



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería Eléctrica y Computación**

MEDIDOR LC UTILIZANDO PANTALLA LCD 2X16 PARA VISUALIZACIÓN CON  
PROGRAMA EMBEBIDO EN UN MICROCONTROLADOR

**TESINA DE SEMINARIO**

Previo a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA ESPECIALIZACIÓN  
TELECOMUNICACIONES**

Presentado por:

**Adrián Eduardo Arroyo Zambrano**  
**Héctor Felipe Cabrera Ulloa**

GUAYAQUIL – ECUADOR

2010

## **AGRADECIMIENTO**

Primero que nada a Dios por poner los eventos en su lugar al momento preciso y por darme siempre no lo que pido sino lo que verdaderamente he necesitado.

A mi esposa Jacqueline Brito Funes quien siempre ha sido el motor de mi superación por todo el apoyo que recibí durante el proceso de la obtención de mi título.

A mis padres Hugo e Inés y a mis hermanos por ser parte importante en mi vida e inculcarme los valores que aplicaré en mi vida profesional.

A los compañeros y profesores que nos dieron la oportunidad y los conocimientos para estar hoy aquí.

*Adrián Arroyo Zambrano*

## **AGRADECIMIENTO**

A Dios sobre todo, quien guía nuestras vidas y nos da fortaleza y sabiduría para seguir día a día; a mi esposa, mis padres y hermanos, quienes con su apoyo espiritual supieron guiarme al final de esta carrera politécnica.

A mis maestros, por el conocimiento, soporte y amistad impartidos a lo largo de todos estos años de estudio.

A mis amigos y compañeros tanto de la ESPOL como de afuera, quienes fueron un soporte con su ayuda y conocimientos a lo largo de esta carrera.

*Héctor Cabrera Ulloa*

## **DEDICATORIA**

A mi esposa Jacqueline sin la cual no finalizaría mi carrera por todas las noches en vela y por todo el amor y apoyo.

A mis padres Hugo e Inés quienes con sus lecciones me enseñaron todo lo que debía saber para llevar hasta el final mi esfuerzo y ser un buen profesional.

A mi abuelo Adriano (+) por enseñarme del trabajo duro y constante que hace al hombre respetable.

*Adrián Arroyo Zambrano*

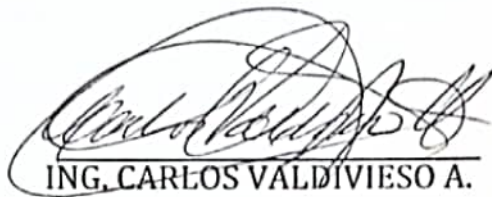
## **DEDICATORIA**

A mi esposa Andrea, a mis padres Héctor y Rebeca, quienes han sido mi apoyo constante.

A mis hermanos Ivo e Isabel, a mi sobrino David, quienes con su apoyo y alegría fueron inspiración para la finalización de la carrera.

*Héctor Cabrera Ulloa*

**TRIBUNAL DE SUSTENTACIÓN**



ING. CARLOS VALDIVIESO A.

**PROF. DEL SEMINARIO DE GRADUACIÓN**



ING. HUGO VILLAVICENCIO V.

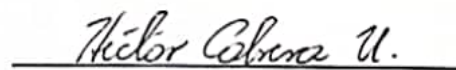
**DELEGADO DEL DECANO**

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)

  
Adrián Arroyo Zambrano

  
Héctor Cabrera Ulloa

## **RESUMEN**

La finalidad del proyecto es desarrollar un medidor de capacitancia e inductancia de rango comercial capaz de tener una precisión igual o menor al 10% que es la precisión de la mayoría de los medidores actuales del mercado.

Se debe presentar el valor medido en la escala correspondiente al valor y presentarla a través de una pantalla gráfica LCD de 16 caracteres por 2 líneas y hacer esto de modo automático sin ajustar el rango de forma manual, el único ajuste manual se realizará para conmutar entre los modos de medición de capacitancia e inductancia.

La principal aplicación del proyecto es la medición de componentes electrónicos pasivos (capacitores e inductores), y aunque es cierto que estos medidores existen en la actualidad su falta de disponibilidad y costo, en algunos casos elevado, lo aleja del poder adquisitivo de gran parte de los radiotécnicos y estudiantes de ingeniería.

