

Sistema de desarrollo para microcontrolador **Motorola** **MC68HC908JK3/JK1**

GUSTAVO A. GALEANO
info@propuestadinamica.com

JUAN ANDRÉS CASTAÑO WELGOS
welgos@telesat.com.co

*Sin invertir grandes cantidades de tiempo y dinero en borradores, programadores o software, las tecnologías actuales de microcontroladores de 8 y 16 bits de **MOTOROLA**, permiten que los usuarios tengan rápidamente un laboratorio donde puedan desarrollar sus aplicaciones en un menor tiempo y con una mayor confiabilidad.*

En el pasado, las tecnologías existentes en el mercado obligaban a pequeños o medianos industriales a invertir mucho dinero en equipos adicionales al microcontrolador, como son los borradores ultravioleta, programadores, y en algunos casos, en un buen software que les permitiera editar, ensamblar, simular, realizar conexiones y programarlo.

Los nuevos microcontroladores y las herramientas de desarrollo disponibles para los usuarios de 8 bits de **MOTOROLA**, permiten todas estas facilidades mediante un programa interno de

fábrica, que permite iniciar proyectos de forma muy sencilla y rápida y sin ningún costo adicional.

Breve descripción del microcontrolador

El MC68HC908JK3 es uno de los miembros de la familia 08 de Motorola, caracterizados por su bajo costo y alto desempeño. Todos los miembros de esta familia utilizan la unidad de procesamiento denominada CPU08 y están disponibles en una gran variedad de presentaciones (20, 28 y 40 pines) y en diversos tamaños de memoria de pro-

grama (1.5K, 4K y 32K). Dentro de las principales características de estos microcontroladores es que son de memoria Flash (borrable y programable eléctricamente) además de contar con 10 canales de conversión de analógico a digital (ADC). En la tabla 1 se pueden observar las características más importantes de la CPU08 y en la tabla 2 las características del microcontrolador MC68HC908JK3, (por sencillez JK3).

Descripción de pines del JK3

En la figura 1 se puede observar la distribución de pines del microcontrolador

Características de la CPU08
Un modelo de programación muy completo, con registro de 16 bits.
Set de instrucciones muy amplio y varios modos de direccionamiento.
Registro de 16 bits y <i>stack pointer</i> manipulable por el usuario.
Instrucciones de transferencia de Memoria a Memoria
Instrucciones de Multiplicación rápida de 8x8
Instrucciones de División rápida de 16/8
Instrucciones de BCD (<i>Binary Coded Decimal</i>)
Fácil soporte de lenguajes de alto nivel como el C.

Tabla 1

Características del MCU JK3:
CPU 08 de 8 bits
Operación interna a 8 MHz
Rango de operación desde 3V
LVI: protección de bajo voltaje.
4 K Bytes para memoria de programa (FLASH)
128 bytes de memoria RAM
10 canales de análogo a digital de 8 bits
15 Entradas/Salidas
2 temporizadores (<i>timers</i>) de 16 bits (Salida por comparación y/o entrada de captura)
Software 100% compatible con el de la familia 05
Versiones HC (cristal o resonador) y HRC (Resistor y capacitor)
Modos de bajo consumo (STOP y WAIT)
COP (<i>Watchdog</i>): perro guardián
8 fuentes de Interrupciones (totalmente vectorizadas)
<i>Pull-downs</i> programables por puerto de entrada

Tabla 2

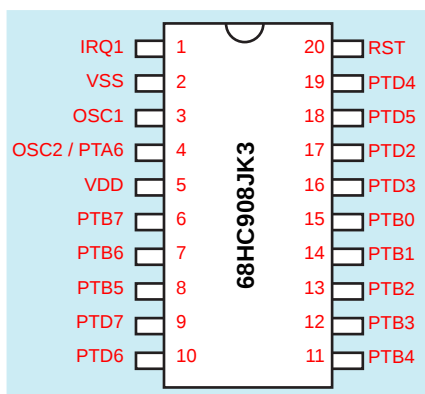


Figura 1. Diagrama de pines del integrado en su presentación de 20 pines

JK3 y en la tabla 3 la función que desempeña cada uno de ellos.

Arquitectura interna del JK3 y modelo de programación de la CPU08

En la figura 2 se puede apreciar un diagrama interno del microcontrolador y en la figura 3 se muestran los registros internos de la CPU08 mediante los cuales se procesan las instrucciones, los operandos y los resultados. La función

básica de cada uno de ellos se describe a continuación.

Acumulador (A): Es un registro de 8 bits de propósito general usado en las operaciones aritméticas y lógicas.

Registro índice (H:X): Es un registro de 16 bits utilizado como apuntador en el modo de direccionamiento indexado.

Puntero de pila (SP): Es un registro de 16 bits que contiene la dirección de la posición disponible en el *stack*.

Contador de programa (PC): Es un registro de 16 bits que contiene la dirección de la siguiente instrucción u operando a ser procesado.

Registro de banderas (CCR): Es un registro de 8 bits que contiene el bit de enmascarado general de interrupciones y 5 banderas de estado, las cuales indican ciertas condiciones originadas por la instrucción previamente ejecutada.

