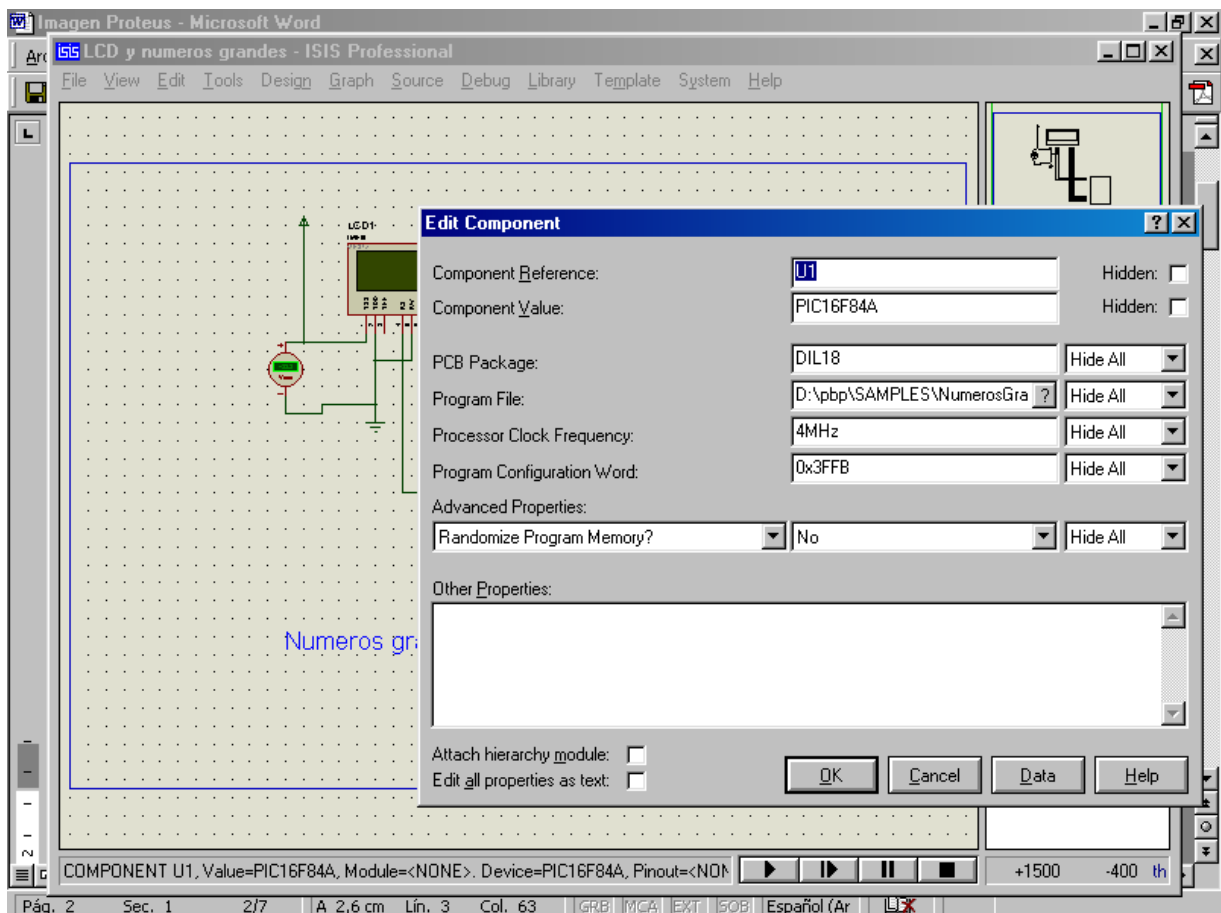


Proyecto: Display LCD para números grandes

Este proyecto nace a partir de una necesidad mía de presentar la información de un equipo en forma legible a la distancia pero utilizando displays del tipo LCD, ya que la cantidad de líneas en un sistema de displays multiplexados de 7 segmentos de leds imposibilitaba continuar con el proyecto.

