

Comment programmer un ADDIN sur graph 85 ?

La graph 85 est une calculatrice programmable. Il y a un menu à l'intérieur : « PRGM », permettant de programmer en un basic modifié. Mais on peut également programmer ce qu'on appelle des « ADDINS ». Les graphs 35+/65+ ne permettent pas de programmer ces addins. Les addins sont des équivalents d'exécutables sous Windows : vous les ouvrez, et quelque chose se lance. On peut créer toutes sortes de programmes : des jeux, ou des applications par exemple...

Ce tutorial a pour but de vous expliquer comment programmer ces fameux addins.

Il faut savoir tout d'abord que vous programmerez vos addins sur votre ordinateur, et non pas sur votre calculatrice, comme c'est le cas avec le basic.

Vous allez en fait taper votre programme sur un **compilateur**, appelé SDK (Software Development Kit). C'est ce compilateur qui se chargera d'adapter votre programme, le **code source**, en un programme compréhensible par votre calculatrice.

Attention précision : je ne vais pas vous expliquer comment programmer en C ou C++, d'autres tutos sont très bien fichus. Mon but ici est de vous expliquer comment programmer ce langage sur votre calculatrice. En gros ce tutorial est fait pour vous si vous vous dites : « Je sais programmer en C/C++, ou un petit peu, et je voudrais créer un addin. Mais voilà, comment m'y prendre ? ».

Vous voulez apprendre à programmer en C/C++ ? Je vous conseille fortement le site du Zéro à [cet endroit](#).

Et bien voilà, vous avez tous les outils en main, même la patience, on peut maintenant commencer ! Hé oui, c'est un peu le but de ce tuto en même temps :p

Avant de montrer vos talents de programmeurs, il va vous falloir récupérer ce fameux SDK. Il est disponible sur le site officiel [ici](#).

Cliquez sur « Accept » pour accepter la licence. Pas d'inquiétude, tout est gratuit.

Ensuite, il va falloir vous inscrire pour récupérer ce SDK. Oui, c'est obligé !

