

Manuel de conception des scripts V1.2.x

Sommaire

<i>I) L'architecture du fichier de script</i>	<i>4</i>
1-1) G�n�ralit�s d'�criture	4
1-2) Architecture g�n�rale.....	4
<i>II) Le champ <CONNECTION></i>	<i>5</i>
2-1) La zone de configuration	5
2-2) Le champ <COMMAND>	6
<i>III) Le champ <MENU=....>.....</i>	<i>8</i>
3-1) La fen�tre de type CONFIRM	9
3-1-1) La zone de configuration	9
3-1-2) Le champ <KWP>	10
3-2) La fen�tre de type ONEVALUE	10
3-2-1) La zone de configuration	10
3-2-2) Le champ <KWP>	11
3-3) La fen�tre de type MULTIVALUE.....	12
3-4-1) La zone de configuration	13
3-4-2) Le champ <KWP>	13
3-4) La fen�tre de type GRID.....	15
3-4-1) La zone de configuration.....	15
3-4-2) Le champ <CLICK>.....	16
3-4-3) Le champ <KWP>	26
<i>IV) Le champ <EXMENU=>.....</i>	<i>29</i>

Introduction

Le logiciel DiagKWP utilise pour diagnostiquer un fichier de configuration. Ce fichier permet de configurer les menus apparents mais aussi les fenêtres qui peuvent apparaître.

Ce document vous explique en détail comment rédiger un script de diagnostique valide pour le programme.

I) L'architecture du fichier de script

1-1) Généralités d'écriture

Les fichiers de script doivent impérativement être placés dans le répertoire « vehicule » du logiciel. Il a pour extension « .veh ».

Il a une architecture très proche du XML avec des balises qui ouvre un champ et d'autre qui les referment.

Vous pouvez sauter des lignes, ajouter des commentaires par les doubles //.

Exemple :

```
// all this command must send at the begining of the connection
<COMMAND>
REPEAT=0;
<KWP>
CDE=12 01 00; // on cherche le nombre d'IOU
TR=[1]/VAR=1; // on place la valeur dans la variable numéro 1
</KWP>
</COMMAND>
```

Dans cet exemple, nous voyons que nous commençons un champ COMMAND, à l'intérieur, il y a une instruction et il y a un champ KWP avec d'autres instructions. Le premier champ KWP est fermé par </KWP> et le champ COMMAND doit être fermé par le champ </COMMAND>

Les // permettent de placer derrière des commentaires ou en première ligne.

Toutes les lignes d'instruction doivent être finies par un « ; ». C'est impératif. A la fin des champs, il ne faut pas mettre de point virgule.

1-2) Architecture générale

Tous les scripts doivent contenir :

- Un champ <CONNECTION> :
Ce champ donne les opérations à effectuer pour se connecter et les actions à effectuer après la réussite de la connexion mais avant de donner la main à l'utilisateur.
- Un ou plusieurs champs <MENU> ou <EXMENU>
Ce sont les champs qui donnent les instructions à exécuter lorsque l'on clique sur le menu. Ils définissent les actions mais aussi les différentes traductions des retours du calculateur.

II) Le champ <CONNECTION>

Le champ <CONNECTION> est défini par une zone de configuration et un ou plusieurs champs de commandes

Exemple :

```
<CONNECTION>
// Début de la zone de configuration
INIT=5Baud;
ECU_ID=E6;
TOOL_ID=F0;
SPEED=10400; //
// Fin de la zone de configuration
<COMMAND>
REPEAT=0;
<KWP>
CDE=12 01 00;
TR=[1]/VAR=1;
</KWP>
</COMMAND>

<COMMAND>
REPEAT=0;
<KWP>
CDE=12 07 00 01;
TR=[4-19]/TABLE=1;
</KWP>
</COMMAND>
<\CONNECTION>
```

2-1) La zone de configuration

La zone de configuration permet de régler les paramètres indispensables d'une connexion ISO 9141 ou 14230.

Elle contient les instructions suivantes :

- INIT=5Baud;

Cette instruction permet de configurer le mode de connexion à utiliser pour se connecter au calculateur. Il peut prendre pour valeur :

- 5Baud : Le logiciel va effectuer une connexion à l'aide du code à 5 baud défini dans la norme ISO 9141
- FAST : Le logiciel va se connecter en respectant la procédure de connexion rapide.

- ECU_ID=E6;
Cette instruction permet de fixer l'adresse du calculateur à diagnostiquer. La valeur est écrite en hexadécimal. Elle est donnée par la documentation constructeur.
- TOOL_ID=F0;
Cette instruction permet de fixer l'adresse du logiciel de diagnostique. La valeur est écrite en hexadécimal. Elle est donnée par la documentation constructeur.
- SPEED=10400;
Cette instruction permet de fixer la vitesse à adopter après une connexion réussie. La valeur est donnée en bauds. La valeur de 10400 baud est la vitesse normale donnée par la norme. Elle n'est donc pas à changer

2-2) Le champ <COMMAND>

Ce champ permet d'exécuter une ou plusieurs commandes dans le but de mémoriser des informations qui pourront servir pour l'affichage ou dans la condition de répétition de menu du champ <EXMENU=...>

Il est formé comme dans l'exemple suivant :

<COMMAND>

REPEAT=0;

COND=[3]/=1; // on répète l'opération tant que l'octet n°3 est = à 1

<KWP>

CDE=12 07 81; // on récupère les noms des entrées sortie

TR=[4-19]/TABLE=1; // on place la valeur dans la TABLE numéro 1

</KWP>

</COMMAND>

- **REPEAT=0 ;**
Cette instruction permet de configurer la période de **répétition inconditionnelle** des commandes des champs <KWP>. Dans ce cas, la valeur est nulle de manière à n'exécuter la/ les commandes qu'une seule fois. Sinon, la valeur est indiquée en ms. Ce délai représente le délai d'attente entre la dernière commande et la première à recommencer.
- **COND=[3]/=1**
Cette instruction indique que la (les) commande KWP doivent être exécuter tant que la valeur entre crochet est égale à 1.

- **KWP**

Le champ **KWP** contient la commande à exécuter et la (les) façon de traduire les valeurs renvoyées par le calculateur. Le champ **<KWP>** contenu dans un champ **<COMMAND>** ne fait JAMAIS de référence à une valeur à afficher. Il est différent en ce sens d'un champ **<KWP>** contenu dans les champs **<MENU=....>**

Le champ **<KWP>** contient les informations suivantes :

- **CDE=12 07 81 ;**

Cette instruction est la commande à envoyer au calculateur. Les valeurs sont obligatoirement séparées par des espaces. Toutes les valeurs sont écrites en hexadécimal.

- **TR=[4-19]/TABLE=1 ;**

L'instruction TR= est la ligne qui explique au logiciel que faire des données renvoyées par le calculateur. Dans le champ KWP d'un champ COMMAND, les seules actions possibles sont de mémoriser les données dans une table ou dans une variable.

