Projets Demon et Eurogam **EDOC138**

Schémas synoptiques sous tcard V1.41

Edition : 1.2
Date : 29 juin 1992
Auteurs : G.ZEHNACKER - J.Y.OBERLE

Centre de Recherches Nucléaires de Strasbourg
CNRS-IN2P3 France

Sommaire

1 Introduction

Le logiciel tcard a été conçu dans le but de permettre la manipulation aisée de l'électronique de commande associée à des détecteurs tels que ceux utilisés dans le cadre des projets DEMON, EUROGAM, etc. Cela passe par l'affichage d'un synoptique décrit dans un fichier .syn dont nous préciserons les règles de formation par la suite.

2 Description des éléments de base

Dans sa version actuelle, tcard intègre treize éléments de base permettant la création rapide de schémas synoptiques. Ces éléments sont décrits ci-dessous.

2.1 Le rectangle (rectangle)

Le rectangle est utilisé pour la représentation des modules électroniques:

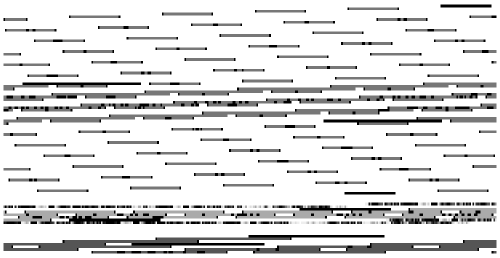


fig.4 *fig.5* *fig. 6*

2.4 Le registre de commande (command register)

Comme leur nom l'indique, ces registres symbolisent les commandes associées aux différents modules électroniques (seuil, retard, durée, etc.):

fig. 13

2.8 L'arc (arc)

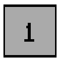


fig. 7

L'arc est une ellipse de couleur différente de celle décrite précédemment:

2.5 L'ellipse (ellipse)

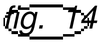


fig. 14

fig. 8

Ces ellipses sont prévues pour les fenêtres "pop-up". Il est par conséquent déconseillé de les utiliser dans un schéma synoptique.

Les ellipses sont destinées à représenter des points de connexion entre lignes ou encore des points de connexion externes comme le montrent les figures 9 et 10 ci-dessous:

2.9 Le pointillé (dash line)

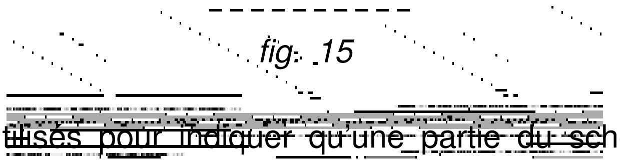


fig. 15

Les pointillés sont utilisés pour indiquer qu'une partie du schéma synoptique est commune à plusieurs voies d'une carte électronique:

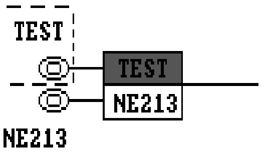


fig. 16

2.10 Le texte (text)

Il permet de placer un texte à une position quelconque sur l'écran:

TEXTE

fig. 17

2.11 Le point de test (scope test point)

Cet élément se présente sous la forme d'une ellipse:

fig. 18

fig. 19

Ces points de test sont placés aux endroits où il est possible de visualiser un signal de la carte sur oscilloscope:

fig. 21

2.13 La fonte (newfont)

Elle permet de changer de fonte pour tout texte imprimé à l'écran, qu'il soit contenu dans un rectangle, un bouton, un zoom, etc. Elle n'a pas de symbole à l'écran.

2.14 Le zoom arrière (unzoom)

Le zoom arrière permet de revenir à l'écran initial si l'on se trouve précisément dans un zoom:

fig. 22

3 Construction d'un schéma synoptique

3.1 Préliminaires

La construction d'un schéma synoptique passe par la création d'un fichier de définition identifié par son suffixe .syn (par exemple demon.syn).
Celui-ci doit contenir le type et les coordonnées des éléments de base que l'on souhaite voir apparaître à l'écran.
Il faut, pour l'écriture de ce fichier, avoir dessiné au préalable le schéma synoptique sur une feuille de papier qui pourra être quadrillée afin de faciliter le

relevé des coordonnées des différents éléments.

