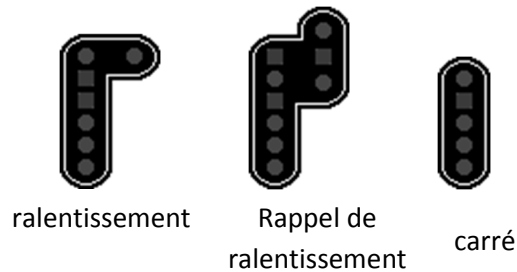


## Notice d'utilisation du programme signaux\_complexes\_GL

Cette notice montre comment utiliser le programme signaux\_complexes\_GL pour utiliser des signaux complexes français avec CDM rail.

Exemples de signaux complexes :



Rappel de  
ralentissement combiné à  
un avertissement

**Matériel nécessaire :**

- Le programme « signaux\_complexes\_GL »
- CDM rail V5.1 mini.
- décodeur de feux « led dekodeur » de **DigitalBahn** équipé de son logiciel « led\_signal\_10 » ou un décodeur **CDF** ou **LEB** ou **LDT-DEC-SNCF** ou **Leb-Modélisme**

<b>Introduction</b>	<b>3</b>
<b>Installation</b>	<b>3</b>
Exclusion de sécurité	4
<b>Fonctionnement</b>	<b>5</b>
Différences entre cantons et zones de détections	6
Signaux complexes virtuels et réels	6
<b>Restrictions et spécificités du programme client pour les signaux</b>	<b>7</b>
<b>Modélisation du réseau</b>	<b>8</b>
<b>Fichier de configuration client-GL.cfg</b>	<b>9</b>
<b>Utilisation du programme signaux_complexes_GL avec CDM rail</b>	<b>11</b>
<b>Modification du programme avec Delphi 7</b>	<b>15</b>
Installation des composants socket (TClientSocket et TServerSocket)	15
Installation du composant MScomm32	15
Défaillances possibles	16

## Introduction

Signaux\_complexes\_GL est une évolution graphique du programme signaux complexes.

Signaux\_complexes\_GL a été écrit en Delphi7 et signaux complexes a été écrit en C++ (avec codeblocs). Ils utilisent le même moteur logique de localisation des trains sur le réseau à partir des évènements des détecteurs de zone.

Ces deux programmes utilisent le même fichier de configuration (config.cfg) qui sont donc interchangeables. Le programme signaux\_complexes\_GL utilise un deuxième fichier de configuration (client\_gl.cfg) qui contient des paramètres qui lui sont spécifiques. Il est décrit en fin de cette notice.

Signaux\_complexes\_GL a deux modes de fonctionnement :

Il peut être client de CDM rail pour piloter les signaux complexes. Il peut être exécuté sur le même ordinateur que CDM rail, ou un ordinateur différent. Dans ce cas ces deux ordinateurs seront reliés via réseau, et il est nécessaire de renseigner l'adresse IP V4 du PC exécutant CDM rail dans le fichier client\_gl.cfg.

Signaux\_complexes\_GL peut être autonome. Dans ce cas il n'a pas besoin de CDM rail pour piloter les accessoires du réseau. Cette fonctionnalité est valide uniquement avec les centrales LENZ et compatibles. Dans ce cas, Signaux\_complexes\_GL doit être relié à la centrale par le bus XpressNet (USB ou Ethernet). Ce mode permet de piloter les locomotives en mode « raquette », et le programme se charge de piloter les signaux complexes. En aucun cas il ne gère la sécurité des convois.

Il existe un mode hybride, qui permet d'utiliser CDM et le programme Signaux\_complexes\_GL de façon autonome. Dans ce cas, il faut mettre 0 dans la variable « Adresse IP du PC exécutant CDM rail ».

## Installation

Il est indispensable de procéder à l'installation du logiciel. Pour cela, faire un clic droit sur le fichier Install.bat ou Install2.bat et sélectionner « Exécuter en tant qu'administrateur ». Il est possible qu'en cas de présence d'un antivirus sur le PC, il refuse de lancer Install.bat. Dans ce cas lancer « installateur.exe » toujours en mode administrateur.

L'installation va installer un composant nécessaire à la communication USB avec la centrale ainsi que ses autorisations dans le registre. Signaux\_complexes\_GL ne pourra pas se lancer si ce composant (mscomm32.ocx) n'est pas installé.

