

# TUTORIAL DE INICIACION A LA PROGRAMACION EN LENGUAJE ENSAMBLADOR PARA MSX

## 1ª PARTE - EL ENTORNO DE DESARROLLO

Hola a tod@s me he decidido a escribir este tutorial para todos aquellos que queréis empezar con esto de la programación para MSX, y no sabéis por donde coger el hilo o que herramientas utilizar.

Si bien no se va explicar paso a paso el lenguaje ensamblador del Z80 si se irán explicado nociones sobre este lenguaje encaminadas a la realización de videojuegos para nuestros MSX's.

Lo primero que os voy a proponer son las herramientas que yo me he montado para poder realizar esta labor de la manera mas fácil, lógicamente vosotros podréis sentirlos mas a gusto con otras o simplemente cambiarlas por alguna que ya estéis utilizando.

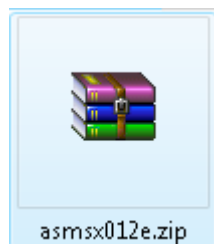
Baja este [Pack](http://www.dimensionzgames.com/wp-content/plugins/download-monitor/download.php?id=5) de recopilación de mis herramientas desde este enlace.  
<http://www.dimensionzgames.com/wp-content/plugins/download-monitor/download.php?id=5>

Crea una carpeta en tu disco duro y llámala C:\MSX y descomprime el [PACK](#) de mis herramientas dentro de esa carpeta para tenerlas todas localizadas a mano.

Para compilar el código en ensamblador que crearemos vamos a utilizar, el que para mi es el mejor compilador cruzado de Z80 para MSX, que no es otro que el [asMSX](#) de KAROSHI.

<http://karoshi.auic.es/index.php?topic=834.0> ( Podéis bajar desde aquí la última versión )

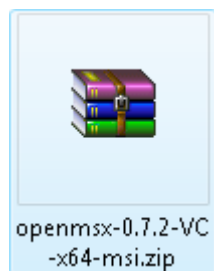
Este compilador no incluye un editor de texto que es donde escribimos el código que queremos compilar, así que buscando y buscando he decidido quedarme con el [EditPlus](#) versión 3.01 que os incluyo en el [Pack](#) y que será el punto de partida para todo nuestro desarrollo.



Vamos al directorio [MSX](#) donde tenemos el [Pack](#) y pulsa doble clic sobre el fichero [asmsx012e.zip](#) y descomprímelo dentro de la carpeta [C:\MSX](#) en esta carpeta el descompresor te ha creado una carpeta [asmsx012e](#)

[C:\MSX\asmsx012e](#)

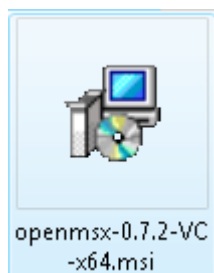
Aquí en este directorio estará nuestro Ensamblador con todos sus ejemplos.



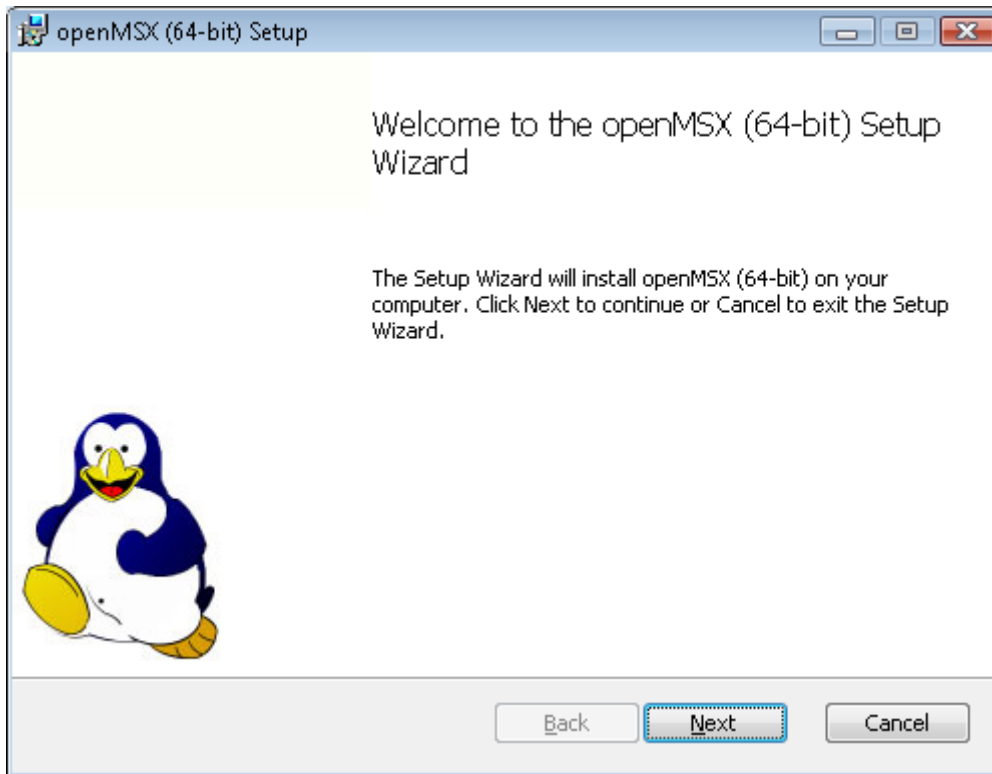
Ahora vamos a descomprimir el Emulador de nuestro querido MSX donde probaremos nuestro código una vez compilado, lógicamente para mí el mejor emulador es el [BlueMSX](#) pero para lanzar y probar el compilado sobre el emulador uso el [OpenMSX](#) por la facilidad de lanzar o abrir el fichero compilado con el emulador.

Pero para probarlo bien y depurar el código en busca de errores uso el [BlueMSX](#).

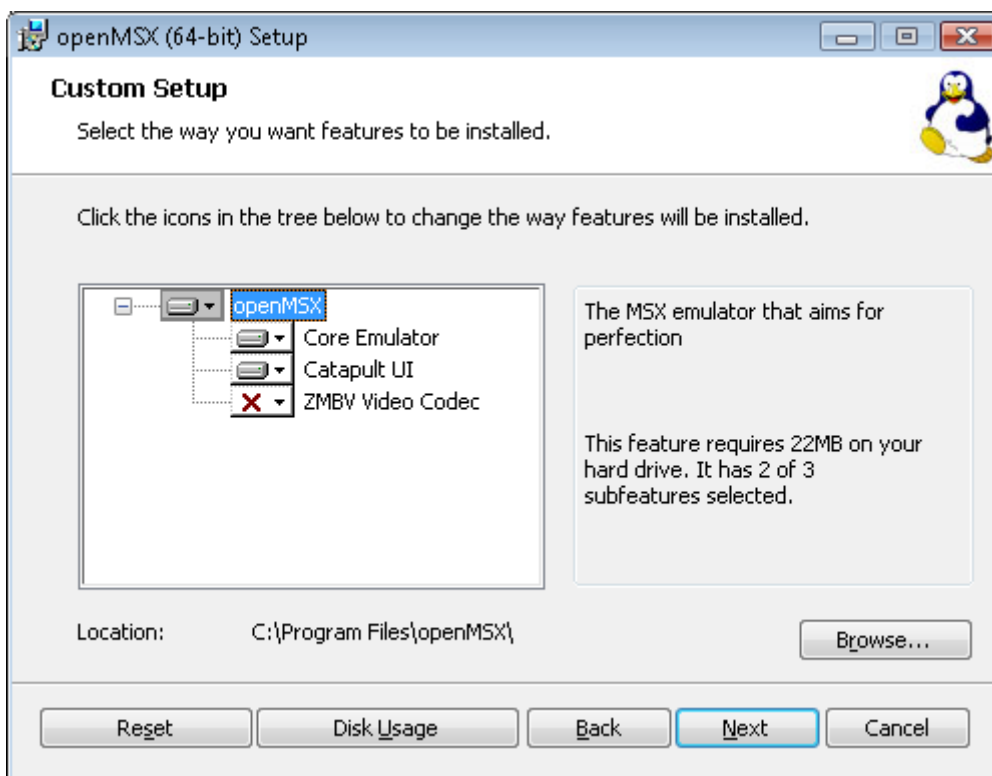
Vamos a descomprimir el OpenMSX dentro de la carpeta del Pack, veras que hay dos ficheros uno [-x64](#) si usas Windows de [64Bits](#) y [-x86](#) si usas Windows de [32 Bits](#).



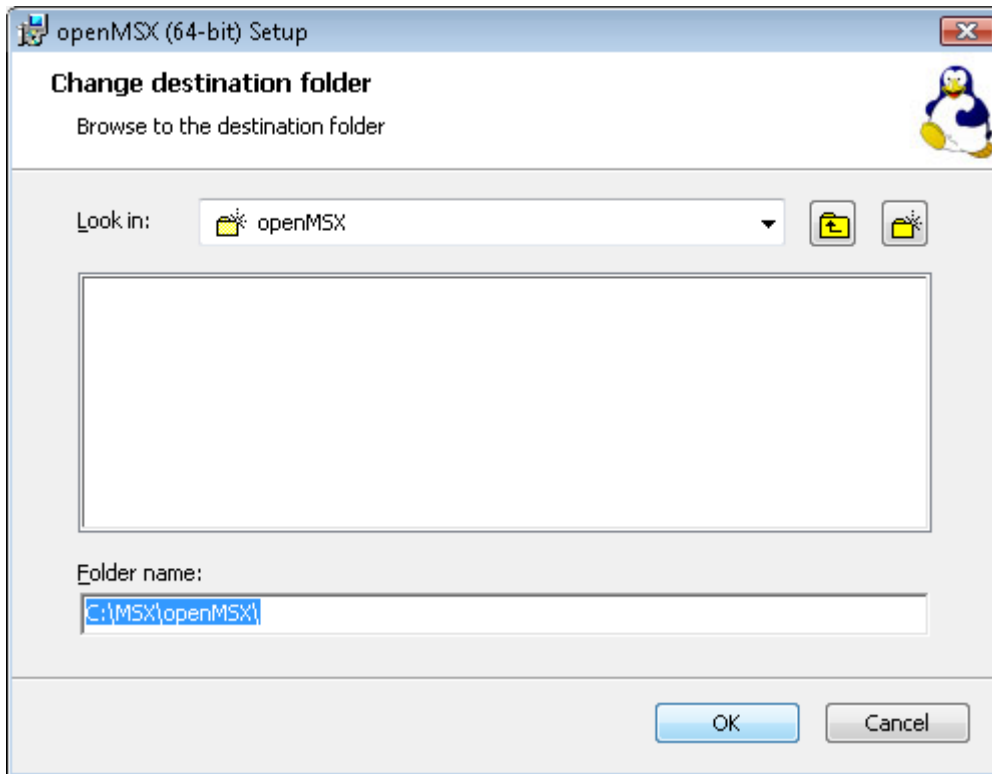
Después de que hayas descomprimido el OpenMSX según tu versión de Windows veras dentro de la carpeta del Pack, el instalador del OpenMSX versión 0.7.2 en mi caso uso Windows 7 Ultimate 64 Bits por eso he descomprimido la versión de 64 Bits. Obvia decir que esta versión es la existente en el momento que se creó este manual. Pulsa doble clic sobre el icono para que comience la instalación.