3.2 Entête du fichier

Le fichier doit obligatoirement débuter par une ligne du type:

```
#         0.    0.          40.5        24.   carte_demon
```

Celle-ci précise les dimensions du schéma qui a été mis sur papier. Plus précisément, les deux premiers nombres représentent les coordonnées du coin supérieur gauche et les deux suivants les coordonnées du coin inférieur droit.

La ligne peut éventuellement être terminée par une chaîne de caractères (qui ne doit pas contenir d'espace). En son absence, il faut mettre un souligné (_).

3.3 Corps du fichier

Le corps du fichier doit contenir la description des éléments de base présents sur le schéma.
Nous allons détailler la syntaxe de ces descriptions pour chaque type d'élément.

3.3.1 Le rectangle

```
R         3.    12.2        7.3   15.         Rectangle
```

La lettre R indique ici que l'on définit un rectangle (les correspondances entre lettres et éléments de base sont rappelées en annexe I).
Les deux premiers nombres correspondent aux coordonnées du coin supérieur gauche du rectangle et les deux derniers à celle du coin inférieur droit.
La chaîne de fin de ligne ("Rectangle") est inscrite dans le rectangle qui est dessiné à l'écran.

3.3.2 Le convertisseur analogique numérique (C.A.N.)

```
A         5.    4.4         7.    6.          1_adc_SYREP
```

La description de cette ligne est analogue à celle du paragraphe précédent sauf en ce qui concerne la chaîne.
En effet, cette dernière doit obligatoirement commencer avec le numéro du C.A.N. (ici 1).
De plus, seules les cinq premières lettres seront affichées dans le C.A.N.

représenté à l'écran. Les suivantes le seront dans la fenêtre "pop up" qui s'ouvrira lors d'un clic sur le C.A.N. Il faut en effet savoir qu'en cliquant sur un de ces symboles, on démarrera ou on arrêtera la formation d'un spectre dans le "Resource Manager" du chassis VXI et on ouvrira une fenêtre "pop up" de réglage du spectre.

3.3.3 Le bouton

B	*8.*	*8.5*	*10.*	*10.9*	*bouton_vertical*
ou					
H	*12.*	*5.*	*13.5*	*5.8*	*bouton_horizontal*

Suivant que l'on souhaite afficher un bouton vertical ou horizontal, celui-ci devra être défini par la lettre B ou H.
La syntaxe de la ligne est, pour les deux types de boutons, identique à celle du rectangle mis à part le fait que la chaîne n'est en aucun cas reportée à l'écran. En pratique, le texte qui sera affiché se trouve dans un autre fichier de définition qui sera vu plus loin. Ce fichier précise également la forme que devra prendre le bouton (double ou simple).

3.3.4 Le registre de commande

C	*3.3*	*4.25*	*3.8*	*4.75*	*1*

La syntaxe de la ligne est identique à celle du rectangle.
Le dernier chiffre ne représente par conséquent non pas une abscisse ou une ordonnée mais bien une chaîne qui sera inscrite dans le registre de commande. On pourra retenir qu'en cliquant sur le symbole d'un registre de commande, on ouvrira une fenêtre "pop up" permettant un réglage de valeur dans les registres des cartes électroniques.

3.3.5 L'ellipse

E	*14.*	*5.*	*15.*	*6.*	*ellipse_1*

Les coordonnées indiquées sont celles du rectangle dans lequel est inscrite l'ellipse.
La chaîne n'est pas reportée sur le schéma.

