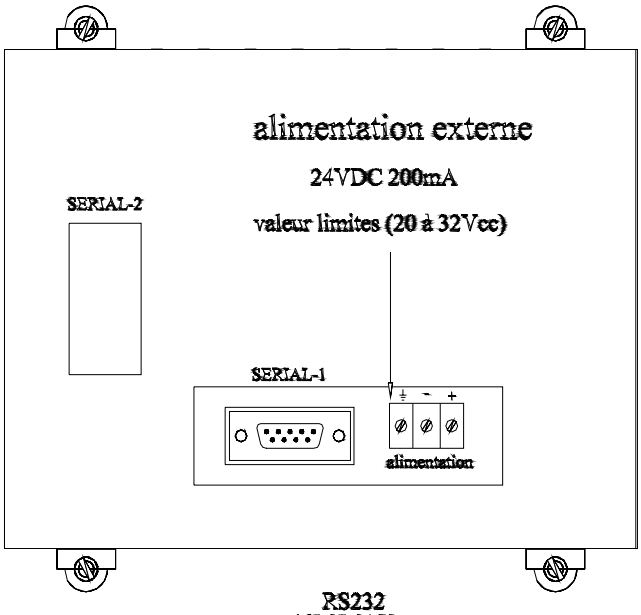
2-1- Conditions d'utilisation

- Tension d'alimentation de 20 à 32Vdc
- Consommation 6W pour le dialog640 et 4W pour le dialog80
- Température de service 0 à 45°C
- Température de stockage –20 à 70°C
- Indice de protection face avant IP65

2-2- RACCORDEMENT

2-2-1- Dialog80

A) Connexion du dialog80



RS232
10M MAX.

OPTION

SERIAL-1 SUBD 9PTS MALE	
PIN	RS232
1	
2	RXD
3	TXD
4	
5	GND
6	
7	
8	
9	

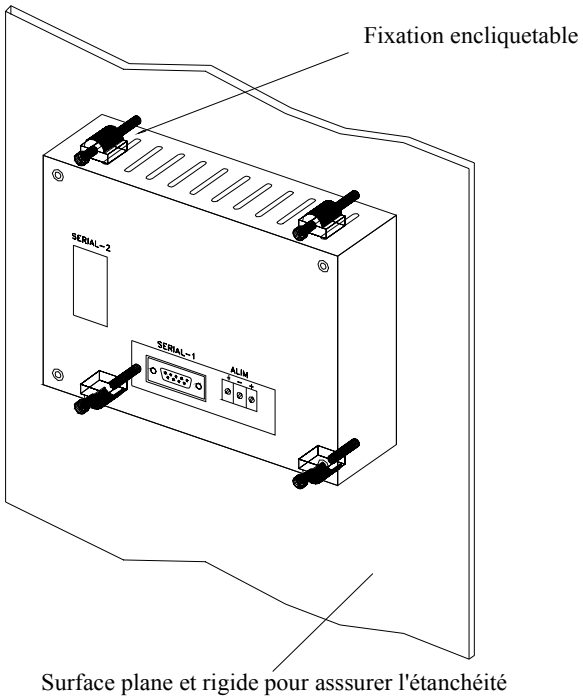
NON ISOLEE

SERIAL-2 SUBD 9PTS FEMELLE			
PIN	RS422	RS485	CANBUS
1			
2			
3	RX-		
4	RX+		
5	GND	GND	GND
6			
7	TX-	TRX-(B)	CANL
8	TX+	TRX+(A)	CANH
9			

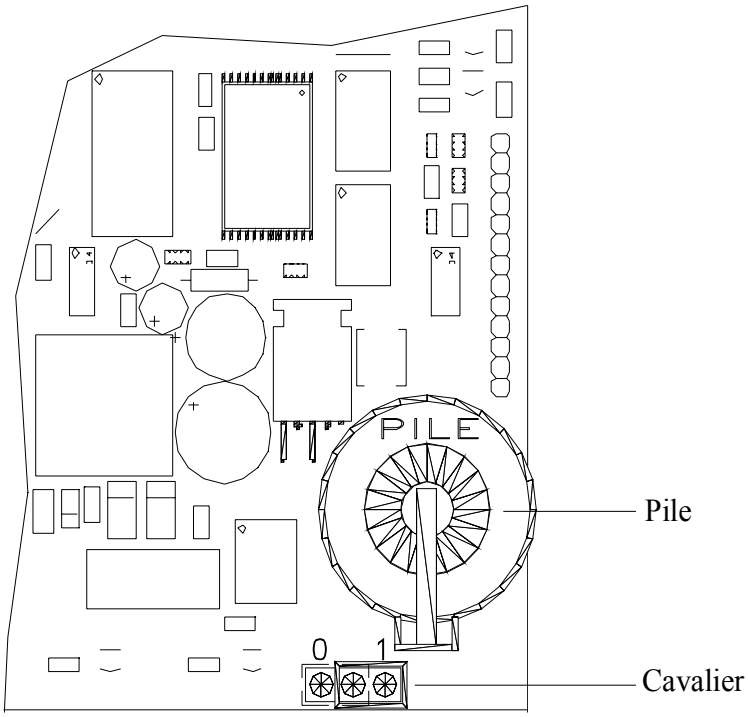
ISOLEE

Sur les modules RS422 et RS485 : Cavalier de validation des résistances (120Ω) de terminaison. Par défaut : validation des résistances.

B) Montage du dialog80



C) Montage de la pile de sauvegarde pour un dialog80



Pile CR2450 : 3V – 500 mA

- Ouvrir le capot arrière
- Insérer la pile dans son support – Vérifier la polarité
- Mettre le cavalier en position 1

Attention : Pour une version sans la pile, le cavalier doit être en position 0

D) Etiquette relégendable

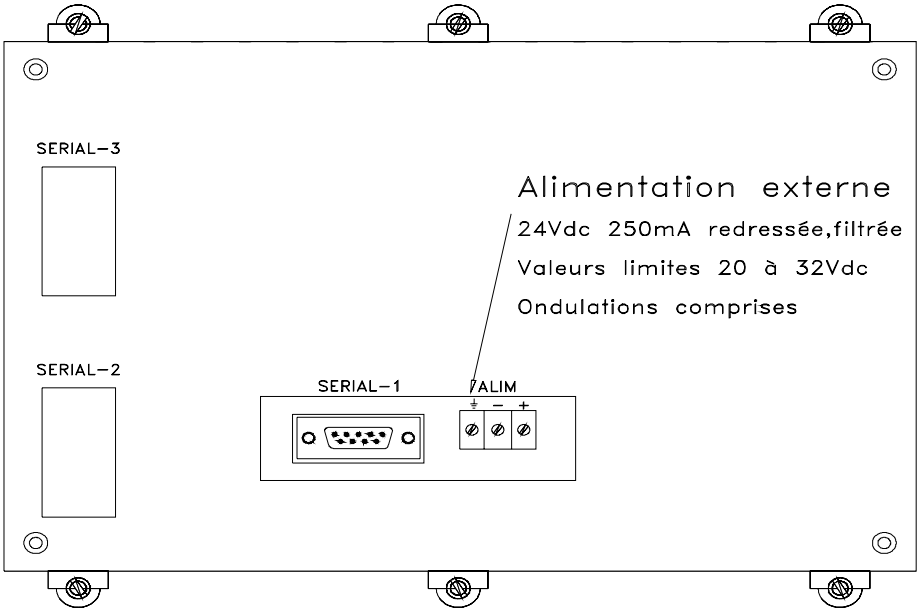
Les touches F1 à F6 peuvent être relégendées. Pour ce faire, il faut utiliser les étiquettes vierges livrées et inscrire les nouvelles affectations.

- Ouvrir le couvercle arrière du terminal
- Retirer l'étiquette déjà en place se situant en bas à gauche
- Introduire la nouvelle étiquette et refermer le couvercle

Attention : l'appareil doit être hors tension

2-2-2- Dialog640

A) Connexion du dialog640



LIASON RS232
L=10M MAX.

OPTION

SERIAL-1 SUBD 9PTS MALE	
PIN	RS232
1	
2	RXD
3	TXD
4	
5	GND
6	
7	
8	
9	

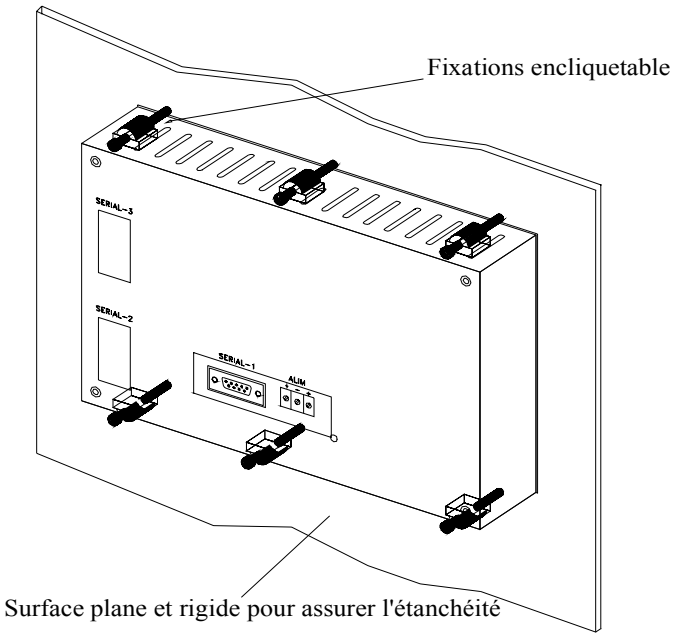
NON ISOLEE

SERIAL-2 SUBD 9PTS FEMELLE			SERIAL-3 SUBD 9PTS FEMELLE
PIN	RS422	RS485	CANBUS
1			
2			
3	RX-		
4	RX+		
5	GND	GND	GND
6			
7	TX-	TRX-(B)	CANL
8	TX+	TRX+(A)	CANH
9			

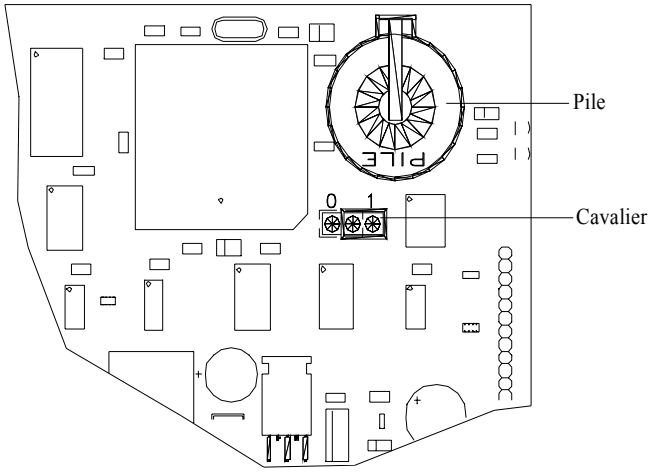
ISOLEE

Sur les modules RS422 et RS485 : Cavalier de validation des résistances (120Ω) de terminaison. Par défaut : validation des résistances.

B) Montage du dialog640



C) Montage de la pile de sauvegarde pour un dialog640



Pile CR2450 : 3V – 500 mA

- Ouvrir le capot arrière
- Insérer la pile dans son support – Vérifier la polarité
- Mettre le cavalier en position 1

Attention : Pour une version sans la pile, le cavalier doit être en position 0

D) Etiquette relégendable

Les touches F7 à F12 peuvent être relégendées. Pour ce faire, il faut utiliser les étiquettes vierges livrées et inscrire les nouvelles affectations.

- Ouvrir le couvercle arrière du terminal
- Retirer l'étiquette déjà en place se situant en bas à gauche
- Introduire la nouvelle étiquette et refermer le couvercle

Attention : l'appareil doit être hors tension

3- PRESENTATION DU LOGICIEL DWIN

3-1- Installation du logiciel DWIN

3-1-1- Configuration du système

Configuration minimale :

- PC 486 DX2 66
- RAM 8 Mo
- Disque dur (15 Mo disponibles)
- Microsoft® Windows™ 95 ou Microsoft® Windows™NT 4.0 (service pack 3)
- Lecteur de CD-ROM (2X)
- Ecran SVGA
- Souris ou autre périphérique de pointage

Configuration recommandée :

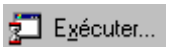
- PC Pentium® 75 ou plus
- RAM 16 Mo
- Disque dur (15 Mo disponibles)
- Microsoft® Windows™ 95 ou Microsoft® Windows™NT 4.0 (service pack 3) ou plus
- Lecteur de CD-ROM (4X)
- Ecran SVGA
- Souris ou autre périphérique de pointage

Cette application ne travaille pas sous Unix, Mac, MS-DOS et Microsoft® Windows 3.11.

3-1-2- Procédure d'installation du logiciel

Le logiciel est fourni sous forme de disquette (à la demande) ou de CD-ROM avec le terminal Dialog 80 ou 640. L'installation du logiciel se fait comme suit :

- Vérifier la configuration requise pour installer le logiciel
- Insérer la disquette ou le CD-ROM dans le lecteur approprié.
- Dans le menu déroulant 

, sélectionner 

.
- Dans la boîte de dialogue « Exécuter », sélectionner 

.
- Dans la boîte de dialogue « Parcourir », sélectionner le lecteur où se situe la disquette ou le CD-ROM.
- Sélectionner 

puis 

dans la boîte de dialogue « Parcourir ».
- Sélectionner 

dans la boîte de dialogue « Exécuter ».

⇨ Le programme d'installation du logiciel DWIN débute.

↳ Le début de l'installation est marquée par une série de boîtes de dialogue guidant l'utilisateur :

- répertoire de destination
- type d'installation (Typique, compacte ou personnalisée)
- sélection du dossier programme

☏ Attention : seul un niveau de répertoire peut être crée.

⇨ **L'installation des fichiers débute et son état d'avancement est indiqué par une barre d'évolution.**

⇨ **L'installation se termine par l'ajout de l'icône de DWIN dans le dossier programme.**

3-2- Architecture

3-2-1- Les répertoires

↳ OS/Dialog80 : contient une copie du système d'exploitation du terminal opérateur Dialog 80.

↳ OS/Dialog640 : contient une copie du système d'exploitation du terminal opérateur Dialog 640.

↳ HELP : contient les fichiers d'aide du logiciel terminal intelligent

Sous l'arborescence de base, on retrouve :

↳ le fichier avec extension EXE pour exécuter le logiciel

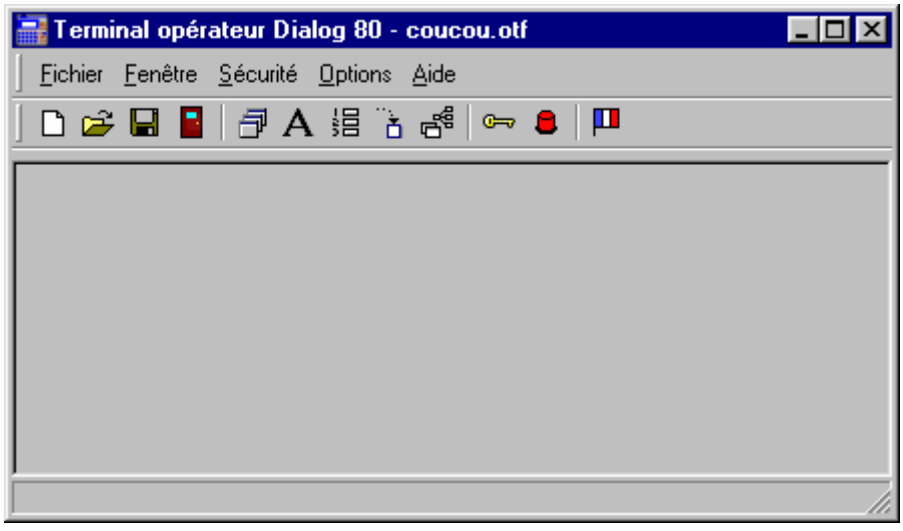
↳ le fichier avec extension STR pour la gestion des langues du logiciel

3-2-2- Contenu d'un projet

Il n'y pas de répertoire prévu pour stocker les projets utilisateurs. C'est donc à l'utilisateur de définir son propre emplacement et d'attribuer un répertoire par projet. Chaque projet est caractérisé par un fichier "Nom Projet.otf". La compilation d'un projet donne des fichiers binaires (Nom Projet.da0 à Nom Projet.da7). La somme des fichiers de type da* permet de connaître la taille du projet compilé.

3-3- Présentation

3-3-1- Ecran initial



La fenêtre principal du logiciel DWIN est composé d'une barre de menu et d'une barre d'icônes ainsi qu'un plan de travail où plusieurs fenêtres peuvent apparaître. La barre de menu est définie comme ceci :

↳ Fichier : regroupe toutes les commandes liées au projet (nouveau, sauver, fermer…) ainsi que les commandes de transfert vers le terminal.

↳ Fenêtre : regroupe les commandes d'affichage des fenêtres liées aux listes de variables, pages, etc..

↳ Sécurité : regroupe les commandes d'affichage des fenêtres liées aux listes des codes d'accès et des alarmes.

↳ Options : regroupe les commandes de paramétrage du logiciel et du terminal

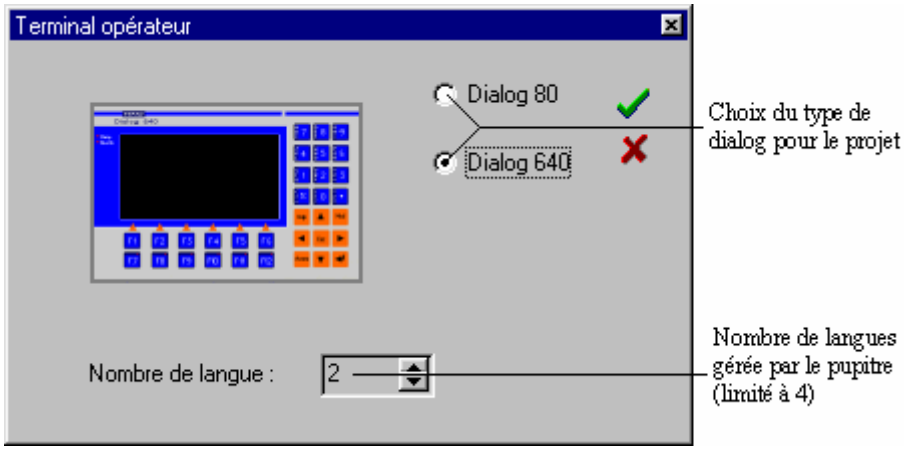
↳ Aide : regroupe les informations d'aide du logiciel

3-3-2- Menu Fichier

Nouveau

Icône :

Action : Cette commande permet à l'utilisateur de définir un nouveau projet. Le dernier projet en cours se trouve alors fermé. Une boîte de dialogue permet de configurer les paramètres nécessaires au projet et en particulier le terminal de destination (dialog 80 et dialog 640) et le nombre de langues. Ce nombre définit combien de langues le terminal sera capable de gérer (1 à 4). Ces paramètres sont important car ils ne peuvent à aucun autre moment être modifiés.



Ouvrir

Icône :

Action : Cette commande ouvre la boîte de dialogue « Ouvrir un projet ». Elle permet à l'utilisateur de spécifier le chemin et le nom du projet à charger (extension OTF). Cette commande a pour effet de fermer le dernier projet en cours dès la validation du projet à ouvrir.

Fermer

Action : Cette commande ferme le projet en cours.

