Réaliser un cadran de TSF avec Galva

Qui a dit : mais c'est presque simple ?

Introduction

Lors de la restauration d'un vieux poste de radio, le cadran est parfois en mauvais état. Lorsqu'il est impossible de le « récupérer », il faut alors envisager d'en recréer un nouveau, et pour rester dans l'esprit de la restauration, le souhait est souvent que le graphisme soit aussi identique que possible à l'original. Cet article traite de manière détaillée la réalisation du graphisme, sans aborder le transfert du graphisme sur un support identique ou proche de l'original.

Il existe un très grand nombre de cadrans, toutefois, il est possible de distinguer trois principaux types de graphismes à réaliser :

1) des échelles de longueurs d'ondes ou de fréquences impliquant de dessiner des graduations sur des segments de droites ou sur des arc de cercles ;

2) l'écriture de noms de stations à des positions bien définies et le marquage des zones de fréquences où ces stations sont reçues ;

3) des dessins, au sens artistique du terme, des logos ou des courbes non géométriques, avec éventuellement des graduations.

La part de chacun de ces types de graphismes est fonction du cadran à réaliser.

Les Fig. 1 à 5 donnent quelques exemples types de cadrans trouvés sur Internet.

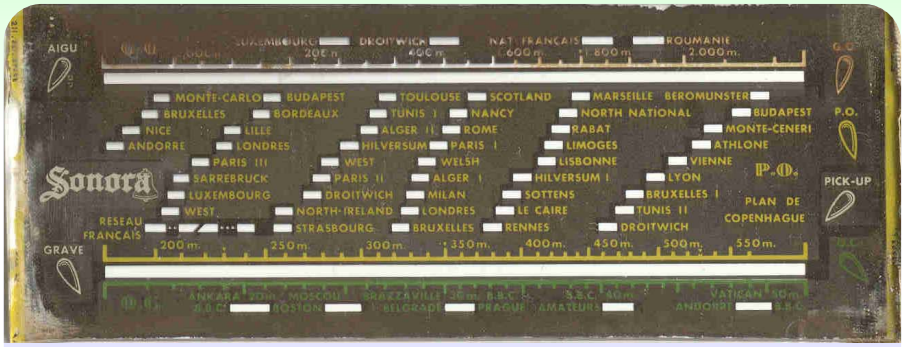

Fig. 1 Fichier Sonora.jpg

Fig. 2 Fichier Dscn3742.jpg

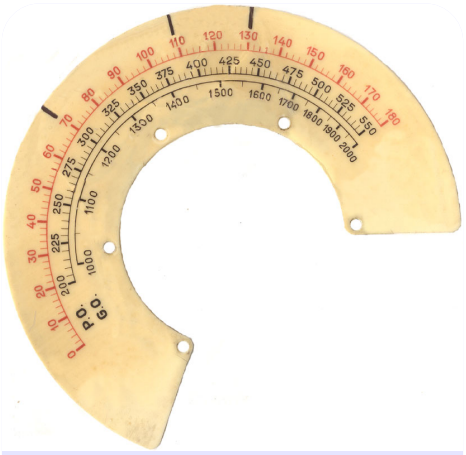

Fig. 4 Fichier Cadran_recadre.jpg

long. En effet, quel que soit le logiciel utilisé pour ce travail, placer les graduations, les noms de toutes les stations et leur position est plutôt un travail de Bénédictin, qui nécessite de la patience et de la persévérance, car il ne peut pas être automatisé. Aussi, il est important de procéder par étapes et avec méthode, pour ne pas avoir à tout

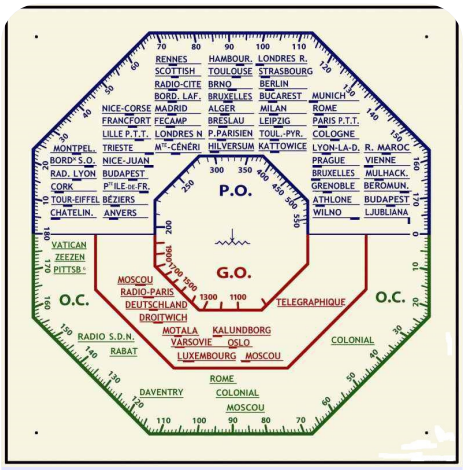

Fig. 3 Fichier CadranRx.jpg

Fig. 5 Fichier SONORA EXCEL.jpg

Réalisation d'un cadran

Afin d'arriver à avoir une reproduction fidèle, il est vivement conseillé d'effectuer le dessin du nouveau cadran avec l'image du cadran original en arrière plan : on dessine par-dessus l'image qu'on supprime à la fin, et on dessine avec une couleur bien visible sur le fond. Ce n'est qu'en fin de travail, que cette couleur est remplacée par les couleurs appropriées.

Reproduire un cadran à l'identique est à priori un travail assez

recommencer, alors qu'on pensait arriver à la fin du travail.