En orden de poder realizar nuestro proyecto vamos a servirnos de los siguientes componentes: primero del corazón del proyecto el microcontrolador de la serie 18F de Microchip el 18F4520 que será el encargado de realizar las mediciones, cálculos y presentación de resultado de las mismas, el comparador LM311 de



National Semiconductors y el oscilador NE555 de Fairchild, así como de las herramientas de simulación Proteus Isis y Ares de Labcenter Electronics y PICC de CCS el compilador del proyecto, y de la herramienta de programación Pickit 3 y su software MPLAB ambas pertenecientes a Microchip.

El programa del microcontrolador se encargará de calcular la frecuencia de oscilación del circuito tanque, el período del circuito astable y de los cálculos de punto flotante y presentación de los mismos así como del proceso de autocalibración del circuito.

Se realizarán las correspondientes simulaciones de cada una de las partes así como del circuito ensamblado dentro del ambiente de simulación de Proteus ISIS para compararlas con los resultados teóricos.

Se realizarán las pruebas del circuito armado para determinar la precisión del mismo y se lo comparará contra modelos comerciales para verificar su precisión contra estos y contra los valores marcados en el capacitor; para tal efecto se utilizarán capacitores de precisión de 1%, también se medirá su respuesta con los capacitores electrolíticos más imprecisos.

## **ÍNDICE GENERAL**

<b>AGRADECIMIENTO .....</b>	<b>II</b>
<b>DEDICATORIA .....</b>	<b>IV</b>
<b>RESUMEN .....</b>	<b>VIII</b>
<b>INTRODUCCIÓN .....</b>	<b>XIV</b>
<b>CAPÍTULO 1 .....</b>	<b>1</b>
<b>1. DESCRIPCIÓN GENERAL DEL PROYECTO .....</b>	<b>1</b>
1.1 Antecedentes .....	1
1.2 Justificación .....	2
1.3 Descripción del proyecto .....	2
1.4 Aplicaciones .....	6
1.5 Funcionamiento .....	7
<b>CAPÍTULO 2 .....</b>	<b>9</b>
<b>2. FUNDAMENTACION TEÓRICA DE RECURSOS UTILIZADOS .....</b>	<b>9</b>
2.1 Hardware .....	10
2.1.1 Microcontrolador PIC 18F4520 de Microchip ....	10
2.1.2 Pantalla de cristal líquido 16x2 .....	12
2.1.3 Temporizador 7555 .....	16

2.1.4 Comparador LM311 .....	18
2.2 Software .....	19
2.2.1 Programador en lenguaje C PCW CCS .....	19
2.2.2 Simulador de circuitos PROTEUS .....	24
<b>CAPÍTULO 3 .....</b>	<b>29</b>
<b>3. DISEÑO E IMPLEMENTACIÓN .....</b>	<b>29</b>
3.1 Diagrama de bloques .....	29
3.2 Funcionamiento del proyecto .....	30
3.3 Diagrama de flujo del algoritmo .....	31
3.4 Código de programación PCW CCS .....	32
3.4.1 Funciones usadas y dinámica del programa .....	47
3.4.1.1 Manejo de interrupción externa .....	47
3.4.1.2 La función Obtener Frecuencia .....	48
3.4.1.3 La función Calibrando .....	49
3.4.1.4 La función Calibrando 2 .....	50
3.5 El programa principal .....	50
3.6 Operación del instrumento .....	54
<b>CAPÍTULO 4 .....</b>	<b>56</b>
<b>4. SIMULACIÓN, PRUEBAS Y DATOS EXPERIMENTALES .....</b>	<b>56</b>
4.1 Implementación y mediciones, cálculo de error .....	56
4.2 Simulación con Proteus .....	57

**CONCLUSIONES**

**RECOMENDACIONES**

**ANEXOS ..... 69**

**BIBLIOGRAFIA ..... 78**

## **ÍNDICE DE FIGURAS**

Figura 1 Microcontrolador PIC 18F4520 .....	11
Figura 2 Pantalla de cristal líquido 16x2 .....	14
Figura 3 Temporizador 7555 .....	16
Figura 4 Comparador LM311 .....	18
Figura 5 Ambiente del programador PCW CCS .....	20
Figura 6 Ambiente del simulador Proteus .....	27
Figura 7 Diagrama de bloques .....	29
Figura 8 Interruptores de selección del circuito .....	55
Figura 9 Obtención de frecuencia mostrada en osciloscopio .....	58
Figura 10 Frecuencia visualizada en osciloscopio y digital .....	58
Figura 11: Frecuencia de calibración del circuito .....	59
Figura 12: Frecuencia de 1 Khz emitida por el PIC .....	60
Figura 13: Frecuencia de 1 Khz emitida por el PIC .....	60
Figura 14: Finalización de calibración inicial del circuito .....	61
Figura 15: Circuito calibrado y listo para realizar mediciones .....	61
Figura 16: Cálculo de capacitancia de 10 pF (modo LM311) .....	62
Figura 17: Cálculo de capacitancia de 200 nF (modo 7555 en modo oscilador astable) .....	63
Figura 18: Cálculo de capacitancia de 1.1 uF (modo 7555 en modo contador de tiempo en alto de pulso) .....	64

