

Características y modo de funcionamiento del programa IC-PROG

(Traducido de la ayuda en línea en Italiano)



Realizado por:
Fernando Remiro Domínguez
Profesor de Sistemas Electrónicos
I.E.S. Juan de la Cierva

INDICE

0	CARACTERÍSTICAS DE IC-PROG	4
1	INSTALACIÓN	4
1.1	INSTALACIÓN HARDWARE DEL PROGRAMADOR.....	4
1.2	INSERTAR UN DISPOSITIVO O UN <i>SMARTCARD</i>	4
2	GENERAL	4
2.1	PANTALLA PRINCIPAL	4
2.2	ÁREA DE CÓDIGO Y DATOS	5
2.3	ÁREA DE CONFIGURACIÓN	6
2.4	OPERACIONES ESTÁNDAR:	8
3	CONFIGURACIÓN DE IC-PROG	8
3.1	INTERFAZ.....	8
3.2	EMPLEO DE LAS API BAJO WINDOWS NT/2000.....	8
3.3	COMUNICACIÓN.....	8
3.4	PRIORIDAD DEL PROCESO	9
3.5	CAMBIO DEL TIEMPO DE RETARDO DE I/O.....	9
4	PROCEDIMIENTO	9
4.1	ARCHIVOS.....	9
4.1.1	<i>Abrir Archivo</i>	9
4.1.2	<i>Guardar como</i>	10
4.1.3	<i>Abrir Datos</i>	10
4.1.4	<i>Archivos Recientes</i>	10
4.1.5	<i>Imprimir</i>	10
4.2	EDICIÓN.....	10
4.2.1	<i>Llenar Buffer</i>	10
4.2.2	<i>Llenar Intervalo de Buffer</i>	10
4.2.3	<i>Copia del buffer</i>	10
4.2.4	<i>Pega en el buffer de memoria</i>	10
4.2.5	<i>Llenar Determinado Buffer</i>	10
4.2.6	<i>Llenar intervalo Buffer de</i>	10
4.2.7	<i>Copia del Buffer de datos</i>	10
4.2.8	<i>Pega al Buffer de datos</i>	11
4.3	BUFFER.....	11
4.3.1	<i>Buffer activo</i>	11
4.3.2	<i>Compara Buffer</i>	11
4.4	AJUSTES	11
4.4.1	<i>Dispositivos</i>	11
4.4.2	<i>Dispositivos Recientes</i>	11
4.4.3	<i>Hardware</i>	11
4.4.4	<i>Control Hardware</i>	12
4.5	OPCIONES.....	12
4.5.1	<i>Confirmación</i>	12
4.5.2	<i>Avisos</i>	13

4.5.3	I2C.....	13
4.5.4	Programación.....	13
	Verificación de dispositivos:	13
4.5.5	Atajos.....	14
4.5.6	Arrastrar y pegar (Drag & Drop).....	14
4.5.7	Smartcard	14
4.6	LENGUA UTILIZADA	15
4.7	SHELL.....	15
4.8	MISCELÁNEA	15
4.9	SMARTCARD (PHOENIX).....	16
4.10	BORRAR AJUSTES.....	16
4.11	COMANDOS	16
4.11.1	Leer Todo.....	16
4.11.2	Programar Todo.....	16
4.11.3	Programar bit de configuración.....	16
4.11.4	Borra Todo	16
4.11.5	Verifica Blanco.....	17
4.11.6	Verificar.....	17
4.11.7	Asistente SmartCard.....	17
4.12	HERRAMIENTAS	17
4.12.1	Cargar Ficheros CDR	17
4.13	VER.....	17
4.13.1	Ensamblador.....	17
4.13.2	Hexadecimal.....	17
4.13.3	Posición del dispositivo.....	17
4.13.4	Informaciones sobre el microcontrolador.....	18
4.13.5	Refrescar.....	18
4.14	AYUDA.....	18
4.14.1	Acerca de.....	18
5	REFERENCIAS	18
5.1	PROGRAMADORES UTILIZADOS.....	18
5.2	INFORMACIONES ESPECÍFICAS PARA LOS PROGRAMADORES	18
5.2.1	Programador JDM (Ludipipo).....	18
5.2.2	Dispositivos que soporta:	18
5.2.3	Dispositivos verificado con éxito.....	20
5.3	OPCIONES DE LÍNEAS DE MANDO	21
5.3.1	IC-Prog Opción1 Opción2 etcétera	21
5.4	EDIT EL BUFFER	22
5.5	INFORMACIONES TÉCNICAS SOBRE LOS PIC.....	22
5.6	SMARTCARD (WAFERCARD).....	23
5.6.1	Selección del oscilador.....	23
5.6.2	WDT, (Watch Dog Temporizador - temporizador del perro guardián). 23	
5.6.3	PWRT (Power on Reset).....	23
5.6.4	RB6 contra RB7.....	24
5.6.5	Para la Programación de PIC	25
5.7	ADAPTADORES.....	25

5.7.1	<i>Adaptador MicroWire para JDM</i>	25
5.8	ADAPTADOR NVM 3060	26
6	INFORMACIONES ADICIONALES	27
6.1	PROGRAMAR LOS EEPROM CON EL PROGRAMADOR JDM	27
6.2	CONTROLAR SI EL PROGRAMADOR FUNCIONA... ..	27
7	AGRADECIMIENTOS	28

0 Características de IC-Prog

El IC-Prog es un programa que corre bajo Windows y que permite la programación de muchos dispositivos y que está probado por un buen número de programadores Hardware.

Los parámetros de la línea de mando pueden ser usados como interfaz con compiladores externos.

1 INSTALACIÓN

1.1 Instalación Hardware del programador

Lo primero es conectar el programador a una de los puertos disponibles sobre el PC en uso, serie o paralelo.

Cerciorarse que todos los interruptores eventualmente presentes en el programador, estén configurados según las recomendaciones hechas por el diseñador del hardware.

1.2 Insertar un dispositivo o un *SmartCard*

Insertar un microcontrolador PIC o un *SmartCard*, teniendo cuidado de que estos tipos de dispositivos son sensibles a las descargas electrostática (ESD) antes de quitar el embalaje de los mismos, cerciorarse de prevenir estas últimas, para evitar la destrucción electrónica de los dispositivos o los equipos.

Además debe de cerciorarse de que todos los pines del dispositivo, estén rectos y que entren perfectamente en el zócalo del programador, teniendo mucho cuidado, por que los pines se rompen con extrema facilidad.

Si se realizan operaciones frecuentes de reprogramación, se aconseja utilizar en el programador un zócalo entre el dispositivo y el zócalo del programador, de tal modo que sea más fácil introducir y sacar el dispositivo del zócalo.

Otra alternativa como solución, es reemplazar el zócalo presente en el programador por un zócalo tipo profesional de fuerza nula (ZFN)

2 General

2.1 Pantalla Principal

La pantalla principal de IC-Prog presenta la información necesaria para programar el dispositivo seleccionado. Todos los dispositivos tienen al menos un área de Código dónde puede ser almacenada información. Los dispositivos como las EEPROM sólo tienen este área de Código.

Otros Dispositivos, como por ejemplo los microcontroladores, tienen varias áreas. Normalmente el área de código contiene el código que será ejecutado por el microcontrolador y el área datos contiene datos fijos como tablas para cálculos, informaciones sobre el autor del *Fireware*, etc....

La mayor parte de los microcontroladores, por ejemplo los PIC también tienen un área de configuración.

Las informaciones sobre la configuración, se realiza en la fase de inicio con algunos parámetros que varían en funcionamiento del microcontrolador utilizado (ver por ejemplo los osciladores en la sección *SmartCard*, los fusibles, etc...). Las informaciones específicas de estas configuraciones pueden ser encontráis en el *data shell* del microcontrolador.

