**CAPÍTULO 2**

**2 FUNDAMENTACIÓN TEÓRICA DE LOS RECURSOS UTILIZADOS**

**Requerimientos para la aplicación del proyecto**

Para la realización de este proyecto utilizaremos los siguientes implementos y/o herramientas:

**- HARDWARE:**

* Microcontrolador PIC 18F4520 de MICROCHIP
* Pantalla de cristal líquido LCD 16 caracteres x 2 líneas
* Temporizador NE555
* Comparador LM311
* Resistencias de distintos valores de acuerdo al circuito implementado
* Capacitancias de distintos valores de acuerdo al circuito implementado
* Inductancias de distintos valores de acuerdo al circuito implementado
* Relés de 5V para proceso de autocalibración y conmutación de modos
* Programador Pickit 3 de Microchip

- **SOFTWARE:**

* Compilador en lenguaje C PCW CCS de CUSTOM COMPUTER SERVICES
* Simulador de circuitos PROTEUS de LAB-CENTER ELECTRONICS
* Programador en lenguaje ensamblador MPLAB de Microchip

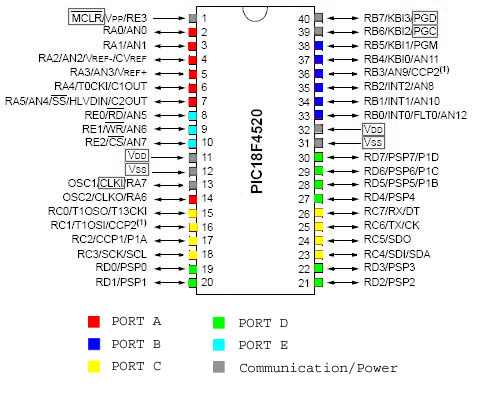
**2.1 HARDWARE**

**2.1.1 MICROCONTROLADOR PIC 18F4520 DE MICROCHIP**

Este dispositivo pertenece a la familia de microcontroladores avanzados PIC18FXXXX, los cuales tienen un alto rendimiento computacional a un costo asequible.

Es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que conforman un controlador. Pertenece a la gama mejorada, poseen una arquitectura abierta pudiéndose expansionar en el exterior al poder sacar los buses de datos, direcciones y control.

Es un computador dedicado. En su memoria reside un programa destinado a gobernar una aplicación determinada, una vez programado y configurado el microcontrolador solamente sirve para gobernar dicha tarea.



**Figura 1: Microcontrolador PIC18F4520**

**CARACTERÍSTICAS PRINCIPALES**

- Tecnología FLASH/EEPROM de baja potencia y alta velocidad.

- Conversor A/D de 10 bits

- Tres sincronizadores (Timers)

- Dos módulos para captura/comparación/PWM

- Módulo Serial Maestro Sincrónico (MSSP) con dos modos de operación

- Módulo USART direccionable, soporta RS485 y RS232

- Detector de bajo voltaje programable

- POR (Power On Reset), PWRT (Power Up Timer), OST (Oscillator Start-up Timer)

- WDR (Watchdog Timer) con oscilador RC incorporado y período programable desde 4 hasta 131 ms

- Protección de código programable

- Modo de ahorro de energía

- Bajo consumo de potencia (menos de 1.6 mA con 5V y 4MHz)

- Tres fuentes externas de interrupción

- Cuatro fuentes de interrupción por cambios de entradas

- Acepta cuatro tipos de osciladores de cristal (hasta 40MHz)

- Acepta dos tipos de osciladores RC externos (hasta 4MHz)

- Acepta dos tipos de relojes externos (hasta 40MHz)

- Amplio rango de voltaje de operación (2V a 5.5V)

**2.1.2 PANTALLA DE CRISTAL LIQUIDO 16X2**

Las pantallas de cristal líquido tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica.

La pantalla consta de una matriz de caracteres distribuidos en dos líneas de 16 caracteres cada línea.

El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.

El módulo LCD ejecuta automáticamente una secuencia de inicio interna en el instante de aplicarle la tensión de alimentación si se cumplen los requisitos de alimentación.

La secuencia de inicio ejecutada es la siguiente:

* CLEAR DISPLAY: Borra el módulo LCD y coloca el cursor en la primera posición (dirección 0)



* FUNCTION SET: Establece el tamaño de interfaz con el bus de datos, número de líneas de la pantalla y tipo de carácter



* DISPLAY ON/OFF CONTROL: Activa o desactiva poniendo en ON/OFF tanto la pantalla como al cursor y se establece si este último debe o no parpadear



* ENTRY MODE SET: Establece la dirección de movimiento del cursor y especifica si la visualización se va desplazando a la siguiente posición de la pantalla o no



Si no se satisfacen las condiciones de alimentación, la secuencia de inicialización habría que realizarla por software, donde las instrucciones que aplica el usuario podrían ser las expuestas anteriormente o cualquier otra según sus propias necesidades.