- **TABLE=**

Le logiciel possède 9 tables de texte numérotées de 1 à 9. Ces tables permettent de mémoriser une liste de texte que peut envoyer un calculateur. Dans l'exemple, le calculateur renvoie le nom de ses entrées et de ses sorties. A chaque fois que la commande est envoyée, le calculateur répond par un nom d'une entrée qui est mémorisé dans la table n°1 les une à la suite des autres.

- **VAR=**

Le logiciel possède 9 variables entières. Elles sont là pour mémoriser une valeur

- Le caractère /

Dans les lignes d'instructions, il permet de séparer les données ou instruction. Dans cet exemple, on sépare **[4-19]** de **TABLE=**.

- **[4-19]**

Les valeurs contenu entre crochet doivent toujours être placée juste après la commande **TR=**. Il permet au programme d'interpréter la valeur renvoyée par le calculateur qui peut être contenu dans une plage d'octets ou une plage de bits. Dans cet exemple, on place dans la TABLE n°1, les octets N°4 à 19 qui représente des codes ASCII. Mais cet exemple est simple. La traduction peut être plus complexe comme celle-ci :

R5[8|5-7]L :

- **R5** défini que l'on doit répéter la traduction sur tous les octets de la trame en partant de l'octet n°5 et ceci jusqu'à la fin de la trame reçu.
- **[8|5-7]** défini une plage de bit dans un octet. Donc on récupère la valeur contenu dans l'octet 8, sur la plage de bit de 5 à 7.
- **L** indique que les octets sont écrits en mode Octet de poids faible puis octet de poids fort. L'inverse est le mode par défaut, on n'indique rien, sinon, on le précise par **H**.











III) Le champ <MENU=....>

Le champ menu permet d'indiquer au programme la ou les actions à exécuter par le logiciel. Ici, tout est défini pour l'affichage au contraire des ligne de traduction de <COMMAND>.

Exemple :

```
<MENU=Lecture-écriture de liste de parametres>
ICO=8;
TYPE=MULTIVALUE;
TITLE=Lecture d'une liste de parametres EEPROM;
REPEAT=0;
LABEL_READ=Lecture;
LABEL_WRITE=Ecriture;
<KWP>
CDE_READ=L21 40;
CDE_WRITE=L3B 40;
TR=[1-64]L/TX=VALUE;
</KWP>
</MENU>
```

Tous les champs <MENU=..> contiennent toujours les informations suivantes :

- ICO=.. ;
Cette instruction permet de définir l'icône associée au menu. Ils sont au nombre de 10 numérotés de 0 à 9.
L'icône  a pour numéro 0
L'icône  a pour numéro 1
L'icône  a pour numéro 2
L'icône  a pour numéro 3
L'icône  a pour numéro 4
L'icône  a pour numéro 5
L'icône  a pour numéro 6
L'icône  a pour numéro 7
L'icône  a pour numéro 8
L'icône  a pour numéro 9
- TYPE=... ;
Cette instruction permet de choisir le type d'affichage à utiliser. Il existe 4 types d'affichages différents :
 - L'affichage CONFIRM
 - L'affichage ONEVALUE
 - L'affichage MULTIVALUE
 - L'affichage GRID

Les types d'affichage sont expliqués plus loin.

- **TITLE=...** ;
Cette instruction initialise le nom de la fenêtre d’affichage ou des colonnes pour les affichages à base de tableau.
- **REPEAT=0** ;
Cette instruction permet de configurer la période de **répétition inconditionnelle** des commandes des champs **<KWP>**. Dans ce cas, la valeur est nulle de manière à n’exécuter la/ les commandes qu’une seule fois. Sinon, la valeur est indiquée en ms. Ce délai représente le délai d’attente entre la dernière commande et la première à recommencer.

Comme les champs contenus sont nombreux et dépendent du type d’affichage, ils sont expliqués dans différents paragraphes.

3-1) La fenêtre de type CONFIRM

Ce type d’affichage permet d’effectuer une opération en demandant la confirmation avant de le faire à l’utilisateur.

Exemple :

```
<MENU=Effacement DTC>
ICO=3;
TYPE=CONFIRM;
TITLE=Clear diagnostic trouble code;
REPEAT=0;
<KWP>
CDE=14;
TR=;
</KWP>
</MENU>
```

3-1-1) La zone de configuration

La zone de configuration possède uniquement les instructions suivantes :

- **ICO=3;**
L’icône du menu
- **TYPE=CONFIRM;**
Le type d’affichage CONFIRM
- **TITLE=Clear diagnostic trouble code;**
Le titre de la fenêtre qui apparaît à l’écran
- **REPEAT=0 ;**
Le délai de répétition. Dans le cas d’une fenêtre CONFIRM, il doit toujours être nul sinon, le logiciel enverra plusieurs fois la même commande. Ce qui n’est pas nécessairement utile.

3-1-2) Le champ <KWP>

Le champ KWP contient deux lignes différentes :

- **CDE=14 ;**
Cette instruction est la commande à envoyer au calculateur si l'utilisateur accepte la question qui est « Etes vous sur de faire ceci ? ». Si l'utilisateur clique sur oui, c'est cette commande qui sera envoyée. Les valeurs sont obligatoirement séparées par des espaces. Toutes les valeurs sont écrites en hexadécimal.
- **TR= ;**
Dans le cas d'un champ CONFIRM, l'affichage ne tiens pas compte des octets transmis. Donc on n'indique aucunes règles de traduction.

3-2) La fenêtre de type ONEVALUE

Ce type d'affichage permet de lire ou/et écrire une valeur du calculateur en fonction d'un numéro. Ce type d'affichage est adapté à l'écriture / écriture d'un paramètre du calculateur.

```
<MENU=Lecture-écriture de parametre>
ICO=8;
TYPE=ONEVALUE;
TITLE=Lecture d'un paramètre EEPROM;
REPEAT=0;
LABEL_READ=Lecture;
LABEL_WRITE=Ecriture;
<KWP>
CDE_READ=L21 40;
CDE_WRITE=L3B 40;
TR=[1-64]L/TX=VALUE;
</KWP>
</MENU>
```

3-2-1) La zone de configuration

- **ICO=8;**
L'icône du menu
- **TYPE=ONEVALUE;**
Le type d'affichage ONEVALUE
- **TITLE=Lecture d'un paramètre EEPROM;**
Le titre de la fenêtre

- **REPEAT=0;**
Cette instruction permet de configurer la période de **répétition inconditionnelle** des commandes des champs **<KWP>**. Dans ce cas, la valeur est nulle de manière à n'exécuter la/ les commandes qu'une seule fois. Sinon, la valeur est indiquée en ms. Ce délai représente le délai d'attente entre la dernière commande et la première à recommencer.
- **LABEL_READ=Lecture;**
Cette instruction initialise le nom du bouton associé à la commande CDE_READ
- **LABEL_WRITE=Ecriture;**
Cette instruction initialise le nom du bouton associé à la commande CDE_WRITE