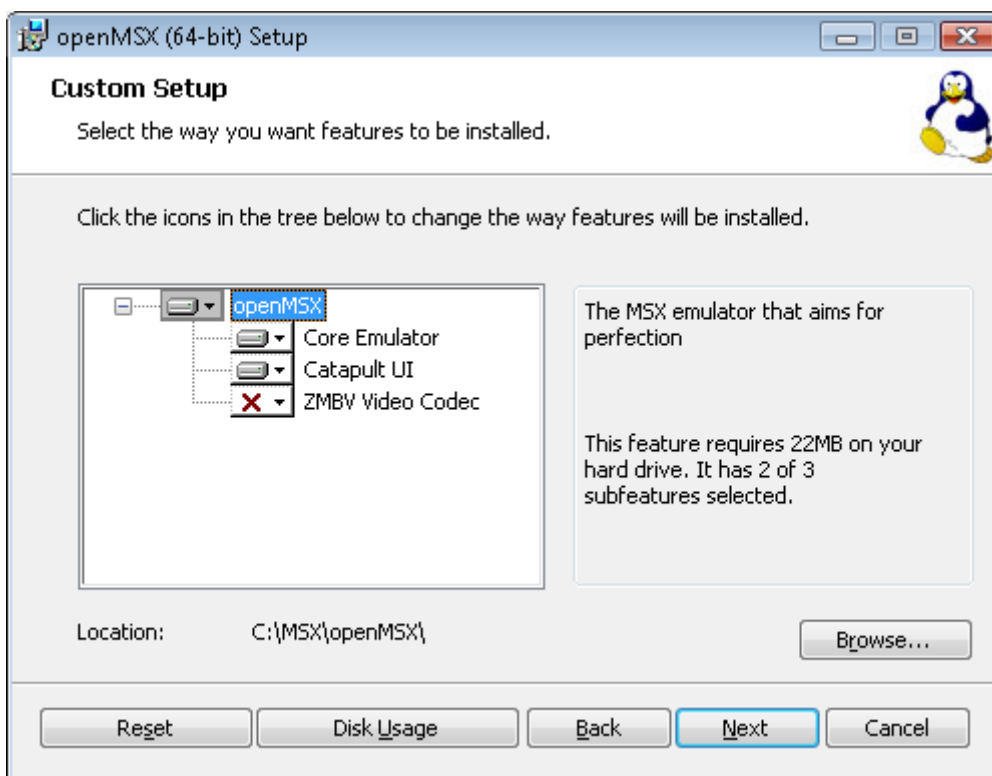
Este el instalador del [OpenMSX](#) pulsamos en el botón “Next”



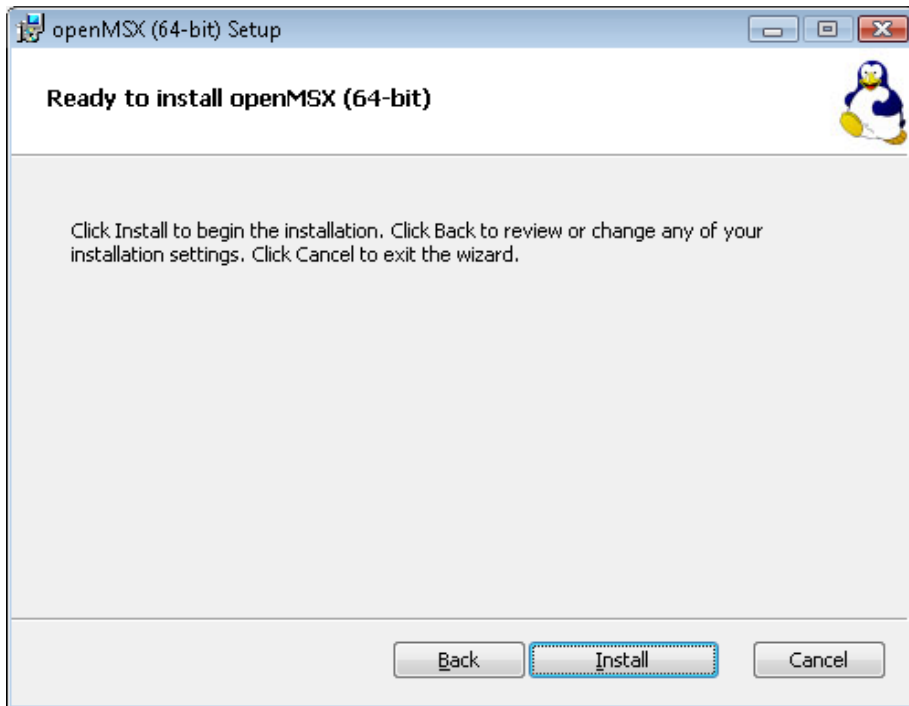
Esta parte es importante cambiar el directorio de Instalación para que lo instale dentro del directorio [c:\MSX](#) ya que después en las macros de nuestro editor lo buscaremos ahí dentro. Pulsa en “Browse”



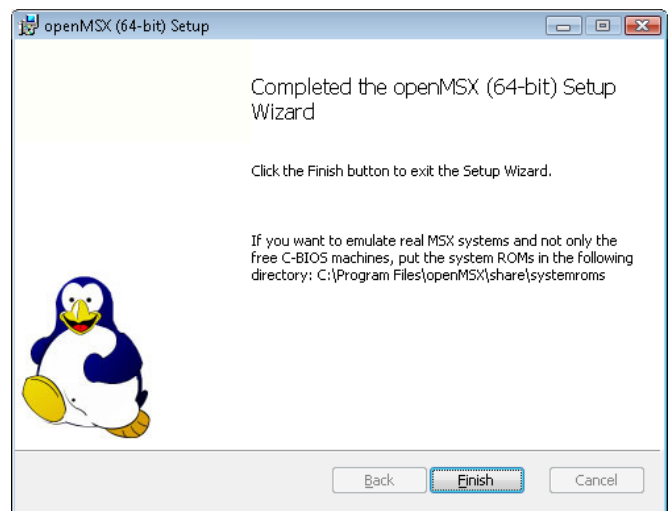
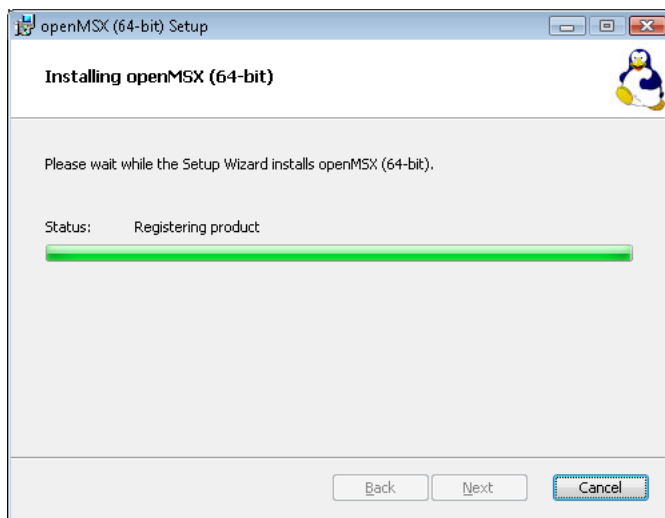
Aquí modifica el **Folder name**: y escribe **C:\MSX\openMSX** y pulsa el botón "OK"



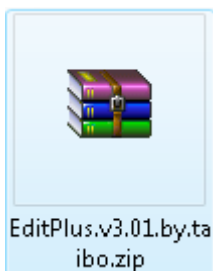
Aquí puedes ver que el directorio ha cambiado y pulsamos el botón "Next"



Finalmente pulsamos en el botón “Install”



Esperas a que se instale, veras estas imágenes cuando pulses el botón “Finish” ya estará instalado.

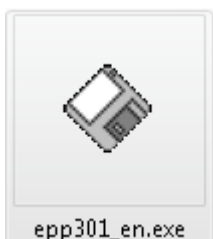


EditPlus.v3.01.by.taibo.zip

Ahora ya podemos instalar el que será el órgano común de nuestra forma de trabajar. Descomprime el fichero EditPlus.v.3.01.by.taibo.zip dentro de la carpeta del Pack.

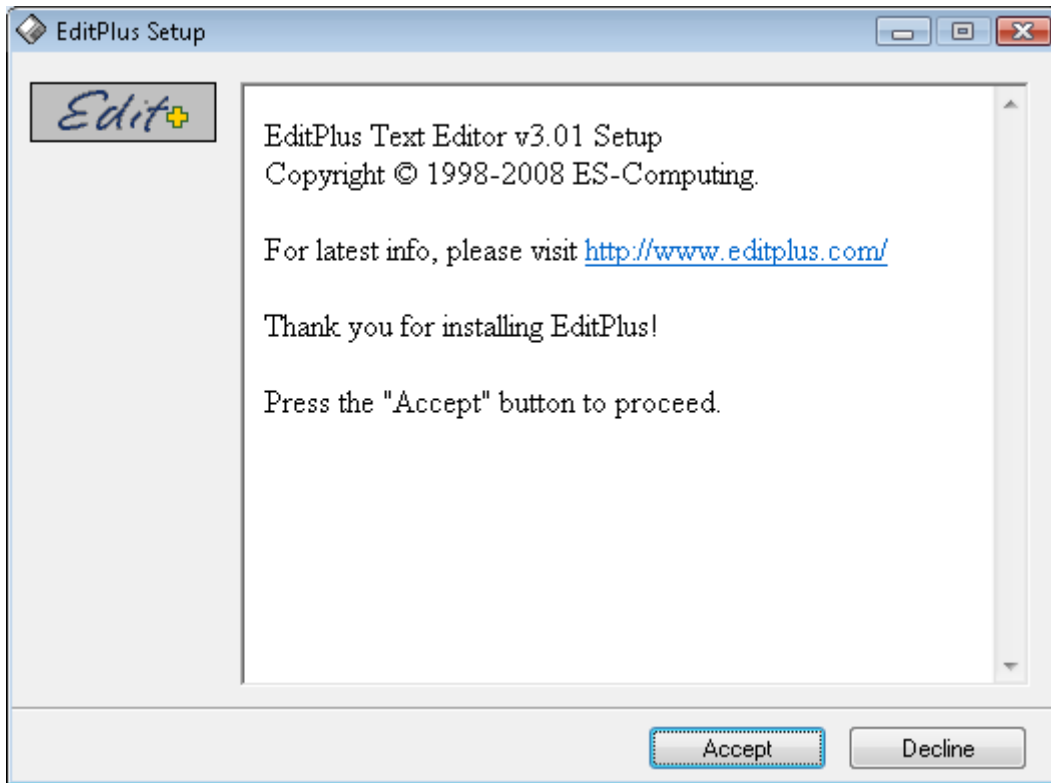
Este es lamentablemente un software de pago, pero merece la pena que lo instales por todas las ventajas que nos da. Lee las instrucciones dentro del txt. del zip

Dentro de la carpeta del Pack donde has descomprimido el EditPlus veras este icono.