## **INTRODUCCIÓN**

El presente proyecto forma parte del seminario de graduación de “Microcontroladores Avanzados” y consiste en el diseño de un “Medidor LC utilizando pantalla LCD 2x16 para visualización con programa embebido en un microcontrolador” cuya fuente de alimentación es una batería de 9V.

El principal objetivo del proyecto presentado es la medición de la capacitancia de capacitores de todos los tipos polarizados y no polarizados desde 1 pf hasta 3300 uF y de inductancia en inductores, choques y bobinas desde 0,01 uH hasta 20 mH.

Esta es una solución económica y práctica para aparatos de medición que en el mercado pueden llegar a costar cientos de dólares con relativamente pocos componentes, buenas prestaciones y precisión.

El capítulo 1 plantea el problema basado en sus antecedentes y lo que perseguimos al diseñar dicho dispositivo, su funcionamiento y aplicaciones y finalmente se enumeran las distintas técnicas y dispositivos similares al construido.

El capítulo 2 describe todas las herramientas utilizadas en el diseño de nuestro dispositivo, describe también los componentes utilizados, sus características y limitaciones.

Describe también el ambiente de trabajo de los programas utilizados para el desarrollo de las aplicaciones embebidas de nuestro controlador y consideraciones adicionales que se deben tomar en cuenta para un mejor uso de éstas.

El capítulo 3 nos da una detallada descripción funcional del circuito, consideración de construcción, así como la interpretación del código embebido y el funcionamiento de cada uno de los bloques del mismo, se explica el proceso de medición, cálculo y calibración del sistema.

El capítulo 4 nos presenta las simulaciones realizadas en el ambiente ISIS, así como las pruebas realizadas en el circuito real, nos presenta los cálculos y comparación de resultados con otros modelos BK Precision y Fluke para determinar el grado de precisión de nuestro aparato.

Finalmente, se muestran los resultados, conclusiones y recomendaciones finales del proyecto.

## **CAPÍTULO 1**

### **1 DESCRIPCIÓN GENERAL DEL PROYECTO**

#### **1.1 Antecedentes**

El mundo actual exige la instrumentación dentro de cualquier proceso que necesite el control en la fabricación de diversos productos, proceso que es necesario controlar y mantener constantes varias magnitudes; y el estudiante exige el conocimiento práctico de los instrumentos que nos permiten cumplir este fin.

El manejo de instrumentos es fundamental en procesos industriales, ya que el estudiante debe mejorar el conocimiento práctico de los instrumentos, el mismo que podrá hacerlo en cualquier laboratorio de instrumentación.

De particular beneficio para los estudiantes será el hecho de que todos los componentes de este equipo de medición son de uso normal en procesos industriales y esto permitirá una excelente familiarización del estudiante con los sistemas de medición industriales.

#### **1.2 Justificación**



Un estudiante que no tiene conocimientos de medición y no ha adquirido conciencia de los instrumentos y métodos de medida de los que se dispone se encuentra en desventaja a un proceso o una máquina nueva.

Por ello la necesidad de conocer los instrumentos de medición y control para saber su funcionamiento, rangos de aplicación, forma de instalación y además parámetros que dan una idea de la importancia de la instrumentación industrial dentro de nuestra formación académica.

La instrumentación es un área de estudio teórica – práctica que sirve como ayuda tanto para el estudiante como para el docente encargado del mismo entender fácilmente la materia.

### **1.3 Descripción del proyecto**

Nuestro proyecto consiste en la medición de los valores de capacitancia e impedancia de elementos pasivos utilizando un microcontrolador avanzado 18F4520, un circuito oscilador basado en un oscilador tipo tanque y cuya salida está determinada por el circuito comparador de bajo voltaje LM311 de National Semiconductors, adicional a esto para capacitancias de alto valor por encima de 20 nF de capacitancia tenemos un circuito oscilador estable basado en el conocido circuito 7555 que no es otra cosa que el 555 temporizador pero basado

en tecnología CMOS para una mejor respuesta ante voltajes de entrada de baja corriente.