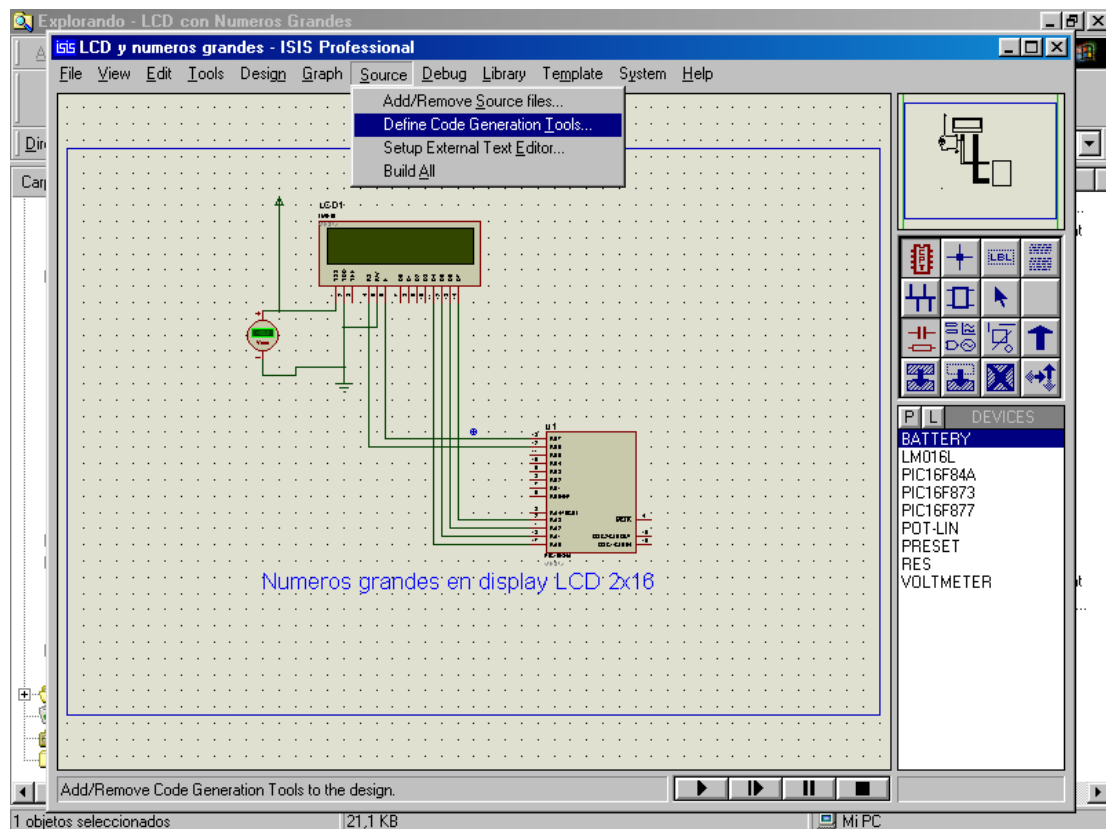
Buscando en la WEB encontré un viejo documento de Scott Edwards, en el cual mostraba como en un display LCD de 4 líneas y 20 caracteres podía mostrar números gigantes, asimilando a un display de 7 segmentos de 1" de altura.

Pensé que no me servía, hasta que se me ocurrió jugar con el generador de caracteres del display LCD, consiguiendo generar los segmentos que permiten mostrar los números en dos líneas de un display común de 2 líneas por 16 caracteres con el controlador 44780, casi universal en estos displays.



Por supuesto, tiene ciertas deficiencias, que se evidencian mas cuando el display tiene demasiada separación entre las matrices de puntos que lo conforman, pero seguramente algunos de ustedes podrán sacar buen provecho de este proyecto, tal como yo lo haré.

