

# TUTORIEL D'INITIATION A LA PROGRAMATION EN LANGAGE ASSEMBLEUR POUR MSX

## 1ere PARTIE – L'ENVIRONNEMENT DE DEVELOPPEMENT

Bonjour à tous, je me décide à écrire le tutoriel pour vous tous afin que ceux qui sont désireux de commencer la programmation pour MSX et qui ne savent pas par où commencer et quels outils utiliser.

On ne va pas expliquer pas à pas le langage assembleur Z80 mais donner des notions de programmation qui emmènent à la réalisation de jeux vidéos pour nos chers MSX.

Le premier sujet abordé sera l'installation d'outils qui vont nous faciliter cette tâche, si vous vous sentez plus à l'aise avec d'autres outils, n'hésitez pas à basculer sur ceux que vous utilisez.

J'ai regroupé dans un [Pack](http://www.megaupload.com/?d=9VS82B9A) l'ensemble des outils présentés via ce lien  
<http://www.megaupload.com/?d=9VS82B9A>

Créez un répertoire dans votre disque dur et appelez-le C:\MSX , décompressez-y le [PACK](#) de mes outils dans ce répertoire pour les avoir sous la main en un seul et même endroit.

Pour compiler le code en assembleur nous allons utiliser ce qui est pour moi le meilleur cross-compileur de Z80 pour MSX qui n'est autre que le [asMSX](#) de KAROSHI.

<http://karoshi.auic.es/index.php?topic=834.0> ( Vous pouvez télécharger ici la dernière version)

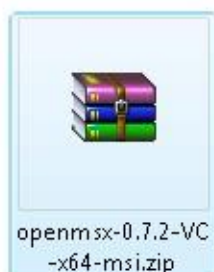
Ce compilateur n'inclus pas d'éditeur de texte dans lequel on va écrire le code que nous voulons compiler. Après de nombreuses recherches j'ai choisi [EditPlus](#) version 3.01 qui est dans le [PACK](#) et qui sera la base de notre développement.



Allez dans le répertoire [MSX](#) où se trouve le [Pack](#) et décompressez le fichier [asmsx012e.zip](#) dans le répertoire [C:\MSX](#) , il va créer dans ce répertoire un répertoire [asmsx012e](#)

[C:\MSX\asmsx012e](#)

Dans ce répertoire se trouve notre assembleur avec tous ses exemples.



Nous allons maintenant décompresser l'émulateur MSX pour jouer le résultat de notre compilation, logiquement le meilleur pour moi c'est [BlueMSX](#) mais pour lancer un programme compilé j'utilise [OpenMSX](#) qui facilite le lancement et l'ouverture de fichier compilé.

Mais pour vérifier le code et déboguer les erreurs j'utilise [BlueMSX](#).

Décompressons [OpenMSX](#) du [Pack](#) vers votre répertoire, vous verrez que vous avez 2 fichiers en fonction de votre configuration [-x64](#) si vous utilisez Windows [64Bits](#) et [-x86](#) si vous utilisez Windows [32 Bits](#).

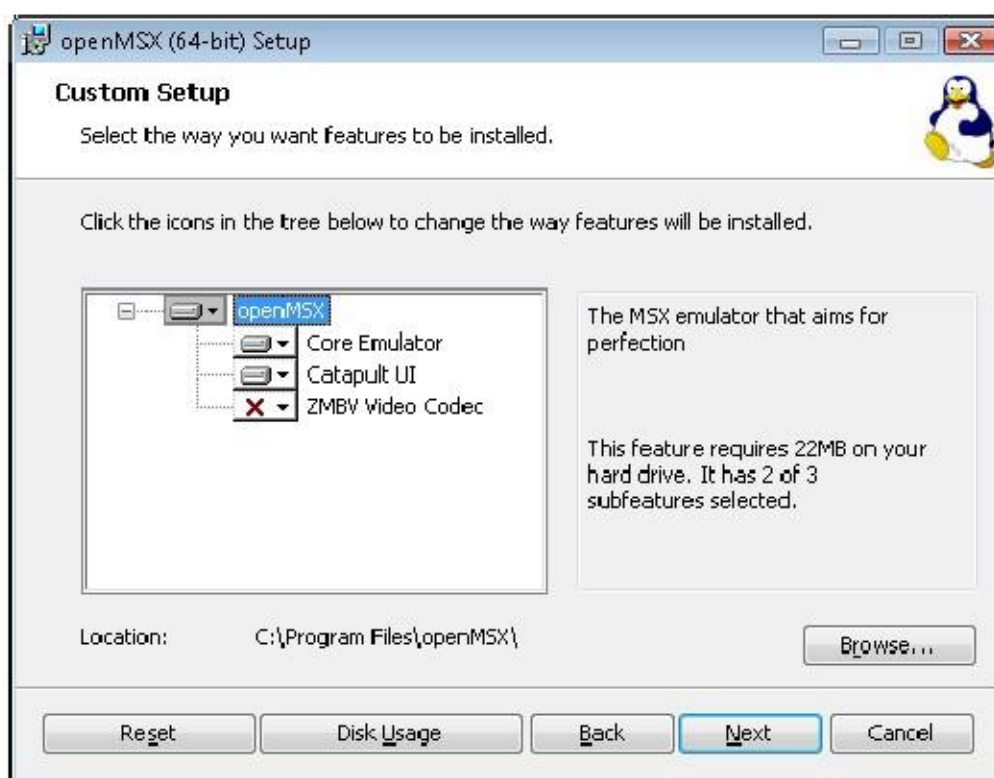


Une fois que l'on a choisi l'OpenMSX suivant votre version Windows du Pack, l'installateur d'openMSX version 0.7.2 dans mon cas j'utilise Windows 7 Ultimate 64 Bits , ainsi j'ai décompressé la version 64 Bits.

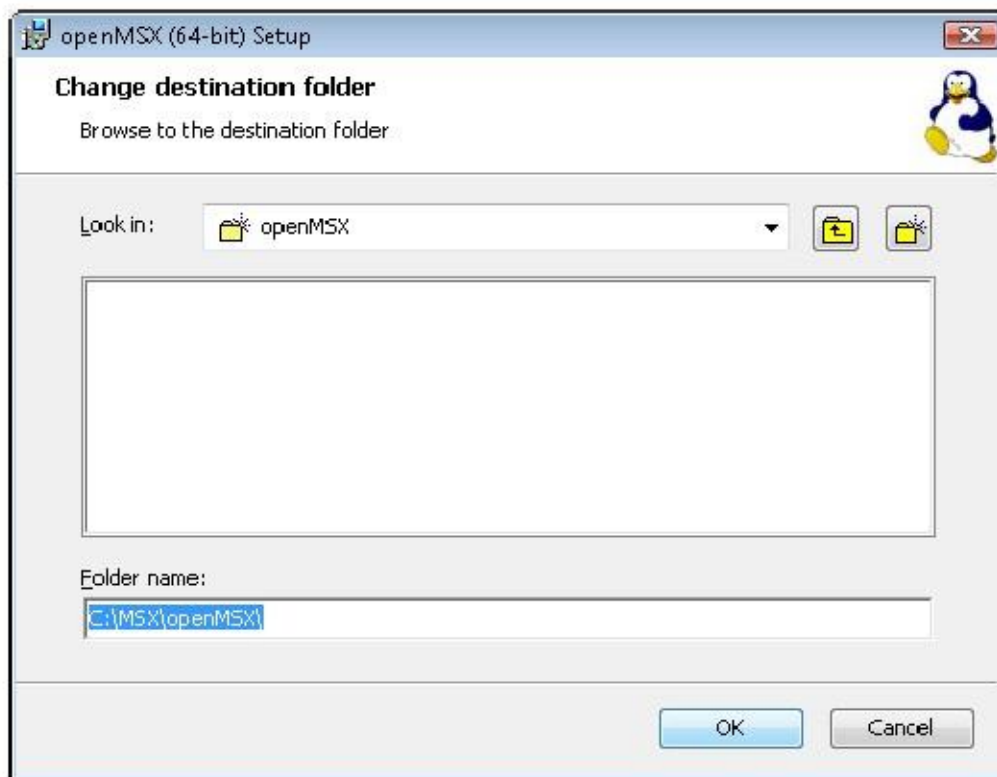
Double-cliquez sur l'icône pour commencer l'installation.



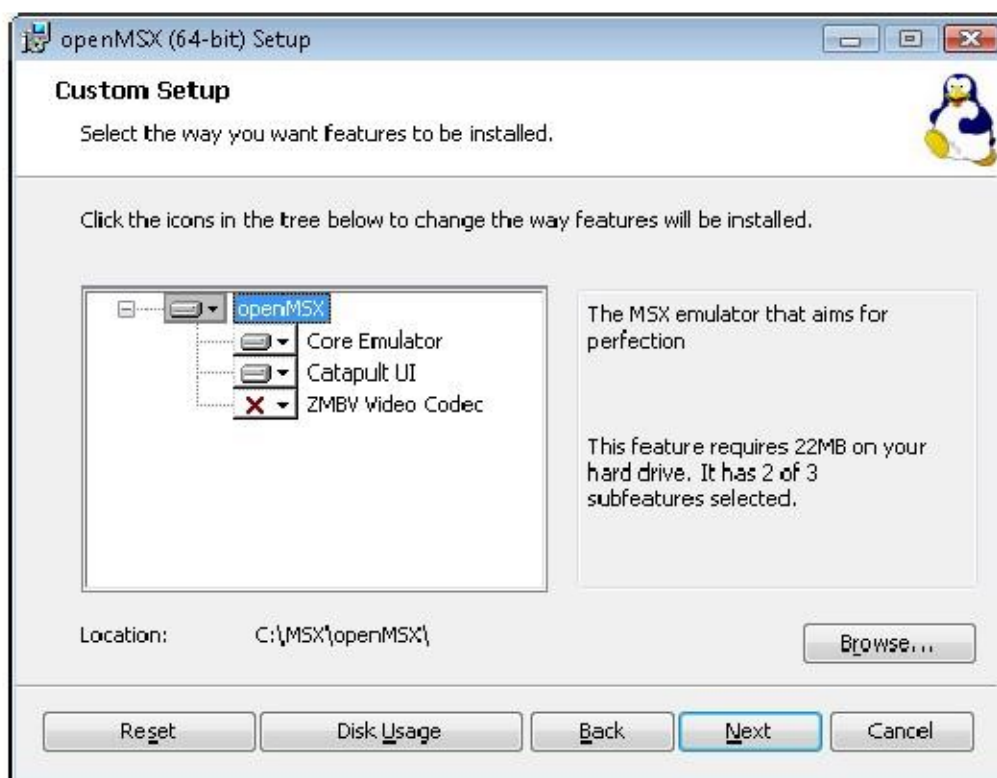
Ceci c'est l'installateur d' [OpenMSX](#) appuyons sur le bouton "Next"