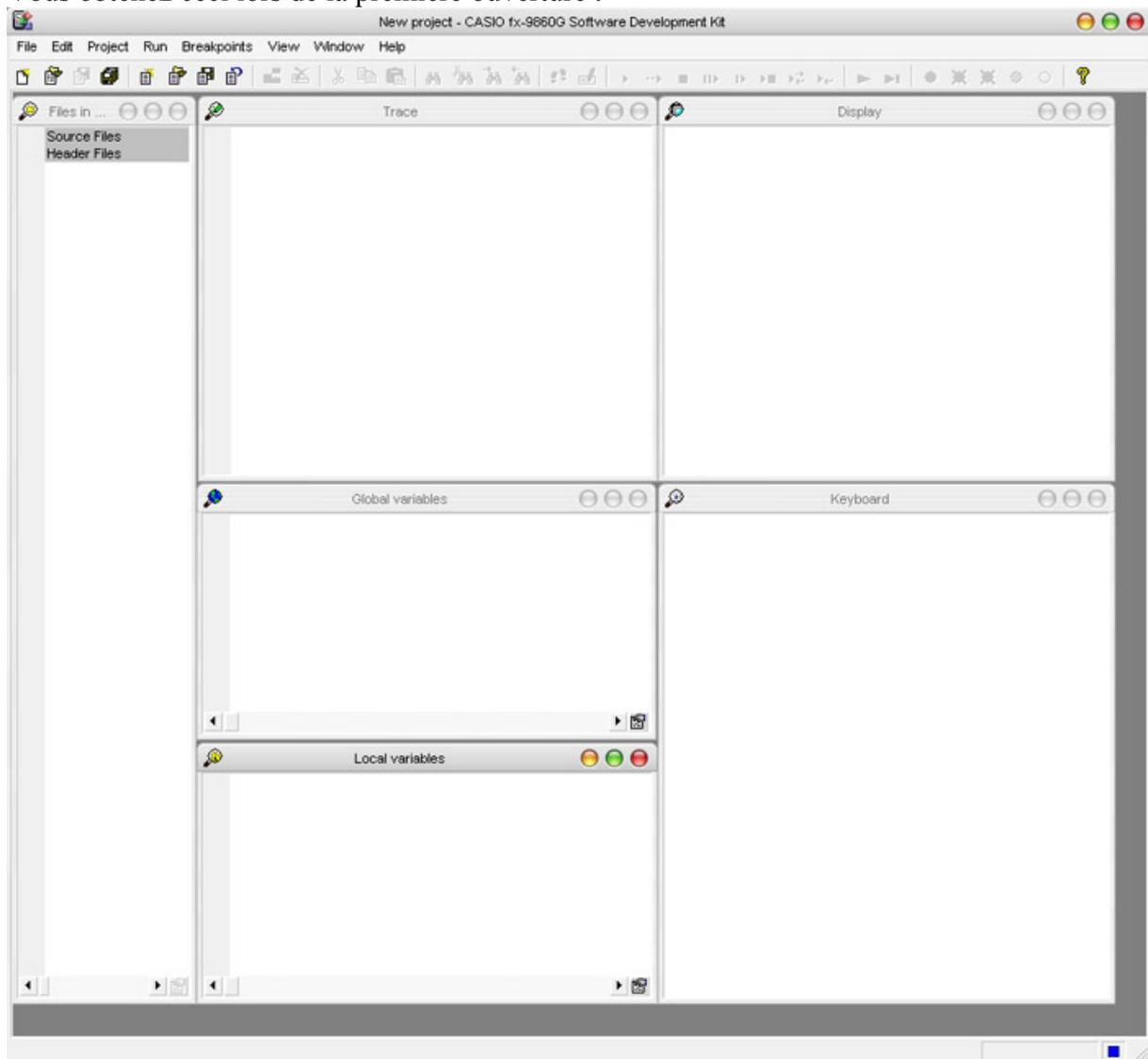
Soit vous avez déjà un compte et vous rentrez votre mail et votre mot de passe, soit vous cliquez sur « Registration » pour vous enregistrer. Si vous vous enregistrez, il va vous demander un code (en fait 2) situés à l'arrière de votre graph 85. Cochez tout d'abord la case « fx-9860G series ». Fx-9860G est en fait le nom américain de votre calculatrice, la graph 85 (tellement plus simple pour nous ^^). Cliquez sur le bouton bleu HELP pour voir où trouver ces codes. Puis continuez normalement l'inscription.

Enfin connectez-vous, histoire de récupérer ce SDK quand même ☺

A votre connexion, vous verrez plein de trucs à télécharger. Le seul truc qui nous interesse, c'est le SDK. Regardez donc dans la partie « fx-9860G series », et cochez le SDK avant de cliquer sur « Next ». Puis cliquez sur « Version 1.00 » (il n'y aura pas de mise à jour dans le futur). Une petite pop up s'ouvre, cliquez simplement sur Download. Et là, on attend, parce qu'ils ont pas de supers serveurs chez Casio :p
Puis installez-le normalement, puis ouvrez-le.

