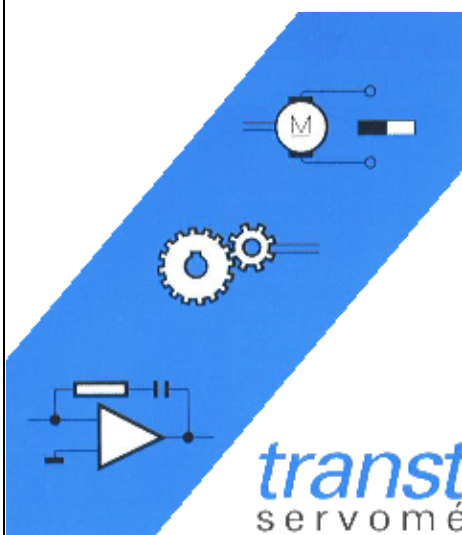


PUPIETRES OPERATEURS

VT50

Pour réaliser votre interface homme/machine
Mise en œuvre rapide

Révision 0



transtechnik
servomécanismes



SOMMAIRE

Ce manuel a pour but de mettre en œuvre rapidement un pupitre opérateur VT50 avec nos cartes d'axes TRIO. Il n'a pas la prétention de décrire toutes les fonctionnalités du système mais en permet une utilisation classique. On exploite les possibilités de la version 4.68 du logiciel de programmation ESA. Pour de plus amples détails, consulter l'aide en ligne, les manuels *405_1200_037_3-Manuel Hardware-Ver 2_11.pdf* et *405_1200_038_3-Manuel Software-Ver 2_11.pdf* dans le CD, ou notre service technique.

Sommaire.....	1
1 VTWIN.....	2
2 Création d'un nouveau projet.....	3
3 Création du programme :.....	4
3.1 Généralités.....	4
3.2 Création des pages et du séquençement entre les pages.....	5
3.3 Insertion d'une variable numérique.....	6
3.3.a Création de la variable.....	6
3.3.b Affichage de la variable.....	7
3.4 Action associée à une touche.....	8
3.5 Message conditionné.....	9
3.6 Opérations Automatiques.....	10
3.7 Zones d'échanges.....	11
3.7.a Les Zones d'entrées.....	12
3.7.b Les Zones d'état.....	13
3.7.c Les Zones de Commandes.....	14
4 Transferts du projet dans le pupitre.....	15
5 Autres possibilités.....	16
5.1 Sauvegarde du projet.....	16
5.2 Récupération du projet present dans le pupitre.....	16
5.3 Mise à jour du projet sans VTWIN.....	16
5.4 Impression du projet.....	16
6 Câblage.....	17
6.1 Connexion VT – PC.....	17
6.2 Connexion VT – TRIO.....	17

(Cliquer pour atteindre le chapitre)

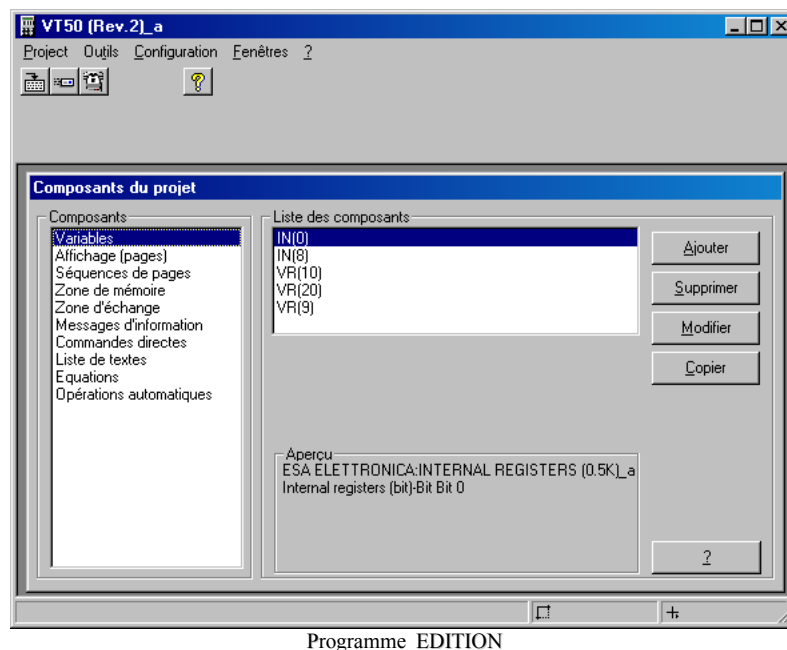
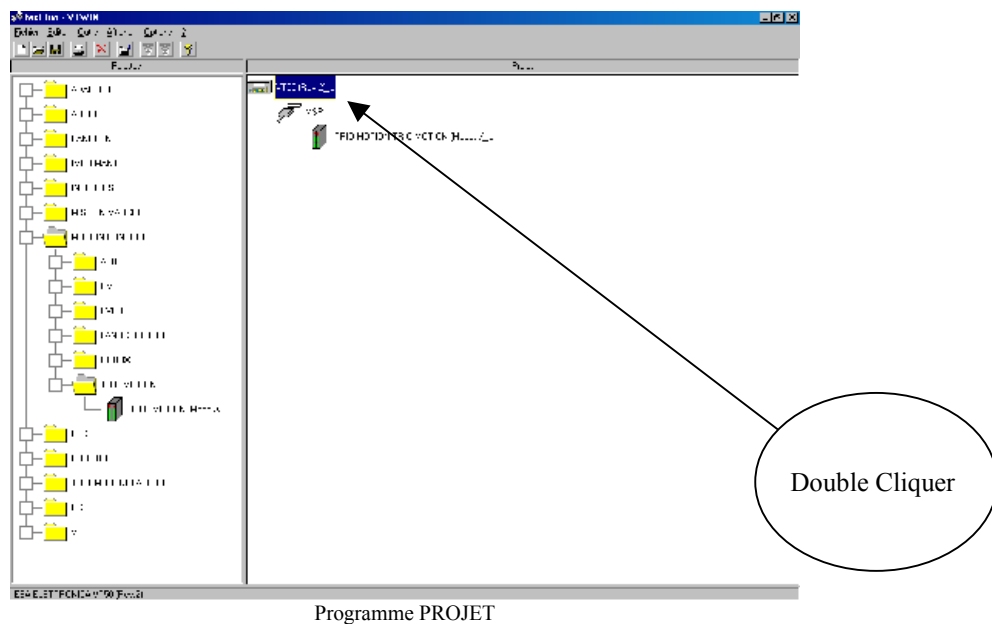
1 VTWIN

VTWIN est une application Windows permettant de programmer le pupitre. Elle fonctionne sous Windows 95 / 98 / Me / 2000 / XP / NT 4.00 SP3 ou supérieur. Dans ce manuel, on étudiera les possibilités de la version 4.68.

Dès sa mise en place dans le lecteur, le programme d'installation doit s'exécuter. Si ce n'est pas le cas, double cliquer sur le fichier Setup.exe. Une fois l'application démarrée, sélectionner 'INSTALLER VTWIN' et suivre les instructions.

VTWIN se décompose en 2 programmes : Un pour créer le projet et définir les différents éléments qui le compose, l'autre pour éditer votre programme et le transférer dans le pupitre. On atteint le second en double cliquant sur votre pupitre dans le premier programme. Pour compréhension, on appellera le premier programme 'PROJET' et le second programme 'EDITION'.

Pour travailler sur le **projet**, fermer l'**édition**



2 CREATION D'UN NOUVEAU PROJET

Tout ce passe dans le programme projet. Il s'agit de déterminer les éléments qui composent votre application:

- Démarrer l'application VTWIN et sélectionner votre pupitre : **VT50 (Rev.2)**.
- Dans la colonne de droite, sélectionner l'élément avec lequel le pupitre doit communiquer : Pour une carte d'axe TRIO, sélectionner **MOTION CONTROL \ TRIO MOTION \ TRIO MOTION (Modbus)**.
- Glisser (en sélectionnant et maintenant le bouton droit de la souris appuyé) cet élément vers **MSP**, sur la partie droite.
- Cliquer avec le bouton droit de la souris sur **MSP**, sélectionnez **Propriétés** et définissez les paramètres de la communication : Baud rate, nombre de bits, nombre de bits de stop, parité.
- Cliquer avec le bouton droit de la souris sur **TRIO MOTION:TRIO MOTION (Modbus)_a**, sélectionnez **Propriétés** et précisez l'adresse (de 1 à 255).

