



Concernant l'auteur

Ingénieur en électronique et informatique, Xavier MONTAGNE a travaillé en France et aux Etats-Unis sur des projets de défense et de super-calculateurs. Spécialiste en systèmes temps réels et en Linux embarqué, il est aujourd'hui directeur technique de la société Pragmatec. Auteur du *Noyau temps réel PICos18* et intervenant auprès d'écoles d'ingénieurs et de formations universitaires, il croit avant tout en l'efficacité des solutions GPL et Open-Sources.

Nano-X

Nano-X est une bibliothèque graphique adaptée aux cibles modestes comme celles utilisées pour les projets embarqués.

La bibliothèque Nano-X revêt 2 formes :

- une syntaxe proche de celle de GTK,
- une syntaxe proche de l'API Windows.

Nous parlerons ici essentiellement de l'interface similaire à l'API GTK, donc plus proche des systèmes Linux.

Nano-X permet le multi-fenêtrage et bien plus encore, le fonctionnement en réseau. Ainsi, un « serveur graphique » est exécuté sur un PC distant (ou un système embarqué distant) et votre carte de développement communiquant avec ce serveur via le réseau Ethernet afin d'afficher des informations distantes.

Notre cas d'emploi est plus modeste car il s'agit d'afficher les objets graphiques sur le LCD. Nous n'utiliserons donc pas les capacités réseau de Nano-X, et de plus, nous n'utiliserons pas de serveur graphique séparé du client graphique (votre application). Nous utiliserons une option de Nano-X dite `LINK_APP_TO_SERVER`, qui consiste à lier votre application aux bibliothèques Nano-X et ainsi obtenir une seule et même application comprenant serveur et client graphique. Le résultat obtenu est un code plus compact mais aussi plus rapide (pas d'utilisation des sockets Linux pour échanger les informations).

Nano-X est présent dans la distribution uClinux Snapgear, aussi il vous sera très facile d'obtenir les bibliothèques Nano-X compilées pour votre plate-forme.

Exemple d'application

Nano-X est capable de gérer plusieurs fenêtres et plusieurs contextes graphiques. Dans l'exemple ci-dessous, nous n'utiliserons qu'une seule fenêtre principale et un seul contexte graphique :

```
GR_WINDOW_ID MainWindow;
GR_GC_ID gc;
GR_FONT_ID font;
// Init client port
if (GrOpen() < 0) exit(1);
MainWindow = GrNewWindowEx(
    GR_WM_PROPS_APPWINDOW,
    "Demo Linux+DVD",
    GR_ROOT_WINDOW_ID, 0, 0, 320,
    240, WHITE);
GrSelectEvents(MainWindow,
    GR_EVENT_MASK_EXPOSURE |
    GR_EVENT_MASK_CLOSE_REQ |
    GR_EVENT_MASK_KEY_DOWN |
    GR_EVENT_MASK_KEY_UP);
GrMapWindow(MainWindow);
gc = GrNewGC();
GrSetGCUseBackground(gc, GR_FALSE);
GrSetGCForeground(gc, FGCOLOR);
\GrSetGCBackground(gc, BGCOLOR);
font = GrCreateFont("System", 18, 0);
GrSetFontAttr(font, GR_TFKERNING |
    GR_TFANTIALIAS, 0);
GrSetGCFont(gc, font);
```

Dorénavant, vous possédez une fenêtre au sens Nano-X du terme et un contexte graphique. Toutes les fonctions graphiques appelées devront faire référence à ces 2 éléments pour permettre à Nano-X de connaître la destination de l'objet graphique.

Pour afficher une image, il vous faudra tout d'abord la charger en mémoire, puis l'afficher :

```
int aImageID = GrLoadImageFromFile(
    "Test.bmp", 0);
GrDrawImageToFit(MainWindow, gc, 10,
    30, 100, 80, aImageId);
```

La fonction `DrawImageToFit` requiert les coordonnées du coin supérieur gauche de votre image (X,Y) ainsi que la largeur et la hauteur de l'image, cette fonction étant en effet capable de faire un stretching de l'image, c'est-à-dire de l'agrandir ou de la réduire selon la taille spécifiée.

Pour afficher du texte, c'est tout aussi simple :

```
GrText (MainWindow,
    gc, 150, 200,
    « Hello World !!”, -1,
    GR_TFASCII|GR_TFBOTTOM);
```

Pour réaliser une jauge comme sur les PC, il suffit d'afficher une image BMP ou GIF de votre choix puis un rectangle plein comme suit :

```
GrFillRect (MainWindow, gc, 10, 10,
    30, 100);
```

Ce rectangle sera situé en coordonnée (10,10), d'une largeur de 30 pixels et d'une hauteur de 100 pixels.

Résultat obtenu

Réaliser une interface graphique sur LCD tactile ne consiste pas pas seulement à afficher quelques widgets à l'écran mais surtout à réaliser un automate afin de réagir aux actions de l'utilisateur et passer d'une page à l'autre.

Vous pouvez sans trop de difficultés réaliser votre propre machine à états finis (votre automate), mais sachez qu'il existe des outils professionnels qui vous permettent de créer de telles applications graphiques en quelques minutes pour les plus simples et quelques heures pour les plus complexes.

Vous pouvez même trouver des plateformes de développement complètes qui intègrent le LCD et l'écran tactile ainsi que les bibliothèques graphiques et les outils logiciels de développement d'applications graphiques.

Conclusion

Cet article vous a montré qu'il était assez simple de réaliser une application graphique sur LCD. Bien entendu, il peut être très long et très fastidieux de franchir soit même chaque étape, à savoir la compilation croisée de Nano-X, du gestionnaire d'écran tactile, de l'application graphique elle-même ou de l'automate de gestion graphique.

Aussi, n'hésitez pas à consulter les différentes offres sur Internet, vous trouverez certainement la plate-forme qui convient parfaitement à votre application et qui vous offre tous les outils logiciels dont vous avez besoin ... 🐣



Sur le réseau

- Site de uClinux, version embarquée de Linux pour microcontrôleurs : <http://www.uclinux.org>
- Site officiel du serveur nano-X : <http://www.microwindows.org>
- Les cartes de développement SIMTEC : <http://www.simtec.co.uk>
- Les cartes de développement Pragmatec : <http://www.pragmatec.net/catalog>