Es importante que la primera instrucción que se envíe realice una espera de unos 15ms o mayor para la completa reinicialización interna del módulo LCD.



**Figura 2: Pantalla de cristal líquido 16x2**

**CARACTERÍSTICAS PRINCIPALES**

- Interfaz para aplicaciones con microcontroladores, permite visualizar 32 caracteres alfanuméricos en dos líneas de 16 caracteres

- Opciones de alimentación: 5V, 12V o dual

- Consumo reducido, del orden de 7.5mW

- Protecciones contra inversión de polaridad y picos de tensión

- Cuenta con un microcontrolador PIC interno que le provee de la librería con todos los comandos para su uso

- Se comanda con una línea Rx. Los datos y comandos se reciben a 9600 bps, 8 bits de datos y sin paridad

- Posee incorporado una tabla completa de caracteres ASCII y además acepta códigos especiales de comando como: borrado de pantalla, cambio de línea, encendido de backlight

- Desplazamiento de los caracteres hacia la izquierda o la derecha

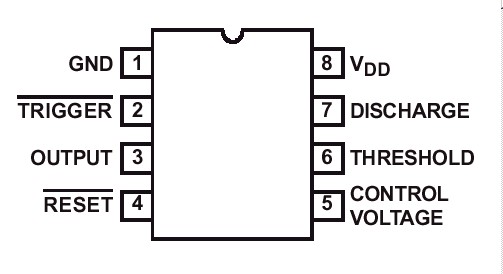
- Movimiento del cursor y cambio de su aspecto

**2.1.3 TEMPORIZADOR 7555**

Es un temporizador CMOS proveyendo una significativa mejora sobre el estándar NE555, siendo al mismo tiempo un reemplazo directo para aquellos dispositivos en la mayoría de las aplicaciones.

Es un económico y versátil circuito integrado temporizador de 8 pines de bajo consumo al estar construido con tecnología CMOS.

Es sencillo de utilizar y requiere de sólo unos pocos componentes adicionales para realizar útiles tareas, no solo relacionadas con la temporización tales como osciladores astables, generadores de rampas y temporizadores secuenciales. A pesar de ser económico se consiguen temporizaciones estables frente a variaciones de tensión de alimentación y de temperatura.

****

**Figura 3: Temporizador 7555**

**CARACTERÍSTICAS PRINCIPALES**

- Equivalente exacto en la mayoría de las aplicaciones del temporizador NE555

- Bajo consumo de corriente (80uA)

- Corrientes de disparo, threshold y reset extremadamente bajas (20pA)

- Operación de alta velocidad (500KHz)

- Operación en amplio rango de voltajes (3V a 16V)

- Puede usarse con altas impedancias para mayores constantes de tiempo

- Temporizaciones desde microsegundos hasta horas

- Opera en modos astable y monoestable

- Duty cycle ajustable

- Estabilidad en frecuencia de 0.005% por ºC.

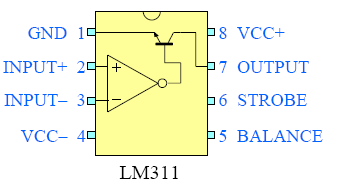
- Aplicaciones:

* Temporización precisa
* Generación de pulsos
* Temporización secuencial
* Generación de retardos
* Modulación PWM

**2.1.4 COMPARADOR LM311**

Es un comparador estándar muy versátil, tiene una salida de colector abierto para poder ajustar la tensión de salida por lo que requiere una resistencia de pull up en la salida. La pata GND se puede llevar a cualquier valor entre las alimentaciones y puede atacar cargas conectadas a GND, VCC positivo o negativo.

Su máxima corriente de salida es de 50 mA por lo que puede fácilmente excitar una salida de relé, también posee una terminal de habilitación o strobe, se alimenta desde los 5 a los 36 voltios y su tiempo de conmutación es de 200 ns a un consumo típico de 5 mA.