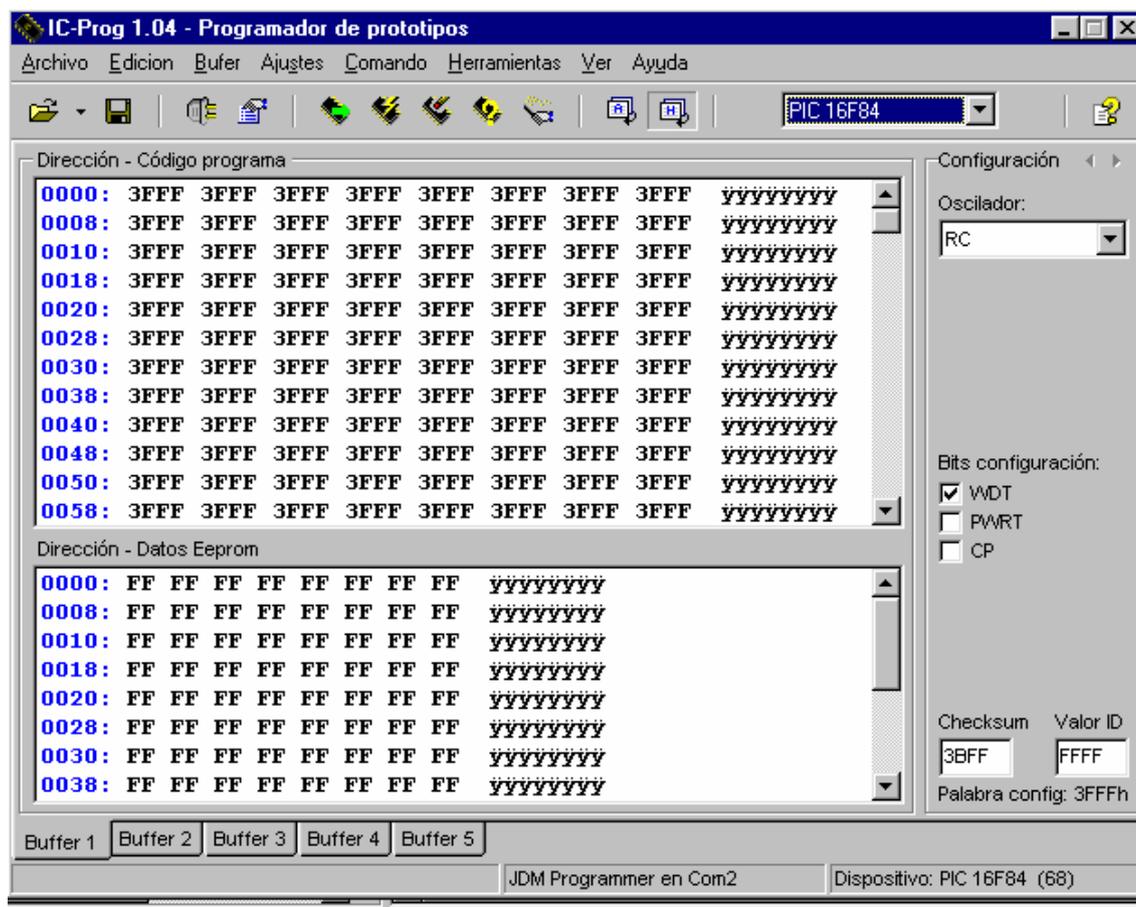


Figura 1.- Ventana principal

El IC-Prog visualiza todas las informaciones de estas áreas de memoria.

2.2 Área de Código y Datos

En el área de código y el área datos se presenta la información en valor hexadecimal y el correspondiente valor ASCII:

La columna de la izquierda del área de código y el área de datos contiene la dirección física de memoria del dispositivo sobre la que se guarda la información. En el centro del campo se presenta el valor hexadecimal y la columna derecha contiene la mismas informaciones, pero en valor ASCII (carácter).

Cada fila en el área de Código presenta dos palabras de 8 bits, o bien una palabra de 16 bits.

Cada dirección de cada fila, presenta las 4 cifras en color azul, que se van incrementado de 8 en 8 palabras.

Una palabra normalmente está compuesta por 16 bits, y el IC-Prog utiliza cifras hexadecimales de 0000 a FFFF para representar los 16 bits.

Obviamente no todos los dispositivos utilizan palabras a 16 bits, pero pueden utilizar 14, 12 y 8 bits por palabra. En estos casos el IC-Prog utiliza igualmente una notación hexadecimal de 4 cifras, indicando respectivamente en el área de código 3FFF, 0FFF, 00FF para 14, 12 u 8 bits respectivamente.

El carácter sólo representa la parte baja de la palabra visualizada, o bien IC-Prog utiliza los 8 bits menos significativos de la palabra, 16 bits, ya que según el estándar ASCII, no extendido, solo se utilizan 256 caracteres, de 0 a 255.

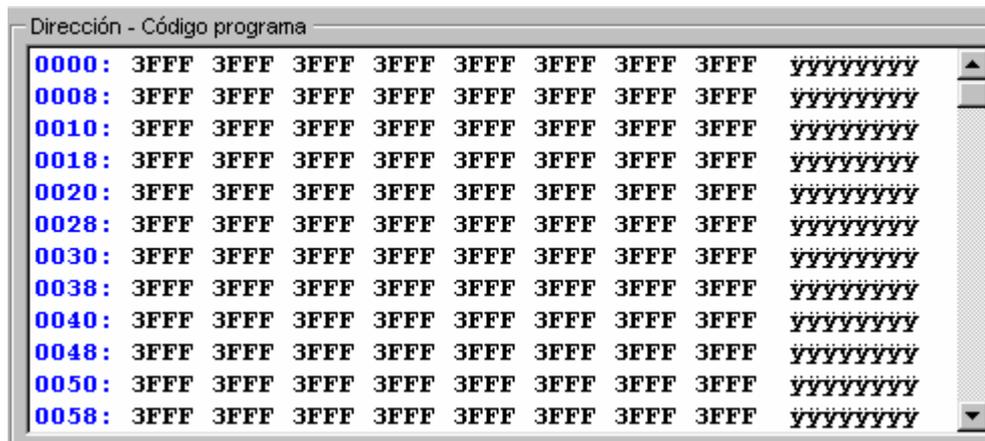


Figura 2.- Área Código

Cada fila de direcciones del área de datos, también está compuesta 8 palabras, pero cada palabra es de 8 bits y no de 16 bits, por tanto, IC-Prog utilizará una notación hexadecimal de 2 cifras que van de 00 a FF. En este área, IC-Prog utilizará los 8 bits de la palabra para visualizar el valor correspondiente ASCII

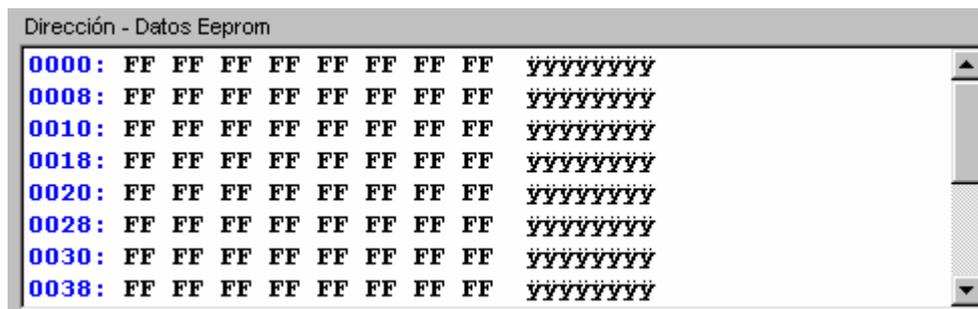


Figura 3.- Área Datos

2.3 Área de configuración

El área de configuración del dispositivo, si en el dispositivo seleccionado, aparece en el apartado de control () que permiten una fácil activación o desactivación de la configuración, sencillamente activando el campo del apartado deseado.

El usuario puede seleccionar fácilmente la configuración deseada, el IC-Prog calculará automáticamente la palabra de la configuración correspondiente.