Tout d'abord, il est important de disposer d'une image du cadran original de la meilleure qualité possible, bien « horizontale» et déformée le moins possible. Un trait tracé horizontalement par dessus doit être le plus parallèle possible à un trait horizontal de l'image et le rapport hauteur/largeur respecté. Ce dernier point ne pose en général pas de problème lorsque l'image est obtenue avec un scanner à plat, par contre, avec un appareil photo, il faut placer celui-ci bien perpendiculairement au centre du cadran et le plus loin possible du cadran pour limiter les déformations. Cette image doit ensuite être placée en arrière plan, et il est conseillé de respecter exactement les dimensions de l'original, afin de ne pas avoir à faire, en fin de travail, une impression avec zoom pour obtenir les bonnes dimensions.

Présentation rapide de Galva

Galva est un programme qui a initialement été spécialement étudié pour dessiner avec précision des cadrans de galvanomètres, de potentiomètres, de CVs, etc. Il est donc particulièrement bien adapté pour la réalisation d'échelles de toutes sortes, dont les formes peuvent êtres courbes ou droites, et les graduations linéaires, fonctions d'une puissance, logarithmiques, spécifiques, manuelle, etc.

Le programme tourne sous Windows, gère la souris, les raccourcis clavier, les couleurs, permet l'insertion de courbes de Bézier, de tracés à la souris et d'images et est utilisable en Français, Anglais, Allemand et Espagnol.

La première version de Galva a été disponible en janvier 2004. Depuis, plusieurs versions se sont succédées et ce programme est, aujourd'hui, aussi très bien adapté pour beaucoup d'applications totalement différentes …

Pour se faire une idée de ses possibilités, le mieux et d'aller voir la présentation sur le site f5bu.fr, sous l'onglet Galva / Présentation, où de nombreux fichiers pdf présentent des réalisations concrètes.

Remarques

Il s'agit d'un interpréteur de commandes, incluant des commandes graphiques, non d'un logiciel pour tout dessiner à l'aide de la souris.

Comme pour beaucoup de logiciels, les possibilités sont tellement nombreuses, qu'un minimum d'apprentissage est recommandé, voir nécessaire. Pour cela il y a un tutoriel, une aide en ligne, des masques de saisies, une aide contextuelle et de très nombreux fichiers exemples sont fournis.

Chercher à effectuer des tracés sans un minimum de connaissances du programme, ne permet pas d'effectuer les bons choix et est, au mieux, source de perte de temps, au pire, source d'abandon. Il est donc recommandé de lire au moins l'introduction de l'aide, de survoler la description des différentes commandes et de faire quelques essais dans le seul but de comprendre la « philosophie » de fonctionnement du programme avant de se lancer dans un vrai projet.

Description rapide

Galva est un interpréteur de commandes, c'est-à-dire qu'il faut écrire une sorte de programme qui décrit votre graphique. Pour cela vous disposez de différentes commandes ou instructions.

Par exemple,
Cercle = 20, 30, 10 va tracer un cercle dont le centre a les coordonnées 20, 30 (le centre est 20 mm à droite et 30 mm au-dessus du point de référence, qui par défaut est le coin inférieur gauche de l'écran ou de la feuille de papier) et de rayon 10 mm.

Autre exemple :
Texte = 20, 40, bleu, GC,Mon Texte va écrire « Mon Texte » en bleu, Gras et Centré en 20, 40.

Écrire des commandes avec un nombre parfois important de paramètres ne vous emballe pas ! Alors, pas de panique, les masques de saisies, mis en place avec la version 2.50, facilitent grandement la saisie des commandes, en donnant une description de la commande et de chaque paramètre et en évitant de connaître l'ordre des paramètres. Un exemple est donné Fig. 8, et un autre sur le site, sous Galva / Téléchargement.

Comme pour toute programmation, cela demande de la logique et un peu de persévérance. Avoir déjà programmé, quel que soit le langage qui a été utilisé, est évidemment un avantage, mais commencer avec Galva peut aussi être une manière de se mettre en « douceur » à la programmation, car le résultat est visuel et on ne risque rien de faire des essais, au contraire.

Il s'agit d'une programmation séquentielle, c'est-à-dire que les commandes sont exécutées dans l'ordre où elles apparaissent dans le code (programme). Si une commande trace un grand rectangle blanc, tout ce qui a été tracé auparavant à cet endroit sera caché par lui. Les commandes Stop et ListeVar permettent de déceler facilement ce genre d'erreur. De plus, en mettant une apostrophe « ' » devant une ligne de commande on la transforme en commentaire, ce qui permet de constater facilement son effet.

Réaliser un cadran avec Galva

Galva est très bien adapté pour réaliser toutes sortes de graduations sur des formes géométriques (point 1) et marquer des noms de postes (point 2). Depuis la version 2.52 il est également possible d'insérer des courbes de Bézier et des dessins réalisés avec la souris ce qui donne accès à quelques réalisations « artistiques » (point 3, avec un exemple donné par les fig. 5, 11 et 12). De plus, il sait parfaitement intégrer des images, qu'elles soient scannées ou créées avec un logiciel graphique et créer aussi manuellement toutes sortes de formes non géométriques par petits segments de droites à l'aide de la souris.