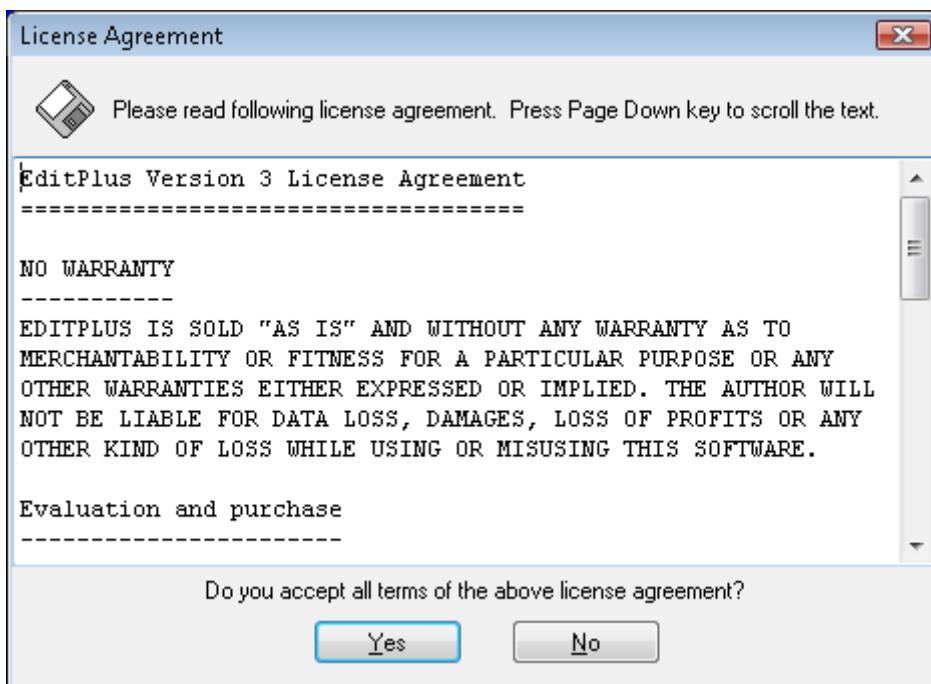


epp301\_en.exe

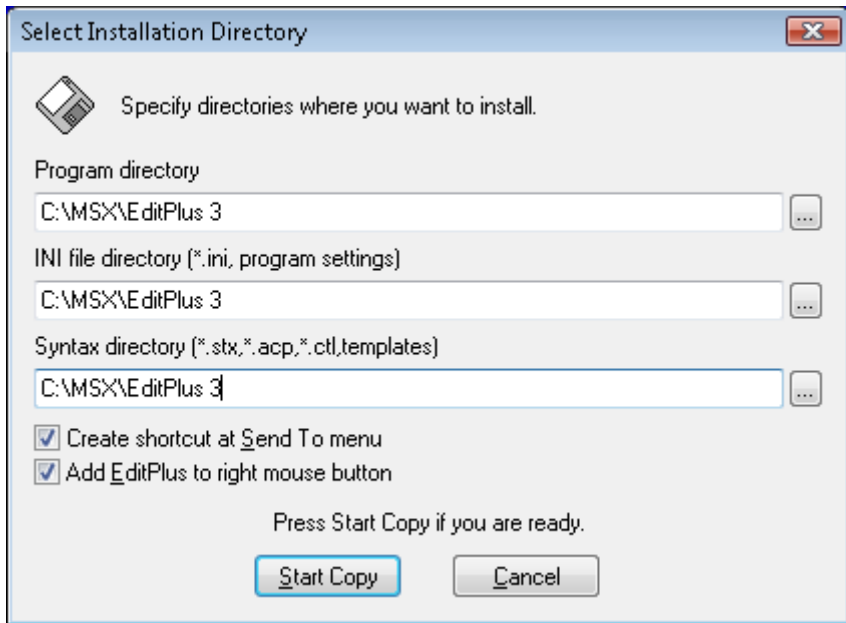
Pulsa doble clic sobre este icono para instalarlo.



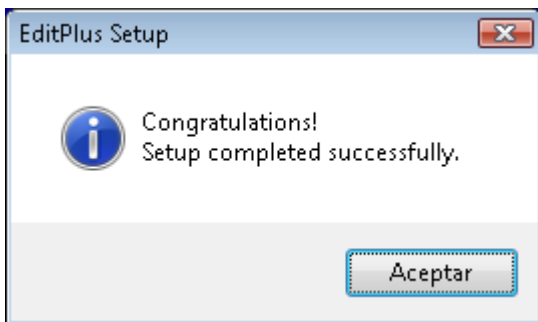
Pulsamos el botón "Accept"



Pulsa el botón "Yes" para aceptar la licencia.

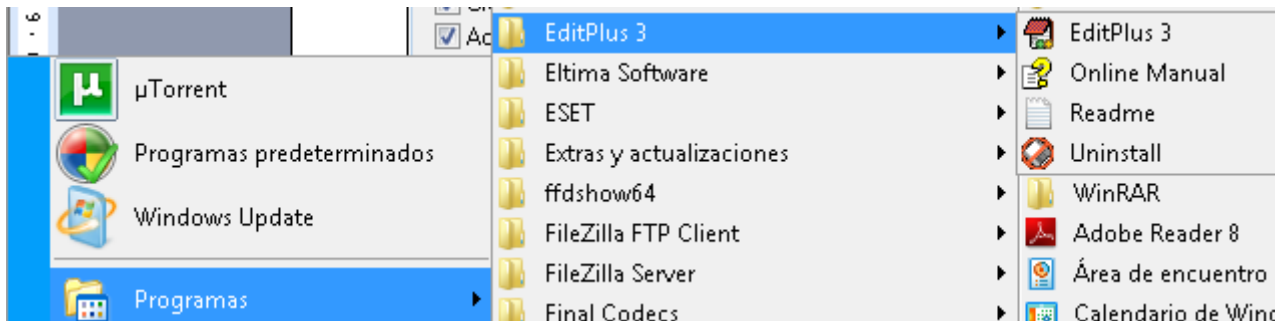


Coloca o escribe los 3 directorios apuntando a [C:\MSX\EditPlus 3](#) como ves en la imagen.

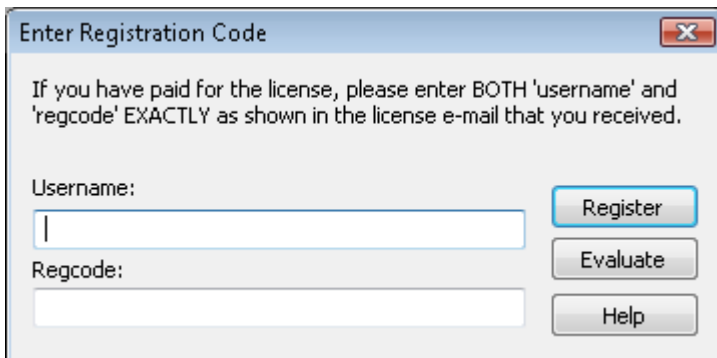


Ya lo tenemos instalado.

Pulsamos el botón “Aceptar”

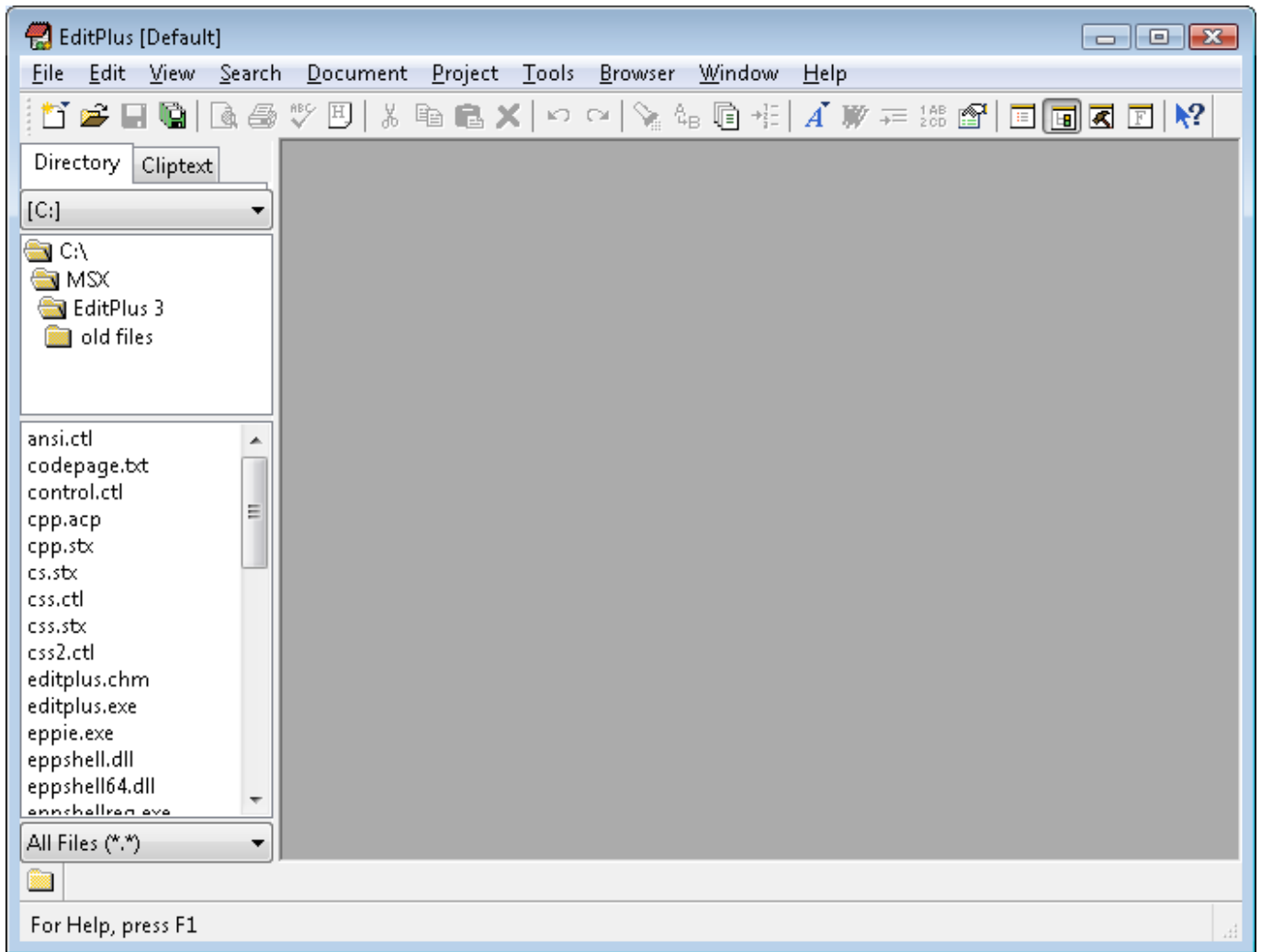


Ahora lo ejecutamos como administrador pulsa en [Programas – EditPlus 3 – EditPlus 3](#), con el botón derecho del ratón y pulsa [ejecutar como administrador](#), si lo haces en Windows vista o Windows 7 si usas Windows XP no hace falta, te recomiendo copiar este acceso al escritorio.

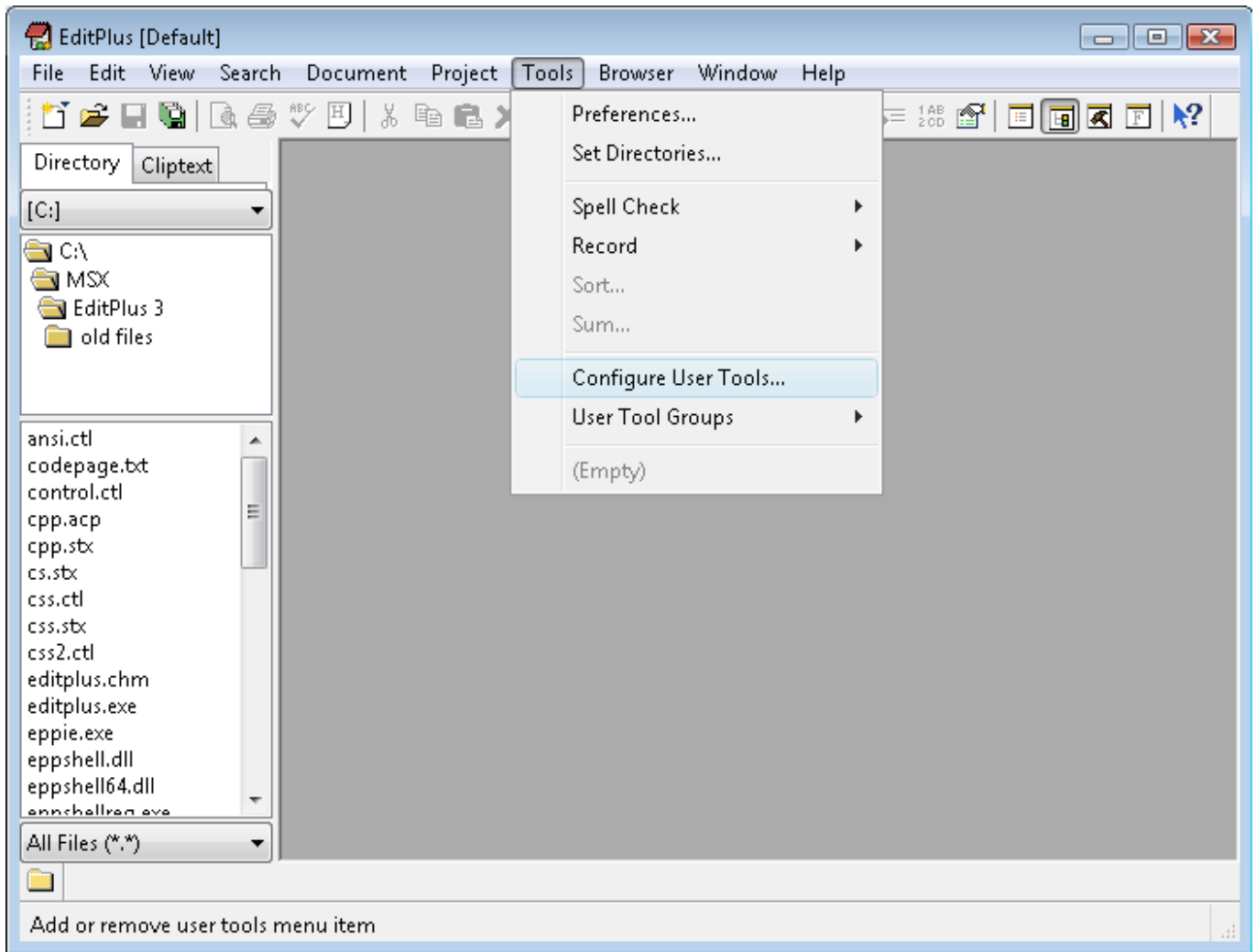


Si tienes los datos para el registro es el momento de introducirlos y Pulsar el botón “[Register](#)” si has leído el .txt te ayudará.