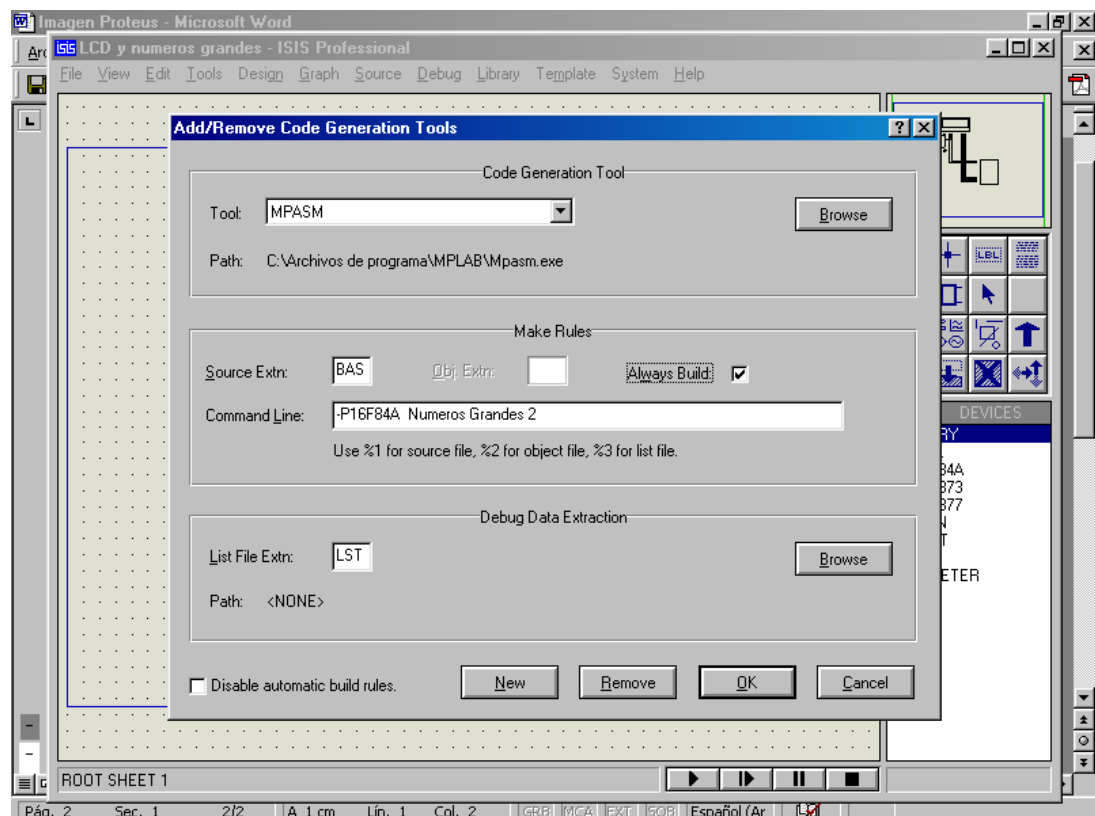
A continuación podrán ver el esquemático directamente en el simulador, ya que a la vez explicare como configurar este ultimo para simular en PBP.



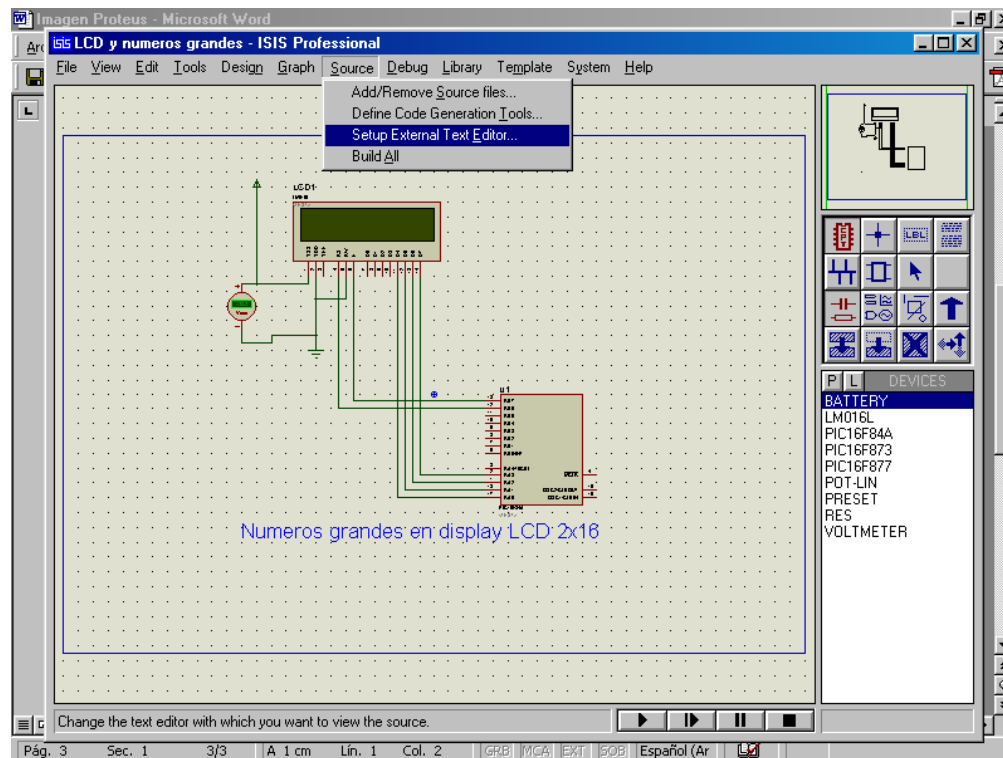
Deberemos explicarle al simulador donde están las herramientas que le permitirán a Proteus ejecutar la simulación.