Réalisation de graduations

Il s'agit là de l'application de base de Galva et cela ne devrait en général poser aucun problème, même si, la plupart du temps, il faut placer les graduations une à une « manuellement ». La réalisation d'échelles graduées en hexagone, comme celles de la Fig. 3 n'étant pas prévue nativement dans Galva, il fallait écrire un petit programme avant la sortie de la

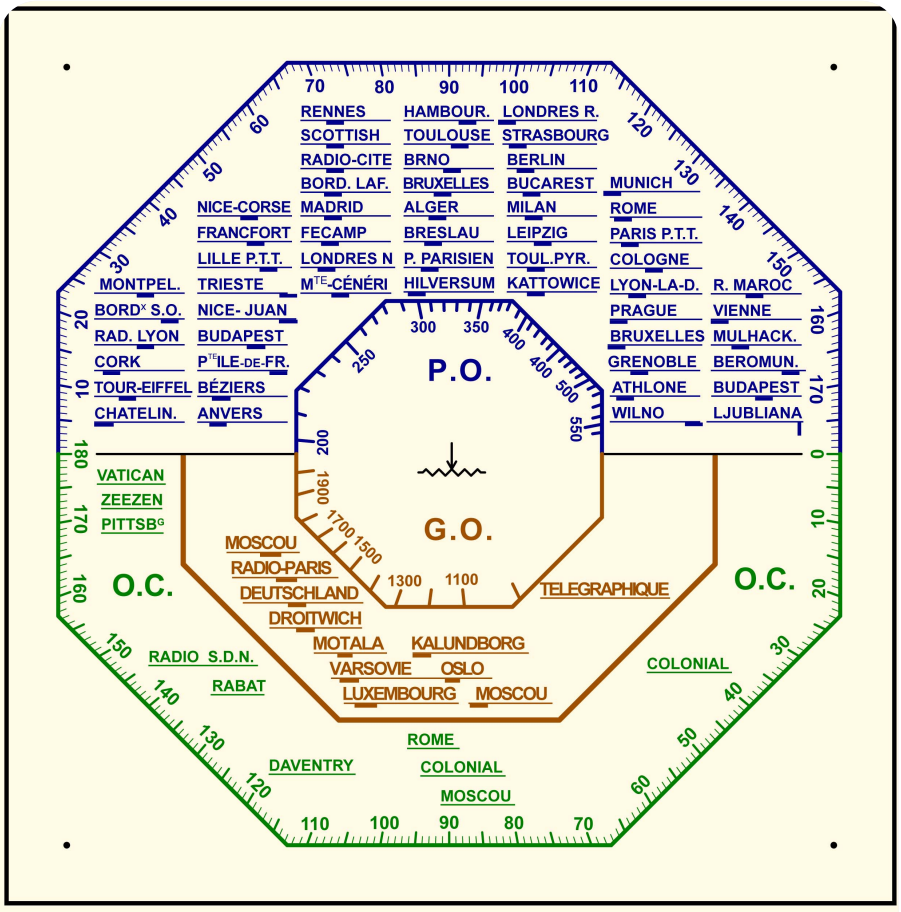

Fig. 6 CadranRx version Galva

version 2.50, qui intègre deux commandes externes spécialement réalisées pour cette fonction. La Fig. 6 donne un exemple de réalisation d'un tel cadran. Ce fichier est disponible au format pdf dans la présentation sur le site, et il fait parti, ainsi que le code source qui le génère, des exemples fournis avec le programme.

Pour l'exemple de la Fig. 1, la Fig. 7 montre une partie du cadran, alors que l'écriture du code est en cours. Voyons un peu comment on en arrive là. Tout d'abord, il faut savoir que, comme presque toujours en programmation, il y a plusieurs manières de faire, et c'est selon les préférences de chacun.

Pour commencer, il faut placer l'image en arrière plan. Il faut pour cela qu'elle apparaisse en premier dans le code pour que les autres tracés viennent par-dessus. Voici à quoi peuvent ressembler les premières lignes de code :

```
Image = Fig1_SONORA.jpg,0,0,1
Couleur = Magenta
CT = 45.4,45.5,n
RefCur = %CTx,%CTy
CadreP = 0,-2.5,270.7,-7
```

Avec le curseur sur la ligne Image =, un clic sur la touche F2 ouvre le masque de saisie de cette commande. Ce masque donne, comme il le ferait pour toutes autres commandes la description de la commande, et dans l'ordre, le détail de tous les paramètres. Ainsi, sur la Fig. 8, on voit que « Fig1_SONORA.jpg » est le nom du fichier graphique, que l'image est placée en 0,0, c'est-à-dire dans le coin inférieur gauche et que le coefficient de taille est de 1 (Ne disposant que de l'image récupérée sur Internet, il m'est impossible de la mettre aux mêmes dimensions que le cadran d'origine, mais c'est sur ce coefficient qu'il faut jouer finement pour y arriver avec précision).

La commande Couleur = définit, pour le temps de la mise au point, la couleur magenta par défaut, pour bien distinguer les nouveaux tracés sur l'image.

La commande CT = (Centre Temporaire) définit un nouveau point de référence par défaut, c'est-à-dire ayant les coordonnées 0,0 pour la suite. Ce point a été choisi à l'origine de l'échelle des longueurs d'ondes pour les P.O., à côté de « GRAVE ». On comprendra pourquoi plus tard.

