

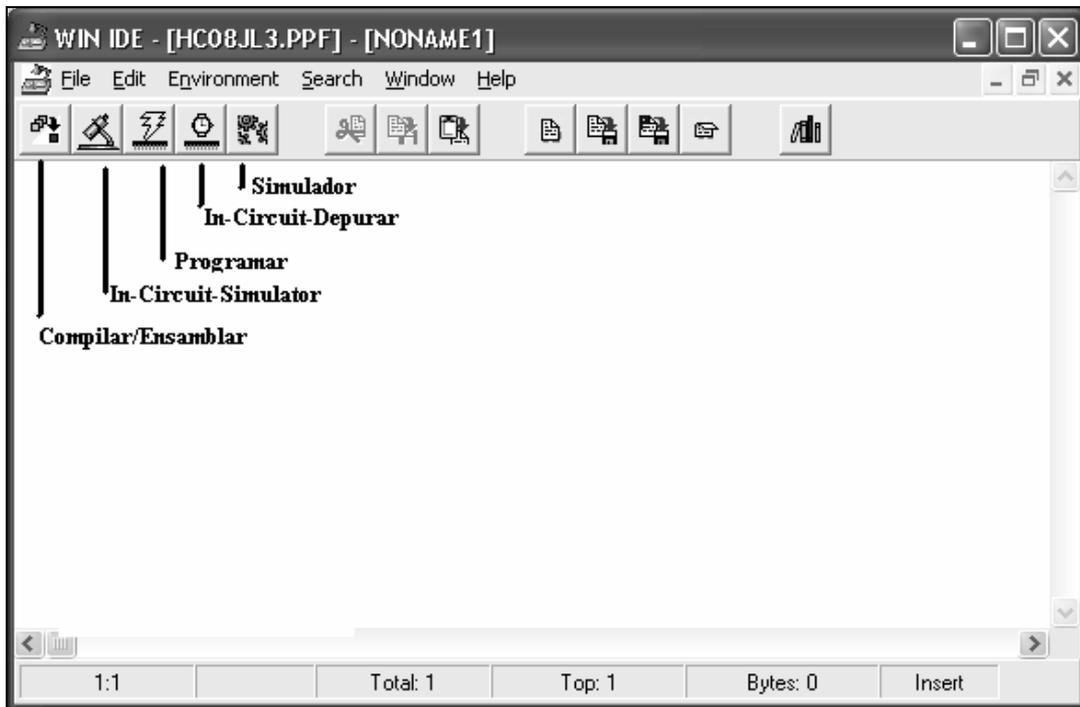
SOFTWARE DE DESARROLLO "WinIDE"

Este documento permitirá al lector conocer el entorno de trabajo correspondiente al software de desarrollo WinIDE y las diferentes herramientas de desarrollo que brinda, como por ejemplo: simulación, Simulación en circuito, Programación del microcontrolador, Emulación, etc.

El software de desarrollo que se puede utilizar para la programación, ensamblado, simulación, edición y depuración de esta familia de microcontroladores se llama WinIDE y se puede descargar en forma gratuita visitando la página www.pemicro.com (Link Motorola M68HC908 kits), en la sección "**LATEST DOWNLOADS**" se escoge la opción "**Find More Downloads**", se escoge la familia deseada, que para el caso es 68HC08 y finalmente se hace clic sobre la opción que diga "**ICS08JLZ v. 1.53**" la cual corresponde al software para desarrollo de aplicaciones con **JK1/JK3/JL3**. Sin embargo, aparece en esta lista todas las herramientas software especializadas para cada referencia de microcontroladores Motorola.