Enregistrer

Icône :

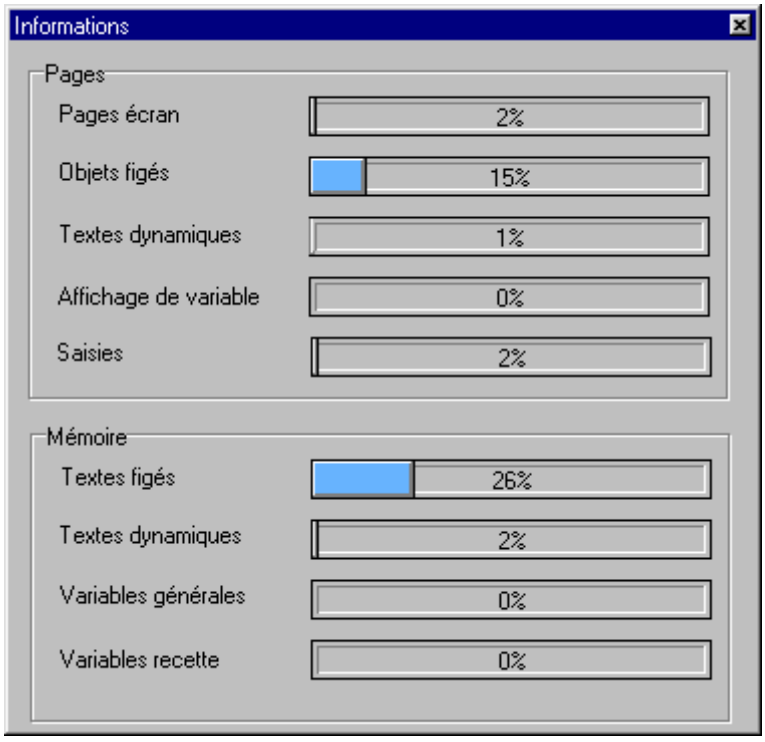
Action : Cette commande enregistre le projet en cours.

Enregistrer sous…

Action : Cette commande permet à l'utilisateur de spécifier le nom de sauvegarde de son projet et l'arborescence de sauvegarde.

Informations

Action : Cette commande ouvre une fenêtre qui indique l'état des ressources occupées par le projet.



Dans le cas ci-dessus, ce sont les informations d'un projet pour un terminal dialog 640. On constate que 2% des pages écran sont utilisées sur un total maximum de 200.

Envoyer le projet

Action : Cette commande permet d'envoyer le code du projet compilé au terminal. Pour réaliser cette opération, il est nécessaire de suivre les explications de la procédure de transfert.

Envoyer les recettes

Action : Cette commande permet d'envoyer au terminal une copie de sauvegarde des recettes, sauvegardée avec la commande recevoir les recettes. Le projet présent dans le terminal et celui du PC doivent être les mêmes. Lors du transfert, il est nécessaire de préciser le port de communication utilisé sur le PC. La procédure pour l'envoi des recettes est décrite dans un chapitre.

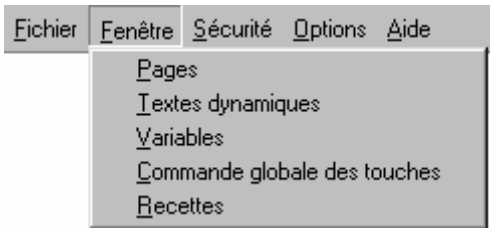
Recevoir les recettes

Action : Cette commande réceptionne les données des recettes provenant du terminal. Pour en faire une copie de sauvegarde, il est nécessaire de sauvegarder le projet. Le projet présent dans le terminal et celui du PC doivent être les mêmes. Lors du transfert, il est nécessaire de préciser le port de communication utilisé sur le PC. La procédure pour la réception des recettes est décrite dans un chapitre.

Quitter

Action : Cette commande permet de quitter le logiciel DWIN.

3-3-3- Menu Fenêtre



Action : Ces commandes valident ou non l'affichage de la fenêtre Pages, Textes dynamiques, Variables, Commande globale des touches ou Recettes.

Ces commandes peuvent être aussi obtenues par les icônes suivantes :

- Pages :
- Textes dynamiques :
- Variables :
- Commande globale des touches :
- Recettes :

3-3-4- Menu Sécurité

Action : Ces commandes valident ou non l'affichage de la fenêtre Codes d'accès ou Alarmes.

Ces commandes peuvent être aussi obtenues par les icônes suivantes :

↳ Code d'accès : 

↳ Alarmes : 

3-3-5- Menu Options

Langues

Icône : 

Action : Cette commande permet d'activer ou non la fenêtre de gestion des langues du terminal.

Protocole

Action : Cette commande permet de faire apparaître la boîte de dialogue liée au protocole de communication entre le terminal et le périphérique. La configuration nécessite le choix du protocole parmi : ModBus, CANopen.

Préférences

Action : Cette commande permet de modifier les paramètres de configuration du logiciel DWIN. En particulier, on trouvera le langage du logiciel DWIN et la durée du rétro-éclairage du dialog 640 après l'appui d'une touche clavier.

3-3-6- Menu Aide

A propos de

Action : Cette commande permet d'afficher les informations de développement et de version du logiciel

Index

Action : Cette commande lance l'aide en ligne associée au logiciel DWIN.

4- TRANSFERT D'UN PROJET

4-1- Description du menu de démarrage du système

Pour obtenir le menu de démarrage du système, il faut appuyer simultanément sur les touches ESC et RETURN à n'importe quel instant du fonctionnement du terminal. Le code d'accès qui est demandé dépend du terminal utilisé. Il est de 80 pour le dialog80 et de 640 pour le dialog640.

Le menu de démarrage comporte un menu PC qui concerne le transfert des programmes et données, un menu INFO qui apporte des renseignements sur la configuration du terminal et une commande QUIT pour retourner au mode de fonctionnement normal.

Le menu PC permet de charger un programme (LOAD PROJECT), de charger les recettes (LOAD DATA) et de sauver les recettes (SAVE DATA). Chacune de ces commandes est détaillée dans une partie spécifique.

Le menu INFO est composée des informations sur le projet (PROJECT), de la version d'OS (OS) et de la configuration de la liaison série (SERIAL). Les informations sur le projet sont : le nom, la date de chargement sur le terminal et la taille du projet. Les informations sur l'OS sont : le numéro de version ainsi que sa date de chargement. Les informations sur la liaison série concernent son type, sa configuration et le protocole utilisé.

4-2- Description du menu de boot du système

Le menu du boot ne peut s'obtenir qu'en appuyant sur la touche ESC au préalable et en alimentant le terminal ensuite. Le code d'accès qui est demandé dépend du terminal utilisé. Il est de 80 pour le dialog80 et de 640 pour le dialog640.

Le menu PC comporte la commande de mise à jour d'une version. La procédure de mise à jour de l'OS est décrite dans un chapitre spécifique (Mise à jour d'une version antérieure). La commande TEST permet le test des touches et des leds du terminal. La commande INFO fournit la version de l'OS et sa date de chargement sur le terminal.

4-3- Mise à jour d'une version antérieure

↳ Pour mettre à jour l'OS du terminal, il faut dans un premier temps relier le terminal avec le PC avec un câble série. Il est important que le PC soit relié sur le port de communication série 1.

↳ Il faut ensuite accéder au menu de boot du terminal. Pour cela, on appuie au préalable sur la touche ESC et on alimente le terminal. Celui-ci demande un code d'accès qui est de 80 pour le dialog80 et de 640 pour le dialog640.

↳ On entre dans le menu PC en appuyant sur la touche 2 du dialog80 ou F1 du dialog640.

↳ Puis, on choisit la commande LOAD en appuyant sur la touche 2 du dialog80 ou F1 du dialog640. Le terminal procède par l'effacement des différents blocs de sa mémoire avant de se mettre en attente d'une réception du nouveau système d'exploitation. Il le signale par le message « Attente programme ».

↳ Sur le PC, on se place sous l'arborescence d'origine de DWIN (C:\Program Files\Serad\Operator terminal). Sous celle-ci, on détecte le répertoire OS dans lequel on trouve le fichier Install.bat. On exécute ce fichier.

↳ Une fenêtre DOS s'ouvre et attend l'appui d'une touche avant d'envoyer la nouvelle version du système d'exploitation.

↳ Lorsque le chargement de l'OS est terminé, il faut appuyer sur une touche du clavier du PC pour quitter le logiciel et ensuite, il faut fermer la fenêtre DOS. Ceci est nécessaire si on veut que le logiciel libère le port de communication.

↳ La version du système d'exploitation est maintenant mise à jour

4-4- Procédure de chargement d'un projet

↳ Pour envoyer un projet sur le terminal, il faut, dans un premier temps, relier le PC avec le terminal au moyen d'un câble série. Le port de communication du PC n'importe pas pour le transfert.

↳ Il faut ensuite accéder au menu de démarrage du terminal en appuyant simultanément sur les touches ESC et RETURN. Le code d'accès demandé est de 80 pour le dialog80 et de 640 pour le dialog640.

↳ On sélectionne le menu PC par la touche 2 du dialog80 ou la touche F1 du dialog640. Puis, on sélectionne la commande LOAD PROJECT par la touche 2 du dialog80 ou la touche F1 du dialog640. On valide la commande par YES par la touche 2 du dialog80 ou la touche F1 du dialog640. Le terminal commence par effacer sa mémoire et se met en attente du projet. Il signale son attente par le message : « Waiting for project ».

↳ A partir de là, on sélectionne le menu « Fichier » et la commande « Envoyer le projet » du logiciel DWIN. On choisit le port de communication auquel est relié le PC avec le terminal et on valide l'envoi.

↳ Le PC affiche l'évolution du chargement du projet par une barre de progression. Dès sa disparition, l'envoi est terminé. Le terminal redémarre et exécute le projet chargé.

4-5- Procédure de sauvegarde des recettes

↳ Pour sauvegarder les données des recettes à partir d'un projet situé sur un terminal, il faut, dans un premier temps, relier le PC avec le terminal au moyen d'un câble série. Le port de communication du PC n'importe pas pour le transfert.

↳ On ouvre le projet correspondant à celui du terminal dans DWIN.

↳ Il faut ensuite accéder au menu de démarrage du terminal en appuyant simultanément sur les touches ESC et RETURN. Le code d'accès demandé est de 80 pour le dialog80 et de 640 pour le dialog640.

↳ On sélectionne le menu PC par la touche 2 du dialog80 ou la touche F1 du dialog640. Puis, on sélectionne la commande SAVE DATA par la touche 5 du dialog80 ou la touche F6 du dialog640. Le terminal se met en attente d'une requête du PC. Il signale son attente par le message : « Waiting for request ».

↳ A partir de là, on sélectionne le menu « Fichier » et la commande « Recevoir les recettes » du logiciel DWIN. On choisit le port de communication par lequel est relié le PC avec le terminal et on valide l'envoi.

↳ Le PC affiche l'évolution de la sauvegarde des recettes par une barre de progression. Dès sa disparition, l'envoi est terminé. Le terminal redémarre et exécute le projet.

↳ Il est nécessaire de sauvegarder le projet sur le PC à partir du menu Fichier.

4-6- Procédure de chargement des recettes

↳ Pour envoyer les recettes sur le terminal, il faut, dans un premier temps, relier le PC avec le terminal au moyen d'un câble série. Le port de communication du PC n'importe pas pour le transfert.

- On ouvre le projet correspondant à celui du terminal dans DWIN. Il faut bien sûr avoir sauvegarder les recettes au mois une fois auparavant sur le PC.

- Il faut ensuite accéder au menu de démarrage du terminal en appuyant simultanément sur les touches ESC et RETURN. Le code d'accès demandé est de 80 pour le dialog80 et de 640 pour le dialog640.

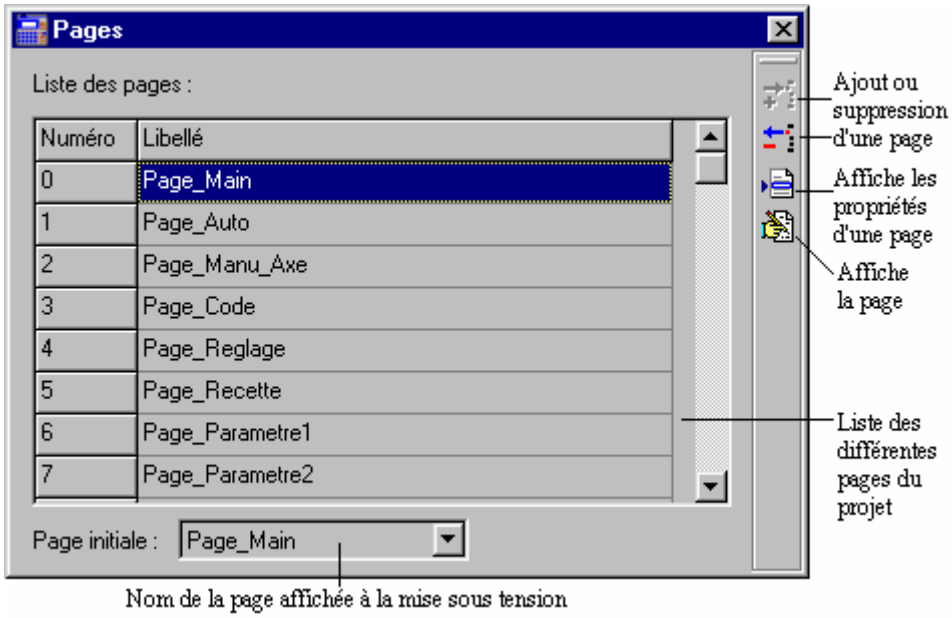
- On sélectionne le menu PC par la touche 2 du dialog80 ou la touche F1 du dialog640. Puis, on sélectionne la commande LOAD DATA par la touche 4 du dialog80 ou la touche F5 du dialog640. On valide la commande par YES par la touche 2 du dialog80 ou la touche F1 du dialog640. Le terminal se met en attente des données provenant du PC. Il signale son attente par le message : « Waiting for data ».

- A partir de là, on sélectionne le menu « Fichier » et la commande « Envoyer les recettes » du logiciel DWIN. On choisit le port de communication auquel est relié le PC avec le terminal et on valide l'envoi.

- Le PC affiche l'évolution du chargement des recettes par une barre de progression. Dès sa disparition, l'envoi est terminé. Le terminal redémarre et exécute le projet chargé en prenant en compte les données des recettes envoyées.

5- LISTE DES PAGES

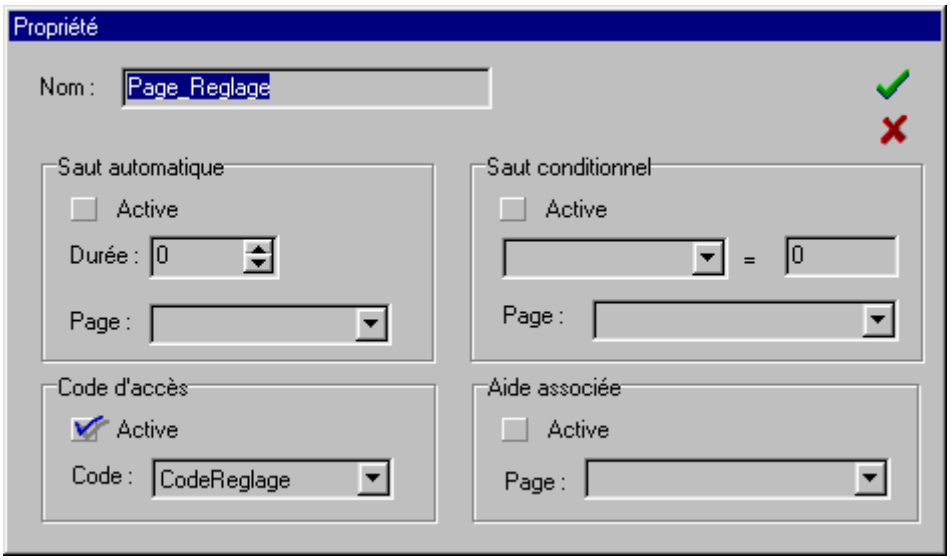
5-1- Présentation générale



Cette fenêtre présente la liste des pages d'un projet. Son activation ou sa désactivation peut être obtenue en agissant sur l'icône . C'est à partir de cette fenêtre que l'on peut venir agir sur les pages écran avec la déclaration, la suppression, la modification des propriétés et l'édition d'une page. Sur cette page, on peut venir déclarer la page initiale qui sera affiché sur le terminal à chaque mise sous tension. Pour que l'on puisse en déclarer une, il faut qu'au moins une page soit déclarée dans la liste. L'édition d'une page s'obtient par la commande . L'éditeur qui sera lancé est différent suivant le type de terminal. Il y a donc un éditeur spécifique pour le Dialog80 et pour le Dialog640. L'activation de l'éditeur peut aussi être obtenue par un double clique gauche sur un emplacement de page déclarée.

5-2- Déclaration d'une nouvelle page

La définition d'une nouvelle page s'obtient soit par un double clique gauche sur un emplacement vide soit par un clique gauche sur l'icône de la fenêtre liste de projet. Sur l'activation de cette commande, la fenêtre des propriétés d'une page apparaît. C'est sur celle-ci que l'on vient configurer les caractéristiques de la nouvelle page. Cette fenêtre des propriétés d'une page peut aussi être obtenue sur une page existante en sélectionnant celle-ci au préalable et par un clique gauche sur l'icône .



Dans la fenêtre des propriétés, on peut venir définir ou modifier le nom de la page. Il n'y a aucune restriction sur le nom de la page.

Les propriétés d'une page concernent principalement son enchaînement avec les autres. Pour que l'une de ces propriétés soit prise en compte, il faut impérativement que la case associée et signalée par Active soit validée.

Saut automatique : Cette propriété caractérise le saut vers une autre page après un délai d'affichage de la page courante. Les paramètres à définir sont :

⇨ la durée qui est exprimée en seconde

⇨ le nom de la page sur laquelle se fait l'enchaînement

```
Exemple  : durée=2s  ; Page= PageSuivante
Un saut vers PageSuivante sera effectué après être resté 2s dans la
PageCourante.
```

Saut conditionnel : Cette propriété caractérise un saut vers une autre page lorsqu'une condition d'égalité se trouve vérifiée. Cet enchaînement ne se produit que si la page courante est affichée sur le terminal et que la condition d'égalité est vérifiée. Le test est une égalité entre une variable définit dans la liste des variables et une valeur fixe. Les propriétés à définir sont :

⇨ le nom de la variable à tester

⇨ la valeur fixe pour le test

⇨ le nom de la page sur laquelle se fait l'enchaînement

```
Exemple  : égalité  : Var=4  ; Page= PageSuivante
Un saut vers PageSuivante sera effectué lorsque Var sera équivalent
à 4 et l'affichage de la PageCourante.
```

Code d'accès : Cette propriété permet d'afficher une page de demande de code d'accès avant l'affichage de la page. L'affichage de la page ne sera effectué que si le code validé par l'utilisateur est correct. Dans le cas contraire, la page de code restera affiché. La page de code d'accès n'est pas à définir car c'est une page prédéfinie et donc standard du logiciel. Les paramètres à définir sont :

⇨ le nom de la variable code d'accès

```
Exemple  : Code=CodePage  ;
Un saut vers PageCourante sera effectué après avoir saisi le bon
code d'accès.
```

Aide associée : Cette propriété permet de définir si une page d'aide est présente sur l'affichage de cette page. Lors de l'affichage de la page sur le terminal, l'association de la page d'aide est signalée par le clignotement de la led Help. L'enchaînement sur celle-ci s'obtient par l'appui sur la touche Help du terminal. Les paramètres à définir sont :

⇨ la page d'aide

```
Exemple  : Page= PageAide
Un saut vers PageAide sera effectué lors de l'appui sur la touche
HELP et l'affichage de la PageCourante. De même, à l'affichage de la
PageCourante, le clignotement de la led Help signale une page
d'aide.
```

5-3- Suppression d'une page

La suppression d'une page s'obtient en sélectionnant au préalable la page à supprimer et par un clique gauche sur l'icône  . La suppression ne provoque aucun décalage des pages situées à un index supérieur. Elle laisse donc un emplacement vide qui pourra être utilisée par une autre page. Si la page supprimée est utilisée par d'autres pages pour des enchaînements, les champs définis sont alors vides. C'est à l'utilisateur de penser à les redéfinir.

5-4- Editeur pour Dialog80

5-4-1- Présentation de l'éditeur

L'éditeur est caractérisé par deux fenêtres.

