

**“DISEÑO E IMPLEMENTACIÓN DE LA TARJETA DE DESARROLLO DEL
MC68HC908GP32 PARA EL LABORATORIO DE
MICROCONTROLADORES DE LA ESCUELA DE INGENIERÍA
ELECTRÓNICA ELÉCTRICA Y TELECOMUNICACIONES DE LA
UNIVERSIDAD INDUSTRIAL DE SANTANDER.**

JOSÉ GREGORIO ARDILA GUTIÉRREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA, 2005**

**“DISEÑO E IMPLEMENTACIÓN DE LA TARJETA DE DESARROLLO DEL
MC68HC908GP32 PARA EL LABORATORIO DE
MICROCONTROLADORES DE LA ESCUELA DE INGENIERÍA
ELECTRÓNICA ELÉCTRICA Y TELECOMUNICACIONES DE LA
UNIVERSIDAD INDUSTRIAL DE SANTANDER.**

JOSÉ GREGORIO ARDILA GUTIÉRREZ

Proyecto de Grado presentado como requisito parcial para optar al título de
Ingeniero Electrónico

DIRECTOR:

MAGISTER: JORGE HERNANDO RAMON

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA, 2005**

RESUMEN

TITULO:

DISEÑO E IMPLEMENTACIÓN DE LA TARJETA DE DESARROLLO DEL MC68HC908GP32 PARA EL LABORATORIO DE MICROCONTROLADORES DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA ELÉCTRICA Y TELECOMUNICACIONES DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER*.

AUTORES:

JOSÉ GREGORIO ARDILA GUTIÉRREZ**

PALABRAS CLAVES: Freescale, Metrowerks Codewarrior, , Microcontrolador, MC68HC908GP32 modo aplicación, modo monitor, Pantalla de cristal líquido.

DESCRIPCIÓN:

En este documento se explica cómo se implemento una tarjeta de aplicación y programación para el *microcontrolador MC68HC908GP32* de *Freescale* para uso en laboratorio de la materia microcontroladores de la escuela de Ingeniería Eléctrica Electrónica y telecomunicaciones de la UIS, también se explica como se implementaron los programas de uso de un teclado y *pantalla de cristal líquido*. Las etapas en las que se dividió el proyecto fueron análisis y selección de los elementos para la implementación, desarrollo de la tarjeta e implementación de los programas.

En el Capítulo 1 se describe el *microcontrolador* utilizado para la implementación del prototipo. Por lo tanto, se describe en forma general el *MC68HC908GP32*, los requerimientos electrónicos para el modo aplicación, modo programación y los requerimientos para el uso de los periféricos de la tarjeta implementada.

En el Capítulo 2 se describe el hardware del proyecto, el circuito de alimentación del proyecto, los circuitos de protección para el microcontrolador los circuitos osciladores implementados, el circuito implementado para la comunicación, los circuitos y sistemas mecánicos para el uso de los periféricos y las características del diseño del circuito implementado.

En el Capítulo 3 de software se describen las funciones y rutinas utilizadas para el uso de un teclado 4x4 y la pantalla de cristal liquido utilizando el compilador de *Metrowerks Codewarrior* versión 3.0 para microcontroladores

* Trabajo de grado

** Facultad de Ingenierías Físico mecánicas, Ingeniería Electrónica, Jorge Hernando Ramón

SUMMARY

TITLE:

DESIGN AND IMPLEMENTATION OF THE CARD OF DEVELOPMENT OF THE MC68HC908GP32 FOR THE LABORATORY DE MICROCONTROLADORES OF THE SCHOOL OF ELECTRIC ELECTRONIC ENGINEERING AND TELECOMMUNICATIONS OF THE INDUSTRIAL UNIVERSITY THE SANTANDER *

AUTHOR:

JOSÉ GREGORIO ARDILA GUTIÉRREZ**

KEYWORDS: *Freescale, Metrowerks Codewarrior, , Microcontroller, MC68HC908GP32 mode application, mode programming, liquid crystal display.*

DESCRIPTION:

In this document it is explained how it was implemented a card of application and programming for the microcontroller MC68HC908GP32 of Freescale for use in laboratory of the subject microcontroller of the school of Electronic Electrical Engineering and telecommunications of the UIS, also is explained as the programs of use of a keyboard and LCD were implemented. The stages in which the project was divided were analysis and selection of the elements for the implementation, development of the card and implementation of the programs.

In Chapter 1 the microcontroller used for the implementation of the prototype is described. Therefore, it is described in general form the electronic MC68HC908GP32, requirements for the mode application, mode programming and the requirements for the use of the peripheral of the implemented card.

In Chapter 2 the hardware of the project is described, the circuit of feeding of the project, the circuits of protection for the microcontroller the implemented oscillating circuits, the circuit implemented for the communication, the mechanical implementation and systems for the use of peripheral and the characteristics of the design of the implemented circuit.

In Chapter 3 of software to the functions and routines used for the use of a keyboard are described 4x4 and the crystal screen I eliminate using the compiler of Metrowerks Codewarrior version 3,0 for microcontroller.

* Degree work

** Physics and Mechanics Engineering College, Electronics Engineering, Jorge Hernando Ramón

CONTENIDO

INTRODUCCIÓN	12
1. MARCO TEÓRICO	13
1.1 MICROCONTROLADOR	13
1.2. EL MC68HC908GP32.....	14
1.3. PANTALLA LCD	17
1.4. TECLADO ALFANUMÉRICO	19
1.5. EAGLE 4.01.....	20
1.6 CODEWARRIOR 3.0	21
2. DISEÑO DE LA TARJETA	23
2.1 TARJETA DE DESARROLLO	23
2.2. MODO MONITOR.....	24
2.3. MODO APLICACIÓN.....	28
2.4. IMPLEMENTACIÓN DEL CIRCUITO DE RESET PARA LA TPM..	30
2.5. REQUERIMIENTOS ELÉCTRICOS PARA EL USO DE CONVERSORES ANALÓGICO DIGITALES.....	31
2.6. IMPLEMENTACIONES ELÉCTRICAS PARA LA COMUNICACIÓN	32
2.7. ESPECIFICACIONES ELÉCTRICAS DE LA TPM	33
2.9. PROTECCIONES ELÉCTRICAS DE LA TPM.....	34
2.11. TECLADO.....	34

3	SOFTWARE DESARROLLADO	39
3.1	MODO DE INICIALIZACIÓN DE LA LCD	39
3.3	MODO DE USO DEL TECLADO	41
	PRUEBAS REALIZADAS	43
	CONCLUSIONES.....	44
	BIBLIOGRAFÍA	46

TABLA DE FIGURAS

Figura 1, Diagrama de pines del mc68hc908gp32 40-pin pdip.....	16
Figura 2, LCD utilizada para implementación del proyecto.....	19
Figura 3, Teclado alfanumérico.....	20
Figura 4, Tarjeta desarrollada.....	24
Figura 5, Modo monitor forzado.....	27
Figura 6, TPM en modo aplicación.....	28
Figura 7, Esquema eléctrico de los osciladores.....	29
Figura 8, Circuito eléctrico de los osciladores.....	30
Figura 9, Circuito de reset.....	30
Figura 10, Circuito interno del reset en el mc68hc908gp32.....	31
Figura 11, Circuito implementado para comunicación.....	32
Figura 12, Cara inferior del circuito (bottom).....	36
Figura 13, Cara superior del circuito (top).....	36
Figura 14, Elementos que se encuentran dentro del circuito.....	37
Figura 15, Esquema eléctrico de la tarjeta.....	38
Figura 16, Diagrama de flujo de inicialización de la lcd.....	39
Figura 17, Diagrama de flujo del teclado.....	42

TABLA DE TABLAS

Tabla1 niveles para modo monitor	26
Tabla 2, distribución de pines de la lcd	40
Tabla 3, instrucciones de la lcd.....	40

TABLA DE ANEXOS

Anexo A, Manual de uso de la tarjeta

Anexo B, Como generar archivos Gerber desde Eagle

Anexo C, Manual basico de uso de Codewarrior

Anexo D, Algoritmos implementados con Codewarrior

INTRODUCCIÓN

Este es un proyecto desarrollado para la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones.

Se diseñó y construyó una tarjeta de desarrollo para el microcontrolador MC68HC908GP32 con fines académicos, esta tarjeta permite al usuario utilizar todas las potencialidades del microcontrolador, MC68HC908GP32 de Freescale, con el propósito de aplicar los conocimientos adquiridos en la materia microcontroladores, en la implementación de proyectos de carácter académico en el laboratorio de esta materia.

La tarjeta se implementó con las siguientes características:

Es un sistema de desarrollo reprogramable de propósito general en un montaje que permite mantenimiento de las partes electrónicas, basado en el microcontrolador MC68HC908GP32, este sistema es capaz de adquirir y procesar señales analógicas, digitales y el uso de “timers”.

Posee un sistema para la comunicación por puerto serie entre una computadora y el microcontrolador implementado dentro de la tarjeta.

Posee un mecanismo que permite la conexión y utilización de un teclado de membrana numérico y una LCD.

En el capítulo uno se encuentra toda la información que hace referencia al proyecto y las herramientas que se utilizaron para desarrollarlo incluyendo un marco teórico de los implementos de la tarjeta.

En el capítulo dos se encuentra la descripción del hardware diseñado y las razones de la selección de sus elementos.

En el capítulo tres se encuentra la descripción del software de apoyo que se desarrolló para facilitar el uso de la tarjeta.

1. MARCO TEÓRICO

En este capítulo se hace referencia a las herramientas y las características de los elementos utilizados en el desarrollo del proyecto.

1.1 MICROCONTROLADOR

Los microcontroladores son circuitos integrados que incluyen en su interior las tres unidades funcionales de un ordenador: CPU, Memoria y Unidades de E/S, se trata de un sistema de control completo en un solo circuito integrado. Además de dicha integración, su característica principal es su alto nivel de especialización, aunque sus prestaciones son limitadas.

Un microcontrolador posee un microprocesador optimizado para ser utilizado para controlar procesos o variables que se pueden medir electrónicamente. Los microcontroladores se usan en una amplia gama de aplicaciones entre las cuales están aplicaciones domésticas y a nivel industrial lo que hace que para el ingeniero electrónico sea muy importante el conocimiento y manejo de los mismos.

En la actualidad los microcontroladores están presentes en nuestro trabajo, en nuestra casa y en nuestra vida en general. Se pueden encontrar controlando el funcionamiento de ratones y teclados en los computadores, en los teléfonos, en los hornos microondas y en los televisores de nuestro hogar. Los microcontroladores son una tecnología que tiende a evolucionar y ampliar su campo de aplicación.

La casi totalidad de los automatismos y electrodomésticos que hoy conocemos, están siendo gobernados por microcontroladores.

Las extensas áreas de aplicación de los microcontroladores, que pueden considerarse ilimitadas, exigen al ingeniero electrónico conocimiento en la

programación y aplicación de los mismos, para implementar soluciones tecnológicas en las áreas que lo necesiten.

Es por esta razón que el conocimiento en esta área se hace de vital importancia en el desarrollo académico de un ingeniero, para aprender a manejar y aplicar los microcontroladores en dispositivos reales. Por tal razón, además del desarrollo teórico, el ingeniero debe poseer un desarrollo integral el cual incluye la práctica y programación con estos circuitos, la cual se realiza a través de una tarjeta de desarrollo para una practica con el microcontrolador MC68HC908GP32.

1.2. EL MC68HC908GP32

El microcontrolador MC68HC908GP32 hace parte de una familia de microcontroladores de 8 bits de muy alto rendimiento, su arquitectura es compatible con la familia 68HC05 y su programación es compatible con la familia de microcontroladores de 16 bits 68HC12, lo que hace fácil y rápida la migración de diseños de una familia a otra.

Tiene la opción de utilizar hasta 8-MHz de frecuencia de bus interno y seguridad en la programación de la memoria "FLASH". Puede trabajar a 5V o 3V dependiendo de la configuración del mismo aunque en el caso de 3V la frecuencia del bus se reduce a la mitad.

Entre sus características más importantes tenemos:

- 32256 bytes de memoria flash.
- Este tipo de tecnología no requiere de voltajes de altos de programación,
- Permite 10000 ciclos de borrado y programado.
- Presenta características de seguridad que no permiten examinar el código programado.

- 512 bytes de memoria RAM
- CPU08 de alto desempeño, código compatible con la CPU05
- Módulo periférico de operación apropiada, COP, “watchdog timer” y monitor de reloj
- Puerto serial de comunicación síncrona, SPI
- Puerto serial de comunicación asíncrona, SCI
- Dos “timers” de canales duales de 16 bits configurables en modo captura, modo comparación y modo PWM.
- Ocho canales analógico/digital de 8 bits.
- Modulo de base de tiempo por interrupciones en tiempo real.
- Registros de entrada y salida direccionables en el mapa de memoria.
- 29 líneas bidireccionales de entrada/salida, con manejo de cargas hasta de 15mA.
- Voltajes de operación de 2.5V a 5.5V.
- Frecuencia de operación interna de 8Mhz a 5 voltios y 4Mhz a 3 voltios
- RESET bidireccional con pull-up interno, IRQ con pull-up interno PORTA pull-up seleccionable.
- 17 fuentes de interrupciones vectorizadas.
- Reset por direccionamiento ilegal de memoria y por instrucción ilegal
- Dos modos de bajo consumo de potencia de WAIT y STOP.
- Dos opciones de reloj: un PLL basado en un oscilador de 32 Khz o un reloj externo
- Interrupción por bajo voltaje LVI interna con dos “trip points” seleccionables
- 16 modos de direccionamiento.

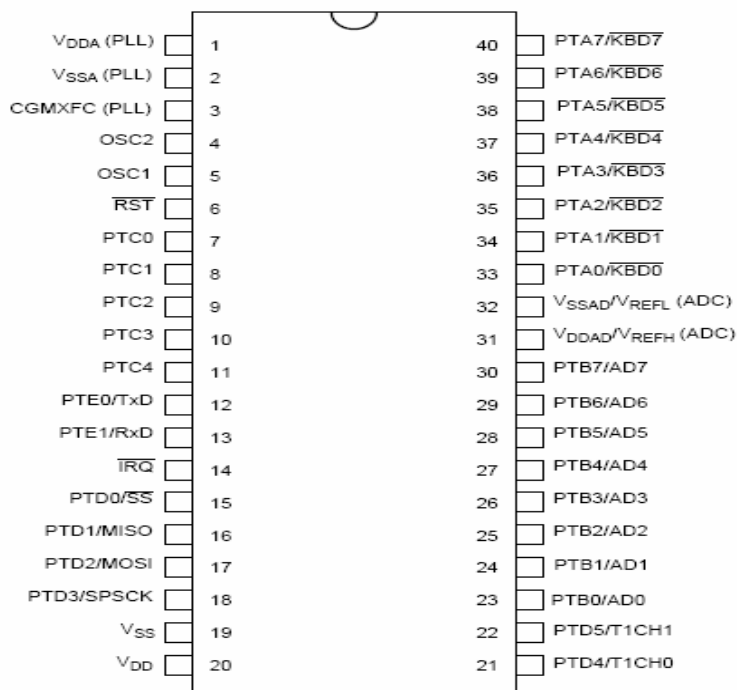
El HCMOS MC68HC908GP32¹ es un microcontrolador de 8 bits con sofisticados periféricos. Su diseño estático, tecnología CISC, y “high-density complementary metal oxide semiconductor” (HCMOS) permite la fabricación

¹ Véase technical data MC68HC908GP32 pag 32 en [http:// www.freescale.com](http://www.freescale.com)

de miembros de la serie GP que operan a frecuencias de 2Mhz en DC, con bajo consumo de potencia. La tecnología HCMOS combina tamaño reducido y alta velocidad con baja potencia y alta inmunidad al ruido. El sistema de memoria del microcontrolador incluye 32 KBytes de memoria Flash y 512 de RAM.

El microcontrolador posee varios mecanismos para proteger el sistema contra errores: un watchdog (COP) protege al sistema contra fallas de software, un reloj monitor genera un reset del sistema en caso de que el reloj este perdido o atrasado, en la Figura 1 se puede apreciar el diagrama de pines del MC68HC908GP32 para un empaquetado tipo PDIP de 40 pines.

Figura 1, diagrama de pines del mc68hc908gp32 40-pin pdip



Fuente: Technical data MC68HC908GP32 Freescale

1.3. PANTALLA LCD

LCD son las siglas en inglés de "Pantalla de Cristal Líquido" ("Liquid Crystal Display"). su implementación y aplicación depende del número de líneas y dígitos por línea.

La LCD es un sistema eléctrico de presentación de datos formado por 2 capas conductoras en medio un material especial cristalino (cristal líquido) que tienen la capacidad de orientar la luz a su paso.

Cuando la corriente circula entre los electrodos transparentes con la forma a representar (por ejemplo, un segmento de un número) el material cristalino se reorienta alterando su transparencia.

Las pantallas LCD se encuentran en multitud de dispositivos industriales y de consumo: máquinas expendedoras, electrodomésticos, equipos de telecomunicaciones, computadoras, etc. Todos estos dispositivos utilizan pantallas las cuales cumplen un estandar. Cada LCD se compone de una pequeña placa integrada que consta de:

- La pantalla LCD.
- Un microcontrolador.
- Una pequeña memoria que contiene una tabla de caracteres.
- Una interfaz de contactos eléctricos, para conexión externa.
- Opcionalmente, una luz trasera para iluminar la pantalla.

El controlador simplifica el uso del LCD proporcionando una serie de funciones básicas que se invocan mediante la interfaz eléctrica, destacando:

- La escritura de caracteres en la pantalla.
- El desplazamiento horizontal de los caracteres de la pantalla (*scrolling*).

La memoria implementa un mapa de bits para cada carácter de un grupo de caracteres, es decir, cada octeto de esta memoria describe los "píxeles" que

deben iluminarse para representar un carácter en la pantalla. Generalmente, se pueden definir caracteres según lo requiera la aplicación, modificando el contenido de esta memoria. Así, es posible mostrar símbolos que no están originalmente contemplados en el juego de caracteres.

El interfaz de contactos eléctricos suele ser de tipo paralelo, donde varias señales eléctricas simultáneas indican la función que debe ejecutar el controlador junto con sus parámetros. Por tanto, se requiere conocer estos comandos para la utilización de la LCD.

Existen dos tipos de pantallas LCD : pantallas de texto y pantallas gráficas para este proyecto se utilizaron pantallas LCD de texto.

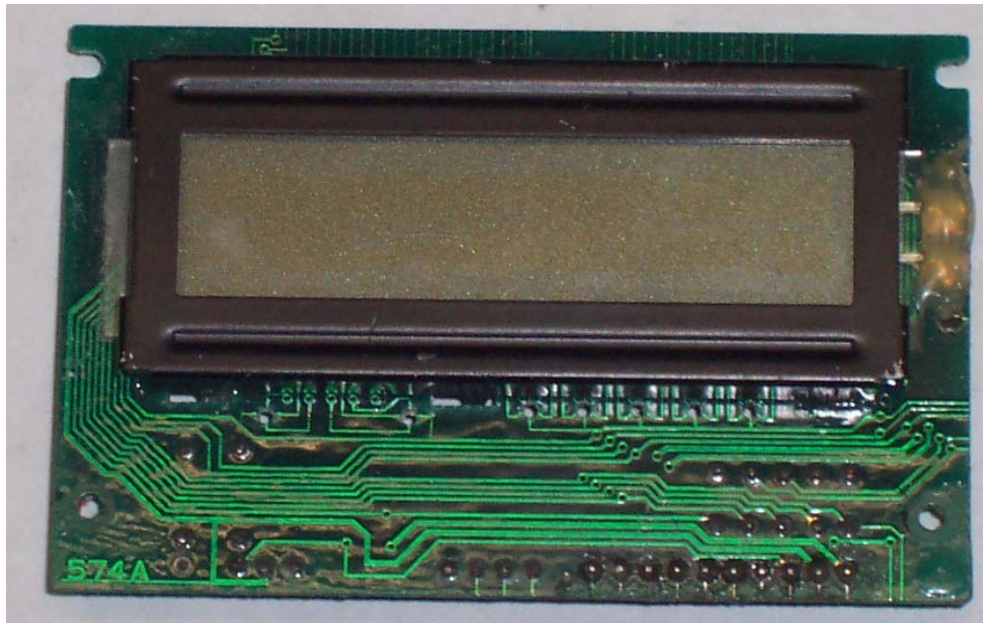
1.3.1. LCD de texto estas tarjetas solamente permiten visualizar cortos mensajes de texto, existen algunos modelos estandarizados en la industria, en función de su tamaño medido en número de líneas y columnas de texto, existen modelos de una, dos y cuatro filas únicamente. El número de columnas típico es de ocho, dieciséis, veinte y cuarenta caracteres.