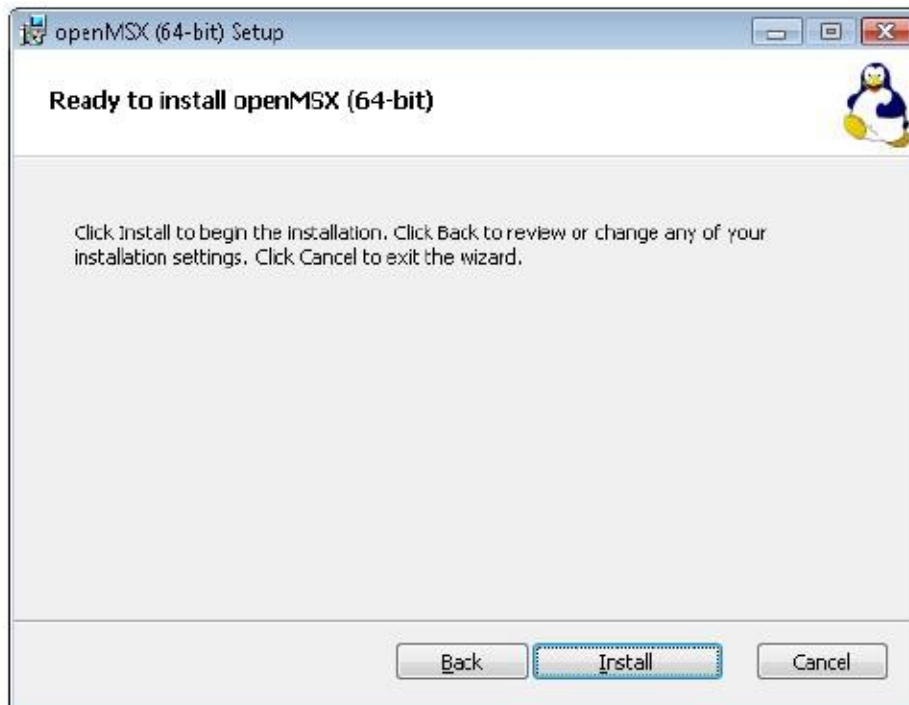
Cette partie est importante, elle change le répertoire d'installation pour qu'on lui indique le notre [C:\MSX](#) qui fera que notre Editeur le trouvera facilement. Appuyez sur "Browse".



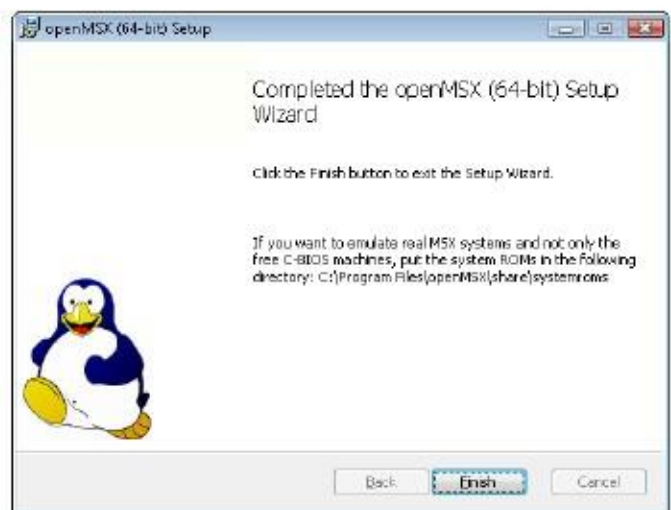
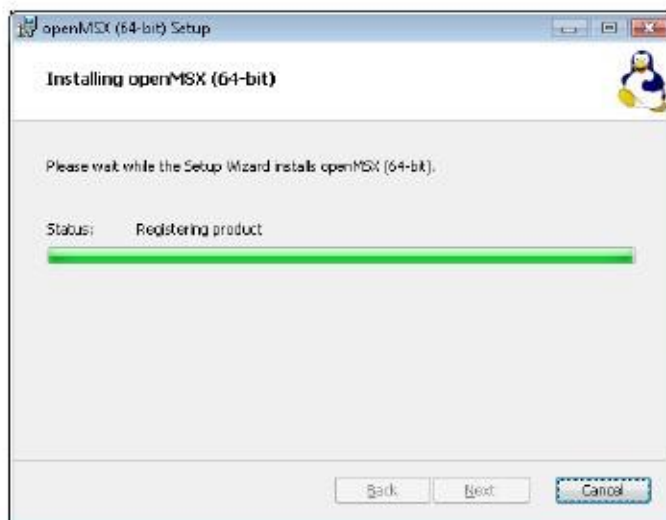
Modifiez le **Folder name:** par **C:\MSX\openMSX** et appuyez sur le bouton “OK”



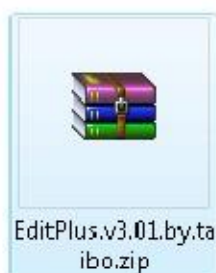
On peut voir lque le répertoire a été modifié, nous appuyons sur le bouton “Next”



Finalement appuyons sur "Install"



Attendez que cela s'installe, vous devriez voir ces écrans puis appuyez sur "Finish" pour finir l'installation.



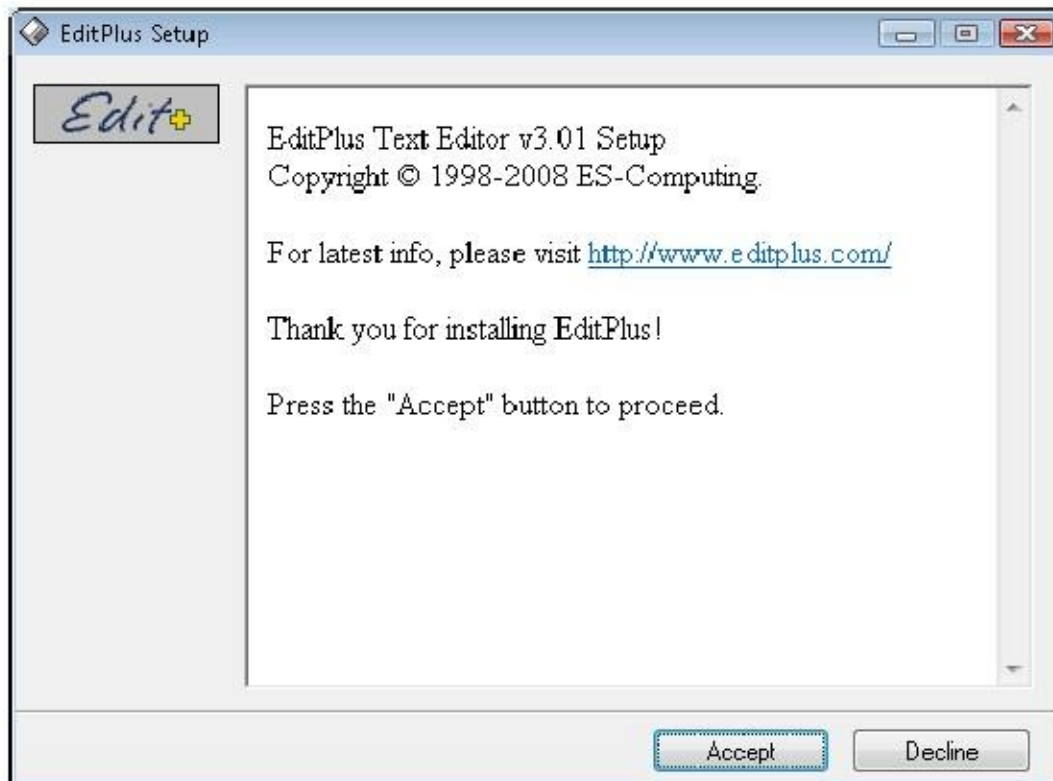
Installons maintenant ce qui sera l'organe commun de notre base de travail. Décompressez EditPlus v.3.01 du Pack.

Malheureusement ce soft est payant, mais mérite qu'on l'installe pour ses nombreux avantages. Lisez les instructions du fichier .txt dans le .zip.

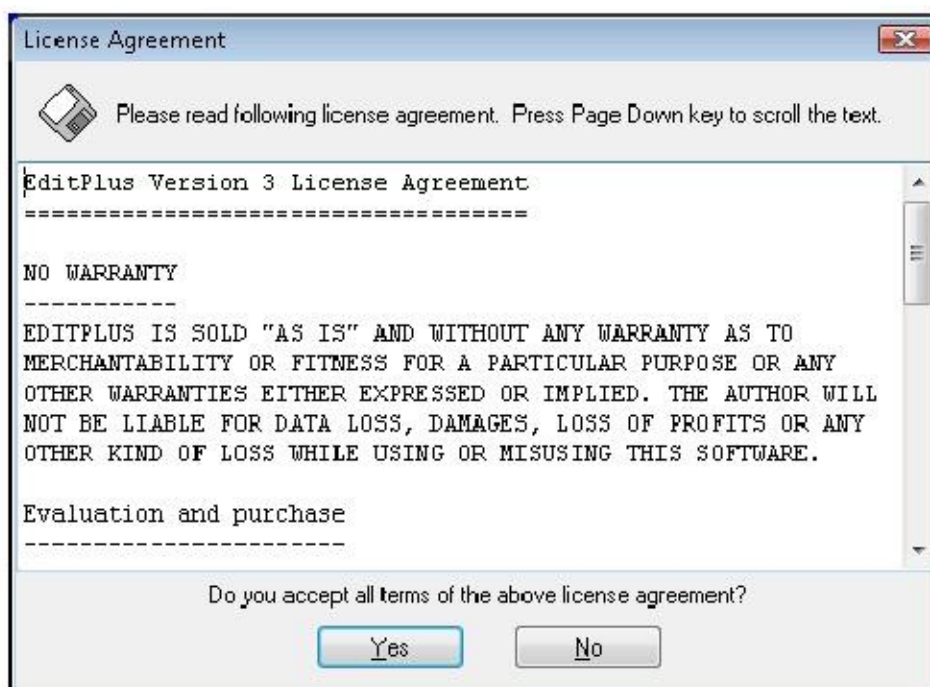
Dans le répertoire du Pack, vous devriez voir cet icône qui est l'EditPlus compressé.



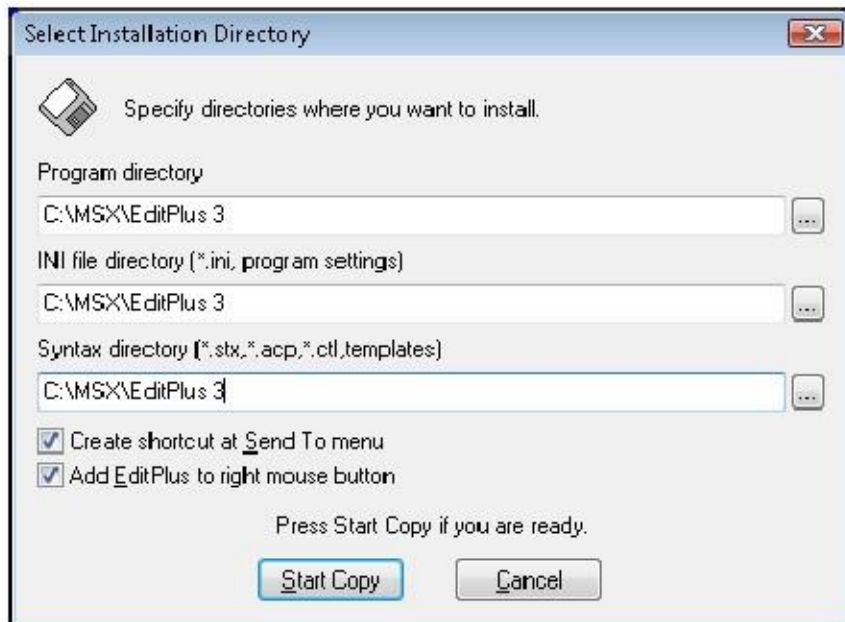
Double-cliquez sur l'icône pour l'installer.