Para programar el PIC 18F4520 que controlará la medición y muestra de estos valores utilizaremos el compilador PCW CCS y los datos serán almacenados en el mismo a través del programador Pickit 3 de Mikroelectronica. El proyecto tiene una capacidad de medición de capacitancias desde 1pF hasta 3.3 mF y de inductancias desde 0.01uH hasta 50 mH

La parte principal del medidor LC es el circuito tanque sintonizado formado por LC (100uH y 1nF) y el NE7555. En este caso esos valores de L y C se toman para formar una referencia que en teoría oscilará:

$$f = \frac{1}{2 * \pi * \sqrt{C * L}} = 555793\text{Hz}$$

Cuando nosotros introducimos un componente a medir (La inductancia en serie a la de referencia y la capacidad en paralelo) se modifica esta frecuencia, la cual la medimos y por medio de simple cálculos se puede determinar el valor del componente desconocido: Por ejemplo, colocamos un capacitor y se mide 330000Hz, entonces:

$$f = \frac{1}{2 * \pi * \sqrt{(C + C_x) * L}} = 330000\text{Hz}$$

Despejando:

$$C_x = \left(\frac{1}{2 * \pi * f}\right)^2 * \frac{1}{L} - C = 1.83nF$$

De la misma manera se hace para L.

Entonces la base del medidor es determinar la frecuencia, para ello hay 2 maneras, midiendo el período de la señal o determinando la cantidad de pulsos en un determinado rango de tiempo.

Ya determinada la frecuencia lo único que queda es determinar el valor aplicando por ejemplo:

$$L_x = \left(\frac{1}{2 * \pi * f}\right)^2 * \frac{1}{C} - L$$

Y aplicar la escala en uH o mH.

Para el proceso de autocalibración del circuito tenemos la ayuda de un relé, en este caso el proceso de calibración procede así:

Primero el circuito toma la frecuencia del capacitor conocido C:

$$f_1 = \frac{1}{2 * \pi * \sqrt{L_1 * C_1}}$$

Luego se activa la salida del reed relé que pone en paralelo a un capacitor de precisión Ca de 1000 pF, de esta manera la frecuencia cambia a:

$$f_2 = \frac{1}{2 * \pi * \sqrt{L_1 * (C_1 + 1000)}}$$

De esta manera tenemos las dos ecuaciones que necesitamos para calcular con precisión los valores de L1 y C1 es decir los capacitores de 1nF y la inductancia de 100 uH que componen normalmente nuestro circuito tanque de esta manera damos mayor precisión al circuito

Las ecuaciones quedan así:

$$C_1 = \frac{f_2^2}{(f_1^2 - f_2^2)} * 1000pF$$

Finalmente,

$$L_1 = \frac{1}{4 * \pi^2 * f_1^2 * C_1}$$

Este circuito se comporta a la perfección para capacitancias de aproximadamente un orden de 20 nF, mas para medir capacitancias más altas tenemos que servirnos del circuito 7555 que en su modo estable posee la capacidad de que su frecuencia de oscilación tiene una relación inversamente proporcional a la capacitancia determinada por la fórmula:

$$f = \frac{1.44}{(R_a + R_b) * C}$$

Donde el factor 1,44/(Ra+Rb) se puede reemplazar por K dado que es una constante con lo cual

$$f = \frac{K}{C}$$

Para el proceso de calibración de este circuito tenemos que calcular la frecuencia simplemente y por medio de esta calculamos la constante K con un capacitor de precisión.

#### **1.4 Aplicaciones**

La principal aplicación del proyecto es la medición de capacitancias e inductancias de elementos pasivos para su posterior utilización en la implementación de circuitos eléctricos y electrónicos brindando confiabilidad al mismo sabiendo que los elementos utilizados son de la medida solicitada, así como comprobar si uno de estos elementos se encuentra defectuoso.

Es dirigido precisamente al estudiante de electrónica o al hobista que desee un aparato de bajo costo y precisión del orden del  $\pm 2\%$  que no debe faltar en ningún laboratorio de estudiante de electrónica.

Otra aplicación sería la de medir bobinas y embobinados de motores de esta manera probar su inductancia, o medir capacitancia en diodos varactores, capacitores variables de sintonizadores de radio así como en bobinas de los sintonizadores de recepción de radiofrecuencias y con una pequeñísima modificación de su código tiene la opción de ser un medidor de ciclo de trabajo y un frecuencímetro de hasta 5 Mhz de precisión.