3.3.6 La fonction

```
F       10.   12.      13.   13.      power_supply
```

Les coordonnées définissent le rectangle (sans texte) symbolisant la fonction à l'écran.
La chaîne de fin de ligne correspond au nom du programme qui sera appelé lors d'un clic dans le rectangle.
Il est à noter que les fonctions sont extérieures à tcard donc développables indépendamment de ce produit.

3.3.7 La ligne

```
L       12.6  3.6      20.7  3.6      ligne_1
```

Les coordonnées précisent les deux extrémités de la ligne.
La chaîne n'est pas affichée sur le schéma.

3.3.8 L'arc

```
O       15.   15.      18.   17.      arc_1
```

La description est identique à celle faite dans le cas de l'ellipse.

3.3.9 Le pointillé

```
P       3.    2.       10.   2.       pointille_1
```

La syntaxe est identique à celle de la ligne.

3.3.10 Le texte

```
T       4.    9.       0.    0.       Texte
```

Les deux premiers nombres correspondent aux coordonnées du coin inférieur gauche du rectangle contenant le texte.
Les deux nombres suivants ne sont pas pris en compte. On peut par convention leur donner la valeur nulle.
En dernière position, se trouve le texte à afficher.

3.3.11 Le point de test

```
V          20.   6.         21.   6.5        L1
```

ou

```
V          18.   3.         19.   3.5        A1
```

Le point de test est symbolisé à l'écran par une ellipse.
Les coordonnées sont celles du rectangle dans lequel est inscrite l'ellipse.
La chaîne est reportée sur le schéma. De plus, le point de test prend une couleur différente suivant qu'il s'agit d'un point logique (commençant par la lettre L; par exemple L1) ou d'un point analogique (commençant par la lettre A; par exemple A1).
En cliquant sur ces points de test, on ouvrira une fenêtre "pop up" permettant l'affichage d'un signal sur un oscilloscope branché sur le "Resource Manager".

3.3.12 Le zoom

```
Z          25.   6.         27.   8.         Temps
```

Les coordonnées sont comme déjà vu celles du coin supérieur gauche et du coin inférieur droit du zoom.
La chaîne est reportée sur le schéma mais elle y est précédée du mot Roci. Ainsi, dans le cas de l'exemple ci-dessus, on pourra lire:

Roci
Temps

3.3.13 La fonte

```
N          0.    0.         0.    0.         times-bold-12
```

Tous les textes affichés à la suite d'une telle ligne le sont dans la fonte décrite en dernière position (*times-bold-12* dans l'exemple). Les chiffres ne sont pas pris en compte.
Attention: tcard ne vérifie pas l'existence de la fonte; il faut par conséquent s'assurer qu'elle est bien disponible avant de l'utiliser.

3.3.14 Le zoom arrière

```
U        10.   5.       12.   6.         Zoom_Arriere
```

Cette fonction dessine un rectangle dont les coordonnées sont définies de manière habituelle. La chaîne de fin de ligne est inscrite dans ce rectangle.
Le zoom arrière est bien évidemment uniquement destiné aux zooms.

3.4 Fin de fichier

La fin du fichier .syn doit être marquée par la ligne suivante:

```
^D       0.    0.       0.    0.         fin_du  fichier
```

Le caractère ^D est obtenu par l'appui simultané des touches Control et D. La chaîne de fin de ligne peut être quelconque mais doit être présente.

3.5 Commentaires

Afin de le rendre plus visible, le fichier peut être commenté.

Le début d'un commentaire est marqué par le signe #.
Par exemple:

```
L    12.   20.      15.   20.        _    # Ligne_horizontale
```

Il faut rester attentif au fait qu'un commentaire ne peut contenir d'espace.

4 Fichiers complémentaires

4.1 Introduction

Pour qu'un schéma synoptique soit exploitable, il faut créer en plus du fichier .syn un ou plusieurs fichiers complémentaires.
L'objet de ce paragraphe est la description de ces fichiers.