Maintenant, les choses sérieuses commencent héhéhé :p :p

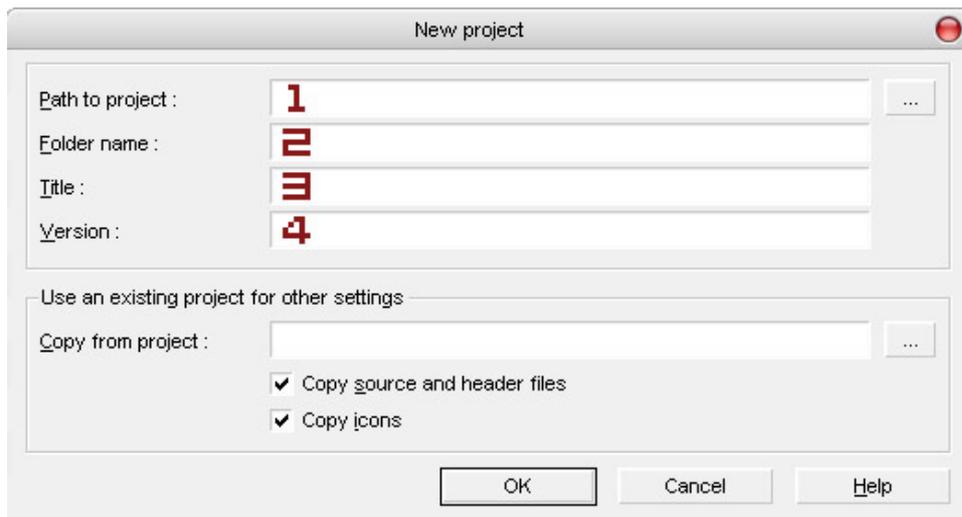
Vous obtenez ceci lors de la première ouverture :



Je pense qu'avant toute chose, vous avez hâte de faire votre 1^{er} addin ^^ J'me trompe ? :D

Alors avant même de vous expliquer dans le détail, cliquez sur Project puis New.

Vous obtiendrez cette fenêtre :



1. Vous devez mettre ici le dossier où se situera votre addin. Je vous conseille de faire un dossier « SDK » où vous mettrez tous vos addins pour vous y retrouver. Une fois que vous vous trouvez dans le dossier désiré, ne mettez aucun nom de fichier, cliquez simplement sur OK.

2. Dans le dossier que vous avez créé en 1, se créera un autre dossier avec cette fois le nom de votre addin. Tapez « 1^{er} prog »

3. Le nom de votre addin. Tapez « essai »

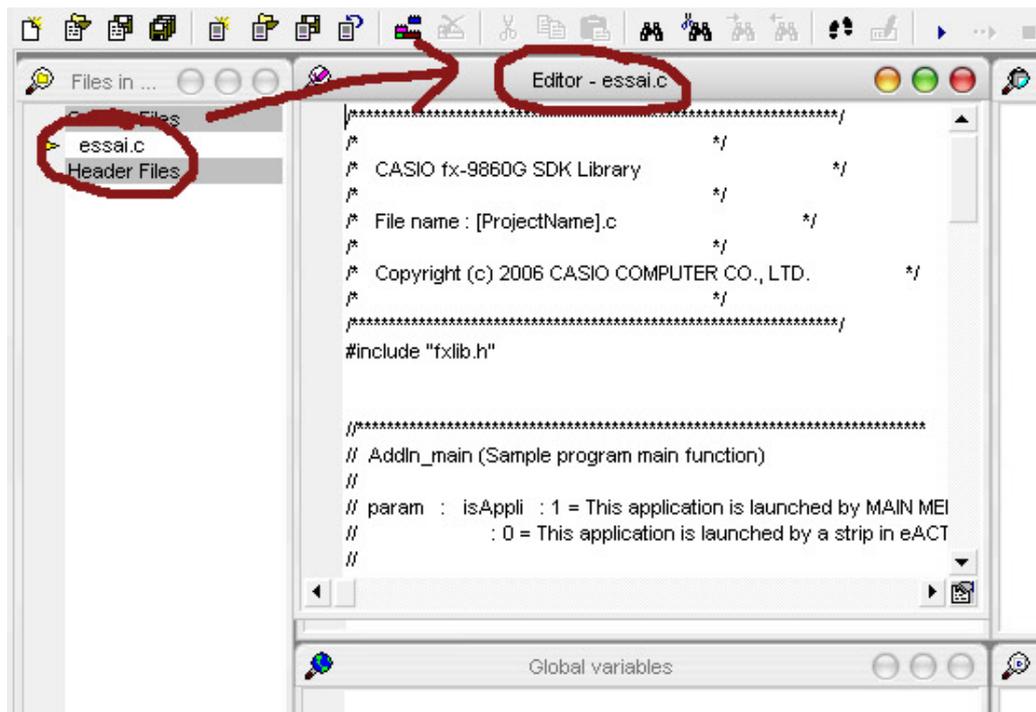
4. La version de votre addin. Mettez 1.