El controlador Hitachi HD44780 se ha convertido en un estándar de industria cuyas especificaciones funcionales son imitadas por la mayoría de los fabricantes. Este controlador cuenta con los siguientes interfaces eléctricos:

- D0-D7: ocho señales eléctricas que componen un bus de datos.
- R/W: una señal que indica si se desea leer o escribir en la pantalla
- RS: una señal que indica si los datos presentes en D0-D7 corresponden a una instrucción u otros parámetros.
- E: una señal para activar o desactivar la pantalla.
- V0: señal eléctrica para determinar el contraste de la pantalla. Generalmente en el rango de cero a cinco voltios.
- Vss y Vdd: señales de alimentación. La señal Vss sirve para encender la luz trasera de la pantalla en algunos modelos.

Para el proyecto se utilizo una LCD de 2 filas y 20 columnas la LCD implementada se presenta en la figura 2.

Figura 2, LCD utilizada para implementación del proyecto



Fuente: autor del proyecto

1.4. TECLADO ALFANUMÉRICO

El teclado es un periférico utilizado para la introducción de órdenes y datos con el cual se emplea un reducido número de pines de entrada, existen distintas clases de teclado, dependiendo del tipo de puerto de comunicación número de teclas y circuito interno implementado.

Para este caso, el teclado es numérico con una matriz de 4x4. Este teclado. Consiste en un grupo de líneas conductoras distribuidas en dos grupos de 4, un grupo de líneas es vertical y las restantes horizontales, estas líneas se encuentran superpuestas unas sobre otras. En los puntos de intersección de las líneas se encuentran los números. Cuando se hace presión en estas

áreas del teclado, las líneas se cortocircuitan permitiendo al usuario saber cual de los puntos ha sido presionado a través de un algoritmo implementado², para el proyecto se utilizo un teclado de membrana de 4x4 como el que aparece en la figura 3.

Figura 3, teclado alfanumérico



Fuente: autor del proyecto

1.5. EAGLE 4.01

Para el diseño de la tarjeta se utilizo el “Software” EAGLE 4.01 version estudiantil³.

Este software presenta las siguientes características:

Es un producto de CadSoft 1988-2002, versión para Windows. Consta de varios editores como lo son de esquemas, trazos y autoenrutamiento, así como de un panel de control, para un mejor manejo de proyectos.

² Véase algoritmo implementado para manejo de un teclado numérico con Codewarrior

³ Para mas información de uso de Eagle vease Acevedo ,Jhon , tesis EVALUACIÓN PROCESOS DE DISEÑO CAD-CAM BIBLIOTECA UIS

Especificaciones Generales

- Area máxima de dibujo 64*64 pulgadas
- Resolución desde 0.1 micrones
- Edición de librerías simple
- Hasta 99 hojas por esquema
- Permite cambios en línea entre esquemático y trazado
- Generación automática de la board (no necesita crear netlist)
- Generación automática de señales de alimentación (para C.I.)
- Cambio entre enrutamiento manual y automático en cualquier momento
- Algoritmo de borre y reintente (*ripup&retry*)
- Soporte completo multicapas hasta 16 capas de señal
- Impresión y Procesamiento CAM⁴

1.6 CODEWARRIOR 3.0

Metrowerks⁵ y Freescale han dejado disponible al público una edición especial del entorno de desarrollo para microcontroladores HC08: CodeWarrior 3.0 la cual es libre y puede compilar hasta 4Kbytes de código C. Una de las ventajas importantes de esta versión es que adiciona el simulador de P&E Microcomputer Systems Inc., el cual cuenta con una máquina virtual que permite simular la CPU, periféricos e interrupciones de todos los microcontroladores HC08 actuales, lo que facilita el proceso de depuración de las aplicaciones desarrolladas en lenguaje C.

⁴ Vease anexo B COMO GENERAR ARCHIVOS GERBER DESDE EAGLE

⁵ Para mas información véase <http://www.metrowerks.com>

Además, el software Codewarrior permite programar en lenguaje de alto nivel, lo cual facilita la programación y desarrollo de algoritmos, pues presenta una fácil interpretación de cada uno de los programas desarrollados, lo cual es tendencia en los circuitos programables.

También incluye gran cantidad de librerías para diversas aplicaciones en distintas áreas de la ingeniería.

2. DISEÑO DE LA TARJETA

En este capítulo se procede a explicar las características que posee la tarjeta y su implementación.

2.1 TARJETA DE DESARROLLO

La **TPM** (Tarjeta de Programación del Microcontrolador) está diseñada con el estándar TTL, para el microcontrolador MC68HC908GP32 de Freescale de encapsulado PDIP-40⁶.

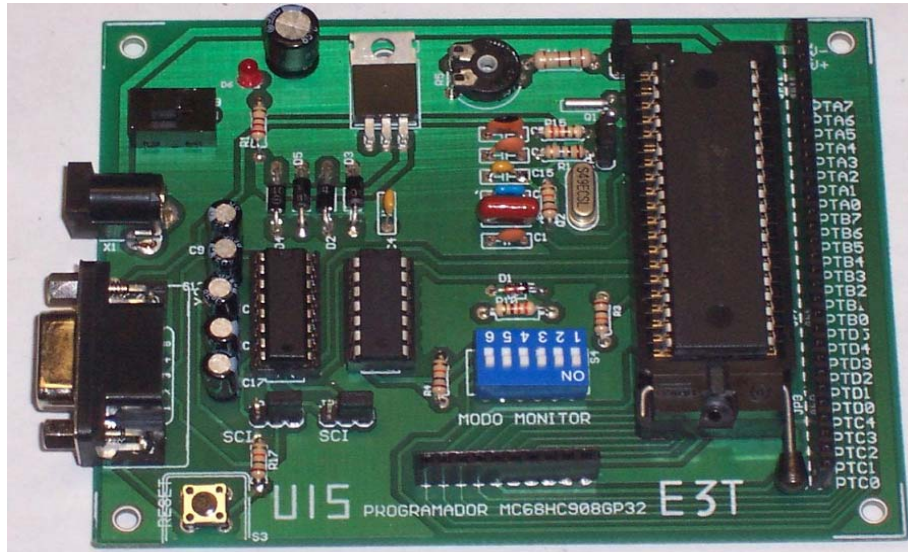
Las características de la tarjeta son:

- Entradas de 8 conversores analógico digitales
- Salidas de 2 TIMER
- Alimentación de 9V DC
- 27 pines de propósito general
- Salida de 5V
- Salida de 9V
- Puerto de comunicación con periféricos implementado con conector DB9
- Implementación del módulo precalador con un cristal de 32Khz.

La tarjeta desarrollada fue construida utilizando CAD-CAM las dimensiones de esta son de 12 x 9 cm esta se puede apreciar en la figura 4

⁶ Ver anexo A: MANUAL DE USO DE LA TARJETA MC68HC908GP32

Figura 4, tarjeta desarrollada



Fuente autor del proyecto

2.2. MODO MONITOR⁷

El modo monitor es una herramienta de programación que permite un completo escaneo del microcontrolador a través de una interfaz de un único cable con el “host” del computador, lo cual implica que utilizando el computador y un software compatible con el modo monitor se permite programar múltiples veces el microcontrolador.

Existen dos maneras de entrar en modo monitor. Entrada monitor estándar y la entrada monitor forzada. Todos los dispositivos 68HC08 soportan la entrada monitor estándar. Usando este método, el dispositivo puede estar gobernado a cualquier frecuencia dentro del rango permitido en modo usuario. Sin embargo, para facilitar la comunicación entre el dispositivo y el ‘host’, el reloj debe seleccionarse para lograr un “baud rate” determinado. En la mayoría de dispositivos 68HC08, el “baud rate” es igual a la frecuencia de

⁷ véase <http://www.silica.com>. guía didáctica del 68HC908

trabajo interna dividida por 256. Una frecuencia de reloj externa típica es de 4.9152 MHz que produce un “baud rate” común de 9600 bits por segundo. En el caso de monitor forzada se requiere que la memoria este en blanco.

2.2.1. Modo monitor estándar Para activar el 68HC908GP32 en modo monitor estándar, hay que conectar el pin PTC0 a VDD⁸ y el pin PTC1 a tierra. El pin PTC3 especifica si la frecuencia interna es la frecuencia del oscilador dividida por 2 o dividida por 4, para el caso de la TPM se divide en 2 por tanto esta conectada a tierra.

Para la entrada en monitor estándar, se debe conectar un oscilador externo a OSC1 y VTST⁹ aplicado al pin IRQ, también activar el pin de comunicación.

Para el 68HC908GP32, este pin es el PTA0. Para esto se necesita usar una resistencia de ‘pull-up’ en el pin de comunicación. Esto mantendrá el estado inactivo de la línea de comunicación en el nivel apropiado.

El modo monitor usa la comunicación bidireccional. half duplex y de no retorno a cero. Esto requiere multiplexar la línea de recepción/transmisión del ‘host’. Que se puede implementar fácilmente con un buffer de tres estados (MC74HC125).

Otro componente, es el convertidor de niveles de tensión entre el protocolo RS-232 y los niveles TTL.

Cuando el nivel del pin PTA7 pasa a 0, con las anteriores especificaciones ver tabla 1, el código de seguridad entra al dispositivo de forma serie sobre el pin PTA0.

⁸ VDD es el voltaje requerido para funcionamiento del microcontrolador en la TPM

⁹ VTST es un voltaje requerido para programación el cual es VDD+2.5

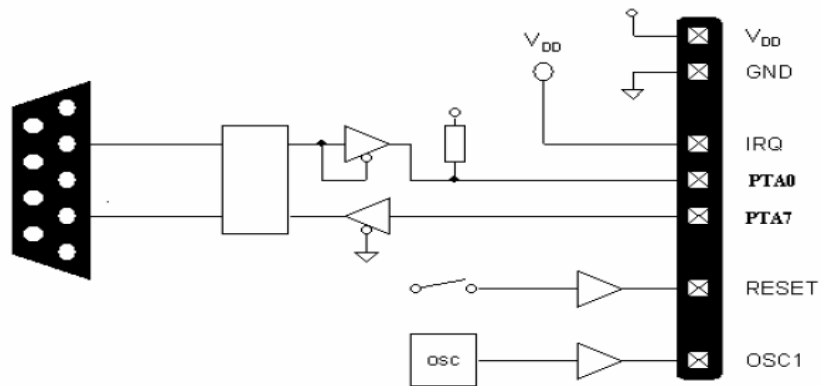
tabla1 niveles para modo monitor

IRQ	RESET	\$FFFE/ \$FFFF	PLL	PTC0	PTC1	PTC3	External Clock ⁽¹⁾	CGMOUT	Bus Frequency	COP	For Serial Communication		
											PTA0	PTA7	Baud Rate ^{(2) (3)}
5V	5V 7.5V	X	OFF	1	0	0	4.9152 MHz	4.9152 MHz	2.4576 MHz	Disabled	1	0	9600
											X	1	DNA

Fuente: Technical data MC68HC908GP32 Freescale

2.2.2. Modo monitor forzado La mayoría de los 68HC908 soportan la entrada de modo monitor forzado. Con este método de programación, se entra automáticamente en el modo monitor cuando parte de la memoria está en blanco, indicado por un vector de reset en blanco. Esto elimina la necesidad de aplicar un voltaje en IRQ y la necesidad de satisfacer condiciones del puerto para la entrada en modo monitor. Hay que tener presente que se puede necesitar un pin adicional para hacer que el monitor trabaje como se quiera. En el 68HC908GP32, se puede activar o se puede desviar la inicialización del PLL poniendo el pin IRQ a VSS o VDD respectivamente. También, se debe poner el pin PTA7 a VSS para indicar que el código de seguridad se transmitirá sobre el puerto de comunicación serie (pin PTA0). Al usar la entrada en modo monitor forzada, las únicas líneas que deben ir a la placa de circuito impreso son la línea de comunicación y la referencia a tierra como se muestra en la figura 5.

Figura 5, Modo monitor forzado



Fuente: Technical data MC68HC908GP32 Freescale

Cuando el vector de reset está en blanco, se puede entrar en modo monitor aplicando VTST o VDD a IRQ. Si se aplica VTST, se puede requerir otra configuración del puerto para entrar en modo monitor. Si VDD se aplica a IRQ, no se requiere ninguna otra configuración del puerto.

La frecuencia interna es la frecuencia externa (oscilador) dividida entre 4 y la velocidad de comunicación es la frecuencia interna dividida entre 256. Esta puede ser gobernada por un cristal de baja frecuencia, típicamente de 32 KHz y un PLL interno. VSS se puede aplicar a IRQ cuando el vector de reset está en blanco, en caso que el PLL está automáticamente acoplado. El PLL se pondrá a la frecuencia interna de 2.4576 MHz cuando el reloj externo (cristal o oscilador) sea de 32.768 Khz. En este caso, la velocidad de comunicación es de 9600 bits por segundo y no se requiere ninguna otra configuración del puerto.

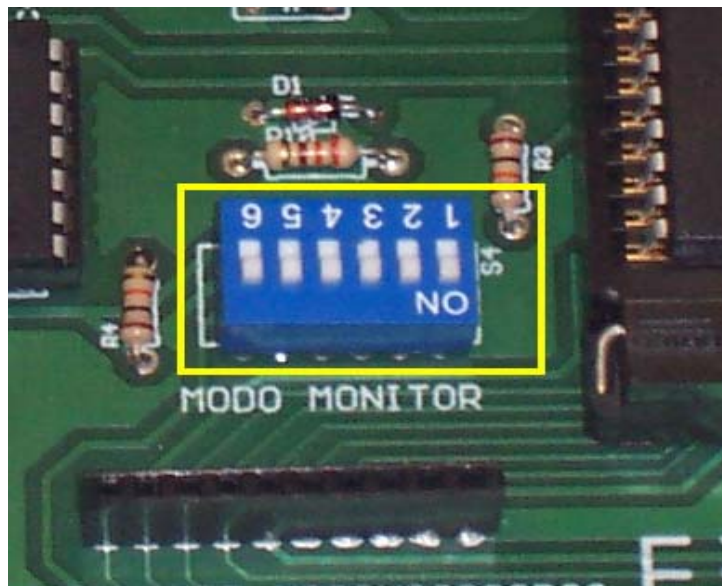
La habilidad de utilizar un reloj con un cristal de baja frecuencia y después aumentar la frecuencia hasta un nivel para producir un baud rate' común, sólo está disponible en una memoria en blanco. También, si el PLL está

acoplado. Entonces los componentes externos de filtrado necesitan estar conectados al pin CGMXFC.

2.3. MODO APLICACIÓN

Para el caso del modo aplicación o usuario, no se tienen requerimientos de entrada de voltajes aparte de los de alimentación y todos los pines de propósito general están disponibles para su uso, para esto es necesario poner los switches de modo monitor en off en la TPM como se muestra en la figura 6.

Figura 6, TPM en modo aplicación



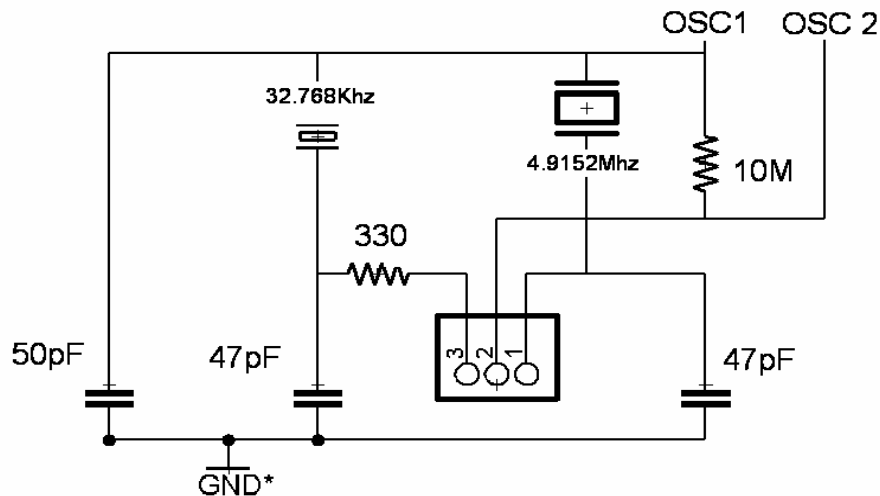
Fuente: autor del proyecto

El valor del cristal externo esta dentro de un rango amplio de valores, pero con el propósito de explorar todas las aplicaciones que presenta el MC68HC908GP32 se implemento dentro de la **TPM** el uso del PLL el cual

permite al usuario manipular la frecuencia de trabajo del microcontrolador por medio de programación.

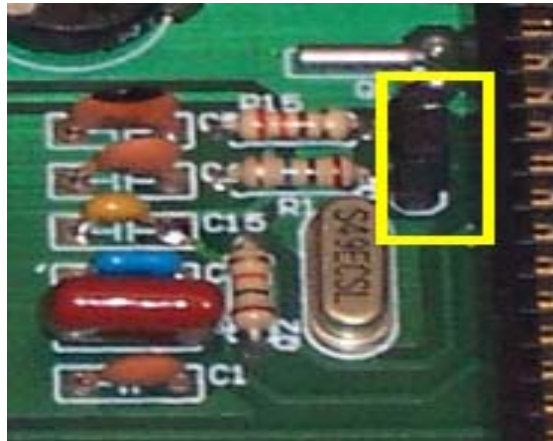
Es por tales razones que la tarjeta tiene implementados 2 cristales uno que se utiliza para modo monitor estándar de frecuencia 4.9152Mhz y otro de frecuencia 32KHZ el esquema eléctrico del circuito implementado puede apreciarse en la figura 7 y el circuito en la figura 8, esta implementación permite al usuario modificar la frecuencia de trabajo por medio de el PLL y un filtro implementado en el pin 3 del microcontrolador que corresponde a CGMXFS .

Figura 7, esquema eléctrico de los osciladores



Fuente: autor del proyecto

Figura 8, circuito eléctrico de los osciladores

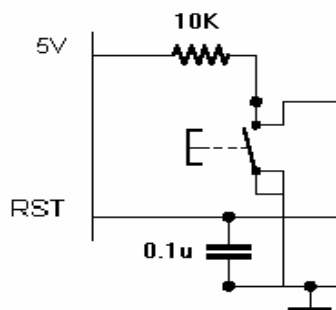


Fuente: autor del proyecto

2.4. IMPLEMENTACIÓN DEL CIRCUITO DE RESET PARA LA TPM

Para la implementación del circuito de reset de la TPM se utilizó un pulsador de cuatro pines, el cual permite la transición de un voltaje de 5V a 0V al pin de reset del microcontrolador el circuito implementado se presenta en la Figura 9 .