Si se desea encontrar mas información sobre módulos, aplicaciones y otra documentación sobre esta familia, puede visitar la página www.mot-sps.com. A continuación se muestra el formulario correspondiente al entorno de trabajo con WinIDE

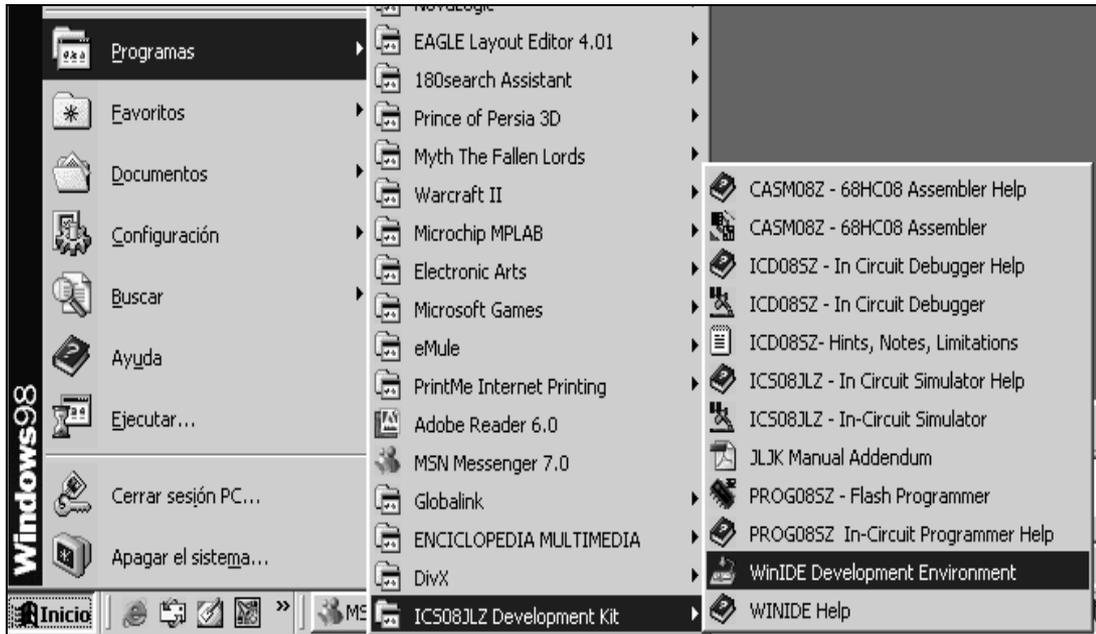
Figura 28. Entorno de Trabajo con WinIDE



Los pasos a seguir para escribir cualquier programa utilizando esta herramienta son los siguientes:

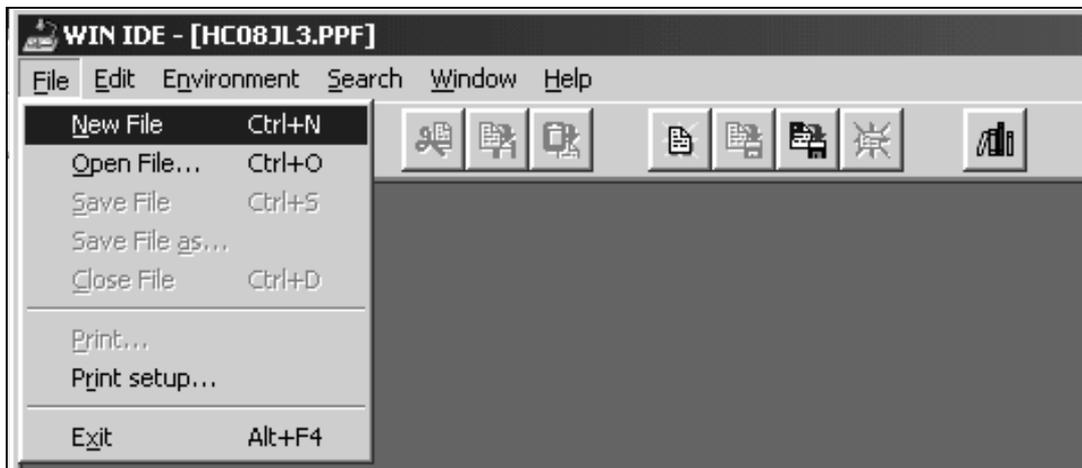
Ejecutar la herramienta WinIDE Development Environment tal como se muestra en la figura