4.2 Le fichier .data

Ce fichier, comme l'indique le titre, doit avoir un nom se terminant par .data (par exemple demon.data).
Il contient des données relatives aux registres de commande, aux boutons et aux points de test.

Il doit débuter par une ligne indiquant le nombre de voies devant être gérées par le logiciel. Par exemple, dans le cas de DEMON, on pourra lire:

```
30    demon    96    voie      1    96
```

On voit qu'il y a 96 voies numérotées de 1 à 96.
30 indique le nombre de lignes du fichier (la première n'étant pas prise en compte).

Les lignes suivantes doivent décrire les caractéristiques des registres de commande (valeurs courante, minimale et maximale):

```
1     seuil    40    mv        0    100
2     retard   50    ns        40   60
3     duree    300   ns        200  400
```

Les registres doivent obligatoirement être définis dans le même ordre que dans le fichier .syn.

La suite du fichier doit préciser les caractéristiques des boutons:

```
1     TEST     NE213     _     test      NE213
2     Sumbus   Sumbus    _     Sumbus    _
```

Les deux derniers mots précisent les deux états possibles du bouton. Lorsqu'il n'y a pas de souligné, le bouton obtenu est double (bouton *TEST*). Dans le cas contraire, il est simple (bouton *Sumbus*).

Le second mot désigne l'état du bouton lorsqu'il est dessiné à l'écran et le premier est un simple commentaire.

Comme les registres de commande, les boutons doivent être définis dans le même ordre dans les fichiers .syn et .data.

Le fichier .data doit finir par la description des boutons de commande. La syntaxe de cette description est très simple:

```
1     s1    A1    _    _    _
2     s2    A2    _    _    _
3     s3    L1    _    _    _
```

Seuls A1, A2 et L1 sont pris en compte par le logiciel, s1, s2 et s3 étant de simples commentaires.

L'ordre de définition doit à nouveau être respecté dans les deux fichiers.

4.3 Les fichiers _roci.syn et _roci.data

Si l'on a défini des zooms dans le premier fichier, il faut créer un fichier supplémentaire dont le nom se termine par _roci.syn (par exemple demon_roci.syn).

Il faut en effet savoir que le fait de cliquer sur un zoom à l'écran provoque l'affichage d'un nouveau schéma synoptique.
Cet affichage se fait à partir du fichier _roci.syn dont la syntaxe est identique à celle du fichier .syn.

Il faut de même, et si nécessaire, créer un fichier _roci.data (par exemple demon_roci.data) contenant les données relatives aux registres de commande, aux boutons et aux points de test du fichier _roci.syn.

Si l'on utilise les programmes de simulation de la carte, il faut de plus savoir que le logiciel tcard a besoin d'une copie du fichier _roci.data en autant d'exemplaires qu'il y a de zooms. Ces copies doivent être numérotées de la manière suivante: _roci_1.data, _roci_2.data, _roci_3.data, etc.

5 Fichiers de commande

La version 1.41 de tcard offre la possibilité d'associer un fichier de commande à chaque bouton. Ce fichier permet l'exécution d'un certain nombre d'ordre lorsque l'utilisateur clique sur un bouton donné.

5.1 Nom des fichiers

Les fichiers de commande sont liés aux boutons par leurs noms qui doivent suivre la syntaxe suivante:

<nom de la carte>_<numéro du détecteur>_b_<numéro du bouton>.button,

où: * *nom de la carte* peut être demon, bgo, etc.
* *numéro du détecteur* est soit le numéro du détecteur soit 0.
Dans le premier cas, si le numéro correspond au synoptique actuellement affiché, le fichier est pris en compte.
Dans le second (numéro nul), le fichier est considéré comme valable quelque soit le numéro du synoptique présent à l'écran.
* *numéro du bouton* est le nombre associé par tcard au bouton.

Ce nombre apparaît dans le champs Choice de la fenêtre de tcard (partie supérieure) lorsque l'on clique sur le bouton.