Par défaut, les coordonnées affichées de la souris (Fig. 9), sont données par rapport au point inférieur gauche du graphique. Aussi, pour déterminer les valeurs xx et yy pour cette commande, il suffit de placer le curseur de la souris au point choisi et de copier les coordonnées de la souris dans le code. Il est aussi possible d'écrire CT = puis de faire un Ctrl-double clic de la souris au point choisi et les coordonnées de ce point viennent s'inscrire dans le code. Pour vérifier qu'elles sont parfaites, ou éventuellement les corriger un peu, le paramètre « n », qui permet de ne pas tracer un signe « + » à cet endroit, a d'abord été omis. Attention, ne pas oublier de faire F4

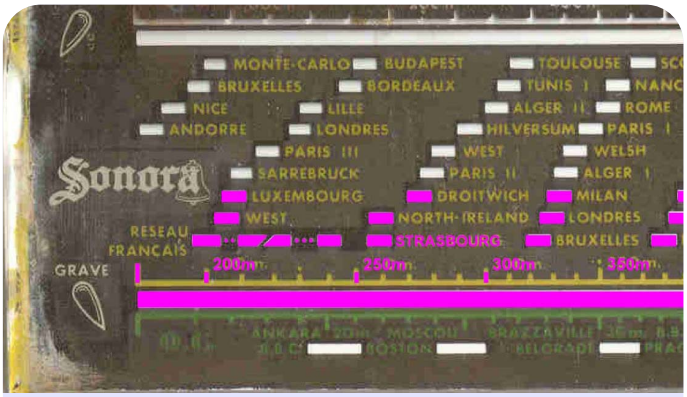

Fig. 7 Vue partielle de la version de travail

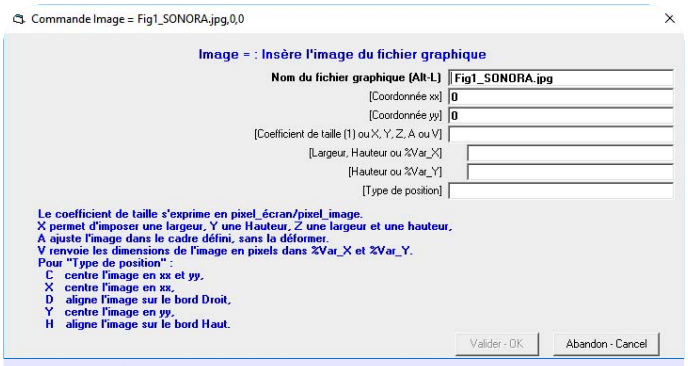

Fig. 8 Masque de saisie de la commande "Image"

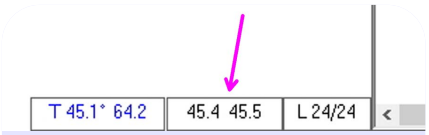

Fig. 9 De gauche à droite, les coordonnées polaires (T pour trigonométriques, A pour celles des commandes Arc), les coordonnées cartésiennes et le numéro de ligne du curseur dans le code par rapport au nombre total de lignes.

pour actualiser le graphique. Un zoom (F7) important (≥ 2) est à utiliser pour obtenir une bonne précision (car la précision des écrans est bien moindre que celle des imprimantes). Dans l'exemple, ce nouveau point se trouve 45.4 mm à droite et 45.5 mm au dessus du point inférieur gauche.

Comme nous venons de déplacer l'origine des coordonnées, il est pratique que l'affichage des coordonnées de la souris soient conformes, plutôt que défini par rapport au coin inférieur gauche. C'est l'objet de la commande RefCur = qui défini les coordonnées du point de références pour l'affichage des coordonnées de la souris. Ainsi, les coordonnées affichées de la souris sont à nouveau celles à utiliser dans la suite du code, et on peut les recopier ou utiliser des Ctrl-double clics de la souris pour entrer directement les coordonnées d'un point. Toujours travailler avec un zoom important et si besoin corriger un peu les valeurs des coordonnées par « essais - erreurs ». A noter qu'un double clic de la souris dans le graphique centre le point sur l'écran. En cas de besoin, Ctrl-F6 repositionne le graphique à l'écran et Ctrl-F8 annule l'effet de la commande RefCur = .

Enfin, la commande CadreP = trace le Cadre Plein qui se trouve sous l'échelle des longueurs d'ondes (c'est là qu'on se rend compte que l'image n'est pas parfaitement horizontale !).

En ce qui me concerne, j'ai d'abord utilisé une commande Cadre = , qui ne dessine que le contour du cadre, et ce n'est qu'une fois satisfait du résultat que j'ai ajouté le « P » pour avoir un cadre plein (ceci est rappelé dans le masque de saisie).

Tout ceci est en fait bien plus rapide à faire qu'à décrire !