En fonction des valeurs précisée dans VTWIN, il faudra écrire les fonctions suivantes dans une tâche TRIO. Ces instructions ne devront être scrutées qu'une seule fois:

```
ADDRESS=1  
SETCOM(9600,8,1,2,2,4)
```

Pour plus de renseignements sur ces instructions et leur paramétrage, consulter le manuel TRIO.

3 CREATION DU PROGRAMME :

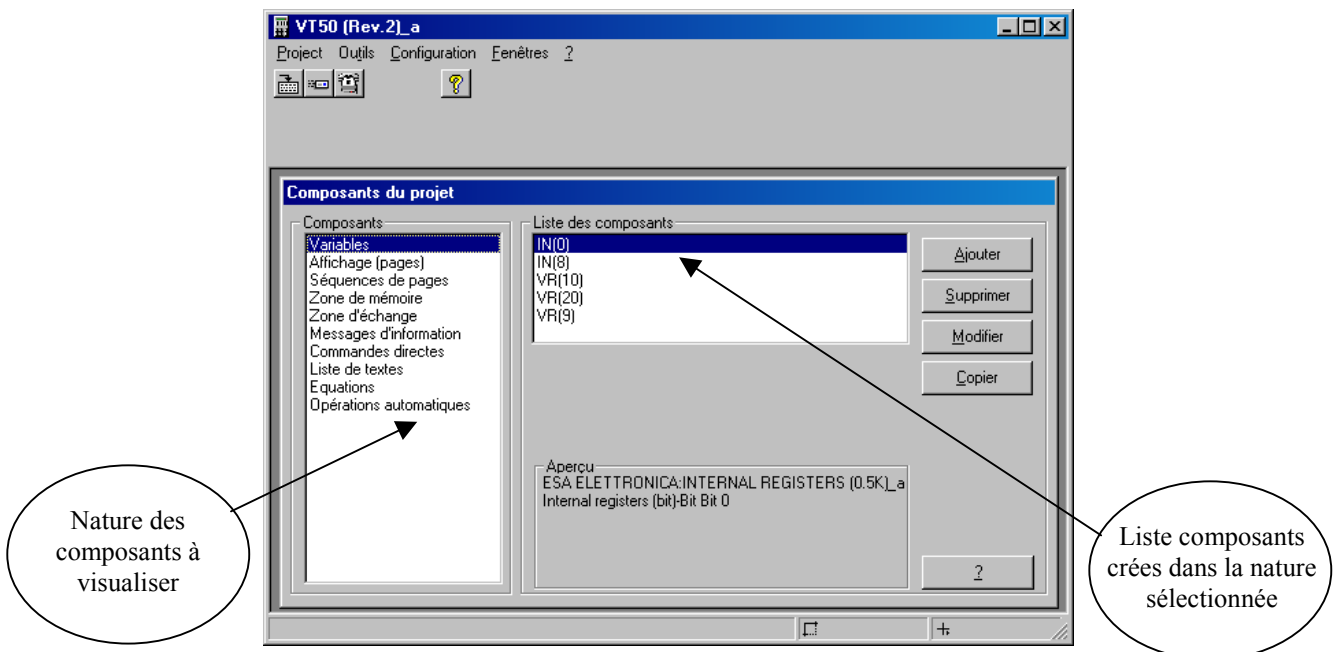
3.1 GENERALITES

Dans le programme '**Edition**', vous avez la possibilité de créer votre application et de la transférer dans le pupitre.

Un programme est constitué de **composants**. Il s'agit d'éléments simples regroupés par nature qui seront assemblés pour réaliser votre application.

Par exemple, supposons une interface simple constituée de 2 pages : 1 page d'accueil et une autre permettant d'afficher une vitesse. En réalité, votre application contiendra 2 composants **page**, 1 composant **variable** et 1 composant **séquence de page** pour définir la manière de passer d'un page à l'autre.

Pour créer un nouveau composant, double cliquer sur la nature du composant à créer (partie gauche) ou sélectionner **ajouter**. Vous pouvez aussi le créer au moment où vous en avez besoin. Par exemple, pour l'application simple, la variable peut être définie comme composant et intégrée dans la page, mais vous pouvez aussi le créer en demandant d'afficher une variable dans la page. Dans les 2 cas, le nouveau composant créé sera rangé selon sa nature et sera utilisable pour d'autres fonctions.



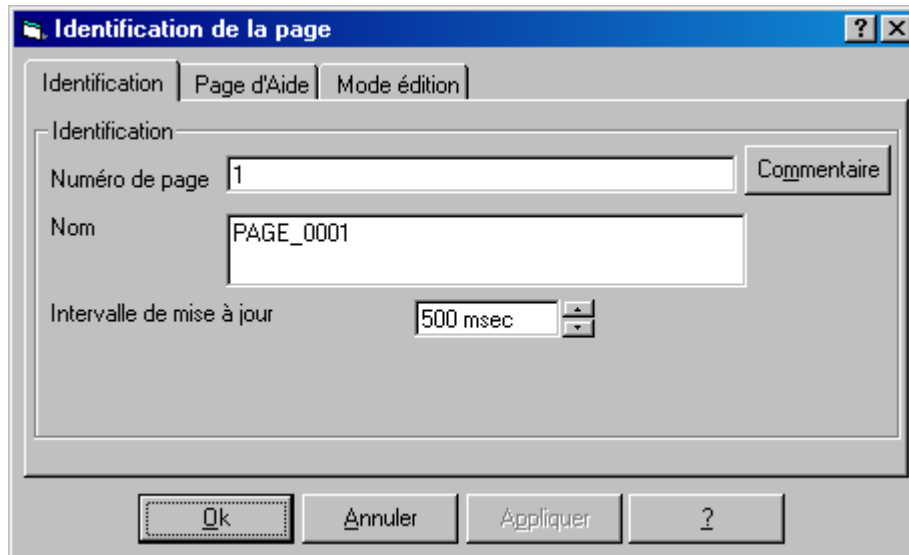
Une fois tous les composants créés et assemblés, il faudra compiler le projet et le transférer dans le pupitre.

Le pupitre exécutera le projet automatiquement à la mise sous tension mais possède aussi d'autres pages prédéfinies pour des fonctions particulières (transfert du programme depuis le PC, messages...). Quand nous sommes dans ces pages spécifiques, nous dirons que nous sommes dans un autre **contexte** que le projet.

Le composant qui communique avec le pupitre (TRIO/LUST...) est appelé le **dispositif**.

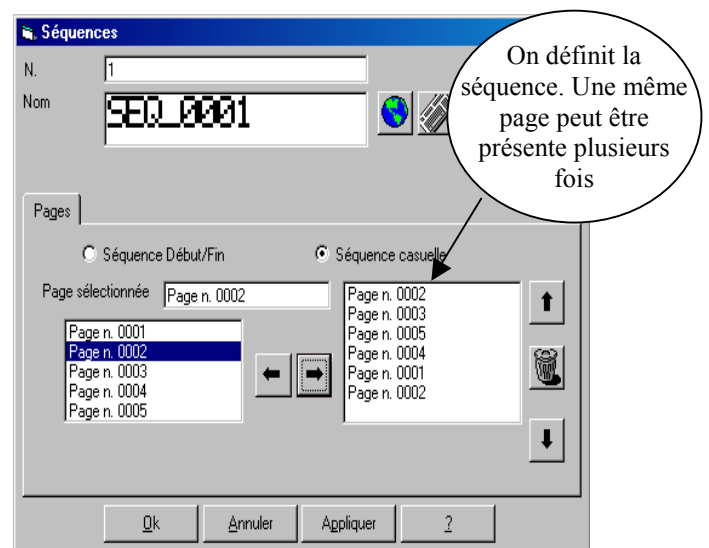
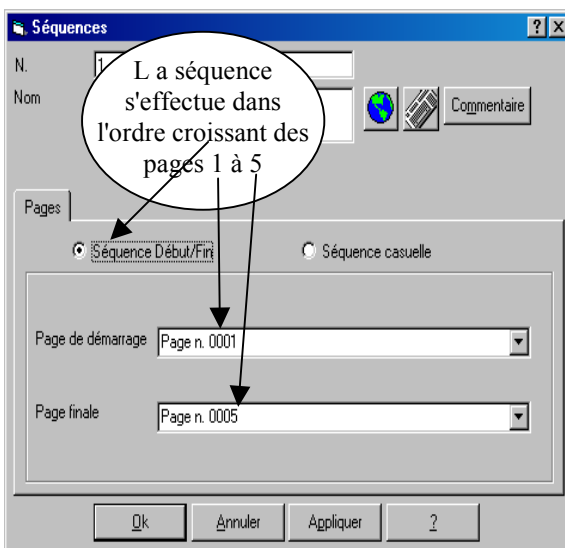
3.2 CREATION DES PAGES ET DU SEQUENCEMENT ENTRE LES PAGES

Pour créer une nouvelle page, double cliquer sur '**Affichage (pages)**' dans la partie '**Composants**'. Dans la fenêtre qui suit, vous pourrez lui donner un numéro, un nom et temps de rafraîchissement (pour la visu des variables qui vont la composer par exemple).