3-2-2) Le champ <KWP>

- **CDE_READ=L21 40;**
Cette instruction définit la commande à envoyer au calculateur lors de l'appui sur la touche ayant le label LABEL_READ. En général c'est une demande de lecture de paramètre. La particularité est qu'il faut mettre un L ou un H devant la liste des commande car le logiciel va envoyer au calculateur la CDE_READ avec le numéro de paramètre puis la valeur. Le L permet de savoir comment classer les octets lorsque la valeur du paramètre est une valeur contenu sur plus de 8 bits. Le L représente l'octet de poids faible après l'octet de poids fort. Par exemple, si le paramètre est le numéro 5, la valeur en hexadécimal à transmettre est ABCD. Le logiciel avec la commande L21 40 va envoyer au calculateur 21 40 05 CD AB. Si la commande avait été H21 40 avec les mêmes conditions, le logiciel enverra 21 40 05 AB CD
- **CDE_WRITE=L3B 40;**
Cette instruction définit la commande à envoyer au calculateur lors de l'appui sur la touche ayant le label LABEL_WRITE. En général c'est une demande de lecture de paramètre. La particularité est qu'il faut mettre un L ou un H devant la liste des commande car le logiciel va envoyer au calculateur la CDE_WRITE avec le numéro de paramètre puis la valeur. Le L permet de savoir comment classer les octets lorsque la valeur du paramètre est une valeur contenu sur plus de 8 bits. Le L représente l'octet de poids faible après l'octet de poids fort. Par exemple, si le paramètre est le numéro 5, la valeur en hexadécimal à transmettre est ABCD. Le logiciel avec la commande L3B 40 va envoyer au calculateur 3B 40 05 CD AB. Si la commande avait été H3B 40 avec les mêmes conditions, le logiciel enverra 3B 40 05 AB CD

- **TR=[1-64]L/TX=VALUE;**

L'instruction TR= est la ligne qui explique au logiciel que faire des données renvoyées par le calculateur.

- **TX=**

La valeur reçue sera placée dans la zone d'édition des données. Le terme VALUE indique que l'on désire un affichage en décimal. VAL_HEX pour un affichage hexadécimal ainsi que VAL_TXT sont aussi possible.

- Le caractère /

Dans les lignes d'instructions, il permet de séparer les données ou instruction. Dans cet exemple, on sépare **[1-64]L** de **TX=**.

- **[1-64]L**

Les valeurs contenu entre crochet doivent toujours être placée juste après la commande **TR=**. Il permet au programme d'interpréter la valeur renvoyée par le calculateur qui peut être contenu dans une plage d'octets ou une plage de bits. Dans cet exemple, on place dans une valeur entière, les octets N°1 à 64 en valeurs entières.

Mais cet exemple est simple. La traduction peut être plus complexe comme celle-ci :

R5[8|5-7]L :

- **R5** défini que l'on doit répéter la traduction sur tous les octets de la trame en partant de l'octet n°5 et ceci jusqu'à la fin de la trame reçu.
- **[8|5-7]** défini une plage de bit dans un octet. Donc on récupère la valeur contenu dans l'octet 8, sur la plage de bit de 5 à 7.
- **L** indique que les octets sont écrits en mode Octet de poids faible puis octet de poids fort. L'inverse est le mode par défaut, on n'indique rien, sinon, on le précise par **H**.

- Autre possibilité dans la ligne de traduction : **V=4|20 :**

Il faut placer cette instruction juste après l'instruction de décodage. Elle permet la mise à l'échelle de la valeur traduite précédemment. Nous allons donc transmettre pour l'affichage Valeur=4* Valeur_traduite+20.

3-3) La fenêtre de type MULTIVALUE

Ce type d'affichage permet de lire ou/et écrire une liste de valeurs du calculateur en fonction d'un fichier de script. Ce type d'affichage est adapté à l'écriture / écriture d'une liste de paramètres du calculateur.

Exemple :

```
<MENU=Lecture-écriture de liste de parametres>
ICO=8;
TYPE=MULTIVALUE;
TITLE=Lecture d'une liste de parametres EEPROM;
REPEAT=0;
LABEL_READ=Lecture;
```

```
LABEL_WRITE=Ecriture;  
<KWP>  
CDE_READ=L21 40;  
CDE_WRITE=L3B 40;  
TR=[1-64]L/TX=VALUE;  
</KWP>  
</MENU>
```

3-4-1) La zone de configuration

- **ICO=8;**
L'icône du menu
- **TYPE=MULTIVALUE;**
Le type d'affichage MULTIVALUE
- **TITLE=Lecture d'une liste de paramètres EEPROM;**
Le titre de la fenêtre
- **REPEAT=0;**
Cette instruction permet de configurer la période de **répétition inconditionnelle** des commandes des champs **<KWP>**. Dans ce cas, la valeur est nulle de manière à n'exécuter la/ les commandes qu'une seule fois. Sinon, la valeur est indiquée en ms. Ce délai représente le délai d'attente entre la dernière commande et la première à recommencer.
- **LABEL_READ=Lecture;**
Cette instruction initialise le nom du bouton associé à la commande CDE_READ
- **LABEL_WRITE=Ecriture;**
Cette instruction initialise le nom du bouton associé à la commande CDE_WRITE

Lors d'un affichage en Multivalue, la/les instruction KWP sont répétée jusqu'à ce que la fin du tableau soit atteinte.

3-4-2) Le champ <KWP>

- **CDE_READ=L21 40;**
Cette instruction définit la commande à envoyer au calculateur lors de l'appui sur la touche ayant le label LABEL_READ. En général c'est une demande de lecture de paramètre. La particularité est qu'il faut mettre un L ou un H devant la liste des commandes car le logiciel va envoyer au calculateur la CDE_READ avec le numéro de paramètre puis la valeur. Le L permet de savoir comment classer les octets lorsque la valeur du paramètre est une valeur contenue sur plus de 8 bits. Le L représente l'octet de poids faible après l'octet de poids fort. Par exemple, si le paramètre est le numéro 5, la valeur en hexadécimal à transmettre est ABCD. Le logiciel avec la commande L21 40 va envoyer au calculateur 21 40 05 CD AB. Si la commande avait été H21 40 avec les mêmes conditions, le logiciel enverra 21 40 05 AB CD

- **CDE_WRITE=L3B 40;**

Cette instruction définit la commande à envoyer au calculateur lors de l'appui sur la touche ayant le label LABEL_WRITE. En général c'est une demande de lecture de paramètre. La particularité est qu'il faut mettre un L ou un H devant la liste des commandes car le logiciel va envoyer au calculateur la CDE_WRITE avec le numéro de paramètre puis la valeur. Le L permet de savoir comment classer les octets lorsque la valeur du paramètre est une valeur contenue sur plus de 8 bits. Le L représente l'octet de poids faible après l'octet de poids fort. Par exemple, si le paramètre est le numéro 5, la valeur en hexadécimal à transmettre est ABCD. Le logiciel avec la commande L3B 40 va envoyer au calculateur 3B 40 05 CD AB. Si la commande avait été H3B 40 avec les mêmes conditions, le logiciel enverra 3B 40 05 AB CD

- **TR=[1-64]L/TX=VALUE;**

L'instruction **TR=** est la ligne qui explique au logiciel que faire des données renvoyées par le calculateur.