Figura 9, circuito de reset

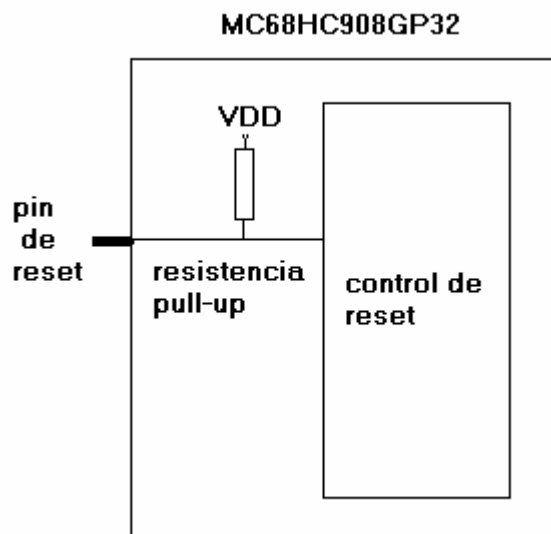


fuentes: autor del proyecto

La razón de este circuito es alimentar la entrada de reset del MC68HC908GP32 con 5V en su estado normal y 0V para el estado de reset.

El microcontrolador posee una resistencia de pull-up que fija el voltaje a 5V para asegurar su funcionamiento en caso de no implementarse el circuito reset. El diagrama interno del reset del microcontrolador se presenta en la Figura 10.

Figura 10, circuito interno del reset en el MC68HC908GP32



Fuente: autor del proyecto

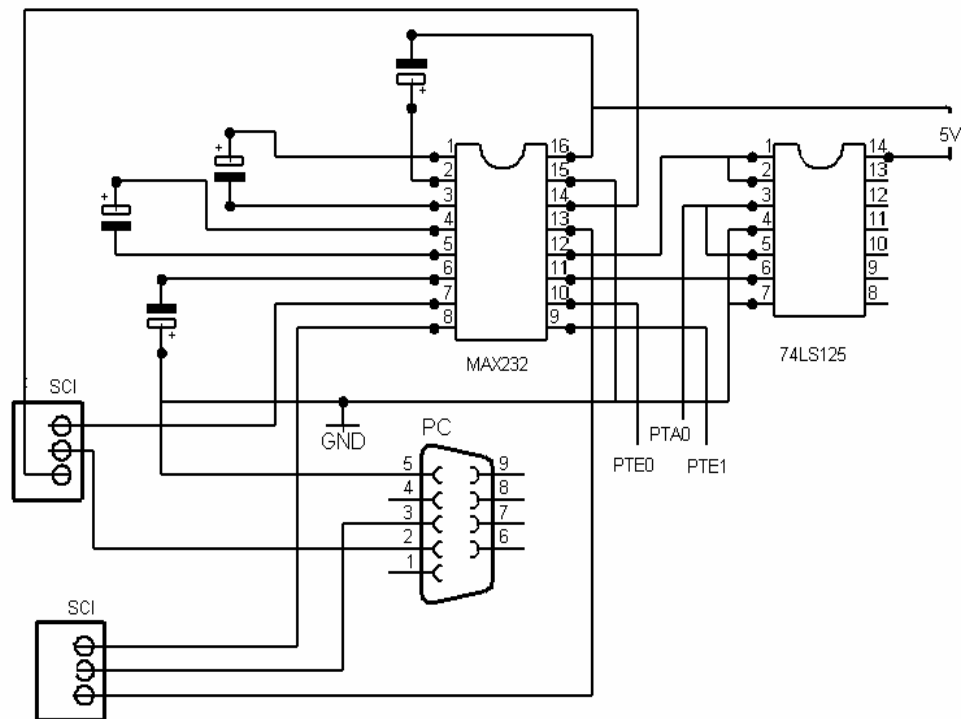
2.5. REQUERIMIENTOS ELÉCTRICOS PARA EL USO DE CONVERSORES ANALÓGICO DIGITALES

Una de los módulos que posee el microcontrolador es el de los conversores analógico digitales (CAD) para su correcto funcionamiento requiere dentro de la tarjeta un filtro, con el fin de disminuir el ruido generado por la alimentación, el cual consiste en un condensador entre la alimentación del conversor y tierra que corresponde al pin 31 y 32 VDDAD y VSSAD el valor de este es de 1uF.

2.6. IMPLEMENTACIONES ELÉCTRICAS PARA LA COMUNICACIÓN

La **TPM** tiene implementada su comunicación utilizando conector DB9, por el cual realiza comunicación según la aplicación, para comunicación utilizando el SPI (Modulo de Interfaz Serial con Periféricos) del microcontrolador utiliza dos líneas de comunicación implementadas para el puerto PTE, para programación utiliza una sola línea implementada para PTA0 es por tanto que la tarjeta se diseño con un mecanismo que permite al usuario utilizar el mismo conector DB9 para comunicación con periféricos y para programación, este mecanismo consiste en cambiar de posición dos “jumpers” que se encargan de habilitar la comunicación con periféricos por interfaz serial (SCI) y la programación por modo monitor.

Figura 11, circuito implementado para comunicación



Fuente: autor del proyecto

2.7. ESPECIFICACIONES ELÉCTRICAS DE LA TPM

La **TPM** esta diseñada para utilizar el microcontrolador a 5V lo cual implica que la máxima corriente que consume el microcontrolador es 150mA y la máxima corriente de salida por cada pin es de 15mA con excepción de PTC0 – PTC4 quienes pueden dar una máxima corriente de 25mA.

El consumo de corriente de la LCD es de 300mA, el regulador lineal 7805 soporta una corriente máxima de 1 Amperio por lo tanto la salida de 5V de la tarjeta esta limitada para aplicaciones de control, no de potencia. La alimentación es proporcionada por una fuente regulada AC/DC de 1200mA, con el propósito de dar mayor comodidad al usuario la TPM tiene habilitado para el usuario dos pines de salida directos de la fuente de alimentación lo cual permite la implementación de otros circuitos sin necesidad de recurrir a otras fuentes.

2.8. ETAPA DE POTENCIA DE LA TPM

La TPM requiere una fuente que le provea 5V para el modo de aplicación y 7.5 para el modo programación lo cual se hace utilizando el regulador lm7805 y un diodo zener de 7.5V.

Se selecciono un L7805C-V por ser un regulador de bajo costo y fácil consecución en el mercado y por cumplir con los requerimientos de potencia necesarios mencionados anteriormente.

La razón por la que se seleccionó un diodo zener es que no consume voltaje para su funcionamiento como ocurre con el regulador lineal 7808, esta ventaja del diodo zener, permite usar una fuente de 9V, y debido a que sólo se requiere este voltaje para el IRQ se hace más eficiente su uso.

2.9. PROTECCIONES ELÉCTRICAS DE LA TPM

La **TPM** tiene una protección en caso de que sea conectada la fuente de alimentación con la polaridad incorrecta, esta protección es un puente de diodos el cual impide que los dispositivos sufran una avería.

La tarjeta posee 2 salidas, las cuales proveen al usuario el voltaje que la fuente este utilizando. En el caso de estar conectada la fuente inversamente las salidas de estos pines también estarán polarizadas inversamente.

En caso de que la **TPM** sea conectada a una fuente de 12V o mayor voltaje el diodo zener limitara la entrada al IRQ a 7.5V, El regulador 7805 también limitara el voltaje de entrada al microcontrolador a 5V, aunque la tarjeta funcione el diseño no esta implementado para ese valor de voltaje de entrada.

2.10 LCD

La LCD se implementó con tres pines de control y 4 pines de datos. correspondientes a los pines de los puertos PTC Y PTD, 2 pines de alimentación, tierra y 5V y un pin conectado a un resistencia variable para graduar la intensidad de la pantalla de la LCD, todos estos pines están implementados para que se conecten la LCD dentro de la misma tarjeta lo cual genera mayor comodidad al usuario.

2.11. TECLADO

El MC08HC908GP32 posee un puerto destinado al uso de teclado el cual no requiere ninguna implementación eléctrica extra, este puerto es PTA y tiene 8 pines de entrada/salida. La tarjeta esta diseñada para conectar un teclado numérico de 4*4.

2.12. DISEÑO DEL PCB

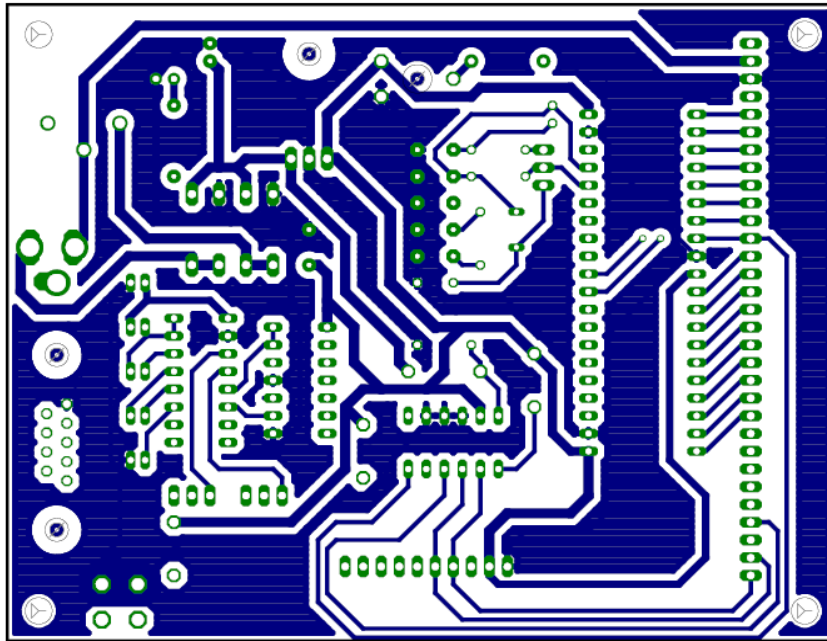
El PCB se diseño utilizando la herramienta EAGLE 4.01 versión estudiantil el cual es un programa que presenta facilidad de uso y auto enrutamiento de los caminos.

Los criterios que se han tenido en cuenta son:

- Los cruces son realizados con un ángulo de 45 y 135 grados para mantener un estándar, aunque las bajas frecuencias de trabajo del circuito no permiten que el ángulo de los caminos afecte al circuito.
- Mantener una distancia de separación entre camino no inferior al mismo ancho el camino.
- Los caminos que corresponden a alimentación deben ser gruesos con el propósito de disminuir la resistencia.
- Tener un amplio apantallamiento de tierra para evitar que el ruido externo afecte la señal generada en el micro.

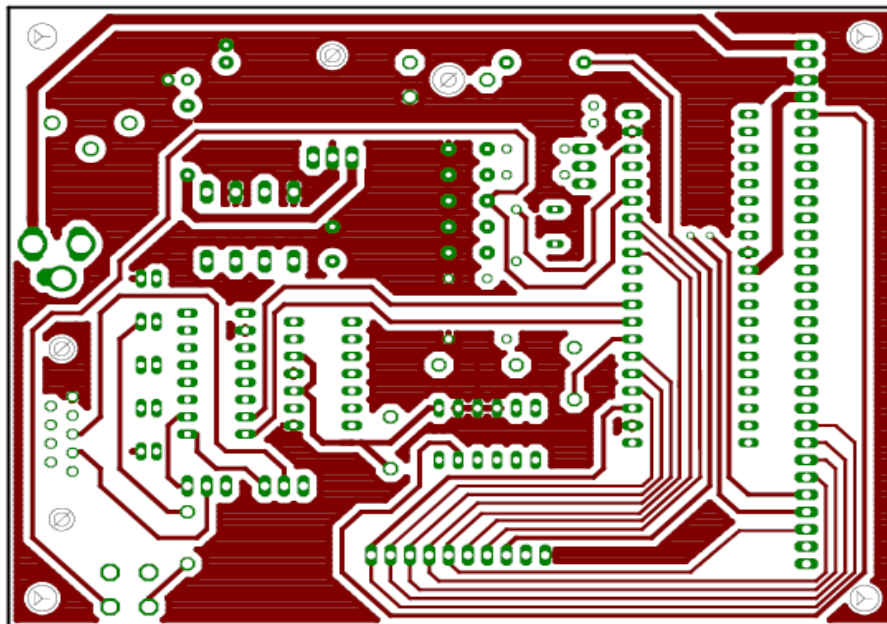
El circuito se implemento en doble cara en un area de 12 x 9 cm. el circuito posee "tought hole y antisolde" esto permite la conexión entre las caras del circuito. La cara inferior BOTTOM es presentada en la figura 10 y la superior TOP es presentada en la figura 11, los elementos que aparecen en la tarjeta están en la figura 12 y el esquema eléctrico de toda la tarjeta esta en la figura 13.

Figura 12, cara inferior del circuito (bottom)



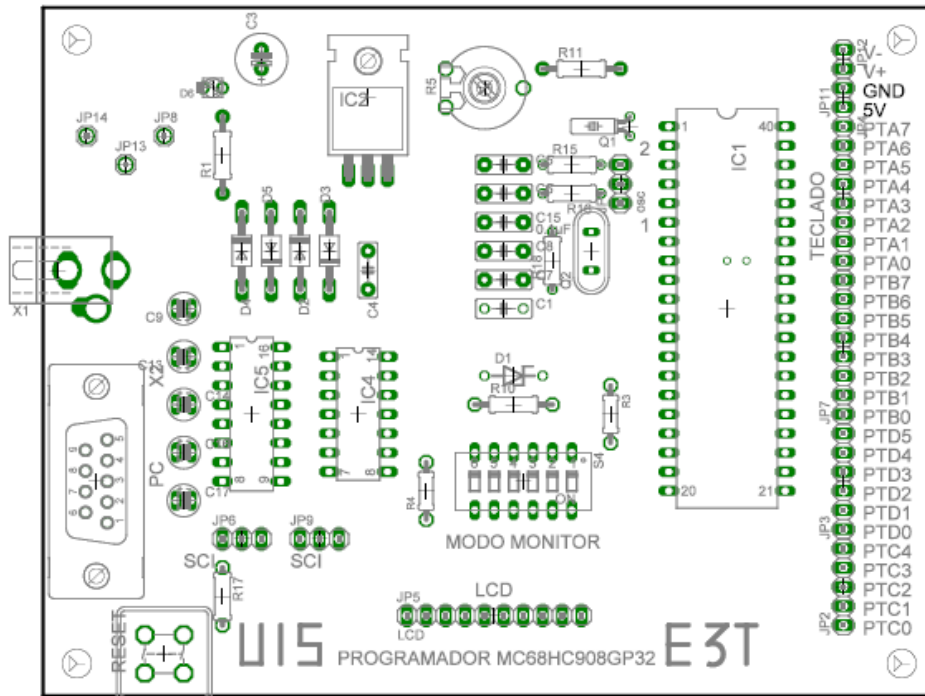
Fuente: el autor del proyecto

Figura 13, cara superior del circuito (top)



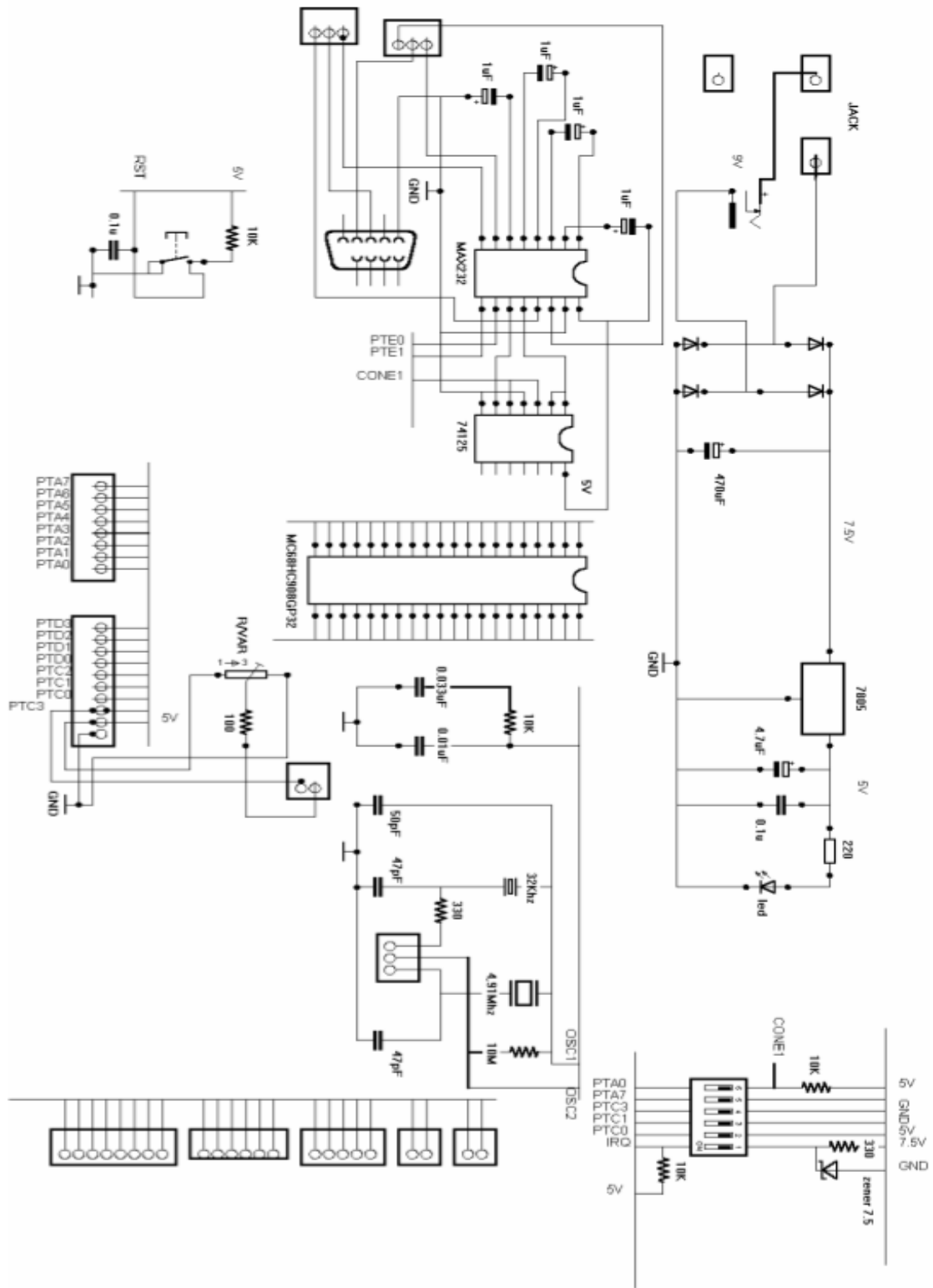
Fuente: el autor del proyecto

Figura 14, elementos que se encuentran dentro del circuito



Fuente: el autor del proyecto

Figura 15, Esquema eléctrico de la tarjeta



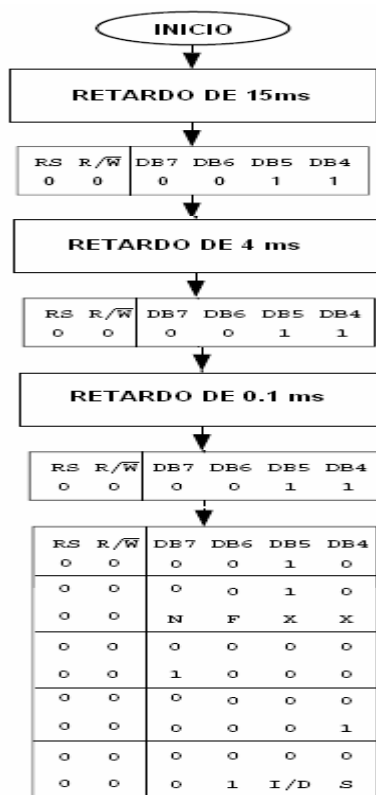
3 SOFTWARE DESARROLLADO

En este capítulo se hace referencia al software desarrollado para la implementación de la tarjeta de desarrollo, para este caso se tiene el programa de inicialización y uso de una pantalla de cristal y el programa para manejo de un teclado matricial de 4x4.

3.1 MODO DE INICIALIZACIÓN DE LA LCD

para la implementación y uso de la tarjeta LCD se hace necesario un algoritmo de inicio en el que se le indica que la tarjeta va a ser usada, este diagrama de flujo se presenta en la Figura 12.