Venons en aux graduations. Il est possible dans chaque commande de graduation(s) de définir l'épaisseur de la ou des graduations, mais il est plus simple et efficace de définir une épaisseur par défaut lorsque la même doit être utilisée plusieurs fois. C'est ce que l'on fait avec les deux premières commandes ci-dessous. En fait, on pourrait simplement écrire EpaisT = 1.2 mm§C, pour définir une épaisseur par défaut de 1,2 mm et des extrémités de traits carrées (donc non arrondis, qui est la valeur par défaut), comme cela est indiqué dans le masque de saisie. Ici, la première ligne correspond à ce que l'on appelle une affectation de variable. On donne la valeur 1.2 à la variable %E (un nom de variable commence toujours par le signe %). L'intérêt d'une telle variable est qu'elle pourra être utilisée de nombreuses fois tout au long du programme et il suffira de changer sa valeur uniquement dans la ligne d'affectation pour que partout dans le programme se soit cette valeur qui soit prise en compte. Ici on l'utilise pour définir l'épaisseur des traits des graduations en remplaçant « 1.2 » par « %E » dans la ligne définissant l'épaisseur des traits.

Pour définir une échelle gradué dans Galva, il faut commencer par définir l'étendue de l'échelle. Il existe pour cela trois commandes : Arc = qui défini une échelle en arc de cercle ; Droite = qui défini une échelle droite horizontale et DroiteV = qui défini une échelle droite verticale. Ici, c'est donc la commande Droite = qu'il faut utiliser. Comme dit, cette commande défini l'étendue de l'échelle, après, il suffit habituellement d'utiliser des commandes indiquant le nombre de graduations et/ou de valeurs, pour que celles-ci soient automatiquement réparties sur l'ensemble de l'échelle. Ici, les graduations ne sont ni linéaires (c'est à dire non équidistantes), ni fonction d'une puissance, ni ..., il faut donc les placer une à une (généralement par essais-erreurs : on entre une valeur, on regarde si la position est bonne, on modifie la valeur, et ceci jusqu'à l'obtention de la bonne position, et ainsi de suite pour toutes les graduations). Aussi, on préfère définir une échelle de 0 à 100 qui fait que les indications de graduations en % de l'échelle correspondent directement à des mm depuis l'origine, ce qui est plus intéressant que des pourcentages de la longueur « quelconque » de l'échelle.

Pour placer des graduations « manuellement », on utilise des commandes Grad1 =. La première ci-dessous défini une graduation de 5,5 mm de long au point origine de l'échelle (à 0 mm de l'origine). On pourrait ainsi placer toutes les graduations avec chaque fois une commande par graduation, mais ceci donnerait un code à rallonge. Pour le faire plus facilement, on utilise une variable %A, qui reçoit la liste des valeurs, séparées par des virgules (le séparateur décimal dans le code de Galva est le point), et une seule commande Grad1 =. Ainsi, la deuxième commande Grad1 = trace les 3 premières « grandes » graduations (200m, 250m, 300m) d'une longueur de 3 mm, volontairement inférieure à la longueur des graduations d'origines, pour mieux distinguer les nouvelles graduations par dessus celles d'origines ; la bonne longueur sera définie ultérieurement) :

```
%E = 1.2
EpaisT = %E mm§C
Droite = 0,100,,n
Grad1 = 0,5.5
%A = 19.6,62.3,99.2
Grad1 = %A§%A,3
```

Pour déterminer les valeurs à mettre dans %A, on peut procéder par essais-erreurs pour chaque valeur, comme expliqué ci-dessus, toutefois, on gagne pas mal de temps en mettant successivement la souris à l'endroit de la graduation suivante à tracer, puis en copiant la valeur x de la coordonnée de la souris à la suite de la liste de valeurs de %A, puis en l'ajustant finement si besoin, jusqu'à l'obtention d'un résultat satisfaisant. Au final, la commande devient :

```
%A = 19.6,62.3,99.2,131.7,161,187,214,241.9
```

Toutes les « grandes » graduations sont alors tracées et il faut refaire la même chose pour les petites graduations (plus courtes), pour lesquelles le code sera, pour

des graduations de 1,5 mm de long, comme le montre si besoin le masque de saisie :

```
%A = 29.7,38.8,46.2,54.6,69.8,77.3, 84.4,91.5, _
105.8,111.8,118.3,125,137.9,143.8, 150.1,156,167.3,172.6,177.9,182.6, _
192.6,198.5,203.2,208.5,219.7,225.2, 230.1,235.8,249.1
Grad1 = %A§%A,1.5
```

Remarques :

● le nombre de petites graduations étant grand (je vous avais dit que c'était un travail de Bénédictin), la ligne de valeurs serait très longue, on utilise alors des lignes de continuations qui suivent les lignes se terminant par « _ » (espace souligné). Tout ceci est évidemment expliqué dans l'aide.

● comme cela est indiqué dans le masque de saisie et expliqué dans l'aide, le premier %A dans la commande Grad1 = est une formule fonction d'une variable qui est précisée par le deuxième %A. Je vous concède que ce n'est sans doute pas évident à comprendre à première vue, mais ce qu'il faut retenir, est que cette façon d'écrire permet de préciser à la commande d'utiliser l'une après l'autre toutes les valeurs contenues dans la variable %A (Pour ceux qui aiment comprendre, un autre exemple est peut-être plus parlant. Si on avait écrit : Grad1 = 2*Sin(%A)§%A,1.5 se sont deux fois le sinus de chaque valeur de %A qui auraient été utilisées).