El cálculo que IC-Prog efectúa, es reconducido al área de final de la configuración. Estas configuraciones varían en base al tipo de microcontrolador utilizado. Para mayor información en necesario, consultar los *data shell* del fabricante.

A menudo un elemento determinado de la configuración puede ser solamente habilitado o inhabilitado. Esto da lugar a la escritura de uno 0 o de un 1 en un bit específico en la palabra de la configuración, si se trata de la configuración del *Power on Reset*, *PWRT*, y la solución adoptada, es que si un elemento de la configuración

utiliza un valor invertido en el dispositivo, el IC-Prog invierte automáticamente el elemento de la configuración, si este último lo necesita. Para mayor información consultar la sección *SmartCard*.



Figura 4.- Área de Configuración

Algunos dispositivos, tienen muchos elementos de configuración, tantos que no es posible visualizar toda la información en el área de configuración. IC-Prog provee este inconveniente creando una segunda, e incluso una tercera, área de configuración que puede ser sencillamente utilizada actuando sobre las flechas presentes al área de configuración (Configuración ◀ ▶).

Si un dispositivo tiene solamente un área de configuración, las flechas presentes estarán inhabilitadas, o bien no están activas. Si un dispositivo tiene más áreas de configuración, las flechas estarán activas y será posible cambiar y visualizar las otras áreas de configuración. Cuando se selecciona un dispositivo, el IC-Prog sabe automáticamente si utilizar una ventana individual o más ventanas para el área de configuración.

Todos los dispositivos tienen al menos un área de Código, por lo tanto, esta parte de la ventana principal siempre está visible.

Algunos dispositivos como las EEPROM, no tienen un área de Datos, en estos casos, el IC-Prog utiliza todo el espacio disponible para visualizar el área de código.

Otros dispositivos no tienen mucha información de configuración y en tal caso el IC-Prog dejará un espacio vacío en el área de configuración: siendo solo visible el valor del *Checksum*.

2.4 Operaciones estándar:

IC-Prog se configura fácilmente, se abre un fichero y luego se programa lo sobre el dispositivo.

Para abrir un fichero, pulsar sobre el menú de la barra de herramientas en **archivo**, y elegir **Abrir datos** : en estas condiciones se abre la ventana de diálogo para abrir el archivo: IC-Prog asume por defecto el filtro "abrir archivo (*.*)", por lo tanto se ven todos los archivos.

Se selecciona el archivo deseado, y se pulsa sobre el botón '**Abrir**'. Cuando se abre un archivo, se ve toda la información contenida en las distintas áreas anteriormente descritas, aparecerá por lo tanto, el área datos, el área de código y el área o las áreas, de configuración, si el dispositivo las posee.

Se puede cambiar eventualmente el contenido del en el área datos o del área de código. Para editar o modificar el archivo, seguir las instrucciones contenidas en la sección Editar el buffer.

Hay que tener presente, que el valor del carácter será puesto en la parte más baja de la palabra. Por ejemplo si se digita el carácter 'A'. el valor de la palabra es 0041 en hexadecimal (sufijo h), que es igual a 65 en decimal (sufijo d).

3 Configuración de IC-Prog

El IC-Prog ha sido diseñado para poder funcionar con numerosos programadores.

Para ver los programadores utilizables y los ajustes de los mismos, seleccionar del menú **Ajustes>Tipo Hardware**

3.1 Interfaz

El software IC-Prog puede controlar al programador a través de acceso directo del puerto al que el programador esté conectado, o bien por las API de Windows (*Application Program Interface*). El empleo de las API se permite solamente para los puertos de comunicación serie.

3.2 Empleo de las API bajo Windows NT/2000.

El empleo del puerto paralela no está permitido en los sistemas operativo Windows NT / 2000, (debido al HAL - *Hardware Abstraction Layer*), debido a que se realiza "llamada" de nivel ring 0 sin autorización. Intentos de control del puerto paralelo bajo tales sistemas operativos, generan un error de "instrucción privilegiada."

3.3 Comunicación

El IC-Prog permite la inversión de las siguientes señales, para poder garantizar una más amplia compatibilidad Hardware de los programadores y los dispositivos.

DATA OUT
DATA IN
CLOCK
MCLR
VCC
VPP

3.4 Prioridad del proceso

- Normal
- Alto
- Tiempo real

La prioridad del proceso habilita al IC-Prog, para calcular, según los distintos casos descritos, tiempos de reloj más adecuados para la programación del dispositivo.

Hay que tener presente que poner la prioridad de proceso sobre el valor Alto, significa crear disminuciones de velocidad en la ejecución de otros programas, porque la CPU dedica más tiempo a la ejecución de IC-Prog, en lugar de ocuparse de otros procesos. El valor por defecto es Normal.

3.5 Cambio del tiempo de retardo de I/O

Cuanto más veloz sea el ordenador, más alto tendrá que ser el retardo de I/O.

Normalmente, por ejemplo usando un Pentium III 650 MHz, con un retardo de I/O de 4 es suficiente.

4 PROCEDIMIENTO

4.1 ARCHIVOS

4.1.1 Abrir Archivo

Este comando carga el código del archivo seleccionado y lo guarda en el buffer de memoria RAM.

El IC-Prog soporta archivos hex en formato Intel de 8-bits y de 16-bits, archivos objeto (obj), y archivos binarios.

Nota.-. Se permite que algunos ensambladores que generan archivos hex no estándar. En estos casos, hace falta seleccionar, el filtro "Any archivos (*.*)" para encontrar el archivo.

Si el archivo hex no incluye informaciones de estándar para el estado de los bits de configuración del dispositivo, serán utilizados estos estándares.

Hace falta indicarle explícitamente a IC-Prog que tipo de archivo se quiere abrir.

Los tipos de archivos actualmente soportados son:

- IHX8 (*. hex, *. h8, *. Sxh)
- IHX16 (*. hex, *. h16, *. Sxh)
- Vía (*. bin)
- Objeto (*. Obj)
- archivos Motorola S (*. mot, *. s19)
- archivos EEPROM (*. e2p, *. eep)
- Any expediente (*. *)

El filtro "Any archivo (*.*)" es de tipo AUTODETECT y permite seleccionar cualquier presente en el directorio.

IC-Prog asume por defecto el filtro "Any archivo (*.*)"

Si se encuentran problemas en la apertura de un archivo, probar a abrir dicho archivo con un tipo de archivo específico de los disponibles. "Any archivo (*.*)" puede abrir los siguientes tipos de formatos: IXH8, IHX16, OBJ Motorola y binario.

Nota.-. Los archivos tipos E2P, EEP no pueden ser vistos a menos que sean otro archivo IHX.

4.1.2 Guardar como

Este comando salva el contenido del buffer de memoria seleccionado en un archivo Hex.

El archivo salvado contiene el programa y los datos de memoria, configuración de los *fuses* y palabra ID.

4.1.3 Abrir Datos

Este comando carga en el buffer del área de datos el archivo seleccionado.

4.1.4 Archivos Recientes

Este comando visualiza el listado de los últimos 8 archivos usados.

4.1.5 Imprimir

Este comando imprime el área de datos del buffer de memoria seleccionado.

4.2 Edición

4.2.1 Llenar Buffer

Este comando sólo llenará el buffer de memoria de código con un valor. El valor puede ser elegido entre decimal, hexadecimal o ASCII.

4.2.2 Llenar Intervalo de Buffer

Este comando sólo llenará el buffer de memoria de código con un valor entre unas direcciones definidas. El valor puede ser elegido entre decimal, hexadecimal o ASCII

4.2.3 Copia del buffer

Este comando copia los datos seleccionados del buffer de memoria en el portapapeles.

4.2.4 Pega en el buffer de memoria

Este comando pega en la posición actual el área de memoria del portapapeles en el buffer de memoria.

4.2.5 Llenar Determinado Buffer

Este comando llenará los datos del buffer de memoria con un único valor. El valor puede elegirse entre decimal, hexadecimal o ASCII.