## 1.5 Funcionamiento de aparatos similares

Nuestro propósito es servirnos de técnicas aplicadas, mejorarlas y obtener lo mejor de cada uno de los aparatos existentes.

Actualmente se puede encontrar en el mercado un capacímetro/inductímetro de hasta 1 uF en capacitancia con autorrango y autocalibración como el LCMeter IIB pero la limitante de este circuito es obviamente el rango de capacitancia porque la mayoría de los capacitores comercialmente usados son del orden de los cientos y miles de microfaradios.

Por lo tanto tomamos de este circuito su principal característica el oscilador tipo tanque controlado por el comparador LM311 y lo usamos para poder medir capacitancias pequeñas e inductancias.

Hay un sinnúmero de diseños de capacímetros basados en cambio de frecuencia de un circuito 555 en modo astable de todos ellos tomamos su mejor característica para ajustar los rangos de las resistencias de carga y descarga  $R_a$  y  $R_b$  además de consejos para calibración de los mismos mas obviamente la limitante de los mismos es que no pueden medir inductancias y su capacidad de medir capacitancias pequeñas es relativamente imprecisa.

Es por todo esto que para que nuestro circuito se diferencie de ellos debe tomar lo mejor de ambos y poder tener la capacidad de autorranego y autocalibración que ambos tienen.

Adicional a esto y de grandes casas comerciales como Fluke o Bk Precision tenemos aparatos en extremo similares cuyos costos son elevados es por esto que nuestro proyecto se diferencia.

## **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

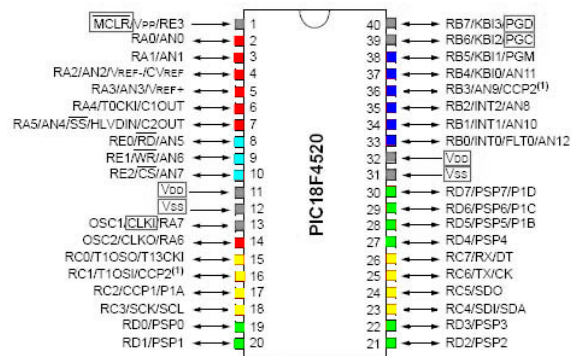
## **2.1 HARDWARE**

### **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 expandir 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)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

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

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

- 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

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

- 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

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	1/D	S

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**

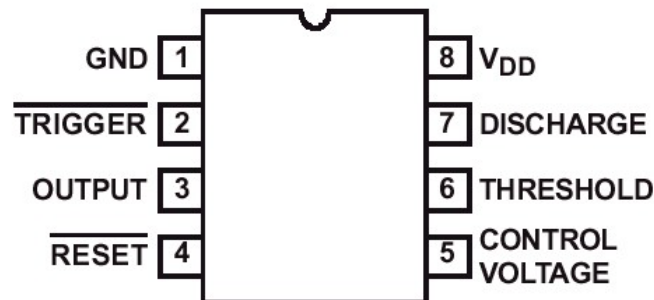
- Interfaz para aplicaciones con microcontroladores, permite visualizar 32 caracteres alfanuméricos en dos líneas de 16 caracteres
- Opciones de alimentación: 5V, 12V o dual
- Consumo reducido, del orden de 7.5mW
- Protecciones contra inversión de polaridad y picos de tensión
- Cuenta con un microcontrolador 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**

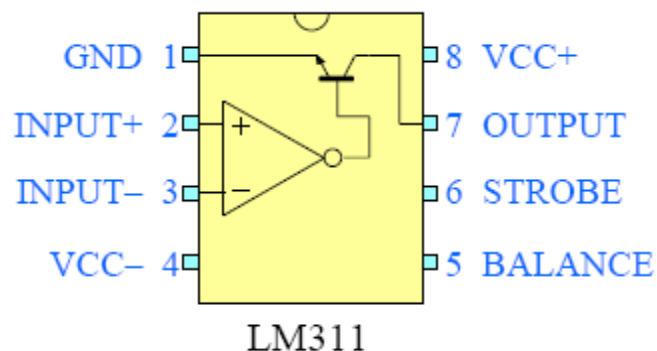
- Equivalente exacto en la mayoría de las aplicaciones del temporizador NE555
- Bajo consumo de corriente (80uA)
- Corrientes de disparo, threshold y reset extremadamente bajas (20pA)
- Operación de alta velocidad (500KHz)
- Operación en amplio rango de voltajes (3V a 16V)
- Puede usarse con altas impedancias 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 °C.
- 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