Figura 29. Inicializar Entorno de Trabajo con WinIDE



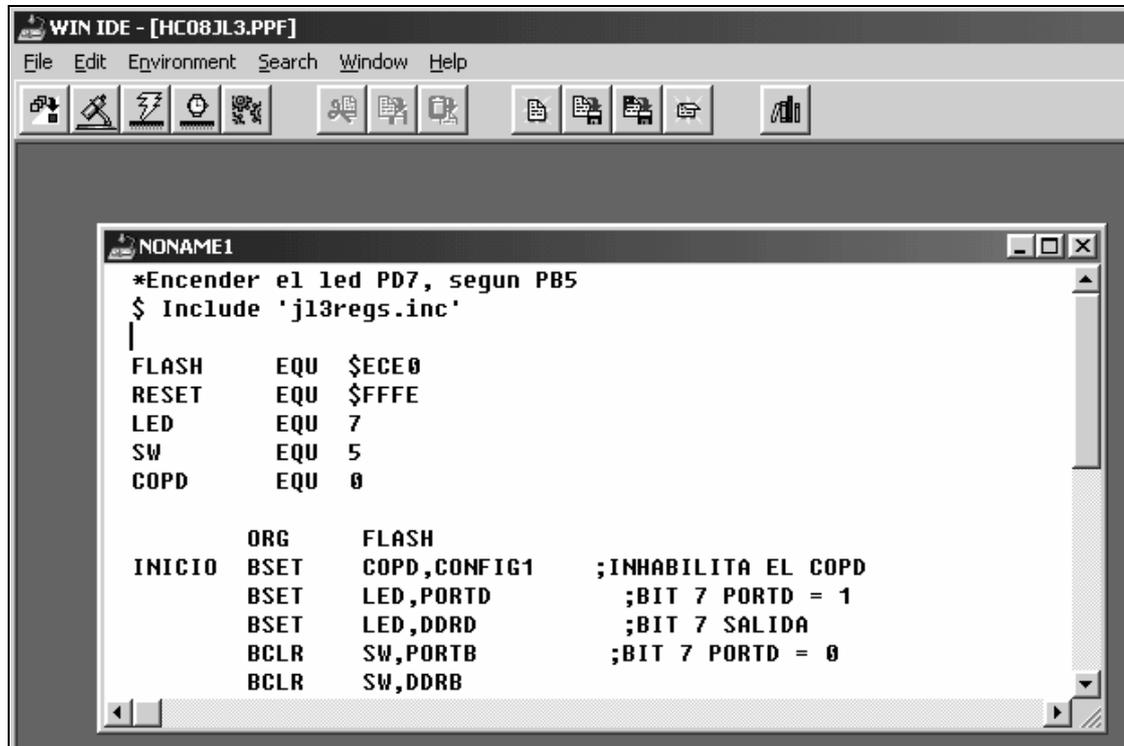
En el menu principal, seleccionar la opción **"File"** y posteriormente **"New File"**

Figura 30. Crear un nuevo archivo ASM con WinIDE



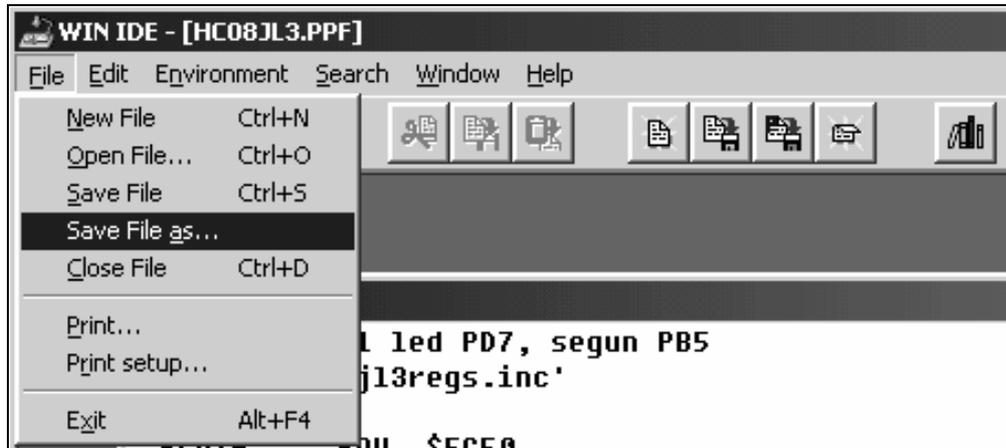
Aparece entonces una ventana con el título **"Sin Título 1"** o **"NONAME1"**; en esta ventana se escribe todo el código correspondiente al programa que deseamos escribir.