L'onglet '**Page d'aide**' permet de définir une page accessible par la touche aide du pupitre et l'onglet '**mode édition**' permet le passage automatique d'un champ à un autre après la modification du premier. Cliquer sur OK pour valider et faire apparaître une page vierge. Vous pouvez ensuite y ajouter des éléments en cliquant sur les icônes puis sur la partie affichage de la page

Même s'il n'y a qu'une seule page, il faut définir une '**Séquence de pages**'. Il s'agit de la première page à afficher et la chronologie de passage de l'une à l'autre. Celle-ci est forcément droite (en 'I', pas en 'Y') et peut être dans l'ordre croissant des numéros des pages ou **casuelle** (on définit l'ordre). Double cliquer sur '**Séquence de pages**' pour en créer une nouvelle :

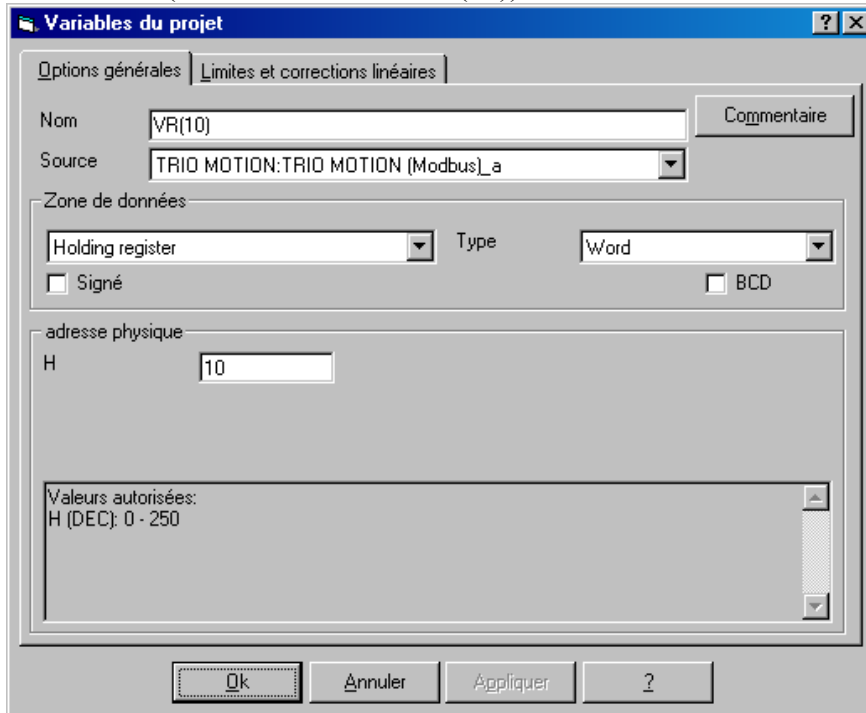


Une page et une séquence de page composent le projet minimum compilable et transférable

3.3 INSERTION D'UNE VARIABLE NUMERIQUE

3.3.a CREATION DE LA VARIABLE

Double cliquer sur '**Variables**' dans la partie '**Composants**'. Préciser un nom, la source, et le type de la variable et son adresse (adresse 10 = Visu VR(10))



Le type de la variable peut être un Bit (Coil) ou une variable (Holding register). Dans le premier cas, cela correspond au paramètre **OP** de la TRIO (les sorties). Dans le deuxième cas, on peut préciser la longueur : **Word** = 16bit = 1 VR, **Dword**=32bits=2 VR, **String** = x Word. Dans le cas d'un '**String**', seule la lecture est possible et on visualise 2 caractères par VR.

STRING : Code ASCII 1er caractère * 256 + Code ASCII 2ème caractère

	VR(x)	
Type de variable	0→32767	32768 → 65535
Word non signé	0→32767	32768 → 65535
Word	0→32767	-32768 → -1

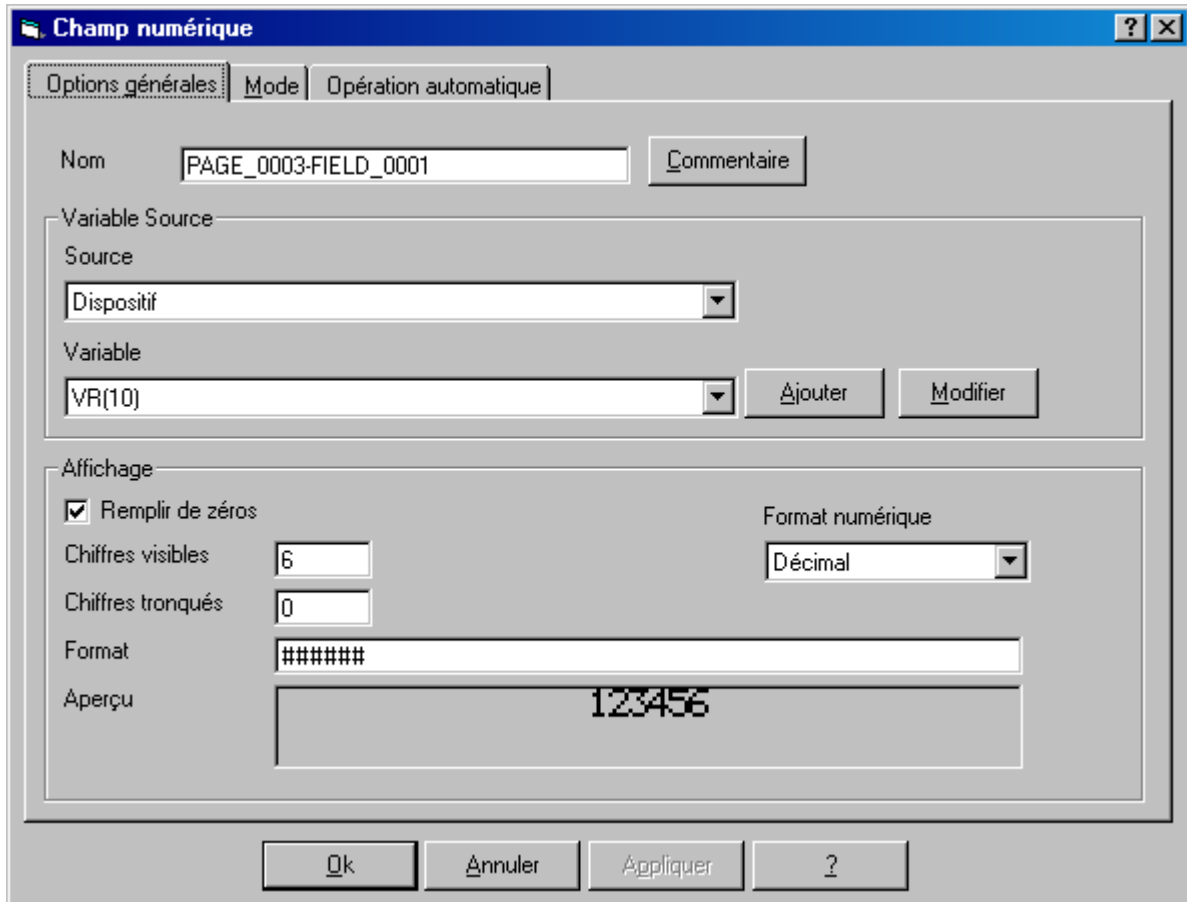
VR(x)	0	1→32767	32768→65535
VR(x+1)	0→65535	0→65535	0→65535
Dword non signé	0→65535	65536→2147483647	2147483648→4294967295
DWord	0→65535	65536→2147483647	-2147843648→-1

Dans l'onglet '**limites et corrections linéaires**' , on peut programmer les paramètres suivants:

- Limites d'entrée : Si une variable peut être modifiée par l'utilisateur, on peut préciser des butées minimales et maximales fixes. Si l'utilisateur entre une valeur hors norme, c'est la butée qui sera prise en compte.
- Correction linéaire : Pour mettre en forme la valeur à afficher. Par exemple, on veut afficher au format 0/10V une entrée analogique de la carte d'axe TRIO qui est lue au format 0/2048. Paramétrer **Terminal Mini** à 0, **Terminal Maxi** à 10, **Périphérique Mini** à 0 et **Périphérique Maxi** à 2048. Ainsi, si la variable représentant l'entrée ana vaut 1024, on affichera 5.