- **TX=**

La valeur reçue sera placée dans la zone d'édition des données. Le terme **VALUE** indique que l'on désire un affichage en décimal. **VAL_HEX** pour un affichage hexadécimal ainsi que **VAL_TXT** sont aussi possible.

- Le caractère /

Dans les lignes d'instructions, il permet de séparer les données ou instruction. Dans cet exemple, on sépare **[1-64]L** de **TX=**.

- **[1-64]L**

Les valeurs contenu entre crochet doivent toujours être placée juste après la commande **TR=**. Il permet au programme d'interpréter la valeur renvoyée par le calculateur qui peut être contenu dans une plage d'octets ou une plage de bits. Dans cet exemple, on place dans une valeur entière, les octets N°1 à 64 en valeurs entières.

Mais cet exemple est simple. La traduction peut être plus complexe comme celle-ci :

R5[8|5-7]L :

- **R5** définit que l'on doit répéter la traduction sur tous les octets de la trame en partant de l'octet n°5 et ceci jusqu'à la fin de la trame reçue.
- **[8|5-7]** définit une plage de bit dans un octet. Donc on récupère la valeur contenu dans l'octet 8, sur la plage de bit de 5 à 7.
- **L** indique que les octets sont écrits en mode Octet de poids faible puis octet de poids fort. L'inverse est le mode par défaut, on n'indique rien, sinon, on le précise par **H**.

- Autre possibilité dans la ligne de traduction : **V=4|20 :**

Il faut placer cette instruction juste après l'instruction de décodage. Elle permet la mise à l'échelle de la valeur traduite précédemment. Nous allons donc transmettre pour l'affichage $Valeur = 4 * Valeur_traduite + 20$.

ATTENTION, cette conversion sera appliqué à toutes les valeurs de paramètres.

3-4) La fenêtre de type GRID

La fenêtre de type grille permet d'afficher des valeurs, des états logiques, de modifier dynamiquement des valeurs. La définition possède trois zones différentes. Une zone de configuration, un champ <CLICK> optionnel et un ou plusieurs champs <KWP>.

3-4-1) La zone de configuration

Dans la zone de configuration, on définit l'icône du menu, le nom des colonnes, le type d'affichage, le délai d'attente entre la dernière commande KWP et l'exécution à nouveau de la première commande, les différents textes fixes du tableau.

- ICO=2;
C'est l'icône du menu.
- TITLE=Label/Pin/Référence/Etat/Defaut/Contact/Commentaire;
Ce sont les noms des différentes colonnes.
- TYPE=GRID/7;
L'affichage un type tableau (GRID) avec 7 colonnes.
- REPEAT=200;
Le délai d'attente entre la dernière commande KWP et l'exécution à nouveau de la première commande est de 200ms.
- LG=ROW=1/CL=FDC968/GR0=EREV 0/GR1=CN 4.8/GR2=TAB[1,1]/GR5=GND;
L'instruction LG définit un ensemble de texte fixe à mettre dans le tableau.
 - ROW=
Cette instruction définit le numéro de ligne de la grille sur lequel s'applique les données qui suivent. Le numéro de ligne commence par 1.
 - CL=FDC968
Cette instruction définit la couleur à appliquer à la ligne concernée. Si cette instruction n'est pas précisée, la couleur sera blanche. La valeur de la couleur doit être donnée en hexadécimal.
 - GR0=EREV 0
L'instruction GR permet de spécifier ce qu'il faut mettre dans la case de la colonne (ici 0). Comme le texte n'est pas une commande réservée (VALUE, VAL_TXT, VAL_HEX, TAB, IMG_GREEN, IMG_RED) le logiciel affiche le texte directement sans conversion d'aucun type.
 - GR1=CN 4.8
L'instruction GR permet de spécifier ce qu'il faut mettre dans la case de la colonne (ici 1). Comme le texte n'est pas une commande réservée (VALUE, VAL_TXT, VAL_HEX, TAB, IMG_GREEN, IMG_RED) le logiciel affiche le texte directement sans conversion d'aucun type.

- GR2=TAB[1,5]

L'instruction GR permet de spécifier ce qu'il faut mettre dans la case de la colonne (ici 2). Ici, on utilise l'instruction TAB. On appelle le texte contenu dans le tableau n°1 et le 6^e éléments (les tableaux de texte sont numérotés à partir de 0) ;

3-4-2) Le champ <CLICK>

Ce champ permet de définir ce que le logiciel doit faire lorsque l'on appui sur une case ou une ligne entière. En effet, il est possible d'afficher lors d'un click de souris sur une case, d'afficher une fenêtre fille qui pourra exécuter une commande pour que le calculateur change une valeur ou qu'il active une sortie...

Exemple :

<CLICK>

EXIT=32 01;

ACTIF=31 01;

CLK=,35/ONOFF/EXIT=GLOBAL/NAME=Commande logique/TITLE=OUT 0 - CN 5.20/ON=30 00 00 01 00 00/OFF=30 00 00 00 00 00;

CLK=,41/PWM/EXIT=GLOBAL/NAME=Commande dynamique en frequence/TITLE=OUT 4 - CN 5.24/DIV=1/TYPE=16L/UNIT=Hz/PAS=1/CDE=30 04 00 VALUE 00;

CLK=,65/BIPWM/EXIT=GLOBAL/NAME=Commande dynamique/TITLE=OUT 22 - CN 5.19/BAR1=Periode/UNIT1=μS/BAR2=PWM/UNIT2=%/TYPE1=16L/TYPE2=8/DIV1=6.4/DIV2=1/CDE=30 16 00 VAL1 VAL2;

CLK=,18/WIPER/EXIT=GLOBAL/NAME=Commande Essui vitre/TITLE=OUT 0 - CN 5.24 - CN5.25/STOP=30 12 01 00 00 00/SLOW=30 12 01 01 00 00/FAST=30 12 01 02 00 00;

CLK=,36/BRIDGE/EXIT=GLOBAL/NAME=Commande du Bridge n°3/TITLE=OUT 13 & 14/STOP=00/SENS1=01/SENS2=02/DIV=50/TYPE=8/UNIT=Hz/PAS=50/CDE=30 11 01 SENS VALUE;

CLK=,1/LIST/EXIT=/CASE=1,2-9/NAME=Ecriture de la date et de l'heure/TITLE=Date et heure/COR=0.25|0:1|0:1|0:1|0:1|0:1|1985:1|-125:1|-125/TYPE=8:8:8:8:8:8:8:8/CDE=2E F9 0B VALUE;

CLK=,18/VALUE/EXIT=/CASE=1,18/NAME=Ecriture du paramètre Impulsion par tour/TITLE=Parametre Impulsion par tour/DIV=0.001/TYPE=16H/CDE=2E F9 1E VALUE;