## **Exclusion de sécurité**

Il est possible que Windows defender détecte par abus un trojan dans Signaux\_complexes\_GL.exe. Dans ce cas il faut ajouter une exclusion de sécurité en suivant le procédé décrit à ce lien :

<https://support.microsoft.com/fr-fr/help/4028485/windows-10-add-an-exclusion-to-windows-security>

Vous pouvez toujours faire scanner le fichier pour le vérifier sur un site antivirus comme *Kaspersky* par exemple :

<https://virusdesk.kaspersky.fr/>

## Fonctionnement

Les signaux gérés de base par CDM rail sont de trois types : carré violet, canton 3 feux ou 4 feux. Néanmoins, CDM rail gère ces feux de la même façon. Pour pouvoir utiliser les feux complexes, il faut utiliser un décodeur externe, piloté par le programme. Ce programme client communique avec le programme serveur (CDM Rail).

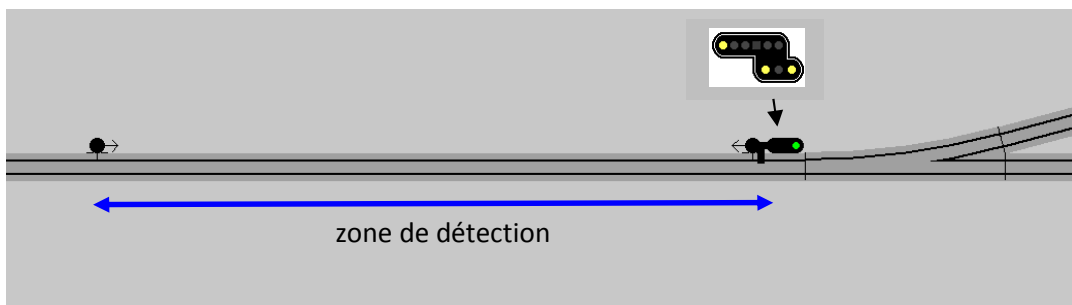
Les décodeurs des signaux complexes étant pilotés par le même bus que les aiguillages, le plan d'adresse des signaux complexes ne doit pas interférer avec les adresses des aiguillages ou des actionneurs.

Les signaux complexes gérés par le programme client n'ont rien à voir avec les signaux déclarés dans le réseau dans CDM rail. La représentation de la signalisation étant obligatoire sous CDM rail elle ne correspond pas forcément à des signaux physiques installés ; CDM rail ne pilotant pas la signalisation complexe (étendue).

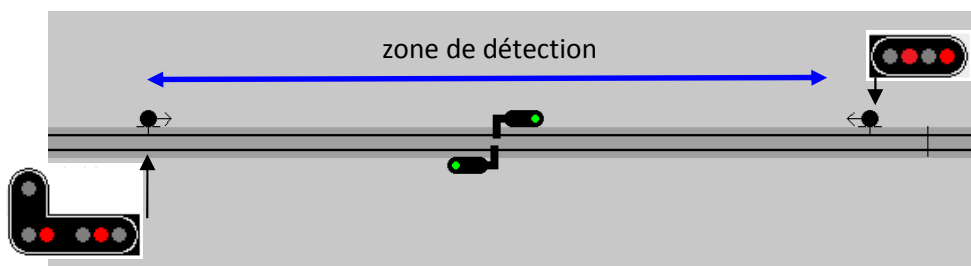
Les signaux complexes du programmes client sont implantables « librement » conformément aux règles de la signalisation française. Ils ne sont pas liés aux signaux de CDM rail. Cela signifie que l'on aura d'une part les signaux implantés dans CDM rail et d'autre part les signaux implantés dans le programme client qui représentent les signaux complexes sur le réseau. A un endroit particulier du réseau on pourra donc avoir un signal CDM et un signal complexe physiquement implanté sur le réseau.

En général on installe les signaux complexes avant un aiguillage ou un grill pour annoncer le rappel de ralentissement ou le carré (et dans ce cas le signal doit être précédent d'un signal ralentissement).