**Figura 4: Comparador LM311**

Entre las aplicaciones del LM311 fabricado por National Semiconductor están:

* Habilitador de cargas conectadas a tierra
* Interfaz entre cualquier voltaje y ttl
* Generador de onda cuadrada (usado en este proyecto)
* Habilitador directo de relés

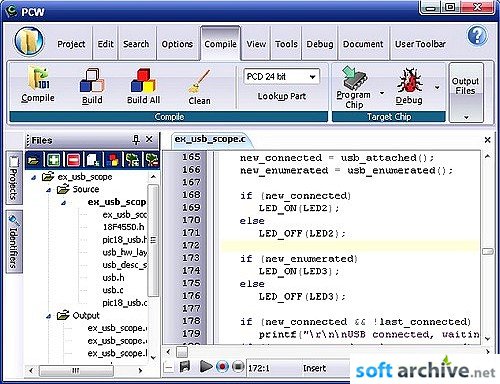
**2.2 SOFTWARE**

**2.2.1 PROGRAMADOR EN LENGUAJE C PCW CCS**

Herramienta para desarrollo de aplicaciones embebidas en microcontroladores PIC. Nos genera ficheros en formato hexadecimal, que es el necesario para programar (utilizando un programador de PIC) un microcontrolador.

Trabaja en un entorno de desarrollo integrado (IDE), que nos permite desarrollar todas y cada una de las fases del programa, desde la edición hasta la compilación, pasando por la depuración de errores.

Este software traduce el código C del archivo fuente (.C) a lenguaje máquina para los microcontroladores PIC, generando así un archivo en formato hexadecimal (.HEX).



**Figura 5: Ambiente del programador PCW de CCS**

**CARACTERÍSTICAS PRINCIPALES**

**-** Generación de código máquina compacto y eficiente.

- Integración perfecta con MPLAB y otros simuladores como PROTEUS para el proceso de depuración.

- Inclusión de biblioteca muy completa de funciones precompiladas para el acceso al hardware de los dispositivos (entrada/salida, temporizadores, conversor A/D, transmisión RS232).

- Incorporación de drivers para dispositivos externos como pantallas LCD, teclados numéricos, memorias EEPROM, relojes en tiempo real.

- Permite integrar partes de código directamente en ensamblador, manteniendo otras partes del programa en C.

* **Utilidades adicionales**

El entorno PCW incluye, además del IDE y del compilador, una serie de utilidades adicionales con las que se amplían las posibilidades de éste, a continuación algunas de ellas:

- Monitor del puerto serie: Terminal que monitorea la entrada y salida del puerto serie del computador.

- Selección de dispositivos: Consta de una base de datos con los dispositivos que puede programar el compilador, incluyendo todas sus características de hardware, de manera que se puede emplear para buscar aquellos dispositivos que cumplan una serie de propiedades comunes.

- Editor de dispositivos: Emplea la base de datos de dispositivos, y permite editar los componentes para modificar sus características hardware, así como añadir nuevos dispositivos o eliminar algunos de ellos.

- Conversor numérico: Realiza conversiones entre los tipos de datos unsigned, signed, hex y float.

- Extracción de datos de calibrado: Permite leer los datos de calibración existentes en la memoria de programa de un determinado dispositivo. Estos datos contienen información particular de cada microcontrolador, y se refieren a posibles problemas especiales que pudieran haber tenido lugar durante el desarrollo y fabricación. Mediante esta opción es posible leer estos datos y grabarlos en un fichero .H o .C que incorporará una directiva # ROM para dicho dispositivo, con lo que cada vez que se programe el microcontrolador se incluirán estos datos de calibrado.

- Desensamblador: Lee un fichero en código máquina y lo traduce a su equivalente en Ensamblador, con lo que se podría insertar este código en un programa en C, mediante las directivas # ASM y # ENDASM.

- Inserción de código ensamblador: directamente en el código C, con lo que determinados procedimientos se pueden implementar directamente en Ensamblador con el ahorro de código y tiempo de ejecución que ello implica.

- Gestión automática de páginas de código: de manera transparente al usuario, insertará las instrucciones necesarias para el cambio de página de código.

- Gestión automática de bancos de memoria: y minimización del cambio, intentando agrupar las variables locales utilizadas en cada función dentro de un mismo banco.

- Mapeo de la memoria del microcontrolador desde el código C: evitando que el compilador asigne automáticamente variables a zonas de memoria que interesa mantener libres.

- Almacenamiento de constantes en memoria de programa: en lugar de la memoria de datos.

- Soporte de punteros y referencias: permitiendo el uso de punteros de 8 y 16 bits, permite el paso de parámetros por referencia de variables mejorando el tiempo de legibilidad del código.

- Eficiente implementación de variables y funciones: permitiendo que éstas sean asignadas a zonas de memoria donde se emplee la menor memoria posible de forma global reutilizando para ello posiciones de memoria.

- Generación del código estrictamente necesario: incluyendo sólo funciones de las bibliotecas precompiladas que son utilizadas en el programa.

**2.2.2 Simulador de circuitos PROTEUS**

Los paquetes de simulación electrónica no suelen incluir modelos de simulación de las diferentes familias de microcontroladores que existen en el mercado, lo que supone una barrera para los diseñadores electrónicos.