Definiremos esto por pasos:

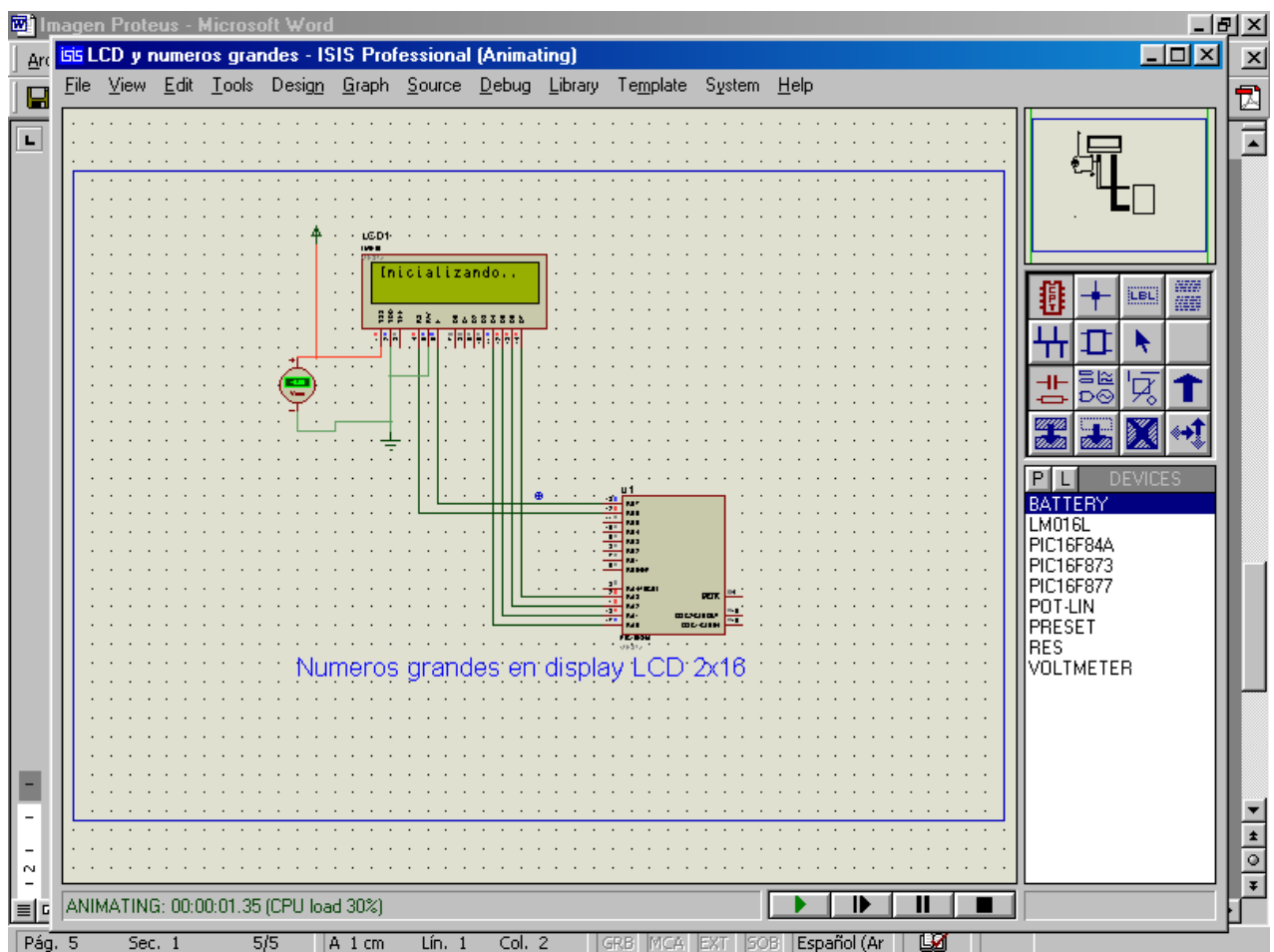
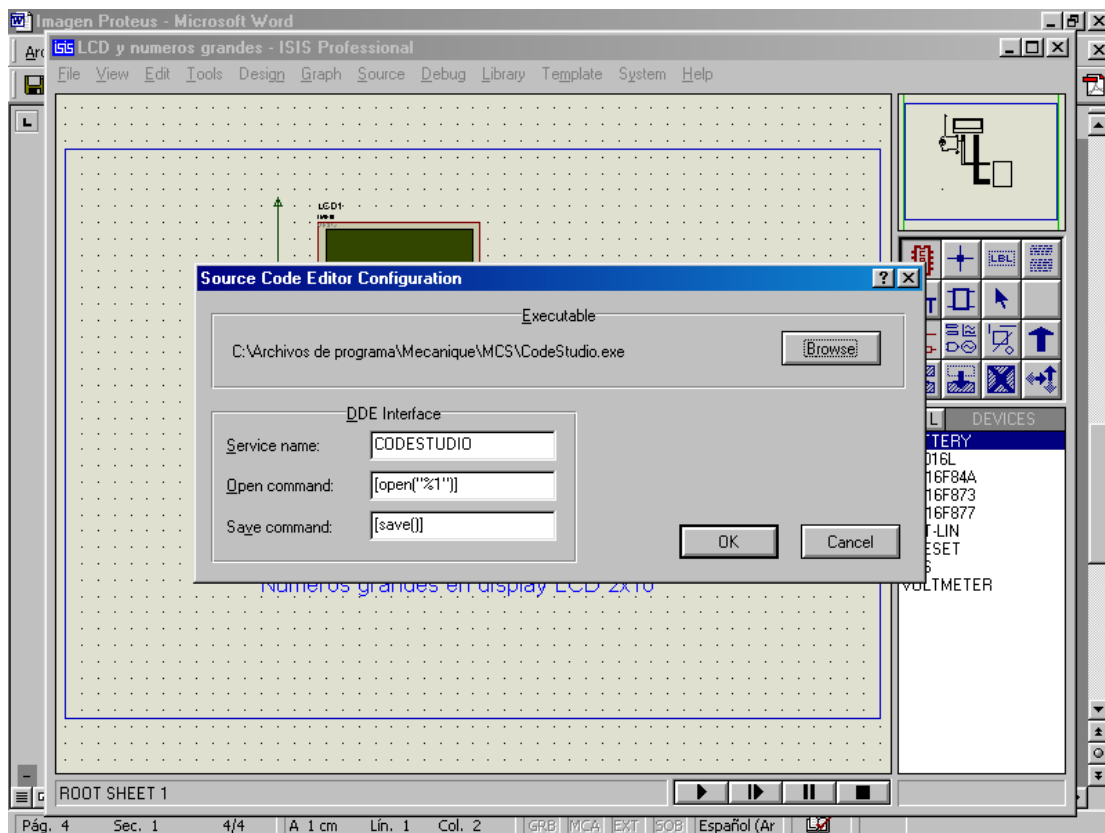
- Como ven en la primer imagen, deberemos decirle al programa donde esta el código para el micro (buscar la ruta directamente en la ventana de búsqueda,