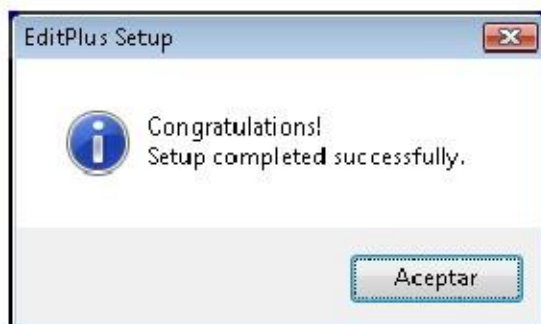
Appuyez sur le bouton "Accept"



Appuyez sur le bouton "Yes" pour accepter la licence.

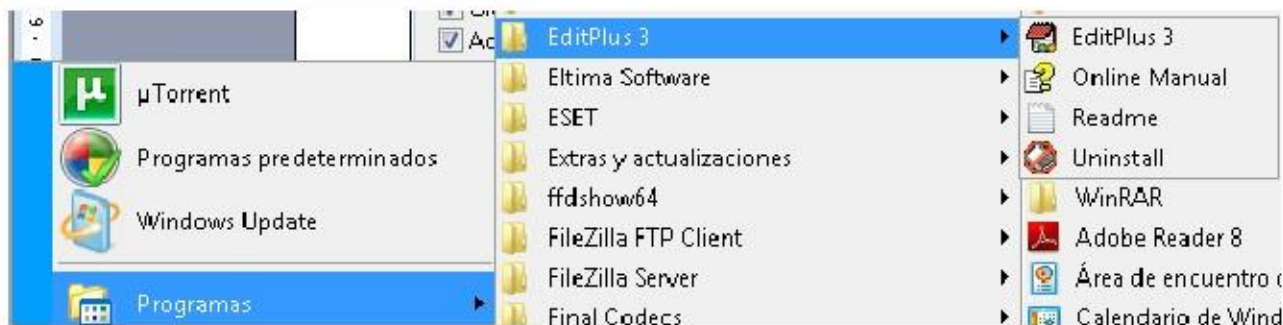


Indiquez les 3 repertoires avec `C:\MSX\EditPlus 3` comme indiqué.

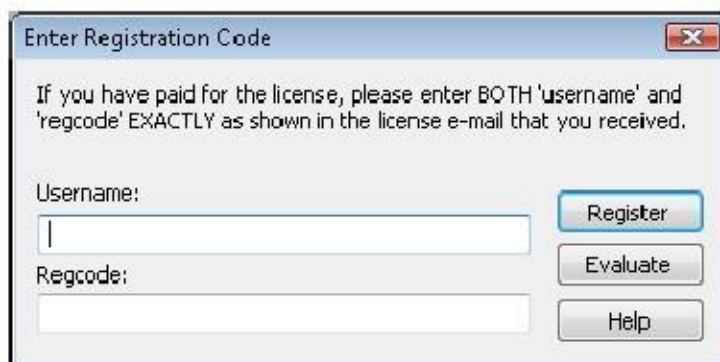


Il est maintenant installé.

Appuyez sur le bouton "Aceptar"



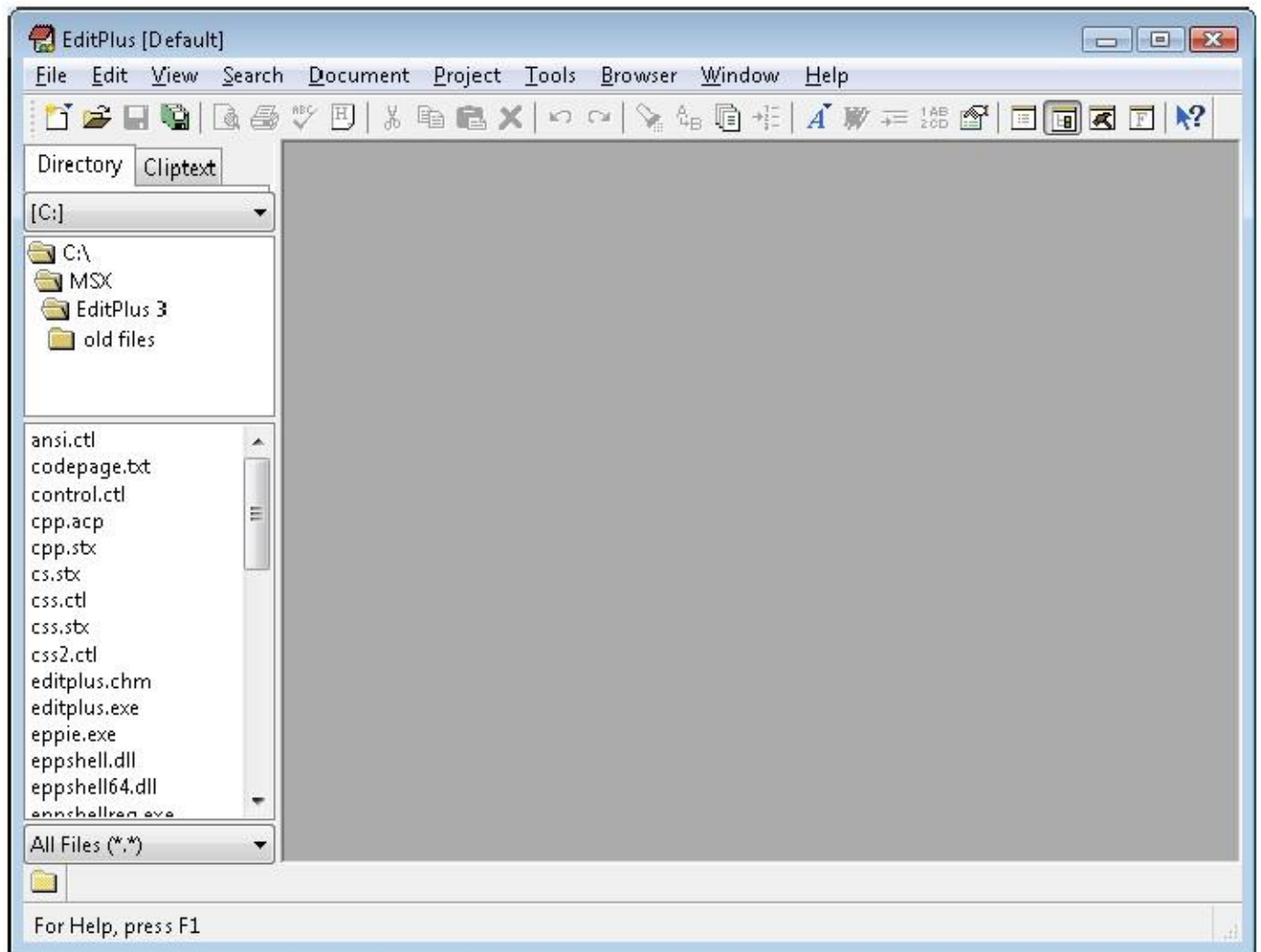
Nous allons executer en mode administrateur pour cela appuyez sur **Programmes – EditPlus 3 – EditPlus 3**, avec le clic droit de la souris et appuyez sur **exécuter en tant qu'administrateur** (pas besoin sous XP).



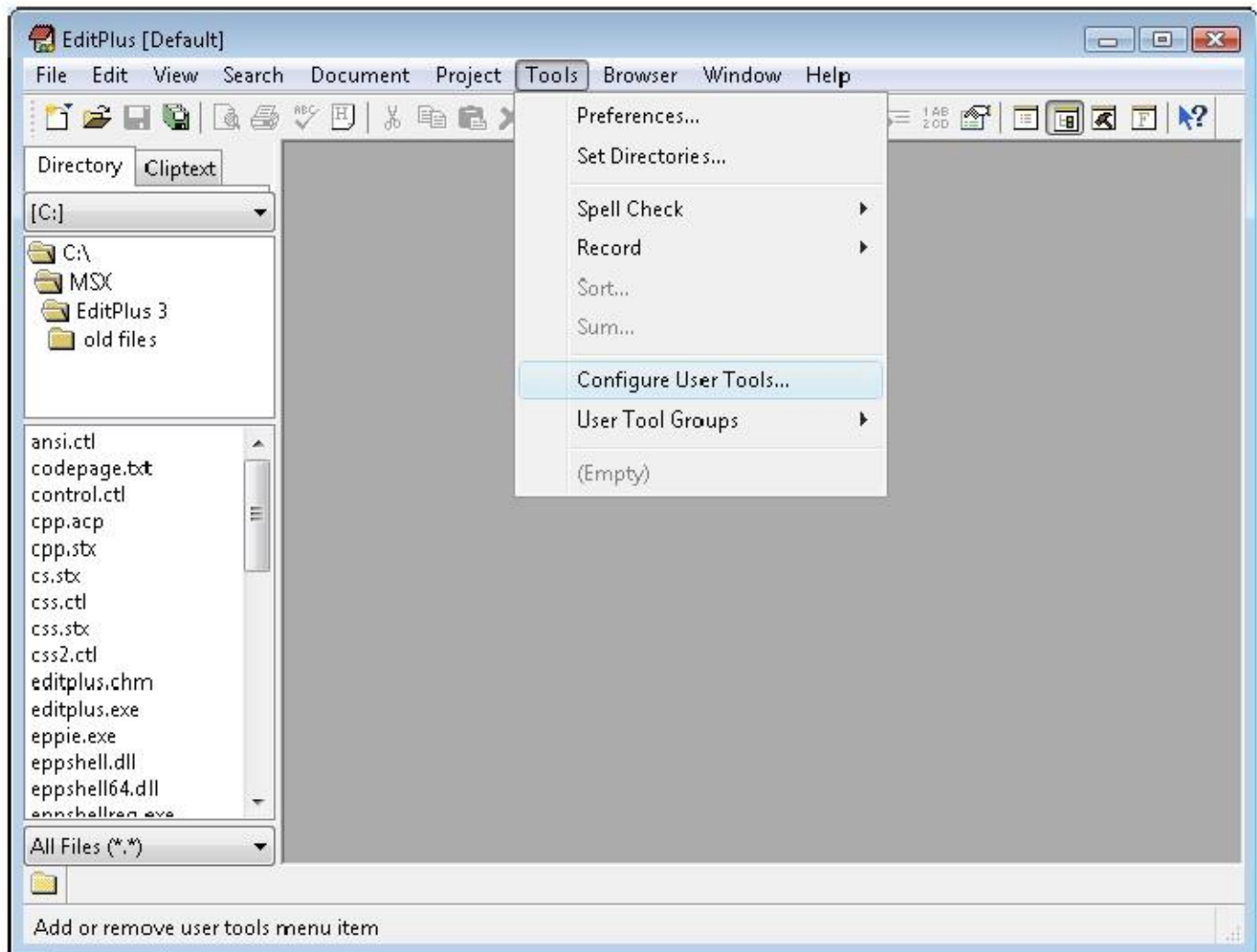
Si vous avez les informations pour vous enregistrer, c'est le moment de le faire et d'appuyer sur le bouton "Register", si vous lisez le .txt cela devrait vous aider.

Sinon appuyez sur "Evaluate" pour utiliser le programme pendant 30 jours.