Ne vous occupez pas du bas.

Cliquez sur OK.

Un avertissement apparaîtra, cliquez sur OK de nouveau.

Dans la colonne de gauche (Files in project), vous voyez « essai.c ». Oui ! c'est bien votre addin ! Double cliquez dessus, et le programme s'ouvre dans la partie Editor, telle la capture ci-dessous :



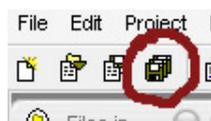
Dans la fenêtre Editor, descendez et modifiez ce texte :

```
locate(1,4);
Print((unsigned char*)"This application is");
locate(1,5);
Print((unsigned char*)" sample Add-In.");
```

En ceci par exemple :

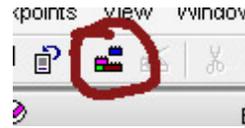
```
locate(1,4);
Print((unsigned char*)"Wahou c'est mon");
locate(1,5);
Print((unsigned char*)" premier Add-In !");
```

Sans le **gras** évidemment.



Cliquez ensuite sur ce bouton :

Ca sauvegarde tout ce que vous avez tapé pour le reprendre plus tard. En aucun cas cela compile votre addin.



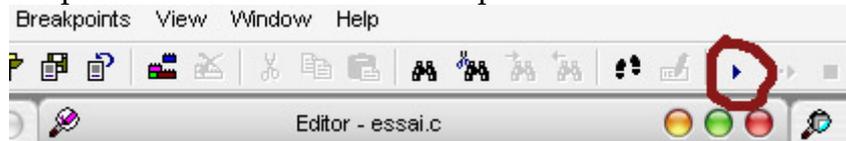
Pour cela, cliquez sur l'icône de compilation :

Si vous n'avez pas sauvegardé, il vous le proposera. Sauvegardez.

Puis dans la fenêtre Build results, plein de blabla avec à la fin « Build has completed », ce qui veut dire que votre addin a été correctement compilé ^^ Si ce n'est pas le cas, c'est que vous avez fait une erreur en modifiant le texte, revérifiez donc.

Bref, votre 1^{er} addin a été compilé ! Vous pouvez maintenant le copier sur votre calculatrice en allant chercher « `essai.g1a` » dans le dossier que vous avez spécifié auparavant. Vous avez la flemme d'aller chercher votre graph ? Soit ! Sachez que le SDK possède un émulateur ! Haha, étonné, n'est-ce pas ?

Cliquez donc sur ce bouton sans plus attendre :



Et là, Ô merveille, une belle graph 85 apparaît sur la droite.

Allez simplement sur l'icône « Debug » du menu de la calculatrice (cliquez sur les boutons de cette dernière comme une vraie). Cliquez sur [EXE] et appréciez ^^

Comme votre programme ne s'appelle pas Debug, pensez à modifier l'icône qui se trouve à côté de votre `.g1a` pour votre calculatrice, avec Paint par exemple qui va très bien. Attention à conserver les dimensions (30x19), le format `.bmp` et le monochrome (pas de couleurs ni de gris !). Pour que les changements fassent effet, recompilez votre addin.

Et voilà, une bonne étape a été franchie, vous connaissez les bases du logiciel, qui ressemblera fortement à l'IDE où vous avez appris à programmer le C (IDE=l'environnement de développement pour créer votre programme, le SDK).

Maintenant nous allons voir plus en détail tout cela. Cliquez sur le « Stop » à côté du « Play » pour arrêter l'émulation.