4.2.6 Llenar intervalo Buffer de...

Este comando llenará el buffer de datos con un único valor entre una serie de direcciones de memoria. El valor puede elegirse entre decimal, hexadecimal o ASCII.

4.2.7 Copia del Buffer de datos

Este comando copia en el portapapeles los datos contenidos en el buffer de memoria seleccionado.

4.2.8 Pega al Buffer de datos

Este comando pega los datos del portapapeles al buffer de datos en la posición actual

4.3 BUFFER

4.3.1 Buffer activo

Este comando permite la selección un buffer de memoria.

4.3.2 Compara Buffer

Este comando permite comparar el buffer de memoria actual con otro buffer de memoria.

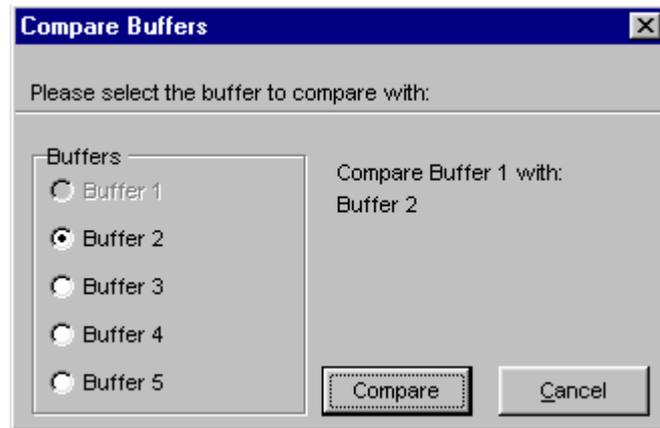


Figura 5.- Pantalla de selección para comparar buffer de memoria

4.4 AJUSTES

4.4.1 Dispositivos

Este comando permite seleccionar el dispositivo a programar.

4.4.2 Dispositivos Recientes

Este comando visualiza la lista de los últimos 8 dispositivos utilizados.

4.4.3 Hardware

Este comando habilita IC-Prog para la configurar el hardware que se quiere utilizar.

El tipo de Hardware, interfaz, puerto y parámetros de comunicación, se definen en esta función:

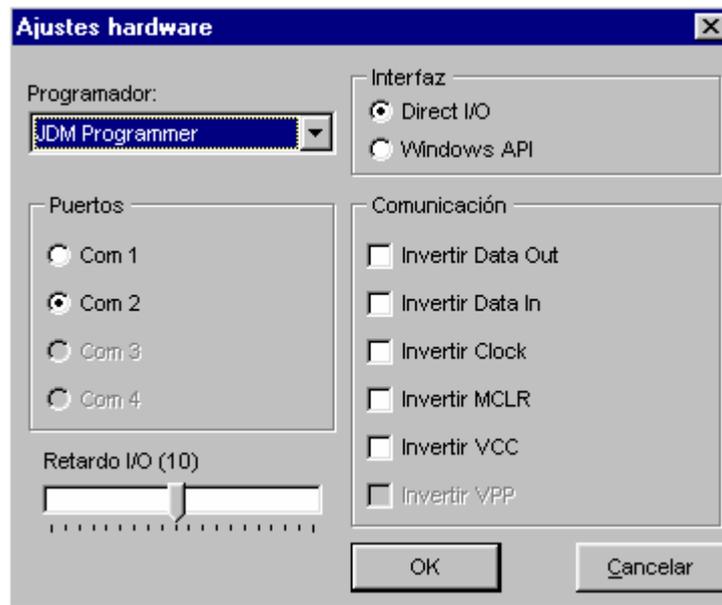


Figura 6.- Menú de configuración del hardware

4.4.4 Control Hardware

Este comando controla si el programador comunica correctamente con IC-Prog.

4.5 OPCIONES

4.5.1 Confirmación

Cuándo el IC-Prog realiza una acción, es posible habilitar la opción de que pregunte la confirmación de dicha acción, para asegurar que el dispositivo no sea programado erróneamente, o bien que no sea anulada la fase de programación del dispositivo.

Hay 4 acciones que permiten habilitar la confirmación con un mensaje:

- Programación del dispositivo
- Borrar un dispositivo
- Protección del código de un dispositivo crítico.
- Sobrescribir un archivo

La protección del código de un dispositivo crítico: Algunos dispositivos son programables una sola vez, es decir, son "*One Time Programmable*." (OTP)

Cuando se programa este tipo de dispositivos, no es posible leer el contenido que tenía antes y esta operación no se puede anular. Si el código que se graba en el dispositivo es, por algún motivo, errado, se puede tirar a la basura, porque es inutilizable.

Con esta opción se verifica que el dispositivo no esté accidentalmente escrito, a menos que no sea preguntado explícitamente al programar el dispositivo.

4.5.2 Avisos

El IC-Prog puede notificar acontecimientos. Hay actualmente puede avisar de:

- Cuando el archivo no contiene la palabra ID
- Cuando el fichero no contiene la palabra de Configuración en el archivo cargado.

Cuando la palabra ID o la palabra de Configuración se pide en el archivo, IC-Prog asume por defecto FFFF.

Para ser conscientes de que el valor cargado no es FFFF h, pero sencillamente que ha sido perdido el valor en el archivo usado, se puede habilitar la opción de aviso correspondiente.

4.5.3 I2C

Sobre este cuadro de diálogo, es posible configurar las opciones que se refieren a los dispositivo I2C, por ejemplo las 24Cxx.

- **Habilitar MCLR junto a VCC:**

Cuando se lee o se programa un dispositivo 24Cxx con una interfaz de tipo *Ludipipo*, la señal MCLR hay que ponerla a nivel lógico alto. Debido a que MCLR necesita estar a nivel alto para crear la tensión de alimentación, VCC.

- **Habilitar PAGE-WRITE**

Normalmente un dispositivo 24Cxx puede ser programado de 2 modos: byte a byte y por escritura de página. La escritura de página es recomendable por ser mucho más rápida, pero desafortunadamente, no todos los dispositivos soportan esta característica.

Por lo tanto, si se posee un dispositivo que no permite seleccionar el modo de escritura de página, inhabilitar esta función. En caso contrario, se aconseja dejar habilitado esta función.

Dirección Hardware:

Cuando se lee o se programa un dispositivo 24Cxx hace falta diseccionarlo.

Los pines de entrada A0, A1, y A2 del dispositivo configuran el direccionamiento y normalmente se conectan a GND, los enlaces son hechos en el zócalo del programador, de este modo, la dirección hardware de estándar es 0.

En el caso en que se necesitara partir con una dirección hardware diferente de 0, se programan los pines de entrada según indica el *data shell* del fabricante, y programar el valor correspondiente en el IC-Prog.

4.5.4 Programación

Verificación de dispositivos:

IC-Prog permite comprobar la programación del dispositivo en dos momentos diferentes:

- Después de que ha sido programado
- Durante la fase de la programación;

Solamente la programación de los PIC permiten la verificación durante la programación

Por defecto, IC-Prog primero programa el dispositivo, finalizada la programación, se rehabilita poco después, e inicializa la fase de verificación.

De este modo, IC-Prog asegura que la programación del dispositivo ha sido efectuada correctamente. Si un microcontrolador PIC está protegido contra la lectura del código, las informaciones programadas no pueden ser vistas en la fase de verificación y por tanto, la rutina de verificación no puede realizarse y dará un error.

Si se elige la opción "Verificar durante la programación" cada palabra programada será verificada en al mismo tiempo, siempre que no esté seleccionada la opción de Protección de Código.

En práctica la verificación de los datos grabados habilitación de la protección de código ocurre en la misma rutina.

4.5.5 Atajos

Con IC-Prog se pueden utilizar 4 atajos predefinidos con la combinación de teclas CTRL+F1, CTRL+F2, CTRL+F3 y CTRL+F4 para seleccionar los dispositivos.