Exemple :



Un signal complexe est toujours associé à la fin d'une zone de détection (comme ci-dessus), ou ci-dessous dans le cas d'une zone de détection de pleine voie :

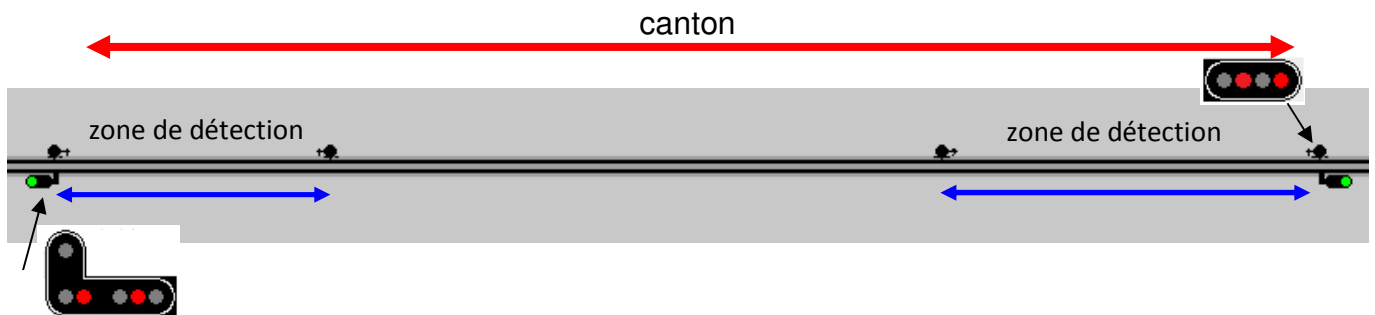


Le signal complexe est bien sûr sensible au sens de circulation de la voie sur lequel il est implanté.

## Différences entre cantons et zones de détections

Une zone de détection est un équipement électrique qui permet la détection d'un train sur le réseau. Un canton peut être constitué de plusieurs zones de détections. Un canton est délimité par des signaux (dits signaux de cantonnement). Deux signaux sont donc les limites d'un canton dans lequel un seul train peut être présent. CDM rail conseille de mettre deux zones de détection par canton en regard de chaque signal d'extrémité, mais CDM rail fonctionne aussi avec une seule zone par canton.

Implantation avec deux zones de détection par canton :

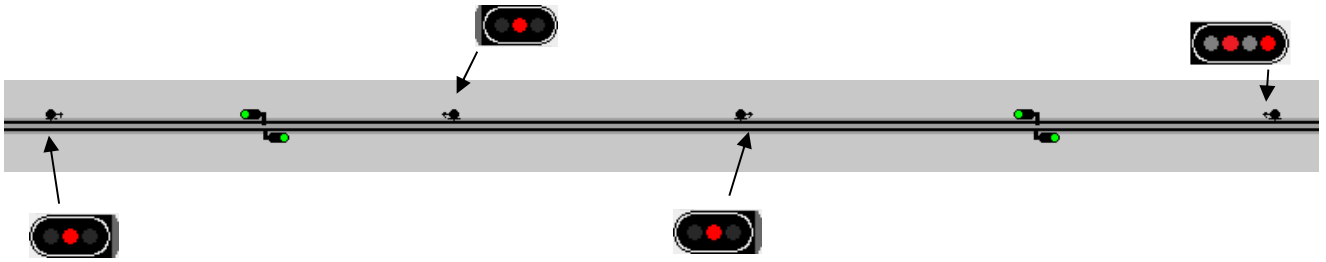


Les signaux complexes sont implantés en limite de canton.

## Signaux complexes virtuels et réels

Il n'est pas obligatoire *d'implanter* un signal complexe si l'on n'en dispose pas (par exemple pour des questions de coûts). Néanmoins pour des raisons de fonctionnement, il faut implanter à la place un signal virtuel (il possède une adresse sur le réseau mais ne sera pas piloté). Ceci permet par exemple de piloter un signal de ralentissement malgré que l'on ne soit pas en possession de son homologue, le signal de rappel de ralentissement. En effet le programme signaux\_complexes va tester le signal suivant pour connaître son état. Il est donc nécessaire de gérer un signal virtuel. Un signal virtuel a un décodeur égal à 0 dans le fichier de configuration.

Implantation avec une zone de détection par canton sur des détecteurs de pleine voie :



Pour les signaux complexes, le canton est encadré de deux signaux (cas d'une voie banalisée pour la circulation dans les deux sens) ou d'un seul signal si la circulation se fait toujours dans le même sens. Les signaux complexes seront implantés sur les fins de zone de détection dans le sens de la circulation, car le changement des signaux complexes se fait sur la retombée de la zone de détection, dans le sens de circulation (front descendant)

Le programme client pilote les signaux complexes du réseau. Il doit donc connaître le réseau. Pour cela, il est nécessaire de renseigner le fichier **config.cfg** qui se trouve dans les fichiers du répertoire du programme de signaux complexes. Il s'agit d'un fichier ASCII modifiable par n'importe quel éditeur de texte (notepad ...)