Figura 31. Escritura de un programa ASM con WinIDE



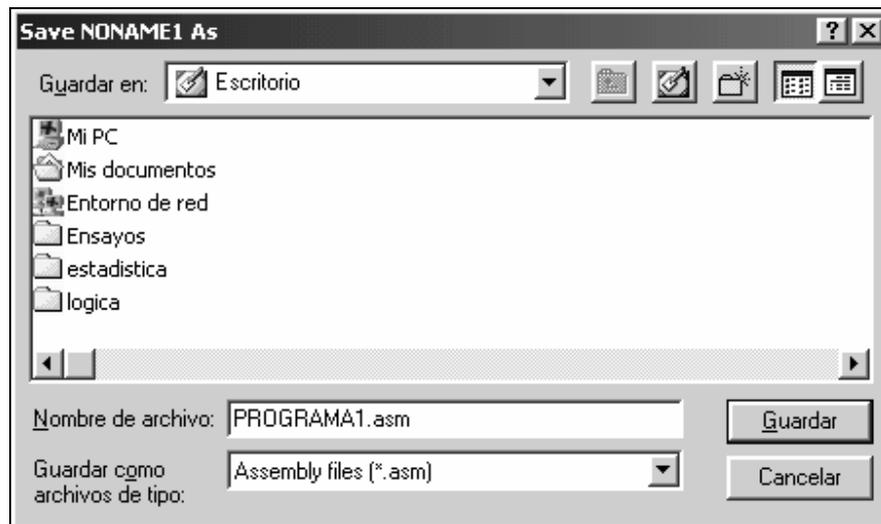
Después de que se escribe el programa, se debe almacenar en el disco duro, bajo un nombre específico. Para ello en el menú principal, seleccionamos la opción **"File"** seguido de la opción **"Save File As"**.

Figura 32. Guardar programa ASM en el disco duro



Para el caso del ejemplo, le podemos dar el nombre "Programa1.asm" y listo. Ha quedado almacenado el programa deseado en el PC.

Figura 33. Formulario para almacenamiento en el Disco



Hasta aquí, simplemente se ha escrito un programa en assembler para realizar una tarea específica. La segunda etapa consiste en verificar o compilar el programa, con el fin de verificar si hay errores de sintaxis, errores por falta de definición de variables o etiquetas, etc. Para compilar el programa, en la barra de herramientas se presenta el siguiente botón :