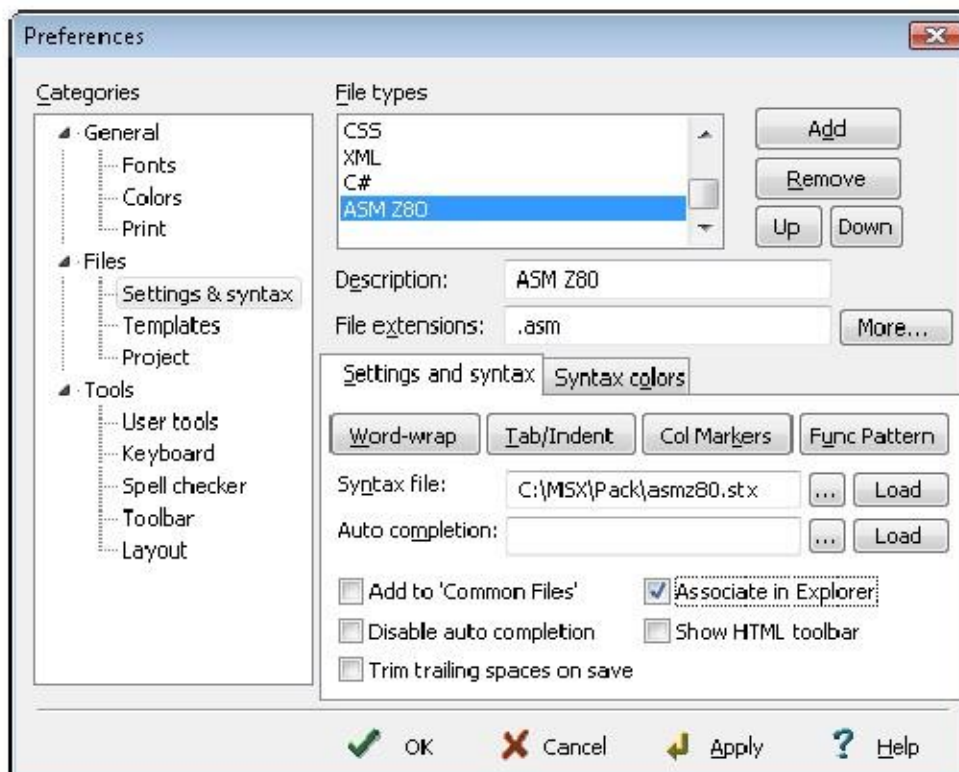




Nous avons ici notre éditeur dans toute sa splendeur, nous allons maintenant faire des manipulations très importantes, une pour ajouter à l'éditeur un colorateur syntaxique du langage machine Z80 à l'aide du fichier [asmz80.stx](#) du Pack.



Pour l'ajouter, allons dans menu **Tools – Configure User Tools...**



1 – D'abord sélectionnons l'option **settings & syntax**.

2 – Appuyez sur le bouton **Add**  
Tapez dans le champ **Description**: ceci **ASM Z80**  
Et dans **File extensions**: **.asm**

3 – Ensuite dans **Syntax file**: appuyez sur le bouton **...** et sélectionnez comme dans l'image suivante le fichier **asmz80.stx** qui est dans le répertoire du Pack.

Cochez **Associate in Explorer**  
Si vous avez la même chose que sur l'image de gauche alors appuyez sur les boutons **Apply** et **OK**



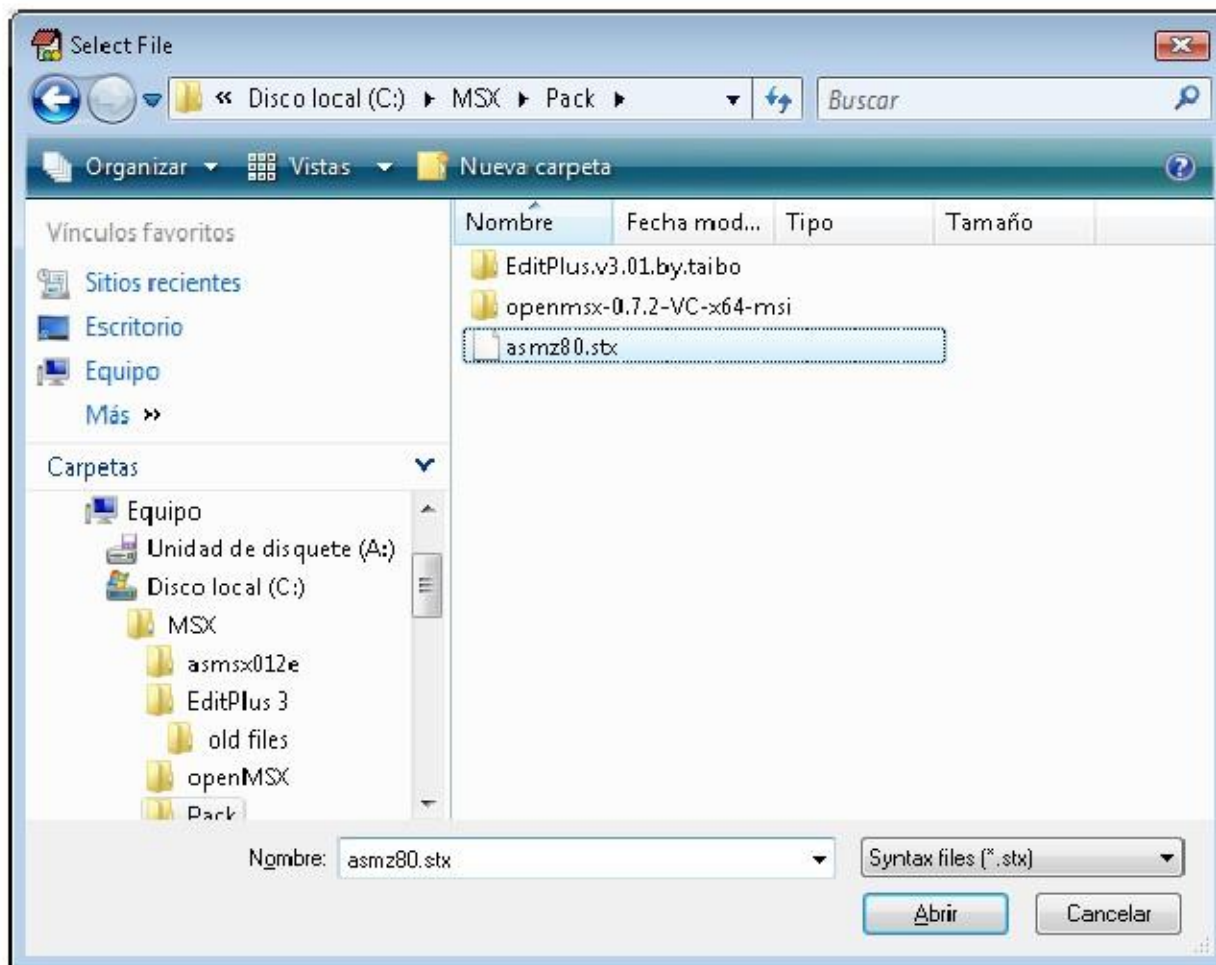
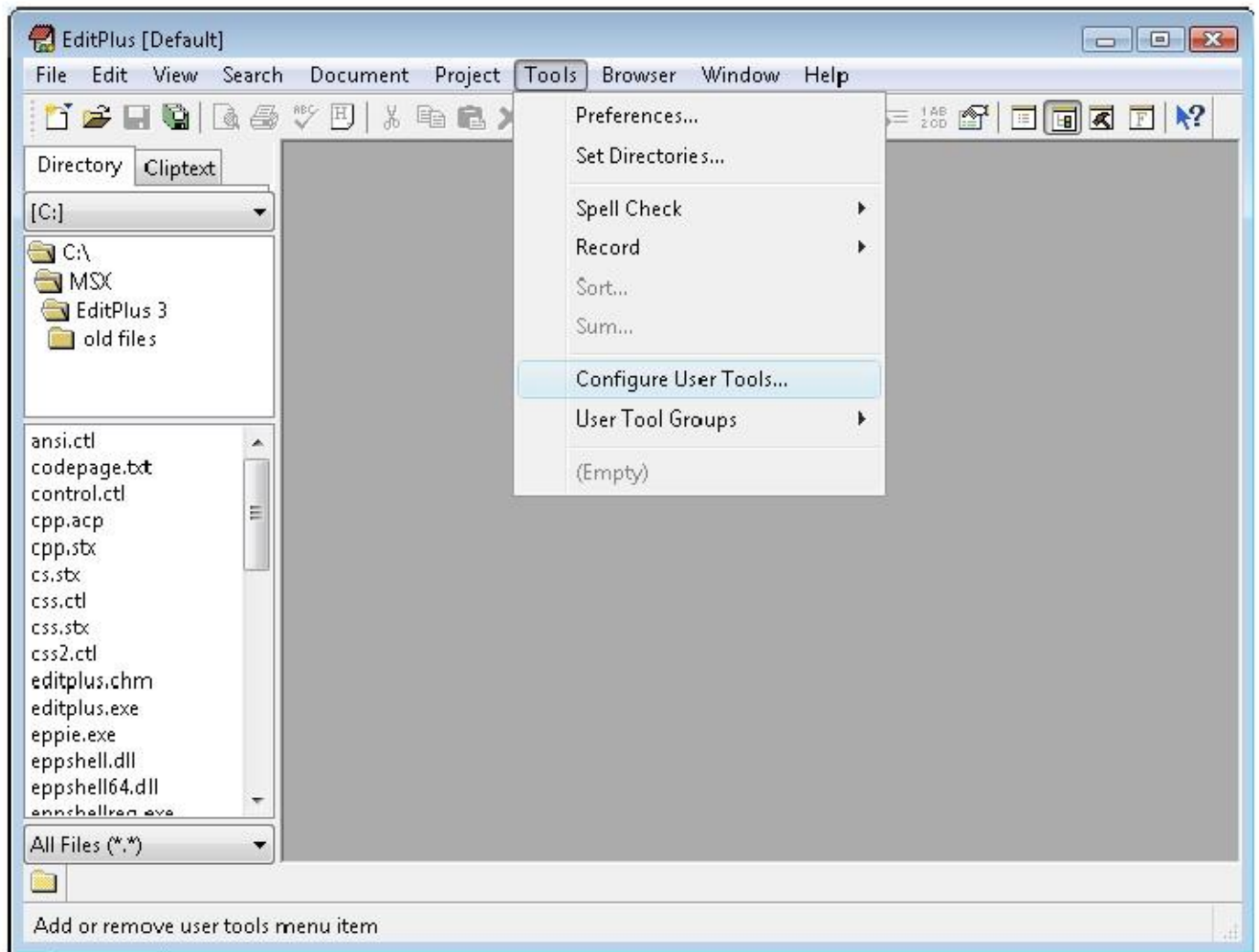
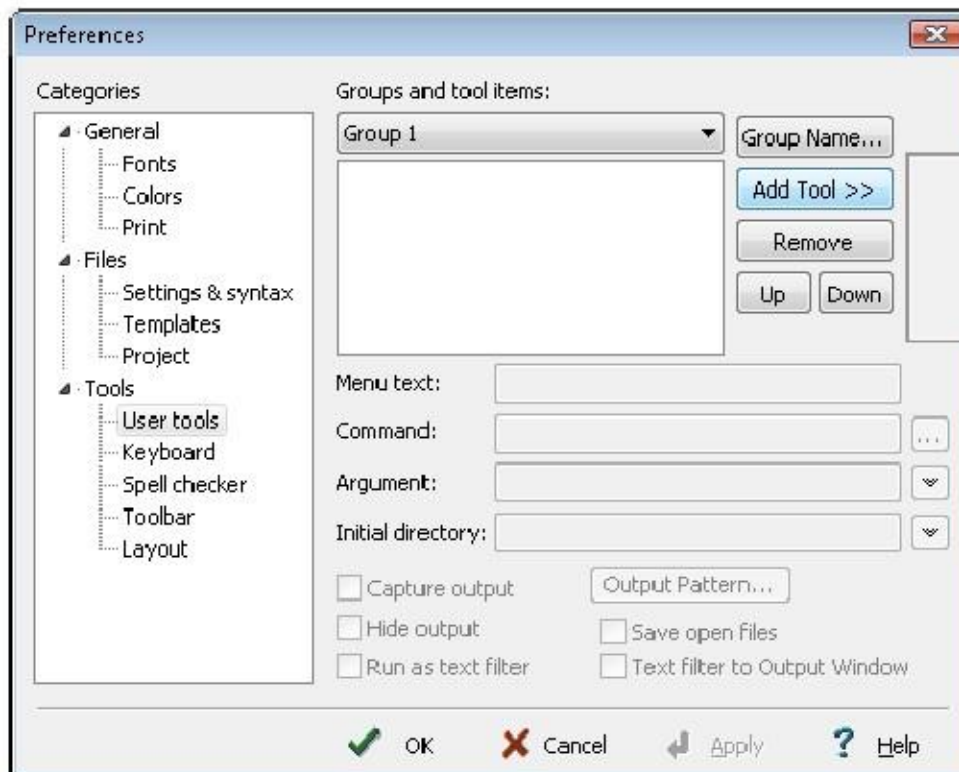


Image qui correspond à l'étape 3 ci-dessus.