## Restrictions et spécificités du programme client pour les signaux

En utilisation avec CDM rail en mode RUN sans itinéraire, les signaux seront positionnés en fonction des aiguillages. C'est à l'opérateur de manœuvrer les aiguillages suffisamment à l'avance avant le passage du train pour que la présentation des signaux soit cohérente.

En utilisation avec CDM rail en mode RUN avec itinéraire(s), les aiguillages sont positionnés 1 canton avant le passage du train, ce qui est trop tard pour l'affichage d'un signal de ralentissement.

Pour une rapidité d'exécution optimale, dans CDM, il faut absolument dévalider les deux options de création des logs dans les menu « Comm IP/créer un fichier de log » et « Autoriser le log des événements de service » (avant de lancer le serveur de l'interface).

En standard, le programme client ne gère pas encore les signaux directionnels.

Le programme signal client gère les signaux violet, blanc, vert, rouge, carré, sémaphore, jaune, jaune clignotant, ralentissement 30 ou 60 et rappel de ralentissement 30 ou 60.

Pour afficher les signaux vert clignotant, rouge clignotant et blanc clignotant, il faut les programmer spécifiquement le programme par Delphi 7.

## **Modélisation du réseau**

La modélisation du réseau est sa description dans le fichier config.cfg. Ce fichier de configuration est identique au programme client « signaux\_complexes

La documentation pour configurer le fichier config.cfg se trouve dans la notice d'utilisation de « signaux complexes ».



## Fichier de configuration client-GL.cfg

Il s'agit d'un deuxième fichier de configuration qui est spécifique à signaux\_complexes\_GL  
Il contient des variables permettant son fonctionnement.

Les variables importantes sont les suivantes :

```
/ Adresse IP V4 du PC sur lequel s'exécute CDM : port  
127.0.0.1:9999
```

Il s'agit du PC sur lequel CDM fonctionne. Signaux\_complexes\_GL utilise un socket (adresse ip + port) pour communiquer avec le serveur de CDM rail. Les deux programmes peuvent donc être sur deux PC différents.

L'adresse 127.0.0.1 représente le PC sur lequel le programme Signaux\_complexes\_GL s'exécute, lorsque les deux programmes s'exécutent sur le même PC, et c'est donc cette adresse ip qu'il faudra utiliser.

Le port doit être le même que celui de CDM rail (9999 par défaut ; modifiable dans CDM rail dans le menu Comm.IP / Paramétrage de la liaison IP. Signaux\_complexes\_gl ne gère pas de nom d'hôte.

---

Uniquement pour les centrales LENZ ou compatibles :

```
/ Adresse IP V4 de l'interface LI-USB Ethernet : port  
/ par défaut le port est 5550  
/ ne pas connecter le port ou mettre 0 si on travaille avec l'interface USB  
192.168.1.23:5550
```

Si l'on veut utiliser Signaux\_complexes\_GL de façon autonome, il faut indiquer par quel moyen il va se connecter à l'interface LENZ : soit par une liaison réseau (wifi ou Ethernet) ou par une liaison USB. Cette ligne contient l'adresse IP et le port de l'interface Lenz LI-USB. L'adresse IP montrée ci-dessus (192.168.1.23) correspond à une interface Lenz\_USB-ETH qui a été configurée avec cette adresse évidemment.

---

Uniquement pour les centrales LENZ ou compatibles :

```
/ port COM de l'adresse USB de l'interface Lenz  
/ attention de 1 à 9 - Si le port de l'interface USB >9, il faut le changer  
/ manuellement dans le gestionnaire des périphériques  
/ mettre 0 si inutilisée ou si CDM est utilisé en parallèle de ce programme (car CDM utilise  
la liaison USB)  
/ Le programme ne tentera pas de se connecter à la centrale si CDM rail est détecté  
3
```

Le deuxième moyen d'utiliser Signaux\_complexes\_GL de façon autonome est d'utiliser le port USB pour qu'il se connecte à l'interface LENZ. Cette ligne contient le numéro de port USB de l'interface LENZ. Ce numéro va de 1 à 9. 0 représente une liaison inutilisée.

```
/ Avec (1) ou sans (0) initialisation des aiguillages au démarrage selon le tableau ci après  
0
```