4.5.6 Arrastrar y pegar (Drag & Drop)

IC-Prog permite arrastrar los archivos de gestión de aplicaciones Windows al área de Código o al área de Datos. Hace falta seleccionar un tipo de archivo específico por cada una de las área. Es posible también, utilizar el filtro "Any expediente (*.*)" Este tipo de filtro es AUTODETECT, para mayores informaciones consultar la sección Abrir Archivo.

4.5.7 Smartcard

IC-Prog permite la utilización de los programadores compatibles "Phoenix." Realmente el programador Phoneix no es considerado un verdadero programador por ello, no es enumerado en el menú Hardware. Sobre la pantalla desplegada, es posible configurar los estados de nuestro smartcard.

➤ Puerto

Selecciona el puerto serie que queremos conectar al interfaz Phoneix

➤ Invertir Reset

La señal de Reset que utiliza el Phoenix puede ser invertido. Esta opción se hace necesaria cuando la interfaz Phoneix invierte la señal de Reset.

IC-Prog invierte la señal de Reset y la interfaz de Phoenix invierte esta última de nuevo, de tal modo que se reestablece la señal con un valor normal. Los programadores Phoenix que tienen la señal de Reposición invertida a menudo son denominados programadores "Smartmouse."

➤ Protocolo

IC-Prog permite actualmente 2 tipos de *boot loader*: compatible Multimac y compatible Secanix. Seleccionar el tipo de *Boot Loader* que se quiere utilizar con el IC-Prog.

➤ Dispositivo:

IC-Prog necesita saber que tipo de dispositivo programa en el Smartcard Wizard. Normalmente el dispositivo utilizado es un 16C84 o 16F84.

Seleccionar el dispositivo que está presente sobre la pantalla.

➤ Frecuencia:

La interfaz Phoenix normalmente está dotado de dos cristales de cuarzos uno a 3.57-8 Mhz y el otro a 6MHz. Elegir el cuarzo que se desea utilizar.

4.6 Lengua Utilizada

IC-Prog dispone de nutrido número de idiomas diferentes. Seleccionar el idioma deseado en la caja de visualización. Algunos idiomas solicitan caracteres que no se localizan en el menú de caracteres predefinido.

IC-Prog usa los caracteres para el idioma seleccionado, pero no por menú. Si se desea también visualizar los caracteres correctos por los menús, hace falta cambiar los *font* de Sistema para usar el menú de caracteres deseado. Esta operación se puede realizar en el Panel Control.

4.7 Shell

La página de los Shell de IC-Prog permite asociar con IC-Prog, las extensiones de los archivos más utilizados con la opción "Habilitar extensión Shell"

Esto quiere decir que si se pica dos veces con el ratón sobre un archivo asociado en Explorar Recursos, este archivo se abre con IC-Prog.

También puede usarse la alternativa del botón derecho del ratón, Explorar Recursos, y seleccionar la ventana que habilita el empleo del Shell.

4.8 Miscelánea

En esta ventana es posible configurar todas las opciones suplementarias que no pudieron ser puestas sobre otras ventanas.

Prioridad del proceso:

En todas las aplicaciones de Windows puede ser definido un nivel de prioridad. Generalmente el nivel de prioridad de una aplicación es puesto sobre Normal. Si es necesario aumentar el nivel de dedicación de IC-Prog, utilizar una prioridad alta. Atención: aumentando el nivel de prioridad, hace el sistema operativo y los demás procesos en ejecución muy lenta, ya que la UCP se ocupa principalmente de los procesos de IC-Prog, dedicando menos tiempo a los demás procesos en ejecución. Si no es necesario tener una velocidad de proceso elevada, programar la prioridad del proceso sobre Normal.

Habilita Driver NT/2000:

Normalmente IC-Prog usa el I/O directo para comunicar con los diferentes tipos de programadores. Con los sistemas operativo Windows NT y 2000 el acceso para utilizar directamente el I/O no se permite y en el *fattispecie* genera una instrucción Privilegiada. Este driver especial del IC-Prog permite utilizar el I/O directo sobre estos sistemas. Cerciorarse que el driver "icprog.sys" se encuentra en el mismo directorio que "icprog.exe". Si se selecciona esta opción IC-Prog instalará el driver en el sistema operativo en uso.

Habilita control VCC por JDM:

Generalmente los programadores JDM, o compatibles, no pueden controlar la tensión VCC. Esto significa que la tensión de alimentación siempre está activa. Algunos microcontroladores PIC no puede ser programados correctamente sin que la tensión de alimentación sea quitada.

Esta opción habilita IC-Prog a controlar la tensión VCC sobre el programador JDM. Esta opción en todo caso no garantiza un trabajo seguro, y por tanto es experimental y sólo para usuarios avanzados. Se aconseja no usar esta opción, porque puede causar fallos de la programación de otro tipo de dispositivos.

Pones todas las salidas a nivel alto (Combi):

Hay algunos programadores especiales que tienen una interfaz paralelo (LPT y una serie, COM).

Para trabajar juntos correctamente, la interfaz paralela necesita ser puesta a nivel alto, cuando está en el estado de no utilización. El empleo de estos medios de selección significa que no ha sido conectado/desconectado el programador y se quiere cambiar la puerta LPT o la puerta COM.

4.9 SmartCard (Phoenix)

Seleccionando la opción Smartcard, Phoenix, IC-Prog se selecciona para usar el hardware Phoenix, a través del archivo de carga apropiado, para programar la EPROM sobre un Smartcard.

Nota.- Esta opción puede ser sólo usada con un puerto serie, porque el tipo de comunicación es asíncrono - estándar (Phoenix).

4.10 Borrar ajustes

Este comando restablece todos los ajustes de IC-Prog llevándolos al estado estándar.

4.11 COMANDOS

4.11.1 Leer Todo

Este comando leerá los datos del dispositivo y los cargará en el buffer de memoria.

Los *fuses* serán programados por el dispositivo utilizado.

Si el dispositivo está protegido contra la lectura de código, la lectura del Código protegido, en el buffer, será un código inutilizable.

Cuando se lee de un microcontrolador cuyo código está protegido (CP), se leen valores diferentes, en todo caso inútiles, según el microcontrolador utilizado.

Por ejemplo los 16C84 llena el buffer con 7FF.

Otros microcontroladores rellenan el buffer con 00 y los microcontroladores AVR rellenan el buffer con 11, 22, 33, etcétera Actualmente no es posible descubrir con el IC-Prog, si los microcontroladores AVR (AT90xxx) o AT89xxx tienen el código protegido.

Se puede saber solamente si el código que se ha leído, incorrecto o es correcto, por consiguiente los bites de los *fuses* para el AT90xxx no pueden ser borrados cuando se lee de uno de estos dispositivos.

4.11.2 Programar Todo

El comando programará el dispositivo con los contenidos del buffer.

El buffer tiene que estar cargado con datos, por el comando Abrir Archivo o por la lectura de un dispositivo ya programado.

Este comando también programa el dispositivo con la configuración expuesta sobre la pantalla.

4.11.3 Programar bit de configuración

Este comando programará los *fuses* del dispositivo con la configuración visualizada en el área de configuración.

4.11.4 Borra Todo

Este comando borra todo el contenido del microcontrolador.

4.11.5 Verifica Blanco

Este comando lee el contenido del microcontrolador y verificación que este último esté vacío.

Nota .- cuando el programa lee por lo tanto todo FFFF h, significa que la memoria del microcontrolador está vacía, mientras si lee 0000 h , la memoria del microcontrolador no está vacío.

4.11.6 Verificar

Este comando averigua que el contenido del dispositivo programado resulta igual al contenido del buffer de memoria seleccionado por el IC-Prog.

4.11.7 Asistente SmartCard

Con este procedimiento, un Smartcard se programará con un Boot Loader que encaminará la programación del EEPROM por el PIC.

El archivo a grabar en la EEPROM podrá ser elegido durante este proceso.

4.12 HERRAMIENTAS

4.12.1 Cargar Ficheros CDR

Este comando habilita la carga de un CRD.