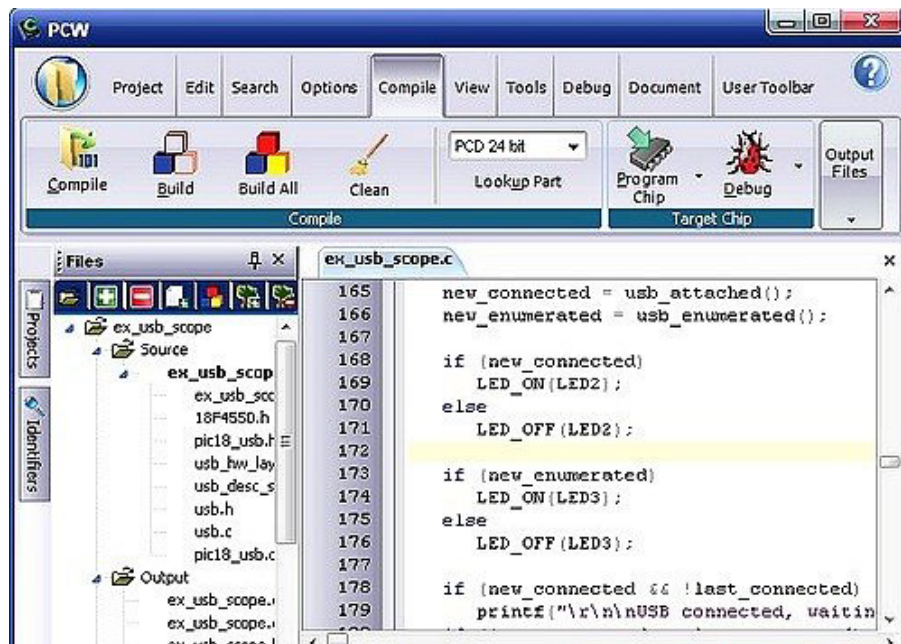
## **2.2 SOFTWARE**

### **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, convertor 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

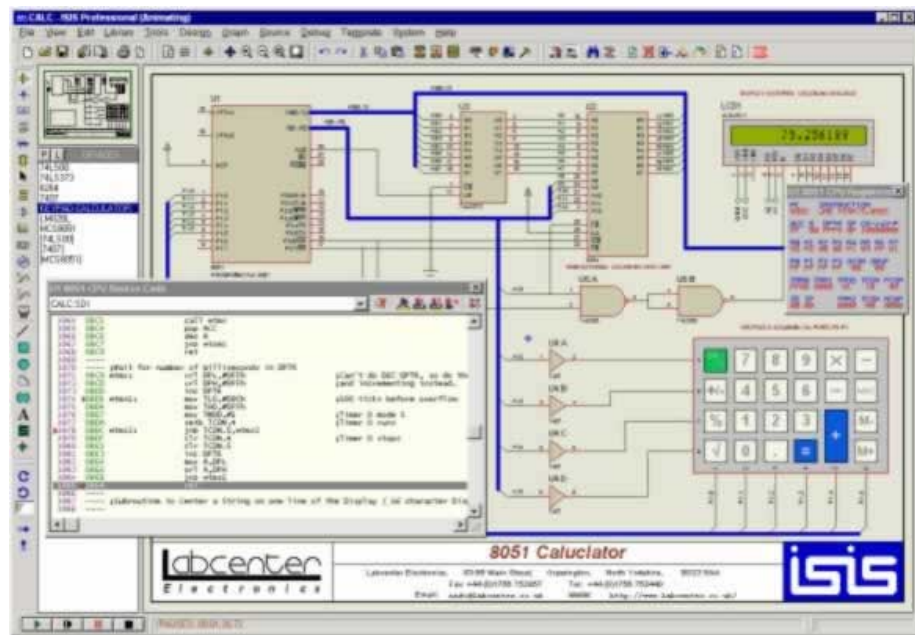


ha escrito en ensamblador como si se ha utilizado un lenguaje de alto nivel, permitiendo interactuar con nuestro diseño

- 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.