En mode autonome, cette variable à 1 permet de lancer ou non (variable à 0) une séquence de positionnement des aiguillages au démarrage selon la description qui suit.

```

/*****
/ fichier de configuration de client-GL.log
/ gily - fliwq - 2018
/*****
/ ce programme permet de commander les signaux complexes avec ou sans
/ CDM rail. Il a deux modes de fonctionnement:
/
/ Mode 1 = sans CDM rail. Le programme ne gère que la centrale LENZ ou compatible.
/ Il faut connecter le PC en USB ou par ethernet à la centrale (nécessite l'interface LI100
ETH)
/
/ Mode 2.1 = Avec CDM rail, ce programme doit être connecté au réseau par l'interface
/ ethernet ; CDM est connecté par USB, et dans ce cas,
/ mettre 0 dans le numéro de port COM (inutilisé) ainsi que la variable avec ou sans
initialisation
/ des aiguillages à 0.
/
/ Mode 2.2 = Avec CDM Rail, ce programme connecté au réseau par l'interface Ethernet et CDM
/ Rail est connecté au réseau par l'USB.
/
/
/ Taille de la fonte de la fenêtre
16
/
/ Adresse IP V4 du PC sur lequel s'exécute CDM : port
127.0.0.1:9999
/
/ Adresse IP V4 de l'interface LI-USB Ethernet : port
/ par défaut le port est 5550
/ ne pas connecter le port ou mettre 0 si on travaille avec l'interface USB
192.168.1.23:5550
/
/ port COM de l'adresse USB de l'interface LI100
/ attention de 1 à 9 - Si le port de l'interface USB LI100 >9, il faut le changer
/ manuellement dans le gestionnaire des périphériques
/ mettre 0 si inutilisée ou si CDM est utilisé en parallèle de ce programme (car CDM utilise
la liaison USB)
/ Le programme ne tentera pas de se connecter à la centrale si CDM rail est détecté
/
3
/
/ Avec (1) ou sans (0) initialisation des aiguillages au démarrage selon le tableau ci après
0
/
/ Valeurs des initialisations des aiguillages au démarrage -----
/ @ aiguillage, position 1=dévié 2=droit, temporisation de manœuvre en 1/10eme de seconde ,
/ 0=pilotage normal 1=pilotage inversé
/ une position différente de 1 ou 2 sera à 1. Une temporisation de 0 sera à 5.
/ la liste doit être terminée par 0,0,0
1,2,3,0
2,2,3,0
(...)

```



signal de ralentissement sur un réseau piloté par CDM Rail + programme client. Il annonce un aiguillage distant pris en pointe dévié à franchir à 30 km/h.  
Ce signal sera suivi d'un signal de rappel de ralentissement 30 km/h (deux feux jaunes verticaux) placé avant l'aiguille.



panneaux directionnels sur signaux complexes annonçant la direction que prendra le train.

## Utilisation du programme signaux\_complexes\_GL avec CDM rail

- Lancer **CDM rail**
- Lancer le serveur d'interface (**interface/démarrer un serveur**)
- Lancer le serveur IP (**comm IP/démarrer le serveur comm IP**).
- Lancer le programme client signaux\_complexes\_GL.

Il va alors afficher l'état des connexions :

### Exemple 1 :

Erreur 10065 socket IP Lenz: Port non connecté

Erreur 10061 socket IP CDM Rail: Connexion refusée

Le programme signaux\_complexes\_GL ne s'est pas connecté l'interface LENZ par réseau.

Le programme signaux\_complexes\_GL ne s'est pas connecté à CDM rail (CDM est pas lancé ou son serveur)

### Exemple 2 :

Erreur 10065 socket IP Lenz: Port non connecté

CDM Rail connecté avec l'ID XX

Le programme signaux\_complexes\_GL ne s'est pas connecté l'interface LENZ par réseau.

Le programme signaux\_complexes\_GL s'est connecté à CDM rail

Pour relancer une connexion à CDM rail, il faut le reconnecter en sélection le menu « interfaces / connecter CDM rail »

Dans CDM rail, passer en mode RUN avec ou sans trains (TCO).

Au lancement du mode TCO ou du RUN, CDM rail positionne les aiguillages. Ces positions sont alors récupérées par le programme client. (ainsi que chaque changement d'aiguillage par la suite).

Il faut un passage sur deux détecteurs consécutifs pour synchroniser un train avec le programme.