3.3.b AFFICHAGE DE LA VARIABLE

Dans une page, sélectionner '**Champ\numérique**' et cliquer sur l'écran pupitre à l'endroit où sera placé l'élément:



On peut préciser un nom, la source est toujours le dispositif (TRIO). Puisque la variable a déjà été créée, on peut la sélectionner, sinon, on peut en créer une autre.

Chiffres visibles : Taille de la visu en nombre de caractères.

Chiffres tronqués : On peut tronquer les chiffres de droite. Ex : Si VR(10)=200, on affiche 200 si on tronque 0 chiffre, 20 si on en tronque un, 2 si on tronque 2. A l'inverse, la valeur entrée par le pupitre sera multipliée par une puissance de 10 pour prendre en compte le tronquage.

Remplir de 0 : Les caractères libres sont remplacés par des 0. Ex : Si VR(10)= 200, on voit 000200

Format : Précise la manière d'afficher. On peut insérer une virgule (#####.## affichera 2.00) ou mettre au format heure ou date (###/###/##### affiche 12/11/1974 si la variable vaut 12111974)

Dans l'onglet **Mode**, on pourra préciser :

Index de champ : S'il y a plusieurs variables dans la page, on précise l'ordre de passage de l'une à l'autre

Mode Lecture : **Continuer** pour une mise à jour périodique de la valeur de la variable ou **Simple** pour une mise à jour unique à l'affichage de la page

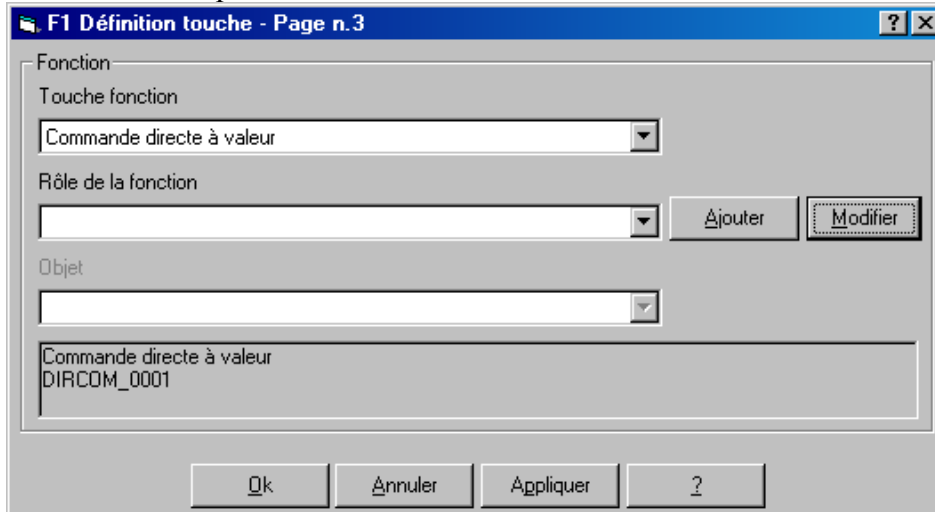
Modification validée : définit la variable en lecture seule ou en lecture/écriture. Si la variable est un '**String**', seule la lecture est possible

Le Bit de Protection sera étudié dans le chapitre 'mot de passe' et l'onglet Opération Automatique est sans intérêt.

3.4 ACTION ASSOCIEE A UNE TOUCHE

On peut associer une fonction aux touches F1 à F5 disponibles sur le pupitre si elles sont précédées de l'appui sur la touche SHIFT. Les actions associées à la touche peuvent être locales (uniquement dans une page) ou globales (valables dans toutes les pages du projet).

Pour définir une action locale, cliquer sur la touche fonction dans l'éditeur de la page. Pour une action globale, aller dans le menu '**Configuration\Définition des touches de raccourci interne**', sélectionner une touche et cliquer sur **Modifier**.



Les fonctions disponibles sont :

Désarmer touche : Enlève localement la fonction globale de la touche

Séquence : Modifie le séquençage des pages (on sélectionne une nouvelle séquence)

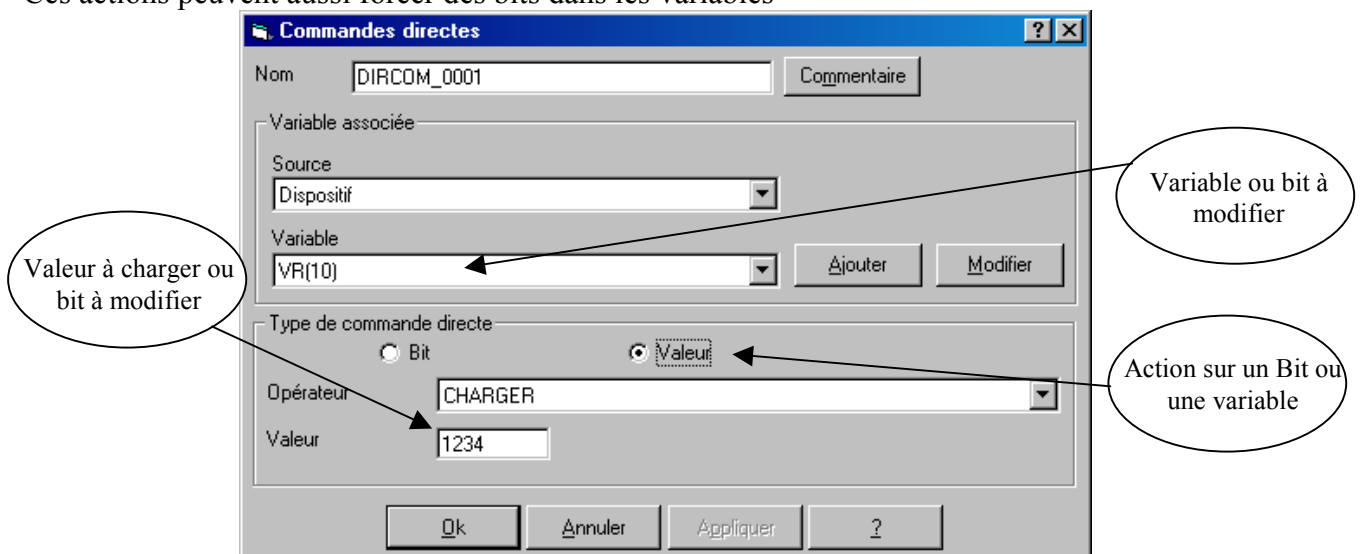
Commande interne : Essentiellement de la visu des infos projets, changement de langage ...

Chargement momentané du bit : Met à 1 un bit sur l'appui de la touche et le met à 0 sur le relâchement

Inverser Bit : Inverse le bit à chaque appui sur la touche

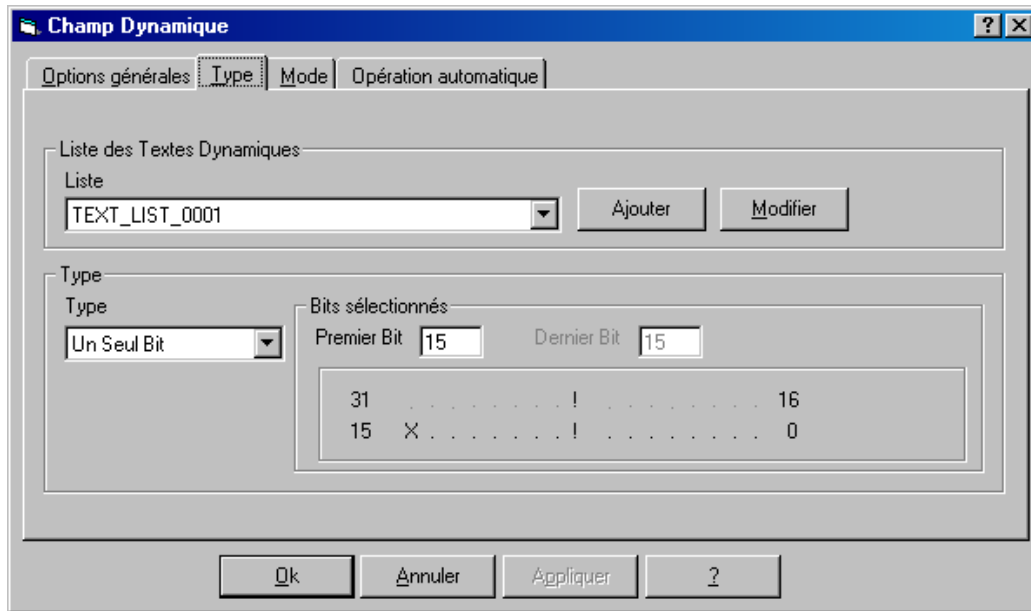
Commande directe a valeur : Ecris une valeur constante dans une variable