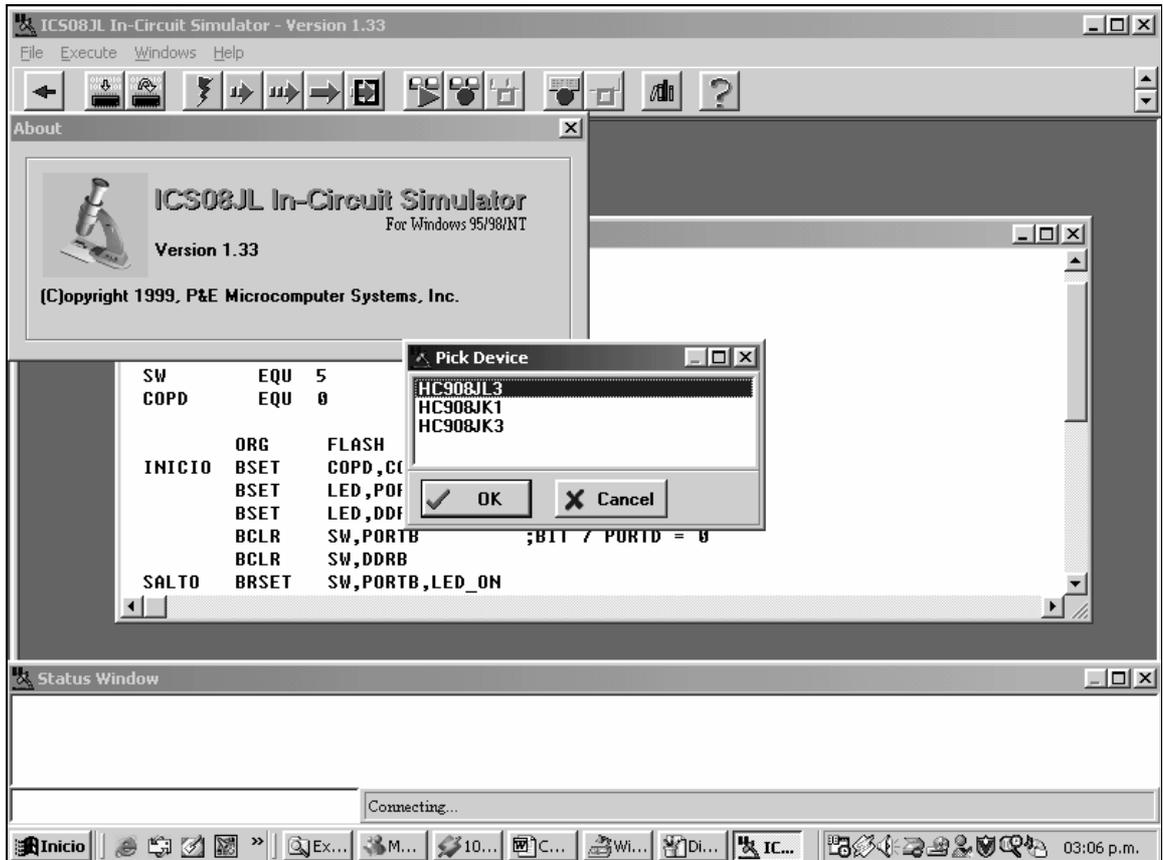
Assemble/Compile File : Al pulsar éste botón, se realiza la compilación o verificación de errores de sintaxis y/o programación realizada. Si en algún caso se presentara un error, la herramienta software indicará la línea en donde éste se encuentra sombreándola de color rojo.

Adicionalmente, el software permite no solamente compilar sino también programar, emular, simular, etc. A continuación se hace explicación de cada una de éstas opciones complementarias y de gran importancia a la hora de realizar cualquier programa en lenguaje Assembler.



In-Circuit Simulator : Esta opción permite realizar una simulación paso a paso con el circuito conectado, es decir, que a medida que se va depurando paso a paso cada línea desde el Software, esta misma línea es ejecutada en el circuito, lo cual brinda una gran ayuda al diseñador en el momento de desarrollar y verificar cualquier aplicación. Al pulsar éste botón, el siguiente formulario:

Figura 34. Entorno In-Circuit Simulator



Este formulario pregunta, ¿Qué tipo de Microcontrolador se va a utilizar para la simulación en circuito?, se debe tener en cuenta que para realizar este tipo de simulación el computador debe estar conectado el programador y el microcontrolador en su respectivo socket conectado desde el programador al circuito perteneciente a la aplicación desarrollada.

Para el caso, se seleccionó la opción HC908JK3; posteriormente aparece el formulario que se ilustra a continuación:

Figura 35. Cuadro de control de Comunicación y programación

Attempting to contact target and pass security...

Target Hardware Type

- Class I - ICS Board with processor installed. Possible emulation cable connection. (Power controlled via serial DTR line)
- Class II - ICS Board without processor. Connected to target via MON08 Cable. (MON08 - No power control)
- Class III - Custom Board (no ICS) with MON08 serial port circuitry built in. (NODTR - No power control)
- Class IV - Custom Board (no ICS) with MON08 serial port circuitry and additional auto-reset circuit built in.

PC Serial Port Configuration

Port: Baud: The serial port is open.

Target MCU Security bytes

- Attempt ALL Known security codes in order
- Attempt FF-FF-FF-FF-FF-FF-FF-FF (Blank Device)
- Attempt FF-FF-FF-FF-FF-FF-FF-FF {From security.ini} (Recent)
- Attempt FF-FF-FF-FF-FF-FF-FF-FF {From security.ini}
- User:
- IGNORE security failure and enter monitor mode.