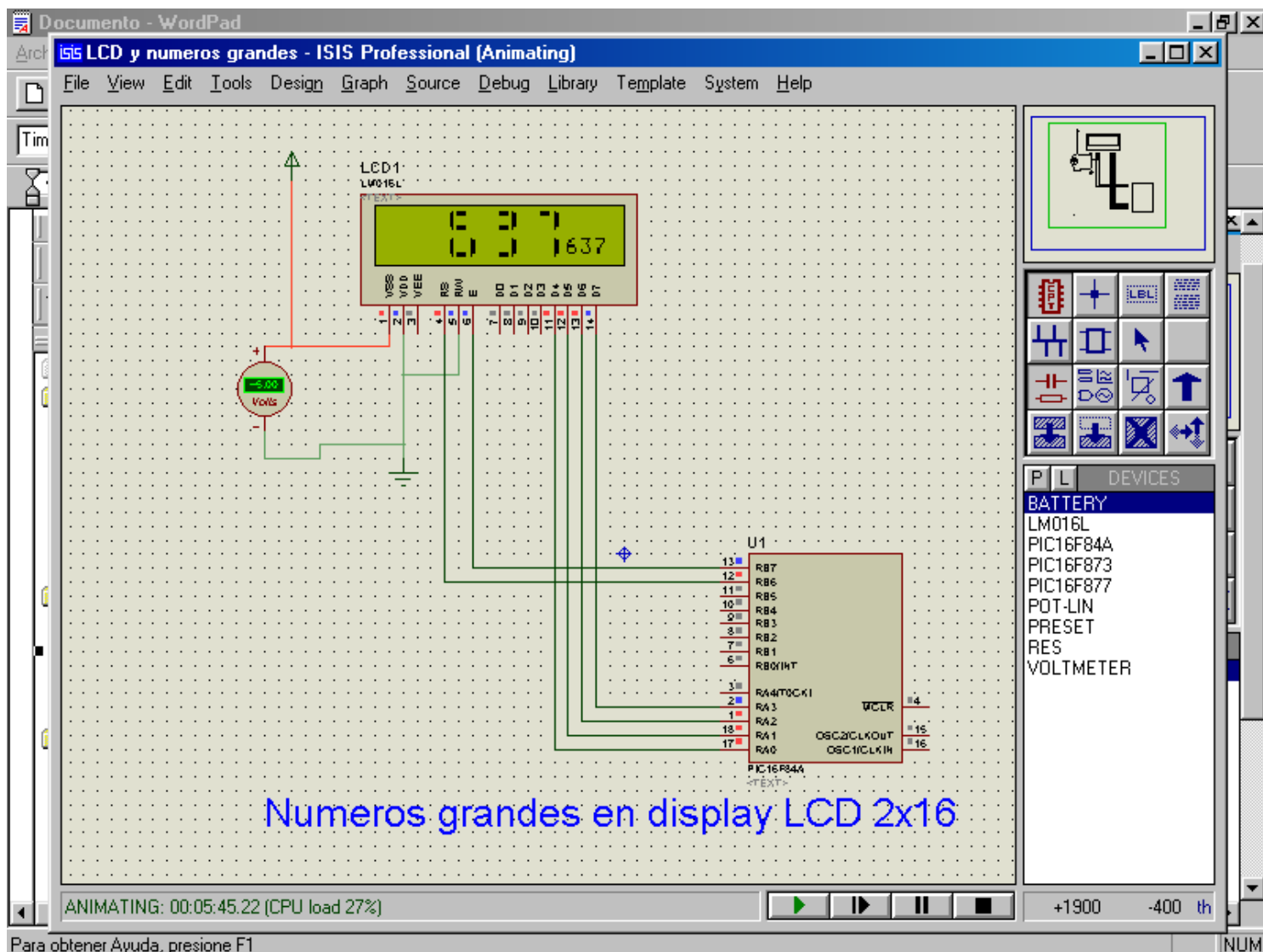


tener cuidado con los nombres largos y el raíz del PBP), la velocidad del reloj, y si se va a empezar con valores al azar en las variables del programa (randomize).



- Luego se le indicara donde están los programas que permiten ensamblar, simular, etc. Esto permite indicarle desde donde obtendrá el archivo .hex que precisa para realizar la emulación. Esto se hace ejecutando la opción de menú tal como se ve en la figura 2 y programándole tal como se ve en la tercera imagen.
- Tener en cuenta de indicar el nombre de la aplicación sin la extensión bas, ya que esta es añadida por el compilador.
- Cuando tengamos esto listo podremos también indicarle donde y con que programa ejecutaremos la simulación. En las imágenes 4 y 5 se puede ver como se selecciona el menú y el Microcode para ejecutar esa simulación.
- En este momento ya tenemos listo todo para realizar la simulación, presionando el botón de inicio de la simulación veremos como se ve en las imágenes 6 y 7 como se simula el código en el Proteus.





Numeros grandes en display LCD 2x16

Espero puedan sacar utilidad del proyecto.