On va créer nos propres boutons, un pour lancer une compilation et un autre pour lancer le fichier compilé (ou ROM) dans l'émulateur [openMSX](#).



De nouveau appuyez sur le menu d'EditPlus [Tools – Configure User Tools...](#)



Sélectionnez à gauche [User tools](#)

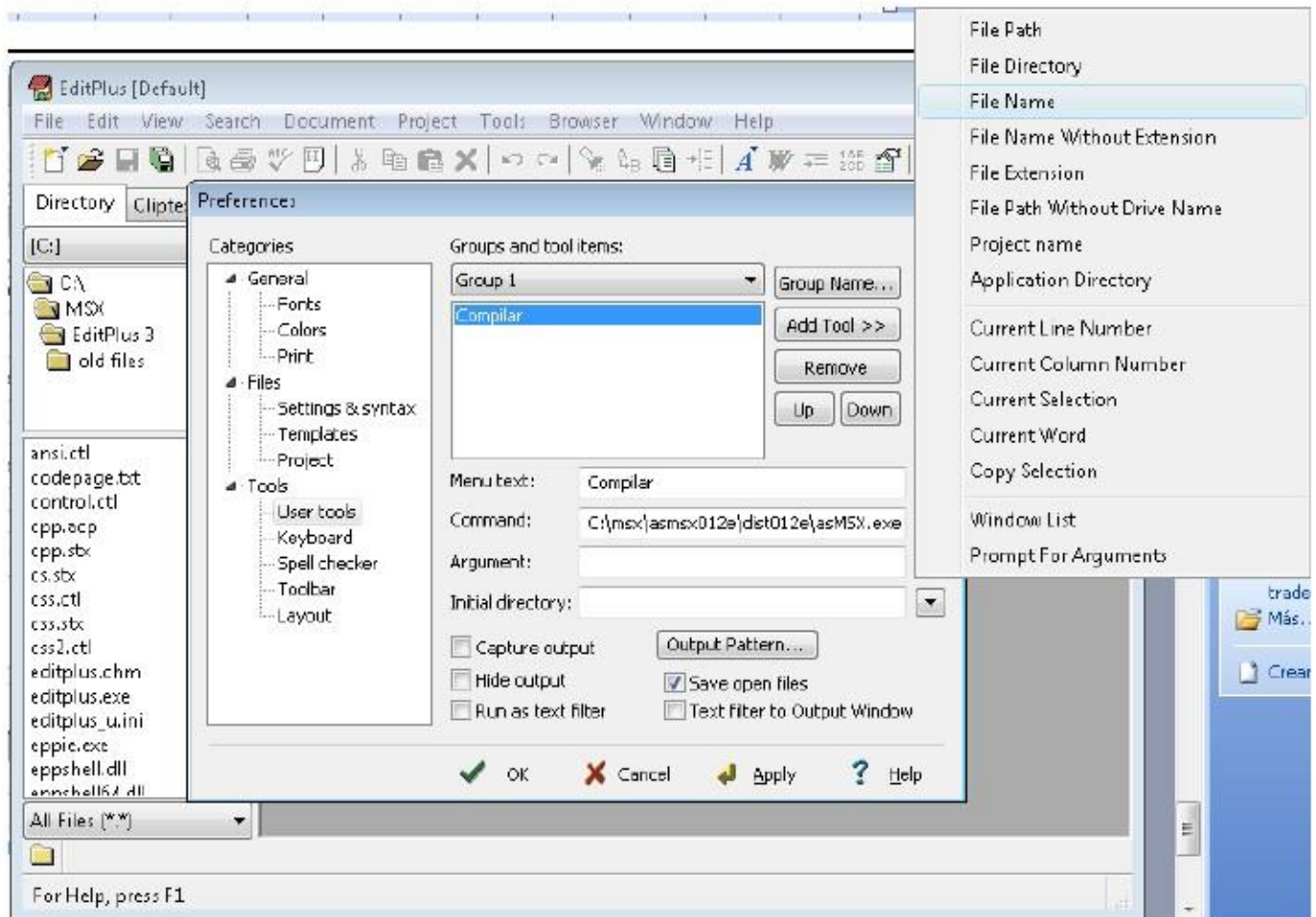
Program  
Help File (\*.hlp)  
HTML Help file (\*.chm)  
Keystroke Recording

Appuyez sur le bouton [AddTool](#) et dans la fenêtre qui s'ouvre prenez l'option [Program](#)

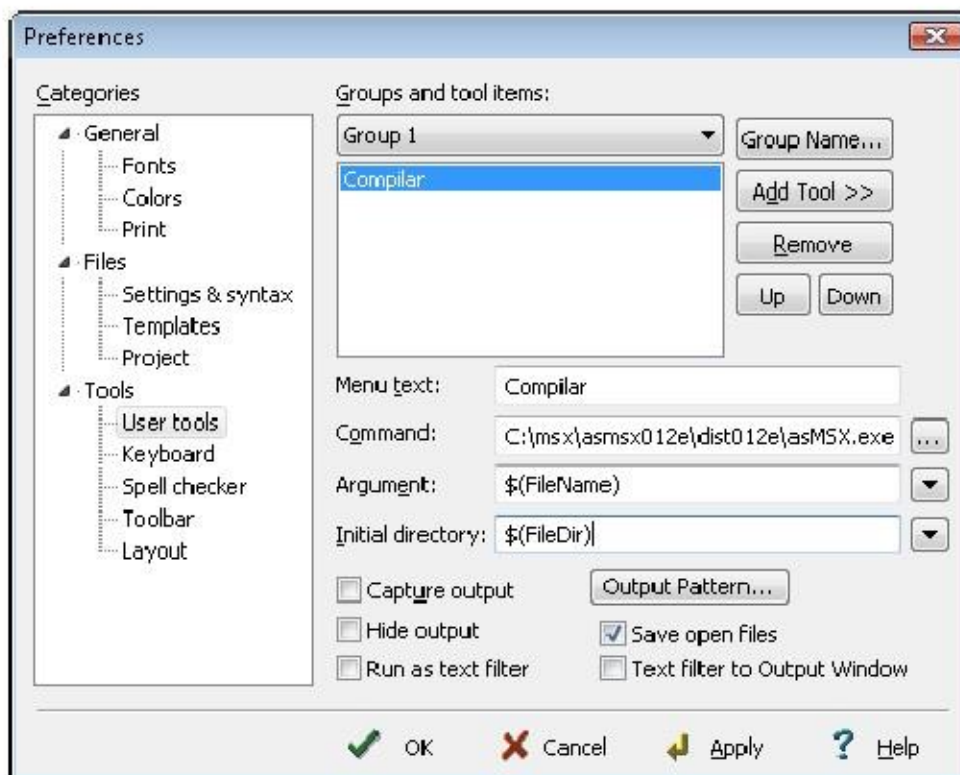
Ensuite dans [Menu text:](#) tapez [Compiler](#)

Dans le champs [Command:](#) Appuyez sur les 3 petits points ...

Voir l'image suivante

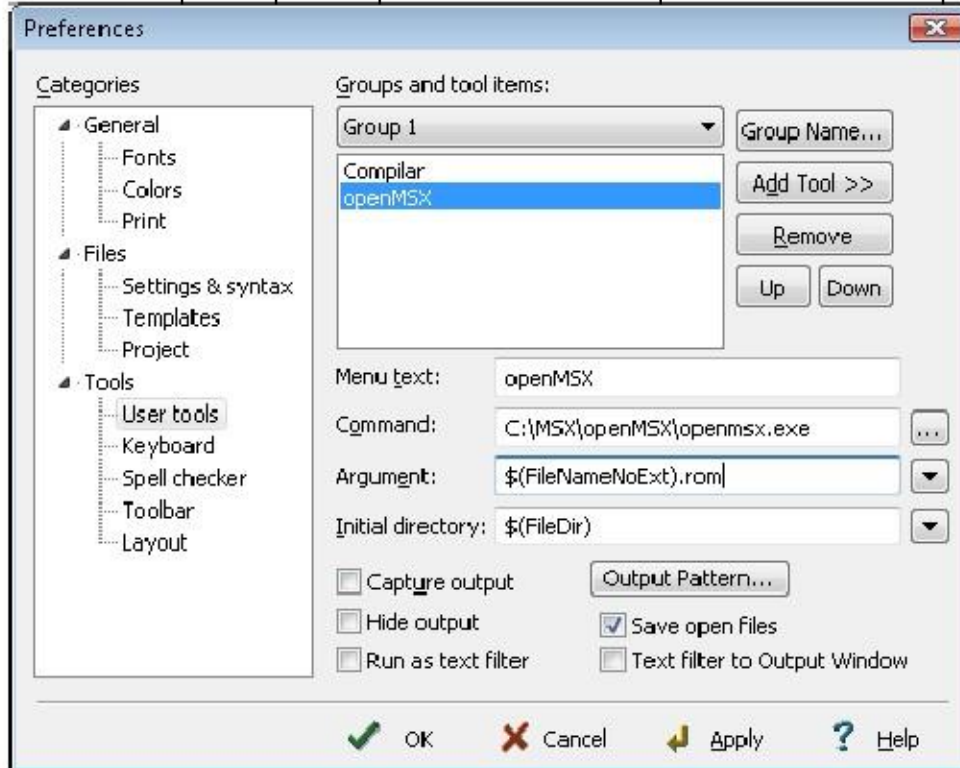


Nous allons indiquer ici le chemin de notre assembleur [C:\msx\asmsx012e\dist012e\asMSX.exe](#)  
 Dans l'option [Argument](#): appuyez sur la flèche et sélectionnez [File Name](#)  
 Dans l'option [Initial directory](#): appuyez sur la flèche et sélectionnez [File Directory](#)



Voici le résultat final que vous devez obtenir pour avoir le bouton 'Compiler'. Appuyez sur le bouton [Apply](#)

Nous allons répéter l'opération pour le second bouton qui lance la ROM dans openMSX.