On peut citer, à titre d'exemple, comme noms de fichiers: demon_0_b_10.button, bgo_3_b_2.button, etc.

5.2 Commandes

Chaque fichier peut contenir plusieurs commandes qui seront exécutées lorsque le nom du fichier correspondra au bouton cliqué.
Ces commandes sont actuellement au nombre de trois:

* *InvSelecBut* qui inverse l'état du bouton cliqué.
  Il faut savoir que le fait de cliquer sur un bouton ne modifie pas automatiquement son état si un fichier de commande associé est trouvé. Cet ordre permet de remédier à cet état de fait.

* *IfButtonHigh* qui permet de modifier l'état d'un autre bouton si le bouton cliqué est à l'état haut (i.e. allumé pour un bouton simple, partie supérieure ou gauche allumée pour un bouton double).
  La syntaxe de cette commande est:
  IfButtonHigh <nom de la carte> <numéro du détecteur> b <numéro du bouton> <on/off>

  nom de la carte, *numéro du détecteur* et *numéro du bouton* ont la même signification que précédemment.
  Il est à noter que si le numéro du détecteur est différent du numéro du synoptique actuellement affiché par tcard, l'ordre est ignoré (la possibilité d'agir sur des cartes non affichées pourra éventuellement être prise en compte par une version future de tcard). Si le numéro est 0, la commande est réalisée sur le synoptique présent à l'écran.
  on/off permet respectivement de mettre le bouton désigné à l'état haut ou bas.

* *IfButtonLow* qui permet de modifier l'état d'un autre bouton si le bouton cliqué est à l'état bas.
  Sa syntaxe est celle de la commande précédente.

Des exemples de commandes peuvent être: *IfButtonLow demon 0 b8 off ;* ou *IfButtonHigh germanium 12 b 1 on ;* ou encore *InvSelecBut ;* , etc.
Il est impératif que chaque commande soit suivie d'un point virgule.

6 Les fichiers images

Afin d'accélérer l'affichage des synoptiques, tcard, dans sa version 1.41, est en mesure d'utiliser les fichiers images (ou .bitmap) créés par tcardtool (voir EDOC 139). Si le fichier image associé à une carte n'existe pas, tcard utilise de manière classique le fichier .syn (qui doit toujours rester présent).

7 Exploitation des fichiers

Pour que les fichiers précédemment créés soient exploitables par tcard, il suffit de tous les placer dans un répertoire sur lequel pointe la variable d'environnement SYNOPTIQUE.
Le plus simple est de nommer ce répertoire SYNOPTIQUE (bien que cela ne soit pas obligatoire et ne dispense pas de créer la variable du même nom).

Ce répertoire doit également contenir les programmes pouvant être lancés à partir d'un clic de la souris sur les rectangles "fonction".

8 Quelques contraintes

tcard est un logiciel très puissant. Cependant, il impose quelques limitations quant au nombre d'éléments de base pouvant être présents simultanément sur un schéma synoptique.
On est ainsi limité à:

80 registres de commandes
80 boutons
80 points de test

D'autre part, il faut savoir que l'on est limité à **80** caractères par champ dans les fichiers.
Par exemple, la ligne suivante est incorrecte:

```
R          3.     12.2        7.3   15.   Ce_champ_est_trop_long_
                                          car_il_dépasse_la_limite_
                                          admissible_de_quatre_
                                          vingts_caractères
```

De plus, les champs doivent obligatoirement être au nombre de six par ligne; on peut au besoin remplacer un champ par un souligné comme dans l'exemple ci-dessous:

```
L          3.    10.        3.    15.        _
```

Il est à noter que toutes ces limites pourront être modifiées, si cela s'avérait nécessaire, dans les prochaines versions de tcard.