Et ainsi de suite pour les autres graduations.

Mise en place des stations

Voyons pour les stations (j'en ai compté 48 + 9, je crois). Il faut pour chaque station écrire le nom et dessiner la position d'accord.

Tout d'abord, il faut trouver une police aussi proche que possible de celle d'origine, pour des questions esthétiques, mais aussi pour des questions de place. En ce qui me concerne, la police Arial correspondant relativement bien, je n'ai pas cherché plus loin pour la démonstration. Pour certains cadrans, il peut arriver qu'il faille resserrer ou écarter un peu les caractères de certaines stations. Ce n'est pas le cas ici, mais en cas de besoins, ceci est possible en utilisant la commande externe GCE_Crenage = à la place de commandes Texte = (c'est le cas pour le cadran de la Fig. 6).

Les stations étant présentées sur 9 lignes horizontales, on va à nouveau utiliser une variable pour ne pas avoir à répéter pour chaque station la valeur de y. Voici donc un exemple de code pour afficher Strasbourg (non, ce n'est pas parce que j'habite à Strasbourg, mais parce que c'est la première station en bas à gauche) :

```
%Y = 10.4
CadreP = 65.5,%Y,6.5r,3.1r,%CB
Texte = 5r,%Y,,*1.5 g,STRASBOURG
```

… et quelques explications. Comme dit, %Y = 10.4 défini la valeur de y pour les deux commandes suivantes, ainsi que pour toutes celles des stations de la même ligne du cadran. Si vous avez suivi jusque là, vous avez compris que pour trouver cette valeur il suffit de placer la souris sur la base du cadre d'accord et de relever la valeur y de la souris. « 65.5 » est la distance en mm entre l'origine de l'échelle et le bord gauche du cadre d'accord, relevé à l'aide de … la souris. « 6.5r » et « 3.1r » définissent les coordonnées de l'angle opposé du cadre, et plutôt que d'écrire « 65.5+6.5 » et « %Y+3.1 » on utilise la possibilité d'utiliser des coordonnées relatives par rapport aux dernières coordonnées utilisées en plaçant un « r » derrière chaque valeur. Cela permet de définir facilement la largeur et la hauteur des petits cadres afin qu'ils soient tous identiques. A remarquer que le choix a été fait ici de donner les mêmes dimensions à tous les petits cadres, alors qu'il y a une ou deux exceptions (volontaires ou non ?) sur le cadran d'origine.

« %CB » (Couleur Blanche) est le nom d'une variable qui doit contenir la couleur des zones d'accord pour les stations en P.O. La ligne d'affectation doit être placée avant sa première utilisation, et le temps des ajustements, on la met également à magenta.

Enfin, la commande Texte = écrit Strasbourg 1,5 fois plus grand que la taille par défaut et en gras.

Il est maintenant possible d'entrer les autres stations de la même ligne du cadran de la même manière, et celles des autres lignes en modifiant à chaque fois la valeur de %Y. Par exemple pour Bruxelles qui est sur la même ligne :

```
CadreP = 111,%Y,6.5r,3.1r,%CB
Texte = 5r,%Y,,*1.5 g,BRUXELLES
```

Il faudrait donc 48 x 2 + 9 lignes. Cela ne pose aucun problème. Toutefois, personnellement je préfère une autre solution qui se présente ainsi (seules les lignes 1 à 4 des stations sont traitées) :

```
%L = 1
E4:
If %L =1 then
 %Y = 10.4
 %T = 16,,51,,65.5,STRASBOURG, 111,BRUXELLES, _
 146.4,RENNES, 191,DROITWICH
ElseIf %L = 2 then
 %Y = 17
  %T = 22,WEST, 66,NORTH-IRELAND, 114.5,LONDRES, _
 151.1,LE CAIRE, 197.4,TUNIS II
ElseIf %L = 3 then
 %Y = 23.3
  %T = 24.2,LUXEMBOURG,76.8, DROITWICH,116.5,MILAN, _
 153.8,SOTTENS,201.3,BRUXELLES I
ElseIf %L = 4 then
 %Y = 29.8
    %T = 26.3,SARREBRUCK, 80.4,PARIS I I,118.6,ALGER I, _
 158,HILVERSUM I,209.5,LYON
'… ici, il faut ajouter les lignes pour %L= 5, 6, 7, 8 et 9
EndIf
%I =1
E5:
 Cadre = %T(%i),%Y,6.7r,3.1r,%CB
 %i==%i+1
 Texte = 4.5r,%Y,,*1.5 g,%T(%i)
 %i==%i+1
 If %i<NbP(%T) Goto E5
 %L==%L+1
 If %L=<9 Goto E4
```

Cela semble un peu compliqué lors de la première lecture, mais en relisant avec les explications, puis en relisant si besoin, ça finit par être … logique.