Figura 16, diagrama de flujo de inicialización de la lcd



una LCD requiere de dos pines de alimentación y un pin de voltaje variable para la intensidad de la pantalla, también requiere de tres pines de control y para este caso de 4 pines de datos, la distribución de los pines de la LCD en la TPM se presenta en la tabla 2

tabla 2, distribución de pines de la LCD

GND	VDD	VAR	RS	RW	E	DATO0	DATO1	DATO2	DATO3
GND	5V	PTC3	PTC0	PTC1	PTC2	PTD0	PTD1	PTD2	PTD3

Fuente: autor del proyecto

Las instrucciones necesarios para cada aplicación de la LCD aparecen en la tabla 3

tabla 3, instrucciones de la LCD

Instrucción	Código										Descripción	Tiempo max de ejecución
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Borra el display y coloca el cursor en la primera posición 0 DDRAM	82µs~1.64ms
Return home	0	0	0	0	0	0	0	0	1	*	Coloca el cursor en la posición de inicio y hace que el display comience a desplazarse desde la posición original. El contenido de la DDRAM no varia	40µs~1.64ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Establece el sentido de desplazamiento de la información en el display. Esta operación se realiza durante la lectura o escritura de la DDRAM	40µs
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	Activa o desactiva poniendo en ON/OFF tanto el display D=0 (off) o D=1(on), como el cursor C=0(off) o C=1(on) y establece si este debe parpadear o no B=0(off) o B=1(on)	40µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	*	*	Mueve el cursor y desplaza el display sin cambiar el contenido de la DDRAM	40µs
Function set	0	0	0	0	1	DL	N	F	*	*	Establece el tamaño de interfase con el bus de datos(DL), el número de líneas del display(N) y la font de los caracteres	40µs
CG RAM address set	0	0	0	1	Dirección CGRAM					Establece la dirección de CGRAM a partir de la cual se almacenan los caracteres de usuario	40µs	
DD RAM address set	0	0	1	Dirección de la DDRAM					Estable la dirección DDRAM a partir de la cual se almacenan los datos a visualizar	40µs		
Read Busy Flag and Address	0	1	BF	Dirección de DDRAM o CGRAM					Lectura del flag de Busy e indica de la dirección de la CGRAM o DDRAM última empleada.	1µs		
Write data into the CG RAM or the DDRAM	1	0	Dato a escribir					Escribe en DDRAM o CGRAM los datos que se quieren presentar en el LCD	40µs			
Read data from the CG RAM or the DDRAM	1	1	Dato a leer					Lee de la DDRAM o CGRAM los datos que se direccionen	40µs			

I/D = 1 Incrementa la dirección del cursor I/D = 0 Decrementa la dirección del cursor S = 1 Desplaza la visualización cada vez que se escribe un dato S/C = 1 Se desplaza la visualización S/C = 0 Se desplaza el cursor R/L = 1 Desplazamiento a la derecha R/L = 0 Desplazamiento a la izquierda DL = 1 Trabaja con bus de datos de 8 bits DL = 0 Trabaja con bus de datos de 4 bits N = 1 La presentación se hace en 2 líneas N = 0 La presentación se hace en 1 línea F = 1 Caracteres de 5x10 dots F = 0 Caracteres de 5x7 dots BF = 1 LCD ocupado en una operación interna BF = 0 LCD disponible para aceptar instrucciones D = 1 Pantalla activa (ON) C = 1 Cursor activo (ON) * Indeterminado	DDRAM: Display Data RAM CGRAM: Character Generator RAM ACC : Address of CGRAM ADD : Address of DDRAM AC: Address counter used for both DDRAM and CGRAM
--	---

fuelle:

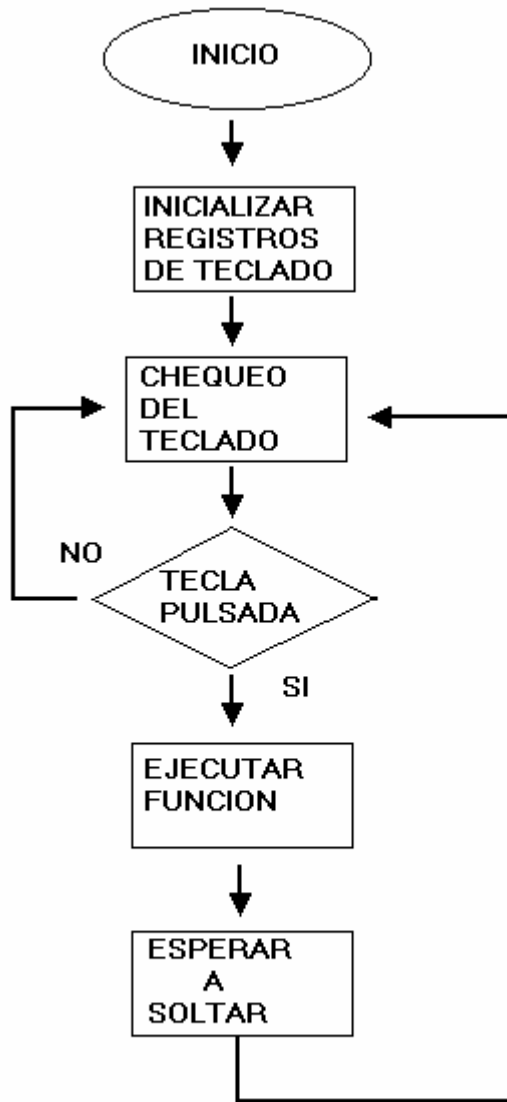
para analizar el algoritmo implementado para control de una LCD utilizando codewarrior diríjase al anexo D

3.3 MODO DE USO DEL TECLADO

Este programa se basa en el uso de un teclado matricial de 4x4, para el uso de este teclado con la TPM se utilizo el puerto PTA el cual presenta la ventaja de poseer resistencias de pull-up configurables por software, lo cual permite fijar los voltajes de entrada en 5V eliminando la posibilidad de ruido externo.

El algoritmo consiste en configurar 4 pines de un puerto como entradas y los restantes como salidas, en los pines de entrada se activa una resistencia de “pull up” la cual fija la entrada a un valor de 5, estos pines se activan en el caso de que el valor en la entrada del pin sea igual a cero. En los cuatro pines configurados como salidas se rota un cero, si una de las entradas presenta un bajo eso indica que el teclado a sido presionado, conociendo cual salida y cual entrada se encuentra en 0 se identifica el valor numérico seleccionado para interpretar mejor el proceso ver diagrama de flujo del teclado Figura 17.

Figura 17, Diagrama de flujo del teclado



Fuente: autor del proyecto

PRUEBAS REALIZADAS

En esta parte se explica como se realizaron las pruebas para cada uno de los módulos de la **TPM**.

- Para las salidas de propósito general se activaron todos los pines en los modos de entrada y salida ejecutando un programa y se observó que el circuito no presentó fallas.
- Para el uso de la LCD se implementó un programa con el cual se activó la pantalla y se visualizaron caracteres alfanuméricos los cuales no presentaron fallas.
- Para el uso del teclado se implementó un programa con el cual se verificó el funcionamiento del mecanismo de conexión de éste a la tarjeta.
- Para el uso del preescalador se implementó un programa que permitiera a la tarjeta trabajar al máximo de su frecuencia, lo cual resultó satisfactorio al visualizarse un aumento en la frecuencia de activación de los pines de salida previamente programados para esta función.
- Para los conversores analógico digitales se probó que todos los pines del puerto donde se encuentran los conversores realizaran conversión A/D y que cumpliera las referencias técnicas. En esta prueba se observó un error inferior al 2% de muestreo, el cual se puede atribuir al ruido del sistema y la calidad de la señal muestreada.
- Para el uso de los TIMER se implementó un programa en el cual se probó el uso de un PWM, lo cual resultó satisfactorio.

CONCLUSIONES

Finalizada la investigación y las pruebas que se realizaron, a continuación se resaltan las conclusiones más destacadas

- Se concluyó que la implementación de una tarjeta de desarrollo para el MC68HC908GP32 que contenga el modo aplicación y el modo programación dentro del mismo circuito no afecta el funcionamiento de cada uno de los modos.
- La implementación de dos cristales dentro de la tarjeta no afecta el funcionamiento de la misma y permite al usuario manejar una de las mas importantes potencialidades del microcontrolador que es el modulo preescalador.
- La implementación de circuitos impresos utilizando el proceso de CAD CAM asegura la homogeneidad entre las tarjetas desarrolladas, pues son hechas con las mismas características.
- Las resistencias de pullup del microcontrolador facilitan y reducen los requisitos para la implementación de un teclado de 4x4

OBSERVACIONES Y RECOMENDACIONES

- Para el caso de implementar una tarjeta para toda la familia 68HC08 asegurarse de usar un reloj externo compatible con todos los miembros de esta familia.
- El microcontrolador esta diseñado para aplicaciones de baja potencia ya que este es un dispositivo de control y sus salidas pueden ofrecer un máximo de 15mA por tanto se recomienda que en caso de requerir mas potencia se use un buffer
- En el momento de programar el microcontrolador puede ocurrir que se genere un reset por 'opcode ilegal' cuando la CPU descifra una instrucción que no esta definida en el opcode o cuando ocurre ruido excesivo o fallas en el proceso debido al usuario. En estos casos se recomienda que reinicie todo el proceso de programación del microcontrolador

BIBLIOGRAFÍA

ACEVEDO, jhon. Evaluación de procesos de diseño CAD-CAM y de fabricación de circuitos impresos (tesis de grado).biblioteca UIS

BROOKS, Doug. 90 Degree Corners:The Final Turn (PDF de ultracad)
.http:// www.ultracad.com

FREESCALE, System design and layout techniques for noise reduction in MCU-based systems (PDF de freescale referencia AN1259).
http://www.freescale.com

FREESCALE, Noise Reduction Techniques for Microcontroller-Based Systems (PDF de freescale referencia AN1705). http://www.freescale.com

FREESCALE, Designing for electromagnetic compatibility with single chip microcontrollers (PDF de freescale referencia AN1263).
http://www.freescale.com

FREESCALE, Datasheet MC68HC08 Microcontrolers (libro de freescale) este libro se adquiere en la página http://www.freescale.com

HIGUERA, Jorge Eduardo Sistemas de desarrollo para un lector/escritor de tarjetas inteligentes diseño y construcción (tesis de grado).biblioteca UIS

MAXIM, Datasheet MAX232 (PDF de MAXIM)
http://www.maxim-ic.com

METROWERK'S, CodeWarrior IDE Tutorial. www.metrowerks.com/pdf

TEXAS INSTRUMENTS, Datasheet SN74125 (PDF de Texas Instruments)
<http://www.ti.com>

ANEXO A: MANUAL DE USO DE LA TARJETA MC68HC908GP32

INTRODUCCIÓN

Este manual es una guía básica para el uso de la tarjeta TPM para el microcontrolador MC68HC908GP32 de Freescale, antes de usar la tarjeta lea todo el manual.

ADVERTENCIA

El microcontrolador esta diseñado para manejar 10mA como máximo en la salida de los pines, recuerde no solicitarle mas corriente pues puede dañarlo.

Asegúrese que la superficie sobre la que pone la tarjeta este limpia y seca y no sea un material conductor para reducir el riesgo de un corto circuito.

1. COMO PROGRAMAR USANDO LA TARJETA

1. Conecte la tarjeta a la alimentación, asegúrese que la polaridad de la fuente sea positivo en el interior y negativo en la periferia del conector, de no conectarse de esta manera la salida de voltaje habilitada en la tarjeta con valores V+ y V- quedara polarizada inversamente aunque el microcontrolador no sufrirá daño ya que se tiene una protección.
2. Conecte el cable DB9 al puerto del computador y cerciórese que este configurado el puerto para ser usado.
3. Ponga los switches de color azul en ON como se muestran en la figura ya que esto hace que el microcontrolador entre en modo monitor al activarse los valores que se muestran en la Tabla 1.

Figura 1, TPM en modo programación

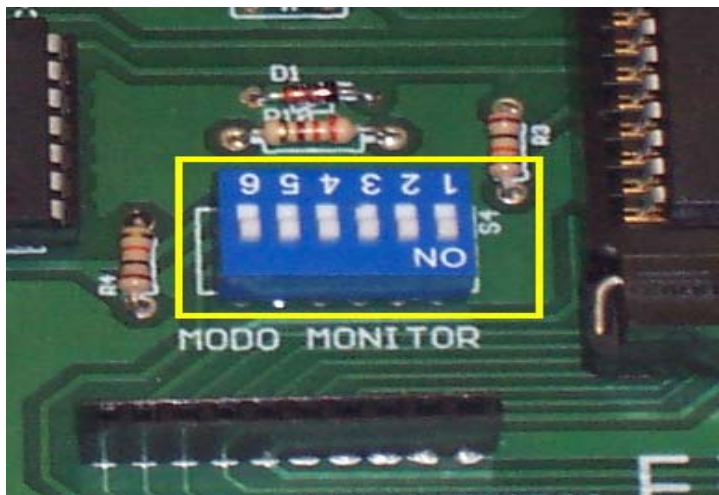


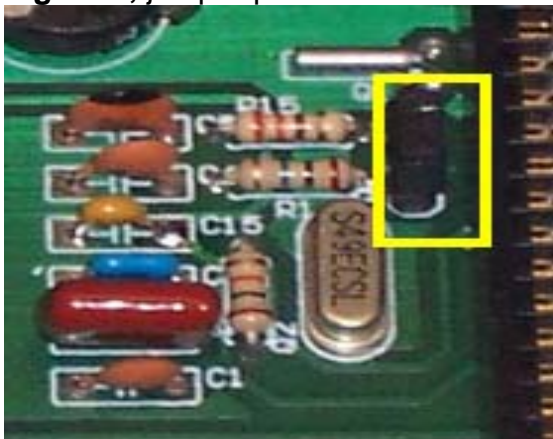
Tabla1, niveles para modo monitor

IRQ	RESET	\$FFFE/ \$FFFF	PLL	PTC0	PTC1	PTC3	RELOJ	FREQ BUS	PTA0	PTA7
VDD+2.5	VDD	XXX	OFF	VDD	GND	GND	4.9142	2.4576	VDD	GND

VDD =5, para la TPM

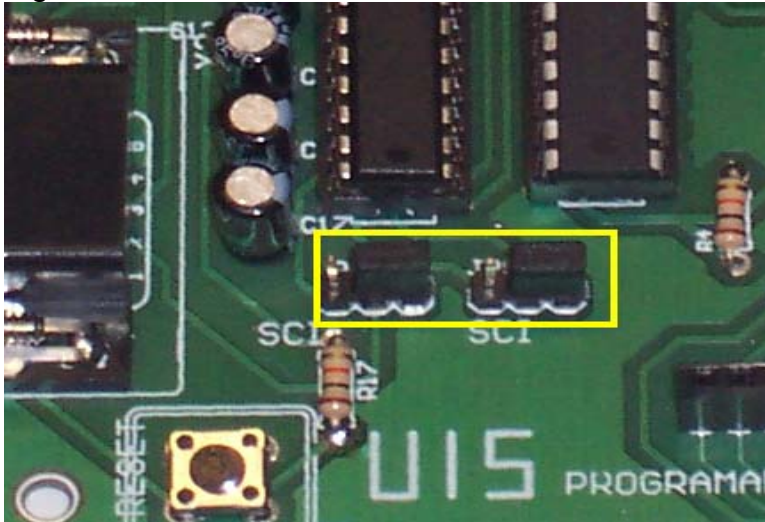
4. Conecte el cristal de 4.9152Mhz como se muestra en la Figura 2, existen 2 cristales controlados por un jumper uno para programación y aplicación y otro para el PLL el cual se conecta al cortocircuitar el pin del centro con el pin 1 para poder utilizar el cristal de 4.9152Mhz, este cristal es para modo programación y aplicación

Figura 2, jumper para selección de cristal



5. Para programar el microcontrolador recuerde mantener los jumper al lado derecho como aparece en la Figura 3, para utilizar el modo comunicación SCI coloque los jumper al lado izquierdo donde aparece el letrero SCI.

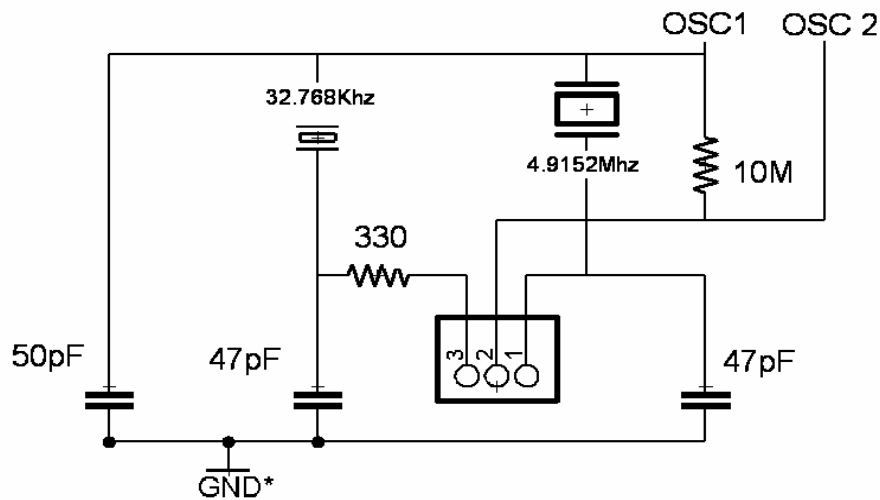
Figura 3 selección de modo de comunicación



2. COMO UTILIZAR EL PRESCALADOR?

En el caso de utilizar el modulo preescalador se hace necesario cortocircuitar el pin del centro con el pin numero 2 recuerde que cuando activa el cristal para utilizar el preescalador no se puede programar por tanto cuando deje de utilizarlo devuélvalo a la posición 1 para que pueda entender como funcionan estos dos cristales en la Figura 4 aparece el esquema eléctrico de los cristales

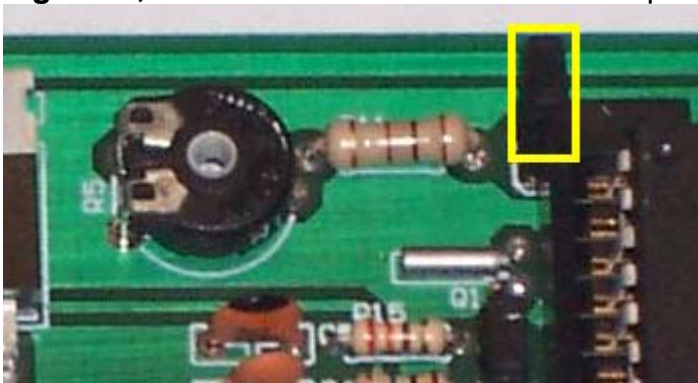
Figura 4, Esquema eléctrico de los cristales



3. COMO UTILIZAR LA PANTALLA LCD

1. Se hace necesario colocar el jumper de la resistencia variable, para de esa forma variar la intensidad de la pantalla LCD como se muestra en la Figura 5

Figura 5, conector de la resistencia variable para la LCD



La tarjeta tiene implementado un mecanismo para insertar una LCD (recuadro amarillo de la Figura 6) pero las salidas de alimentación, resistencia variable y los pines de control y datos también están disponibles al usuario, por los pines que se muestran en la Figura 6 y se indican en la Tabla 2.