Mapa de memoria

En la figura 4 se muestra el mapa de memoria completo del JK3. Allí se encuentran las direcciones de los registros internos, la posición que ocupa la RAM y la parte asignada al programa del usuario. De igual manera, se observa la zona en la cual se encuentra almacenado el programa de fábrica (ROM MONITOR) que nos permite realizar simulación, *debug* y programación del microcontrolador. En la zona alta de memoria se encuentran los vectores de interrupción y en ellos se encuentra la dirección (2 bytes) con la cual se carga el contador de programa (PC) cuando se presenta alguna de las interrupciones.

Sistema de desarrollo

El proyecto que realizaremos consiste en un sistema de desarrollo que soporta los microcontroladores **JK3** y el **MC68HC908JK1 (JK1)** de la familia **08** de **MOTOROLA**. Este proyecto se puede adquirir bajo la referencia **K-218** de **CEKIT**. En la figura 5 se observa la conexión básica con la computadora.

El sistema consta de un software y de una tarjeta en la cual se inserta el microcontrolador con los elementos básicos para su funcionamiento; la guía de montaje de esta tarjeta se puede apreciar en la figura 6. Posee entrada para alimentación de DC, conversores de nivel TTL a RS-232 que permiten conexión serial a una computadora para simulación y programación, circuito oscilador basado en cristal y LED rojo indicador de alimentación.

El jumper J1 selecciona el modo de funcionamiento. En la posición «PROG» el sistema arranca en modo SIMULACION-PROGRAMACION, permitiendo mediante el software instalado, realizar simulación y posterior programación del chip. J1 en la posición «APP» permite que el sistema arranque en modo aplicación; es decir, ejecuta el programa grabado por el usuario.

SW1 es un suiche de dos polos y dos posiciones, que permite remover la alimentación completa del sistema cuando el software del PC así lo solicite para validar el código de seguridad interno del

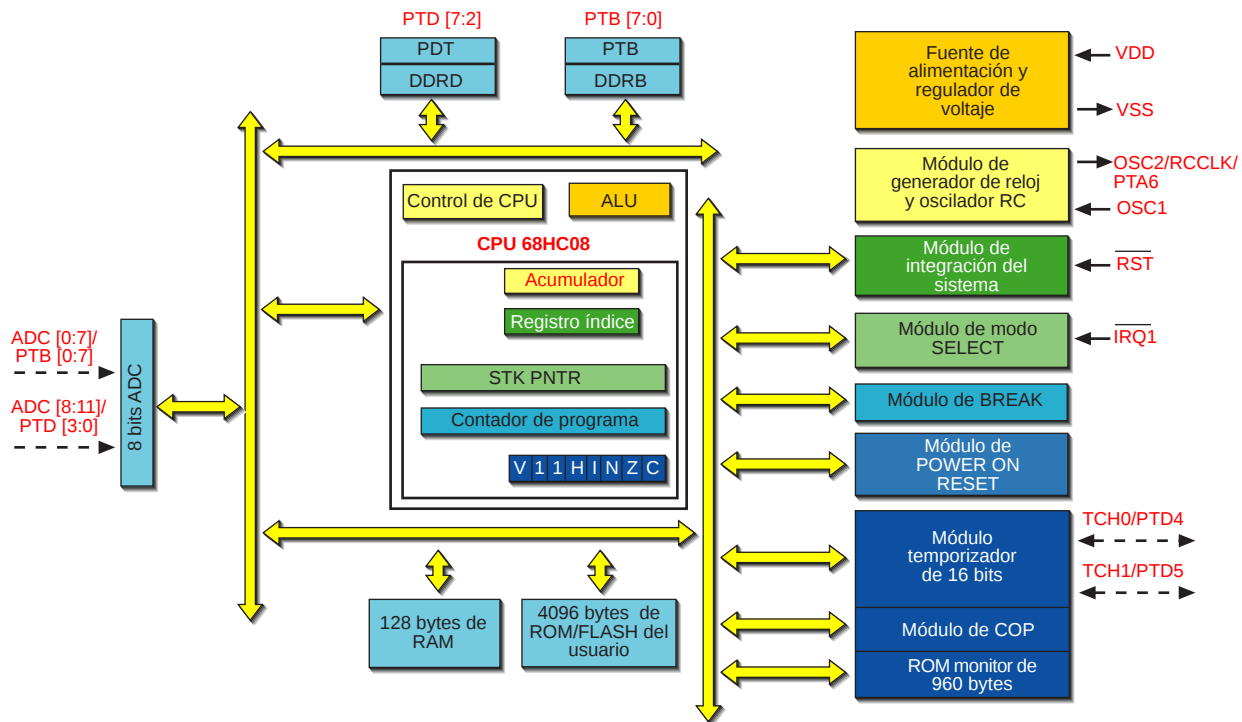


Figura 2. Diagrama de bloques internos del microcontrolador Motorola 68HC908JK3

micro. El pulsador RST está conectado a la señal de reset del micro permitiendo el control por el usuario de esa función.

J3, J4 y J5 corresponden a tres de los pines del microcontrolador (PTB3, PTB2 y PTB1); para poder entrar al modo simulación-programación, ellos deben tener colocado el jumper respectivo. Si el usuario utiliza estos pines en su aplicación, deberá remover los jumpers cuando arranque el sistema en modo aplicación (J1 en posición APP).