Status: **Invalid Response or No Response to last attempt to contact target.**

- 0. ICS Hardware loopback detected: **N**
- 1. Device echoed some security bytes: **N (Port?)**
- 2. Device echoed all security bytes: **N**
- 3. Device signaled monitor mode with a break: **N**
- 4. Device entered monitor mode: **N**
- 5. Reset was Power-On Reset:
- 6. ROM is accessible (un-secured):

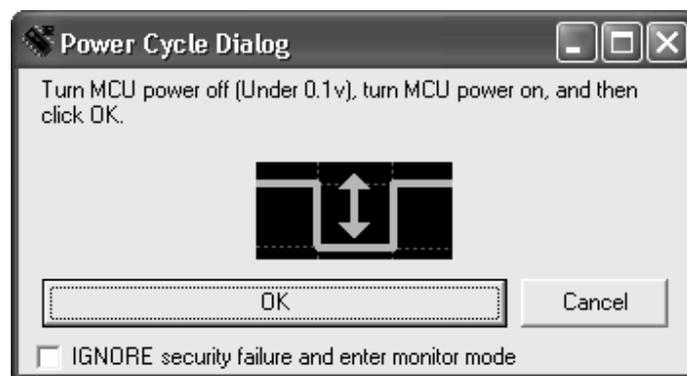
Show this dialog before attempting to contact the target 68HC08 board

En este formulario se configuran las características para realizar la comunicación con el programador, para el caso de los Microcontroladores JK1/JK3/JL3 se escoge la opción “**Class III**”, entablado comunicación por el puerto deseado (COM1 o COM2) y finalmente a una velocidad de comunicación de 9600 Baudios.

Cuando se han configurado los parámetros y se va a entablar una comunicación con el circuito programador aparece una ventana de verificación en la comunicación; esta ventana se ilustra a continuación, la cual muestra que debe hacerse una apertura y cierre del interruptor presente en el programador para corroborar

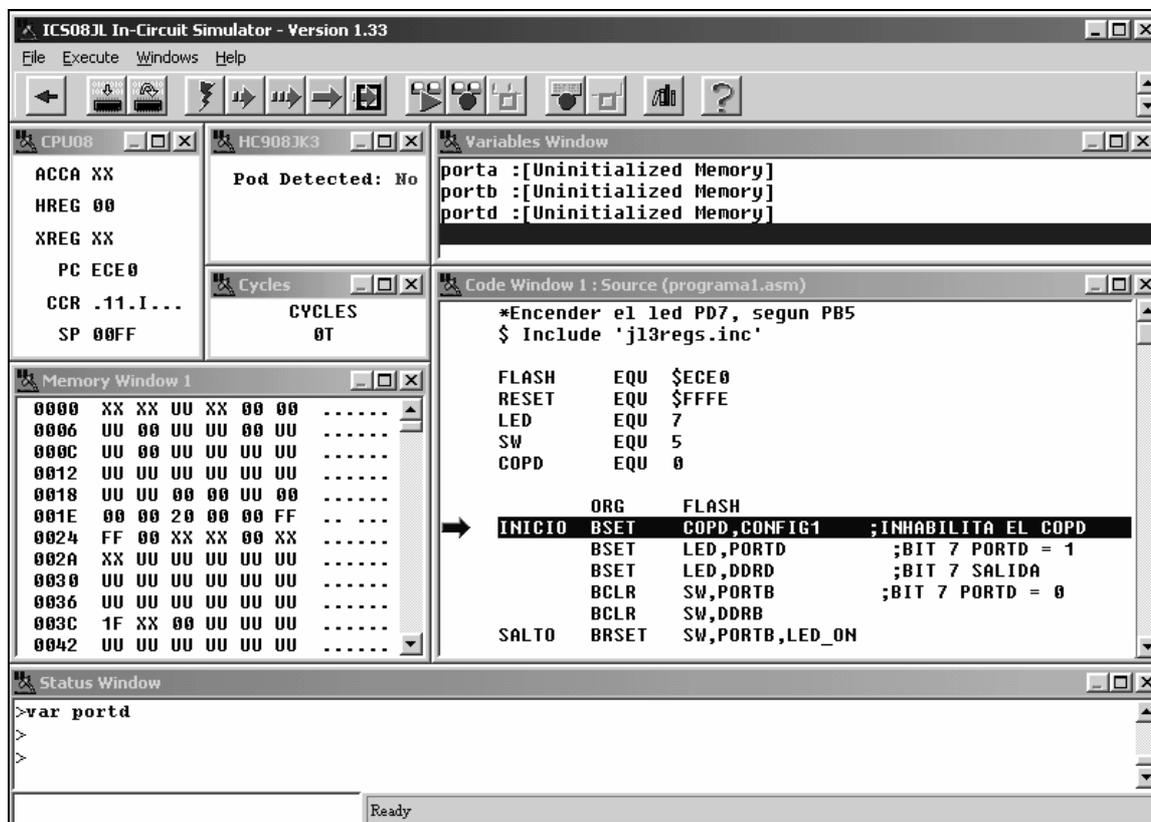
que el proceso de comunicación se encuentra en optimas condiciones.