La première fenêtre correspond à l'éditeur pleine page. Dans celui-ci, on voit la face avant du dialog 80 et une barre d'outil située sur la droite. Sur cet écran on peut configurer :

↳ Les touches 2, 3, 4, 5 du terminal par un clique gauche sur celle-ci

↳ Les touches F1, F2, F3, F4, F5, F6 et leurs leds par un clique gauche sur la touche correspondante

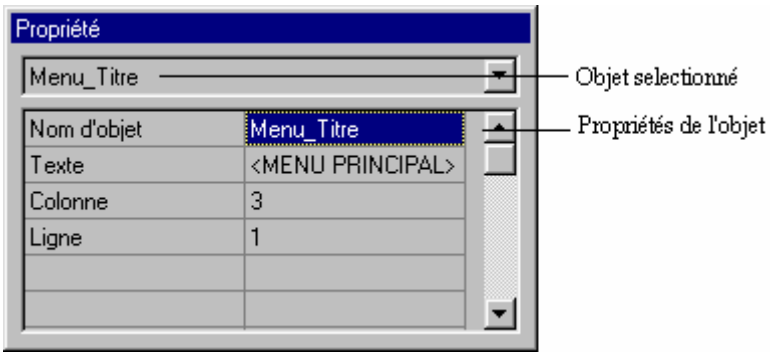
↳ La touche ESC par un clique gauche sur celle-ci

↳ L'écran du terminal

La barre d'outil est constituée des icônes : Validation de l'édition de la page et annulation de l'édition de la page . Elle est aussi constituée des icônes suivantes qui permettent l'ajout d'objets sur l'écran du terminal :

- Texte statique
- Texte dynamique
- Affichage de variable numérique
- Affichage de variable alphanumérique
- Saisie de variable numérique
- Saisie de variable alphanumérique

L'icône permet de pointer un objet de l'écran afin de modifier son dimensionnement ou ses propriétés. Cette icône est directement validée dès l'ajout d'un objet sur l'écran.



La dernière icône permet de valider ou non la deuxième fenêtre qui concerne les propriétés des objets situées sur l'écran du terminal. Dans cette fenêtre, on peut sélectionner l'objet désiré qui est caractérisé par un nom et l'affichage de ses propriétés. La sélection d'un objet peut aussi être obtenue par un clique gauche sur l'objet situé sur l'écran du terminal. La liste des propriétés située sur la colonne de gauche est différente suivant l'objet sélectionné. Les champs de la colonne de droite correspondent aux caractéristiques des propriétés de la colonne de gauche et sont donc modifiables par l'utilisateur. Toute modification de l'un de ces champs entraîne une modification sur l'objet de l'écran. De même toute modification apportée sur un objet de l'écran du terminal entraîne une modification sur ses propriétés.

5-4-2- Texte statique

Ce type d'objet permet d'afficher un texte fixe à l'écran. Ce texte peut être défini par l'utilisateur.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace le texte statique en bougeant la souris (glisser-déplacer).

Les propriétés de cet objet sont :

Nom d'objet : Il caractérise le nom de l'objet et doit être unique.

Texte : Il correspond au texte qui sera affiché. La longueur de l'objet se dimensionne automatiquement en fonction de la longueur du texte. Si le texte inscrit est plus long que la taille d'affichage du terminal, les caractères en trop seront éliminés à l'affichage.

Colonne : Cette caractéristique définit la colonne de l'écran où se situera le premier caractère du texte à afficher.

Ligne : Cette caractéristique définit la ligne de l'écran où le texte sera affiché.

5-4-3- Texte dynamique

Ce type d'objet permet d'afficher un texte dynamique à l'écran. Le texte affiché dépend d'une variable index qui référence un texte de la liste des textes dynamiques.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace le texte dynamique en bougeant la souris (glisser-déplacer).

⇨ Par un clique sur l'un des bords gauche ou droite de l'objet et en le maintenant enfoncé, on peut modifier la longueur de texte autorisé à l'affichage.

Les propriétés de cet objet sont :

Nom d'objet : Il caractérise le nom de l'objet et doit être unique.

Colonne : Cette caractéristique définit la colonne de l'écran où se situera le premier caractère du texte à afficher.

Ligne : Cette caractéristique définit la ligne de l'écran où le texte sera affiché.

Longueur : Cette propriété spécifie la longueur de l'objet et donc le nombre de caractères autorisé à l'affichage. Si le texte à afficher possède un nombre de caractères plus important que la zone d'affichage de l'objet, les caractères en trop n'apparaîtront pas à l'écran.

Numéro : Elle définit l'adresse de départ du texte de la liste dynamique lorsque la variable codage est de type BCD ou binaire. Lorsque l'état de la variable est nulle, il représente directement l'adresse de départ. Lorsque la variable codage est de type bit, cette propriété indique le numéro du bit qui agira comme index. Par exemple pour une valeur de 4 et pour une propriété de codage de type BCD ou binaire, on aura un affichage des textes dynamiques situés à partir de l'adresse 4. Pour une propriété de codage de type Bit, on aura un affichage qui dépendra de la variation du quatrième bit de la variable index (le bit de poids faible est le bit 1).

Bit0 : Cette propriété n'est valide que si la propriété codage est de type Bit. Elle définit la valeur de l'index lorsque le bit est à 0. Si on indique 25, le texte affiché, lorsque le bit est 0, sera celui située à l'adresse 25 de la liste des textes dynamiques.

Bit1 : Cette propriété n'est valide que si la propriété codage est de type Bit. Elle définit la valeur de l'index lorsque le bit est à 1. Si on indique 26, le texte affiché, lorsque le bit est 1, sera celui située à l'adresse 26 de la liste des textes dynamiques.

Nombre : Cette propriété n'est valide que si la propriété codage est de type Binaire ou BCD. Elle définit le nombre maximum de textes que peut afficher l'objet. Il permet de connaître l'index haut qui est alors l'addition des propriétés Numéro+Nombre-1.

Variable : Cette propriété définit la variable index qui indiquera le texte de la liste dynamique qui sera affiché. Le texte affiché correspondra à la valeur Numéro + valeur de la variable, sachant que l'index est compris dans la fourchette Numéro et Numéro+Nombre-1.

Codage : Cette propriété définit le type de codage de la variable index. Suivant le choix défini, les propriétés ne sont pas les mêmes. Le type bit n'autorise l'évolution de l'index sur un seul bit et donc deux types de textes dynamiques sont possibles. Les propriétés associées à cette

propriété sont : Bit0, Bit1 et Numéro. Le type BCD indique que l'évolution de la variable index se fait suivant un format BCD (par exemple : 01010011b représente un index de 93). Le type binaire indique un format binaire de l'index (pour l'exemple précédent : 01010011b , l'index est de 83). Les propriétés associées à BCD et à binaire sont : Nombre et Numéro.

```
Exemple 1:
Codage=binaire ; Numéro=5 ; Nombre=15 ; Variable=VarTextDyn
Si contenu de VarTextDyn=0h, alors affichage du texte dynamique n°5
Si contenu de VarTextDyn=1h, alors affichage du texte dynamique n°6
Etc..
Si contenu de VarTextDyn=0Ah, alors affichage du texte dynamique n°15
Etc..
Si contenu de VarTextDyn=0Eh, alors affichage du texte dynamique n°19
Si contenu de VarTextDyn>0Eh ou <0, alors aucun affichage de texte
Exemple 2:
Codage=BCD ; Numéro=5 ; Nombre=15 ; Variable=VarTextDyn
Il faut que le format de la variable VarTextDyn soit de type BCD.
Si contenu de VarTextDyn=0h, alors affichage du texte dynamique n°5
Si contenu de VarTextDyn=1h, alors affichage du texte dynamique n°6
Etc..
Si contenu de VarTextDyn=10h, alors affichage du texte dynamique n°15
Etc..
Si contenu de VarTextDyn=14h, alors affichage du texte dynamique n°19
Si contenu de VarTextDyn>14h ou <0, alors aucun affichage de texte
Exemple 3:
Codage=Bit ; Numéro=5 ; Bit0=55 ; Bit1=32 ; Variable=VarText
Si contenu de VarText=XXX0XXXXb, alors affichage du texte dynamique n°55
Si contenu de VarText=XXX1XXXXb, alors affichage du texte dynamique n°32
```

5-4-4- Affichage d'une variable numérique

Ce type d'objet permet d'afficher la valeur d'une variable numérique à l'écran.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace la variable numérique en bougeant la souris (glisser-déplacer).

Les propriétés de cet objet sont :

- **Nom d'objet :** Il caractérise le nom de l'objet et doit être unique.
- **Colonne :** Cette caractéristique définit la colonne de l'écran où se situera le premier caractère de la valeur à afficher.
- **Ligne :** Cette caractéristique définit la ligne de l'écran où la valeur sera affichée.
- **Variable :** Cette propriété indique la variable à afficher.
- **Signe :** Cette propriété définit si le signe de la variable doit être affiché.
- **Partie entière :** Cette propriété définit le nombre de chiffre affiché avant la virgule.
- **Partie décimale :** Cette propriété définit le nombre de caractère affiché après la virgule.
- **Alignement :** Cette propriété définit l'alignement de la valeur par rapport à son cadre (droite ou gauche)
- **Base :** Cette propriété définit la base d'affichage de la valeur (Décimale, hexadécimal, binaire)

```
Exemple :
Base=Décimale ; Signe=oui ; Partie entière=3 ; Partie décimale=2 ;
Variable=VarNum
Si contenu de VarNum=+3.25, alors affichage de la valeur +3.25
Si contenu de VarNum=-2255.25, alors affichage de la valeur -2255.2
Si contenu de VarNum=+255.2564, alors affichage de la valeur +255.25
```

5-4-5- Affichage d'une variable alphanumérique

Ce type d'objet permet d'afficher la valeur d'une variable alphanumérique à l'écran.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace la variable alphanumérique en bougeant la souris (glisser-déplacer).

⇨ Par un clique sur l'un des bords gauche ou droite de l'objet et en le maintenant enfoncé, on peut modifier la longueur de texte autorisé à l'affichage.

Les propriétés de cet objet sont :

↳ **Nom d'objet :** Il caractérise le nom de l'objet et doit être unique.

↳ **Colonne :** Cette caractéristique définit la colonne de l'écran où se situera le premier caractère de la valeur à afficher.

↳ **Ligne :** Cette caractéristique définit la ligne de l'écran où la valeur sera affichée.

↳ **Variable :** Cette propriété indique quelle est la variable à afficher.

↳ **Longueur :** Cette propriété indique le nombre maximum de caractères à afficher.

5-4-6- Saisie d'une variable numérique

Ce type d'objet permet de définir un champ de saisie d'une valeur numérique sur l'écran.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace le champ de saisie en bougeant la souris (glisser-déplacer).

Les propriétés de cet objet sont :

↳ **Nom d'objet :** Il caractérise le nom de l'objet et doit être unique.

↳ **Colonne :** Cette caractéristique définit la colonne de l'écran où se situera le premier caractère de la valeur à afficher.

↳ **Ligne :** Cette caractéristique définit la ligne de l'écran où la valeur sera affichée.

↳ **Variable :** Cette propriété indique la variable numérique qui sera affichée et concernée par la saisie.

↳ **Aide :** Cette propriété indique si une page d'aide est associée à la saisie. Si cette propriété est à oui, cela signifie que lorsque le navigateur « > » est positionné sur cet objet, la led « help » clignote et la page d'aide associée peut-être affichée avec la touche « Help ».

↳ **Page d'aide :** Cette propriété indique le nom de la page qui servira d'aide. Pour que celle-ci soit affichée, il faut que la propriété « Aide » soit activée.

↳ **Code d'accès :** Cette propriété indique si un code d'accès doit être spécifié au moment de la modification de la valeur (dès l'appui sur la touche MOD).

↳ **Nom code :** Cette propriété indique quel est le code d'accès de la liste des codes d'accès qui sera demandé.

↳ **Signe :** Cette propriété définit si le signe de la variable doit être affiché.

↳ **Partie entière :** Cette propriété définit le nombre de chiffre affiché avant la virgule. Si le nombre de caractère saisie pour la partie entière est supérieur au nombre N définit dans cette propriété, seules les N premiers caractères seront affichés.

↳ **Partie décimale :** Cette propriété définit le nombre de caractère affiché après la virgule. Si le nombre de caractère saisie pour la partie décimale est supérieur au nombre N définit dans cette propriété, seules les N premiers caractères seront affichés.

↳ **Alignement :** Cette propriété définit l'alignement de la valeur par rapport à son cadre (droite ou gauche)

↳ **Base :** Cette propriété définit la base d'affichage de la valeur (Décimale, hexadécimal, binaire)

↳ **Limite :** Cette propriété définit si des limites doivent être appliquées lors de la saisie.

↳ **Limite mini :** Cette propriété définit la limite basse possible de la valeur saisie.

↳ **Limite maxi :** Cette propriété définit la limite haute possible de la valeur saisie.

↳ **Navigateur :** Cette propriété indique l'ordre d'évolution du navigateur « > ». Cette indicateur « > » permet d'indiquer quel est le champ de saisie qui pourra être modifié sur l'appui de la touche MOD. Le passage de ce navigateur d'un champ de saisie à un autre s'obtient à l'aide des touches flèche vers le haut et flèche vers le bas. Pour ordonnancer l'évolution du navigateur, on indique un numéro d'ordre à la propriété Navigateur. Si pour un champ de saisie cette propriété est équivalente à 1, cela signifie que le champ sera toujours sélectionné sur l'affichage de la page. Un appui sur la flèche de direction bas permet de passer au champ de saisie ayant la propriété Navigateur à 2 (etc…). Pour revenir en arrière, il faut utiliser la touche flèche de direction haut. Tous les numéros attribués à la propriété Navigateur des différents champs de saisie doivent être différents (1,2,3,…). Si plusieurs champs possèdent la même valeur de propriété Navigateur, alors un seul d'entre eux pourra posséder le curseur de sélection « > ». De même, on peut ne pas autoriser le positionnement du curseur sur un champ de saisie en attribuant une valeur 0 à sa propriété Navigateur.

Exemple :
Ordre de création des différents champs de saisie d'une page
```
Champ de saisie n°1  : Navigateur=2
Champ de saisie n°2  : Navigateur=1
Champ de saisie n°3  : Navigateur=3
Champ de saisie n°4  : Navigateur=0
Champ de saisie n°5  : Navigateur=3
Champ de saisie n°6  : Navigateur=8
```
A l'affichage de la page, c'est le Champ de saisie n°2 qui possède le curseur de sélection. L'appui sur la touche flèche de direction bas donne le curseur de sélection au Champ de saisie n°1. Un nouvel appui sur la même touche donne le curseur de sélection au champ de saisie n°5. En effet, les champs de saisie n°3 et n°5 ont la même propriété, mais seul le dernier créé pourra posséder le curseur de sélection. Un nouvel appui sur la touche flèche de direction bas donne le curseur de sélection au Champ de saisie n°6. L'appui sur la touche flèche de direction haut réalise le passage du curseur de sélection dans le sens inverse. Le champ de saisie n°4 ne pourra jamais posséder le curseur de sélection.

↳ **Confirmation :** Cette propriété si elle est validée permet de définir si un écran de confirmation doit être affiché après la saisie du champ (appui sur ENTER). L'écran de confirmation n'est pas à définir par l'utilisateur. Pour confirmer, l'utilisateur doit appuyer sur la touche 2. Et pour annuler la confirmation, il doit appuyer sur la touche 3.

5-4-7- Saisie d'une variable alphanumérique

Ce type d'objet permet de définir un champ de saisie d'une valeur alphanumérique sur l'écran.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace le champ de saisie en bougeant la souris (glisser-déplacer).

Les propriétés de cet objet sont :

Nom d'objet : Il caractérise le nom de l'objet et doit être unique.

Colonne : Cette caractéristique définit la colonne de l'écran où se situera le premier caractère de la valeur à afficher.

Ligne : Cette caractéristique définit la ligne de l'écran où la valeur sera affichée.

Variable : Cette propriété indique qu'elle est la variable alphanumérique qui sera affichée et concernée par la saisie.

Aide : Cette propriété indique si une page d'aide est associée avec la saisie. Si cette propriété est à oui, cela signifie que lorsque le navigateur « > » est positionné sur cet objet, la led « help » clignote et la page d'aide associée peut-être atteinte avec la touche « Help ».

Page d'aide : Cette propriété indique le nom de la page qui servira d'aide. Pour que celle-ci soit affiché, il faut que la propriété « Aide » soit activée.

Code d'accès : Cette propriété indique si un code d'accès doit être spécifié au moment de la modification de la valeur (dès l'appui sur la touche MOD).

Nom code : Cette propriété indique quel est le code d'accès de la liste des codes d'accès qui sera demandé.

Longueur : Cette propriété définit le nombre maximum de caractère alphanumérique qui peuvent être saisie.

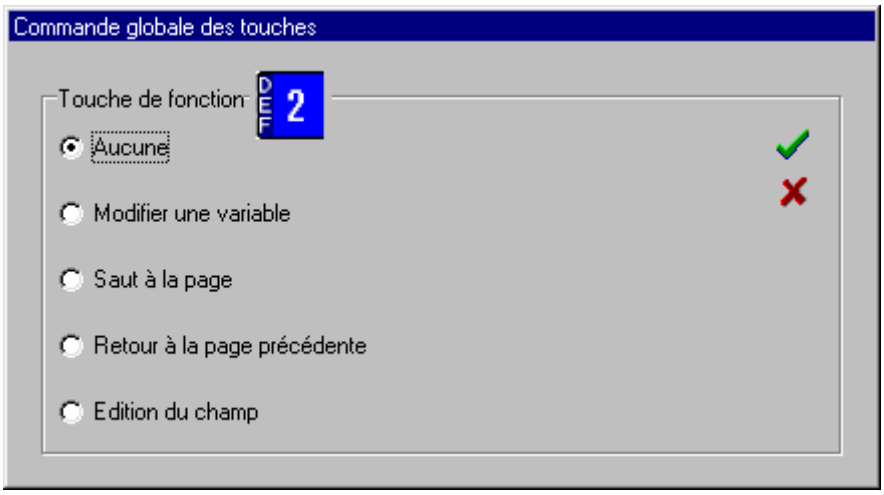
Navigateur : Cette propriété indique l'ordre d'évolution du navigateur « > ». Cette indicateur « > » permet d'indiquer quel est le champ de saisie qui pourra être modifié sur l'appui de la touche MOD. Le passage de ce navigateur d'un champ de saisie à un autre s'obtient à l'aide des touches flèche vers le haut et flèche vers le bas. Pour ordonnancer l'évolution du navigateur, on indique un numéro d'ordre à la propriété Navigateur. Si pour un champ de saisie cette propriété est équivalente à 1, cela signifie que le champ sera toujours sélectionné sur l'affichage de la page. Un appui sur la flèche de direction bas permet de passer au champ de saisie ayant la propriété Navigateur à 2 (etc…). Pour revenir en arrière, il faut utiliser la touche flèche de direction haut. Tous les numéros attribués à la propriété Navigateur des différents champs de saisie doivent être différents (1,2,3,…). Si plusieurs champs possèdent la même valeur de propriété Navigateur, alors un seul d'entre eux pourra posséder le curseur de sélection « > ». De même, on peut ne pas autoriser le positionnement du curseur sur un champ de saisie en attribuant une valeur 0 à sa propriété Navigateur.

Exemple :

```
Ordre de création des différents champs de saisie d'une page
Champ de saisie n°1  : Navigateur=2
Champ de saisie n°2  : Navigateur=1
Champ de saisie n°3  : Navigateur=3
Champ de saisie n°4  : Navigateur=0
Champ de saisie n°5  : Navigateur=3
Champ de saisie n°6  : Navigateur=8
A l'affichage de la page, c'est le Champ de saisie n°2 qui possède le curseur
de sélection. L'appui sur la touche flèche de direction bas donne le curseur
de sélection au Champ de saisie n°1. Un nouvel appui sur la même touche donne
le curseur de sélection au champ de saisie n°5. En effet, les champs de saisie
n°3 et n°5 ont la même propriété, mais seul le dernier créé pourra posséder le
curseur de sélection. Un nouvel appui sur la touche flèche de direction bas
donne le curseur de sélection au Champ de saisie n°6. L'appui sur la touche
flèche de direction haut réalise le passage du curseur de sélection dans le
sens inverse. Le champ de saisie n°4 ne pourra jamais posséder le curseur de
sélection.
```

Confirmation : Cette propriété si elle est validée permet de définir si un écran de confirmation doit être affiché après la saisie du champ (appui sur ENTER). L'écran de confirmation n'est pas à définir par l'utilisateur. Pour confirmer, l'utilisateur doit appuyer sur la touche 2. Et pour annuler la confirmation, il doit appuyer sur la touche 3.