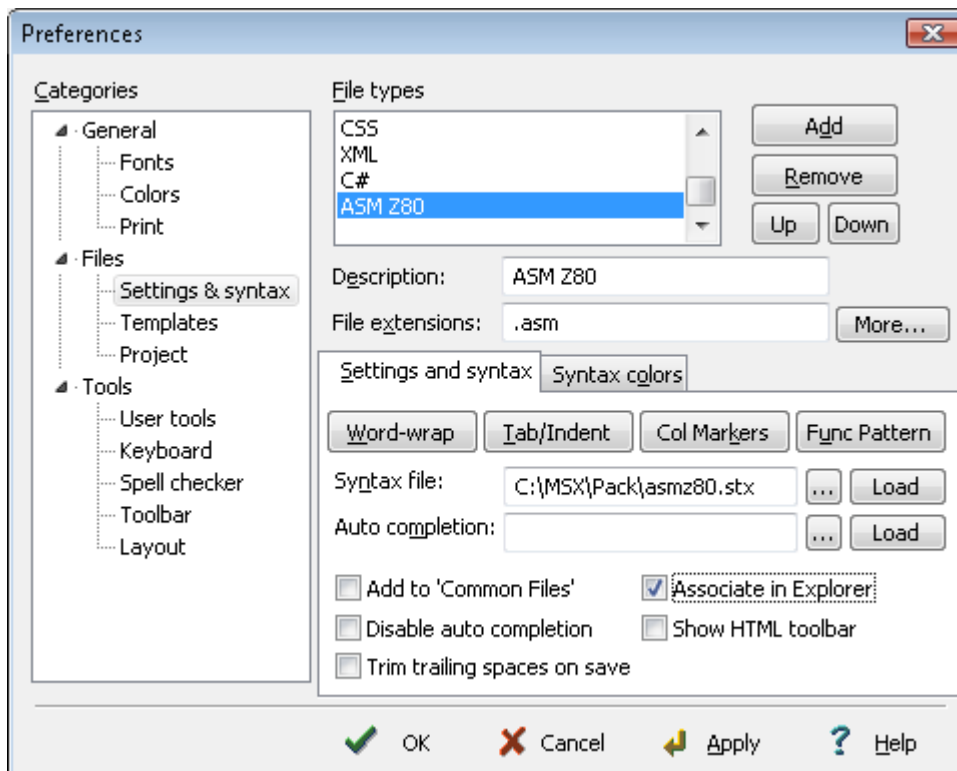
De lo contrario pulsa el botón “[Evaluate](#)” para que te dejen usar el programa durante 30 días.



Aquí tenemos nuestro editor en todo su esplendor, ahora vamos a realizar los pasos más importantes uno será, incorporarle un añadido para que nos muestre por colores la sintaxis del lenguaje en ensamblador del Zilog Z80, el fichero que vamos a incorporar al EditPlus es [asmz80.stx](#) en el Pack.



Para agregarlo vamos al menú **Tools – Configure User Tools...**



1 - Primero en la izquierda tienes que señalar la Opción de **settings & syntax** una vez señalado.

2 - Pulsamos el botón **Add** teclaea en el campo **Description:** pon **ASM Z80** Y en **File extensions:** **.asm**

3 - Ahora pulsa en **Syntax file:** en el botón **...** y selecciona en la imagen que ves debajo de este cuadro el fichero **asmz80.stx** que esta en el directorio del Pack.

Marca **Associate in Explorer** Si ves la imagen como la foto de la izquierda ya puedes pulsar los botones **Apply** y **OK**



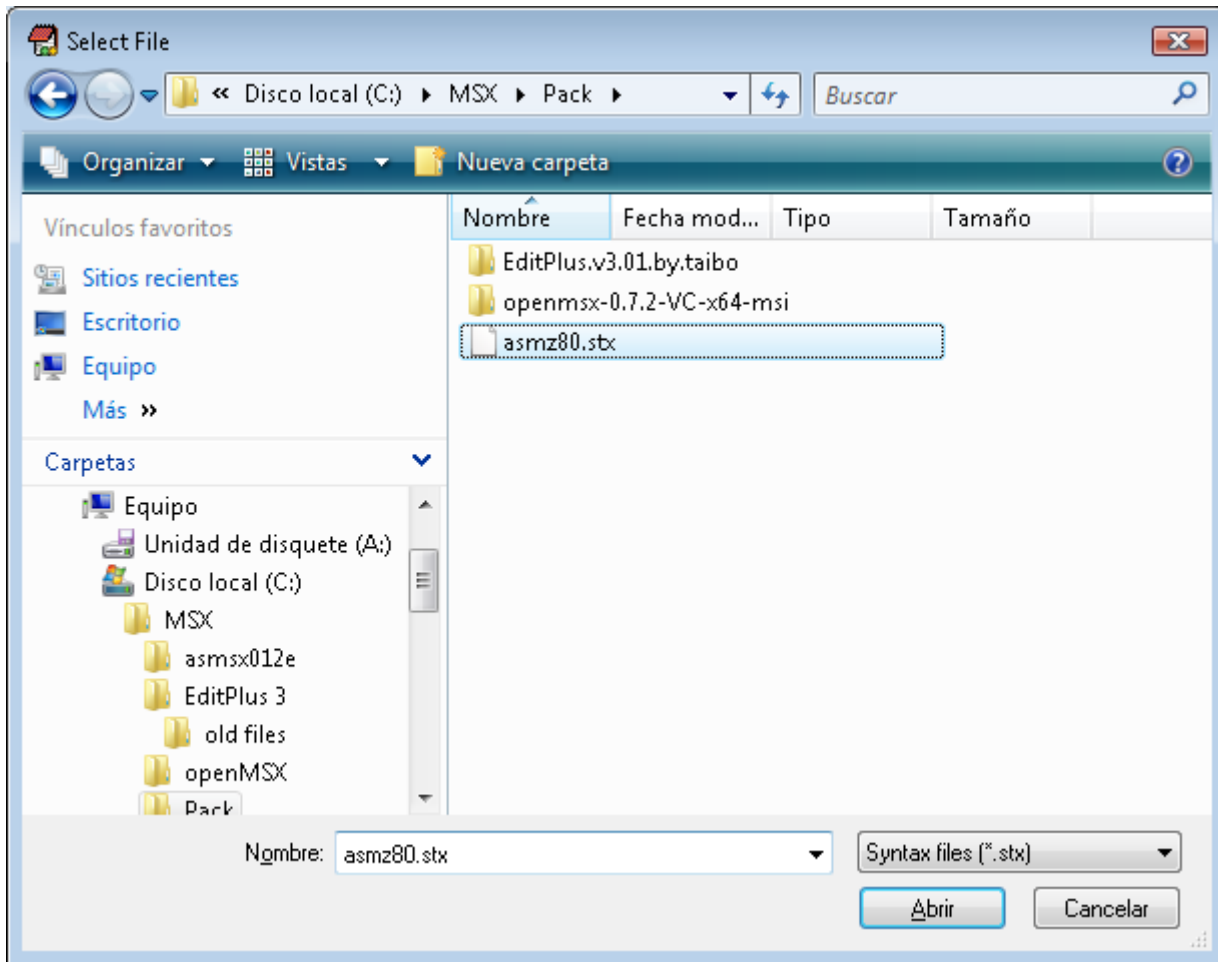
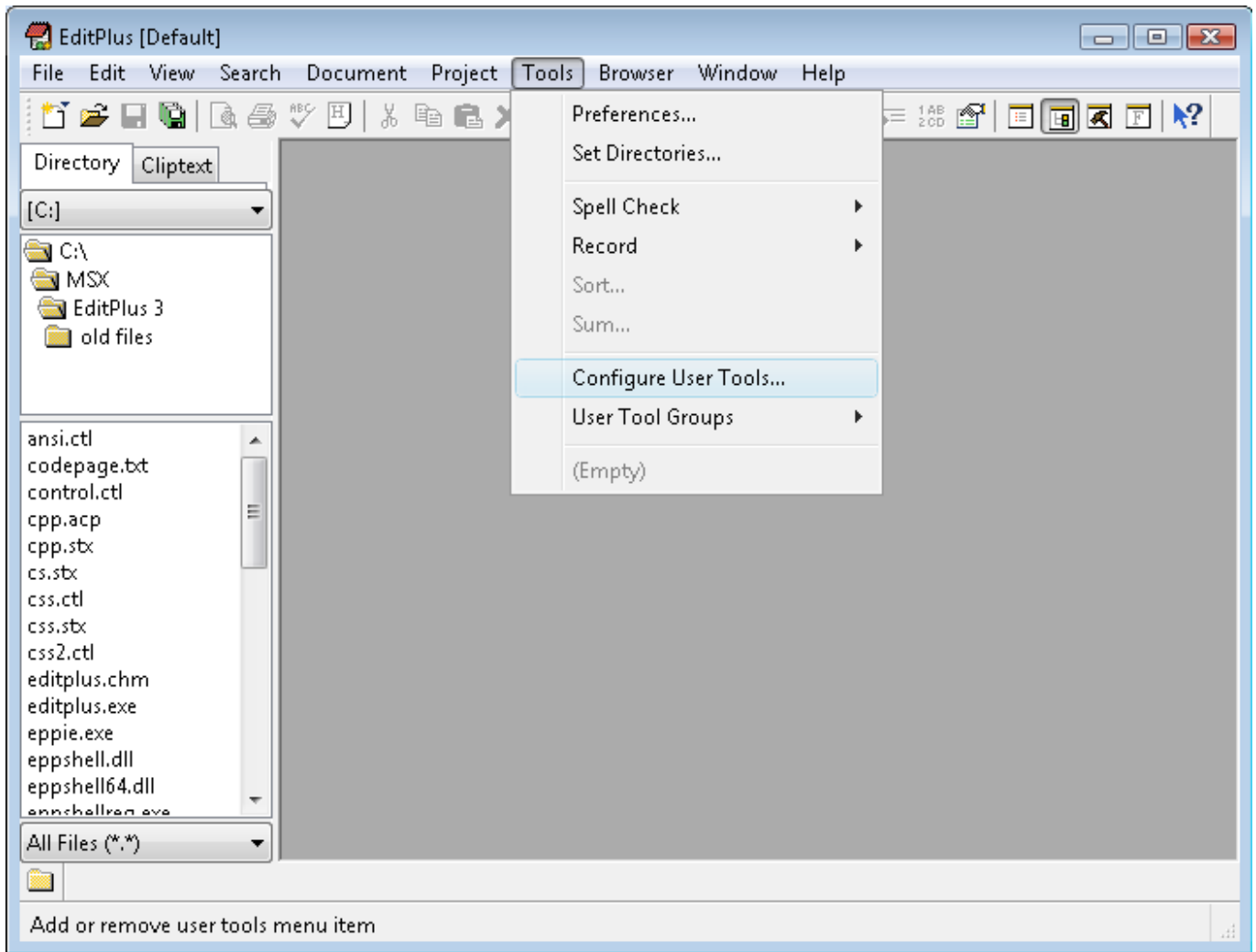
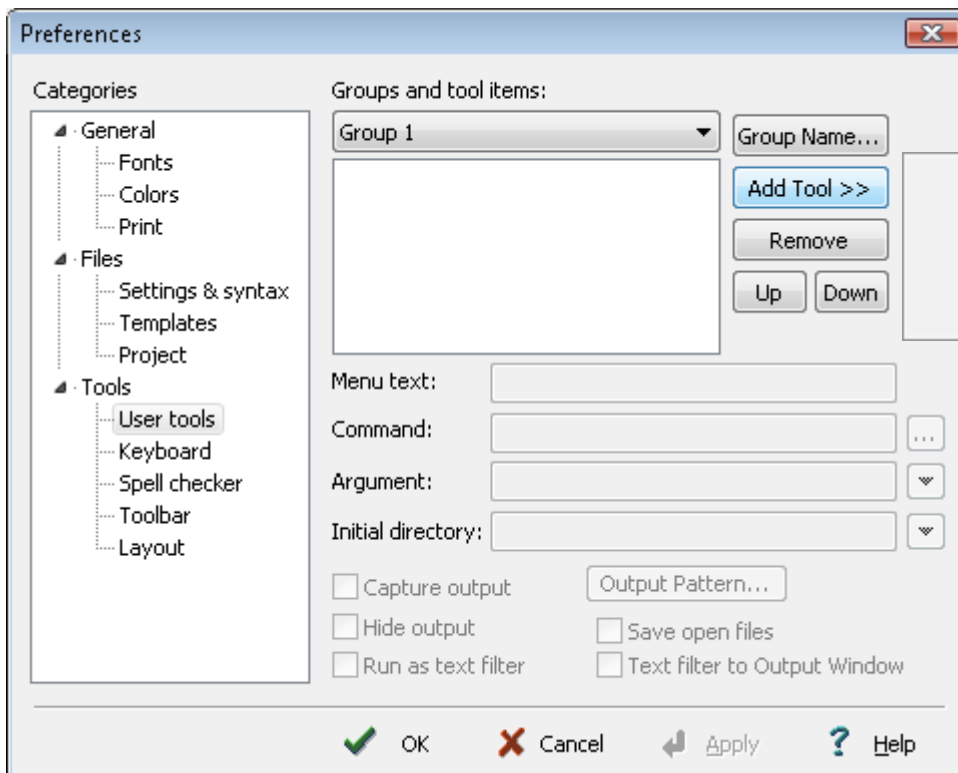