Appuyez de nouveau sur le bouton [Add Tool](#) et dans la nouvelle fenêtre choisissez [Program](#)

Dans [Menu text](#): tapez [openMSX](#)

Dans [Command](#): appuyez sur les 3 points ... et sélectionnez le répertoire où est installé openMSX.

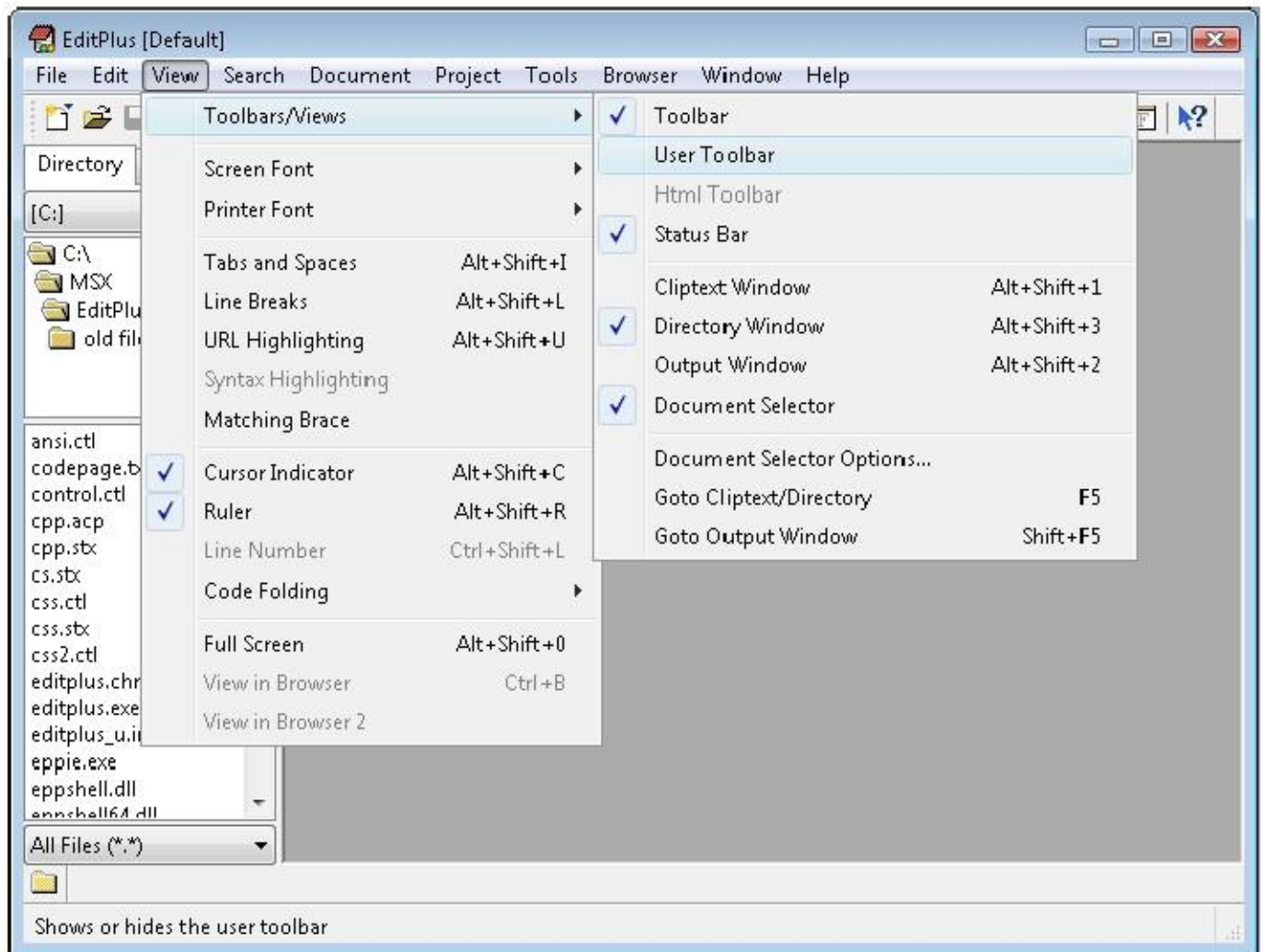
Qui doit être [C:\MSX\openMSX\openmsx.exe](#)

Dans [Argument](#): appuyez sur la flèche et sélectionnez [File Name Without Extension](#) et ajoutez [.rom](#) à la fin.

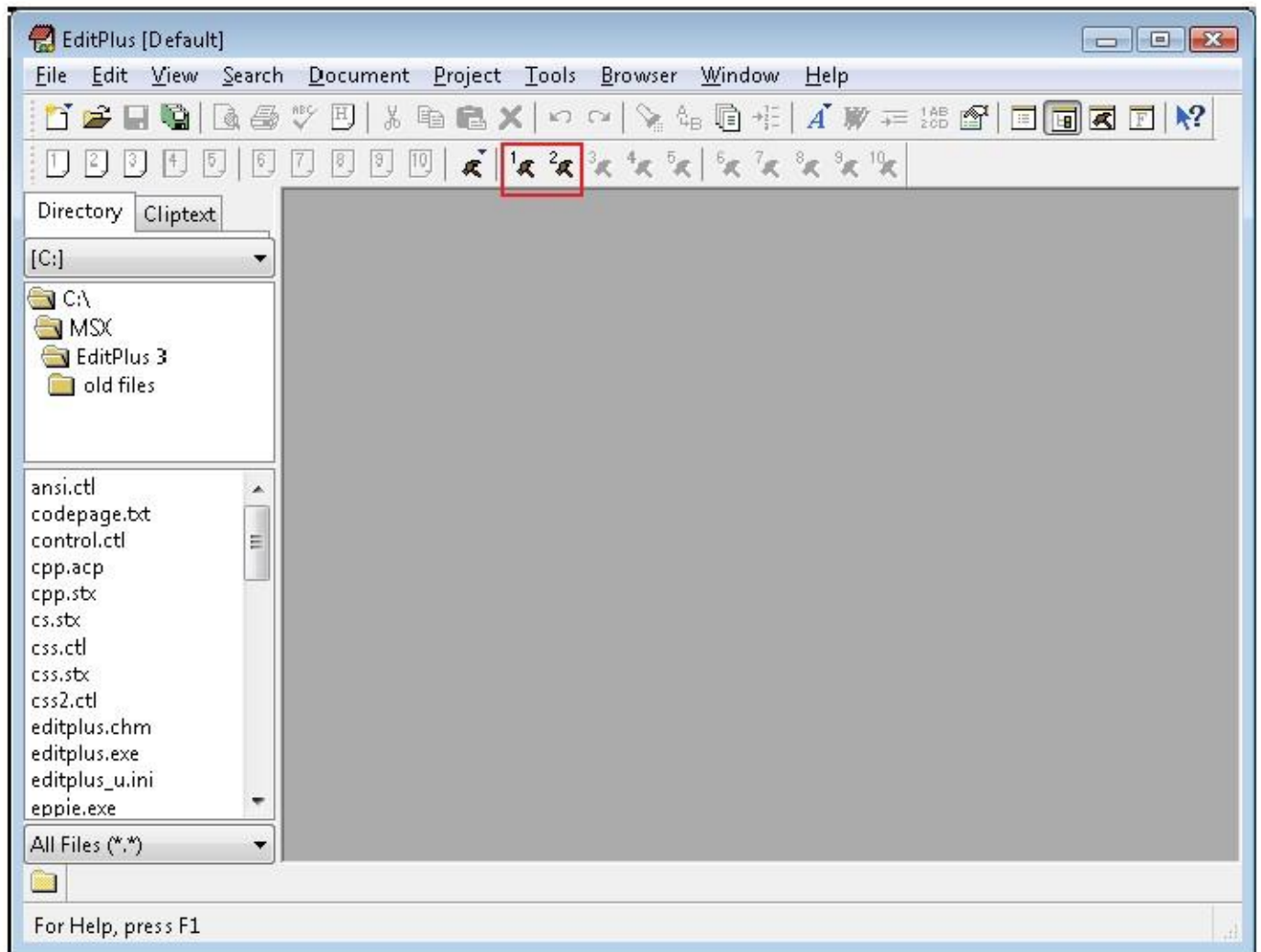
Dans l'option [Initial directory](#): appuyez sur la flèche et sélectionnez [File Directory](#)

Dans l'image ci-dessus, voici à quoi doit ressembler le résultat.

On peut alors appuyer sur le bouton [Apply](#) et le bouton [OK](#)

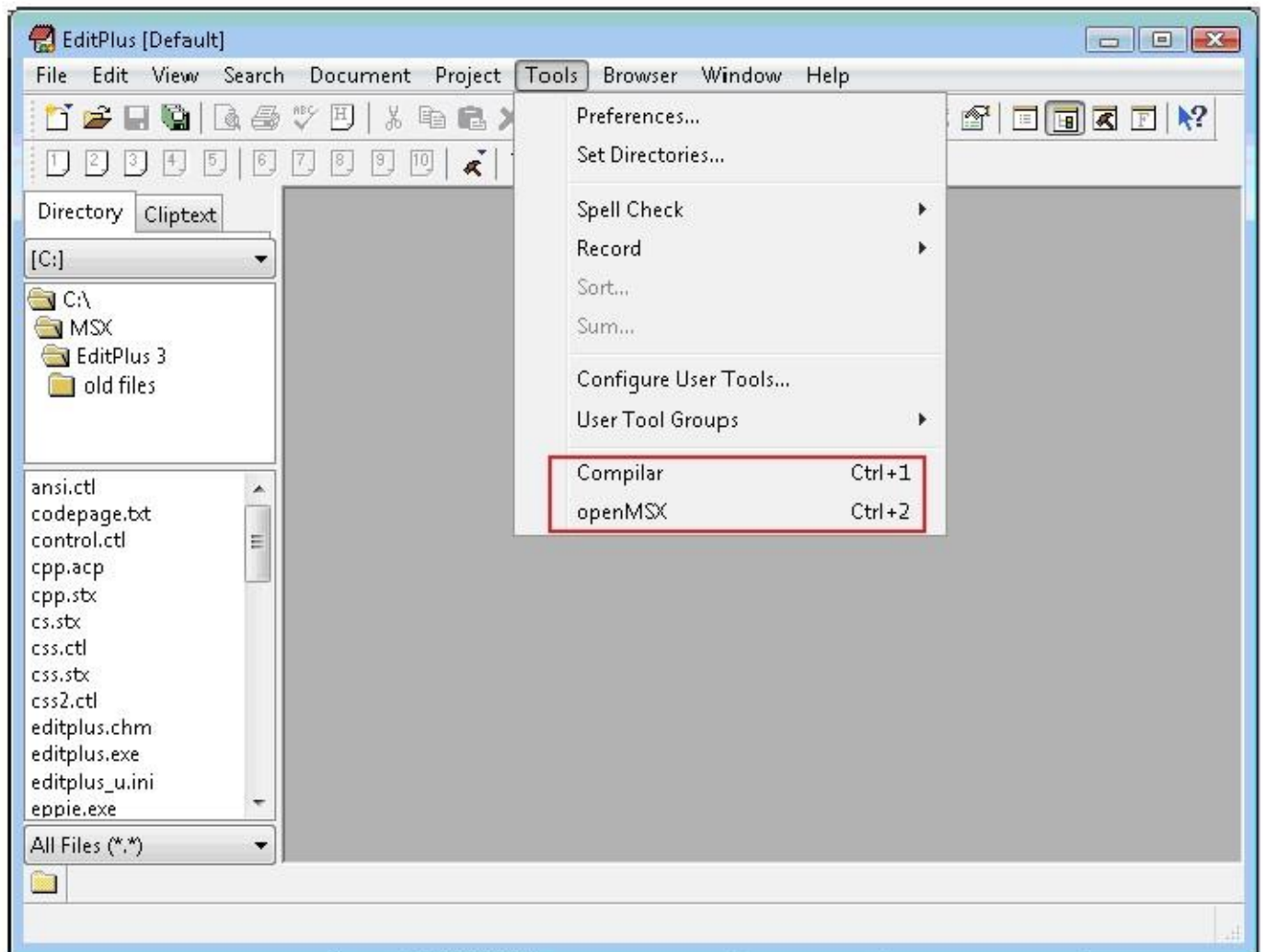


On va maintenant afficher dans la barre des icones nos 2 boutons qui sont [Compiler](#) et [openMSX](#). Dans le menu, sélectionnez [view – Toolbars/Views – User Toolbar](#) pour afficher la User Toolbar.