5-4-8- Programmation des touches dynamiques simples



La fonction définie pour une touche ne peut être exécutée que sur l'affichage de la page courante. Leur programmation est prioritaire par rapport à la programmation qui peut être faite de manière globale sur les touches. Les fonctions associées à une touche sont au nombre de 4 :

Modifier une variable : Cette fonction permet sur l'appui de la touche de modifier un bit d'une variable. Pour cette fonction, il faut préciser la variable et le numéro du bit à modifier (le bit n°1 correspond au bit de poids faible de la variable). La modification du bit peut être de plusieurs types (paramètre à spécifier) :

⇨ directe : le bit est à 1 lorsque la touche est enfoncée et à 0 lorsque celle-ci est relâchée.

⇨ flip-flop : un appui sur la touche provoque le basculement du bit de l'état 1 à 0 ou de l'état 0 à 1.

⇨ bit set : L'appui sur la touche provoque le passage du bit à 1.

⇨ bit reset : L'appui sur la touche provoque le passage du bit à 0.

Saut à la page : L'appui sur cette touche permet d'afficher une autre page. Cette page est à spécifier et à définir par l'utilisateur. Sur l'appui de la touche une page de confirmation peut être intercalée. Celle-ci est prédéfinie et n'a pas besoin d'être définie par l'utilisateur.

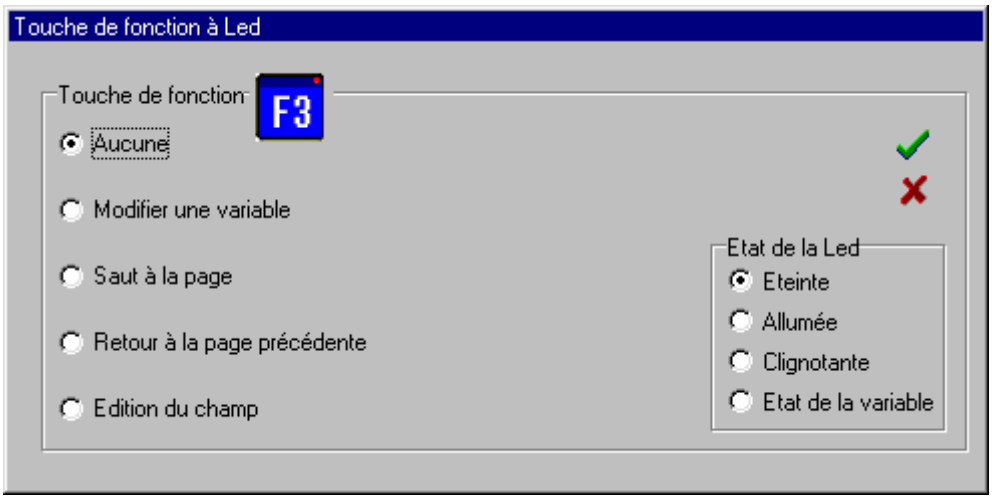
Retour à la page précédente : L'appui sur la touche permet de revenir à la page affichée précédemment.

Edition du champ : L'appui sur la touche permet de lancer la saisie d'un champ qui doit être spécifié par l'utilisateur.

```
Exemple  1:
Nom de la Page  : PageCourante
Touche 2  :    Fonction  : Modifier une variable  ;
          Type=directe  ; Variable=Var  ; Bit n°6
Touche 3  :    Fonction  : Modifier une variable  ;
          Type=flip-flop  ; Variable=Var  ; Bit n°3
Touche 4  :    Fonction  : Modifier une variable  ;
          Type=bit set  ; Variable=Var  ; Bit n°5
Touche 5  :    Fonction  : Modifier une variable  ;
          Type=bit reset  ; Variable=Var  ; Bit n°5
Lors de l'affichage de la PageCourante toutes les opérations suivantes donnent
les résultats suivant  :
⇨ Touche 2 enfoncée  : Var=XX1XXXXXb
⇨ Touche 2 relâchée  : Var=XX0XXXXXb
⇨ Si Var=XXXXX0XXb et un appui sur touche 3  : Var=XXXXX1XXb
⇨ Si Var=XXXXX1XXb et un appui sur touche 3  : Var=XXXXX0XXb
⇨ Un appui sur la touche 4  : Var=XXX1XXXXb
```

```
⇨ Un appui sur la touche 5  : Var=XXX0XXXXb
Exemple  2:
Touche déclarée en globale  :
Touche 4  :    Fonction  : Retour à la page précédente
Touche 5  :    Fonction  : Retour à la page précédente
Nom de la Page  : PageCourante
Touche 2  :    Fonction  : Saut à la page  ;
          Nom de la page  : PageSuivante
Touche 3  :    Fonction  : Retour à la page précédente  ;
Touche 4  :    Fonction  : Edition du champ  ;
          Nom du champ=ChampPageCourante  ;
Lors de l'affichage de la PageCourante toutes les opérations suivantes donnent
les résultats suivants  :
⇨ Un appui sur la touche 2 provoque l'affichage de  PageSuivante. Les
fonctions déclarées pour les touches dynamiques de PageCourante ne sont plus
valables (Uniquement celle déclarée dans PageSuivante).
⇨ Un appui sur touche 3  provoque le retour à la dernière page affichée.
⇨ Un appui sur la touche 4  provoque l'édition du champ de saisie
ChampPageCourante. Cette fonction est prioritaire par rapport à la fonction
définie au niveau globale.
⇨ Un appui sur la touche 5  provoque le retour à la dernière page affichée.
```

5-4-9- Programmation des touches de fonctions à leds



La fonction définie pour une touche et une led ne peut être exécutée que sur l'affichage de la page courante. Leur programmation est prioritaire par rapport à la programmation qui peut être faite de manière globale sur les touches. Les fonctions associées à une touche sont au nombre de 4 :

Modifier une variable : Cette fonction permet sur l'appui de la touche de modifier un bit d'une variable. Pour cette fonction, il faut préciser la variable et le numéro du bit à modifier (le bit n°1 correspond au bit de poids faible de la variable). La modification du bit peut être de plusieurs types (paramètre à spécifier) :

⇨ directe : le bit est à 1 lorsque la touche est enfoncée et à 0 lorsque celle-ci est relâchée.

⇨ flip-flop : un appui sur la touche provoque le basculement du bit de l'état 1 à 0 ou de l'état 0 à 1.

⇨ bit set : L'appui sur la touche provoque le passage du bit à 1.

⇨ bit reset : L'appui sur la touche provoque le passage du bit à 0.

Saut à la page : L'appui sur cette touche permet d'afficher une autre page. Cette page est à spécifier et à définir par l'utilisateur. Sur l'appui de la touche une page de confirmation peut être intercalée. Celle-ci est prédéfinie et n'a pas besoin d'être définie par l'utilisateur.

Retour à la page précédente : L'appui sur la touche permet de revenir à la page affichée précédemment.

Edition du champ : L'appui sur la touche permet de lancer la saisie d'un champ qui doit être spécifié par l'utilisateur.

L'état de la led associée à une touche peut être pilotée de 4 manières différentes :

Eteinte : la led ne sera jamais allumée

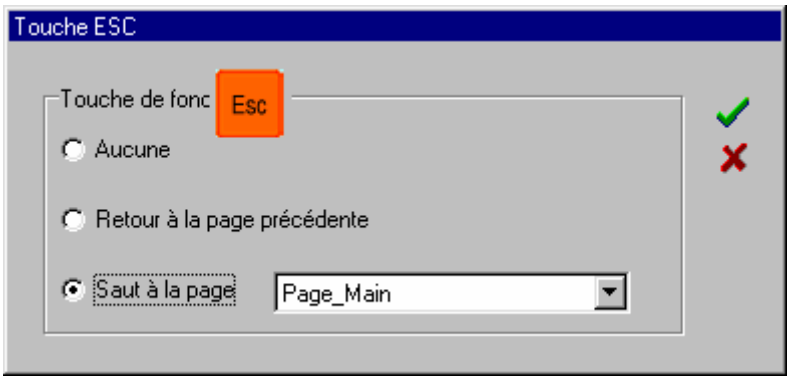
Allumée : la led sera toujours allumée

Clignotante : la led clignotera

Etat de la variable : la led sera l'image du bit de la variable spécifiée dans les paramètres de la fonction de la touche Modifier une variable.

```
Exemple 1:
Nom de la Page : PageCourante
Touche F1 :  Fonction Touche : Modifier une variable ;
             Type=directe ; Variable=Var ; Bit n°6
             Fonction Led : Etat de la variable
Touche F2 :  Fonction Touche : Modifier une variable ;
             Type=flip-flop ; Variable=Var ; Bit n°3
             Fonction Led : Etat de la variable
Touche F3 :  Fonction Touche : Modifier une variable ;
             Type=bit set ; Variable=Var ; Bit n°5
             Fonction Led : Etat de la variable
Touche F4 :  Fonction Touche : Modifier une variable ;
             Type=bit reset ; Variable=Var ; Bit n°5
             Fonction Led : Etat de la variable
Lors de l'affichage de la PageCourante toutes les opérations suivantes donnent
les résultats suivants :
⇨ Touche F1 enfoncée : Var=XX1XXXXXb et led allumée
⇨ Touche F1 relâchée : Var=XX0XXXXXb et led éteinte
⇨ Si Var=XXXXX0XXb et un appui sur touche F2 : Var=XXXXX1XXb et led allumée
⇨ Si Var=XXXXX1XXb et un appui sur touche F2 : Var=XXXXX0XXb et led éteinte
⇨ Un appui sur la touche F3 : Var=XXX1XXXXb et led allumée
⇨ Un appui sur la touche F4 : Var=XXX0XXXXb et led éteinte
Exemple 2:
Touche déclarée en globale :
Touche F3 :  Fonction Touche : Retour à la page précédente
             Fonction led : éteinte
Touche F4 :  Fonction : Retour à la page précédente
             Fonction led : clignotante
Nom de la Page : PageCourante
Touche F1 :  Fonction : Saut à la page ;
             Nom de la page : PageSuivante ;
             Fonction led : éteinte ;
Touche F2 :  Fonction : Retour à la page précédente ;
             Fonction led : allumée ;
Touche F3 :  Fonction : Edition du champ ;
             Nom du champ=ChampPageCourante ;
             Fonction led : clignotante ;
Lors de l'affichage de la PageCourante, les leds sont dans l'état suivant :
⇨ led F1 : éteinte
⇨ led F2 : allumée
⇨ led F3 : clignotante
⇨ led F4 : clignotante
Toutes les opérations suivantes donnent les résultats suivants :
⇨ Un appui sur la touche F1 provoque l'affichage de PageSuivante. Les
fonctions déclarées pour les touches dynamiques et les leds de PageCourante ne
sont plus valables (Uniquement celle déclarée dans PageSuivante).
⇨ Un appui sur touche F2 provoque le retour à la dernière page affichée.
⇨ Un appui sur la touche F3 provoque l'édition du champ de saisie
ChampPageCourante. Cette fonction est prioritaire par rapport à la fonction
définie au niveau global.
⇨ Un appui sur la touche F4 provoque le retour à la dernière page affichée.
```

5-4-10- Programmation de la touche ESC



La fonction définie pour cette touche ne peut être exécutée que sur l'affichage de la page courante. Sa programmation est prioritaire par rapport à la programmation qui peut être faite de manière globale sur cette touche. Les fonctions associées à cette touche sont au nombre de 2 :

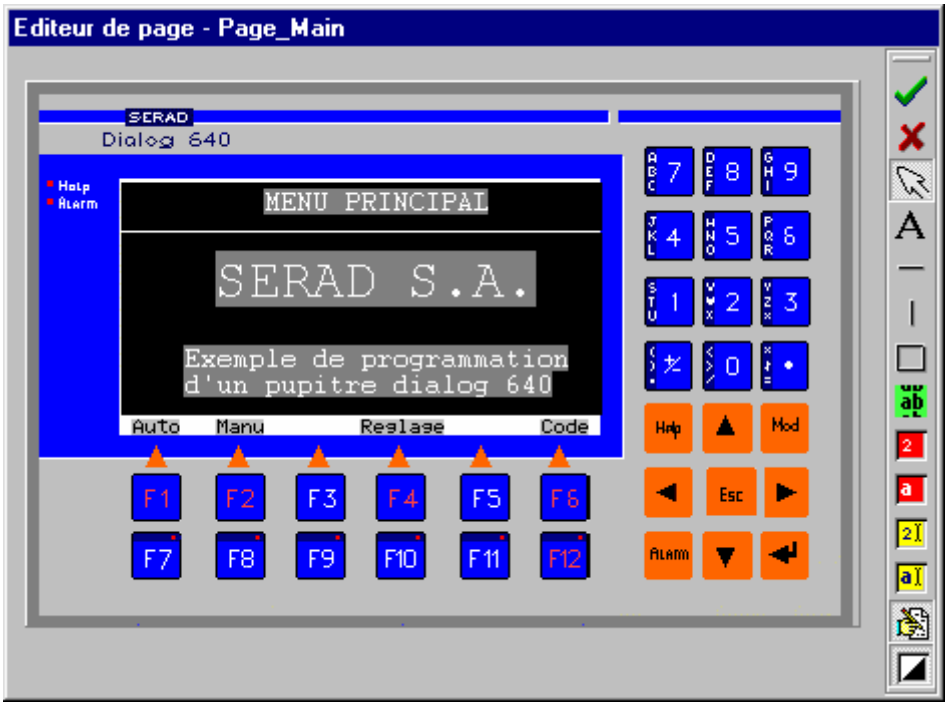
↳ **Retour à la page précédente :** L'appui sur la touche permet de revenir à la page affiché précédemment.

↳ **Saut à la page :** L'appui sur cette touche permet d'afficher une autre page. Cette page est à spécifier et à définir par l'utilisateur.

5-5- Editeur pour Dialog640

5-5-1- Présentation de l'éditeur

L'éditeur est caractérisé par deux fenêtres.



La première fenêtre correspond à l'éditeur pleine page. Dans celui-ci, on voit la face avant du dialog 640 et une barre d'outil située sur la droite. Sur cet écran on peut configurer :

↳ Les touches F1, F2, F3, F4, F5, F6 du terminal par un clique gauche sur celle-ci

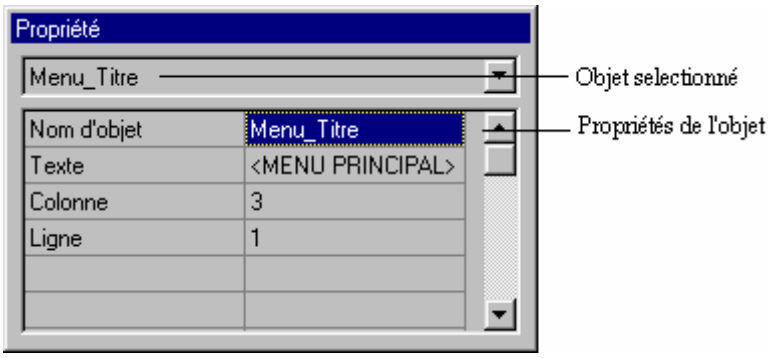
↳ Les touches F7, F8, F9, F10, F11, F12 et leurs leds par un clique gauche sur la touche correspondante

- La touche ESC par un clique gauche sur celle-ci
- L'écran du terminal

La barre d'outil est constituée des icônes : Validation de l'édition de la page et annulation de l'édition de la page . Elle est aussi constituée des icônes suivantes qui permettent l'ajout d'objets sur l'écran du terminal :

- Texte statique
- Texte dynamique
- Affichage de variable numérique
- Affichage de variable alphanumérique
- Saisie de variable numérique
- Saisie de variable alphanumérique
- Affichage d'une ligne verticale
- Affichage d'une ligne horizontale
- Affichage d'un rectangle
- Change la couleur du fond d'écran

L'icône permet de pointer un objet de l'écran afin de modifier son dimensionnement ou ses propriétés. Cette icône est directement validée dès l'ajout d'un objet.



La dernière icône permet de valider ou non la deuxième fenêtre qui concerne les propriétés des objets situées sur l'écran du terminal. Dans cette fenêtre, on peut sélectionner l'objet désiré qui est caractérisé par un nom et l'affichage de ses propriétés. La sélection d'un objet peut aussi être obtenue par un clique gauche sur l'objet situé sur l'écran du terminal. La liste des propriétés située sur la colonne de gauche est différente suivant l'objet sélectionné. Les champs de la colonne de droite correspondent aux caractéristiques des propriétés de la colonne de gauche et sont donc modifiables par l'utilisateur. Toute modification de l'un de ces champs entraîne une modification sur l'objet de l'écran. De même toute modification apportée sur un objet de l'écran du terminal entraîne une modification sur ses propriétés.

5-5-2- Police de caractères

Pour la série des dialog640, la taille des caractères peut-être différente. Lors de l'édition d'une page, les outils texte statique, texte dynamique, affichage d'une variable numérique, affichage d'une variable alphanumérique, saisie d'une variable numérique et saisie d'une variable

alphanumérique ont les mêmes propriétés que ceux de l'éditeur dialog80 à une différence près : il possède en plus une propriété concernant la police de caractère utilisée. Cette propriété correspond à une valeur variant de 1 à 8. Avec pour les polices de 1 à 4, une couleur d'écriture en noir sur fond blanc et pour les polices de 5 à 8, une couleur d'écriture en blanc sur fond noir. Ensuite chaque numéro correspond à une taille différentes d'affichage.

- valeur 1 et 5 : taille de 3×4 mm soit 16 lignes de 40 caractères
- valeur 2 et 6 : taille de 4×7 mm soit 9 lignes de 30 caractères
- valeur 3 et 7 : taille de 5×8 mm soit 8 lignes de 26 caractères
- valeur 4 et 8 : taille de 7×10 mm soit 6 lignes de 17 caractères

5-5-3- Affichage d'une ligne verticale

Ce type d'objet permet d'afficher une ligne verticale à l'écran.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace la ligne verticale en bougeant la souris (glisser-déplacer).

⇨ Par un clique sur l'un des bords haut ou bas de l'objet et en le maintenant enfoncé, on peut modifier la hauteur de la ligne.

Les propriétés de cet objet sont :

- **Nom d'objet :** Il caractérise le nom de l'objet et doit être unique.
- **X1 :** Cette caractéristique définit la position en X de la ligne verticale et est exprimée en pixel (de 1 à 240).
- **Y1 :** Cette caractéristique définit la position limite haute en Y de la ligne verticale et est exprimée en pixel (de 1 à 128).
- **Y2 :** Cette caractéristique définit la position limite basse en Y de la ligne verticale et est exprimée en pixel (de 1 à 128). La position Y2 doit être supérieure à la position Y1.
- **Couleur :** Cette propriété définit la couleur du trait (1 : noir ; 0 : blanc).

5-5-4- Affichage d'une ligne horizontale

Ce type d'objet permet d'afficher une ligne horizontale à l'écran.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace la ligne horizontale en bougeant la souris (glisser-déplacer).

⇨ Par un clique sur l'un des bords gauche ou droite de l'objet et en le maintenant enfoncé, on peut modifier la longueur de la ligne.

Les propriétés de cet objet sont :

- **Nom d'objet :** Il caractérise le nom de l'objet et doit être unique.
- **X1 :** Cette caractéristique définit la position limite gauche en X de la ligne verticale et est exprimée en pixel (de 1 à 240).
- **X2 :** Cette caractéristique définit la position limite droite en X de la ligne verticale et est exprimée en pixel (de 1 à 240). La position X2 doit être supérieure à la position X1.