Pour les 3 derniers choix, créer un nouveau composant '**action directe**' en cliquant sur **Ajouter**. Ces actions peuvent aussi forcer des bits dans les variables



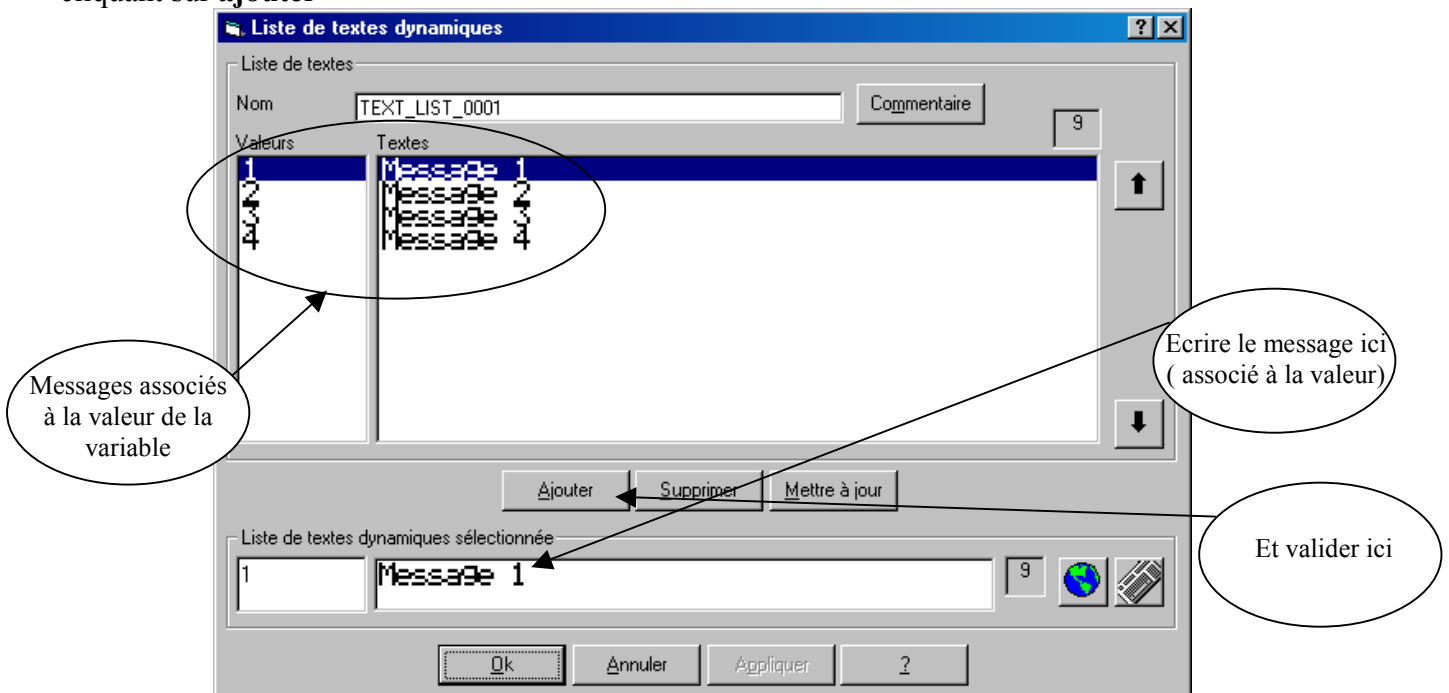
3.5 MESSAGE CONDITIONNE

Le principe de cette fonction est d'afficher une chaîne de caractères en fonction de la valeur d'une variable. Elle est très utilisée pour, par exemple, afficher ou modifier une information. Dans une page, sélectionner '**Champ\Dynamique**' et cliquer sur l'écran pupitre à l'endroit où sera placé l'élément:



Dans le premier onglet 'Options générales' on définit simplement quelle variable va conditionner le message. Le troisième onglet '**Mode**' reprend les mêmes fonctionnalités de gestion de lecture/écriture, affichage rafraîchi ou non... (voir de chapitre 'affichage de la variable')

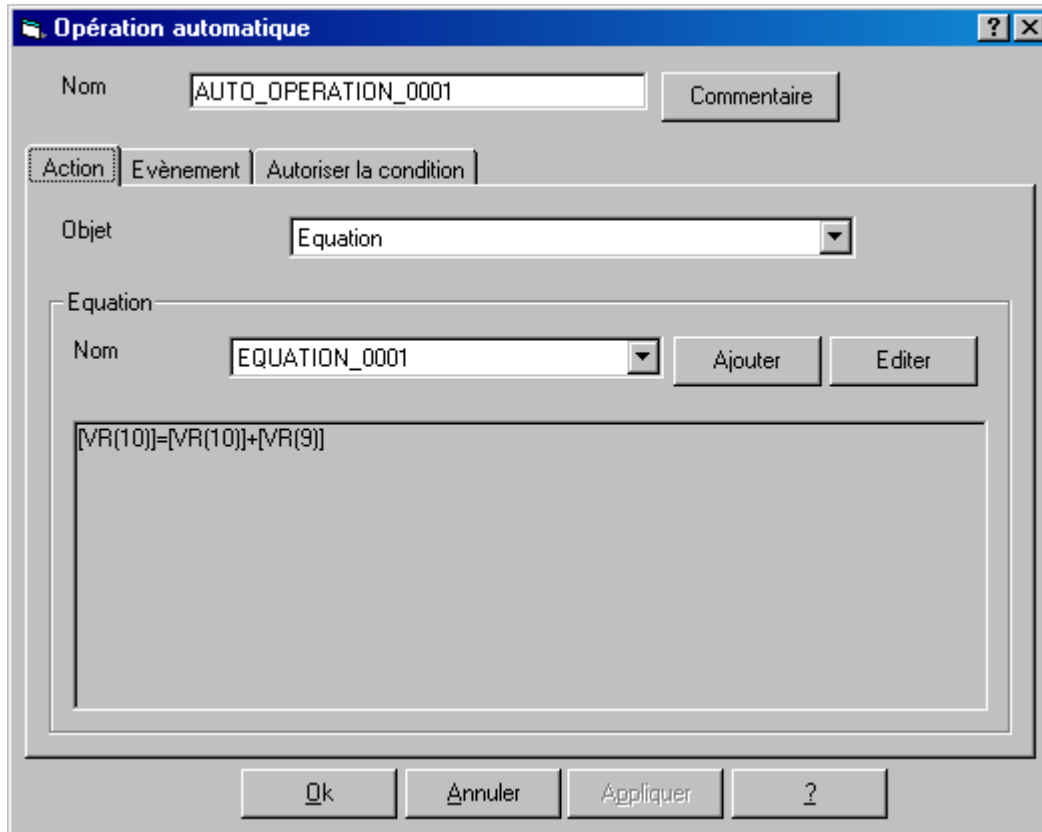
Dans **Type**, on définit le type de la source (Valeur, un bit dans un mot, ou un groupe de bits). Il faut ensuite créer les chaînes de caractères associées avec un nouveau composant '**Liste de texte**' en cliquant sur **ajouter**



Voir le chapitre '**Zones d'entrée**' pour une fonction similaire mais plus complète

3.6 OPERATIONS AUTOMATIQUES

Cette possibilité permet de réaliser des fonctions de manières automatiques. On crée un nouveau composant 'Opérations automatiques' en cliquant dans la partie 'Composant':



Dans l'onglet '**Autoriser la condition**', on peut préciser quel événement peut conditionner l'opération automatique. Ainsi, l'opération ne sera effective que si une variable Modbus est égale, entre, inférieure ou égale, supérieure ou égale à des constantes.

Dans l'onglet '**Evènement**', on précise quel événement va démarrer l'opération automatique. Il peut s'agir d'une variable Modbus égale à une constante ou entre des constantes, à pression de la touche RETOUR comme fin de définition d'une donnée ou une temporisation interne au VT50.

Dans le premier cas, l'opération est exécutée sur le front montant de l'information **vraie**. Dans le dernier cas, configurer la temporisation dans le menu '**configuration\temporisateur**'.

Dans l'onglet '**Action**', on précise la fonction réalisée par l'opération automatique si la condition d'autorisation est vraie et si l'évènement a eu lieu. Il peut s'agir d'une fonction (comme on peut les associer aux touches fonctions) ou une équation sur des variables Modbus.