4.13 VER

4.13.1 Ensamblador

Este comando visualiza el área de código en lenguaje ensamblador.

Nota.- La visualización del ensamblador puede utilizarse solamente para los microcontroladores (PIC) de 12 y de 14 bits

4.13.2 Hexadecimal

Este comando visualiza el área de código en formato hexadecimal.

4.13.3 Posición del dispositivo

Este comando visualiza la posición de inserción del dispositivo para el programador seleccionado.

Ejemplo: microcontrolador PIC 16F84



4.13.4 Informaciones sobre el microcontrolador

Este comando visualiza los detalles y las características del dispositivo que se va a programar.

4.13.5 Refrescar

Este comando refresca la ventana principal de IC-Prog.

4.14 AYUDA

4.14.1 Acerca de...

Este comando visualiza la versión utilizada y la información para contactar con el autor de IC-Prog.

5 REFERENCIAS

5.1 Programadores utilizados

Los siguientes programadores son manejados correctamente por el software IC-Prog:

- Programador JDM (Ludipipo)
- Programador Conquest
- Programador TAFE
- Programador Paralelo TAIT 'Clásico'
- Programador TAIT
- Programador Fun-card
- Programador SCHAER
- Programador ProPic II
- Programador STK200
- Programador AN589
- Programador WILLEPRO
- Programador Fluffy
- Programador DL2TM

5.2 Informaciones específicas para los programadores

En esta sección se indican algunas sugerencias para hacer funcionar de la mejor manera el programador utilizado.

5.2.1 Programador JDM (Ludipipo)

- Cuando los EEPROM se programan 24XX16 con el programador JDM, desconecte el pin N° 7, WP (*Write Protect*) Protección de Escritura, de la EEPROM unido en el zócalo, y poner este pin a masa, GND (*Ground*).
- Se aconseja de no utilizar la opción de (*Enable VCC control for JDM*) Habilita control VCC por JDM. Esta opción todavía está en fase de experimentación y por tanto no debe utilizarse, para usos comunes. Inhabilitarla.

5.2.2 Dispositivos que soporta:

Aquí se indica la relación de los dispositivos normalmente utilizados.

EEPROM 4-Hilos:

59C11, 59C22, 59C13

EEPROM I2C:

24C01, 24C02, 24C04, 24C08, 24C16, 24C32, 24C64/65, 24C128, 24C256, 24C512

PCF8572 or 8572 = 24C01

PCF8582 or 8582 = 24C02

PCF8592 or 8592 = 24C04

SDE2516 = 24C??

SDE2526 = 24C??

SDA2546 = 24C??

SDA2586 = 24C??

SDA3546 = 24C??

SDA3586 = 24C??

24LC21 = 24C01

¡Pin7, VCLK, tiene que estar conectado a VCC!!. Soporta toda la serie C y LC .
La serie CS no se ha utilizado todavía.

Microcontroladores Flash:

89C1051, 89C2051, 89C4051

EEPROM IM-Bus:

NVM3061 MDA2060, MDA2061, MDA2062

En estos momentos en fase experimental

Dispositivos Microchip PIC:

12C508, 12C508A, 12C509, 12C509A, 12CE518, 12CE519 12C671, 12C672, 12CE673, 12CE674, 16C433, 16C54, 16C56, 16C61, 16C62A, 16C62B, 16C63, 16C63A, 16C64A, 16C65A, 16C65B, 16C66, 16C67 16C71, 16C72, 16C72A, 16C73A, 16C73B, 16C74A, 16C76, 16C77, 16C84, 16F83, 16F84, 16F84A 16C505 16C620, 16C621, 16C622, 16C622A, 16F627

16F628 poner RB4 a GND.

16C715 16F870, 16F871, 16F872, 16F873, 16F874, 16F876

16F877 poner RB3 a GND.

16C923, 16C924

EEPROM Microwire:

93C06, 93C46, 93C57, 93C56, 93C66, 93C76, 93C86

93C13 = 93C06

93C14 = 93C46

Toda la serie C e LC está confirmada.

La serie CS no está comprobado todavía

EEPROM serie modernas:

AK6420, AK6440, AK6480 BR9010, BR9020, BR9040, BR9080

Ahora en vía experimental

Dispositivi Scenix (solo rev 5.2):

SX18AC, SX28AC

Ahora en vía experimental

EEPROM SPI:

25010, 25020, 24040 25080, 25160, 25320, 25640

Microcontrollore SPI:

90S1200, 90S2313, 90S2323, 90S2333, 90S2343 90S4414,
90S4433, 90S4434, 90S8515, 90S8535 89S53, 89S8252.

5.2.3 Dispositivos verificado con éxito

Programador JDM

24C01, 24C02, 24C04, 24C08, 24C16
24C32, 24C64/65, 24C256
59C11
93C46, 93C06, 93C66
PIC 16C84, 16F84, 16F84A
PIC 16C71, 16C72
PIC 16C64A
PIC 12C508, 12C508A, PIC12C509, 12CE518
PIC 12CE674
PIC 16C505
PIC 16F872

Es importante mantener a nivel bajo el Pin PB3 de los 872 mientras es programado. La primera vez hace falta inhabilitar esta función,

PIC 16F877, 16F876, 16F874, 16F873
PIC 16C715
AT90S8515
AT90S2323
25020
89S53

Programador CONQUEST

- 24C16
- PIC 16F84

Programador TAFE

- AT89C2051

Programador seriee TAIT

- 24C16, 24C32, 24C64/65
- PIC 16C84, PIC16F84, 16F84A
- PIC 12C508, 12C508A,12C509, 12C509A
- PIC 16C622A
- PIC 16F873, PIC 16F874, PIC 16F876
- PIC 16C622A, PIC 16C63A, PIC 16C66

Programador paralelo TAIT

16C54

Programador FUN-CARD

AT90S1200
AT90S2313-10PI
AT90S2323-10PI
AT90S2343-10PI
AT90S8515-8PI

Programador SCHEAR

16F873
16F84, 16F84A
24C16, 24C32

Programador STK200

90S1200
90S1200A
90S2313
90S8515

Programador ProPic II

(Propic I : 12C508A and 16F84)
12C508A, 16F84, 16F84A
24C04, 24LC16
12C509A
PIC16F870 , PIC16F872, PIC16F873 , PIC16F876, PIC16F877
¡Hace falta conectar Rb3 a masa, GND!!! OK.

93C06, 93C46 (**Necesita de adaptador**)

5.3 Opciones de líneas de mando

Direccionar el IC-Prog con la línea de mando siguiente:

5.3.1 IC-Prog Opción1 Opción2 etcétera

Opciones:

- **p** Programa el dispositivo
- **v** Verificación el dispositivo
- **r** Lenguaje del dispositivo
- **I** Devuelve la ventana escondida
- **x** Habilita la programación de las SmartCard (Por la programación del PIC)
- **t00** Tipo de la instrumentación, dónde 00 puede ser cualquier dispositivo válido, * No probado!!! *
- **g00** Tipo del programador, dónde 00 puede ser cualquier programador válido, * No probado!!! *
- **o0** Tipo de puerto utilizado, dónde 0 puede ser cualquier numero del 1 al 4 que corresponde a LPT o a COM,) * No probado!!! *
- **INOMEFILE** Cargar un archivo, actualmente solo pueden ser cargados archivos IHX8,

- **sNOMEFILE** guarda un archivo como NOMEFILE, actualmente solo pueden ser salvados solamente archivos IHX8,
- **q** Salir después del último comando
- **fxxxx** Fundidos Word = xxxhx
- **ft?** Selecciona un tipo de archivo específico: 0 = cualquier archivo, 1 = IHX8, 2 = IHX16 etcétera El valor 0 es usado por defecto.

Ejemplo:

IC-Prog lprova.hex - p - v - los - f3EEA - q

El archivo *prova.hex* es cargado, a su vez programada (- p, con la modalidad Word indicado (- f3EEA) pregunta con el mensaje de confirmación ‘¿Quieres verdaderamente programar el dispositivo?’