La tarjeta posee un conector en línea macho de 3 pines (CN2), el cual permite la conexión directa al PC (para simulación-programación) y mediante los conectores en línea CN3, CN4 y CN5 se tiene acceso a todos los pines del chip, con los cuales podemos conectar el sistema de desarrollo a nuestra aplicación.

Adicionalmente, el sistema cuenta con un LED verde conectado a uno de los pines del microcontrolador (PD7) y un jumper que conmuta entre +VDD y tierra conectado al pin PB5; mediante estos dos elementos ilustraremos el manejo de salidas y entradas.

Nombre del pin	Descripción	Entrada/salida	Nivel de voltaje
VDD	Alimentación positiva	Entrada	5v ó 3v
VSS	Tierra	Salida	0v
RST	Reset activo en bajo con resistencia interna de pull-up	Entrada	VDD
IRQ1	• Interrupción externa • Posee resistencia interna de pull - up • Pin usado para seleccionar el modo de arranque (programación o aplicación)	Entrada	VDD
OSC1	Entrada del oscilador	Entrada	Análogo
OSC2	Salida del oscilador	Salida	Análogo
PTB (0:7)	• Puerto I/O de 8 bits • 8 entradas análogas, ADC(0:7)	Entrada/salida	VDD
PTD (2:7)	• Puerto I/O de 6 bits • PTD (3:2) 2 canales de ADC, ADC(8:9) • PTD (4:5) 2 pines de temporizador • PTD (6:7) salida en colector abierto de máximo 25 mA	Entrada/salida	VDD
		Entrada/salida	Análogo
		Entrada/salida	VDD
		Entrada/salida	VDD
		Entrada/salida	VDD

Tabla 3. Descripción de las funciones de cada uno de los pines

Ensamble

Con la guía de ensamble de la figura 6 y la lista de materiales, realice el montaje de los componentes en el impreso de referencia K-218. Tenga precaución de soldar primero los componentes de bajo nivel, tales como resistencias y condensadores; después puede continuar con los transistores y demás elementos. Al finalizar, remueva los excesos del fundente de la soldadura con algún limpiador electrónico de contactos o con thinner.

Realice el ensamble del cable de interfaz con la computadora, haciendo los puentes como se ilustra en la figura 7 (en el conector DB9: el pin 4 unido con el pin 6 y el pin 7 unido con el pin 8). En la figura 9 se muestra el aspecto físico de la tarjeta.

Pruebas iniciales de la tarjeta

Para hacer las primeras pruebas a la tarjeta, puede seguir el siguiente procedimiento.

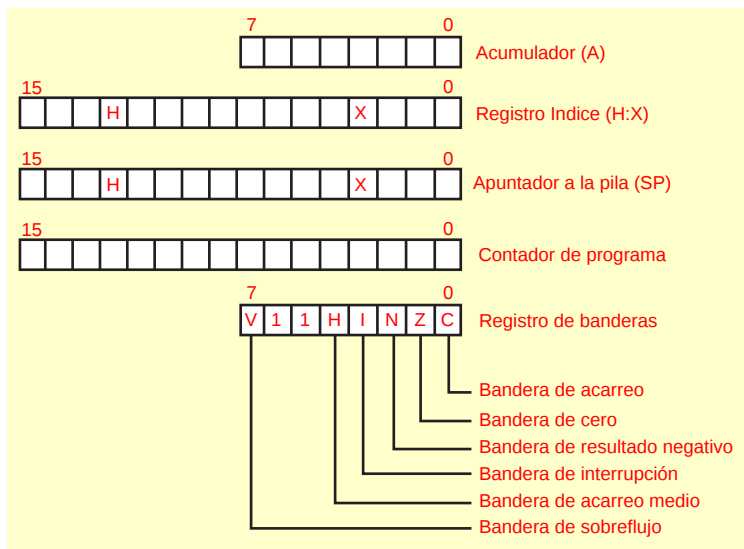


Figura 3. Estructura de los registros internos y ubicación de las banderas

1. Sin el microcontrolador JK3, alimentar el circuito con un adaptador de 12VCD con terminación redonda. Tenga precaución con la polaridad: el terminal externo debe ser el positivo (+) y el terminal interno, el negativo (-) o conexión de tierra.
2. Realice la medida de voltaje entre **TP4** (GND) y **TP3** (5V), debe ser alrededor de 5 VDC; de lo contrario revise el montaje del circuito. Realice la medida de voltaje entre **TP4** (GND) y **TP2** (IRQ); debe ser alrededor de 8 VDC; de lo contrario revise el montaje del circuito.
3. Remueva la alimentación de la tarjeta y coloque el micro **JK3** en la base.