Y1 : Cette caractéristique définit la position horizontale de la ligne et est exprimée en pixel (de 1 à 128).

Couleur : Cette propriété définit la couleur du trait (1 : noir ; 0 : blanc).

5-5-5- Affichage d'un rectangle

Ce type d'objet permet d'afficher un rectangle sur l'écran du terminal.

⇨ L'ajout d'un objet s'obtient en sélectionnant l'icône  et par un clique gauche sur l'écran du terminal.

⇨ En cliquant sur l'objet et en maintenant enfoncé le bouton gauche, on déplace le rectangle en bougeant la souris (glisser-déplacer).

⇨ Par un clique sur l'un des bords gauche ou droite de l'objet et en le maintenant enfoncé, on peut modifier la longueur du rectangle.

⇨ Par un clique sur l'un des bords haut ou bas de l'objet et en le maintenant enfoncé, on peut modifier la hauteur du rectangle.

⇨ Par un clique sur l'un des coins du rectangle et en le maintenant enfoncé, on peut modifier en même temps la longueur et la hauteur du rectangle.

Les propriétés de cet objet sont :

Nom d'objet : Il caractérise le nom de l'objet et doit être unique.

X1 : Cette caractéristique définit la position en X du coin gauche du rectangle. Cette position est exprimée en pixel (de 1 à 240).

X2 : Cette caractéristique définit la position en X du coin droit du rectangle. Cette position est exprimée en pixel (de 1 à 240).

Y1 : Cette caractéristique définit la position en Y du coin haut du rectangle. Cette position est exprimée en pixel (de 1 à 128).

Y2 : Cette caractéristique définit la position en Y du coin bas du rectangle. Cette position est exprimée en pixel (de 1 à 128).

Cadre : Cette propriété définit la couleur du trait d'encadrement du rectangle (1 : noir ; 0 : blanc).

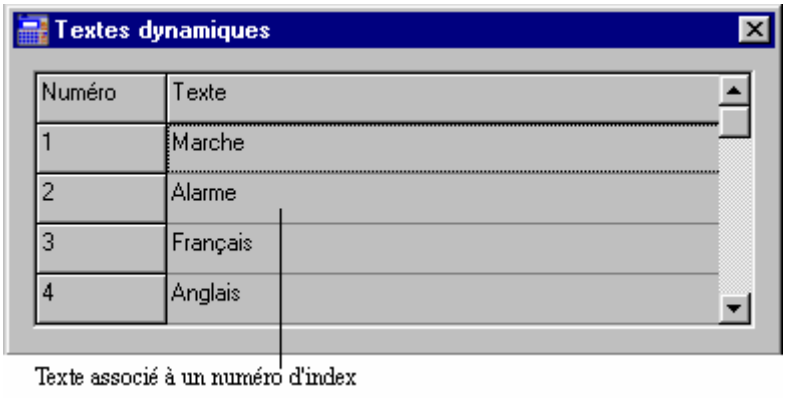
Remplissage : Cette propriété définit la couleur du remplissage du rectangle (1 : noir ; 0 : blanc).

5-5-6- Modification du fond d'écran

L'appui sur cette icône  provoque le changement de la couleur du fond d'écran. Lorsqu'il est activé le fond est blanc, désactivé le fond est noir. Il est recommandé de réaliser des pages écran possédant une couleur noire prédominante par rapport à la couleur blanche, ceci afin d'augmenter la durée de vie de l'afficheur.

6- TEXTES DYNAMIQUES

6-1- Présentation générale



Dans cette fenêtre, on définit les textes qui pourront être affichés par un objet de type Texte dynamique. Ils sont limités à 1500 pour un dialog640 et à 3000 pour un dialog80.

Par exemple, prenons le cas d'un objet de type texte dynamique qui doit afficher l'état de la machine. La propriété codage est de type binaire. Il y a 3 paramètres à spécifier :

- ⇨ le nombre (Nombre) de textes possibles à afficher
- ⇨ la valeur de l'offset de départ dans la liste (Numéro)
- ⇨ la variable d'indexage du texte (Variable)

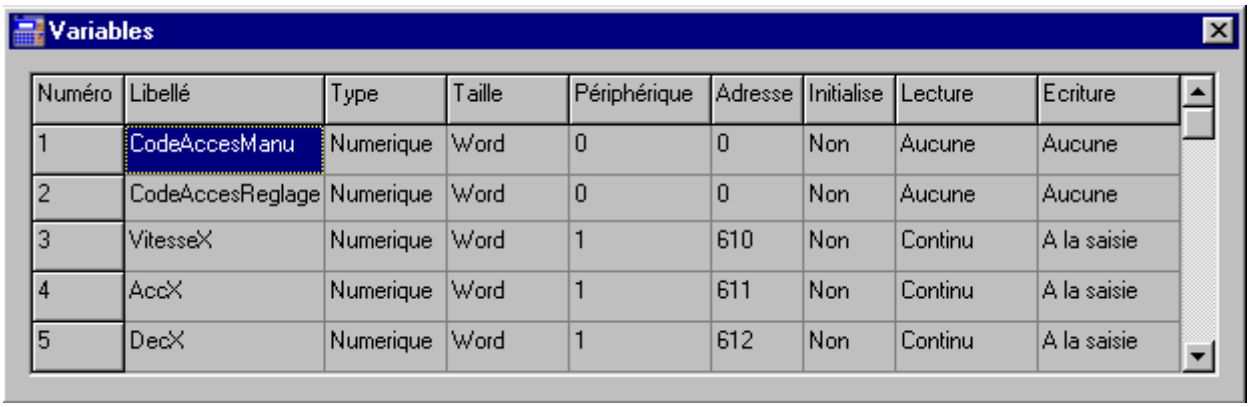
Les messages à afficher sont au nombre de deux (Marche et Alarme). La propriété Nombre doit être mise à 2. Si maintenant, ces messages sont placés aux numéros 1 et 2 de la liste, cela signifie que l'adresse de départ (Numéro) devra être de 1. Il ne reste plus qu'à créer une variable de type numérique et de taille Word (suffisant) dans la liste des variables. Cette variable devra représenter l'image de l'état de la machine. C'est à dire 0 pour « Marche » et 1 pour « Alarme ».

Si ces messages étaient placés en 45 et 46, la propriété Nombre serait toujours égale à 2. Par contre, l'adresse de départ (Numéro) devra être de 45. La valeur de la variable index devra être égale à 0 ou 1. Attention, la solution consistant à positionner l'adresse de départ à 1 et à faire évoluer la variable entre 44 et 45 ne fonctionne pas car Nombre autorise l'affichage que si variable vaut 0 ou 1.

Si le nombre de messages étaient de 45 (d'où Nombre=45) par exemple, l'affichage serait alors autorisée pour une évolution de la variable entre 0 et 44.

7- LISTE DES VARIABLES

7-1- Présentation générale



Numéro	Libellé	Type	Taille	Périphérique	Adresse	Initialise	Lecture	Ecriture
1	CodeAccesManu	Numerique	Word	0	0	Non	Aucune	Aucune
2	CodeAccesReglage	Numerique	Word	0	0	Non	Aucune	Aucune
3	VitesseX	Numerique	Word	1	610	Non	Continu	A la saisie
4	AccX	Numerique	Word	1	611	Non	Continu	A la saisie
5	DecX	Numerique	Word	1	612	Non	Continu	A la saisie

Cette fenêtre présente la liste des variables utilisées par un projet. Son activation ou sa désactivation peut-être obtenue en agissant sur l'icône . C'est à partir de cette fenêtre que l'on peut déclarer, supprimer ou modifier une variable et ses propriétés. Le nombre de déclaration de variables est limité à 5000 pour les dialog80 et dialog640.

7-2- Déclaration et modification d'une variable

Pour déclarer une nouvelle variable, l'utilisateur doit éditer un libellé vide (double clique gauche sur la case), y entrer un nom caractérisant sa variable et le valider par la touche entrée. Lorsqu'une variable est déclarée, toutes ces propriétés sont initialisées par défaut.

Taille : définit le type utilisé par la variable. Ce type peut-être un Word (16 bits signé) ou un DWord (32 bits signé).

Type : est la représentation de la variable sous une forme numérique (valeur) ou alphanumérique (suite de caractères). Un caractère a une taille de 1 octet. Ce qui implique qu'une variable alphanumérique de taille Word ne pourra pas stocker plus de 2 caractères et 4 pour une taille de type DWord. Dans le cas où l'on voudrait qu'une variable ait une taille plus importante, il faut laisser autant de champs libres que la taille souhaitée. Ces champs libres doivent se situer juste après la déclaration de la variable et tous ces champs sont forcément de la même taille que la première (même s'il n'ont pas la même définition). De même la taille d'une variable alphanumérique est définie en fonction du nombre de caractères maximum qui pourront être affichés ou saisis par les objets définit dans les pages. Par exemple, si on veut afficher (ou saisir) un maximum de 10 caractères, la variable alphanumérique associée devra comporter une réservation minimum de 10 octets. Ce qui implique qu'il faut soit 5 emplacements pour une variable déclarée en Word et 3 emplacements pour une variable déclarée en DWord. Si la variable est définit en Word à l'emplacement 26, les emplacements 27, 28, 29, 30 devront être laissés libres. Si au contraire, elle est définit en DWord à la même adresse, les emplacements 27, 28 seulement seront laissés libres. Nous conseillons malgré cela de définir les champs libre avec un nom portant le nom de la variable principale plus un indice (0….1). De même, il est préférable de définir les variables alphanumériques à un emplacement particulier parmi les 5000 emplacements possibles pour les séparer des variables numériques. Ainsi, il sera plus facile de laisser des emplacements libres en cas de modification de la taille d'une variable.

Périphérique : Cette propriété indique le numéro de périphérique lié avec le terminal.

↳ **Adresse :** Cette propriété indique la valeur de l'adresse mémoire du périphérique où la variable peut-être lue et/ou écrite. Attention cette adresse doit être accessible sur le périphérique.

↳ **Initialise :** Cette propriété permet lors de la mise sous tension du terminal d'envoyer la variable vers le périphérique concerné. Cette propriété permet au terminal de servir de mémoire sauvegardée pour le périphérique.

↳ **Lecture :** Le terminal peut lire de plusieurs façons une variable d'un périphérique :

⇨ de manière continue : la variable est lue tout durant le fonctionnement du terminal. Dans le cas d'un texte dynamique, c'est la variable d'indexage qui est lue.

⇨ de manière unique : la variable est lue à chaque fois qu'une page va être affichée et qu'elle nécessite l'affichage de la variable. Dans le cas d'un texte dynamique, c'est la variable d'indexage qui est lue.

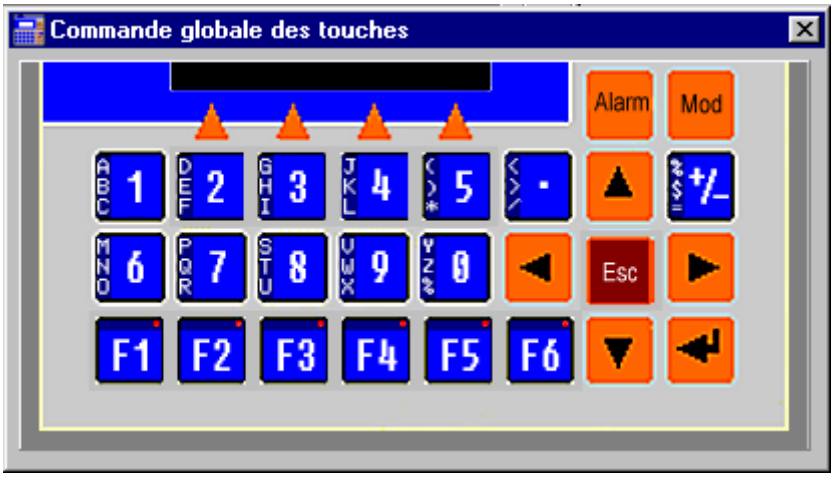
↳ **Ecriture :** Le terminal envoie l'état de la variable au périphérique lorsqu'une saisie de la variable a été validée.

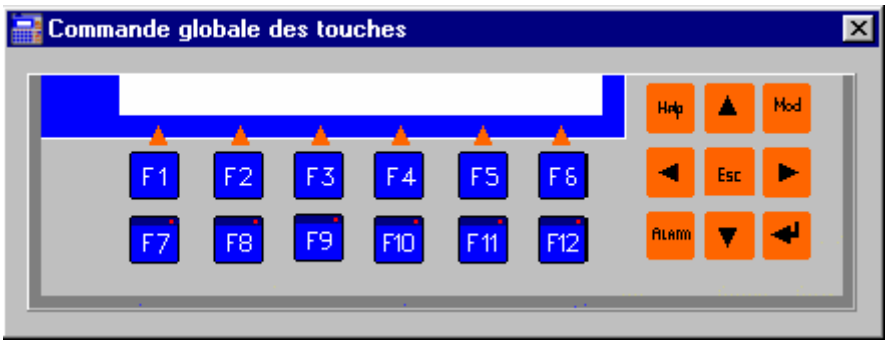
7-3- Suppression d'une variable

La suppression d'une variable se fait en supprimant tous les caractères qui composent le nom de la variable et en validant la case par entrée. La validation entraîne la suppression de tous les champs suivants.

8- COMMANDE GLOBALE DES TOUCHES

8-1- Présentation générale





L'accès à l'une de ces fenêtres (suivant le type de terminal) se fait par l'icône  . A partir de cette fenêtre, on peut définir une fonction particulière associée à une touche. La fonction associée sera globale. C'est à dire qu'elle sera tout le temps active sauf si la fonction de la touche est définie localement lors de l'édition d'une page et que cette page est présente sur l'écran. Ceci indique donc qu'une fonction de touche déclarée en locale sera prioritaire sur une fonction de touche déclarée en globale.

Les touches dont on peut définir une fonction sont les touches :

⇨ 2, 3, 4, 5, F1, F2, F3, F4, F5, F6 pour le dialog80

⇨ F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12 pour le dialog640

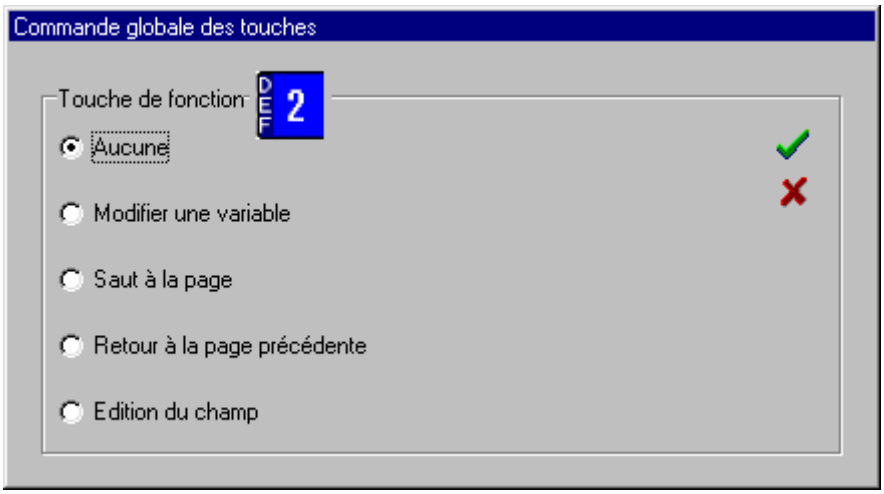
Sur chaque terminal, on peut aussi programmer la touche ESC. De même les leds associées à certaines touches peuvent être pilotées. Pour pouvoir définir une fonction pour une touche, il faut réaliser un clique gauche sur celle-ci.

8-2- Programmation des touches dynamiques

Les touches auxquelles on peut associer une fonction sont :

⇨ 2, 3, 4, 5 pour le dialog80

⇨ F1, F2, F3, F4, F5, F6 pour le dialog640



Les fonctions associées à une touche sont au nombre de 3 :

↳ **Modifier une variable :** Cette fonction permet sur l'appui de la touche de modifier un bit d'une variable. Pour cette fonction, il faut préciser la variable et le numéro du bit à modifier (bit 1 est le bit de poids faible de la variable) . La modification du bit peut être de plusieurs types (paramètre à spécifier) :

⇨ directe : le bit est à 1 lorsque la touche est enfoncée et à 0 lorsque celle-ci est relâchée.

⇨ flip-flop : un appui sur la touche provoque le basculement su bit de l'état 1 à 0 ou de l'état 0 à 1.

⇨ bit set : L'appui sur la touche provoque le passage du bit à 1.

⇨ bit reset : L'appui sur la touche provoque le passage du bit à 0.

↳ **Saut à la page :** L'appui sur cette touche permet d'afficher une autre page. Cette page est à spécifier et à définir par l'utilisateur. Sur l'appui de la touche une page de confirmation peut-être intercalée. Celle-ci est prédéfini et n'a pas besoin d'être définie par l'utilisateur.

↳ **Retour à la page précédente :** L'appui sur la touche permet de revenir à la page affiché précédemment.

```
Exemple 1:
Touche 2  :    Fonction  : Modifier une variable  ;
          Type=directe  ; Variable=Var  ; Bit n°6
Touche 3  :    Fonction  : Modifier une variable  ;
          Type=flip-flop  ; Variable=Var  ; Bit n°3
Touche 4  :    Fonction  : Modifier une variable  ;
          Type=bit set  ; Variable=Var  ; Bit n°5
Touche 5  :    Fonction  : Modifier une variable  ;
          Type=bit reset  ; Variable=Var  ; Bit n°5
Lors du fonctionnement du terminal et à condition qu'une fonction des touches
2, 3, 4, 5 n'ait pas été programmée localement dans une page, toutes les
opérations suivantes donnent les résultats suivants  :
⇨ Touche 2 enfoncée  : Var=XX1XXXXXb
⇨ Touche 2 relâchée  : Var=XX0XXXXXb
⇨ Si Var=XXXXX0XXb et un appui sur touche 3  : Var=XXXXX1XXb
⇨ Si Var=XXXXX1XXb et un appui sur touche 3  : Var=XXXXX0XXb
⇨ Un appui sur la touche 4  : Var=XXX1XXXXb
⇨ Un appui sur la touche 5  : Var=XXX0XXXXb
Exemple 2:
Touche déclarée en locale dans PageCourante  :
Touche 4  :    Fonction  : Retour à la page précédente
Touche 5  :    Fonction  : Retour à la page précédente
Touche déclarée en globale  :
Touche 2  :    Fonction  : Saut à la page  ;
          Nom de la page  : PageSuivante
```

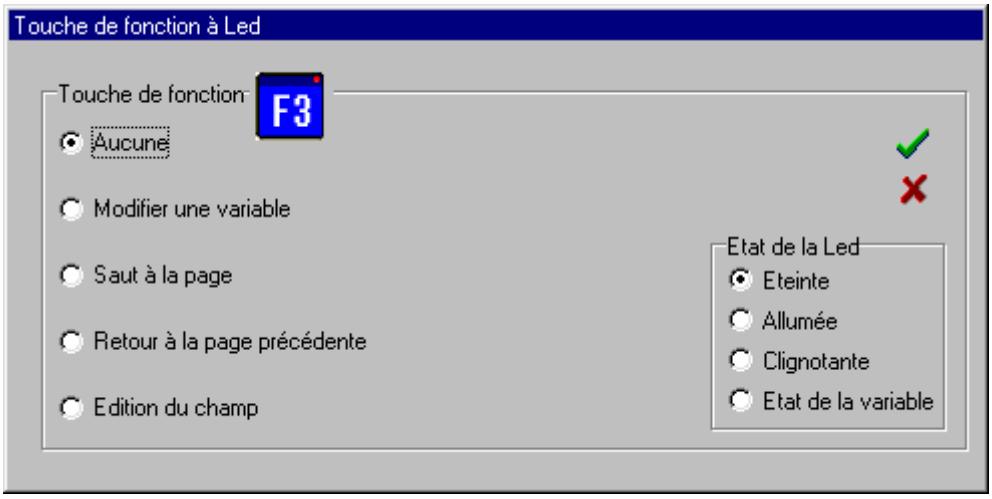
```
Touche 3  :    Fonction  : Retour à la page précédente  ;
Touche 4  :    Fonction  : Saut à la page  ;
          Nom de la page  : PageSuivante
Lors de l'affichage de la mise sous-tension toutes les opérations suivantes
donnent les résultats suivant  :
⇨ Un appui sur la touche 2 provoque l'affichage de  PageSuivante.
⇨ Un appui sur touche 3  provoque le retour à la dernière page affichée.
⇨ Un appui sur la touche 4 provoque le saut à Page suivante, sauf lorsque
PageCourante est affichée où il  provoque le retour à la dernière page
affichée.
⇨ Un appui sur la touche 5  provoque le retour à la dernière page affichée
lorsque PageCourante est affichée.
```

8-3- Programmation des touches de fonctions à leds

Les touches auxquelles on peut associer une fonction sont :

⇨ F1, F2, F3, F4, F5, F6 pour le dialog80

⇨ F7, F8, F9, F10, F11, F12 pour le dialog640



Les fonctions associées à une touche sont au nombre de 3 :

↳ **Modifier une variable :** Cette fonction permet sur l'appui de la touche de modifier un bit d'une variable. Pour cette fonction, il faut préciser la variable et le numéro du bit à modifier (bit 1 est le bit de poids faible de la variable) . La modification du bit peut être de plusieurs types (paramètre à spécifier) :

⇨ directe : le bit est à 1 lorsque la touche est enfoncée et à 0 lorsque celle-ci est relâchée.