Sucesivamente el dispositivo será verificado (- v, con la aparición del mensaje ‘verificación efectuada con éxito!’)

Cuándo IC-Prog ha acabado su ejecución, será cerrado automáticamente (-q.)

5.4 Edit el Buffer

Hay muchos modos para editar el buffer:

- Se entra en el área del buffer de memoria. Si se introduce un valor que supera la dimensión de una palabra, word, por el dispositivo utilizado, IC-Prog la cambiará automáticamente. Por lo tanto, si se introduce por ejemplo FABC y se utiliza un PIC a 14bit IC-Prog cambiará el carácter F principal en un 3.
- Haciendo doble click en una posición de memoria. Se' pueden escribir cifras decimales, hexadecimales o el valor en ASCII.
- Usado el Menú Edit. En caso es posible escribir en el buffer, para llenarlo todo o también en un punto específico. Usando la tecla derecha del ratón. Será posible aplicar los mismos métodos indicados anteriormente.

5.5 Informaciones técnicas sobre los PIC

La mejor fuente de informaciones sobre los PIC es Internet.

En Internet es posible encontrar las respuestas a las preguntas más frecuentes, FAQ, especificaciones, precauciones en el empleo, ensambladores, simuladores, programas experimentales, y... ¡mucho más!!!

Algunos enlaces interesantes son recomendados seguidamente.(Si estos enlaces por alguna razón, estuvieran mal o cancelados, se aconseja utilizar los motores de búsqueda disponible en Internet, Altavista, Google, Lycos, etc...)

Home page IC-Prog Software: www.ic-prog.com

Email IC-Prog Software: bgijzen@wanadoo.nl

David Tait's enlaces de PIC: <http://www.man.ac.uk/~mbhstdj/piclincs.html>

Tecnología Microchip: <http://www.microchip.com/>

WinPhoenix: <http://www.thoic.com/decrypt/>

5.6 SmartCard (WaferCard)

Originariamente los SmartCard se basaban en un solo PIC, recientemente han sido dotadas con un PIC y un EEPROM para aumentar la capacidad de memoria disponible, debido a que para algunas aplicaciones la memoria presente en el PIC no es suficiente..

Típicamente un GoldCard contiene un PIC 16x84 y un EEPROM 24LC16. El PIC denominado con la sigla 16x84 puede ser un 16C84 o el modelo más nuevo, el 16F84.

Las EEPROM 24LC16 es a 16K bites, 2K Byte, (256x8x8 bit).

Los 16x84 tienen 1024 byte de código y 64 byte de área de datos.

5.6.1 Selección del oscilador

Los tipos de Osciladores o Clock, son 4 y seleccionan la modalidad de funcionamiento del amplificador interior del pic sobre la entrada del clock.

XT--> funcionamiento Oscilador a cuarzo Externo, estándar,

La lista completa es la siguiente:

Sigla	Nombre completo	Descripción
LP	Low Power	Oscilador de cristal de cuarzo o resonador cerámico de baja potencia
XT	Cristal/Resonador	Oscilador o resonador cerámico. Se trata de un oscilador estándar que permite una frecuencia de reloj entre 100KH < y 4 MHz
HS	High Speed Cristal/Resonador	Oscilador de cristal o resonador de alta velocidad. Es un oscilador de una frecuencia de reloj comprendida entre 4 MHz y 20 MHz
RC	Resistor/Capacitor	Oscilador de bajo coste, formado por una resistencia y un condensador

5.6.2 WDT, (*Watch Dog Temporizador* - temporizador del perro guardián)

Siempre está puesto en OFF.

5.6.3 PWRT (*Power on Reset*)

➤ debe ser puesto en ON para el PIC 16C84

➤ debe ser puesto en OFF para el PIC 16F84

El bit PWRT tiene que ser invertido la segunda vez que el PIC 16C84 o bien el PIC 16F84 se utilizan.

Por ejemplo, si es programado bajo IC-Prog un microcontrolador del tipo 16C84, y se carga un archivo hex Intel, por ejemplo de un archivo o de otro microcontrolador, y luego se modifica bajo el IC-Prog, el dispositivo por ser un 16F84,

se verá que la condición de PWRT se invierte, señal que se ve sobre el apartado de control PWRT.

5.6.4 RB6 contra RB7

RB6/RB7 identifican el pin del PIC que son comúnmente usados para comunicar con el pad del Smartcard (Para las GoldCard - MultiMac típicamente se usa RB7)

El archivo Hex por el PIC puede tener dos condiciones, RB6, RB7 o ambas, es importante, por lo tanto, seleccionar el código hex correcto para el mismo Smartcard.



Figura 7.- Gold Smartcard (o Wafer Card)

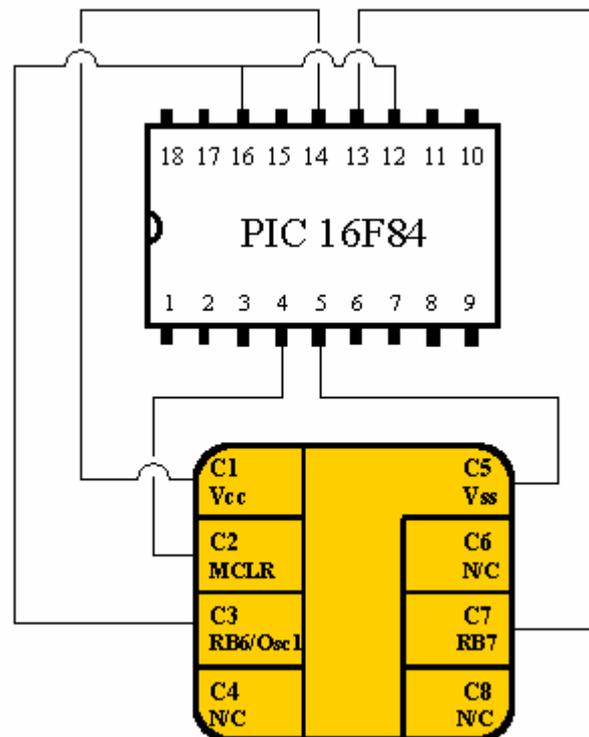


Figura 8.- conexión de una tarjeta que contiene solo el PIC

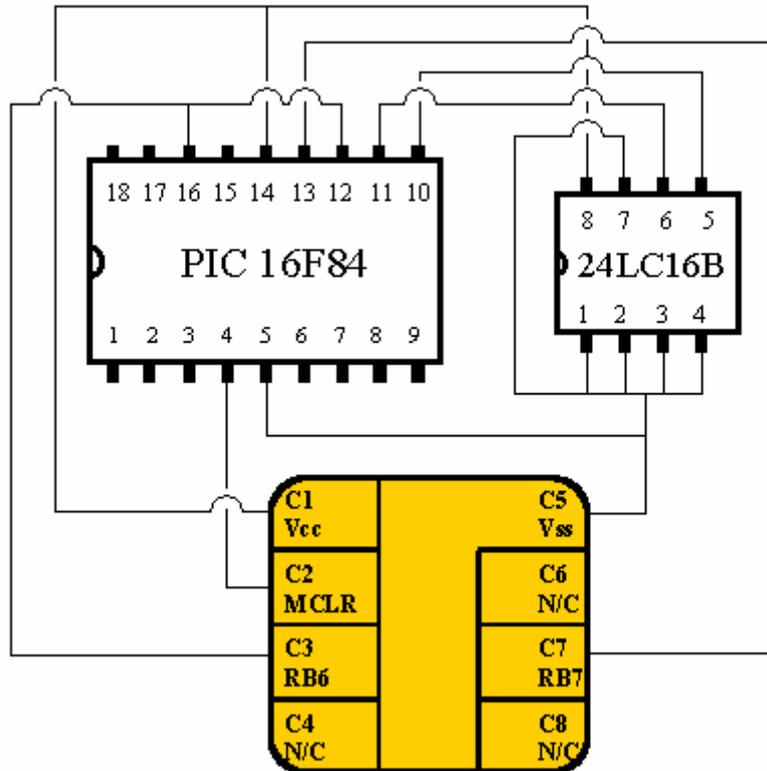


Figura 9.- Conexión de una tarjeta conectada a un PIC 16F84 y una EEPROM 24LC16

5.6.5 Para la Programación de PIC

La programación directa del EEPROM no es posible con los Smartcards cuyo conexiones de la EEPROM no están directamente conectados a los bloques sobre el Smartcard.