Figura 36. Ventana de verificación de comunicación con Tarjeta



Establecidos estos parámetros, estamos listos para comenzar a simular la aplicación desarrollada. El formulario que aparece a continuación corresponde al formulario que permite realizar la simulación del programa de acuerdo con los requerimientos del usuario, permitiendo visualizar diferentes regiones del microcontrolador como son: Variables, Memoria, Ciclos, Registros Especiales y Código de programa, sombreando la línea a ejecutar

Figura 37. Entorno de Simulación



Dentro de los botones más utilizados en la barra de herramientas se pueden mencionar

- ❖  **Reset** : Reinicializa la ejecución del Programa
- ❖  **Step** : Ejecuta una instrucción del programa cada vez que se presione
- ❖  **Múltiple Step** : Ejecuta múltiples instrucciones con solo una pulsación
- ❖  **Go** : Ejecuta el programa como si estuviera en tiempo real, sin visualizar la línea que se está ejecutando

❖  **Stop** : Detiene la ejecución del programa

Para el caso de las alternativas ofrecidas por WinIDE, se presentan casos muy semejantes el descrito para el caso de "In-Circuit Simulator", en donde hay necesidad de configurar la comunicación con el programador (En el caso de escoger la opción "Programmer" ó "In-Circuit Debugger") o solamente indicar ejecución paso a paso, multiples pasos o ejecución completa en el caso de seleccionar "Simulation" .

 **Programmer** : Opción que permite almacenar el programa en el Microcontrolador, en otras palabras, dar la orden para quemar el microcontrolador con el programa deseado.

 **In-Circuit Debugger** : Esta función realiza una tarea muy semejante a la mencionada en "In-Circuit Simulator", con la diferencia de que el software ejecuta no paso a paso a la velocidad del usuario, sino a una velocidad animada, es decir, cada instrucción se ejecuta en un lapso de tiempo fijo y a la vez rápido, emulando la función de ejecución en tiempo real.

 **Simulation** : La opción de simulación, como su nombre lo indica, permite simular el programa paso a paso, o ejecutado con puntos de ruptura para agilizar su evaluación con la diferencia de que ésta vez no se tiene conectado el circuito en el momento de simular, a diferencia de cómo ocurría en "In-Circuit Simulator"

Otras opciones presentes en la barra de herramientas y que son de gran ayuda son las siguientes:



Open File : Abre un archivo ASM almacenado previamente



Save File : Guarda el programa codificado en la ventana de código como un archivo con extensión ASM (Assembler)



Save Project : Guarda un proyecto



Close File : Cierra un archivo ASM abierto



Cut : Borra de la ventana de código el texto seleccionado



Copy : Copia en la ventana de código un texto previamente seleccionado guardándolo en el portapapeles



Paste : Pega el contenido del portapapeles en la ventana de código, correspondiente a un texto previamente seleccionado