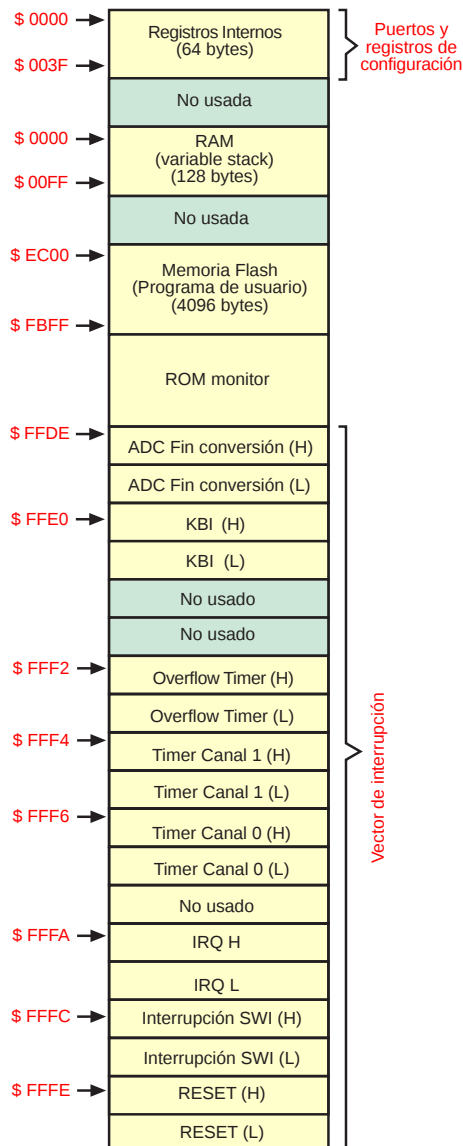


Figura 4. Mapa de la memoria interna del microcontrolador

Lista de materiales	
Resistencias 1/4 w, 5%	
R1	10 MΩ
R2	10 Ω
R3, R4 R5, R6	10 KΩ
R13, R15, R16	10 KΩ
R7	100 KΩ
R8, R10	470KΩ
R9, R11	3.3 KΩ
R12, R14	100 Ω
Condensadores	
C1, C8	22uF / 50V electrolíticos
C2, C3, C6, C7	104 / 50V cerámicos (0.1μF)
C4, C5	22pF / 50V cerámicos
Semiconductores	
D1	Diodo LED Rojo 3mm
D2	Diodo LED Verde 3mm
D3	Diodo rectificador 1N4148
Q1	Transistor NPN 2N2222
Q2	Transistor PNP 2N2907
Circuitos Integrados	
IC1	Microcontrolador MC68HC908JK3CP
IC2	Regulador 78L08 (ó LM7808)
IC3	Regulador 78L05 (ó LM7805)
IC4	Compuerta 74HC125 (ó 74LS125)
Conectores	
CN1	Conector hembra adaptador de corriente terminal redondo
CN2	Conector blanco 1x3 Vertical
CN3	(opcional) Conector blanco 1x4 Vertical
J1	Conector para jumper de 3 pines en línea.
J2, J3, J4, J5	Conector para jumper de 2 pines en línea.
Accesorios electromecánicos y otros	
SW1	Suiche 2 polos 2 posiciones pequeño
X1	Cristal 4.9152MHz (ó 5MHz)
RST	Pulsador reset negro 4 pines
TP1, TP2, TP3, TP4, TP5	Espadines para puntos de prueba
Carcaza plástica para DB9	
1.5 mt de cable de 3 ó más líneas	
Base maquinada 20 pines 0.3"	
5 Jumpers (o micropuentes)	
Circuito impreso K-218	

Software de desarrollo

El software mediante el cual se puede realizar edición, ensamblado, simulación, programación y *debug* es completamente gratuito y puede ser bajado de internet visitando la página www.pemicro.com (link: **Motorola M68HC908 Kits**). La documentación del microcontrolador y notas de aplicación para el manejo de todos los módulos internos del procesador, se pueden encontrar en www.mot-sps.com. Para manipular este sistema, tenga en cuenta el siguiente procedimiento.

1. Instale el software **ICSJL** de pemicro.
2. Para verificar el funcionamiento de la tarjeta con el software **ICSJL**, conecte el circuito al puerto serial de la computadora (DB9) y ejecute el programa **ICS08JLZ Development Kit > WinIDE Development Environment**. Allí se accede el segundo ícono de izquierda a derecha (**In-Circuit Simulator (EXE1)**), figura 10.
3. Elija el puerto por el cual fue conectada la tarjeta y 9600 Baud como velocidad de comunicación, figura 11.
4. Remover la selección del **Checkbox "Serial Port DTR controls target power"**, para que el software le indique cuándo encender y apagar el circuito (esto es necesario para validar el código de seguridad, en caso que el MCU esté programado previamente).
5. Presionar el botón **RETRY**, para intentar comunicación con el circuito, si no obtiene comunicación luego de varios intentos de encendido y apagado, revise el circuito cuidadosamente y repita el proceso.
6. Al obtener comunicación con el circuito, debe aparecer en la pantalla una nueva interface, en la que se tienen ventanas de puertos, zonas de memoria y programa en *assembler*, el cual puede ejecutarse paso por paso en la tarjeta, figura 12.
7. Para ilustrar la evaluación del hardware con la ayuda del software **ICSJLZ**, el circuito provee una salida (LED **D2**) y una entrada (Jumper **J2**). Coloque y remueva el Jumper **J2** de la tarjeta. Este cambio debe verse reflejado en la zona de puertos del software (esta es la forma como se pueden evaluar que las entradas de su proyecto funcionan adecuadamente). Para evaluar la salida acceda la línea