⇨ flip-flop : un appui sur la touche provoque le basculement su bit de l'état 1 à 0 ou de l'état 0 à 1.

⇨ bit set : L'appui sur la touche provoque le passage du bit à 1.

⇨ bit reset : L'appui sur la touche provoque le passage du bit à 0.

↳ **Saut à la page :** L'appui sur cette touche permet d'afficher une autre page. Cette page est à spécifier et à définir par l'utilisateur. Sur l'appui de la touche une page de confirmation peut-être intercalée. Celle-ci est prédéfini et n'a pas besoin d'être définie par l'utilisateur.

↳ **Retour à la page précédente :** L'appui sur la touche permet de revenir à la page affiché précédemment.

L'état de la led associée à une touche peut-être pilotée localement de 4 manières différentes :

↳ **Eteinte :** la led ne sera jamais allumée

- **Allumée :** la led sera toujours allumée
- **Clignotante :** la led clignotera
- **Etat de la variable :** la led sera l'image du bit de la variable spécifiée dans les paramètres de la fonction de la touche Modifier une variable.

Exemple 1:

```
Touche F1  :   Fonction Touche  : Modifier une variable  ;
               Type=directe  ; Variable=Var  ; Bit n°6
               Fonction Led  : Etat de la variable
Touche F2  :   Fonction  Touche : Modifier une variable  ;
               Type=flip-flop  ; Variable=Var  ; Bit n°3
               Fonction Led  : Etat de la variable
Touche F3  :   Fonction  Touche : Modifier une variable  ;
               Type=bit set  ; Variable=Var  ; Bit n°5
               Fonction Led  : Etat de la variable
Touche F4  :   Fonction  Touche : Modifier une variable  ;
               Type=bit reset  ; Variable=Var  ; Bit n°5
               Fonction Led  : Etat de la variable
```

Lors de la mise sous tension du terminal, toutes les opérations suivantes donnent les résultats suivants :

⇨ Touche F1 enfoncée : Var=XX1XXXXXb et led allumée
⇨ Touche F1 relâchée : Var=XX0XXXXXb et led éteinte
⇨ Si Var=XXXXX0XXb et un appui sur touche F2 : Var=XXXXX1XXb et led allumée
⇨ Si Var=XXXXX1XXb et un appui sur touche F2 : Var=XXXXX0XXb et led éteinte
⇨ Un appui sur la touche F3 : Var=XXX1XXXXb et led allumée
⇨ Un appui sur la touche F4 : Var=XXX0XXXXb et led éteinte

Exemple 2:

Touche déclarée en locale sur PageCourante :

```
Touche F3  :   Fonction  Touche : Retour à la page précédente
               Fonction led  : éteinte
Touche F4  :   Fonction  : Retour à la page précédente
               Fonction led  : clignotante
```

Touche déclarée en globale :

```
Touche F1  :   Fonction  : Saut à la page  ;
               Nom de la page  : PageSuivante  ;
               Fonction led  : éteinte  ;
Touche F2  :   Fonction  : Retour à la page précédente  ;
               Fonction led  : allumée  ;
Touche F3  :   Fonction  : Saut à la page  ;
               Nom de la page  : PageSuivante  ;
               Fonction led  : clignotante  ;
```

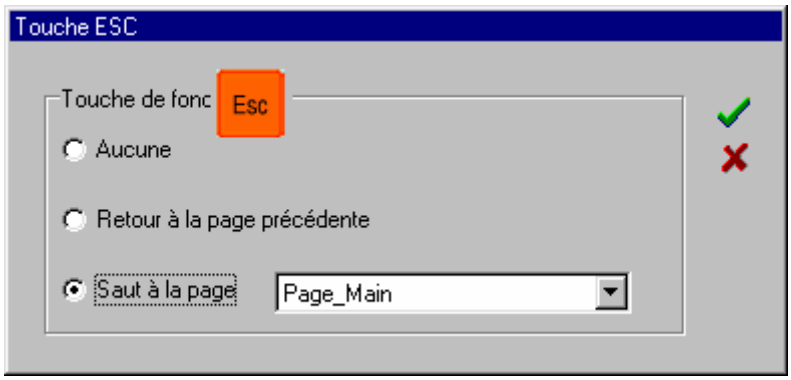
Lors de la mise sous-tension, les leds sont dans l'état suivant :

⇨ led F1 : éteinte
⇨ led F2 : allumée
⇨ led F3 : éteinte sur la page PageCourante et clignotante sur les autres
⇨ led F4 : clignotante sur la page PageCourante

Toutes les opérations suivantes donnent les résultats suivants :

⇨ Un appui sur la touche F1 provoque l'affichage de PageSuivante. Les fonctions déclarées pour les touches dynamiques et les leds de PageCourante ne sont plus valables (Uniquement celle déclarée dans PageSuivante).
⇨ Un appui sur touche F2 provoque le retour à la dernière page affichée.
⇨ Un appui sur la touche F3 provoque le saut à PageSuivante, sauf lorsque PageCourante est affichée où elle provoque le retour à la dernière page affichée.
⇨ Un appui sur la touche F4 provoque le retour à la dernière page affichée.

8-4- Programmation de la touche ESC



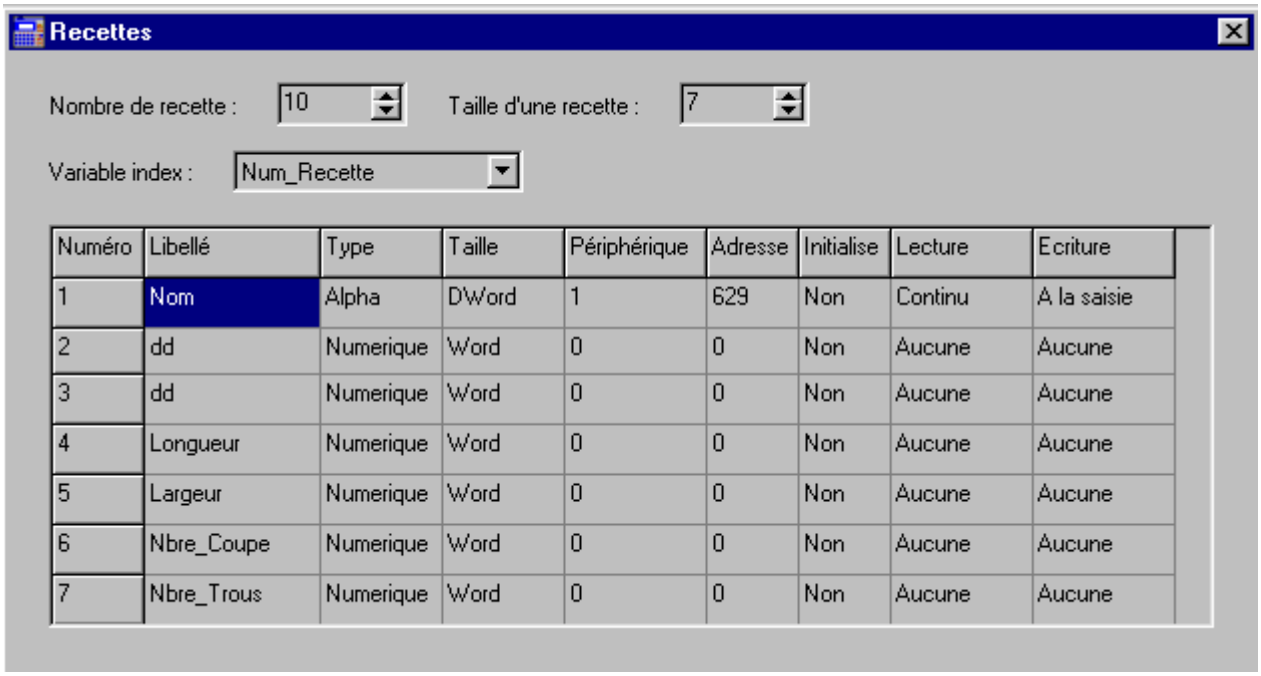
Les fonctions associées à cette touche sont au nombre de 2 :

↳ Retour à la page précédente : L'appui sur la touche permet de revenir à la page affichée précédemment.

↳ Saut à la page : L'appui sur cette touche permet d'afficher une autre page. Cette page est à spécifier et à définir par l'utilisateur.

9- LES RECETTES

9-1- Présentation générale



Numéro	Libellé	Type	Taille	Périphérique	Adresse	Initialise	Lecture	Ecriture
1	Nom	Alpha	DWord	1	629	Non	Continu	A la saisie
2	dd	Numerique	Word	0	0	Non	Aucune	Aucune
3	dd	Numerique	Word	0	0	Non	Aucune	Aucune
4	Longueur	Numerique	Word	0	0	Non	Aucune	Aucune
5	Largeur	Numerique	Word	0	0	Non	Aucune	Aucune
6	Nbre_Coupe	Numerique	Word	0	0	Non	Aucune	Aucune
7	Nbre_Trous	Numerique	Word	0	0	Non	Aucune	Aucune

Le terme recette caractérise un ensemble de paramètres liés à un produit. Par exemple pour la découpe d'une pièce, celle-ci peut-être décrite par un nom, des longueurs correspondant aux longueurs de coupe d'un côté etc.. . Cet ensemble de paramètres constitue une recette. Une recette est donc caractérisée par sa taille qui représente le nombre N de variables. Pour un système une seule recette ne convient pas. On définit autant de recettes qu'il peut alors y avoir de produits différents. Le nombre de recettes correspond donc au nombre M de recettes.

Le passage d'une recette à une autre se fait par l'intermédiaire d'une variable index. En effet suivant la valeur de celle-ci, on aura une seule recette de valide. Si l'on veut afficher la valeur d'une variable recette par exemple la longueur d'un côté d'une pièce à l'aide d'un champ d'affichage d'une variable numérique, on définit la propriété variable de ce champ avec la variable recette à afficher. Le terminal affichera, alors, la valeur de la longueur suivant la valeur de la variable index. Si celle-ci vaut 1, on aura à l'écran la longueur correspondant à la recette n°1. Si au contraire elle vaut 5, on aura la valeur de longueur correspondant à la recette n°5. Pour passer d'une recette à une autre, il faut donc modifier la valeur de la variable index. Il faut alors prévoir un champ de saisie pour modifier la recette sélectionnée.

Pour les terminaux Dialog80 et Dialog640, on peut réserver jusqu'à 10000 variables recettes. C'est à dire que le produit : N×M doit être inférieur à 10000. N est le nombre de variables par recette quel que soit le type (la définition d'un Word réserve quand même la place d'un DWord). Et M est le nombre de recettes. Par exemple, si on a besoin de 250 variables pour décrire un produit, on ne pourra pas avoir plus de 40 recettes différentes. Dans cet écran, on définit la taille et le nombre de recettes ainsi que les variables associées. Les valeurs de chaque variable associées aux recettes seront saisies sur le terminal.

9-2- Déclaration d'une recette

Pour définir une recette, on définit dans un premier temps la taille d'une recette. Cette taille peut-être modifier à tout moment et c'est elle qui spécifie le nombre de lignes nécessaire pour la déclaration des variables de la recette. On définit ensuite les variables d'une recette. Les

caractéristiques des variables d'une recette (Type, taille, etc… et déclaration d'une chaîne de caractères) sont identiques aux caractéristiques des variables (voir liste des variables) elle-même. Une fois cette opération terminée, on peut définir le nombre de recettes et la variable index de la recette. La variable index correspond à une variable située dans la liste des variables.

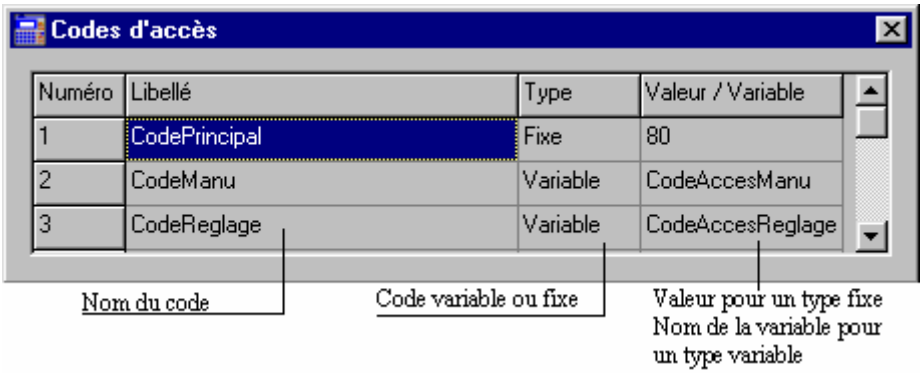
9-3- Suppression d'une recette

La suppression d'une variable recette s'obtient en supprimant le libellé de la variable désirée. Lors que ceci est réalisé, tous les champs associés à cette variable sont vides.

La suppression d'une recette complète nécessite la suppression de chaque variable la constituant et l'initialisation à 1 des champs : taille et nombre de recettes.

10- CODE D'ACCES

10-1- Présentation générale



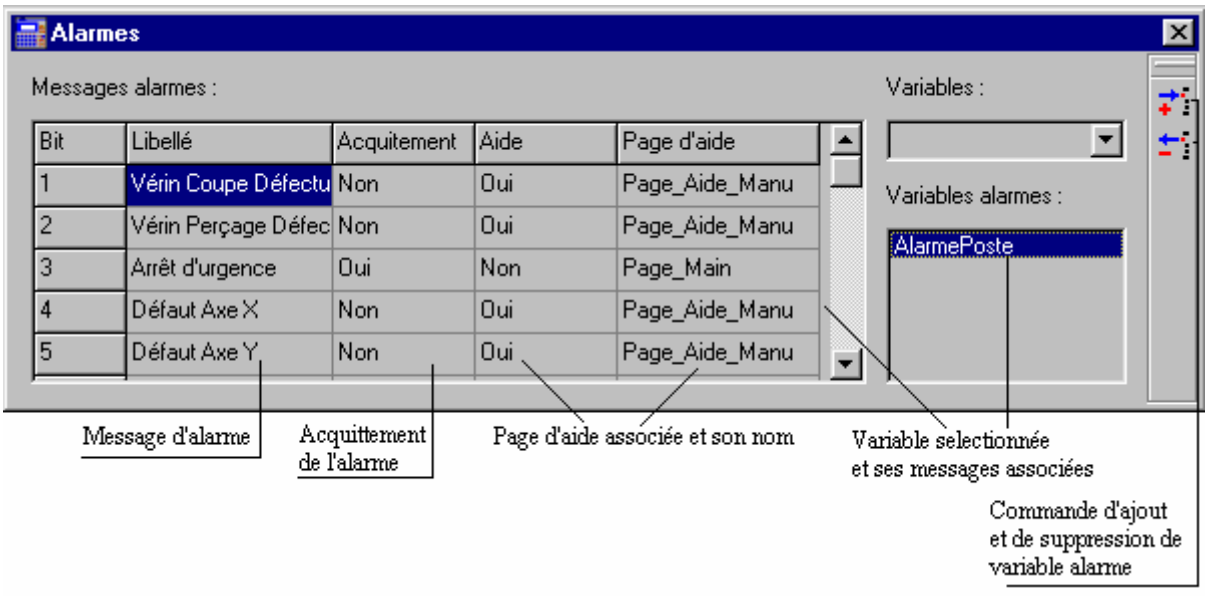
La liste des codes d'accès permet de définir le type du code. Celui-ci peut être de deux types :

- **fixe :** Dans ce cas, la valeur est à spécifier sur 4 chiffres maximum
- **variable :** Dans ce cas, on peut spécifier une variable numérique (variable de taille Word suffisant). Le contenu de celle-ci correspondra à la valeur du code.

Les codes d'accès peuvent être définis pour un enchaînement entre pages ou lors d'une saisie d'une valeur numérique ou alphanumérique. Les propriétés de ces éléments nécessitent de spécifier l'une des variables indiquées dans la liste code d'accès. La page code d'accès est prédéfinie et n'est donc pas à réaliser par l'utilisateur. Elle est réalisée de façon à ce que lorsque l'utilisateur saisit son code, les chiffres entrés sont remplacés par une étoile. Le code d'accès ne peut pas excéder plus de 4 chiffres. Etant donné que le code correspond à une valeur numérique, il est préférable de ne pas utiliser les codes d'accès commençant par un zéro type : 0XXX, 00XX, 000X et 0000. Car dans ce cas, les codes respectifs XXX, XX, X et rien ou seulement 0 sont valables.

11- ALARMES

11-1- Présentation générale



Cette fenêtre regroupe toutes les variables alarmes déclarées pour le système. Chaque bit d'une variable correspond à une alarme (bit à 0 = absence d'alarme, bit à 1 = présence d'une alarme). La détection d'une alarme à n'importe quel instant provoque le clignotement de la led « Alarm ». Dès que cette led clignote, l'utilisateur peut visualiser l'élément ou les éléments qui ont déclenchés l'alarme par l'appui sur la touche ALARM. La page d'alarme qui apparaît alors est une page prédéfinie énumérant les différents libellés des alarmes actives. Sur cette page, on peut voir aussi une flèche dans le coin haut gauche. Celle-ci indique quelle est l'alarme sélectionnée et permet de connaître ainsi les caractéristiques associées avec ce niveau d'alarme. Pour pouvoir sélectionner et connaître les autres caractéristiques d'une autre alarme, l'utilisateur peut utiliser les touches flèches vers le haut et flèche vers le bas. Les caractéristiques d'une alarme sont : la page d'aide et le mode d'acquittement. La page d'aide est une page prédéfinie par l'utilisateur qui peut être atteinte lorsque l'alarme associée est présente et que celle-ci se trouve sélectionnée dans la page d'alarme. Une page d'aide associée à une alarme se manifeste par le clignotement de la led « Help » et nécessite donc l'appui sur la touche Help pour y accéder. Une alarme peut être définie sans acquittement. C'est à dire que dès la disparition de l'alarme, celle-ci ne sera plus signalée. Avec le mode d'acquittement, le signal d'alarme ne disparaît que si le défaut associée a bien disparu et que l'utilisateur l'a acquitté. L'acquittement d'une alarme se fait dans la page d'alarme en appuyant sur la touche 2 pour le dialog80 et F1 pour le dialog640 lorsque l'alarme a acquitté est sélectionnée.