On utilise ici plusieurs puissantes possibilités de la programmation. Il y a ce que l'on appelle deux boucles : la première, la principale, celle où l'on passe le plus, trace les cadres et les noms des stations dont les données sont dans %T. Elle va de E5: jusqu'à If %i<NbP(%T) Goto E5. Cette dernière commande permet une modification conditionnelle du déroulement du programme et son sens est : Si %i<Nombre_de_va-

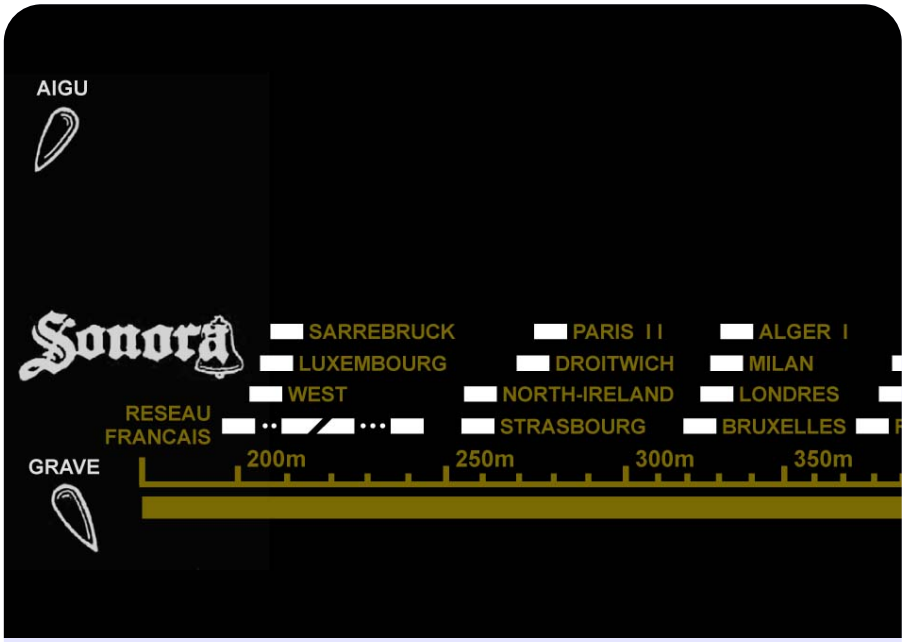

Fig. 10 Version finale incomplète

leurs_dans_%T Aller_à E5 (NbP() étant une fonction qui renvoie le nombre de paramètres contenu dans l'argument, c'est-à-dire ici dans %T).

La deuxième boucle va de E4: j'usqu'à la fin et traite successivement les différentes lignes de stations. La variable %L, qui représente le numéro de la ligne à traiter, est initialisée à 1 au début, puis incrémentée de 1 à l'avant dernière ligne à chaque passage dans la boucle. A noter que, comme précisé dans le masque de saisie, le double signe égal « == » permet d'affecter le résultat numérique de l'expression à une variable. De la 3e ligne jusqu'à EndIf, on utilise des commandes d'exécutions conditionnelles : les commandes « If … then », « Si … alors » en anglais, quasi universelles en programmation ; « ElseIf … then », « SinonSi … alors » ; « Else », « Sinon » et « EndIf », « FinDuSi » permettent l'exécution de lignes de code différentes selon la valeur de %L. Les variables %Y et %T reçoivent ainsi des valeurs en fonction de la valeur de %L, c'est-à-dire de la ligne à gérer.

Les commandes Cadre = et Texte = ressemblent à celles utilisées pour « STRASBOURG », mais la valeur de x et le nom de la station sont remplacés par ce que l'on appelle des variables indexées, c'est à dire qu'on prend la %ie valeur de %T. Exemple : si %L=2 et %i=5, alors %T(%i)=114.5 et si %i=6 %T(%i)=LONDRES.

Voici encore les lignes pour écrire « RESEAU FRANCAIS » et tracer les pointillés et la zone d'accord particulière, interrompue, juste à côté :

```
Texte = 14.1, 13.1,%C,*1.5gd,RESEAU%CRFRANCAIS
PolyP = %CB,28.6,10.4, 0r,3.1r, 8.r,0r, -3.1r,-3.1r
PolyP = %CB,43.5,10.4, 0r,3.1r, -4.9r,0r, -3.1r,-3.1r
CercleP = 25.2, 12,.5,,,%CB
CercleP = 27, 12,.5,,,%CB
CercleP = 45.3, 12,.5,,,%CB
CercleP = 47.3, 12,.5,,,%CB
CercleP = 49.3, 12,.5,,,%CB
```

L'écriture du texte ne pose pas de problème. Après la lettre g pour avoir le texte en gras, la lettre « d » demande un alignement du texte à droite et on utilise la variable interne %CR pour inséré un retour chariot - saut de ligne. On aurait aussi pu écrire une commande Texte = séparée pour chaque ligne de texte.