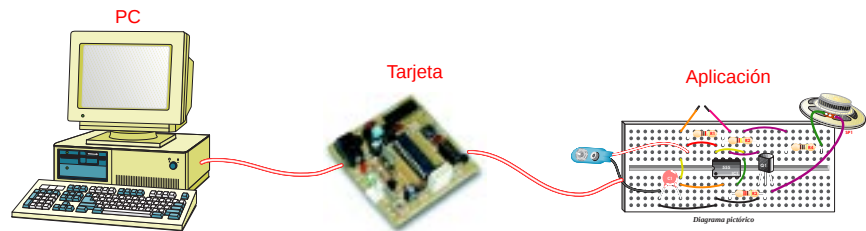
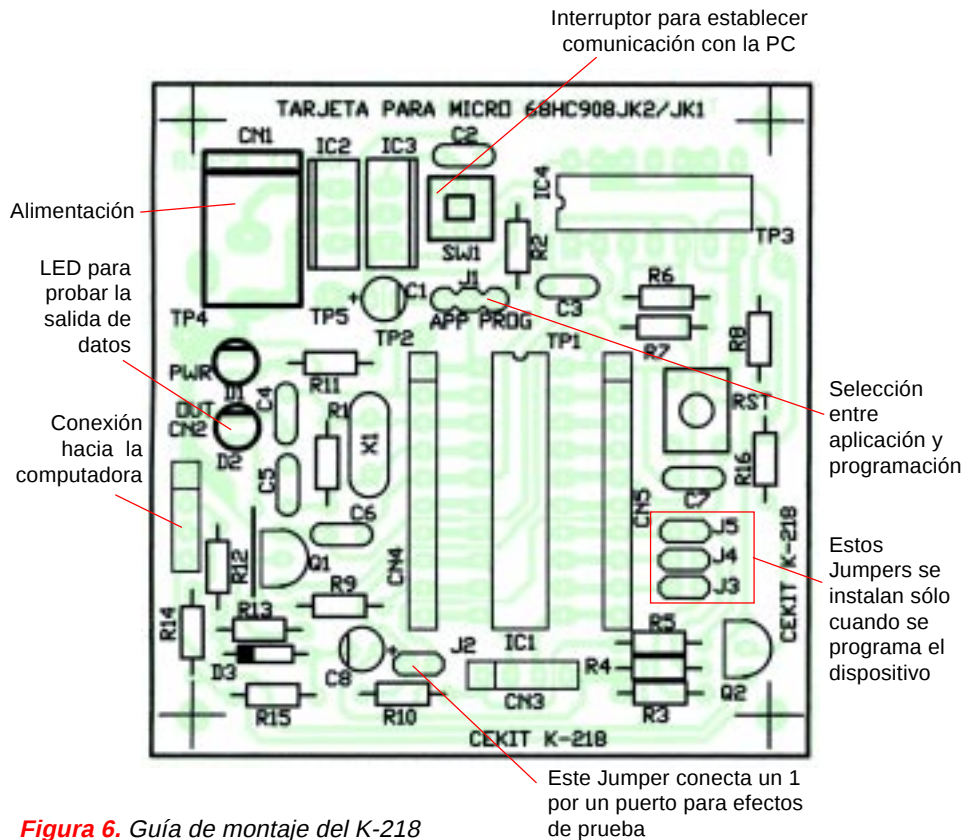


Figura 5. Conexión básica de la tarjeta con la computadora y con la aplicación



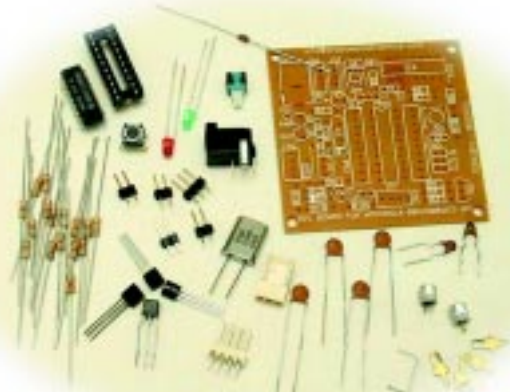


Figura 8. Componentes necesarios para el ensamble de la tarjeta. Todo este conjunto lo puede adquirir bajo la referencia K-218 de CEKIT S.A.