Permite solamente programar la EEPROM a través del el PIC. IC-Prog utilizando la programación de la EEPROM a través del PIC: hace falta averiguar si el programador utilizado tiene la capacidad de programar las EEPROM. A tal fin es aconsejable utilizar un programador Phoenix o Smartmouse para programar la EEPROM.

5.7 ADAPTADORES

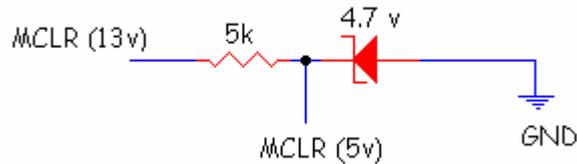
5.7.1 Adaptador MicroWire para JDM

El programador JDM respalda la tecnología MicroWire, sencillamente añadiendo los siguientes componentes:

- Una resistencia de 5K Ohm,
- Un diodo Zener de 4,7 Voltios.

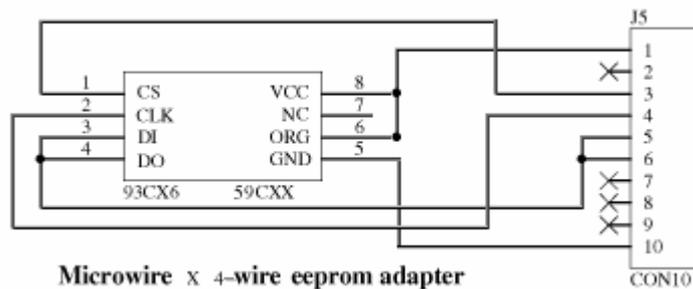
Conectar la señal MCLR a un terminal de la resistencia. Conectar al otro terminal de la resistencia con el cátodo del Zener, conectar el ánodo del Zener a masa,

(GND). El cátodo del Zener (que está conectado a la resistencia): este último será la nueva señal MCLR.



¡De ahora en adelante, se tendrá una señal MCLR de 5 solos Voltios en vez de de 13 Voltios!

Seguidamente se representa el adaptador para los dispositivo MicroWire, 93Cxx y 4-Wire, 59Cxx,:



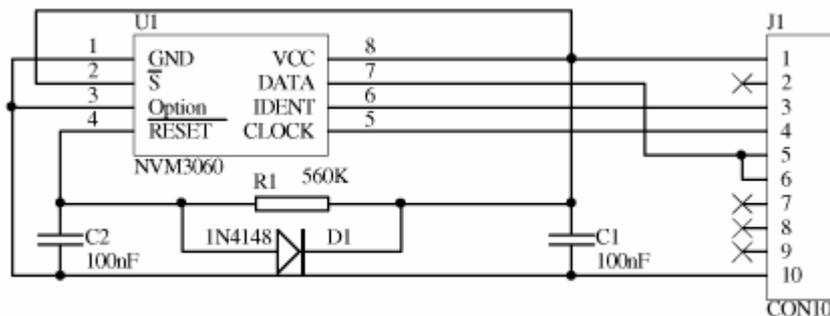
- 1 = VCC
- 3 = MCLR (5v)
- 4 = CLOCK
- 5 = DATA OUT
- 6 = DATA IN

(En la mayor parte de los programadores las señales de DATA-IN y DATA-OUT coinciden con una única señal. Si no coincidieran, conectarlos como se muestra en la figura.

5.8 Adaptador NVM 3060

Los NVM3060 son EEPROM serie IM-AUTOBÚSES que pueden ser programada casi con cualquier programador.

Pero hace falta utilizar un adaptador especial



NVM3060 eeprom adapter

Conectar los hilos sobre el conector 10 como se muestra en la figura del programador que se desea utilizar.

1 = VCC

3 = MCLR (5v, para utilizar el adaptador de MicroWire)

4 = CLOCK

5 = DATA OUT

6 = DATA IN

(Si se tienen señales de DATA-IN y DATA-OUT separados, conectarlas juntas.)

6 INFORMACIONES ADICIONALES

6.1 Programar los EEPROM con el programador JDM

Con un programador JDM o con cualquiera otro programador compatible, es posible programar PIC y EEPROM (24Cxx).

Para poder programar tales dispositivos, es obligatorio insertar el PIC o la EEPROM en el zócalo del programador: en la posición y dirección hacia la muesca del zócalo mirar el esquema del programador JDM.

No existe de momento para este programador otros métodos para programar estos dispositivos si no se introducen en el zócalo del mismo programador.

El único modo para poder programar un EEPROM que esté presente dentro de un GoldCard o, soldada sobre un WaferCard, es usar a un programador de SmartCard Phoneix, o compatibles.

En tal caso, para la programación del PIC se utiliza un archivo especial especial, llamado "Loader", es posible escribir y/o leer del EEPROM: ejemplos de estos loader son Multimac303 y otros...

Como se ha dicho anteriormente, para enviar los datos al PIC del archivo, es necesario utilizar a un programador de SmartCard Phoneix, o compatible.

Este programador Phoneix envía los datos en forma serial al PIC, luego por el PIC se programa el EEPROM.

El software a utilizar, junto al programador Phoneix puede ser IC-Prog, o bien una alternativa, el software llamado WinPhoneix.

Nota.- IC-Prog sólo puede ser usado con aplicaciones Loader Multimac, o compatibles.

Parece que los archivos COP usan otra secuencia de comandos...

En todo caso, WinPhoenix se ha especializado en estas cuestiones, por lo tanto, es mejor probar el propio archivo con WinPhoenix.

Bonny Gijzen,

6.2 Controlar si el programador funciona...

Para ver si un programador trabaja correctamente con IC-Prog, es posible utilizar el método siguiente:

Controlar si el programador funciona correctamente con otro software; Si éste no funciona, entonces es un problema Hardware

Si por el contrario funciona, controlar las señales que son invertidos y luego aplicar las mismas modificaciones a IC-Prog.

En IC-Prog es posible controlar las señales del programador y eventualmente "invertir " en el menú Ajustes > Hardware;

Es posible. por ejemplo, averiguar si invirtiendo la señal DataOut, la señal que está presente sobre el programador es de 5 Voltios o bien no (Nota.- Una señal Invertida es invertida...), por lo tanto, de este modo es posible controlar todas las señales que él puede invertir;

Por último, hay que averiguar, cual es el valor del retardo de I/O.

Si el programador no funciona correctamente, es posible aumentar el retraso, normalmente la posición optima es 4, pero es posible aumentar este valor si necesario.

Estas informaciones deberían ser suficientes para probar el propio programador con IC-Prog.

Bonny Gijzen,

7 Agradecimientos

Autor del software IC-Prog: Bonny Gijzen

Autor del archivo de la guía en italiano: Antonio Guerra

Idiomas que existen:

- Español - José de ZdS electrónica
- Francés - Richard Klein a la dirección www.varicap.com/
- Portugués - Gil por la traducción y Fernando Carvalho, el brasileño, por la inicial traducción
- Holandés - Wilfred de Wolf
- Italiano - MM9800 y también a freesat.cjb.net
- Ruso - Sergei y Alex Kokaiko
- Griego - Con Tanidis
- Polaco - Rajmund Komosinski
- Turco - Ayhan
- Alemán - Christoph Heitkamp

Las secciones de esta Guía han sido llevadas a cabo por un archivo creado por Tord Andersson.