9 Quelques conseils

Le premier conseil concerne les rectangles.
Quelques essais montrent que le texte (la chaîne de fin de ligne) qui y est inscrit par le logiciel n'est pas centré.
Pour remédier à ceci, il est conseillé d'y inscrire un texte à l'aide de la fonction T:

```
R          10.   15.        15.   17.5       _         # Pas_de_texte
T          16.8  12.        0.    0.         Rectangle   # Texte
```

En pratique, ceci ne sera plus nécessaire pour la prochaine version de tcard qui centrera automatiquement le texte.

Le second conseil concerne tous les éléments de base pouvant contenir du texte (C.A.N., point de test, etc.).
Il est prudent de les prévoir assez large pour que le texte puisse y tenir (la police utilisée par défaut prenant assez de place notamment lors de l'utilisation de majuscules).

On peut également faire une remarque à propos des fonctions.
Il a été vu plus haut que la chaîne de fin de ligne n'était pas inscrite dans le rectangle représentant la fonction. Si l'on souhaite cependant y voir apparaître du texte, il est possible d'utiliser la fonction T comme dans le cas du rectangle.

D'autre part, il faut savoir que la gestion des fonctions par tcard autorise l'emploi de fonctions cachées sous d'autres éléments (par exemple sous un bouton ou un point de test). Ceci permet de lancer un programme spécifique lors d'un clic sur un de ces éléments.
Pour qu'une fonction soit placée *sous* un élément, elle doit être déclarée *avant* cet élément dans le fichier .syn (la lecture de ce fichier se faisant linéairement).

Enfin, pour la réalisation d'un schéma synoptique, on pourra s'aider de celui, fictif, donné en annexe II ainsi que de ses fichiers .syn et .data reproduits en annexes III et IV.

10 Conclusion

tcard offre une multitude de symboles pour représenter les différents modules des cartes électroniques.
Cependant, si les besoins s'en font sentir, il pourra être envisagé une extension de ces symboles, donc des objets nécessaires pour piloter l'électronique.

Annexe I
Correspondance entre lettres et éléments de base

A : convertisseur analogique numérique (adc)

B : bouton vertical (button)

C : registre de commande (command register)

E : ellipse (ellipse)

F : fonction (function)

H : bouton horizontal (button)

L : ligne (line)

N : fonte (newfont)

O : arc (arc)

P : pointillé (dash line)

R : rectangle (rectangle)

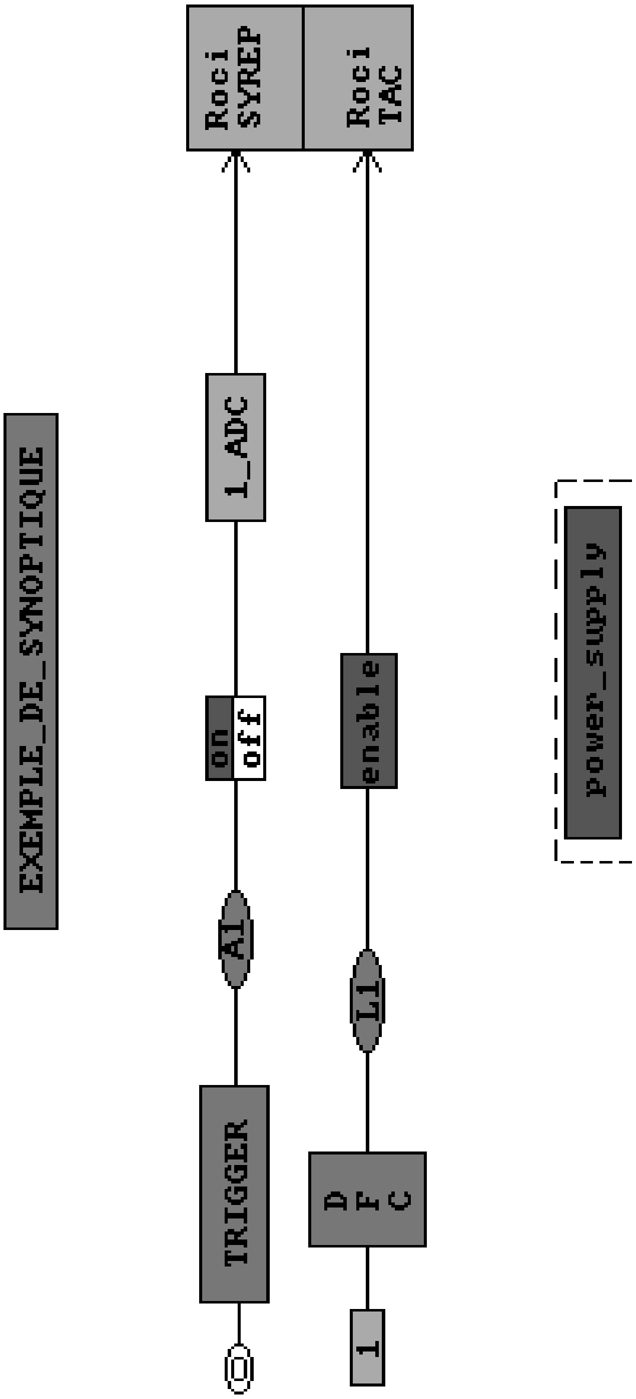
T : texte (text)