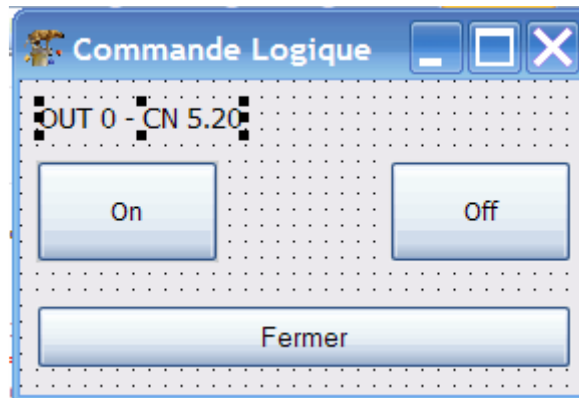
</CLICK>

Explication des instructions :

- EXIT=32 01;
L'instruction EXIT= définit une commande à envoyer au calculateur pour arrêter le forçage d'une sortie. En effet, certains calculateurs demandent d'envoyer une commande particulière pour quitter le mode de diagnostic actif. Si cette instruction est absente, le bouton Mode Forcé en bas de la grille n'apparaîtra pas.
- ACTIF=31 01;
L'instruction ACTIF= permet de demander au calculateur de passer dans le mode de diagnostic actif ou il est possible de forcer l'état des sorties.
C'est la première instruction que le logiciel enverra si on clique sur une case.

- CLK=,35/ONOFF/EXIT=GLOBAL/NAME=Commande logique/TITLE=OUT 0 - CN 5.20/ON=30 00 00 01 00 00/OFF=30 00 00 00 00 00;

Cette instruction CLK= indique quel type de fenêtre fille doit être affichée et ce qu'il faut faire. Dans ce cas, « ,35 » indique au logiciel que si on clique sur la ligne 35 (la valeur avant la virgule définit la colonne et comme elle est omise, le logiciel en conclut que c'est valable pour toute la ligne). Ensuite on indique le type de fenêtre : ici ONOFF qui ressemble à :



Ce type de fenêtre permet d'activer ou de désactiver une sortie.

L'instruction EXIT= permet d'indiquer si on doit envoyer une commande lorsque l'on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n'envoie pas de trame.

L'instruction NAME= fixe le nom de la fenêtre.

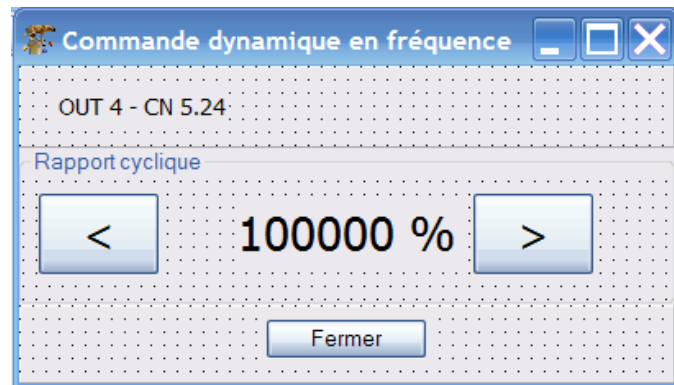
L'instruction TITLE= permet de fixer le texte au dessus des boutons.

L'instruction ON= permet de définir la commande à envoyer lorsque l'on clique sur le bouton ON.

L'instruction OFF= permet de définir la commande à envoyer lorsque l'on clique sur le bouton OFF.

- CLK=,41/PWM/EXIT=GLOBAL/NAME=Commande dynamique en fréquence/TITLE=OUT 4 - CN 5.24/DIV=1/TYPE=16L/UNIT=Hz/PAS=1/CDE=30 04 00 VALUE 00;

Cette instruction CLK= indique quel type de fenêtre fille doit être affichée et ce qu'il faut faire. Dans ce cas, « ,41 » indique au logiciel que si on clique sur la ligne 41 (la valeur avant la virgule définit la colonne et comme elle est omise, le logiciel en conclut que c'est valable pour toute la ligne). Ensuite on indique le type de fenêtre : ici PWM qui ressemble à :



Ce type de fenêtre permet d'activer une sortie qui a une activation proportionnelle. L'instruction EXIT= permet d'indiquer si on doit envoyer une commande lorsque l'on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n'envoie pas de trame.

L'instruction NAME= fixe le nom de la fenêtre.

L'instruction TITLE= permet de fixer le texte au dessus des boutons.

L'instruction DIV permet de définir un diviseur pour la mise à l'échelle de la valeur à envoyer au calculateur. Ici, le diviseur est de 1. La valeur envoyée au calculateur est donc $\text{Valeur_envoyée} = \text{Valeur_affichée} / \text{DIV}$

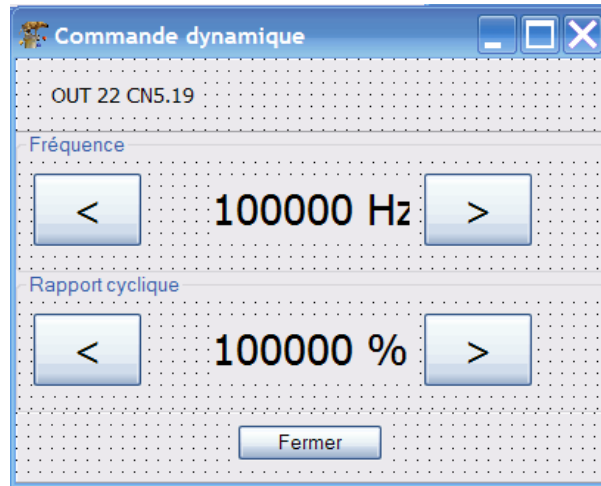
L'instruction TYPE=16L définit le nombre d'octet de la valeur à transmettre au calculateur. Ici, la valeur est de 16 bits. Le classement des octets se fait en mode L soit l'octet de poids faible avant celui de poids fort.

L'instruction UNIT= définit le texte d'unité à afficher derrière la valeur.

L'instruction PAS= définit le pas qu'il faut appliquer lors d'un appui sur un bouton. Le pas peut être un nombre réel.

L'instruction CDE= permet de définir la commande à envoyer lorsque l'on clique sur un des boutons < ou >. Il faut mettre à l'intérieur de la trame à envoyer le terme VALUE pour indiquer au logiciel ou mettre la valeur à envoyer au calculateur.

- CLK=,65/BIPWM/EXIT=GLOBAL/NAME=Commande dynamique/TITLE=OUT 22 - CN
5.19/BAR1=Periode/UNIT1=μS/BAR2=PWM/UNIT2=%/TYPE1=16L/TYPE2=8/DIV1=6.4/DIV2=1/CDE=30 16 00 VAL1 VAL2;
Cette instruction CLK= indique quel type de fenêtre fille doit être affichée et ce qu'il faut faire. Dans ce cas, « ,65 » indique au logiciel que si on clique sur la ligne 65 (la valeur avant la virgule définit la colonne et comme elle est omise, le logiciel en conclut que c'est valable pour toute la ligne). Ensuite on indique le type de fenêtre : ici BIPWM qui ressemble à :



Ce type de fenêtre permet d'activer une sortie qui a deux paramètres proportionnels modifiables

L'instruction EXIT= permet d'indiquer si on doit envoyer une commande lorsque l'on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n'envoie pas de trame.