Esta limitación es superada por el entorno de desarrollo PROTEUS VSM, que trata al microcontrolador como un componente más del circuito a desarrollar., ofreciendo la posibilidad de simular código microcontrolador de alto y bajo nivel. Esto permite el diseño tanto a nivel de hardware como software y realizar la simulación en un mismo y único entorno

Para ello se suministran tres potentes entornos como son:

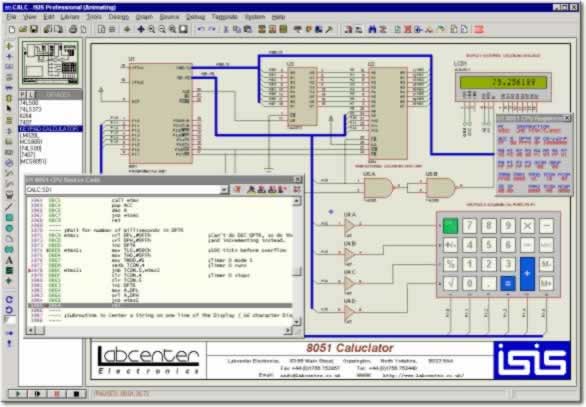
- ISIS que es el módulo de captura de esquemas,

- VSM (Virtual System Modeling) que es el módulo de simulación incluyendo PROSPICE y

- ARES el cual es módulo para la realización de circuitos impresos (PCB).

La característica más importante de PROTEUS VSM es la capacidad de depurar programas fuente de distintos lenguajes de programación. Además de aceptar el archivo de programación .HEX, también admite ficheros D39, COD, COF y BAS. Al utilizar estos archivos se puede abrir una ventana de código fuente llamada SOURCECODE mediante la cual se puede seguir el programa fuente línea a línea de código.

* **ISIS**
* Permite realizar esquemas que pueden ser simulados en el entorno VSM o pasados a un circuito impreso en el entorno ARES
* Posee una buena colección de librerías de modelos para dibujar, simular o para las placas; además permite la creación de nuevos componentes, su modelización para su simulación e incluso la posibilidad de solicitar al fabricante (Lab-center Electronics) que cree un nuevo modelo
* **VSM**
* Simulador para esquemas electrónicos que contienen microprocesador
* El núcleo de VSM es ProSPICE, que combina un núcleo de simulación analógica usando el estándar SPICE3F5, con modelos animados de los componentes electrónicos, tanto si el programa se ha escrito en ensamblador como si se ha utilizado un lenguaje de alto nivel, permitiendo interactuar con nuestro diseño
* Permite incluir en la simulación de circuitos el comportamiento completo de los microcontroladores más conocidos del mercado. PROTEUS es capaz de leer los ficheros con el código ensamblado para los microprocesadores de las familias PIC, AVR, 8051, HC11, ARM/LPC200 y BASIC STAMP y simular perfectamente su comportamiento. Incluso puede ver su propio código interactuar en tiempo real con su propio hardware pudiendo usar modelos de periféricos animados tales como pantallas LED o LCD, teclados, terminales RS232, simuladores de protocolos I2C, etc. Proteus es capaz de trabajar con los principales compiladores y ensambladores del mercado.
* **ARES**
* Herramienta para la elaboración de placas de circuito impreso con posicionador automático de elementos y generación automática de pistas, que permite el uso de hasta 16 capas.
* Con ARES el trabajo duro de la realización de placas electrónicas recae sobre el PC en lugar de el diseñador.



**Figura 6: Ambiente del simulador PROTEUS de LAB-CENTERS ELECTRONICS**

**CARACTERÍSTICAS PRINCIPALES**

* Entorno de diseño gráfico de esquemas electrónicos (ISIS) extremadamente fácil de utilizar y dotado de herramientas para facilitar el trabajo.
* Entorno de simulación prospice mixto entre el estándar SPICE3F5 y la tecnología exclusiva de modelación de sistemas virtuales (VSM).
* Entorno de diseño de placas de circuito impreso (ARES) de altas prestaciones con bases de datos de 32 bits, posicionador automático de elementos y generación automática de pistas con tecnologías de autocorte y regeneración.
* La lista de redes (NetList) son compatibles con la mayoría de los programas de realización de PCB.
* Se puede visualizar la RAM (Registros especiales y datos), además de la EEPROM y la memoria de programa. Se pueden establecer puntos de detención para la depuración.
* Posee una variada instrumentación virtual que nos facilita el análisis de circuitos. Estos dispositivos se pueden insertar en los circuitos, mostrando las medidas de tiempo real según se simula.