Imagen que sale cuando pulsamos en el paso 3 descrito más arriba.

Ahora vamos a crear nuestros propios botones uno será para [Compilar](#) y el otro Ejecutar el Compilado o fichero ROM en el Emulador [OpenMSX](#).



De nuevo pulsamos en el menú del EditPlus en **Tools – Configure User Tools...**



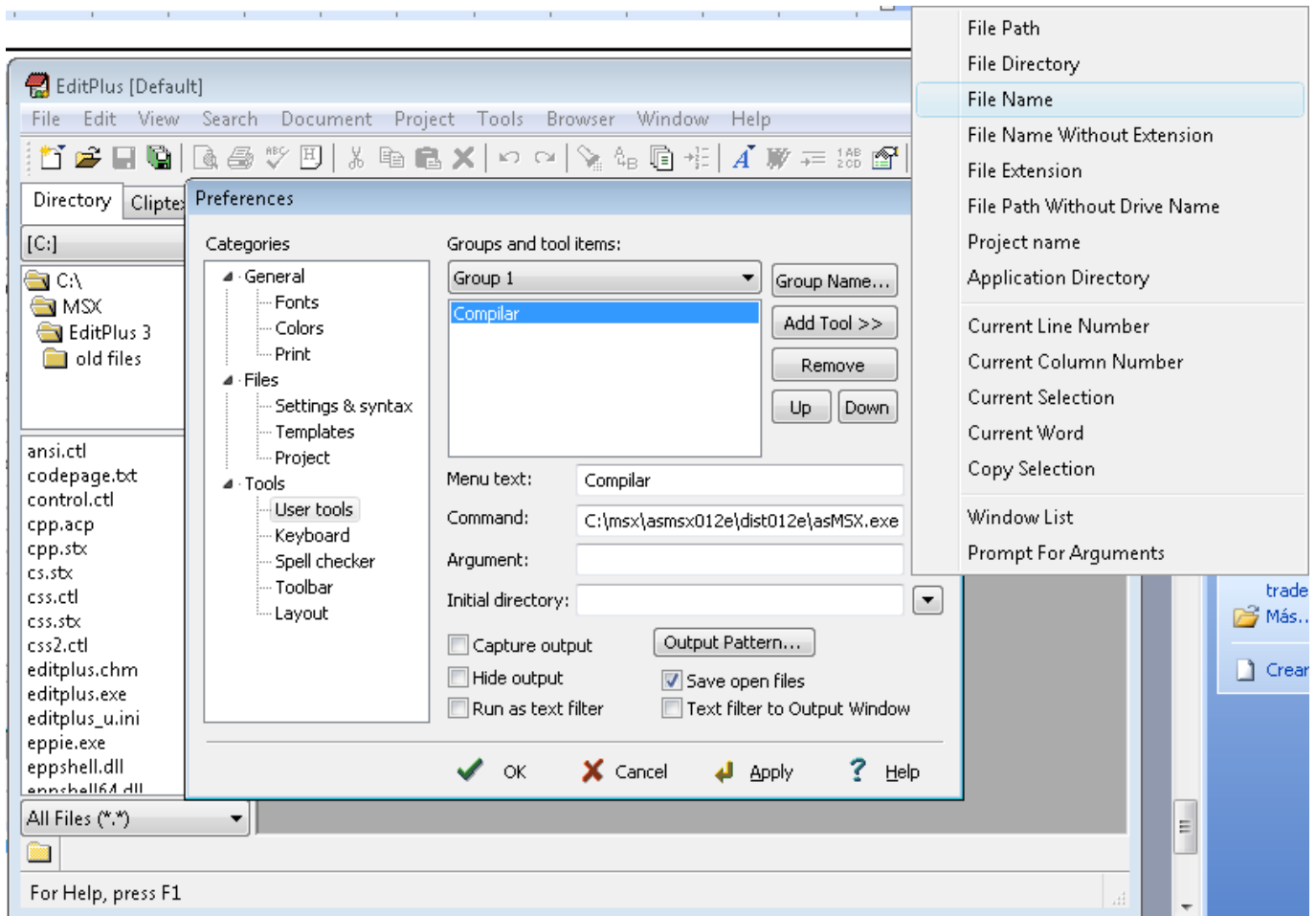
Primero en la Izquierda señala **User tools**

Ahora pulsa en el botón **AddTool** y en la ventana emergente elige la opción **Program**

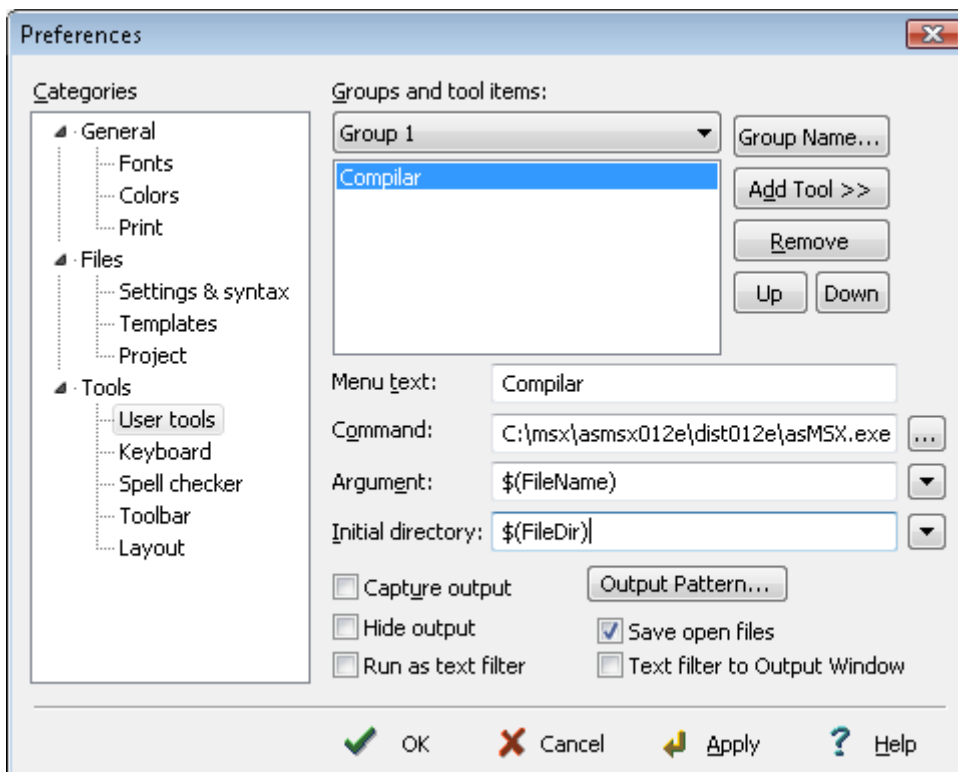
Ahora en **Menu text:** teclea **Compilar**

Ahora pulsa en **Command:** en los 3 puntos ...

Fíjate en la imagen de abajo.

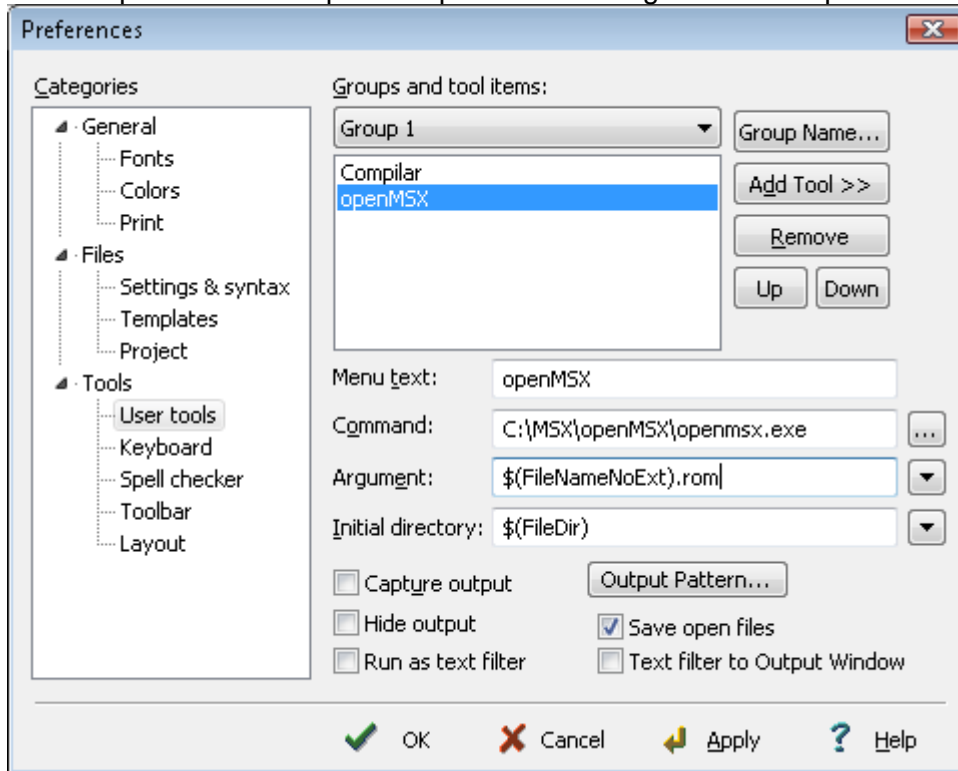


Aquí le decimos la ruta de nuestro ensamblador `c:\msx\asmsx012e\dist12e\asMSX.exe`  
 En la opción **Argument**: pulsa la flecha y selecciona **File Name**  
 Y en la opción **Initial directory**: pulsa la flecha y selecciona **File Directory**



Este es el resultado final que tienes que ver para el botón **Compilar**. Pulsa el botón **Apply**

Ahora repetimos todo el proceso para crear el segundo botón que lanza la ROM en el OpenMSX



Pulsa de nuevo en el botón **Add Tool** y en la ventana emergente elige la opción **Program**

Ahora en **Menu text:** teclea **openMSX**

Pulsa en **Command:** en los 3 puntos **...** y selecciona el directorio donde esta instalado el OpenMSX