MGLSOFT

```

*****
' * Name      : Numeros Grandes2.BAS *
' * Author    : Marcos Lazcano - MGLSOFT *
' * Notice    : Copyright (c) 2003 MGLSOFT *
' *           : All Rights Reserved *
' * Date      : 27/03/03 *
' * Version   : 2.0 *
' * Notes     : Muestra en un display de 2x16 numeros grandes en *
' *           : dos lineas, conformando un numero posible de ver *
' *           : a mas distancia por el usuario. *
' *           : Para utilizar este formato se definen un juego de *
' *           : simbolos en formato de 5x8 puntos y se guardan en *
' *           : el area de memoria CGRam del display, luego las *
' *           : rutinas van mostrando cada caracter segun como se *
' *           : conforme el numero a mostrar. Esto para cada linea. *
' *           : Para utilizarlo se debera incluir el codigo de *
' *           : usuario en reemplazo de la rutina llamada DEMO. *
' *           : Cuando quiera mostrar una variable en numeros *
' *           : grandes se debera escribir este en ValDisp y *
' *           : luego llamar a la rutina GranNum. *
*****

' Vaya mi agradecimiento a los usuarios de la lista TODOPIC
' y a Scott Edwards por inspirarme con su codigo para displays
' de 4 lineas por 20 caracteres.
' De la misma forma ahora comparto este codigo, quien lo use debe
' sentirse liberado de cualquier demanda por derechos, asi como
' implicitamente al utilizarlo me liberara de cualquier demanda
' por daños que este codigo o su mala utilizacion pudieran producirle.

' Define registros del LCD y bits
DEFINE LCD_DREG      PORTA
DEFINE LCD_DBIT      0
' Primer bit de datos en PortA.0, en modo a 4 bits
DEFINE LCD_RSREG     PORTB
DEFINE LCD_RSBIT     6
' Señal del LCD RS en PortB.6
DEFINE LCD_EREG      PORTB
DEFINE LCD_EBIT      7
' Señal del LCD Enable en PortB.7
' Definicion de constantes
I CON $FE          ' Prefijo para comando LCD.
ClrLCD CON 1       ' Instruccion para borrar el display LCD.
cgRAM CON 64       ' Direccion 0 de CG RAM del LCD.
columna CON 1      ' Columna inicial donde arranca el mensaje.
' Grupo de variables tipo Word
Puntee VAR WORD    ' Puntero para la EEPROM.
UsoPat VAR WORD    ' Patron de numeros a enviar al LCD.
ValDisp VAR WORD   ' Valor a ser mostrado como numeros grandes.
decada VAR WORD    ' Variable para encontrar ceros nulos y no mostrarlos.
' Grupo de variables tipo Byte
temp VAR BYTE      ' Variable temporal.
nibble VAR BYTE    ' Indice para encontrar patrones de numeros en tablas.
digito VAR BYTE    ' Digito actual para el display.
linea VAR BYTE     ' Linea actual del LCD.
nib0 VAR BYTE      ' Primer nibble obtenido.
nib1 VAR BYTE      ' Segundo " "
nib2 VAR BYTE      ' Tercer " "
nib3 VAR BYTE      ' Cuarto " "

' Guardado de los patrones de bits en EEPROM en el momento de la