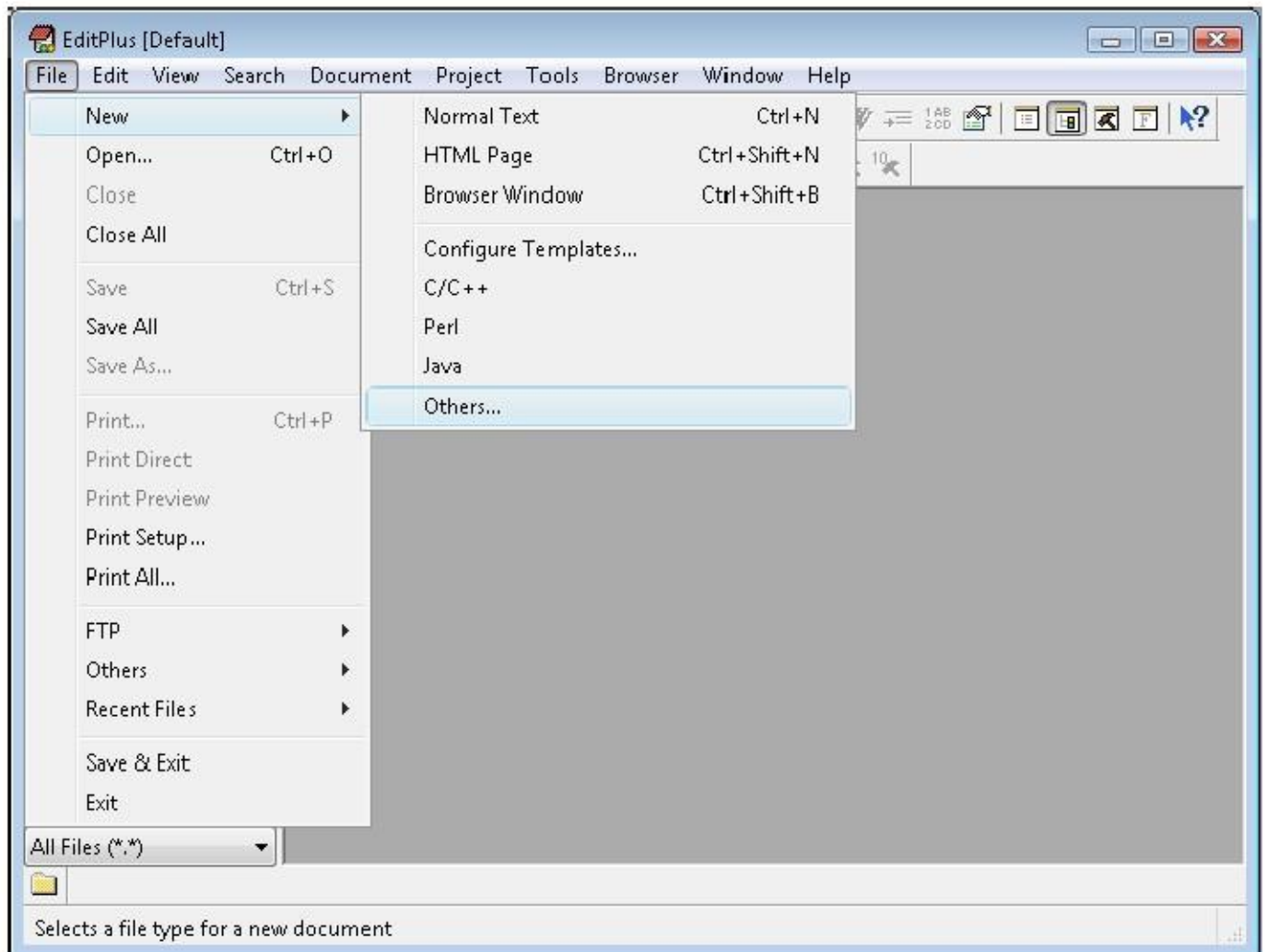
Et voilà... On peut admirer nos 2 boutons [Compiler](#) et [openMSX](#) encadrés en rouge sur l'image ci-dessus : marteau 1 et marteau 2.



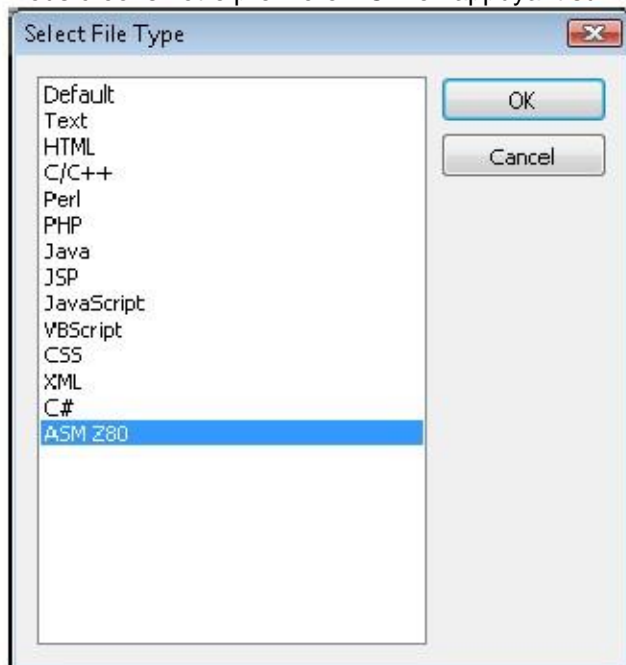


On peut aussi appeler ces 2 boutons via le menu **menu – Tools** ou par les touches de raccourci **Control+1** et **Control+2**

Nous allons vérifier si cela fonctionne en créant notre première ROM d'MSX avec un Hola Mundo qui utilise le BIOS.



Nous créons notre première ROM en appuyant sur le menu **File – New – Others**



Sélectionnez **ASM Z80** et appuyez sur **OK**.

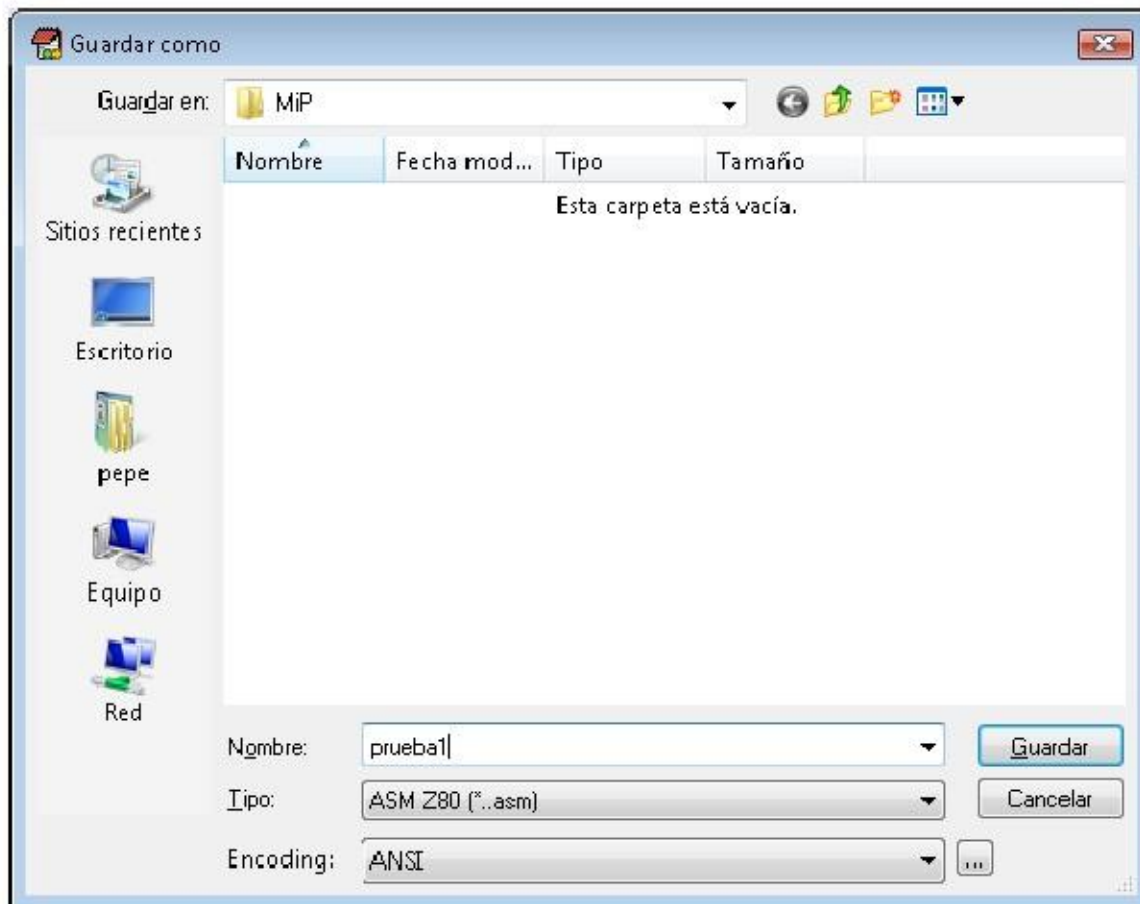
Maintenant copiez/collez le texte ci-dessous dans l'éditeur :

```
;-----
; Nom de notre programme
; Hola Mundo
;-----
; DEFINIR CONSTANTES
;-----
; Nous n'avons pas défini de constantes
;
; Variables du système
;         FORCLR equ F3E9h                ; Foreground colour
;-----
; DIRECTIVES POUR L'ASSEMBLEUR ( asMSX )
;-----
.bios                ; Définir les noms des appels au BIOS
.page 2              ; Définir que le code démarrera à l'adresse 8000h
.rom                  ; on indique que l'on crée une ROM
.start INICIO         ; Début du code de notre programme

;-----
INICIO:
; DEBUT DU CODE
;-----
;
;         Call    INIT_MODE_SC0          ; initialise le mode d'écran
;         Call    IMPRI_MENSAJE          ; imprime un message à l'écran
FIN:
;
;         Jp      FIN                    ; idem à 100 GOTO 100
;-----
INIT_MODE_SC0:
; INITIALISE LE MODE ECRAN
;-----
; BASIC: COLOR 15,0,0
; Passe le fond en noir
;
;         ld      hl,FORCLR              ; Couleur de premier plan
;         ld      [hl],15                ; blanc
;         inc     hl                     ; Couleur de fond
;         ld      [hl],1                 ; noir
;         inc     hl                     ; Couleur du bord
;         ld      [hl],1                 ; noir
;
;         call    INITXT                  ; set SCREEN 0
; call INIT32; set SCREEN 1
; call INIGRP; set SCREEN 2
; call INIMLT; set SCREEN 3
; SCREEN 0 : texte de 40 x 24 sur 2 couleurs
; SCREEN 1 : texte de 32 x 24 sur 16 couleurs
; SCREEN 2 : graphiques de 256 x 192 sur 16 couleurs
; SCREEN 3 : graphiques de 64 x 48 sur 16 couleurs
;
;         ret
;-----
;
;-----
IMPRI_MENSAJE:
; ROUTINE QUI IMPRIME LE MESSAGE A L'ECRAN
;-----
;
;
;         ld      h,01                    ; positionne la colonne
;         ld      l,01                    ; et la ligne
; ldhl,0101h ; on pourrait aussi le faire de cette manière
;         call    POSIT                    ; on positionne le curseur au début de l'endroit où on veut écrire
;         ld      hl,texto                 ; on charge hl avec le texte du message
;
; bucle:
;         ld      a,[hl]                   ; on charge le premier caractère dans a
;         or      a                         ; on vérifie si on a atteint la fin du texte (0)
;         ret     z                         ; et on quitte la routine dans le cas où la comparaison est zéro
;         call    CHPUT                    ; on écrit le caractère à la position du curseur
;         inc     hl                       ; on incrémente hl pour qu'il pointe à la lettre suivante
;         jr      bucle                    ; si on est pas arrivé à la fin on recommence la routine
;-----
;
; texto:
;         .db "Hola Mundo",0
```