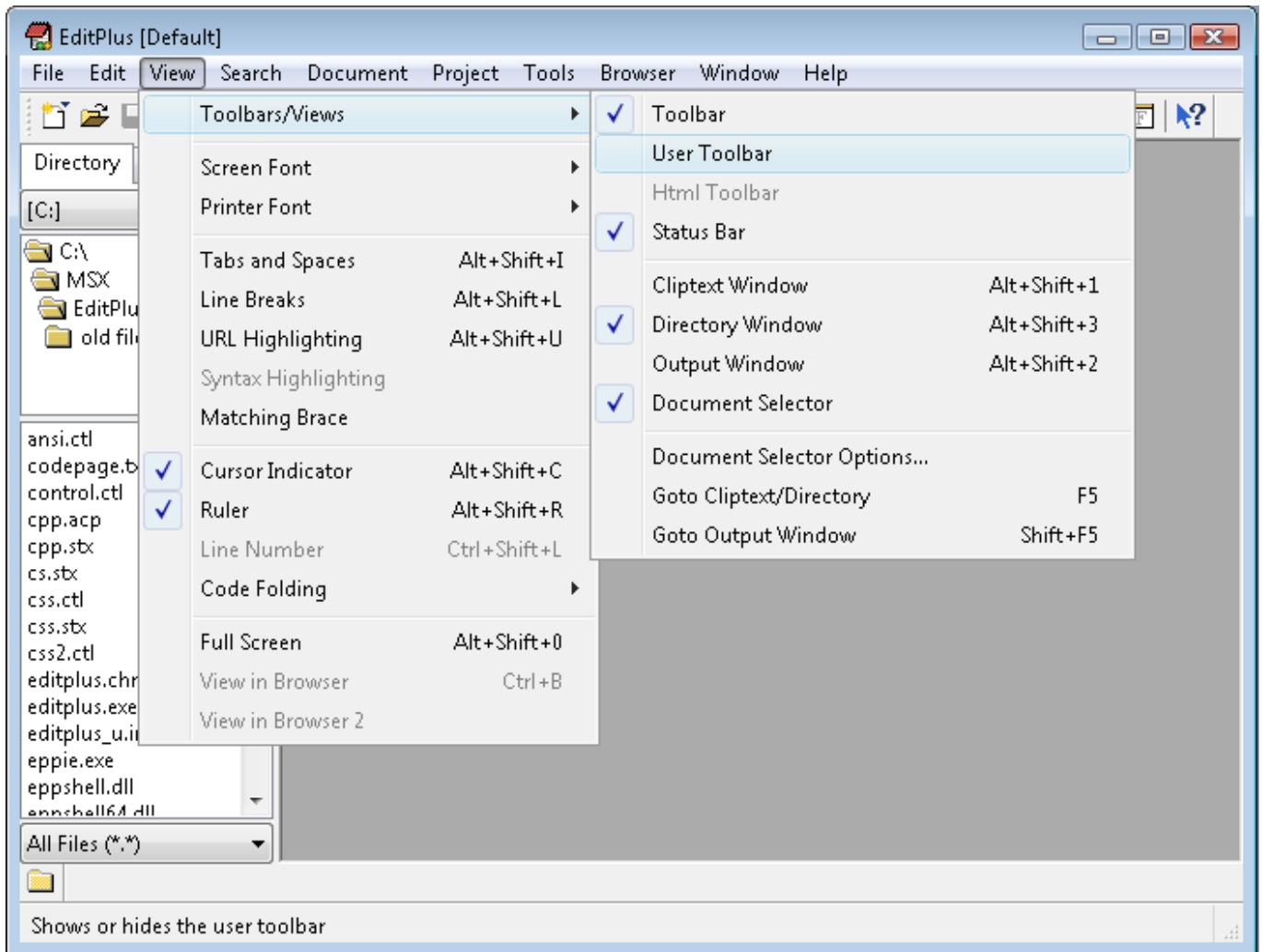
Que supongo que sera **C:\MSX\openMSX\openmsx.exe**

En la opción **Argument:** pulsa la flecha y selecciona **File Name Without Extension** y añade **.rom** al final

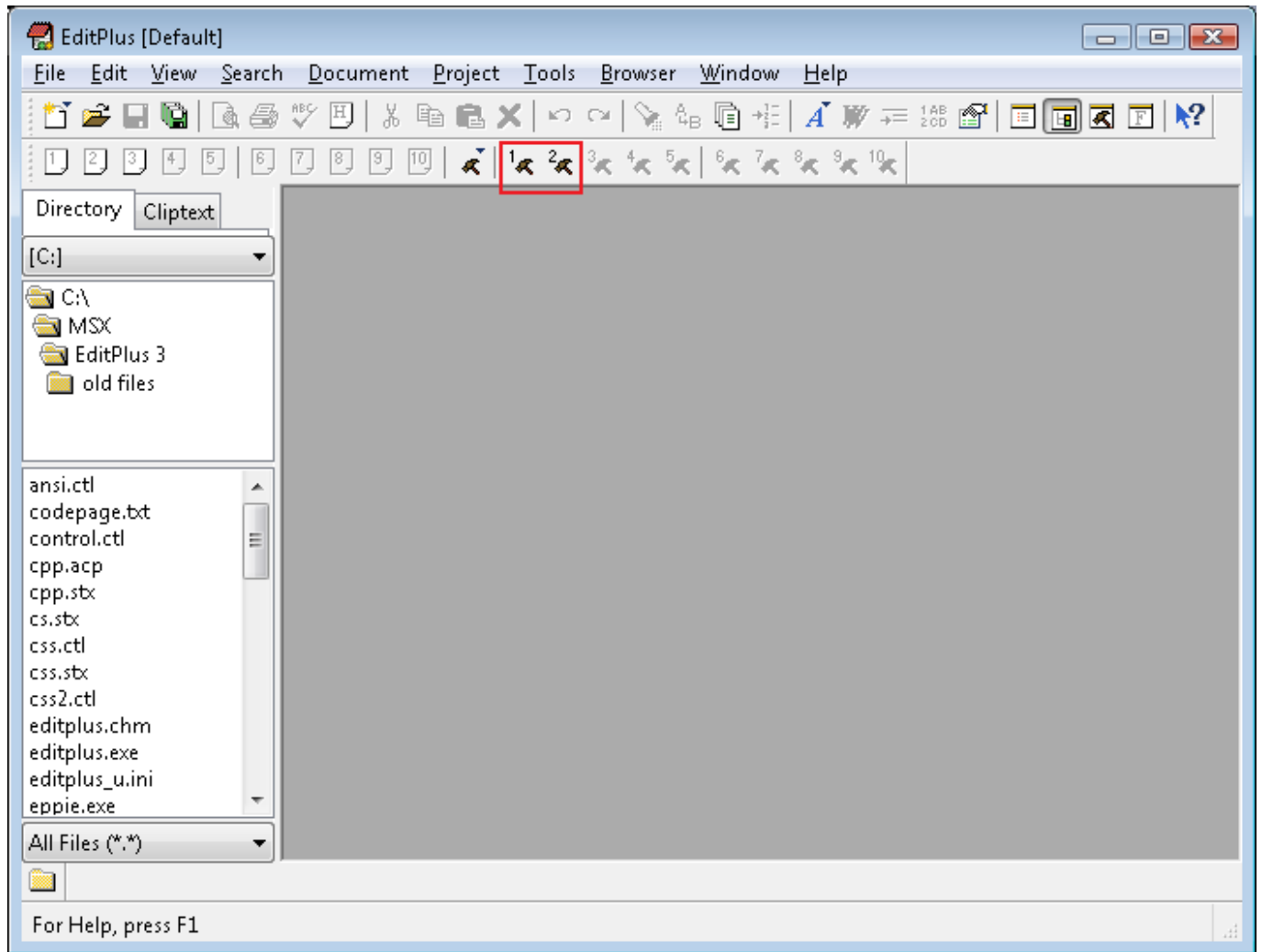
Y en la opción **Initial directory:** pulsa la flecha y selecciona **File Directory**

Fijate en la imagen de arriba de este texto, asi es como tiene que quedar la ventana.

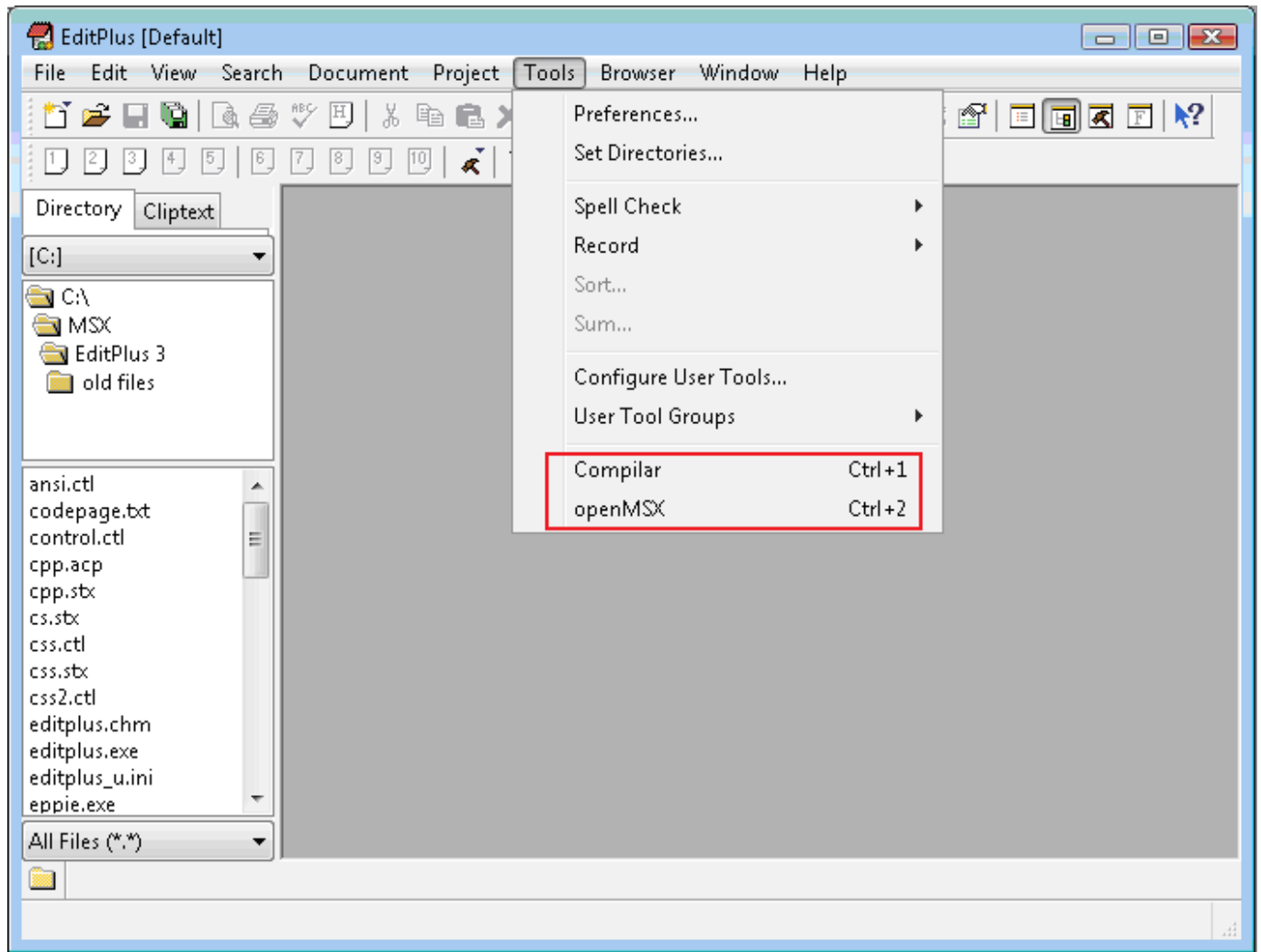
Ahora ya puedes pulsar el botón **Apply** y el botón **OK**



Ahora vamos a mostrar en la barra de iconos nuestros dos nuevos botones el [Compilar](#) y el [OpenMSX](#). Pulsa en menú [view – Toolbars/Views – User Toolbar](#) para que se muestre la barra User Toolbar.