Figura 6, pines de conexión de la LCD

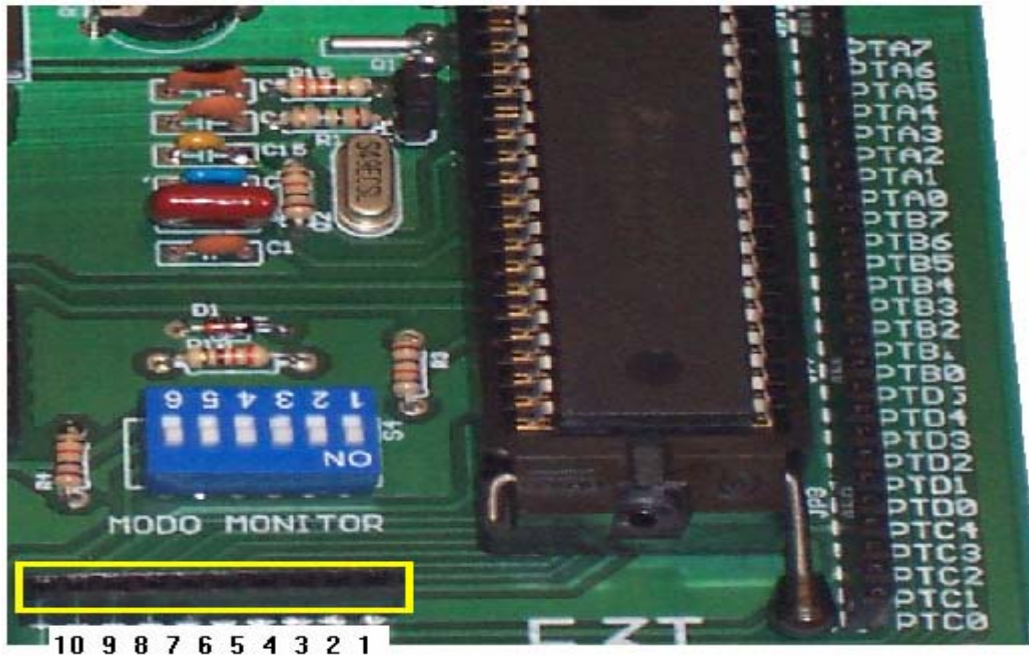


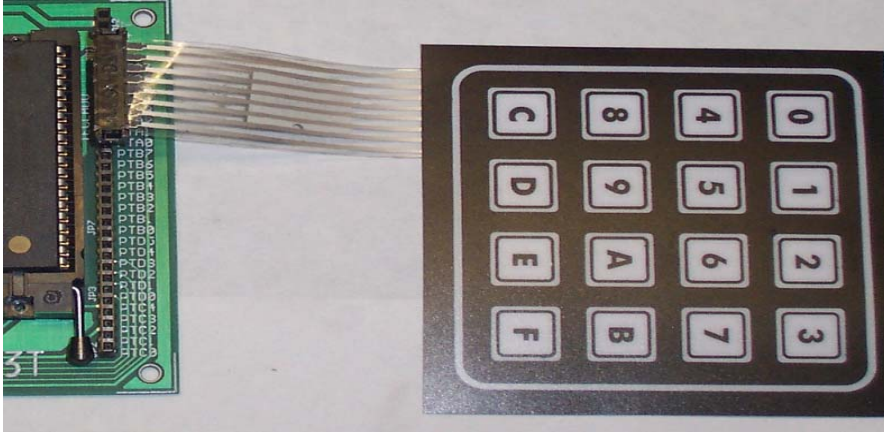
Tabla 2, Conexiones de la LCD a la TPM

1	GND	GND
2	5V	5V
3	PTC3	Intensidad de la pantalla
4	PTC0	R/S
5	PTC1	R/W
6	PTC2	E
7	PTD0	DATO1
8	PTD1	DATO2
9	PTD2	DATO3
10	PTD3	DATO4

4. COMO USAR EL TECLADO

Para esto inserte en el puerto PTA el teclado, tenga en cuenta que la tarjeta tiene escrito teclado como se observa en la Figura 7

Figura 7, Conexión del teclado a la TPM



5. COMO USAR EL MODULO DE COMUNICACIÓN?

Para utilizar el modo comunicación SCI coloque los jumper cortocircuitando el centro con el pin del lado izquierdo donde aparece el letrero SCI, **OJO**; recuerde devolver los jumper al lado derecho, si va a utilizar el modo programación.

6. MODO APLICACIÓN

1. Para el modo aplicación se hace necesario que los switches de color azul estén en OFF esto inhabilita la comunicación con el computador. Se recomienda que el cambio de los switches a off se haga con la tarjeta apagada
2. Encienda la tarjeta

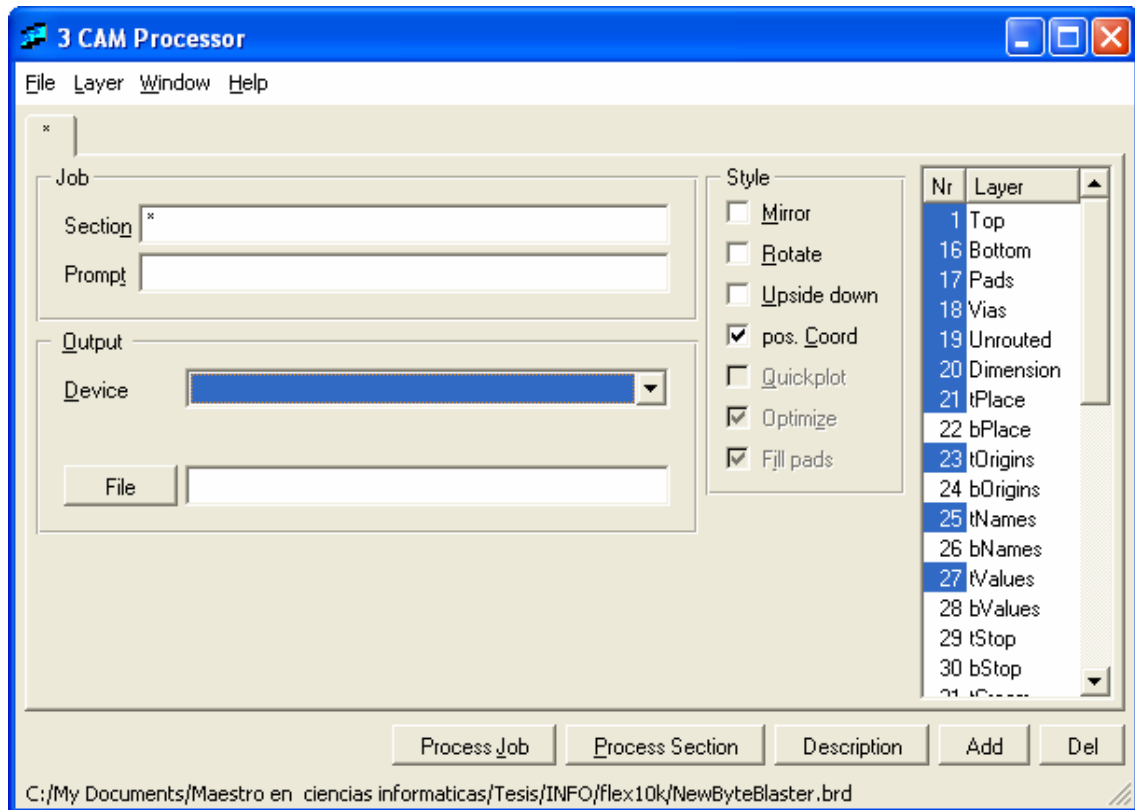
ANEXO B: COMO GENERAR ARCHIVOS GERBER DESDE EAGLE

INTRODUCCIÓN

Este anexo es una guía practica para generar archivos en formato Gerber para la producción de circuitos impresos utilizando CAD-CAM.

Para el caso del proyecto desarrollado estos son los archivos requeridos por la empresa microcircuitos en la ciudad de Cali.

- Una vez terminado el diseño, chequeado los errores (DRC) y editado el circuito (suavizar ángulos y líneas); se procede a llamar el CAM procesor activando con el mouse el logo del CAM procesor que aparece en la barra de tareas de la pantalla del eagle donde se realiza la board.(Archivo .brd)



El CAM procesor es una herramienta del eagle encargada de generar

archivos o impresiones de sus diseños que requieren los fabricantes de circuitos impresos.

Los archivos gerber se generan en el CAM procesor de la siguiente manera:

-En la casilla **DEVICE**, lugar donde se encuentra la maquina que va a leer estos archivos, se escoge la opción **GERBER_RS274X**.

-En la casilla **OUTPUT** se da el nombre y la ruta a donde quiere llevar los archivos gerber. Para cada archivo una extensión diferente.(Ej: C:\Mi_diseño\diseño.bot, donde “diseño” es el nombre de su circuito, y “C:\Mi_diseño” la ruta donde guardare los archivos)

-En el recuadro **LAYERS** es donde se activan las caras usadas para cada archivo gerber, las cuales se seleccionan dependiendo de la extensión.

-Las opciones de *mirror, rotate, upside down, pos. coord., quickplot, optimize* y *fill pads* se activan dependiendo de las necesidades del fabricante de la tarjeta o del diseñador de la misma.

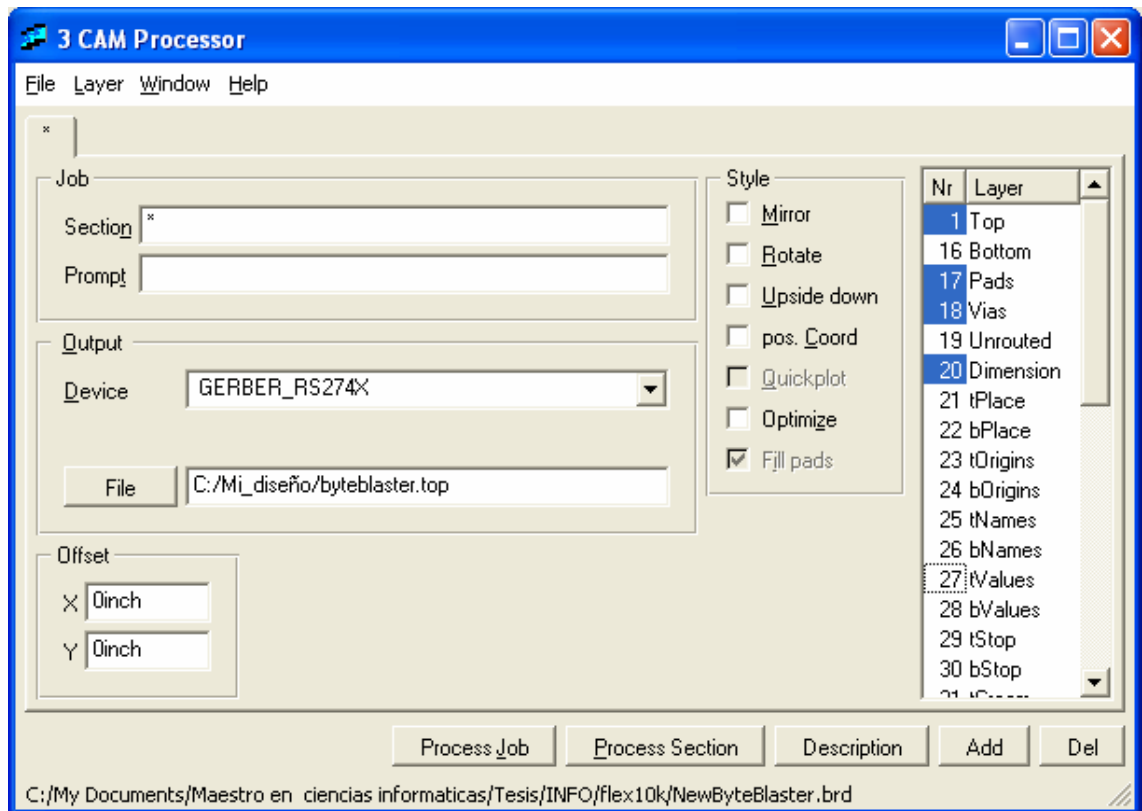
-La casilla de **offset** se modifica cuando se usa la impresora y queremos que el dibujo quede centrado o que quepan dos dibujos en una sola hoja.

En el siguiente listado se dan las extensiones de cada archivo gerber seguido de las caras necesarias, así como la explicación de su generación.

Pasos a seguir:

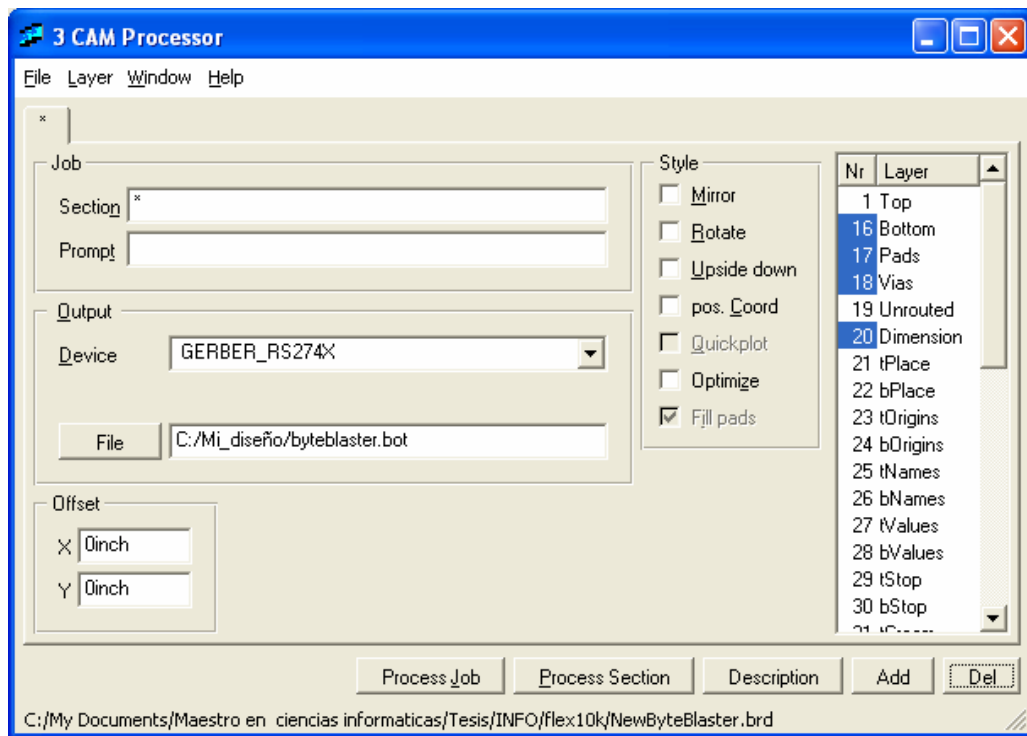
Suponga que se desea generar los archivos de un circuito llamado “**byteblaster**”:

1. **byteblaster.top**: (*pistas lado componentes*) (*Layers: top, pads, vias, dimension*)



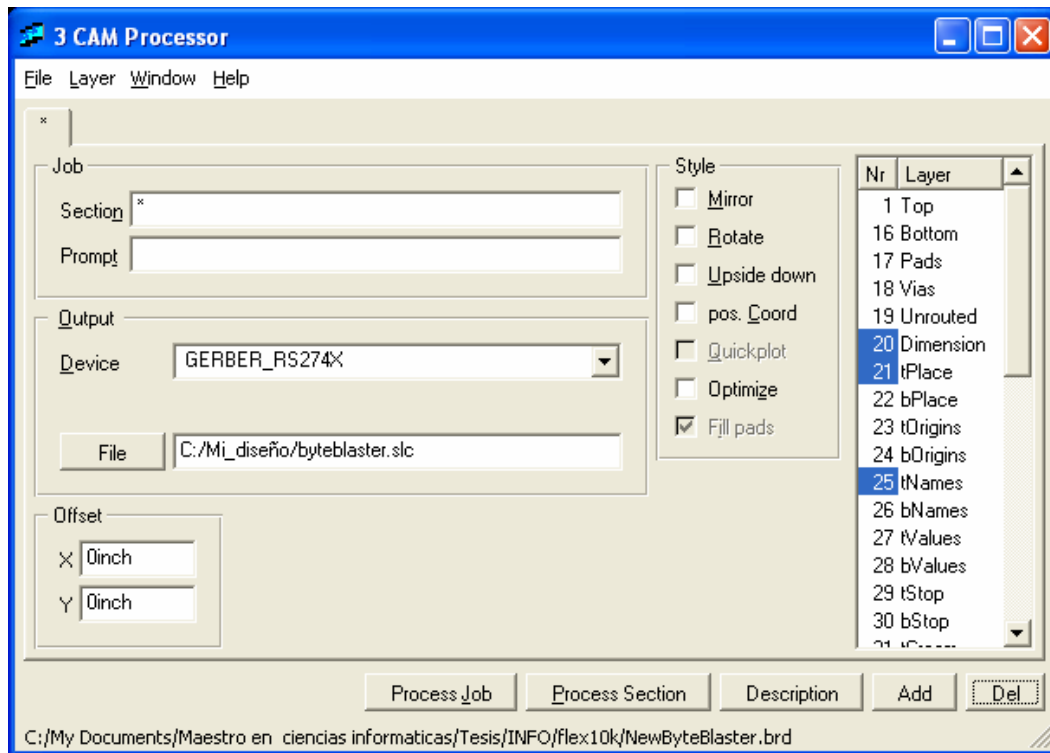
Se presiona **Process Job** aparece un indicador y si desaparece solo sin mensajes de error se genero el archivo con éxito.

2. **byteblaster.bot** (pistas lado soldaduras) - bottom, pads, vias, dimension



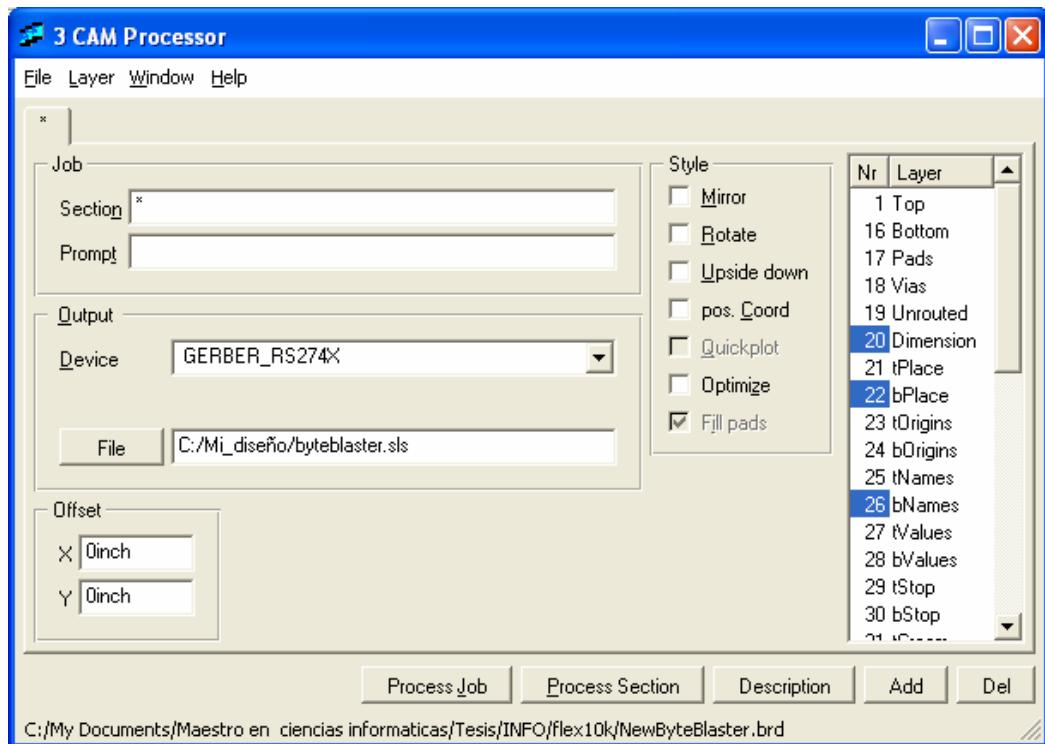
Se presiona **Process Job** aparece un indicador y si desaparece solo sin mensajes de error se genero el archivo con éxito.