11-2- Ajout et configuration d'une alarme

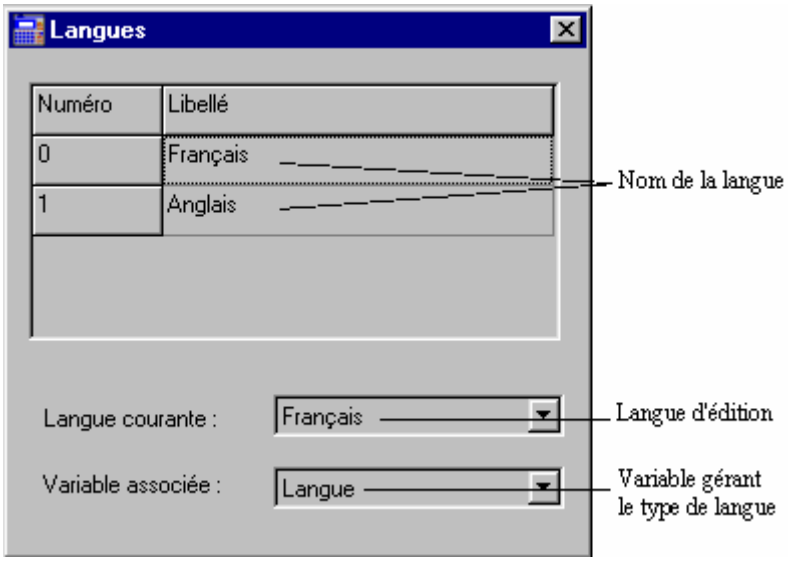
Pour ajouter une alarme, il faut sélectionner une variable image des alarmes dans la liste des variables. Puis à l'aide de l'icône  , on l'ajoute parmi la liste des variables alarmes. Dans cette liste, on la sélectionne. Si cette variable alarme n'a jamais été configurée, la liste de messages vierge apparaît. Pour la configurer, il faut définir les libellés qui correspondent aux messages d'alarmes. Chaque libellé représente un bit de la variable alarme. Si cette variable est de type Word, on aura 16 messages différents et 32 si c'est une variable de type DWord. Pour chaque message d'alarme, on peut définir son mode d'acquittement et la page d'aide associée. Pour définir un message d'alarme, on doit d'abord rentrer le libellé et ensuite ses paramètres (Acquittement, Aide, etc…).

11-3- Suppression d'une alarme

Pour supprimer une variable alarme de la liste, il suffit de sélectionner la variable à supprimer et d'activer l'icône  . Pour supprimer un message d'alarme associé à une variable alarme, il suffit de supprimer le libellé et toute la configuration associée disparaît.

12- LES LANGUES

12-1- Présentation générale



Le nombre de langues pouvant être programmé dépend du choix défini lors de la déclaration du projet. Il peut aller de 1 à 4 au maximum. Pour définir le nom de la langue, il suffit de double cliquer sur l'un des libellés et de le spécifier. La langue courante par défaut est celle ayant le numéro 0. La variable associée permet au terminal de connaître la langue à afficher. Si celle-ci vaut 0, le terminal affichera les textes dans la langue 0. Si elle vaut 1, ce sera la langue 1 etc…. Cette variable pourra être modifiée à tout moment sur le terminal et donc changer la langue. Pour définir les textes d'une langue, il faut d'abord avoir saisi tous les textes des différentes pages dans la langue par défaut. Ensuite, à partir de cette fenêtre des langues, on modifie la langue courante. Tous les textes des pages ne sont pas modifiés ou supprimés, mais restent dans la langue par défaut. Il ne reste plus qu'à les traduire dans la nouvelle langue. Après enregistrement du projet, vous pouvez passer de la langue par défaut à une autre et vous observerez que les textes apparaissent sur les pages dans la langue correspondante.

13- PROTOCOLES

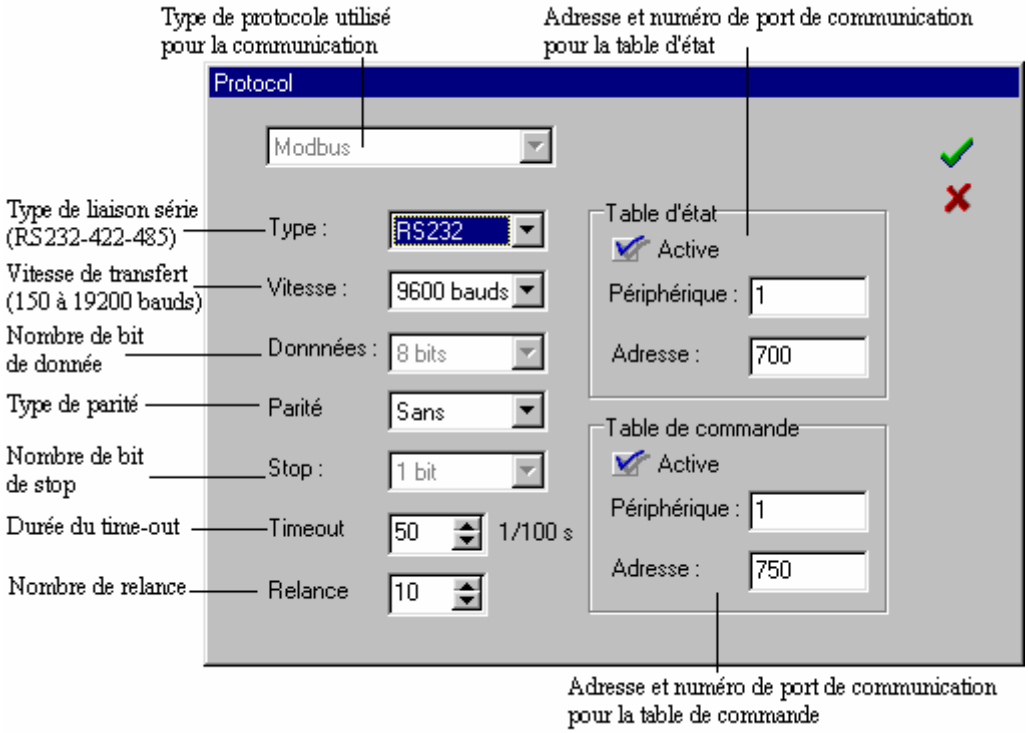
13-1- Définition du protocole

Pour déclarer le protocole de communication utilisé par le terminal dialog80 ou dialog640, il faut sélectionner la commande « Protocole » du menu Options. La sélection provoque l'activation d'une boîte de dialogue contenant les différents paramètres. Les terminaux dialog80 et dialog640 sont capables de gérer les protocoles suivants :

- MODBUS RTU MAITRE
- CANOPEN

13-2- MODBUS RTU MAITRE

13-2-1- Déclaration du ModBus

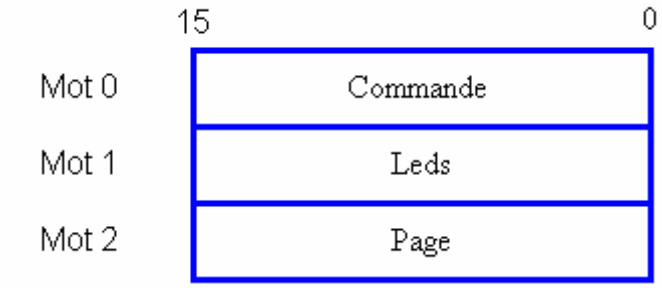


Dans cette fenêtre, on définit le protocole de communication ModBus. Ce choix nécessite un paramétrage de certaines caractéristiques de la communication. Parmi ces caractéristiques, il y a la configuration de la liaison série avec son type, sa vitesse, le nombre de données, la parité, le nombre de stop, le time-out et le nombre de relance autorisée. Il faut aussi définir la table d'état et la table de commande si celles-ci sont utilisées par le périphérique. Outre l'activation de l'une de ces tables, il faut spécifier le numéro du périphérique et l'adresse de départ de ces tables dans le périphérique. Une fois déclarée et le projet transmis au terminal, la table d'état est émise en continu vers le périphérique et la table de commande est lue en continu.

Attention : Les adresses du périphérique définies pour les différents éléments des tables doivent correspondre à des adresses qui peuvent être lues et/ou écrites.

13-2-2- Présentation de la table de commande

La table de commande est constituée de 3 mots consécutifs stockés dans le périphérique. Ils sont lus de façon continue par le terminal. Suivant l'état de ces mots, le pupitre exécute des commandes indiquées par le périphérique : forçage de l'état des leds, affichage d'une page….



↳ Le mot 0 permet d'activer ou non chacune des commandes disponibles. Le bit0, s'il est à l'état 1, permet de forcer l'état des leds contenu dans le mot 1. Le bit1, s'il est à l'état 1, permet d'afficher la page dont le numéro est contenu dans le mot 2. Le bit2, s'il est à l'état 1, permet d'inhiber les touches clavier. Le bit3, s'il est à l'état 1, permet de forcer le buzzer du terminal.

↳ Le mot 1 est réservé pour définir l'état des leds. Ce mot est la constitution de deux octets Led ON (bit8 à bit15) et Led OFF (bit0 à bit7) dont chacun des bits permet de définir un état de chacune des leds. L'attribution des bits pour les leds est la suivante :

Led F1 à F6 pour le dialog80 : bit 0 à bit 5 et bit 8 à bit 13

Led F7 à F12 pour le dialog640 : bit 0 à bit 5 et bit 8 à bit 13

Led HELP : bit 6 et bit 14

Led ALARM : bit 7 et bit 15

Pour une combinaison des deux bits des octets Led ON et Led OFF correspond un état des leds :

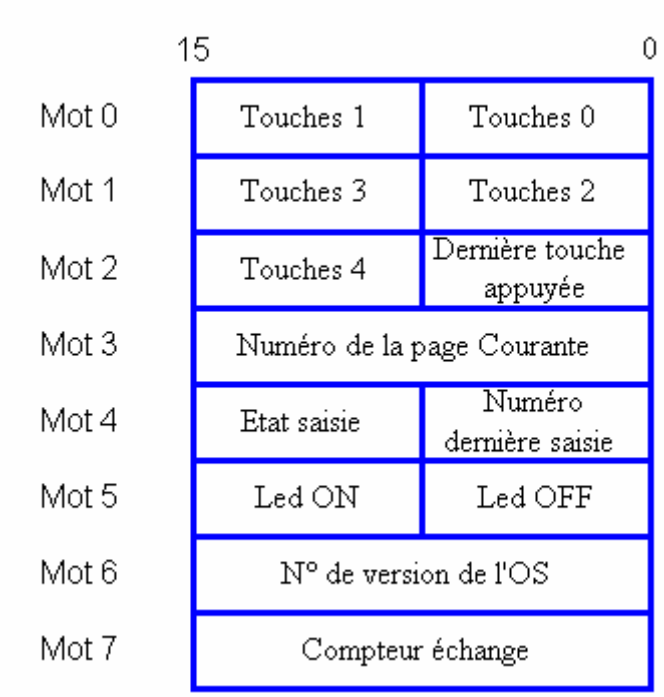
⇨ Led ON et Led OFF = 0 :Led éteinte

⇨ Led ON = 1 et Led OFF = 0 :Led clignotante

⇨ Led ON et Led OFF = 1 :Led allumée

13-2-3- Présentation de la table d'état

La table d'état est envoyée de façon continue dans le périphérique. Elle est constituée de la manière suivante :



↳ Chaque bit des octets « Touches 0 » à « Touches 4 » correspond à l'état d'une touche. Lorsque l'un de ces bits est à 1, il correspond à l'état d'enfoncement d'une touche. Dans l'autre cas, la touche est relâchée. En raison du moins grand nombre de touches du Dialog80, l'octet « Touche 4 » est inutilisé pour ce terminal. Les correspondances des bits avec les touches est donnée ci-dessous suivant le type de terminal utilisé.

Touches[0]								
Bits	0	1	2	3	4	5	6	7
Dialog 80	F1	F2	F3	Enter	F. Bas	F6	F5	F4
Dialog 640	F7	F8	F12	Aucune	Aucune	F11	F10	F9
Touches[1]								
Bits	0	1	2	3	4	5	6	7
Dialog 80	6	7	8	F. Droite	ESC	F. Gauche	0	9
Dialog 640	F1	F2	F6	Aucune	Aucune	F5	F4	F3
Touches[2]								
Bits	0	1	2	3	4	5	6	7
Dialog 80	1	2	4	+/-	F. Haut	.	5	4
Dialog 640	Aucune	9	Enter	F. Droite	mod	.	3	6
Touches[3]								
Bits	0	1	2	3	4	5	6	7
Dialog 80	Aucune	Aucune	Aucune	mod	Alarm	Help	Shift	Aucune
Dialog 640	Aucune	8	F.Bas	Esc	F.Haut	0	2	5
Touches[4]								
Bits	0	1	2	3	4	5	6	7
Dialog 80	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
Dialog 640	Aucune	7	Alarm	F. Gauche	Help	+/-	1	4

↳ L'octet dernière touche appuyée donne la valeur de la touche qui a été appuyée en dernier. La correspondance entre les touches et les valeurs décimales est la suivante :

⇨ les touches F1 à F12 : de 97 à 108 (61h à 6Ch)

⇨ les touches Right, Left, Up, Down : de 28 à 31 (1Ch à 1Fh)

⇨ la touche ESC : 27 (1Bh)

⇨ la touche MOD : 77 (4Dh)

⇨ la touche ALARM : 76 (4Ch)

⇨ la touche HELP : 69 (45h)

⇨ la touche POINT : 46 (2Eh)

⇨ la touche RETURN : 13 (0Dh)

⇨ la touche SHIFT : 72 (48h)

⇨ la touche SIGN : 83 (53h)

⇨ les touches 0 à 9 : de 48 à 57 (30h à 39h)

↳ Numéro de la page courant renvoie l'index de la page courante affichée sur le terminal. Cet index correspond à celui indiqué dans la liste des pages.

↳ Numéro dernière saisie indique le numéro de champ de saisie qui a été activé en dernier. Le numéro renvoyé correspond au numéro d'ordre de création du champ de saisie.

↳ Etat saisie renvoie l'état de la dernière saisie. Le bit8 correspond à une saisie qui s'est déroulée correctement et qui a été validée par RETURN. Le bit9 correspond à l'indication d'une saisie en cours. Les autres bits sont inutilisés.

↳ Les octets Led ON et Led OFF permettent de connaître l'état des leds du dialog80 et du dialog640.

Led F1 à F6 pour le dialog80 : bit 0 à bit 5 et bit 8 à bit 13

Led F7 à F12 pour le dialog640 : bit 0 à bit 5 et bit 8 à bit 13

Led HELP : bit 6 et bit 14

Led Alarm : bit 7 et bit 15

Pour une combinaison des deux bits des registres Led ON et Led OFF correspond un état de la led.

⇨ Led ON et Led OFF = 0 : led éteinte

⇨ Led ON = 1 et Led OFF = 0 : led clignotante

⇨ Led ON et Led OFF = 1 : led allumée

↳ Le registre numéro de version fournit la version du système d'exploitation du terminal sous forme BCD. (Par exemple 0100 pour une version 1.0)

↳ Le registre compteur d'échange s'incrémente de 1 à chaque échange entre le périphérique et l'automate. Il peut notamment servir à détecter des coupures de communications.

13-3- CANopen

13-3-1- Introduction

Le bus CAN (Controller Area Network) est apparu au milieu des années 80 pour répondre aux besoins de la transmission de données dans le secteur automobile. Ce type de bus permet d'obtenir des taux de transfert jusqu'à 1Mbit/s.

Les spécifications du CAN définissent 3 couches parmi le modèle OSI : la couche physique, la couche liaison des données et la couche application. La couche physique définit le mode de transmission des données en fonction du support de transmission. La couche liaisons des données représente le noyau du protocole CAN puisque cette couche est responsable de la trame à envoyer, de l'arbitrage, de la détection des erreurs, etc… . La dernière couche est la couche

application appelée aussi CAL (CAN Application Layer). Celle-ci est donc une description générale du langage pour les réseaux CAN qui offre de nombreux services de communication.

CANopen est un type de réseau qui est basé sur le système du bus série et de la couche application CAL. CANopen ne propose qu'une partie des services de communication offerte par CAL. Ce sont les avantages nécessaires dont ont besoins les ordinateurs ayant des performances réduites et des capacités de stockage faible.

Le CANopen est, par conséquent, une couche application standardisée par les spécifications du CIA (CAN In Automation) : DS-201…DS-207.

Le gestionnaire du réseau permet une initialisation simplifiée du réseau. Le réseau peut être étendu avec tous les composants que l'utilisateur désire.

Le bus CAN est un bus multi-maître. Contrairement aux autre bus de terrain, ce sont les messages qui sont identifiés et non les modules connectés. Les éléments du réseau sont autorisés à envoyer leurs messages à chaque fois que le bus est libre. Les conflits sur le bus sont résolus par un niveau de priorité donné aux messages. Le bus CAN émet des messages qui sont divisés en 2032 niveaux de priorités. Tous les éléments du réseau ont les mêmes droits et donc cette communication n'est seulement possible que sans bus maître.

Chaque élément décide lui-même lorsqu'il veut envoyer des données. Il est cependant possible de faire envoyer des données par un autre élément. Cette demande est effectuée par la trame distante.

Les spécifications du CANopen (DS-201…DS-207) définissent les caractéristiques techniques et fonctionnelles que nécessitent un appareil individuel pour être associé sur le réseau. Le bus CANopen fait une distinction entre les appareils serveurs et les appareils clients.

13-3-2- La communication CANopen

Le profil de la communication du CANopen permet de spécifier les informations pour l'échange de données en temps réel et des paramètres. Le CANopen utilise des services optimisés suivant les différentes sortes de données.

- PDO (Process Data Object)
  - ⇨ Echange de donnée en temps réel
  - ⇨ Identifiant à haute priorité
  - ⇨ Transmission synchrone ou asynchrone
  - ⇨ Maximum de 8 octets (un message)
  - ⇨ Format prédéfini
- SDO (Service Data Object)
  - ⇨ Accède au dictionnaire des objets d'un appareil
  - ⇨ Identifiant à basse priorité
  - ⇨ Transmission asynchrone
  - ⇨ Données distribuées dans plusieurs télégrammes
  - ⇨ Données adressées par un index

Les caractéristiques diffusées par le CAN sont reçues et évalués par tous les appareils connectés. Chaque service d'un appareil CAN est paramétré par un COBID (Communication OBject Identifier). Le COBID est un identifiant qui caractérise le message. C'est ce paramètre qui permet d'indiquer à un appareil si le message doit être traiter. Pour chaque service (PDO ou SDO), il est nécessaire de spécifier un COBID à l'émission (envoi d'un message) et un COBID à la réception (récupération de massage). Pour le premier SDO serveur, le COBID est fixe est ne

peut pas être modifié à distance. De plus, il est calculé à partir du NODE-ID. Le NODE-ID est le paramètre qui caractérise l'appareil et qui permet d'accéder de façon unique à l'appareil.

PDO (Process Data Object)

C'est un échange de donnée arbitré entre deux modules. Les PDO peuvent transférer alternativement des synchronisations ou des événements contrôlés pour réaliser la demande d'envoi des messages. Avec le mode d'événements contrôlés, la charge du bus peut être réduite au minimum. Un appareil peut, donc, réaliser une communication à haute performance avec un faible taux de transfert.

L'échange de donnée avec le PDO utilise les avantages du CAN :

- L'envoi de message peut être déclenché par un événement asynchrone. (événements contrôlés)
- L'envoi de message peut être déclenché sur la réception d'un événement de synchronisation.
- Récupération par une trame à distance.

SDO (Service Data Object)

C'est un échange de donnée point à point. Un appareil vient faire une demande d'accès dans la liste d'objets d'un SDO. Le SDO renvoie une information correspondant au type de requête fait par le demandeur. Chaque SDO peut être client (demande d'un message) et/ou serveur (attente d'une requête). Un SDO serveur ne peut pas faire de demande envers un autre SDO par contre lui peut répondre à toute demande d'un SDO client. Contrairement aux PDO, les SDO doivent suivre un protocole de communication particulier. La trame envoyée est composée de 8 octets :

- Domain Protocol (Octet 0) : il définit la commande (Upload, Download,….)
- Index sur 16 bits (Octet 1 et 2) : il définit l'adresse du dictionnaire des objets
- Sub-index sur 8 bits (Octet 3) : il définit l'élément de l'objet sélectionné dans le dictionnaire
- Paramètre (Octet 4 à 7) : Il définit la valeur du paramètre lu ou écrit.