FINAL DEL TEXTO QUE TIENES QUE COPIAR menos esta línea

Nous sommes prêt à compiler et exécuter la compilation avec openMSX  
Ainsi, appuyez sur le bouton que nous avons créé Compiler ou le bouton défini 1

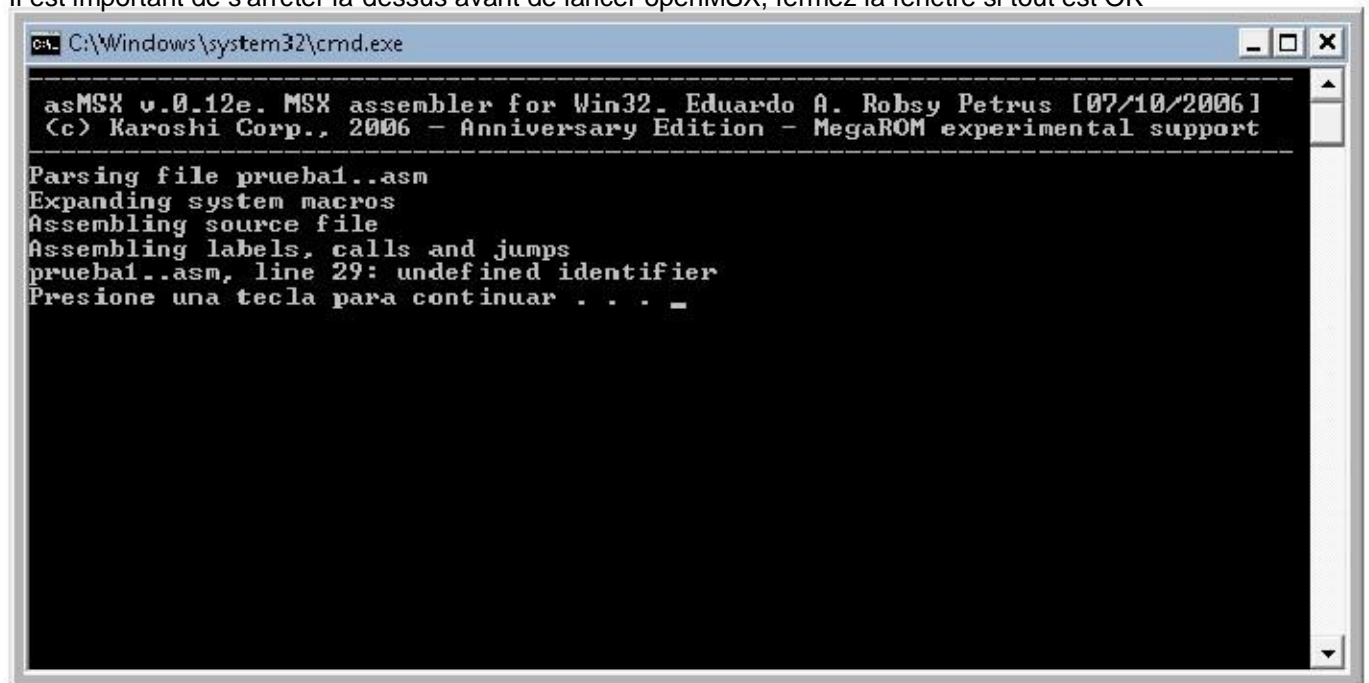


La première fois que l'on compile il faut sauvegarder auparavant le fichier **.ASM** , j'ai créé un répertoire appelé MiP et j'ai nommé le fichier **holamundo.asm** appuyez sur le bouton **Sauvegarder**.



On peut observer que **asMSX** a compilé notre code en assembleur sans erreurs.

Il est important de s'arrêter la-dessus avant de lancer openMSX, fermez la fenêtre si tout est OK



```
C:\Windows\system32\cmd.exe

asMSX v.0.12e. MSX assembler for Win32. Eduardo A. Robsy Petrus [07/10/2006]
(c) Karoshi Corp., 2006 - Anniversary Edition - MegaROM experimental support

Parsing file prueba1.asm
Expanding system macros
Assembling source file
Assembling labels, calls and jumps
prueba1.asm, line 29: undefined identifier
Presione una tecla para continuar . . . _
```

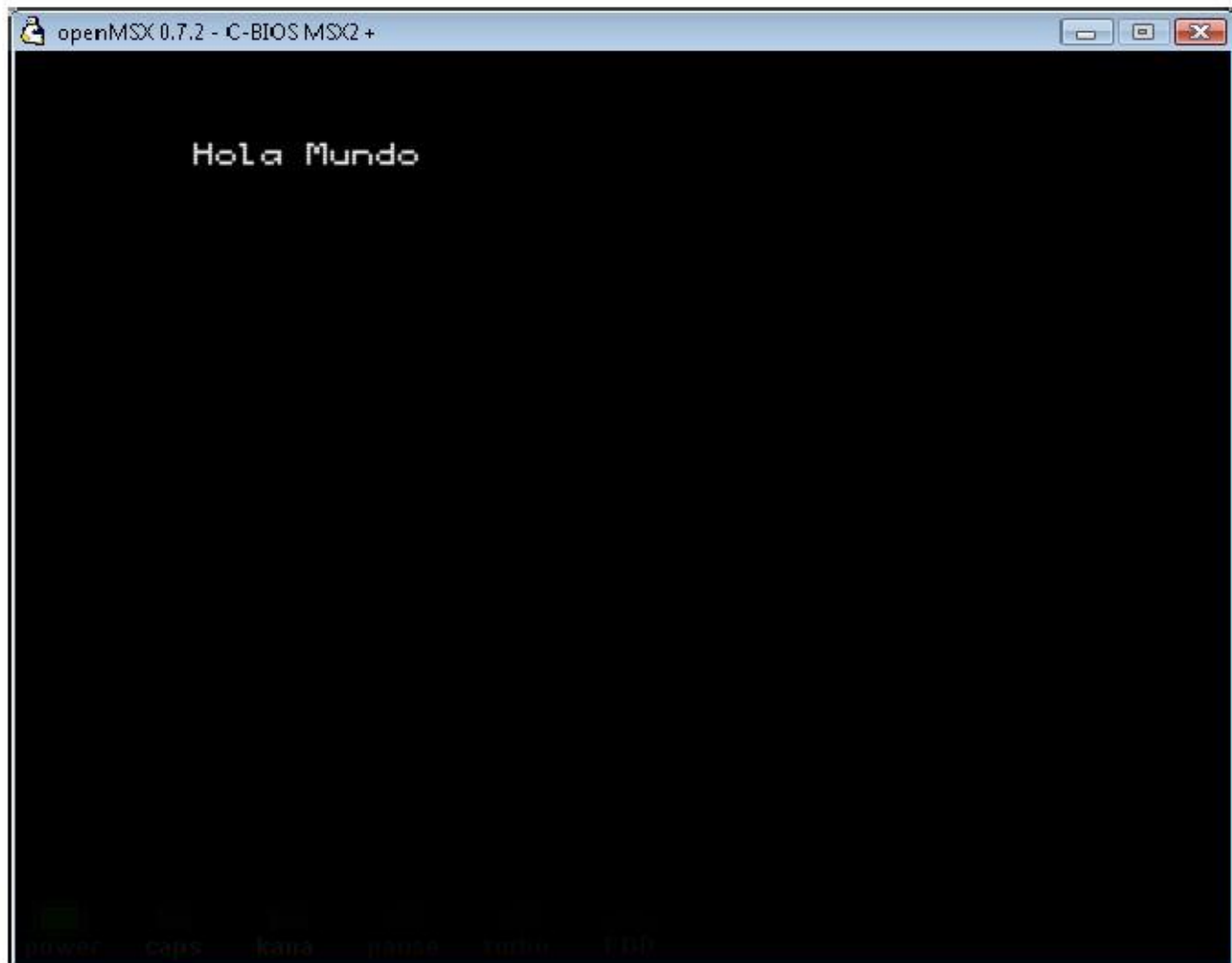
Ci-dessus j'ai volontairement provoqué une erreur de syntaxe pour que vous voyez le cas d'erreur, asMSX vous donne la ligne de l'erreur dans votre code, ici il faut revenir sur la ligne 29 dans l'éditeur de texte pour corriger l'erreur.

MAINTENANT QUE NOUS AVONS LE CODE NOUS ALLONS LE VERIFIER

Appuyez sur le bouton de lancement d'openMSX ou le bouton défini 2



On peut voir qu'openMSX charge son BIOS comme si c'était un vrai MSX2



Et voilà on obtient le résultat final.

On peut fermer l'émulateur et les autres fenêtres noires qui servent de lanceur.

Nous avons créé notre première ROM pour MSX avec une incroyable facilité. Vous pouvez maintenant modifier la colonne et la ligne de début du texte à l'écran très simplement et relancer une compilation et le vérifier dans l'émulateur.

J'espère que cela vous est utile et on se retrouve pour le prochain tutoriel où je vous expliquerai plus en détail le code que nous venons de voir dans cet article et les routines du BIOS MSX et ensuite comment vérifier notre code via le débogueur de blueMSX

P.D. Le reste des programmes du Pack MSX sont pour de prochains chapitres .

José Vila Cuadrillero

"ES DETESTABLE ESA AVARICIA ESPIRITUAL QUE TIENEN, LOS QUE SABIENDO ALGO, NO PROCURAN LA TRANSMISION DE ESOS CONOCIMIENTOS."

Miguel de Unamuno

Escritor y Filósofo.

( Bilbao 1864 - Salamanca 1936 )