3. **byteblaster.slc** (componentes lado componentes) - *tplace, tnames, dimension*



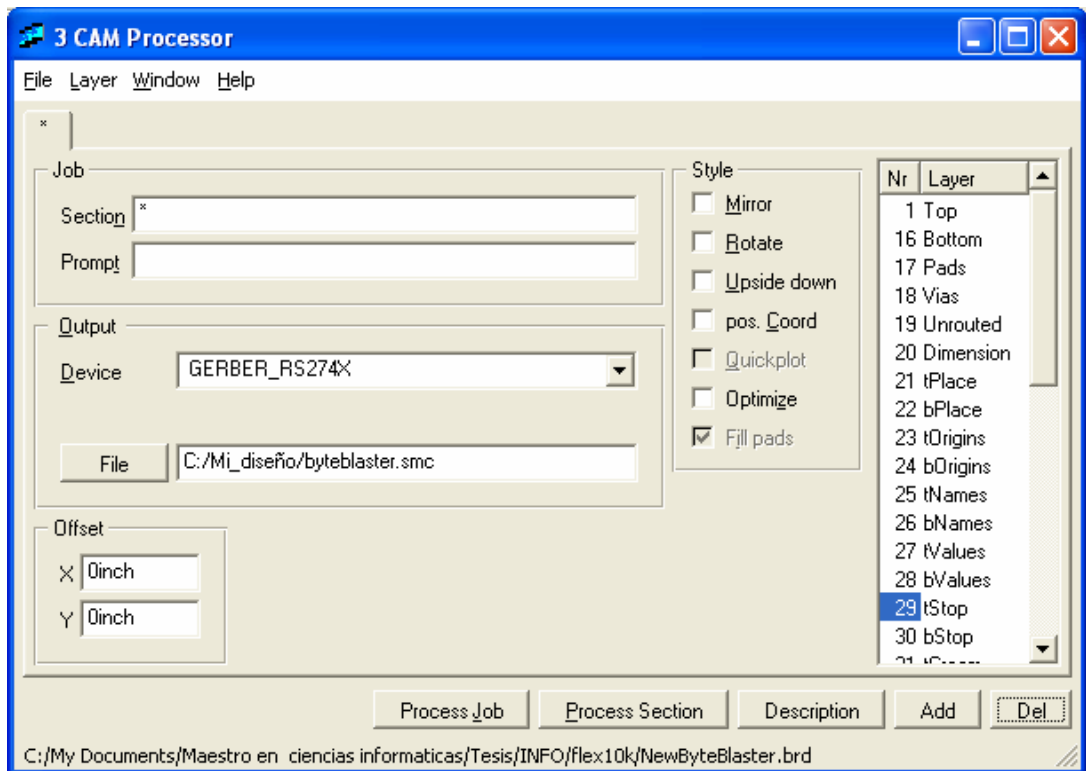
Se presiona **Process Job** aparece un indicador y si desaparece solo sin mensajes de error se genero el archivo con éxito.

4. **byteblaster.sls** (componentes lado soldaduras) - *bplace*, *bnames*, *dimension*



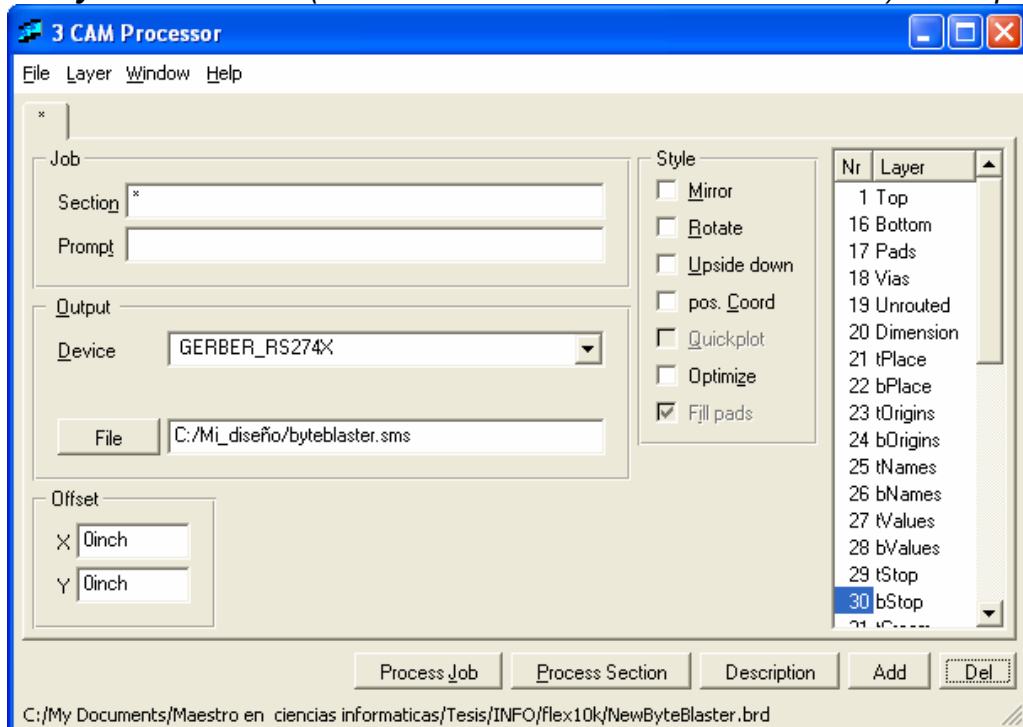
Se presiona **Process Job** aparece un indicador y si desaparece solo sin mensajes de error se genero el archivo con éxito.

5. **byteblaster.smc** (mascara de soldaduras lado componentes) - tstop



Se presiona **Process Job** aparece un indicador y si desaparece solo sin mensajes de error se genero el archivo con éxito.

6. *byteblaster.sms* (mascara de soldaduras lado soldaduras) - bstop



Se presiona **Process Job** aparece un indicador y si desaparece solo sin

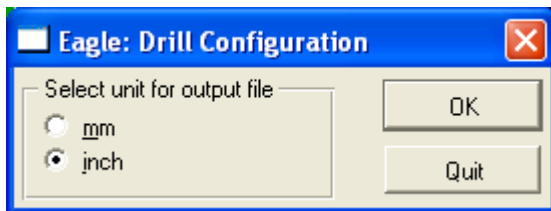
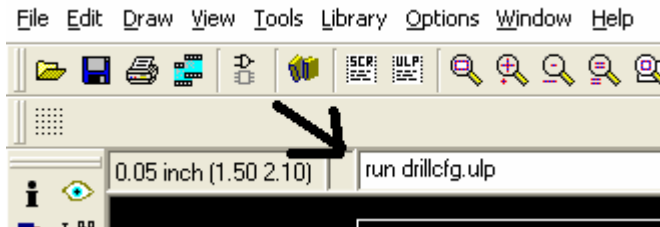
mensajes de error se genero el archivo con éxito.

7. **byteblaster.ncd** (perforaciones) - drills, holes

Se debe de modificar la casilla **DEVICE** y seleccionar *excellon*, de inmediato aparecerá una casilla llamada RACK en la cual se debe colocar la ruta a un archivo .drl donde se encuentran las brocas usadas en los diseños.

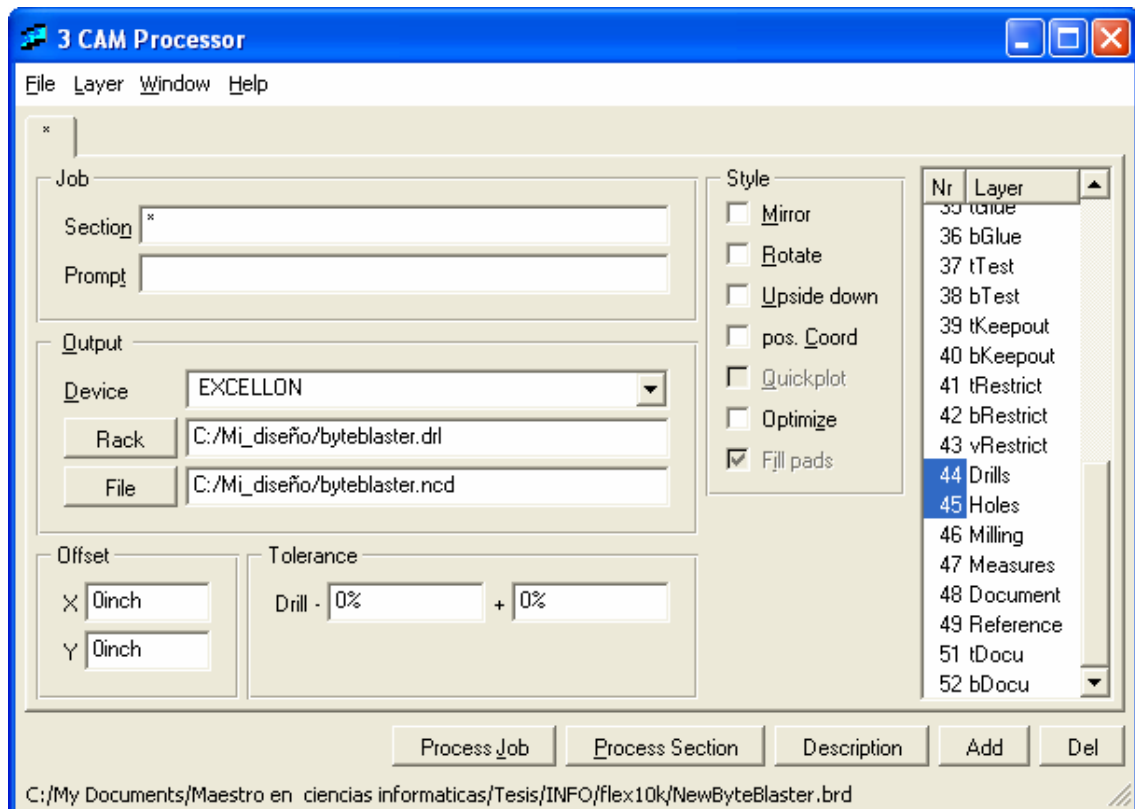
El eagle no tiene creado este archivo inicialmente, para ello, debemos ejecutar:

run drillcfg.ulp



Escoja inch, y presione OK hasta que le pregunte donde desea guardarlo, elija la misma carpeta donde ha estado trabajando y póngale el mismo nombre (En este caso C:\Mi_diseño\byteblaster.drl)

En la casilla casilla llamada RACK en se debe colocar la ruta del archivo .drl



Se presiona **Process Job** aparece un indicador y si desaparece solo sin mensajes de error se genero el archivo con éxito.

8. copie el contenido de la carpeta. **(Deben haber 10 archivos incluyendo un .gpi)**

Este listado es la forma general para los archivos gerber de una tarjeta doble faz, para tarjetas de una faz (un solo lado con cobre) no se generan los archivos de extensión top, sls y smc.

Las modificaciones que puede sufrir este listado dependen de las características del diseño, es decir, si se van a usar mas caras para cada archivo si las necesita.

ANEXO C: MANUAL BASICO DE USO DE CODEWARRIOR 3.0

INTRODUCCIÓN

Este es un manual básico para el uso de Codewarrior en la programación del microcontrolador MC68HC908GP32 utilizando lenguaje C

TABLA DE CONTENIDO

- COMO INICIAR PROCESADOR EXPERTO

En la ventana de inicio elija el icono file y luego new, después de esto aparecerá la pantalla de new en la cual usted elegirá HC(S)08 New Project Wisard (Figura 1) luego en New Project Wizard Page 1 elija el microcontrolador con el cual va a trabajara para este caso el MC68HC908GP32 (Figura 2) luego presione siguiente,

Figura 1

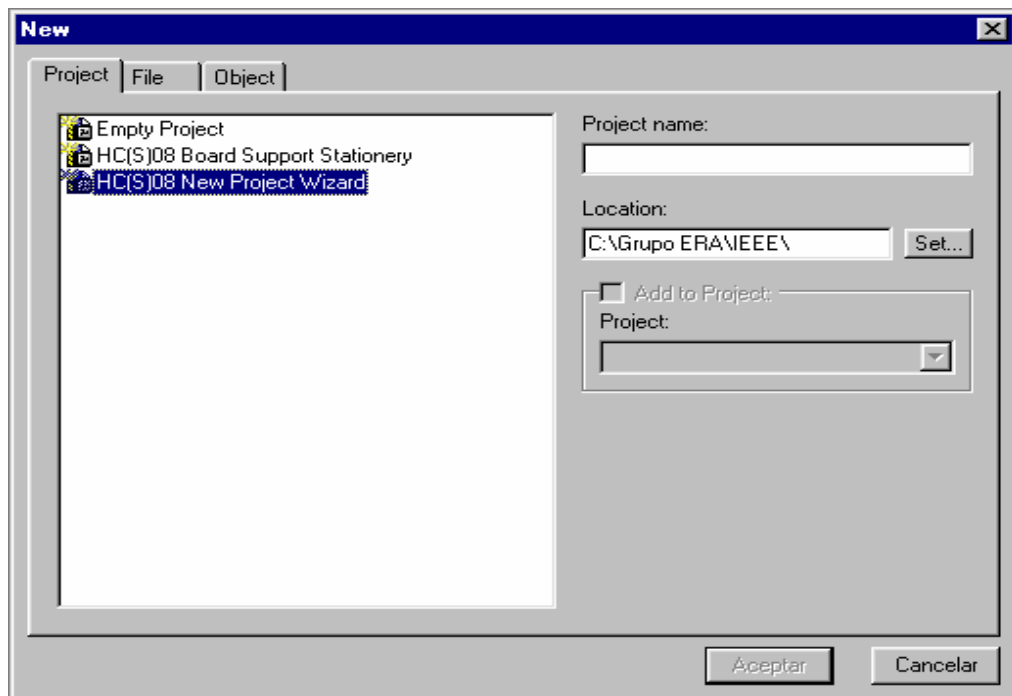
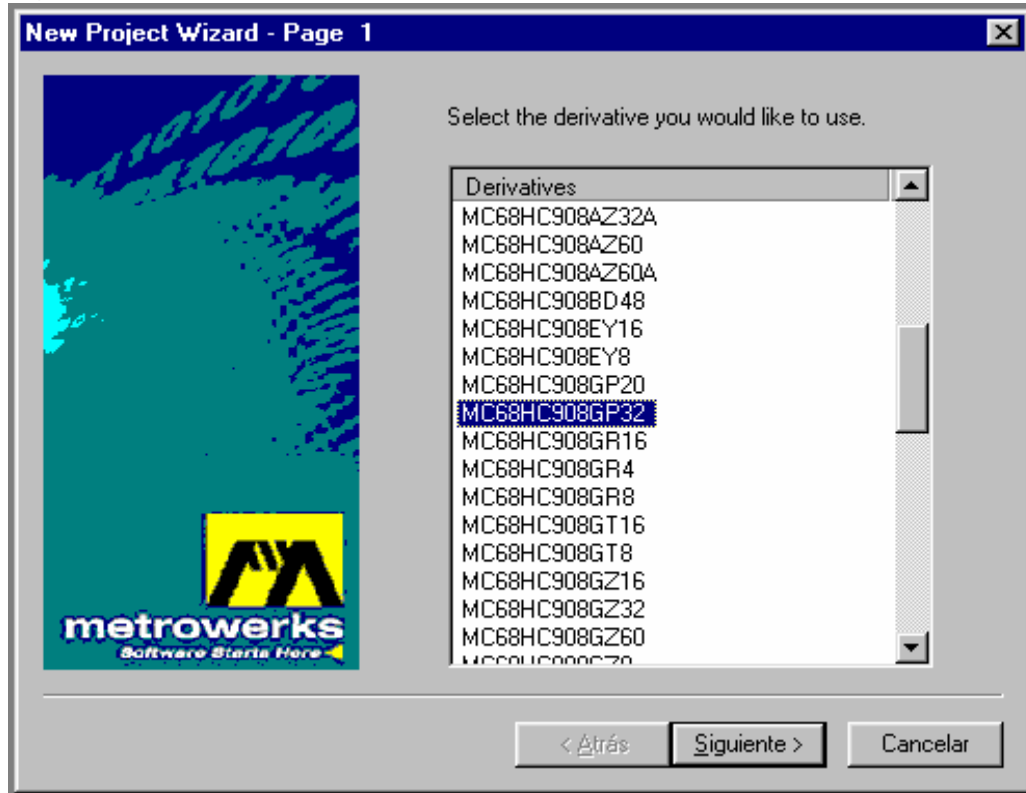


Figura 2



Esto genera la ventana New Project Wizard Page 2 donde usted elije los lenguajes para programar el microcontrolador para mayor facilidad en la programación señale los tres como aparece en la Figura 3.

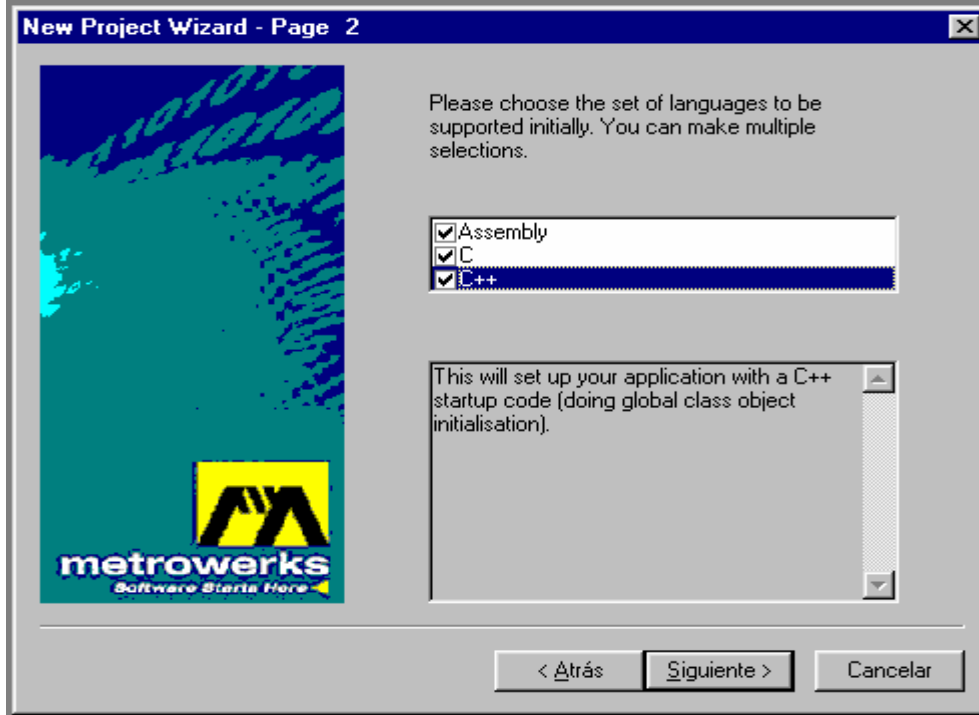
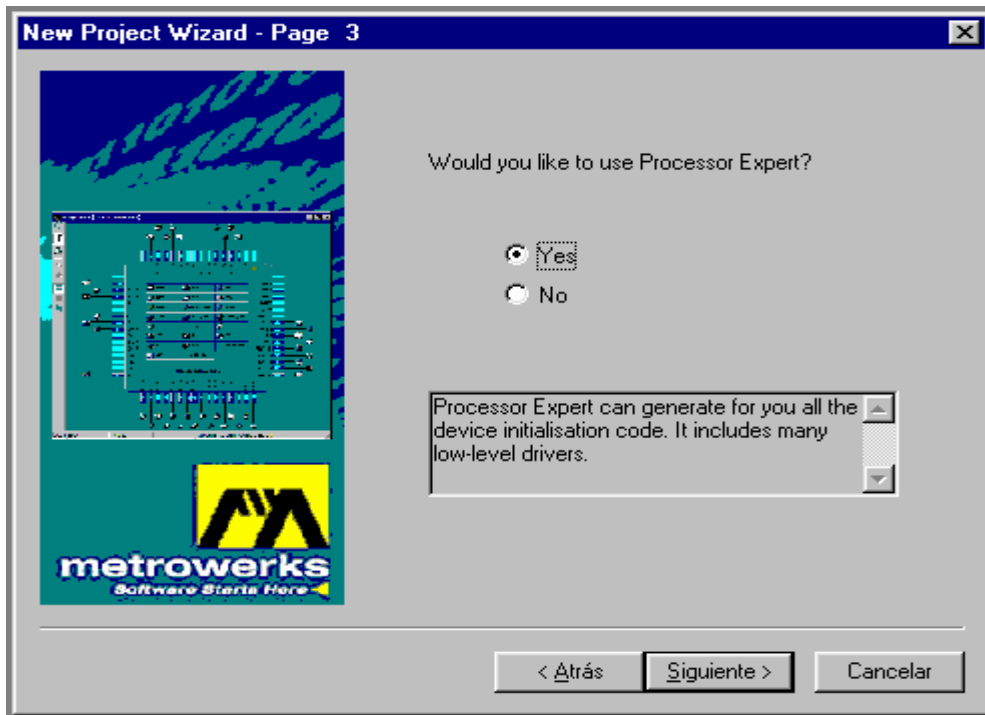


Figura 3

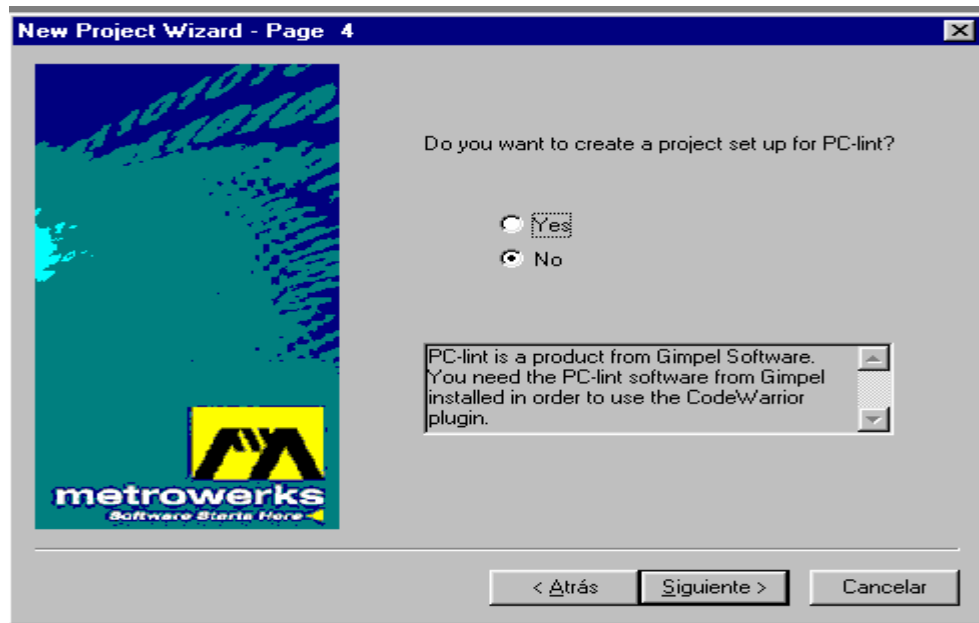
Oprima siguiente lo que generara la ventana New Project Wizard Page 3 (Figura 4) donde tendrá la opción de usar Processor Expert en este caso se recomienda usarlo ya que es una interfaz de fácil manejo y con mayores prestaciones para el usuario.