Exemple : Incrémenter la variable VR(10) de TRIO toutes les secondes.

Dans le menu '**configuration\temporisateur**', sélectionner la temporisation Timer1 et indiquer sa **valeur** : 1s

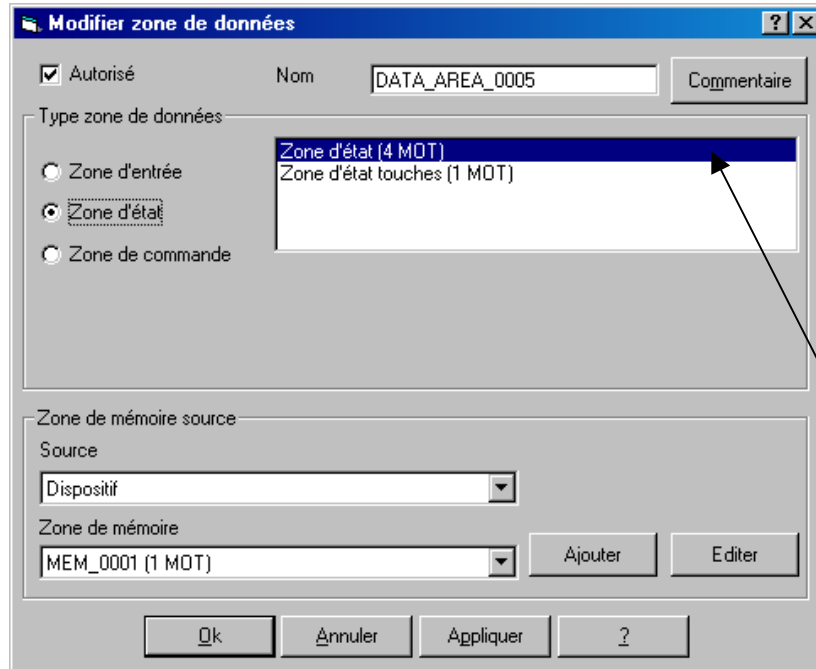
Rien à préciser dans l'onglet '**Autoriser la condition**'

Dans l'onglet '**évènement**', sélectionner temporisateur

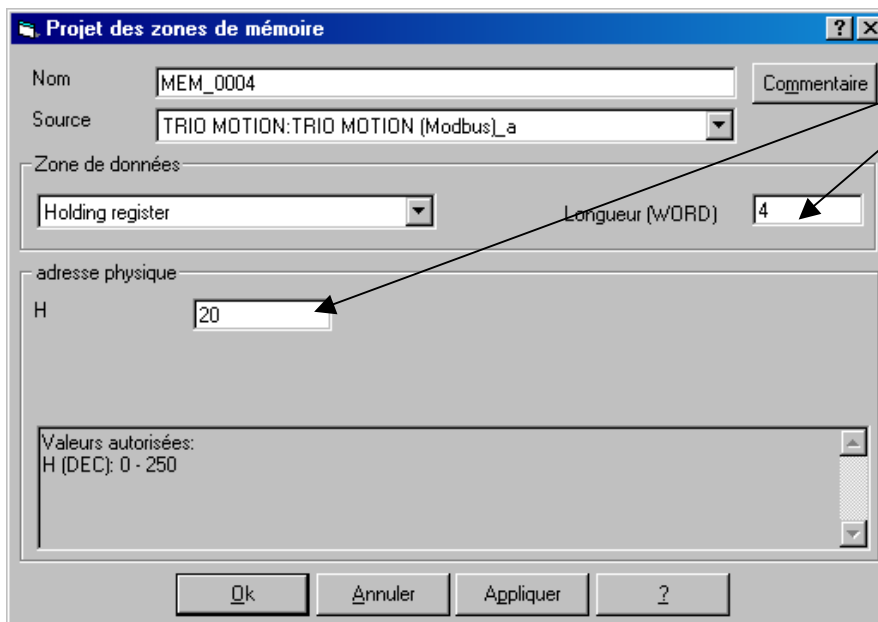
Dans l'onglet '**Action**', préciser **Equation** et **Ajouter** une équation $VR(10)=VR(10)+VR(9)$. Dans TRIO, fixer $VR(9)=1$. Toutes les secondes, VR(10) sera incrémenté par le VT

3.7 ZONES D'ECHANGES

Les zones d'échanges permettent de configurer des variables qui seront en communication constante avec TRIO. Ces variables auront de plus des actions plus profondes quant au fonctionnement du VT. Double cliquer sur '**Zone d'échange**' dans la partie '**Composants**'.



Les zones d'échanges peuvent être des zones d'entrée, zones d'état, zones de commande. Pour chaque type de zone, on définit une ou plusieurs variables permettant de communiquer les informations. Dans le cas où il en faut plusieurs, il s'agit d'un nouveau composant '**zones de mémoires**' éditables de la même manière qu'une variable et où l'on peut préciser le nombre de VR concernés dans le champ **longueur**. Les zones ne nécessitent pas toutes la même longueur de zone de mémoire. Voir dans la sélection de la zone la longueur nécessaire.



On définit une zone de mémoire de 4 variables VR(10)→ VR(24)

3.7.a LES ZONES D'ENTREES

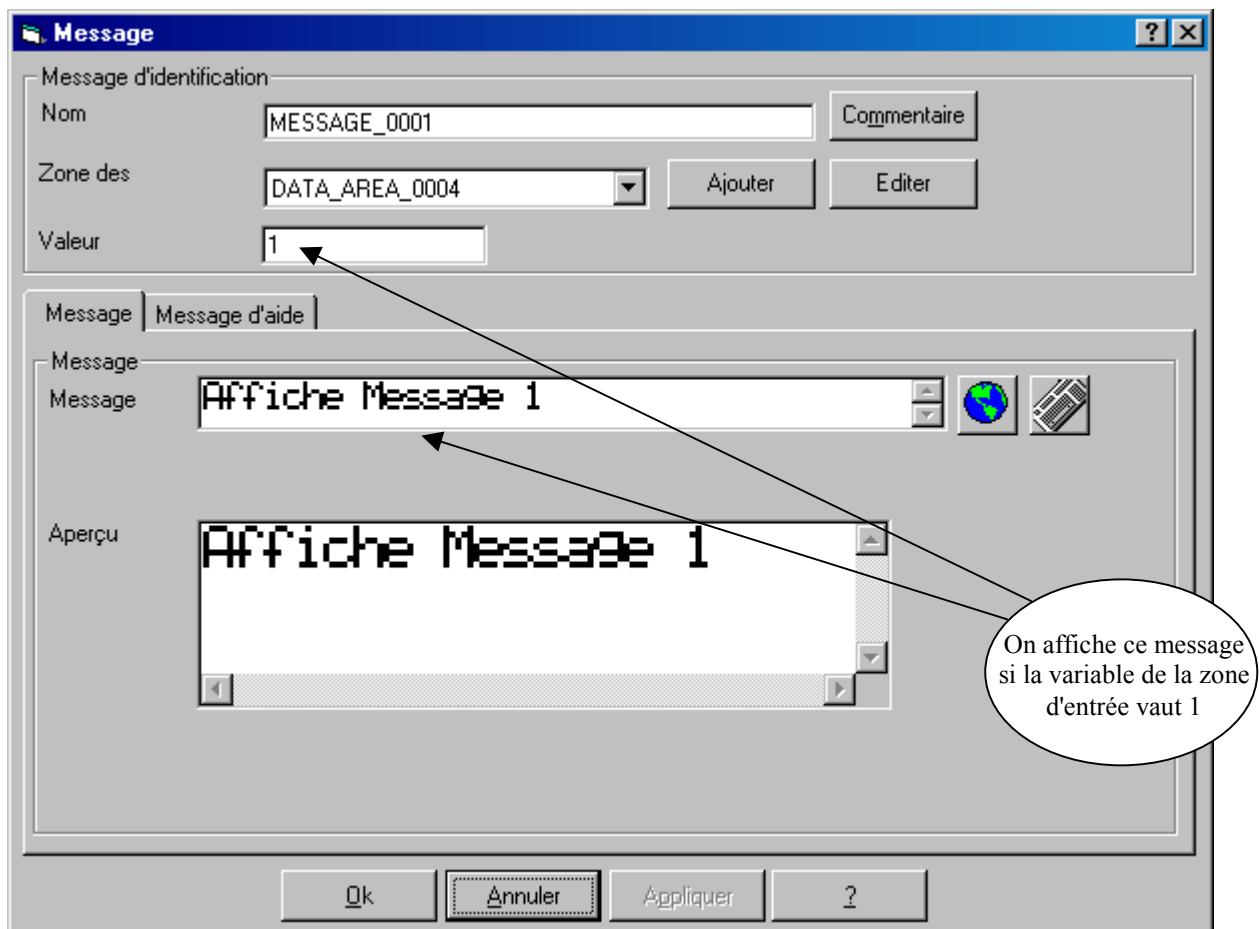
D'abord, les zones d'entrées permettent de visualiser des messages sur le pupitre. Les messages sont associés à une zone de mémoire (Max 8mots) sous forme de bits ou à une variable. La gestion du message en bit permet de visualiser plusieurs messages en même temps (si plusieurs bits sont à 1) et de les faire défiler. Pour autoriser ce défilement, sélectionner '**défilement automatique**' dans le menu '**configuration\propriétés du projet\messages**'

Sélectionner **zone d'entrées** puis '**zone de message (1...8mots)**' pour la gestion en bits ou '**zone de message à valeur**' pour la gestion avec une variable.