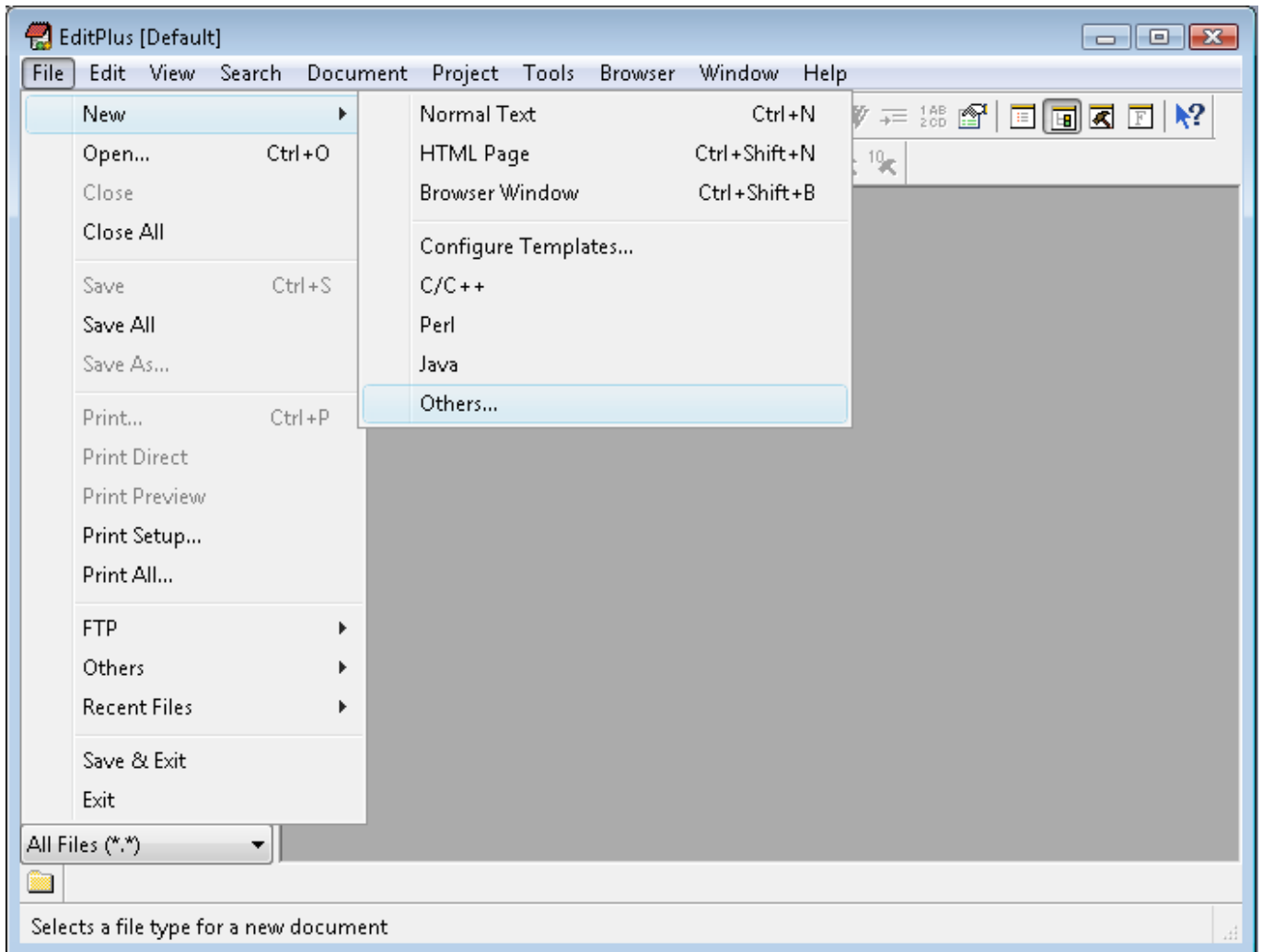


Y voila... aquí puedes ver nuestros botones [Compilar](#) y [OpenMSX](#) en el **recuadro rojo** martillo 1 y 2

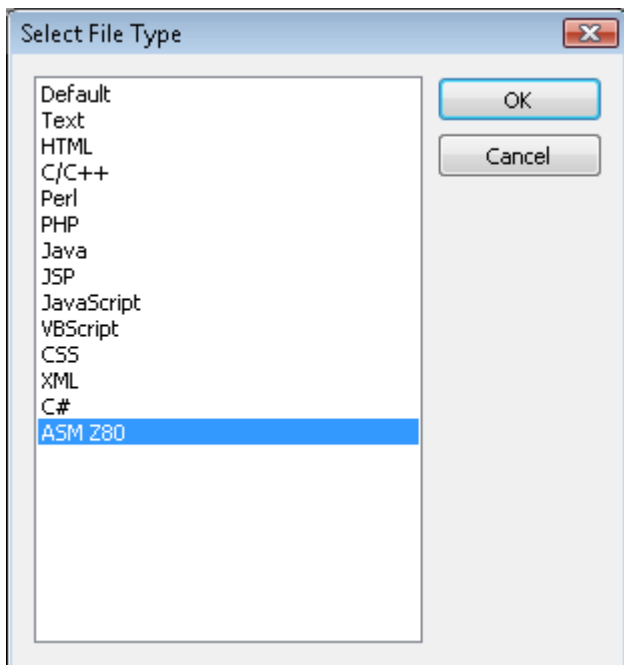


También puedes llamar a nuestros botones en [menú – Tools](#) o con [Control+1](#) y [Control+2](#)

Vamos a probar si funciona todo nuestro tinglado a la perfección, os voy a enseñar a crear vuestra primera ROM de MSX con un Hola Mundo muy sencillito usando la BIOS.



Vamos a crear nuestra primera ROM de MSX pulsa en menú [File – New – Others](#)



Selecciona [ASM Z80](#) y pulsa el botón [OK](#).

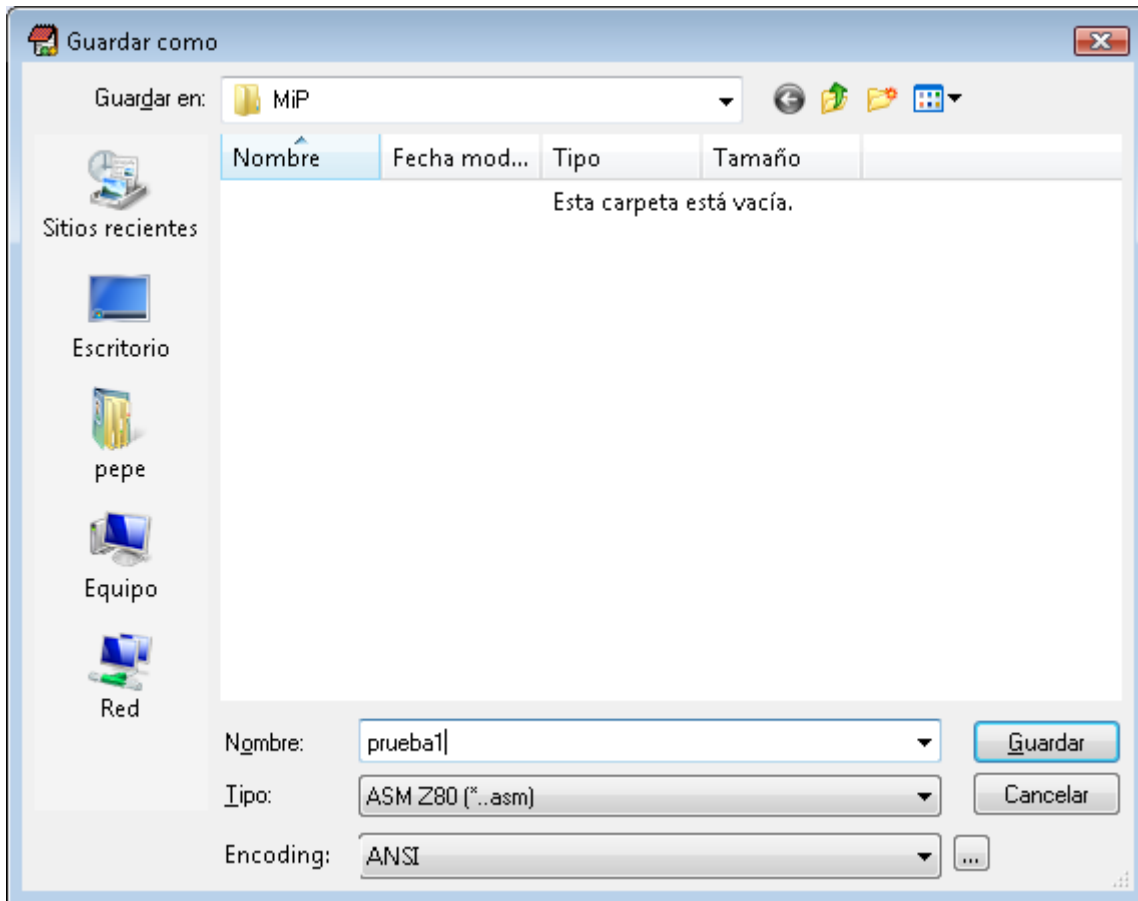


AHORA COPIA Y PEGA ESTE TEXTO EN TU EDITOR menos esta línea

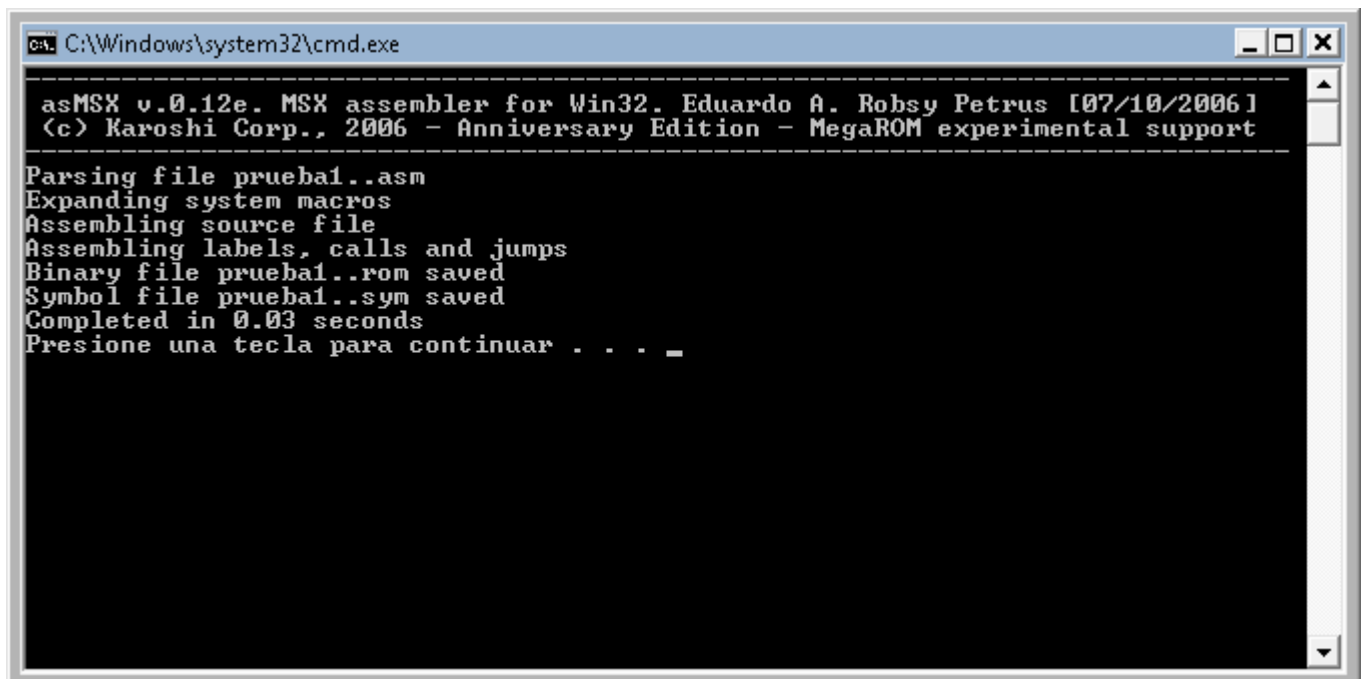
```
;-----  
; Nombre de nuestro programa  
; Hola Mundo  
;-----  
; DEFINIR CONTANTES  
;-----  
; no definimos ninguna constante  
  
; Variables de sistema MSX  
FORCLR equ 0f3e9h ; Foreground colour  
;-----  
; DIRECTIVAS PARA EL ENSAMBLADOR ( asMSX )  
;-----  
.bios ; Definir Nombres de las llamadas a la BIOS  
.page 2 ; Definir la dirección del código irá en 8000h  
.rom ; esto es para indicar que crearemos una ROM  
.start INICIO ; Inicio del Código de nuestro Programa  
  
;-----  
INICIO:  
; INICIO DEL PROGRAMA  
;-----  
call INIT_MODE_SC0 ; iniciar el modo de pantalla  
call IMPRI_MENSAJE ; imprimir el mensaje en pantalla  
FIN:  
jp FIN ; esto es como 100 goto 100  
  
;-----  
INIT_MODE_SC0:  
; INICIALIZA EL MODO DE PANTALLA  
;-----  
; BASIC: COLOR 15,0,0  
; Establecer el fondo de color Negro  
ld hl,FORCLR  
ld [hl],15 ; Color del primer plano  
inc hl ; blanco  
ld [hl],1 ; Color de fondo  
inc hl ; negro  
ld [hl],1 ; Color del borde  
 ; negro  
  
call INITXT ; set SCREEN 0  
; call INIT32 ; set SCREEN 1  
; call INIGRP ; set SCREEN 2  
; SCREEN 0 : texto de 40 x 24 con 2 colores  
; SCREEN 1 : texto de 32 x 24 con 16 colores  
; SCREEN 2 : gráficos de 256 x 192 con 16 colores  
; SCREEN 3 : gráficos de 64 x 48 con 16 colores  
;  
ret  
  
;-----  
;-----  
IMPRI_MENSAJE:  
; RUTINA QUE IMPRIME EL TEXTO EN PANTALLA  
;-----  
;  
ld h,01 ; situamos la Columna  
ld l,01 ; y la Fila donde escribir en la pantalla  
; ld hl,0101h ; también podemos hacerlo de esta manera  
call POSIT ; BIOS fijar el cursor donde empezara a escribir  
ld hl,texto ; ponemos HL apuntando al texto del mensaje  
@@bucle:  
ld a,[hl] ; leemos el primer carácter y lo metemos en A  
or a ; comprobamos si hemos llegado al final del texto  
ret z ; salimos de la rutina en el caso de que sea un cero  
call CHPUT ; escribimos ese carácter en la posición del cursor  
inc hl ; incrementamos HL para que apunte a la siguiente letra  
jr @@bucle ; como no hemos llegado al final continuamos escribiendo  
;-----  
texto:  
.db "Hola Mundo",0
```