Figura 9. Aspecto final de la tarjeta ensamblada

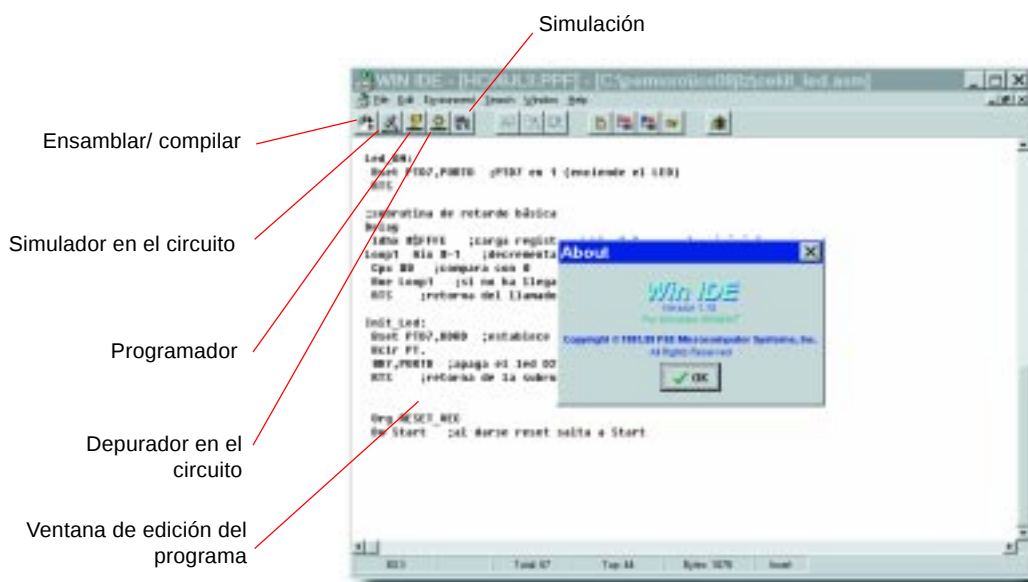


Figura 10. WINICE es el programa que permite la edición de los programas y además integrar en un solo entorno los procesos de ensamble o compilación, programación y depuración

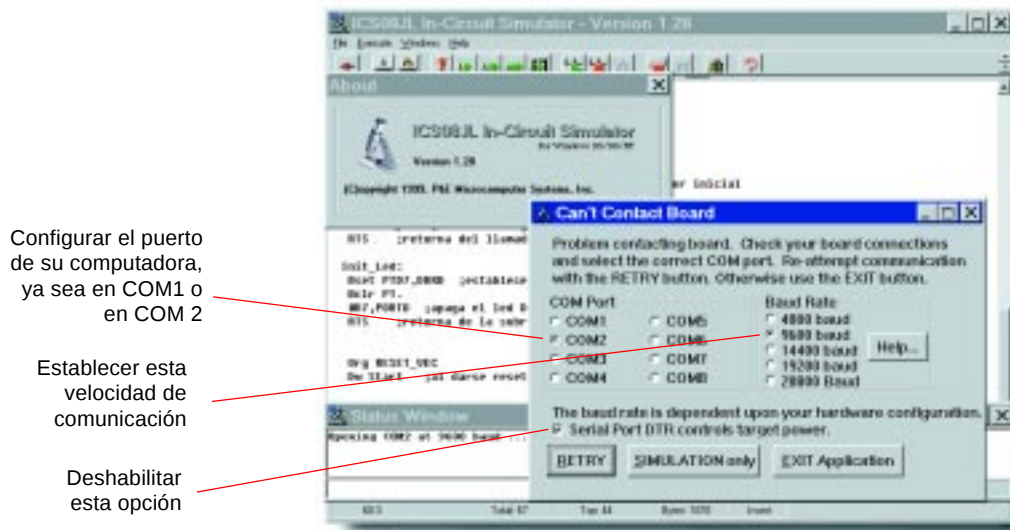


Figura 11. Para entrar al simulador se debe configurar el puerto serial a una velocidad de 9600 Baudios y deshabilitar la opción "Serial Port DTR controls target power"

cediendo el botón **SIMULATION only** (esta simulación es recomendable para zonas de programa que no tengan interacción directa con el hardware).

En la figura 13 se muestra una tabla resumen de todo el conjunto de instrucciones que puede utilizar para programar este tipo de dispositivos.

Programa ejemplo

El ejemplo que ilustramos a continuación consiste en un sencillo programa que varía la frecuencia de oscilación de encendido del LED D2, dependiendo del estado de la entrada J2 (jumper puesto, menor frecuencia de oscilación). Para la programación del MCU con el programa ejemplo, utilice los pasos que siguen a continuación.

1. Acceda el tercer icono del software WinIDE Development environment (**programmer** (EXE2)).
2. Cargue en memoria el algoritmo de programación del MCU JK3 (**908_JK3.08p**).
3. Especificar el archivo a programar en el icono 7 (de izquierda a derecha el icono con disquete).
4. Borrar el MCU presionando el icono 6 (de izquierda a derecha el icono con borrador sobre el chip).
5. Programar el chip presionando el icono 8 (de izquierda a derecha el icono de rayo sobre el chip).
6. Para ensayar el programa en la tarjeta, cambie el jumper J1 de la posición 2-3 (**PROG**) a la posición 1-2 (**APP**), desenergice y energice la tarjeta nuevamente para proporcionar un reset.
7. Para volver al modo simulación *in-circuit* y programación, cambie J1 de la posición 1-2 (**APP**) a la posición 2-3 (**PROG**), remueva la alimentación y colóquela nuevamente. 