L'instruction NAME= fixe le nom de la fenêtre.

L'instruction TITLE= permet de fixer le texte au dessus des boutons.

L'instruction BAR1= permet de fixer le nom de la zone de première zone de modification de valeur.

L'instruction UNIT1= définit le texte d'unité à afficher derrière la valeur de la zone 1.

L'instruction BAR2= permet de fixer le nom de la zone de la zone n°2 de modification de valeur.

L'instruction UNIT2= définit le texte d'unité à afficher derrière la valeur de la zone 2.

L'instruction TYPE1=16L définit le nombre d'octet de la valeur à transmettre au calculateur de la zone 1. Ici, la valeur est de 16 bits. Le classement des octets se fait en mode L soit l'octet de poids faible avant celui de poids fort.

L'instruction TYPE2=8 définit le nombre d'octet de la valeur à transmettre au calculateur de la zone 1. Ici, la valeur est de 8 bits. Il n'y a pas de définition de classement des octets.

L'instruction DIV1 permet de définir un diviseur pour la mise à l'échelle de la valeur à envoyer au calculateur pour la zone 1. Ici, le diviseur est de 6.4. La valeur envoyée au calculateur est donc $\text{Valeur_envoyée} = \text{Valeur_affichée} / 6.4$

L'instruction DIV2 permet de définir un diviseur pour la mise à l'échelle de la valeur à envoyer au calculateur pour la zone 1. Ici, le diviseur est de 1. La valeur envoyée au calculateur est donc $\text{Valeur_envoyée} = \text{Valeur_affichée} / 1$

L'instruction CDE= permet de définir la commande à envoyer lorsque l'on clique sur un des boutons < ou >. Il faut mettre à l'intérieur de la trame à envoyer le terme VAL1 et le terme VAL2 pour indiquer au logiciel où mettre les valeurs à envoyer au calculateur.

- CLK=,18/WIPER/EXIT=GLOBAL/NAME=Commande Essui vitre/TITLE=OUT 0 - CN 5.24 - CN5.25/STOP=30 12 01 00 00 00/SLOW=30 12 01 01 00 00/FAST=30 12 01 02 00 00;

Certain calculateur possède un essuie vitre piloté. La fenêtre de type WIPER est faite spécialement pour le piloter. Elle ressemble à :



L'instruction EXIT= permet d'indiquer si on doit envoyer une commande lorsque l'on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n'envoie pas de trame.

L'instruction NAME= fixe le nom de la fenêtre.

L'instruction TITLE= permet de fixer le texte au dessus des boutons.

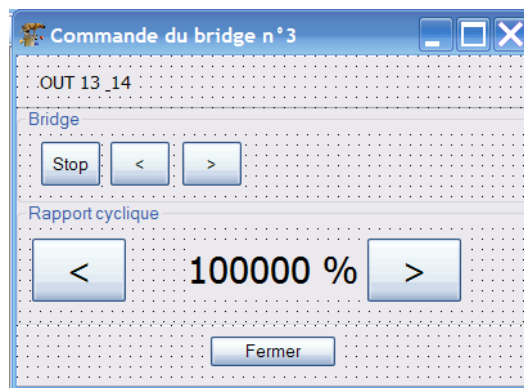
L'instruction STOP= permet de définir la commande à envoyer au calculateur lorsque l'utilisateur clique sur le bouton STOP.

L'instruction SLOW= permet de définir la commande à envoyer au calculateur lorsque l'utilisateur clique sur le bouton >.

L'instruction FAST= permet de définir la commande à envoyer au calculateur lorsque l'utilisateur clique sur le bouton >>.

- CLK=,36/BRIDGE/EXIT=GLOBAL/NAME=Commande du Bridge
n°3/TITLE=OUT 13 &
14/STOP=00/SENS1=01/SENS2=02/DIV=50/TYPE=8/UNIT=Hz/PAS=50/CDE=30
11 01 SENS VALUE;

Ce type d’affichage permet de piloter le pont en H d’un calculateur. Il est donc possible de changer le sens et de piloter la puissance délivrée.



L’instruction EXIT= permet d’indiquer si on doit envoyer une commande lorsque l’on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n’envoi pas de trame.

L’instruction NAME= fixe le nom de la fenêtre.

L’instruction TITLE= permet de fixer le texte au dessus des boutons.

L’instruction STOP= permet de définir la commande à envoyer au calculateur lorsque l’utilisateur clique sur le bouton STOP pour arrêter toute commande du bridge.

L’instruction SENS1= permet de définir la commande à envoyer au calculateur lorsque l’utilisateur clique sur le bouton < pour que la calculateur place le bridge dans le sens 1.

L’instruction SENS2= permet de définir la commande à envoyer au calculateur lorsque l’utilisateur clique sur le bouton > pour que la calculateur place le bridge dans le sens 2.

L’instruction DIV permet de définir un diviseur pour la mise à l’échelle de la valeur à envoyer au calculateur. Ici, le diviseur est de 50. La valeur envoyée au calculateur est donc Valeur_envoyée=Valeur_affichée / 50

L’instruction TYPE=8 défini le nombre d’octet de la valeur à transmettre au calculateur. Ici, la valeur est de 8 bits.

L’instruction UNIT= défini le texte d’unité à afficher derrière la valeur.

L’instruction PAS= défini le pas qu’il faut appliquer lors d’un appui sur un bouton. Le pas peut être un nombre réel. Il est ici de 50.

L'instruction CDE= permet de définir la commande à envoyer lorsque l'on clique sur un des boutons < ou >. Il faut mettre à l'intérieur de la trame à envoyer le terme VALUE pour indiquer au logiciel ou mettre la valeur de commande proportionnelle et le terme SENS pour indiquer ou mettre la valeur du SENS à envoyer au calculateur.

- CLK=,1/LIST/EXIT=/CASE=1,2-9/NAME=Ecriture de la date et de l'heure/TITLE=Date et heure/COR=0.25|0:1|0:1|0:1|0:0.25|0:1|1985:1|-125:1|-125/TYPE=8:8:8:8:8:8:8:8/CDE=2E F9 0B VALUE;

Ce type d'affichage permet de modifier toute une liste de paramètres en même temps. Cette fenêtre est adaptée à la modification d'une liste de paramètres contenus dans une seule trame KWP. Les données doivent obligatoirement être dans l'ordre d'apparition dans la trame, sinon des erreurs peuvent se produire.