Figura 4



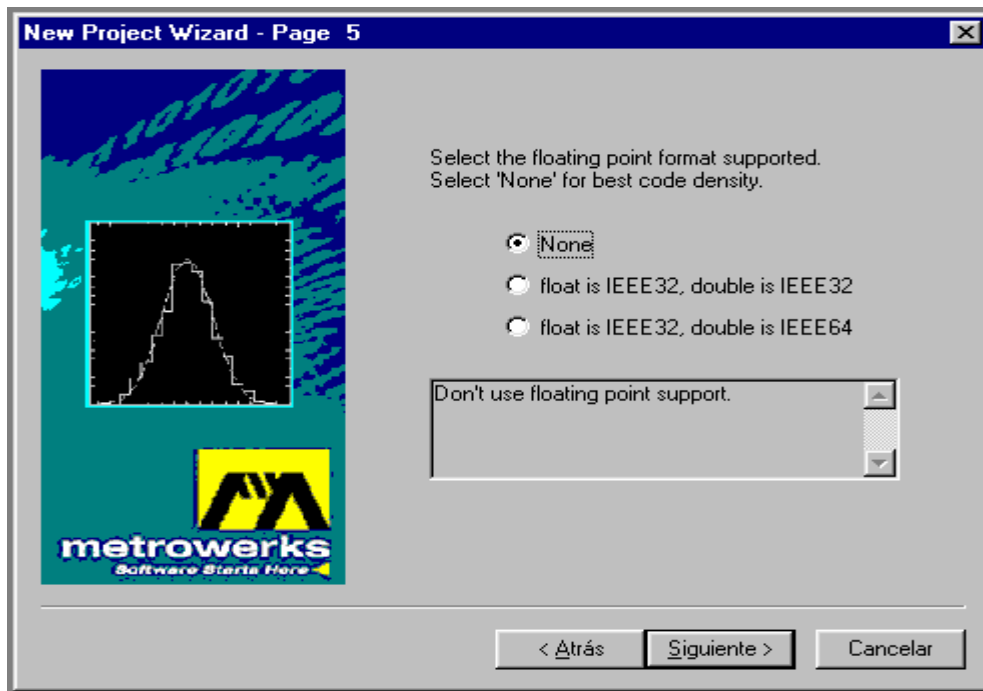
Cuando presione siguiente aparecerá una nueva ventana New Project Wizard Page 4 (Figura 5) la cual le da la opción de utilizar PC-lint el cual es un archivo necesario para la utilización de software de Gimpel para nuestro caso seleccione no.

Figura 5



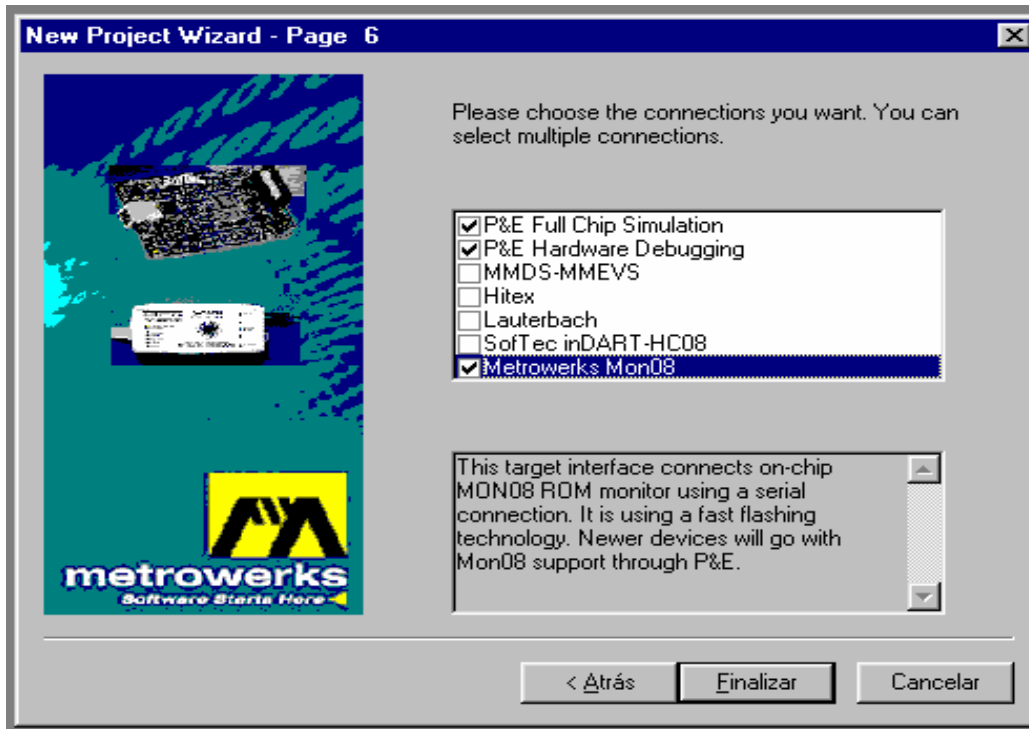
Cuando presione siguiente aparecerá una nueva ventana New Project Wizard Page 5 (Figura 6) la cual le da la opción de utilizar punto flotante y palabras de 32 o 64 bits, lo cual se selecciona según la necesidad del usuario y del algoritmo a generarse.

Figura 6



La siguiente ventana que se genera (figura 7) permite al usuario utilizar varios de los compiladores generados por las empresas que crean las tarjetas de programación en este caso recomendamos que siempre seleccione los 3 que aparecen en la figura, por defecto el programa compilara con P&E debugging programa necesario para realizar comunicación y ejecución con el microcontrolador en tiempo real, Metrowerks Mon08 este programa permite al usuario programar el microcontrolador y con P&E Full Chip Simulation usted puede simular el algoritmo desarrollado. En el caso de que usted use una tarjeta de programación de otra compañía seleccione la opción que esta le recomiende ya que puede que use comunicación USB o un protocolo diferente.

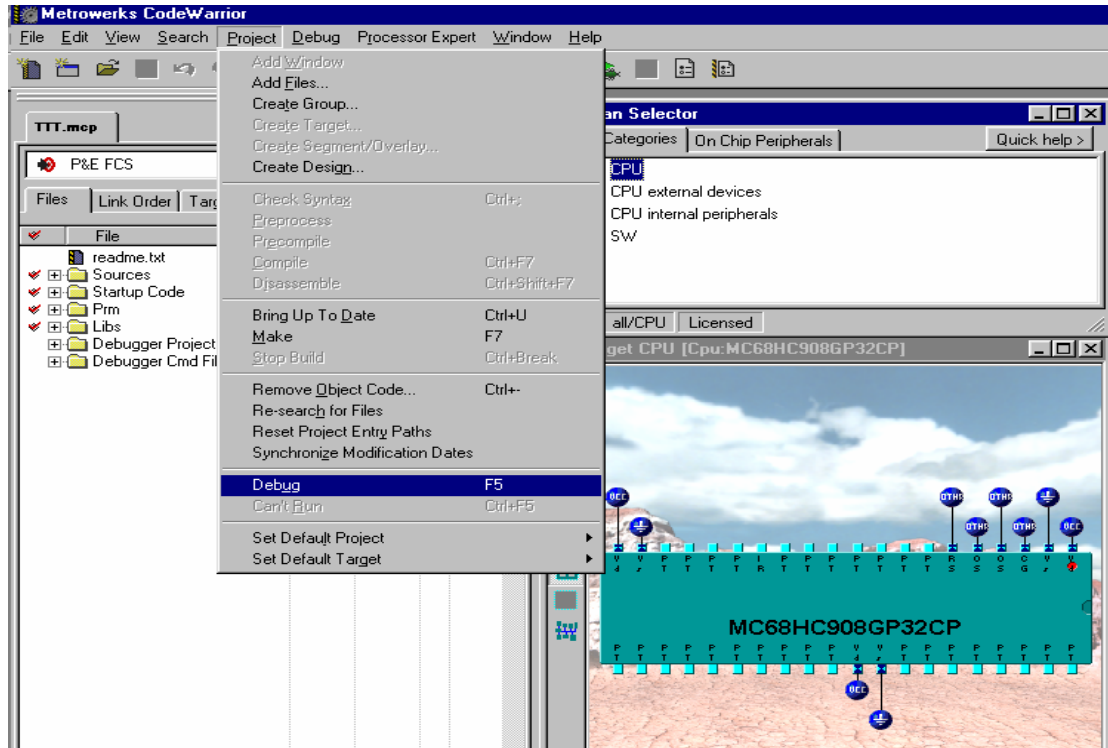
Figura 7



Después de presionar finalizar se generaran las ventanas necesarias para poder utilizar Codewarrior.

Para cargar los archivos de C y poder compilar su programa utilizando este lenguaje, debe seleccionar el icono Project y seleccionar Debug o en su defecto teclear F5 como aparece en la Figura 8

Figura 8



Esto genera los archivos que convierten los programas de C a ensamblador y las librerías necesarias para utilizar los BEANS, también genera el True-Time Simulator, como aparece en la Figura 9 y Figura 10.

Figura 9

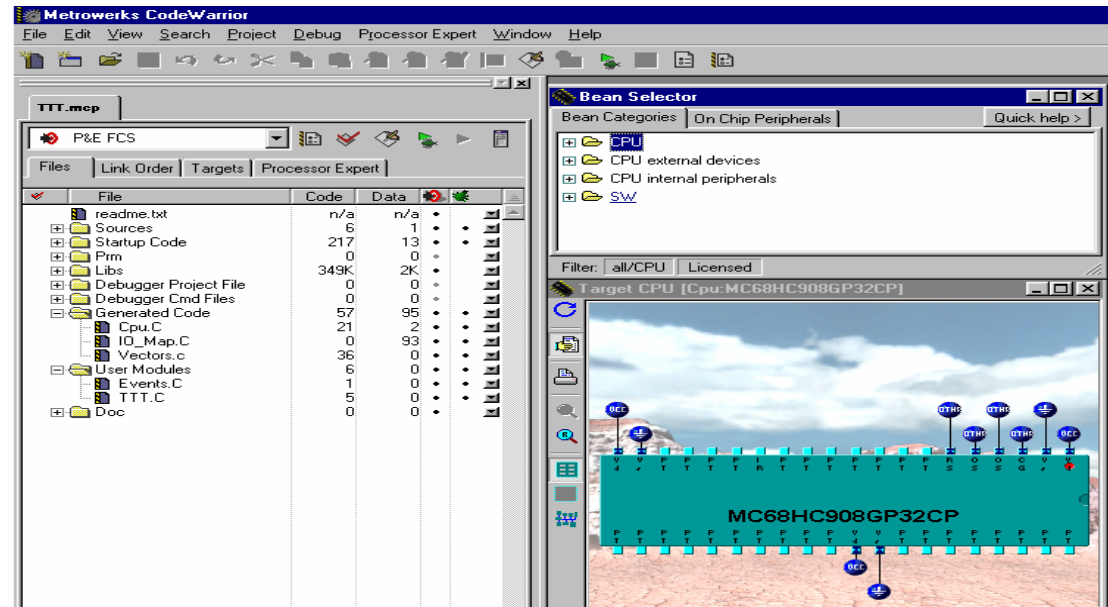
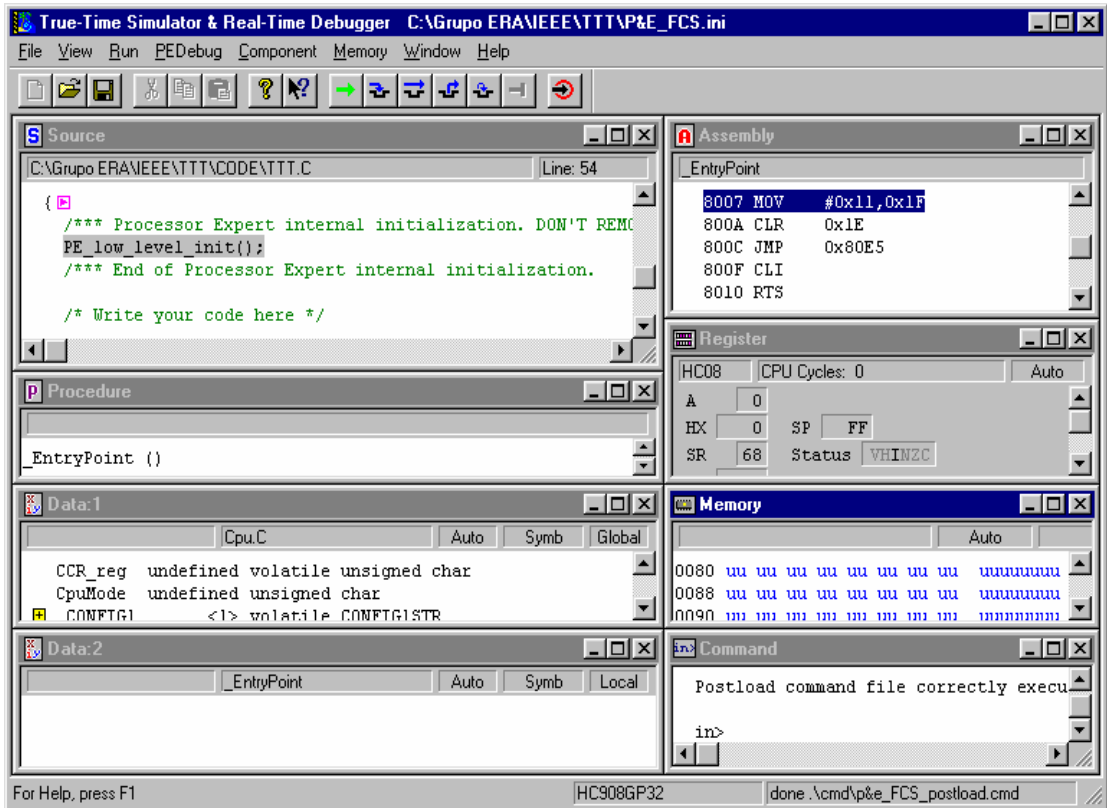


Figura 10



Las carpetas generadas Generate Code y User Modules son necesarias para programar en lenguaje C.

En la carpeta Generate Code aparecen las librerías que usted adicione como lo son los Beans y programas ya desarrollados como la librería matemática, CPU, Vectors y otras.

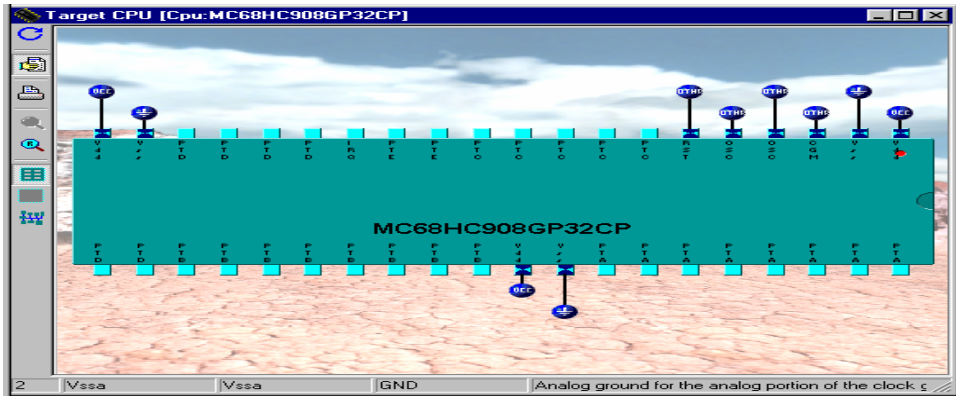
En User Modules aparece el programa que usted esta implementando para este caso TTT.C, para implementar el algoritmo este debe ser escrito en el espacio que señala write your code here,

La ventana de programacion posee tres sub ventanas en las cuales se encuentra Tarjet CPU, BEAN Selector y TTT.mcp.

Tarjet CPU:

En esta ventana aparece el empaquetado del microcontrolador y los pines seleccionados para aplicación y alimentación (Figura 11).