FINAL DEL TEXTO QUE TIENES QUE COPIAR menos esta línea

Ya estamos preparados para compilar y ejecutar el compilado en el OpenMSX  
Así que pulsa el botón que hemos creado Compilar o botón definido 1



La primera vez que compilas tienes que guardar antes el fichero **.ASM** creo una carpeta llamada MiP y le pongo como Nombre: **HolaMundo.asm** y pulso el botón **Guardar**.



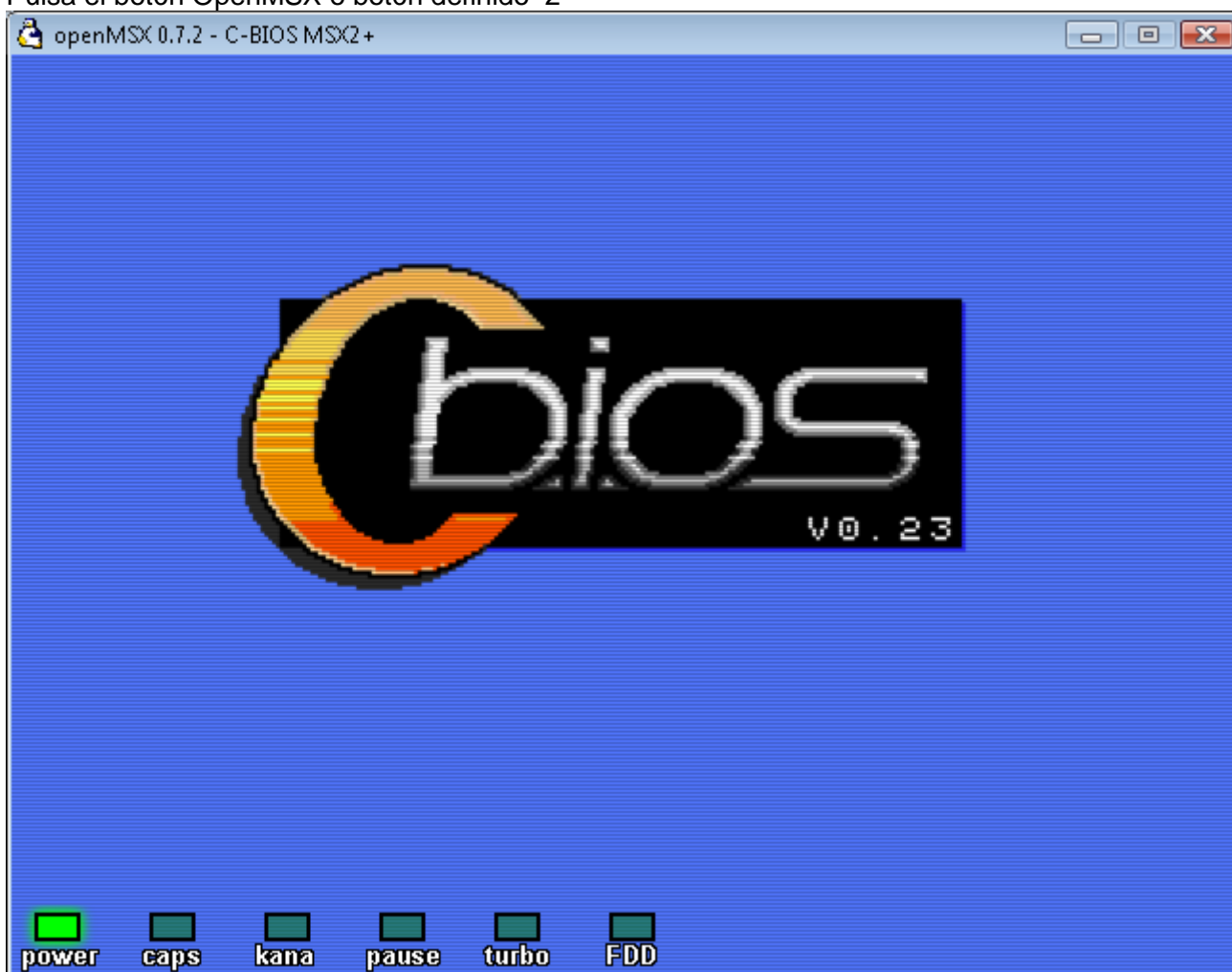
Aquí puedes ver que el **asMSX** ha compilado nuestro código en ensamblador sin ningún problema. Es importante que te fijes en esto antes de lanzar el OpenMSX. Cierra la ventana si todo está OK.

```
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 . . . _
```

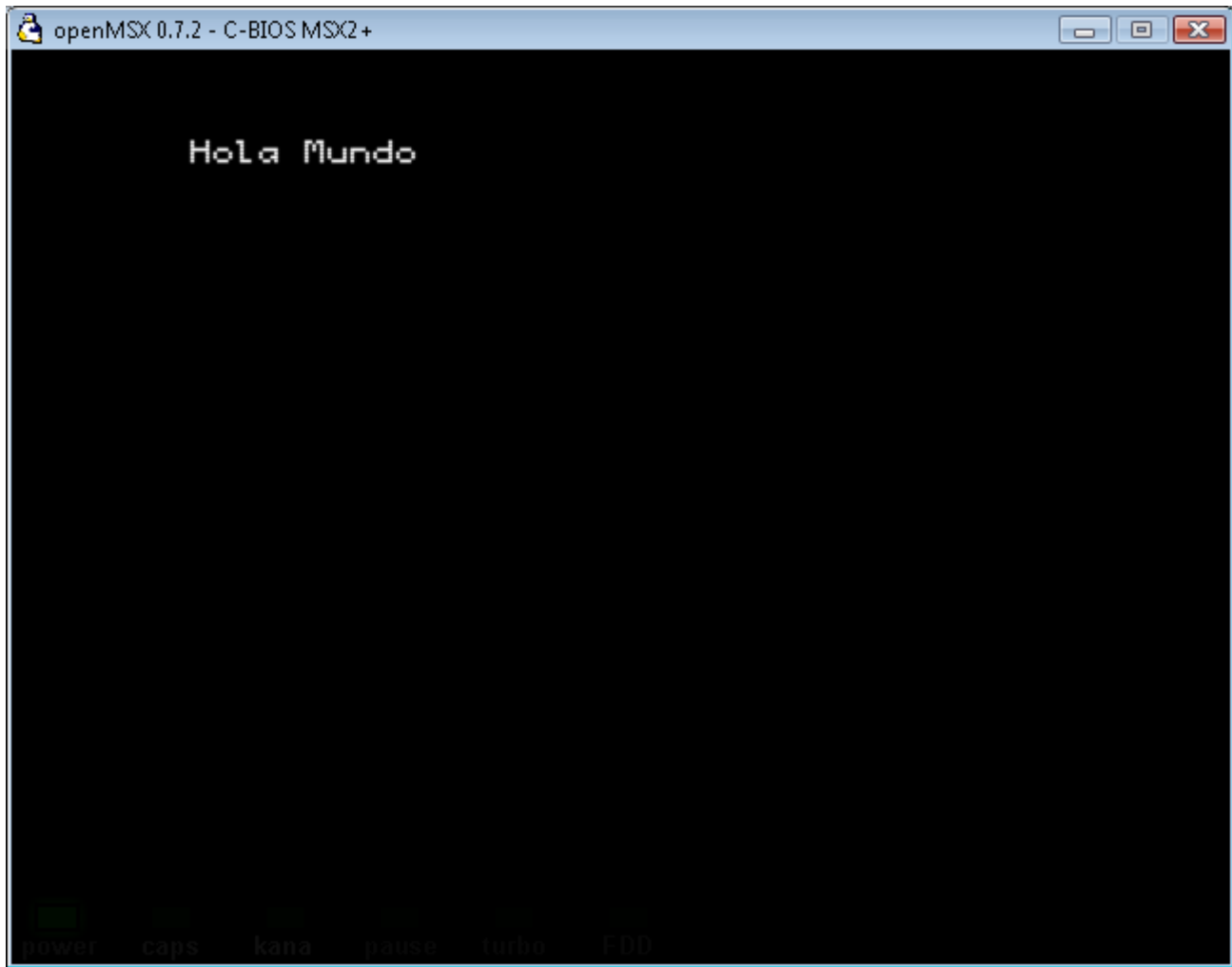
Aquí he forzado un error de sintaxis para que veas que en caso de error, el asMSX te dice en que línea o líneas de tu código has cometido un error de sintaxis, o algo que no has definido bien, fíjate en el número de línea y repasa con el editor de texto la línea 29 a ver que error habríamos cometido.

**AHORA QUE TENEMOS EL CODIGO BIEN VAMOS A PROBARLO**

Pulsa el botón OpenMSX o botón definido 2



Aquí puedes ver como arranca el OpenMSX cargando su BIOS como si fuera un MSX real



Y voila aquí tenemos el resultado final.

Puedes cerrar el emulador OpenMSX y cerrar la otra ventana negra que queda al cerrarlo.

Ya hemos creado nuestra primera ROM para MSX increíble y fácil. Ahora modifica la Fila y la Columna donde tiene que salir el texto en pantalla, para que veas lo fácil que es modificar compilar y lanzar.

Espero que sea de vuestro total agrado y nos vemos en el próximo tutorial. Donde explicare un poco el código de este artículo, donde se miran las rutinas de la BIOS del MSX y como se puede depurar nuestro código con el Debugger del BlueMSX.

P.D. el resto de programas de Pack del MSX es para futuros capítulos.

**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 )**