The screenshot shows a window titled "Ecriture de la date et de l'heure". Inside, there is a section labeled "Date et heure" and a table with the following data:

Parametres	Valeur	Commentaire
Secondes	8	
Minutes	8	
Heure	14	
Mois	7	
Jour	19,75	
Annee	2009	
Offset Minut	0	
Offset Heur	-1	

Below the table, there is a text input field, an "OK" button with a green checkmark, a "Fermer" button, and a large right-pointing arrow button.

L'instruction EXIT= permet d'indiquer si on doit envoyer une commande lorsque l'on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n'envoi pas de trame mais qu'il y a un mode de déconnection par un bouton en bas de la page.

L'instruction CASE=1,2-9 permet de définir la plage de donnée à copier entre la grille du programme principale et la fenêtre de liste de paramètres. La valeur à mettre avant la virgule est le numéro de la colonne. Ensuite, après la virgule, vient la plage des

lignes de valeurs qui devra être copiée soit ici les lignes de 2 à 9 inclus soit 8 paramètres.

L'instruction NAME= Ecriture de la date et de l'heure : fixe le nom de la fenêtre.

L'instruction TITLE= Date et heure : permet de fixer le texte au dessus de la grille.

L'instruction COR=0.25|0:1|0:1|0:1|0:0.25|0:1|1985:1|-125:1|-125 permet de définir les valeurs de correction à appliquer aux paramètres avant l'envoi de la trame. Il faut impérativement définir autant de valeur de correction qu'il y a de paramètres sinon, tous les paramètres ne seront pas envoyés. Chaque groupe de correction est séparé par un « : ». La pente est séparé de l'offset par « | ». Pour rappel, la valeur affichée dans la case est $Y = \text{Pente} * X + \text{Offset}$. Avec X étant la valeur brute de la trame et Y, la valeur mise à l'échelle.

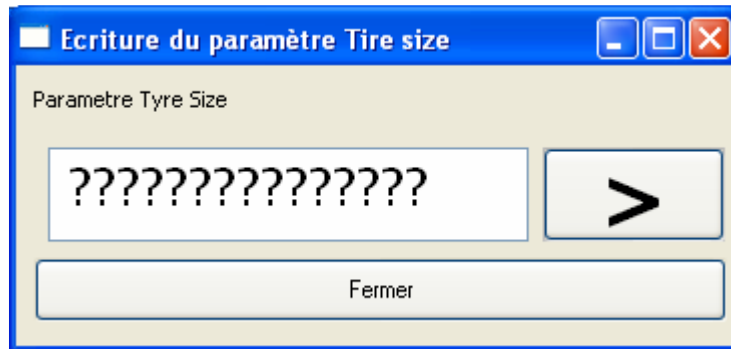
L'instruction TYPE=8:8:8:8:8:8:8:8 définit le nombre de bit de chaque paramètre à transmettre au calculateur. Chaque longueur doit être séparé par « : ». Ici, tous les paramètres sont transmis avec une longueur de 8 bits. Au dessus de 8 bits, il faut spécifier l'ordre des octets en ajoutant « L » si le sens de lecture est octet de poids faible vers octet de poids fort ou « H » dans l'autre cas (ce cas est le mode par défaut et il peut-être omis). Comme pour la correction, Il faut impérativement définir autant de valeur de type qu'il y a de paramètres sinon, tous les paramètres ne seront pas envoyés.

L'instruction CDE=2E F9 0B VALUE permet de définir la commande à envoyer lorsque l'on clique sur le bouton >. Il faut mettre à l'intérieur de la trame à envoyer le terme VALUE pour indiquer au logiciel ou mettre la valeur de commande contenant la suite d'octet de chaque paramètre.

NOTA : après ce genre de fenêtre, lorsque l'on quitte la fenêtre, le logiciel rafraîchi la grille principale en interrogeant à nouveau le calculateur.

- CLK=,18/VALUE/EXIT=/CASE=1,18/NAME=Ecriture du paramètre Impulsion par tour/TITLE=Parametre Impulsion par tour/DIV=0.001/TYPE=16H/CDE=2E F9 1E VALUE;

Ce type d'affichage est adapté à la modification d'un paramètre unique qui est modifiable par une trame unique ne contenant que ce paramètre.



L'instruction EXIT= permet d'indiquer si on doit envoyer une commande lorsque l'on ferme la fenêtre. Il faut donc mettre la commande à envoyer ou mettre GLOBAL pour indiquer que la fermeture n'envoie pas de trame mais qu'il y a un mode de déconnection par un bouton en bas de la page.

L'instruction CASE=1,18 permet de définir la case de donnée à copier entre la grille du programme principale et la zone d'édition de la fenêtre valeur. La valeur à mettre avant la virgule est le numéro de la colonne. Ensuite, après la virgule, vient le numéro de la ligne.

L'instruction NAME= Ecriture de la date et de l'heure : fixe le nom de la fenêtre.

L'instruction TITLE= Date et heure : permet de fixer le texte au dessus de la grille.

L'instruction DIV permet de définir un diviseur pour la mise à l'échelle de la valeur à envoyer au calculateur. Ici, le diviseur est de 1. La valeur envoyée au calculateur est donc Valeur_envoyée=Valeur_affichée / DIV

L'instruction TYPE=16L définit le nombre d'octet de la valeur à transmettre au calculateur. Ici, la valeur est de 16 bits. Le classement des octets se fait en mode L soit l'octet de poids faible avant celui de poids fort.

L'instruction CDE= permet de définir la commande à envoyer lorsque l'on clique sur le bouton >. Il faut mettre à l'intérieur de la trame à envoyer le terme VALUE pour indiquer au logiciel ou mettre la valeur à envoyer au calculateur.

NOTA : après ce genre de fenêtre, lorsque l'on quitte la fenêtre, le logiciel rafraîchi la grille principale en interrogeant à nouveau le calculateur.

3-4-3) Le champ <KWP>

A l'intérieur du menu, on peut avoir plusieurs commandes KWP à émettre. Pour un affichage GRID, dans le champ KWP, il y a une instruction CDE et une ou plusieurs instructions TR=. Elles sont définies comme suit l'exemple :

```
<KWP>
CDE=21 03 01;
TR=[6|6]/ROW=1/GR3=IMG_GREEN;
TR=[12|4-7]/V=50|0/ROW=43/GR3=VALUE;
TR=[21|0]/ROW=9/GR4=IMG_RED;
TR=[22|5]/ROW=17/GR3=[PWM mesure,Period Measure];
TR=[3-12]/ROW=9/GR3=VAL_TXT;
TR=[3]/ROW=9/GR3=VAL_HEX;
TR=[3]/ROW=9/GR3=FILE(DTC.DTC);
```

<\KWP>

Explication des champs :

- CDE=21 03 01;
Cette instruction définit la commande à envoyer au calculateur.
- TR=[6|6]/ROW=1/GR3=IMG_GREEN;
TR=[12|4-7]/V=50|0/ROW=43/GR3=VALUE;
TR=[21|0]/ROW=9/GR4=IMG_RED;
TR=[22|5]/ROW=17/GR3=[PWM mesure, Periode Measure];
TR=[3-12]/ROW=9/GR3=VAL_TXT;
TR=[3]/ROW=9/GR3=VAL_HEX;
TR=[3]/ROW=9/GR3=FILE(DTC.DTC);

L'instruction **TR=** est la ligne qui explique au logiciel que faire des données renvoyées par le calculateur.