L'affichage dans le programme client donne l'état des pilotages des signaux et l'état des mémoires de zone. Il est possible de sélectionner l'affichage de l'un ou de l'autre dans le menu Debug et « afficher changement des détecteurs » (menus à bascule).

La fenêtre debug peut être fermée à tout instant et rappelée par le bouton « affiche debug »



## **Arrêt des logiciels**

Pour l'arrêt des logiciels, procéder dans l'ordre :

1. Arrêter le mode Run,
2. Arrêter le serveur Comm IP sur CDM Rail, ou fermer directement CDM rail
3. Fermer le programme client.

## Modification du programme avec Delphi 7

Signaux complexes a été écrit en Delphi7. Il est téléchargeable gratuitement ici (après avoir créé un compte)

<https://delphi.developpez.com/telecharger-gratuit/delphi7-perso/>

la clé est la suivante

Numéro : YH?Z-2WDEGK-S48529-3AS3

Clé : G5N-D95

Il reste à configurer Delphi pour y intégrer les composants additionnels Sockets et TmsComm.

### Installation des composants socket (TClientSocket et TServerSocket)

Menu composants / installer les paquets, cliquer sur ajouter

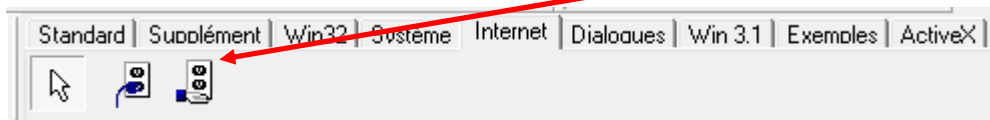
bouton ajouter un paquet de conception

naviguer dans c:\programmes\borland\Delphi7\bin

et choisir le fichier dclsockets70.bpl

cliquer sur OK

Les composant ClientSocket et  
ServeurSocket apparaissent  
dans l'onglet Internet



### Installation du composant MScomm32

Menu composants / importer un contrôle activeX / cliquer sur ajouter

chercher le fichier mscomm32.ocx (qui est dans le répertoire signaux\_complexes\_gl)

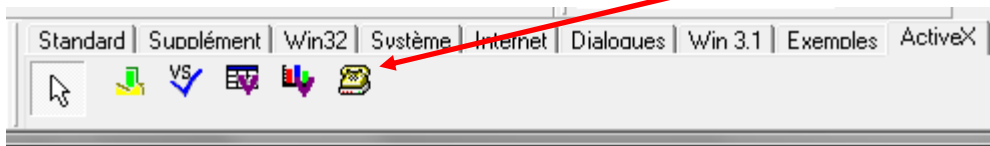
cliquer sur ok

Dans la liste, sélectionner « Microsoft Comm Control 6.0 (Version 1.1)

Cliquer sur installer puis sur cliquer unité

Sauvegarder

Le composant TMSComm  
apparaît dans l'onglet ActiveX



## Défaillances possibles

En cas de « défaillance ouverture fiche à l'ouverture du source du programme » sous Delphi:

Fermer Delphi

Editer en ascii le fichier UnitPrinc.dfm

Supprimer les lignes concernant le composant MSCommRelais:TMSComm (voir ci dessous)  
Sauvegarder

Réouvrir Delphi et le projet paquet météo

1. faire glisser le composant TMScomm dans la fiche FormPrinc

Si une erreur apparait (Tmscomm existe déjà), il faut supprimer la ligne de déclaration du composant TMSCOM dans le fichier (UnitPrinc) (MSComm1: TMSComm; )

2. renommer le composant en MSComm1

3. Affecter la procédure d'interruption MSComm1Comm dans le champ onComm du composant MSComm.(y mettre MSComm1Comm)

Fini

---

Lignes à supprimer dans UnitPrinc.dfm:

```
object MSCommxxx: TMSComm
  Left = 440
  Top = 86
  Width = 32
  Height = 32
  OnComm = MSCommRelaisComm
  ControlData = {
    2143341208000000ED030000ED03000001568A64000006000000010000040000
    000200008025000000000800000000000000000003F00000011000000}
end
```

---

En cas d'exception « classe non enregistrée » à l'exécution de signaux\_complexes\_GL :  
il faut enregistrer mscomm32.ocx dans le registre avec la commande :

```
regssvr32 mscom32.ocx
```



L'environnement de travail sous Delphi une fois tout installé, avec le projet ouvert :