Páginas donde se puede encontrar información adicional sobre estos microcontroladores:

<http://ebus.mot-sps.com/ProdCat/psp/>

www.digitaldna.com

www.mot-sps.com/products/index.html

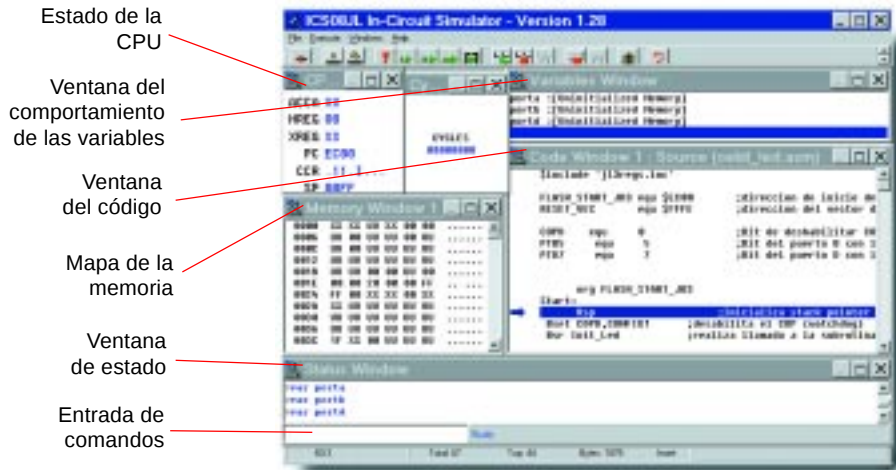


Figura 12. Pantallazo del simulador ICS08SL. Esta aplicación permite simular los programas en el circuito

```
$Include 'j13regs.inc' ; Archivo de definición de registros del 68HC908JK3.

FLASH_START_JK3 equ $EC00 ;dirección de inicio de la flash
RESET_VEC equ $FFFE ;dirección del vector de reset

;definición de bits
COPD equ 0 ;bit de CONFIG1 para habilitar/deshabilitar el watchdog
PD7 equ 7 ;bit de conexión a la salida (Led D2)
PB5 equ 5 ;bit de conexión de la entrada (jumper J2)

org FLASH_START_JK3
Start:
    Rsp ;inicializa stack pointer
    Bset COPD,CONFIG1 ;desabilita el COP (watchdog)
    Bsr Init_Led ;realiza llamado a la subrutina de estado inicial del
    led

LoopD:
    Bsr PB5,PORTB,Half1 ;verifica el estado de la entrada J2
    Lda #255T ;carga el AccA con el valor 255 decimal
    Bra Dly1
Half1: Lda #70T ;J2 colocado ¿ carga el AccA con un valor menor
    Bsr Delay ;invoca subrutina de retardo
    Deca ;decrementa el AccA
    Bne Dly1

    Bsr Led_ON ;enciende Led D2

    Bsr PB5,PORTB,Half2 ;verifica el estado de la entrada J2
    Lda #255T ;carga el AccA con el valor 255 decimal
    Bra Dly2
Half2: Lda #70T ;J2 colocado ¿ carga el AccA con un valor menor
    Bsr Delay ;invoca subrutina de retardo
    Deca ;decrementa el AccA
    Bne Dly2

    Bsr Led_OFF ;apaga Led D2
    Bra LoopD

Led_OFF:
    Bclr PD7,PORTD ;PD7 en 0
    RTS

Led_ON:
    Bset PD7,PORTD ;PD7 en 1
    RTS

;subrutina de retardo básica
Delay
    ldhx #$FFFE ;carga registro doble H:X con valor inicial
    Loop1
    Aix #-1 ;decrementa el registro H:X
    Cpx #0 ;compara con 0
    Bne Loop1 ;si no ha llegado continua el retardo
    RTS ;retorna del llamado

Init_Led:
    Bset PD7,DDRD ;establece el pin PD7 del MCU como salida
    Bclr PD7,PORTD ;apaga el led D2
    RTS ;retorna de la subrutina

Org RESET_VEC
Dw Start ;al darse reset salta a Start
```