Ensuite, les zones d'entrées affichent les messages dans un page spécifique et on peut faire automatiquement le saut dans cette page si un message est à afficher (sinon, on la visualise avec la touche '**info**' du pupitre). C'est donc une fonction très pratique pour afficher des défauts machine. Pour autoriser le saut de page automatique, sélectionner '**changement automatique du contexte**' dans le menu '**configuration\propriétés du projet\messages**'.

Enfin, en plus de messages conditionnés, on peut associer une page d'aide pour chaque message accessible avec la touche '**aide**' du pupitre.

Pour la création des messages, créer autant de composants 'Message d'information' que de messages différents et y associer un bit ou une valeur. Double cliquer sur '**Messages d'information**' dans la partie '**Composants**'.



3.7.b LES ZONES D'ETAT

Les zones d'état permettent de lire des informations du pupitre. Sélectionner **zone d'état** puis '**zone d'état(4mots)**' pour la lecture de l'état du pupitre ou '**zone d'état des touches**' pour visualiser quelle touche du pupitre est enfoncée.

Etat du pupitre :

Cette zone d'état doit être associée à une zone mémoire de 4 variables. Ces variables seront donc régulièrement mises à jour par le VT pour communiquer, entre autre, le numéro de page affichée.

	1e Variable	2e Variable	3e Variable	4e Variable
Bit 0	Watchdog : toujours à 1	Contient une valeur différente de 0 si l'on est dans une page du projet et 0 si le pupitre est dans un autre contexte (voir 3e variable)	Contient le numéro de la page affichée ou une valeur d'identification si on est dans un autre contexte (2e variable =0) : Message d'information : 0 Répertoire des séquences : 1 Pages de service driver : 2 Informations sur le projet : 3	Position du curseur Contient une valeur qui identifie le champ où se trouve le curseur et 0 si on n'est pas dans une page du projet
Bit 1	A 1 quand un champ est en train d'être modifié			
Bit 2	A 1 quand il existe au moins 1 message à afficher			
Bit 3	A 1 quand il existe au moins 1 message arrivé et pas encore consulté			
Bit 4				
Bit 5				
Bit 6	A 1 quand la dernière commande envoyée par le périphérique n'a pas été exécutée			
Bit 7				

Etat des touches :

Associées à une zone mémoire d'un mot, cette zone d'échange permet de connaître la dernière touche enfoncée.

Bit 0	Témoin	Bit 8	F1
Bit 1	ESC	Bit 9	F2
Bit 2	Flèche gauche	Bit 10	F3
Bit 3	Flèche droite	Bit 11	F4
Bit 4	Editer	Bit 12	F5
Bit 5	Retour	Bit 13	HELP
Bit 6	Flèche bas	Bit 14	INFORMATIONS
Bit 7	Flèche haut	Bit 15	Non utilisé

A chaque pression sur une touche, le témoin est mis à 1 ainsi que le bit représentant la touche. Le dispositif peut ensuite mettre le témoin à 0 pour vérifier qu'elle touche a été enfoncée. Dès que le témoin repasse à 1, c'est qu'une touche vient d'être pressée.

3.7.c LES ZONES DE COMMANDES

Cette zone permet de commander le fonctionnement du VT par l'intermédiaire d'une zone de mémoire de 4 mots. Le premier mot précise le type de commande à passer, les 3 autres permettent de passer des arguments à la commande. Les commandes permettent entre autre de demander un changement de page :

Fonction	1e Variable	2e Variable	3e Variable	4e Variable
Force séquence de page	1	N°Séquence	N° Page	N° champ et Masque de protection à bit
Force page	2	N°Page	N° champ et Masque de protection à bit	
Force champ	3	N° champ et Masque de protection à bit		
Masque de protection en bit	4	Masque de protection à bit		
Force contexte	5	N° Contexte (Voir 3e variable dans Etat pupitre)		
Autorise le défilement des messages	6	0 : NON Autre: OUI		
Force langage	7	N° langage		
Autorise la priorité au message (saut automatique dans la page)	8	0 : NON Autre: OUI		
Fonctionnement des messages	9	Bit 0 : Défilement Bit 1 : Priorité		
Temps de défilement des messages	10	0 à 60 sec		

Par exemple, si la zone de mémoire de 4 mots à partir de VR(10) a été définie pour une zone de commande, mettre VR(11)=5 et VR(10)=2 pour forcer l'affichage de la page 5

4 TRANSFERTS DU PROJET DANS LE PUPITRE

VTWIN permet de créer son application sans pupitre (offline). Afin d'effectuer les modifications sur le pupitre, il faut y transférer le programme. Seul le programme compilé est transféré dans le pupitre.

Suivre la procédure suivante :

- Dans le programme **éditeur**, sélectionner '**outils\compiler projet**'
- Dans la fenêtre '**Project Editor**' qui s'ouvre, répondre '**OUI**' pour sauvegarder le projet.
- Dans la fenêtre '**Compilateur de projet**', cliquer sur '**compiler**'
- La compilation s'effectue et s'il n'y a ni erreur ni avertissement, on peut continuer. Sinon, **quitter** la fenêtre et corriger les erreurs puis recommencer la procédure au début.
- Cliquer sur '**Quitter**' pour fermer la fenêtre du compilateur
- La fenêtre du programme '**projet**' revient au premier plan. Cliquer sur le nom de votre projet dans la barre des tâches pour refaire apparaître le programme '**édition**'.
- Sélectionner '**outils\transférer projet**'
- La fenêtre '**Downloader**' s'affiche. Sélectionner votre port de communication et la vitesse de transmission (par défaut COM1, 115 200 bd). La case '**Mise à jour terminal**' doit être cochée pour transférer votre projet. La case '**Mise à jour firmware du terminal**' permet de transférer les caractéristiques de communication du VT50 en fonction du dispositif qui sera connecté. Il faut la cocher au premier des transferts pour configurer le pupitre. Un fois le firmware chargé, le projet peut être transféré seul.
- Préparer le VT50. Le port RS232 qui sert à programmer le pupitre est peut-être celui utilisé pour communiquer avec votre dispositif. Il faut donc préciser au VT de l'utiliser pour recevoir le firmware : Eteindre le pupitre puis rallumer le en pressant la touche F5 quelques secondes. Dans le choix qui apparaît à l'écran du pupitre, sélectionner PC avec F2. Le pupitre est prêt à recevoir les informations du PC. Cette page s'appelle '**page de transfert**'.
- Sur le PC, dans la fenêtre '**Downloader**', sélectionner '**OK**'. Le programme 'projet' revient au premier plan et une plus petite fenêtre '**Downloader**' permet de visualiser l'avance du transfert.

Une fois le transfert fait, le pupitre réalisera un reset automatique et après son initialisation, les pages du projet seront accessibles.

5 AUTRES POSSIBILITES

5.1 SAUVEGARDE DU PROJET

Pour sauvegarder le projet complet (édition et projet) dans un fichier, sélectionner '**Fichier\enregistrer**' dans le programme **projet** VTWIN créera un fichier *.V TS. Ce fichier pourra être relu par VTWIN par la suite.

Il est également possible de sauvegarder plusieurs versions du projet. Chacune des versions sauvegardées sera sauvé dans un dossier du répertoire du projet principal. Le nom des versions est le même que celui du projet auquel est ajouté REVxx ou xx est le numéro de la version (1 pour la plus récente). De plus, un fichier texte sera créé pour contenir les informations sur cette version. Cliquer sur '**Fichier\versions**' dans le programme **projet**. Vous pouvez paramétrer le nombre maximum de versions à sauvegarder, comprimer ou pas les sauvegardes (~4 fois moins gros), sauvegarder automatiquement à la fermeture de VTWIN, sauvegarder ponctuellement et ouvrir ou effacer les anciennes versions.