```

```

' grabacion del micro.
PatBits0 DATA 16,24,24,24,24,24,24,16 ' Media barra lateral derecha.
PatBits1 DATA 1,3,3,3,3,3,3,1 ' Media barra lateral izquierda.
PatBits2 DATA 31,31,0,0,0,0,0,0 ' Barra superior sola.
PatBits3 DATA 0,0,0,0,0,0,31,31 ' Barra inferior sola
PatBits4 DATA 31,31,0,0,0,0,31,31 ' Barra superior e inferior.
PatBits5 DATA 2,6,14,30,14,6,2,0 ' Flecha Izquierda.
PatBits6 DATA 8,12,14,15,14,12,8,0 ' Flecha Derecha.
PatBits7 DATA 0,0,0,0,0,0,0,0 ' Todo vacio.

CLEAR
'ValDisp = 1234 'Usado para pruebas...

PAUSE 1000 ' espera la normalizacion del LCD al arranque.
LCDOUT I,ClrLCD ' Borrar el LCD.
LCDOUT "Iniciando.."
PAUSE 500

' -----
' Grabar los patrones de bits en la CG Ram para luego ser utilizados
' -----
LCDOUT I,cgRAM ' Entrar en la CG RAM.
FOR Puntee = 0 TO 63 ' guardar los patrones de bits en el LCD
    READ Puntee,temp
    LCDOUT temp
NEXT Puntee
' una vez guardados los datos en la CGRam hay que borrar el LCD
LCDOUT I,ClrLCD ' Borrar el LCD.
PAUSE 1

' -----
' Demo: muestra el numero puesto en ValDisp en Numeros Grandes
' -----
Demo:
GOSUB MiGranNum ' con esta linea muestro el numero en grande
LCDOUT DEC ValDisp ' esta otra lo hace en chico o normal
ValDisp = ValDisp + 1
PAUSE 500 ' pausa para mostrar numeros
GOTO Demo

' -----
' Rutina que muestra los numeros grandes
' -----
MiGranNum:
FOR linea = 0 TO 1 ' solo dos lineas del LCD
    decada = 1000
    LOOKUP linea,[128,192],temp
    temp = temp + columna ' referencia del inicio del mensaje
    LCDOUT I,temp
    FOR digito = 3 TO 0 STEP -1 ' hasta 4 digitos solamente
        nibble = ValDisp DIG digito
        GOSUB BuscaPatron
        IF ValDisp = 0 AND digito = 0 THEN saltar10
        'comentar la siguiente linea si se necesitan los ceros a la izquierda
        IF ValDisp < decada THEN blankIt1
        saltar10:
        GOSUB BuscaNibble
        LCDOUT nib2,nib1,nib0
        ' usar esta linea si se quieren visualizar los caracteres enviados al
        ' display utilizando el mismo display. Comentar la anterior...
        'lcdout dec nib2,dec nib1,dec nib0

```

```

        GOTO continua1
blankIt1:
        LCDOUT " " " 'elimina ceros a la izquierda
        continua1:
        IF digito = 0 THEN saltar11
        'lcdout 32 ' agregarla si se quiere espaciar entre numeros
saltar11:
        decada = decada/10
        NEXT digito
NEXT linea
'lcdout $FE,$D0, dec ValDisp dig 0
RETURN

' -----
' Aca se definen los patrones utilizados para enviar numeros grandes al LCD
' -----
BuscaPatron:
        BRANCH linea,[uno,dos]
                ' 0 1 2 3 4 5 6 7 8 9
                ' -----
uno:
        LOOKUP2 nibble,[120,770,740,740,130,147,147,720,140,140],UsoPat
        ' comentar linea superior para utilizar UsoPat DIG x , de lo contrario
        ' comentar linea inferior si se utiliza mascara en la rutina BuscaNibble
        'lookup2 nibble,[120,$770,$740,$740,$130,$147,$147,$720,$140,$140],UsoPat
        RETURN
dos:
        LOOKUP2 nibble,[130,770,137,730,770,730,130,770,130,770],UsoPat
        ' comentar linea superior para utilizar UsoPat DIG x , de lo contrario
        ' comentar linea inferior si se utiliza mascara en la rutina BuscaNibble
        'lookup2 nibble,[130,$770,$137,$730,$770,$730,$130,$770,$130,$770],UsoPat
        RETURN

' -----
' Aca se definen los nibles utilizados para enviar numeros grandes al LCD
' Observar las dos formas distintas que se pueden utilizar.
' Una trabaja con numeros en notacion Decimal y el otro lo hace en Hexa
' Probar ambas formas y se veran las diferencias de tamaño de codigo...
' -----
BuscaNibble:
        'nib0 = UsoPat & $000F
        'UsoPat = UsoPat >> 4
        'nib1 = UsoPat & $000F
        'UsoPat = UsoPat >> 4
        'nib2 = UsoPat & $000F
        'UsoPat = UsoPat >> 4
        'nib3 = UsoPat & $000F
        ' comentar lineas superiores para utilizar UsoPat DIG x , de lo contrario
        ' comentar lineas inferiores si se utiliza la rutina BuscaNibble
        nib0 = UsoPat DIG 0
        nib1 = UsoPat DIG 1
        nib2 = UsoPat DIG 2
        nib3 = UsoPat DIG 3
        UsoPat = 0
        RETURN

```