Manipulación de bit			Salto			Lectura/escritura					Control			Memoria/registros									
DIR	DIR	REL	DIR	INH	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX				
MSB	0	1	2	3	4	5	6	9EE	7	8	9	A	B	C	D	9ED	E	9EE	F				
																				LSB			
0	5 3 3	BSET0 DIR	2 REL	4 NEG DIR	1 INH	1 INH	4 NEG IX1	5 NEG SP1	3 NEG IX	7 INH	7 RTI 1 INH	2 REL	2 SUB IMM	3 SUB DIR	4 SUB IX2	5 SUB SP2	3 SUB IX1	4 SUB SP1	2 SUB IX				
1	5 3 3	BCLR0 DIR	2 REL	4 CBEQ DIR	1 INH	1 INH	4 CBEQ IX1	5 CBEQ SP1	3 CBEQ IX	7 INH	4 RTS 1 INH	2 REL	2 CMP IMM	3 CMP DIR	4 CMP IX2	5 CMP SP2	3 CMP IX1	4 CMP SP1	2 CMP IX				
2	5 3 3	BSET1 DIR	2 REL	4 MUL DIR	1 INH	1 INH	4 MUL IX1	5 MUL SP1	3 MUL IX	7 INH		2 REL	2 SBC IMM	3 SBC DIR	4 SBC IX2	5 SBC SP2	3 SBC IX1	4 SBC SP1	2 SBC IX				
3	5 3 3	BCLR1 DIR	2 REL	4 COM DIR	1 INH	1 INH	4 COM IX1	5 COM SP1	3 COM IX	7 INH	9 SWI 1 INH	2 REL	2 CPX IMM	3 CPX DIR	4 CPX IX2	5 CPX SP2	3 CPX IX1	4 CPX SP1	2 CPX IX				
4	5 3 3	BSET2 DIR	2 REL	4 LSR DIR	1 INH	1 INH	4 LSR IX1	5 LSR SP1	3 LSR IX	7 INH	1 TAP 1 INH	2 REL	2 AND IMM	3 AND DIR	4 AND IX2	5 AND SP2	3 AND IX1	4 AND SP1	2 AND IX				
5	5 3 3	BCLR2 DIR	2 REL	4 LDX DIR	1 INH	1 INH	4 LDX IX1	5 LDX SP1	3 LDX IX	7 INH	1 TPA 1 INH	2 REL	2 BIT IMM	3 BIT DIR	4 BIT IX2	5 BIT SP2	3 BIT IX1	4 BIT SP1	2 BIT IX				
6	5 3 3	BSET3 DIR	2 REL	4 ROR DIR	1 INH	1 INH	4 ROR IX1	5 ROR SP1	3 ROR IX	7 INH	2 PULA 1 INH	2 REL	2 LDA IMM	3 LDA DIR	4 LDA IX2	5 LDA SP2	3 LDA IX1	4 LDA SP1	2 LDA IX				
7	5 3 3	BCLR3 DIR	2 REL	4 ASR DIR	1 INH	1 INH	4 ASR IX1	5 ASR SP1	3 ASR IX	7 INH	2 PSHA 1 INH	2 REL	2 A/S IMM	3 STA DIR	4 STA IX2	5 STA SP2	3 STA IX1	4 STA SP1	2 STA IX				
8	5 3 3	BSET4 DIR	2 REL	4 LSL DIR	1 INH	1 INH	4 LSL IX1	5 LSL SP1	3 LSL IX	7 INH	2 PULX 1 INH	2 REL	2 EOR IMM	3 EOR DIR	4 EOR IX2	5 EOR SP2	3 EOR IX1	4 EOR SP1	2 EOR IX				
9	5 3 3	BCLR4 DIR	2 REL	4 ROL DIR	1 INH	1 INH	4 ROL IX1	5 ROL SP1	3 ROL IX	7 INH	2 PSHX 1 INH	2 REL	2 ADC IMM	3 ADC DIR	4 ADC IX2	5 ADC SP2	3 ADC IX1	4 ADC SP1	2 ADC IX				
A	5 3 3	BSET5 DIR	2 REL	4 DEC DIR	1 INH	1 INH	4 DEC IX1	5 DEC SP1	3 DEC IX	7 INH	2 PULH 1 INH	2 REL	2 ORA IMM	3 ORA DIR	4 ORA IX2	5 ORA SP2	3 ORA IX1	4 ORA SP1	2 ORA IX				
B	5 3 3	BCLR5 DIR	2 REL	4 DBNZ DIR	1 INH	1 INH	4 DBNZ IX1	5 DBNZ SP1	3 DBNZ IX	7 INH	2 PSHH 1 INH	2 REL	2 ADD IMM	3 ADD DIR	4 ADD IX2	5 ADD SP2	3 ADD IX1	4 ADD SP1	2 ADD IX				
C	5 3 3	BSET6 DIR	2 REL	4 INCA DIR	1 INH	1 INH	4 INCA IX1	5 INCA SP1	3 INCA IX	7 INH	1 RSP 1 INH	2 REL	2 JMP IMM	3 JMP DIR	4 JMP IX2	5 JMP SP2	3 JMP IX1	4 JMP SP1	2 JMP IX				
D	5 3 3	BCLR6 DIR	2 REL	4 TSTA DIR	1 INH	1 INH	4 TSTA IX1	5 TSTA SP1	3 TSTA IX	7 INH	1 NOP 1 INH	2 REL	2 BSR IMM	3 JSR DIR	4 JSR IX2	5 JSR SP2	3 JSR IX1	4 JSR SP1	2 JSR IX				
E	5 3 3	BSET7 DIR	2 REL	4 MOV DIR	1 INH	1 INH	4 MOV IX1	5 MOV SP1	3 MOV IX	7 INH	1 STOP 1 INH	2 REL	2 LDX IMM	3 LDX DIR	4 LDX IX2	5 LDX SP2	3 LDX IX1	4 LDX SP1	2 LDX IX				
F	5 3 3	BCLR7 DIR	2 REL	4 CLR DIR	1 INH	1 INH	4 CLR IX1	5 CLR SP1	3 CLR IX	7 INH	1 WAIT 1 INH	2 REL	2 AUX IMM	3 STX DIR	4 STX IX2	5 STX SP2	3 STX IX1	4 STX SP1	2 STX IX				
DIR	Directo																						
INH	Inferente																						
INN	Inmediato																						
IX	Indirecto																						

Figura 13. En esta tabla se muestra el Set de instrucciones de este microcontrolador, incluyendo el número de ciclos de máquina que gasta cada una de ellas