5.2 RECUPERATION DU PROJET PRESENT DANS LE PUPITRE

Seul le programme compilé est transmis au pupitre. Ainsi, il n'est pas possible de récupérer un programme éditable (et donc modifiable) depuis un pupitre déjà programmé. Cependant, on peut récupérer le programme compilé et le transférer dans un nouveau pupitre pour en faire une copie.

Dans le programme '**édition**', cliquer sur '**outils\sauvegarder/restaurer**'. Cliquer sur **sauvegarder** ou **restaurer** selon que vous voulez respectivement récupérer le fichier d'un pupitre ou l'envoyer dans le pupitre. Sélectionner un nom (extension *.bin) puis la page **Downloader** apparaît. Placer alors le pupitre en page de transfert (F5 à la mise sous tension puis F2) et cliquer sur OK.

5.3 MISE A JOUR DU PROJET SANS VTWIN

Afin de permettre de transférer un projet sans VTWIN, il est possible de créer un logiciel indépendant contenant le projet compilé et une fonction de communication.

Cliquer sur '**outils\ Créer disque de mise à jour terminal opérateur**' dans le programme **projet**. Si vous avez déjà utilisé cette fonction, vous pouvez sauvegarder la procédure et la rappeler dans la première fenêtre. Sinon, il faut en créer une en précisant le type de pupitre, le port de communication et la vitesse de transfert, **Télécharger firmware et projet** et préciser le nom d'un fichier *.bin préalablement créé avec la fonction du chapitre précédent, et le type de fichier à créer (sur disquette, sur disque dur avec un seul fichier ou plusieurs fichiers de la taille d'un disquette).

VTWIN créera alors un répertoire contenant entre autre le fichier *.bin et un programme **Setup.exe**. Ce dernier programme permettra de transférer le fichier *.bin à la manière du '**downloader**'.

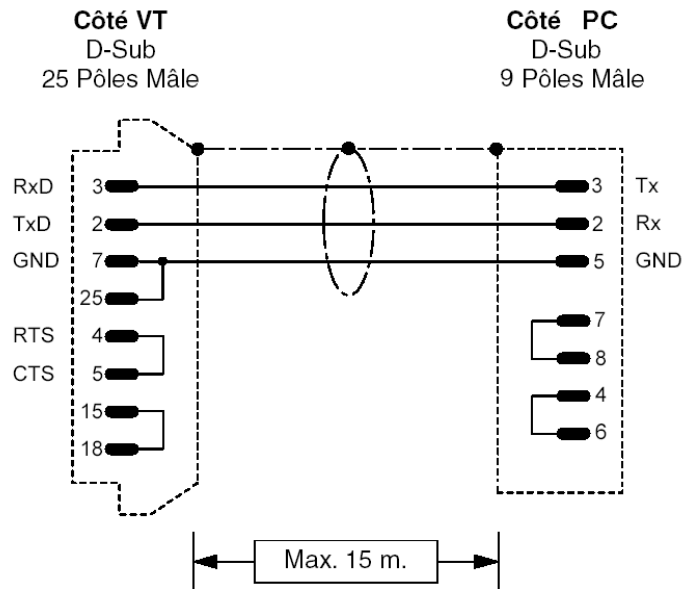
Il suffira donc de double cliquer sur setup.exe et de suivre les instructions pour transférer un projet depuis un PC ne possédant pas VTWIN

5.4 IMPRESSION DU PROJET

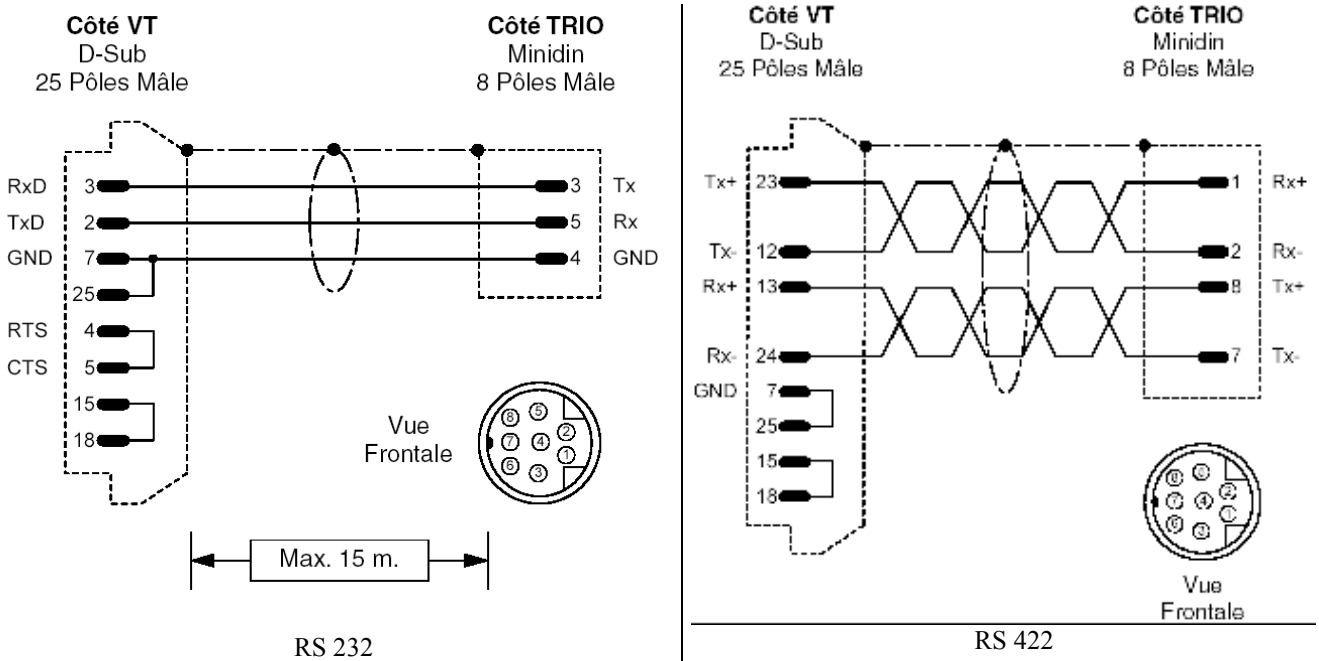
Afin de créer une documentation complète pour votre application, il est possible de paramétrer un format d'impression. Chaque paramétrage pourra être sauvé dans un modèle et réutilisé par la suite. Cliquer sur '**outils\imprimer**' dans le programme **projet**. Un combo Box vous permettra de préciser sur quel modèle vous voulez imprimer un projet. Pour créer un nouveau modèle, cliquer sur '**nouveau**' et sélectionner quels composants doivent apparaître à l'impression, créer une couverture, un index, une mise en page...

6 CABLAGE

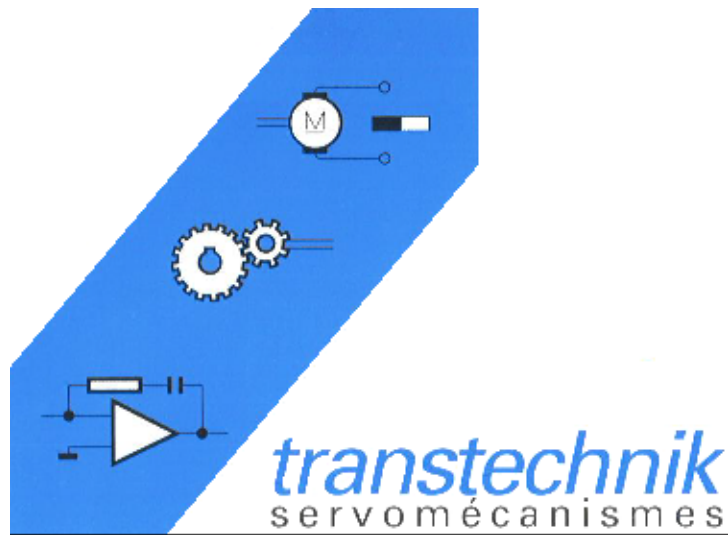
6.1 CONNEXION VT – PC



6.2 CONNEXION VT – TRIO



Par vue frontale, on entend vu par la TRIO (coté pins face à nous)



B.P.46 - Z.A. Ahuy Suzon
17rue des Grandes Varennes
21121 AHUY
Tel : 03 80 55 00 00
Fax : 03 80 53 93 63
E.Mail : infos@transtechnik.fr