Pour tracer les deux trapèzes rectangles, on utilise deux commandes PolyP = (Polygone_Plein) et on défini les coordonnées des quatre angles des polygones. Ceux-ci peuvent être entrés par des doubles clics de la souris. Toutefois, pour avoir des formes bien symétriques seul les coordonnées

Fig. 11 Version de travail

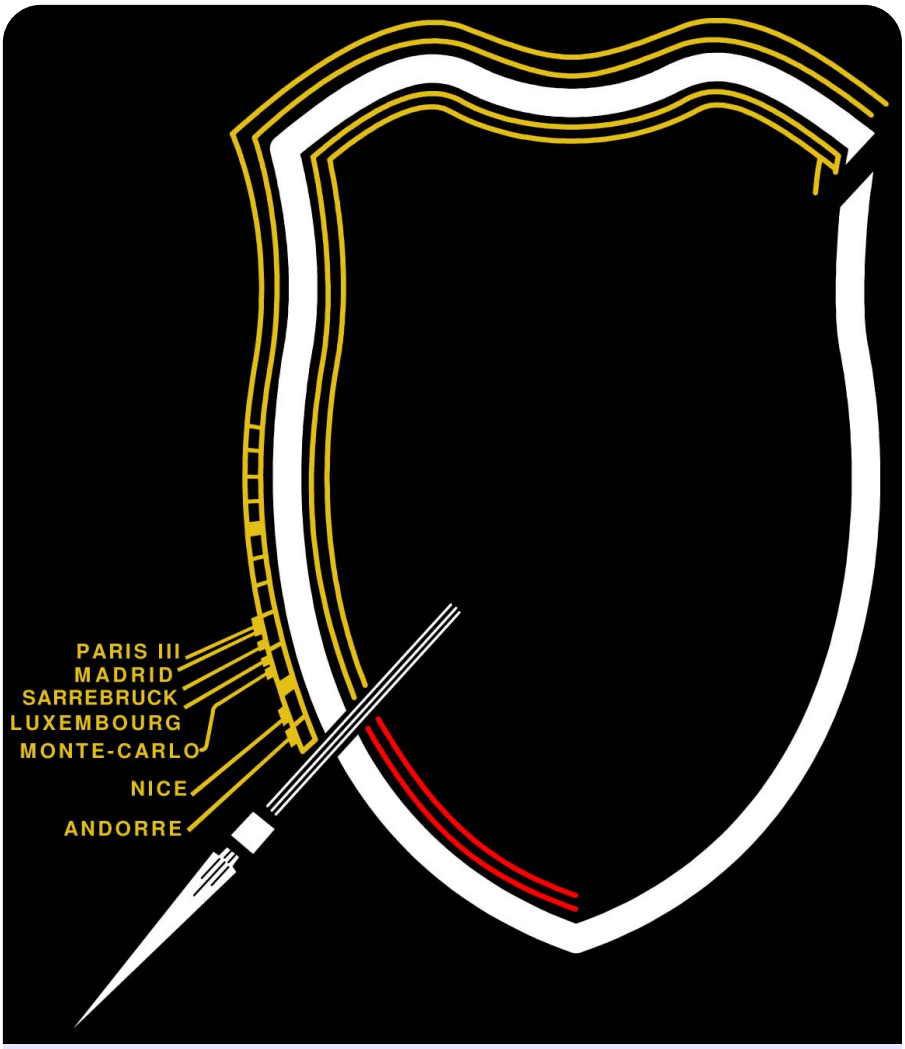

Fig. 12 Version finale incomplète

du point de départ ont à chaque fois été entrés avec la souris, puis des coordonnées relatives (des déplacements) ont été utilisées. 0r,3.1r précise, par exemple, un déplacement nul en x et un déplacement de 3,1 mm vers le haut.

Pour tracer les cinq points, cinq commandes `CercleP =` sont utilisées.

Afin de montrer comment on procède pour les formes non géométrique, le logo « Sonora » et les deux dessins pour les graves et aigus ont été récupérés de l'image d'origine et « arrangés » « vite fait » avec un logiciel de retouche photos et/ou de traitement d'images, pour être intégrés comme images dans le cadran final. La Fig. 10 montre le résultat pour les éléments qui ont été traités.

Autre exemple

A titre d'exemple, les figures 11 et 12 montrent un début de réalisation du cadran de la Fig. 5, utilisant des courbes de Bézier. Une courbe de Bézier est définie par les coordonnées des points de départ et d'arrivé et par les vecteurs de directions pour ces deux points. Les courbes se gèrent l'une après l'autre, et il faut choisir judicieusement les points : tous les points anguleux, à priori les points d'inflexions (lorsqu'une courbe passe de convexe à concave ou inversement), surtout s'il y en a plus d'un entre deux points anguleux, plus d'autres points selon la courbe à réaliser (en pratique, il est souvent avantageux d'ajouter quelques points pour faciliter le travail).

Après, il faut ajouter les graduations de longueurs d'ondes. La commande `Grad1 =` n'est pas utilisable ici, car elle ne permet de définir que la longueur d'une graduation, pas son point de départ. C'est donc la commande `SectP =` (Secteur_Plein) qui est utilisée.

Après, il faut ajouter les noms de stations et leur plage d'accord. ...

Je ne vous le fait pas dire, c'est un travail de … Bénédictin, mais le résultat est à la hauteur de l'effort fourni si la précision, la patience et la persévérance sont au rendez-vous !

Accès aux fichiers

L'ensemble des fichiers dont il est question ici, ainsi que ceux pour les cadrans des Fig. 2, et 4 sont téléchargeables sous forme d'un fichier Cadrans_TSF.zip sur le site, dans la rubrique « Téléchargements ».

Jean-Paul Gendner, F5BU
auteur du programme Galva
Site : f5bu.fr