Le gestionnaire de réseau comporte un mode simplifié de démarrage du réseau. La configuration du réseau n'est pas nécessaire dans tous les cas. La configuration par défaut des paramètres est donc parfois suffisante. Si l'utilisateur désire optimiser le réseau CANopen ou augmenter ses fonctionnalités, il peut alors modifier lui-même ces paramètres. Dans les réseaux CANopen, tous les appareils ont les mêmes droits et l'échange des données est directement régulé entre chaques appareils participants.

Le profil d'un appareil définit les paramètres nécessaires pour une communication. Le contenu de ce profil est spécifié par le constructeur. Les appareils ayant le même profil sont directement interchangeables. La plupart des paramètres sont décrits par le constructeur. Le profil possède aussi des emplacements vides qui correspondent aux futures extensions de fonctionnalités des constructeurs.

Dans la plupart des bus maître/esclave, l'efficacité du maître détermine le comportement de tout le réseau. De plus, les esclaves ne peuvent pas directement communiquer entre eux. Toutes ces caractéristiques augmentent, donc, le nombre d'erreurs de transmission. CANopen élimine tous ces désavantages. Le comportement temporel peut être spécifié individuellement pour chaque tâche respective des appareils participants. Ainsi, le système entier de communication n'a pas besoin de plus d'efficacité si seulement certains appareils participants nécessitent plus de performance. De plus, une tâche automatique peut être séparée pour chacun des appareils participants. Ainsi, les performances disponibles du contrôleur du réseau peuvent être utilisées de manière optimales et peuvent être augmentées à tout instant par adjonction de nouveaux appareils participants.

Le mapping des variables utilisées lors des échanges de type PDO permet d'utiliser de manière optimale la bande passante actuelle du bus. CANopen détermine les valeurs en défaut de tous les paramètres.

13-3-3- Caractéristiques

Le logiciel DWIN offre les possibilités suivantes pour programmer les échanges avec le protocole CANopen :

- Un serveur SDO par défaut.
- Un client SDO pour accéder aux variables telles que les automates et les cartes PC.
- Un tableau d'affectation des périphériques permettant d'associer chaque variable du pupitre avec un emplacement dans le dictionnaire d'un équipement CANopen.

13-3-4- Dictionnaire

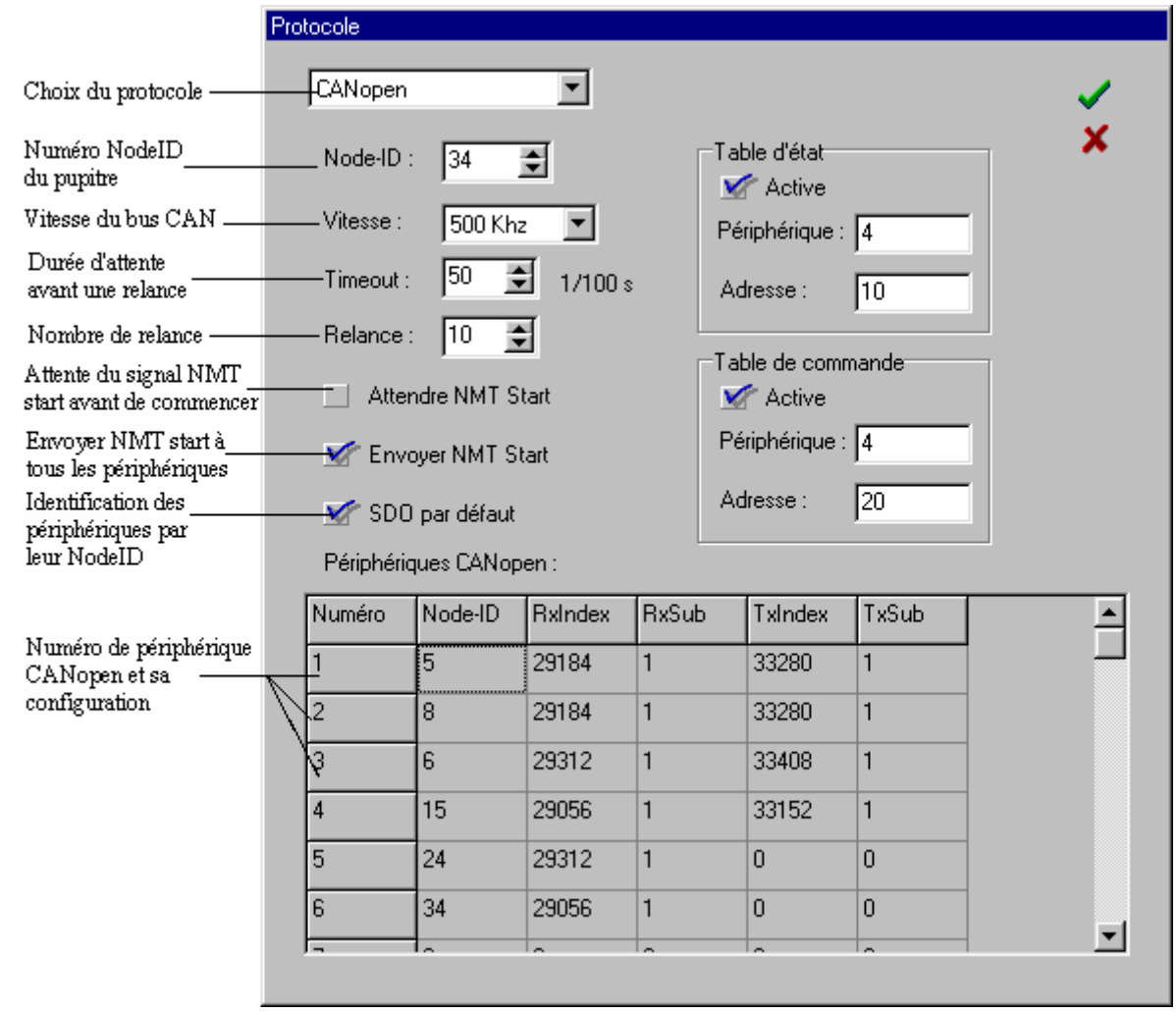
Le dictionnaire contient les différents paramètres et variables de la carte CANopen du terminal. Ces paramètres sont complètement transparents pour l'utilisateur et ne sont donc pas présentés. Par contre, il est nécessaire de retrouver les informations importantes des autres appareils pour pouvoir réaliser l'échange de variables. Voici présenté ci-dessous une partie du dictionnaire de la commande numérique MCS32 EX. Cette partie du dictionnaire rassemble toutes les variables avec leur index et les sub-index.

Index	Sub-idx	Nom	Type	Attr.	Défaut
7180	de 1 à FEh	Lecture de variables 32 bits	32 bits signé	ro	aucune
7200	de 1 à FEh	Lecture de variables 8 bits	8 bits non signé	ro	aucune
7280	de 1 à FEh	Lecture de variables 16 bits	16 bits non signé	ro	aucune
8180	de 1 à FEh	Ecriture de variables 32 bits	32 bits signé	wo	aucune
8200	de 1 à FEh	Ecriture de variables 8 bits	8 bits non signé	wo	aucune
8280	de 1 à FEh	Ecriture de variables 16 bits	16 bits non signé	wo	aucune

Ce sont ces informations qui sont nécessaire pour échanger des variables avec cet appareil. Il est aussi nécessaire de connaître les paramètres situés à l'index 1200h, 1201h, etc…. Ces paramètres correspondent aux COBID serveur de l'appareil. Les paramètres de l'index 1200h sont fixés par le NODE-ID. Cette valeur est située à l'index 100Bh et peut-être fixé sur l'appareil par des interrupteurs ou par logiciel. Pour tout autre type d'appareil, il est nécessaire de posséder le dictionnaire et de connaître les différents index mentionnés. En plus de cela, il faut repérer ce qui caractérise l'appareil (index des variables, entrées, sorties, etc…).

13-3-5- Paramétrage

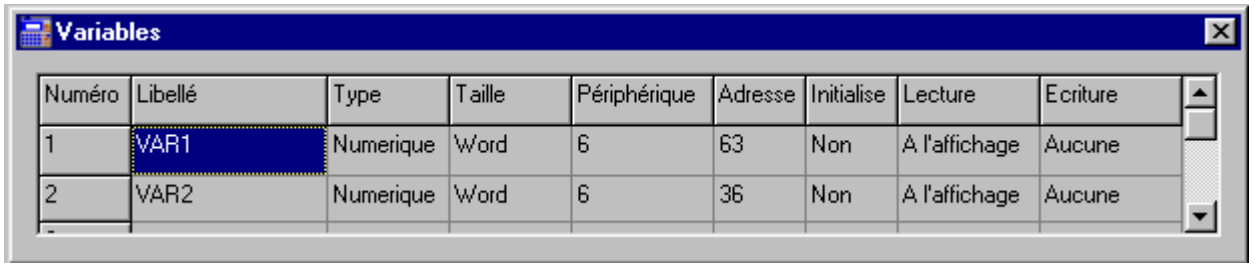
Le pupitre opérateur n'est pas paramétrable à distance par un autre périphérique CANopen. Il est donc nécessaire de le paramétrer à l'aide de la fenêtre de paramétrage du protocole CANopen. Celle-ci est définie en trois parties : la configuration du protocole, de la table des périphériques et de la table d'état et de commande.



↳ La configuration du protocole permet de définir le pupitre dans l'environnement CANopen. Cette configuration nécessite de connaître la configuration du réseau afin d'en déduire le NODE-ID et la vitesse. Le NODE-ID caractérise un matériel sur le réseau et permet donc de configurer le premier SDO serveur. De même, on configure le timeout et le nombre de relance nécessaire. On définit ensuite si le pupitre envoie un ordre d'initialisation à tous les périphériques (Envoyer NMT start) ou s'il doit le recevoir (Attendre NMT start). A aucun moment, ces deux fonctions ne doivent être validées en même temps. L'option « SDO par défaut » permet de définir si l'on veut dialoguer avec le premier SDO serveur directement. Dans ce cas, il suffit de préciser le NODE-ID de chaque appareil du réseau CAN avec lequel on veut discuter. Dans l'autre cas, il est nécessaire de spécifier le COBID à la réception et le COBID à l'émission du SDO serveur de l'appareil concerné.

↳ Un périphérique désigne un emplacement (en écriture et en lecture) dans le dictionnaire d'un équipement CANopen. Un périphérique caractérise l'accès à un champ qui peut correspondre à un tableau de variables. Et donc, deux périphériques peuvent appartenir au même équipement (même NODE-ID ou même COBID). Un périphérique est caractérisé par l'équipement auquel il appartient (NODE-ID ou COBID), le numéro de l'index et le numéro de base du sub-index du dictionnaire. On pourra ensuite attribuer à une variable de la liste des variables un numéro de périphérique qui correspond au numéro de la table des périphériques et une adresse qui correspond à une valeur de décalage du sub-index. Ce qui signifie que le numéro de périphérique fixe l'appareil en relation et l'index du dictionnaire. Le sub-index est fixé par l'addition du sub-index définit dans la table des périphériques et l'adresse de la variable. Par exemple, on désire pouvoir lire deux variables (VAR1 et VAR2) d'un périphérique ayant un NODE-ID de 22. Les emplacements à lire sont situés à l'index 7180h et aux sub-index 40h et 25h dans le dictionnaire de l'appareil. Des périphériques ont déjà été déclarés dans la table des

périphériques, ce qui implique de déclarer notre périphérique au numéro 6. Dans ce cas, on spécifie 34 en décimal (22h) pour le champ NODEID du périphérique numéro. On définit pour RxIndex la valeur de 29056 en décimal (7180h) et pour RxSub la valeur de 1. Les champs TxIndex et TxSub correspondent à l'écriture et ne sont donc pas nécessaire dans notre cas. On peut quand même les spécifier à des valeurs correspondants à une écriture des variables.



Numéro	Libellé	Type	Taille	Périphérique	Adresse	Initialise	Lecture	Ecriture
1	VAR1	Numerique	Word	6	63	Non	A l'affichage	Aucune
2	VAR2	Numerique	Word	6	36	Non	A l'affichage	Aucune

On peut ensuite définir les variables dans la liste des variables. Pour les deux variables, on spécifie le périphérique auquel elles sont liées soit le numéro 6 de la table des périphériques. Puis pour chacune d'entre elle, on définit son adresse qui correspond au décalage par rapport au sub-index. Et donc pour l'une, on aura 63 (40h-1h= 3Fh) en décimal et l'autre 36 (25h-1h= 24h) en décimal. Toutes ces étapes sont nécessaires pour configurer les variables. Il est donc nécessaire de définir autant de périphériques qu'il y a d'index. Le nombre de périphériques est limité à 16 déclarations.

↳ La configuration de la table d'état et de la table de commande nécessite la configuration d'au moins un périphérique dans la table des périphériques. En effet, il est nécessaire pour ces deux tables de pouvoir se référencer par rapport à un numéro de périphériques. De même, l'adresse à spécifier sert de valeur de décalage pour le sub-index. La définition de ces deux tables est la même que celle qui est faite dans le protocole MobBus.

13-3-6- Exemple : Liaison CANopen entre un pupitre et une MCS

Le paramétrage de la communication entre une MCS et un pupitre consiste à attribuer un NodeID a chacun. Une communication par SDO du pupitre vers la MCS est alors possible. Les COBID par défaut des serveurs SDO sont 600h+NodeID en réception et 580h+NodeID en émission. On paramètre donc les clients respectifs en conséquence.

↳ Initialisation de la MCS

```
'Démarrage de la carte à 500KBits/s sur le nœud 1
StartCan(Can1,1,5)
'COBID ClientSDO Rx (Mcs) = COBID ServerSDO Tx (terminal)
CanSetup&(Can1,1280h,1,582h)
'COBID ClientSDO Tx (Mcs) = COBID ServerSDO Rx (terminal)
CanSetup&(Can1,1280h,2,602h)
```

↳ Initialisation du pupitre

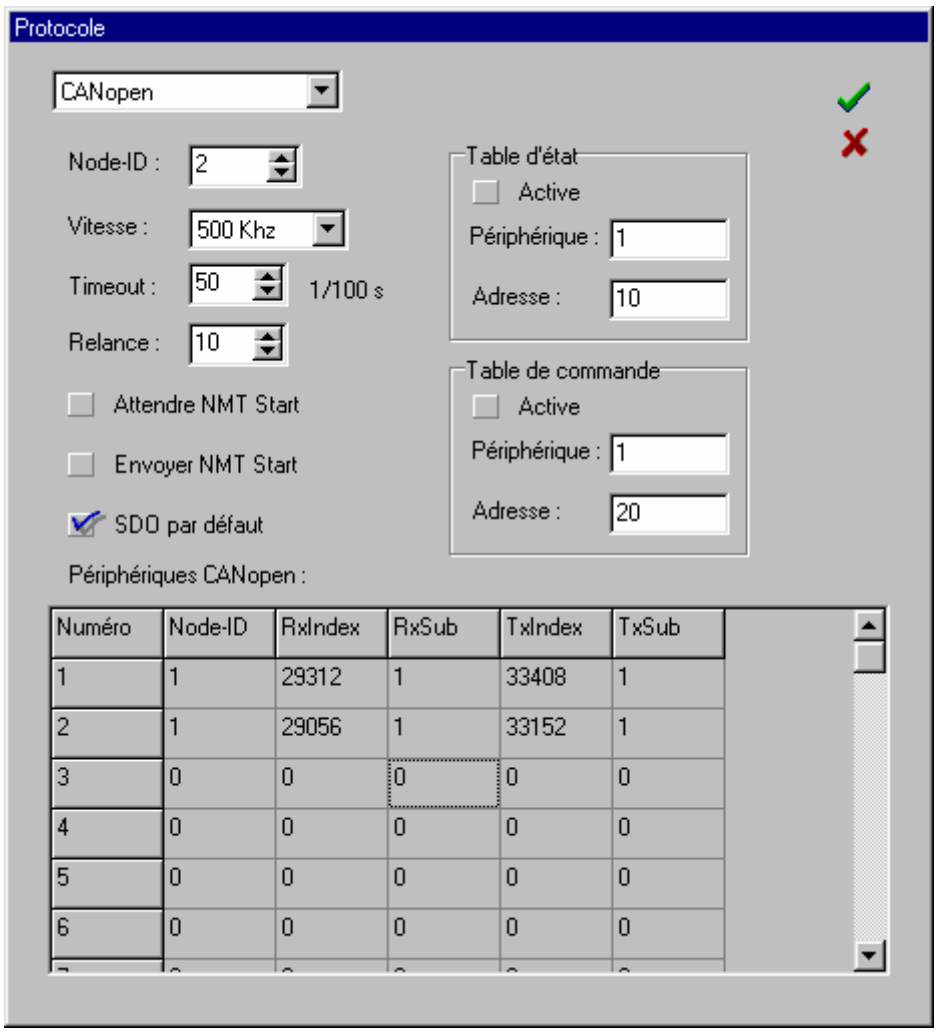
Les périphériques 1 et 2 du pupitre sont déclarés pour accéder au tableau "Lecture et écriture de variables 16 bits non signé" et "Lecture et écriture de variables 32 bits non signé" de la MCS.

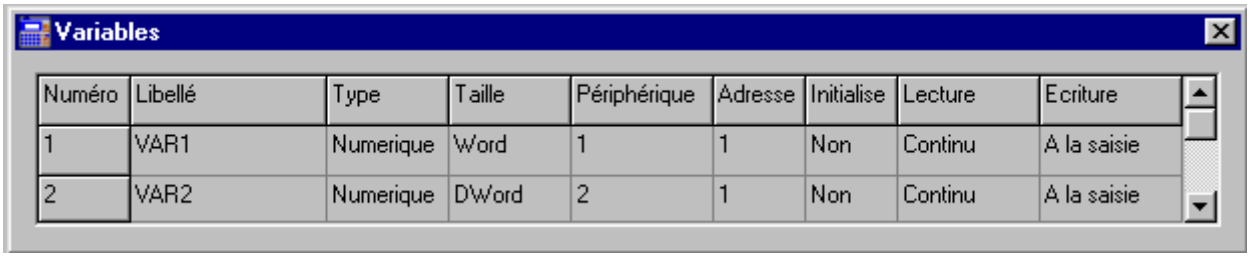
Les deux périphériques désignent le même NodeID mais les index et sous-index sont différents. En choisissant d'utiliser les SDO par défaut, le pupitre calcule les COBID en fonction des NodeID. Par la suite, en associant les variables au périphériques on associe chaque variable à un tableau CANopen différent.

Dans cet exemple, on utilise deux variables pour la communication entre la MCS et le pupitre. Une première variable (VAR1) comporte les ordres qui seront transmis à la MCS. L'autre variable (VAR2) indique la position de l'axe X et sera affichée par le pupitre.

La variable VAR1 devra être écrite dans le tableau "Ecriture de variables 16 bits non signé" du dictionnaire de la MCS et la variable VAR2 devra être lue dans le tableau "Lecture de variables 32 bits non signé" du dictionnaire de la MCS.

La configuration du pupitre est la suivante :





La variable VAR2 sera affichée de manière continue à l'écran du terminal. La variable VAR2 sera modifié par deux touches différentes. L'appui sur une touche modifiera l'état du bit n°1 et l'appui sur une autre touche modifiera l'état du bit n°2.

La MCS utilise ces informations de la façon suivante :

```
O%=CanLocal%(Can1,1)                    'Lecture de l'ordre
```

```
P&=RealToLong(Pos_S(X))
CanLocal&(Can1,1,P&)                              'Lecture de la position
If (O%=0) And (Move_S(X)=1) Then Stop(X)          'Stop si pas d'ordre
If (O%=1) And (Move_S(X)=0) Then Stti(X=+)        'Bit 1 = Jog+
If (O%=2) And (Move_S(X)=0) Then Stti(X=-)        'Bit 2 = Jog-
```

Index

A

C

D

E

I

L

M

P

S

T