U : zoom arrière (unzoom)

V : point de test (scope test point)

Z : zoom (zoom)

Annexe II
Exemple de schéma synoptique



Annexe III
Listing du fichier exemple.syn

```
#     0.     0.          22.5   13.5        Exemple_de_synoptique

#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                        Lignes
#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

L     3.25   4.          18.    4.          _
L     3.5    6.          18.    6.          _

#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                       Fonctions
#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

F     10.75 5.5          11.75 6.5          enable       #  fonction_cachée
F     9.5    9.5         13.    10.5        power_supply

  #   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                        C.A.N.
#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

A     14.4   3.5         15.6   4.5         1_ADC_SYREP

#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                        Boutons
#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

B     11.    3.5         11.5   4.5         bouton_double
B     10.75 5.5          11.75 6.5          bouton_simple

#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                    Registres de commande
#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

C     2.5    5.75        3.5    6.25        1

#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                         Ellipses
#   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
E       2.75    3.75            3.25    4.25            _
E       2.9     3.9             3.1     4.1             _

#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                                 Pointillés
#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

P       9.      9.              9.      11.             _
P       9.      9.              13.5    9.              _
P       13.5    9.              13.5    11.             _

#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                                Rectangles
#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

R       4.      3.5             6.5     4.5             _
R       4.      5.              4.75    7.              _
R       8.25    0.5             14.25 1.5               _

#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                               Points de test
#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

V       8.      3.75            9.      4.25            A1
V       7.      5.75            8.      6.25            L1

#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                                   Zoom
#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Z       18.     3.              19.5    5.              SYREP
Z       18.     5.              19.5    7.              TAC

#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#                                 Textes
#    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

T       9.2     1.2             0.      0.              EXEMPLE_DE_SYNOPTIQUE
T       4.5     4.2             0.      0.              TRIGGER
T       4.3     5.7             0.      0.              D
T       4.3     6.2             0.      0.              F
T       4.3     6.7             0.      0.              C
T       10.3    10.1            0.      0.              power_supply
```

```
T     17.9  4.         0.    0.          \                # fleche
T     17.9  4.3        0.    0.          /                # fleche
T     17.9  6.         0.    0.          \                # fleche
T     17.9  6.3        0.    0.          /                # fleche

^D    0.    0.         0.    0.          fin_de_fichier
```

Annexe IV
Listing du fichier exemple.data

```
5       exemple _de_synoptique                  1       voie            1       1
1       seuil_DFC                               40      mv              0       100
1       bouton_double                           on      _               on      off
2       bouton_simple                           enable          _       enable          _
1       signal_trigger                          A1      _               _       _
2       signal_DFC                              L1      _               _       _
```