- ROW=
Cette instruction définit le numéro de ligne de la grille sur lequel s'applique les données qui suivent. Le numéro de ligne commence par 1.
- Le caractère /
Dans les lignes d'instructions, il permet de séparer les données ou instruction. Dans cet exemple, on sépare **[1-64]L** de **TX=**.
- CL=FDC968
Il est possible d'indiquer une couleur pour la ligne. Cette instruction définit la couleur à appliquer à la ligne concernée. Si cette instruction n'est pas précisée, la couleur sera blanche à moins d'avoir défini la couleur de la ligne dans la zone de configuration par l'instruction **LG=**. La valeur de la couleur doit être donnée en hexadécimal.

○ **[1-64]L**

Les valeurs contenu entre crochet doivent toujours être placée juste après la commande **TR=**. Il permet au programme d'interpréter la valeur renvoyée par le calculateur qui peut être contenu dans une plage d'octets ou une plage de bits. Dans cet exemple, on place dans une valeur entière, les octets N°1 à 64 en valeurs entières.

Mais cet exemple est simple. La traduction peut être plus complexe comme celle-ci :

R5[8|5-7]L :

- **R5** défini que l'on doit répéter la traduction sur tous les octets de la trame en partant de l'octet n°5 et ceci jusqu'à la fin de la trame reçue.
- **[8|5-7]** défini une plage de bit dans un octet. Donc on récupère la valeur contenue dans l'octet 8, sur la plage de bit de 5 à 7.
- **L** indique que les octets sont écrits en mode Octet de poids faible puis octet de poids fort. L'inverse est le mode par défaut, on n'indique rien, sinon, on le précise par **H**.

○ Autre possibilité dans la ligne de traduction : **V=4|20 :**

Il faut placer cette instruction juste après l'instruction de décodage. Elle permet la mise à l'échelle de la valeur traduite précédemment. Nous allons donc transmettre pour l'affichage Valeur=4* Valeur_traduite+20.

○ **GR3=.... ;**

L'instruction GR permet de spécifier ce qu'il faut mettre dans la case de la colonne (ici 0). Si le texte n'est pas une commande réservée (VALUE, VAL_TXT, VAL_HEX, TAB, IMG_GREEN, IMG_RED) le logiciel affiche le texte directement sans conversion d'aucun type.

- **VALUE** => l'affichage correspond à la valeur réelle traduite
- **VAL_HEX** => l'affichage correspond à la valeur réelle affichée en hexadécimal
- **VAL_TXT** => l'affichage est un texte dont les octets représentent des codes ASCII.
- **TAB[x,y]** => l'affichage correspond au texte contenu dans le tableau de valeurs n°x et d'indice n°y.
- **IMG_GREEN** => l'affichage correspond à un voyant vert qui prend pour état : éteint si la valeur traduite est nulle, allumée si la valeur traduite est différente de 0.
- **IMG_RED** => l'affichage correspond à un voyant rouge qui prend pour état : éteint si la valeur traduite est nulle, allumée si la valeur traduite est différente de 0.
- **[txt0,txt1,txt2,txt3,txt4]** => l'affichage correspond au texte dont l'indice est égal à la valeur traduite. Si valeur=0, on affiche le txt0, si valeur=1, on affiche txt1 et ainsi de suite. Si la valeur dépasse la liste, le logiciel indique indéfini.
- **FILE(DTC.DTC)** => le programme va ouvrir le fichier de type texte DTC.DTC et rechercher à l'intérieur si la valeur lu du calculateur est présente à l'intérieur, si oui, il affiche le champ de la même ligne.

Par exemple : si DTC contient les données suivante
1 ; défaut de vanne EGR ;

2 ; défaut de clignotant ;
3 ; défaut de moteur essui vitre.

Lors de la lecture, le calculateur renvoi l'information 2, le programme va afficher « défaut de clignotant ».

IV) Le champ <EXMENU=>

Pour certain calculateur, il est possible de répéter un menu en fonction d'une valeur précédemment lu dans celui-ci. Par exemple :

Si vous avez une architecture électronique ayant un boîtier maître et trois boîtiers esclaves identiques. En interrogeant le boîtier maître, il peut vous l'indiquer. Il est donc possible de configurer un EXMENU commun à tous les boîtiers esclaves. Et un MENU pour le boîtier maître.

La particularité est qu'il y a des instructions supplémentaires pour gérer la répétition.

Exemple :

```
<EXMENU=Diagnostic IOU>
ICO=0;
EXREPEAT=1; // la condition de répétition du menu se trouve dans la variable N°1
TITLE=Label/Pin/Référence/Etat/Defaut/Contact/Commentaire;
TYPE=GRID/7;
REPEAT=100;

LG=ROW=1/GR0=TAB[1,51];
LG=ROW=2/CL=FDC968/GR0=EREV 0/GR1=CN 4.16/GR2=TAB[1,52]/GR5=GND;
LG=ROW=3/CL=FFF3DD/GR0=EREV 1/GR1=CN 4.17/GR2=TAB[1,53]/GR5=GND;

<CLICK>
EXIT=32 01;
ACTIF=31 01;
CLK=,26/ONOFF/EXIT=GLOBAL/NAME=Commande logique/TITLE=OUT 0 - CN 5.24 -
CN5.25/ON=30 00 EXV 01 00 00/OFF=30 00 EXV 00 00 00;
CLK=,30/ONOFF/EXIT=GLOBAL/NAME=Commande logique/TITLE=OUT 1 - CN
5.7/ON=30 01 EXV 01 00 00/OFF=30 01 EXV 00 00 00;
<\CLICK>

<KWP>
CDE=21 03;
TR=[6|6]/ROW=1/GR3=IMG_GREEN;
TR=[6|7]/ROW=2/GR3=IMG_GREEN;
<\KWP>
<\EXMENU>
```

Comme l'écriture est très proche du menu MENU, il est traité uniquement des différences. Donc le champ <EXMENU=...> se termine par <\EXMENU>. Le logiciel a besoin de l'instruction EXREPEAT=. Elle permet d'indiquer au logiciel dans quelle variable est contenu le nombre de répétition du menu. Le logiciel affichera dans la zone de menu le titre suivant EXMENU= en ajoutant l'indice (qui commence par 1.

Le logiciel va ensuite répéter les instructions contenues dans le champ<EXMENU..> autant de fois que la valeur de la variable pointée par EXREPEAT. Le logiciel va également

incrémenter le numéro d'indice des tableaux. Ainsi TAB[1,51] sera incrémenté à chaque répétition. Dans notre exemple, TAB[1,51] deviendra pour le second menu TAB[1,54], pour le troisième menu, il deviendra TAB[1,57] et ainsi de suite...