Figura 11



BEAN Selector:

En esta ventana se pueden seleccionar las aplicaciones de cada uno de los pines Figura 12

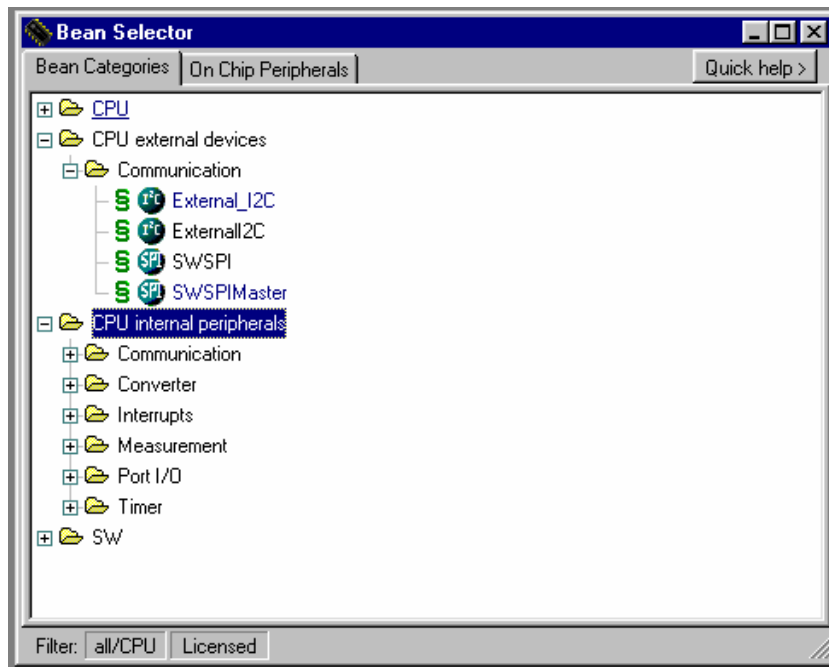


Figura12

CPU:

Contiene la información que define el microcontrolador y el empaquetado del mismo

CPU external devices:

Contiene los BEANS y la información sobre la comunicación que realiza el microcontrolador utilizando protocolo I2C y comunicación sincrónica serial SWSPI

CPU internal Peripherals:

Contiene las carpetas:

Comunicación:

Contiene el BEAN para realizar comunicación sincrónica

Converter:

Contiene los BEANS para realizar conversión análoga a digital

Interrupts:

Contiene los BEANS para realizar interrupciones

Measurement:

Contiene los BEANS para realizar la función de captura del Timer

Port I/O:

Contiene los BEANS para realizar la función de bits de entrada y salida

Si usted elige PortI/O puede elegir entre un bit o un grupo de bits de un puerto.

Timer:

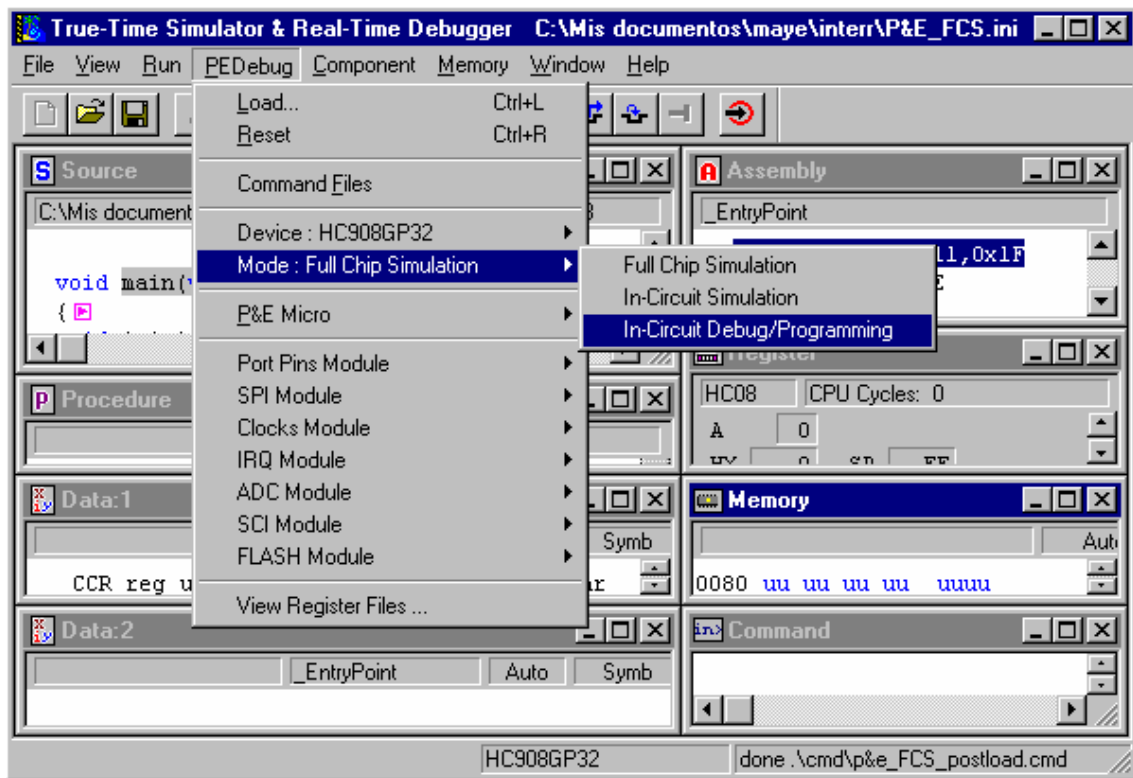
Contiene los BEANS para realizar las funciones del TIMER

- **COMO CARGAR EL PROGRAMA AL MICROCONTROLADOR**

Para cargar el algoritmo generado, este no debe presentar ningún error, para analizar que errores presenta se señala make (la imagen de una mano escribiendo) o F7 la cual le indica que errores posee el algoritmo.

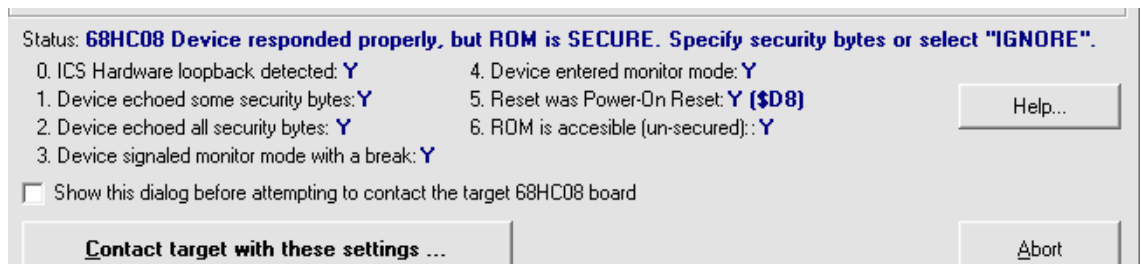
Cuando el programa no presenta fallas se señala debug (la flecha verde) o F5. el cual abre la ventana de True Time Simulator & Real Time Debugger en esta ventana se señala PEDebug, mode: Full Chip Simulation, In Circuit Debug/ Programming como aparece en al Figura 13

Figura 13



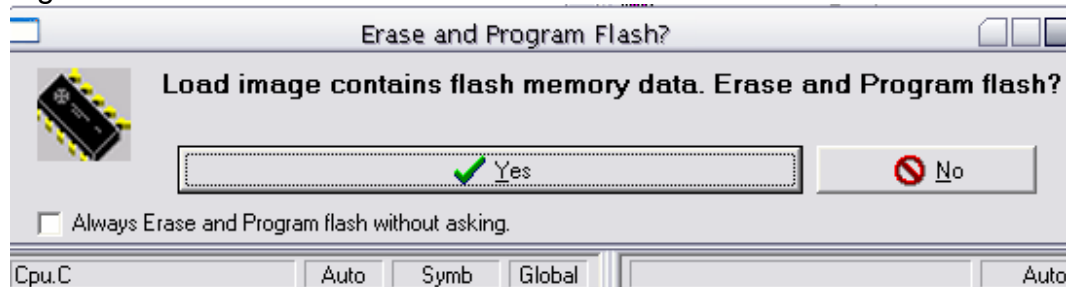
Cuando se selecciona In Circuit Debug/ Programming se genera la pantalla de programación para la cual si se esta utilizando la TPM PROGRAMADOR MC68HC908GP32 E3T se debe seleccionar en Target Hardware Type la opción Class III e IGNORE security después de esto debe seleccionar contact target.

Ahora bien, si el modo monitor es correcto los estados mostrados en la ventana tiene que estar con una Y azul como aparece en la Figura 14.



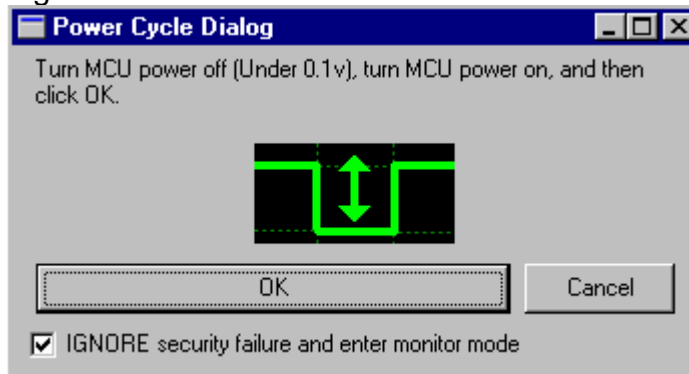
Una vez cumplidas estas condiciones, se pulsa Contact target with these settings, con lo cual se abre la ultima ventana que en su primera opción nos pregunta si queremos hacer un borrado de memoria (Figura 15)

Figura 15



para el proceso de programación se hace necesario seleccionar yes después de esto se genera la ventana Power Cycle Dialog como se muestra en la Figura 16 en la cual se selecciona ignore security failure, después se apaga y se prende la tarjeta TPM y se presiona OK.

Figura 16



El anterior proceso se repite varias veces hasta que la pantalla de Power Cycle Dialog deje de aparecer, después de programar varias veces el microcontrolador el proceso se hace mas corto y no se requiere prender y apagar la tarjeta mas de dos veces.

COMO DEFINIR LOS PINES Y LOS PUERTOS

para definir los puertos se utiliza el comando DDRX donde X es el puerto seleccionado, los pines del puerto seleccionados como salidas son definidos con un 1 y los de entrada con un cero, por ejemplo

DDRA=15; esto define los primeros 4 pines del puerto A como salidas y los siguientes cuatro como entradas.

COMO DEFINIR LOS REGISTROS DE UN PUERTO

Después de definir el puerto se pueden definir los registros de este utilizando el comando PTX donde X es el nombre del puerto a utilizar, por ejemplo:

PTA=variable;

Tambien se pueden definir cada uno de los pines del puerto ejemplo
PTA_PTA0=1;

COMO DEFINIR LAS RESISTENCIAS DE PULL-UP DE UN PUERTO

Para esta aplicación cerciórese que el puerto tenga implementadas las resistencias de pull-up, revise el datasheet del microcontrolador donde aparecen todas las especificaciones técnicas, una vez echo esto utilice el comando PTXPUE donde X es el puerto, las resistencias se habilitan si el valor es un uno, ejemplo:

PTAPUE=15;

En este caso se habilitaron las resistencias de pull-up de los cuatro primeros pines del puerto A.

COMO UTILIZAR LOS CONVERTORES ANALÓGICO DIGITALES

Para utilizar el conversor analógico digital se requiere primero configurarlo, como se explica a continuación

ADCLK=0x50; selecciona ADICLK en 1 con lo cual se selecciona el bus del reloj , selecciona ADIV1 en 1 con lo cual la entrada del reloj se divide en 4 (ver pag 96 technical Data MC68HC908GP32)

ADSCR=0x1F; pone ADCH0,ADCH1,ADCH2,ADCH3,ADCH4 en 1 con lo cual se apaga el el conversor(ver pag 94 technical Data MC68HC908GP32)

ADSCR=0x20; pone ADCO en 1 con lo cual se pone el conversor en conversión continua (ver pag 92 technical Data MC68HC908GP32)

Valor=ADR; se define una variable para el vector de registro ADR donde se guardan los datos analógicos que están convirtiéndose a valores digitales (ver pag 95 technical Data MC68HC908GP32)

COMO UTILIZAR EL PWM

Para utilizarlo el modulo PWM se requiere primero configurarlo, como se muestra a continuación

```
T1SC = 48; (ver pag 361 technical Data MC68HC908GP32 )          /*
MS0A=1,TOV0=1 */
T1SC0 = 18; /* inicia modo PWM MS0A=1,ELS0B=1,ELS0A=1,TOV0=1 */
T1SC0 = 30;
T1CH0H = (byte)(30000 >> 8); /* se define valor del ciclo útil del PWM*/
T1MODH = (byte)(60000 >> 8); /* se define el valor del periodo */
T1CH0L = (byte)30000;
T1MODL = (byte)60000;
```

ANEXO D PROGRAMAS IMPLEMENTADOS EN CODEWARRIOR

1 PROGRAMA IMPLEMENTADO EN CODEWARRIOR 3.0 PARA USO DE UNA PANTALLA LCD

El programa se implemento como una librería llamada “lcd4datos” la cual se puede llamar desde la ventana donde el usuario escribe el algoritmo para esto debe incluirla dentro de las librerías y llamarla desde el programa como un archivo de extensión .h

programa: lcd4datos.c

```
void inicio_lcd(void)
{
void comando(byte dato);
    void comando2(byte dato);
    void retardo(byte tiempo);
    DDRC=7;
    DDRD=15;

    retardo(15); //retardo de 15mseg
    comando(3); //function set interface 8 bits 3 nibble alto
    retardo(4); //retardo de 4.1mseg
    comando(3); //function set interface 8 bits 3 nibble alto
    retardo(1); //esperar por mas de .1mseg
    comando2(3); //function set interface 8 bits 3 nibble alto
    comando2(2); //function set interface 4 bits 3 nibble alto
    comando2(2); //function set interface 4 bits 3 nibble alto
    comando2(8); // se especifica lineas y fuentes de la ICD
    comando2(0); //display off 0 nibble alto
    comando2(12); //display off 12 nibble bajo
    comando2(0); //display clear 0 nibble alto
    comando2(1); //display clear 1 nibble bajo
    retardo(1); //esperar por mas de .1mseg
    comando2(0); //entry mode set 0 nibble alto
    comando2(6); //entry mode set 6 nibble bajo

}

//Comando de Control 8 bits
void comando(byte dato)
```

```

{
    PTC=0;
    PTD=dato;
    PTC=4;
    PTC=0;
}

//Comando de Control 4 bits
void comando2(byte dato)
{
    PTC=0;
    PTD=dato;
    PTC=4;
    PTC=0;
    asm
    {
        LDA #13
        L2: DECA
        BNE L2
    }
}

//Arreglo de los datos a enviar
void datos(byte letra)
{
    void enviar_dato(byte dato);

    byte nible_H, nible_L;
    nible_H = letra & 0xF0;
    nible_H = nible_H >> 4;
    nible_L = letra & 0x0F;
    enviar_dato(nible_H);
    enviar_dato(nible_L);
}

//Envia el nible
void enviar_dato(byte dato)
{
    PTC_PTC0=1;
    PTD=dato;
    PTC_PTC2=1;
    PTC_PTC2=0;
    asm
    {
        LDA #13

```

```

                L2:  DECA
                   BNE  L2
            }
    }

void retardo(byte tiempo)
{
    void unmsseg(byte tiempo);

    int y;
    for(y=0;y<=tiempo;y++)
    {
        unmsseg(1);
    }
}

/* Retardo de 1 milisegundo */

void unmsseg(byte tiempo)
{
    int a,b;
    for(b=0;b<12;b++)
    {
        for(a=0;a<=tiempo;a++);
    }
}

```

Este programa permite al usuario representar caracteres numéricos en una LCD utilizando dos comandos, comando2 permite la selección de filas y columnas lo cual se hace llamando la primera fila con un 8 y la 2 fila con un 12 ejemplo: comando2(8).

Para la visualización de datos se utiliza el comando datos() que permite al usuario poner el símbolo alfanumérico dentro del paréntesis.

Archivo de extensión .h
Lcd4datos.h

```

#ifndef __lcd4datos_H
#define __lcd4datos_H

```

```

/*Include shared modules, which are used for whole project*/
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "Cpu.h"

void inicio_lcd(void);
void comando(byte dato);
void comando2(byte dato);
void retardo(byte tiempo);
void datos(byte letra);
void enviar_dato(byte letra);
void unmsseg (byte tiempo);

```

2 PROGRAMA IMPLEMENTADO EN CODEWARRIOR 3.0 PARA USO DE UN TECLADO

El programa se implemento como una librería llamada “teclado4x4” la cual se puede llamar desde la ventana donde el usuario escribe el algoritmo para esto debe incluirla dentro de las librerías y llamarla desde el programa como un archivo de extensión .h

teclado4x4.c

```

void teclado4x4 (void)
{
int tecla0,tecla1,tecla2,tecla3,tecla4,tecla5,tecla6,tecla7,tecla8,
tecla9,teclaA,teclaB,teclaC,teclaD,teclaE,teclaF,i;

PTAPUE=240;
DDRA=15;

tecla0=0;
tecla1=0;
tecla2=0;
tecla3=0;
tecla4=0;
tecla5=0;
tecla6=0;
tecla7=0;
tecla8=0;
tecla9=0;

```

```

teclaA=0;
teclaB=0;
teclaC=0;
teclaD=0;
if(PTA_PTA4&PTA_PTA5&PTA_PTA6&PTA_PTA7==1)
{
  for(i=0;i<100;i++)
  {
    PTA =14;
    if (PTA_PTA4==0)
      teclaF=1;
    if (PTA_PTA5==0)
      teclaB=1;
    if (PTA_PTA6==0)
      tecla7=1;
    if (PTA_PTA7==0)
      tecla3=1;
  }
}
if(PTA_PTA4&PTA_PTA5&PTA_PTA6&PTA_PTA7==1)
{
  for(i=0;i<100;i++)
  {
    PTA =13;
    if (PTA_PTA4==0)
      teclaE=1;
    if (PTA_PTA5==0)
      teclaA=1;
    if (PTA_PTA6==0)
      tecla6=1;
    if (PTA_PTA7==0)
      tecla2=1;
  }
}
if(PTA_PTA4&PTA_PTA5&PTA_PTA6&PTA_PTA7==1)
{
  for(i=0;i<100;i++)
  {
    PTA =11;
    if (PTA_PTA4==0)
      teclaD=1;
    if (PTA_PTA5==0)
      tecla9=1;
    if (PTA_PTA6==0)
      tecla5=1;
  }
}

```

```

        if (PTA_PTA7==0)
            tecla1=1;
    }
    if(PTA_PTA4&PTA_PTA5&PTA_PTA6&PTA_PTA7==1)
    {
    for(i=0;i<100;i++)
    {
        PTA =7;
        if (PTA_PTA4==0)
            teclaC=1;
        if (PTA_PTA5==0)
            tecla8=1;
        if (PTA_PTA6==0)
            tecla4=1;
        if (PTA_PTA7==0)
            tecla0=1;

    }
    }
    }
    }
    }
    if(tecla0==1)
    { //coloque la funcion que desea para la tecla 0//
    //datos('0');
    }
    if(tecla1==1)
    { //coloque la funcion que desea para la tecla 1//
    //datos('1');
    }
    if(tecla2==1)
    { //coloque la funcion que desea para la tecla 2//
    //datos('2');
    }
    if(tecla3==1)
    { //coloque la funcion que desea para la tecla 3//
    //datos('3');
    }
    if(tecla4==1)
    { //coloque la funcion que desea para la tecla 4//
    //datos('4');
    }
    if(tecla5==1)
    { //coloque la funcion que desea para la tecla 5//

```

```

//datos('5');
}
if(tecla6==1)
{//coloque la funcion que desea para la tecla 6//
//datos('6');
}
if(tecla7==1)
{//coloque la funcion que desea para la tecla 7//
//datos('7');
}
if(tecla8==1)
{//coloque la funcion que desea para la tecla 8//
//datos('8');
}
if(tecla9==1)
{//coloque la funcion que desea para la tecla 9//
//datos('9');
}
if(teclaA==1)
{//coloque la funcion que desea para la tecla A//
//datos('A');
}
if(teclaB==1)
{//coloque la funcion que desea para la tecla B//
//datos('B');
}
if(teclaC==1)
{//coloque la funcion que desea para la tecla C//
//datos('C');
}
if(teclaD==1)
{//coloque la funcion que desea para la tecla D//
//datos('D');
}
if(teclaE==1)
{//coloque la funcion que desea para la tecla E//
//datos('E');
}
if(teclaF==1)
{//coloque la funcion que desea para la tecla F//
//datos('F');
}
}
}

```

Este programa permite al usuario utilizar un teclado y definir las funciones de cada tecla dentro de la librería según lo considere el usuario. Para utilizar esta librería se necesita el archivo de extensión .h, llamado teclado4x4.h

```
teclado4x4.h
#ifndef __teclado4x4_H
#define __teclado4x4_H

/*Include shared modules, which are used for whole project*/
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "Cpu.h"

void teclado4x4(void);
```