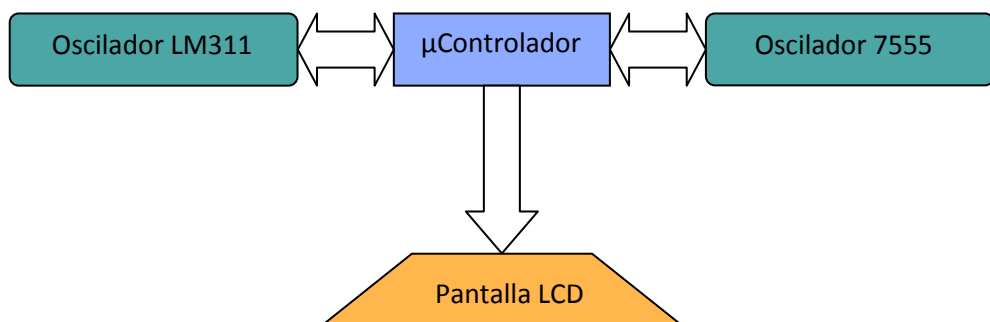
- La lista de redes (NetList) son compatibles con la mayoría de los programas de realización de PCB.
- Se puede visualizar la RAM (Registros especiales y datos), además de la EEPROM y la memoria de programa. Se pueden establecer puntos de detención para la depuración.
- Posee una variada instrumentación 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.

## CAPÍTULO 3

### 3 DISEÑO E IMPLEMENTACIÓN

#### 3.1 Diagrama de bloques del Proyecto

El circuito del medidor está constituido por 4 partes principales que realizan sistemáticamente cada operación necesaria para obtener el valor de la capacitancia o inductancia conectada al circuito. A continuación se detalla el diagrama de bloques explicando el mismo:



**Figura 7: Diagrama de bloques del proyecto**

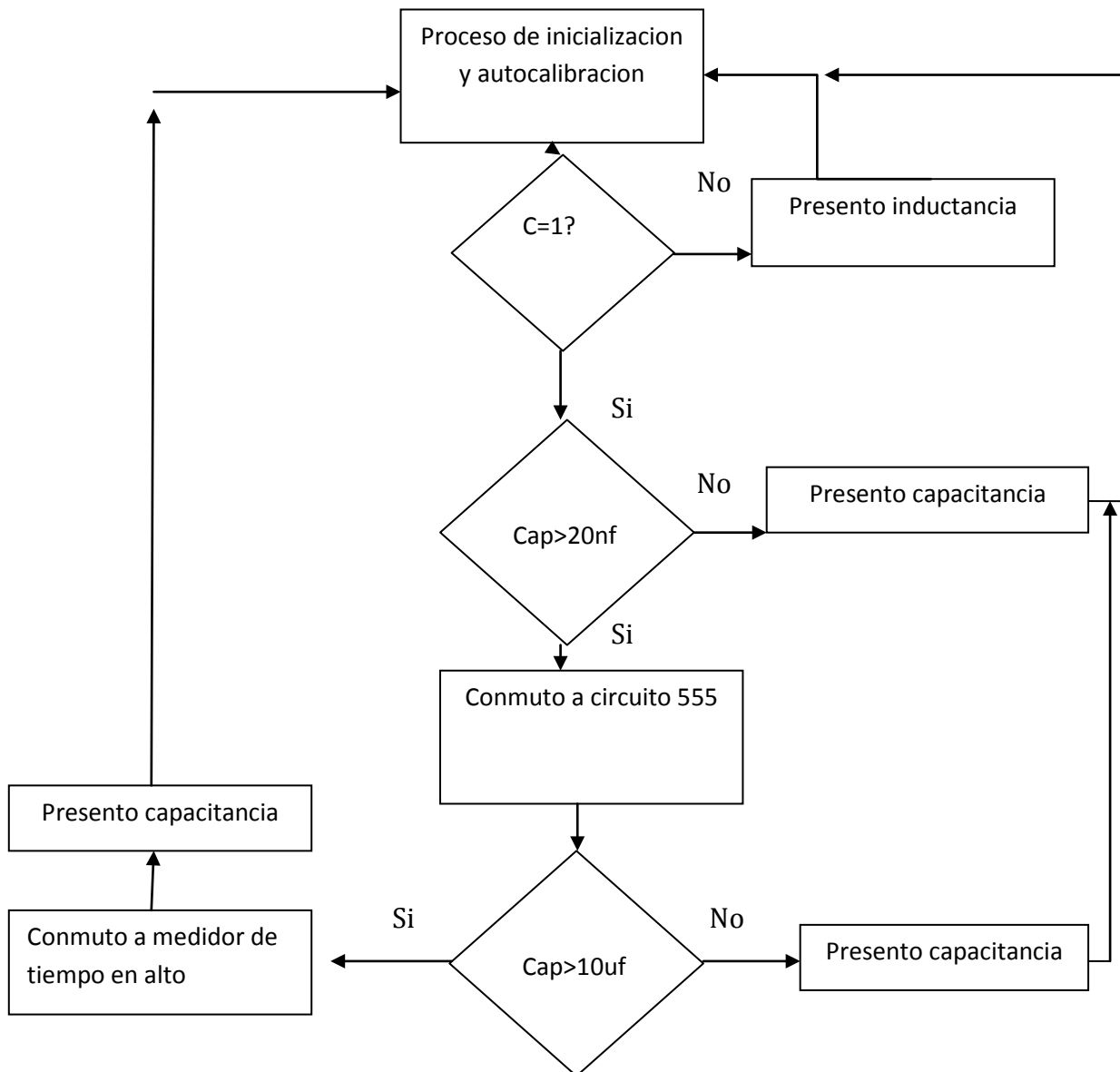
### 3.2 Funcionamiento del Proyecto

El programa inicia con la asignación de constantes y de variables así como la definición de la rutina de interrupción externa que se usa aquí, posterior a esto realiza la autocalibración de ambos circuitos de oscilación.

Para calibrar el oscilador con LM311 toma dos frecuencias, una con el capacitor C y la inductancia L y luego cierra el relé que pone en paralelo al capacitor de calibración  $C_{cal}$ , luego de esto calcula el valor de C y L y posteriormente procede a realizar la calibración del circuito 555 misma que procede de la misma manera usando un relay de conmutación con un capacitor de mas capacidad para calcular los valores de C2 y R del circuito, luego de esto se coloca como 000pf.

El siguiente paso con el oscilador LM311 es el valor de la capacitancia; si esta resulta ser menor de la que el circuito resonante necesita para oscilar es decir el equivalente a un capacitor de 20 nF, conmuta este capacitor para ser medido con el circuito oscilador 555, no sin antes haber preguntado por el modo si es Capacitancia o Inductancia, enseguida el 555 calcula su valor de capacitancia en base a la frecuencia, si la frecuencia es menor a un valor equivalente a un capacitor de 10 uf esta se conmuta no a medir la frecuencia sino el tiempo en alto esto se logra por medio de la interrupción externa y un oscilador de 1 Khz externo, para conveniencia y precisión se logra con el PIC de gama baja 12F629.

### 3.3 Diagrama de flujo del algoritmo



### 3.4 Código de programación PCW CCS

```

#include <18F4520.h>

#fuses HS,NOWDT,NOPROTECT,NOLVP

#use delay(clock=20000000)

#bit t1_overflow=0x0C.0

#include <lcd.c>

#bit TMRIF=0XF9E.0

#byte portb=6

#byte portc=9

int1 flagToggleFlanco=0, flagHayDatos=0;

int16 t1=0x00, t2=0x00, tt=0x00;

int ciclos8, ciclos;

int8 digito0, digito1, digito2, digito3, decimal0, decimal1;

int32 mod1, mod2, mod3, mod4, mod5, mod6;

int32 freq, freq2, cap, cap2, microcap;

float capf, frecf, frecf2, capacitor, microcapac, microinduc;

float inductor, frecal1, frecal2, capacitor2, k1, k2, capcal;

float inducal, capacitorfix, capacitorfix2, resiscal;

int16 freqc_high;

int16 freqc_low;

#int_ext

void handle_ext_int(){

    if(flagToggleFlanco==0){

        t1=get_timer1();
    }
}

```

```
    ext_int_edge(0,H_TO_L);
    flagToggleFlanco=1;}
    else {
    t2=get_timer1();
    ext_int_edge(0,L_TO_H);
    flagToggleFlanco=0;
    set_timer1(0);
    if(flagHayDatos==0)
        {flagHayDatos=1;
        }}
}

void obtener_freq()
{
    ciclos8=0;
    ciclos=0;
    freqc_high=0;
    TMRIF=0;
    set_timer1(0);
    setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
    while (ciclos!=0xFF)
    {
        while (ciclos8!=0xFF)
        {
            if (TMRIF)
            {
```



```
    TMRIF=0;freqc_high++;
  }
else
{
  delay_cycles(5);
}
  delay_cycles(62);
  ciclos8++;
}
  delay_cycles(216);
  ciclos++;
}
  delay_cycles(211);
  setup_timer_1(T1_DISABLED);
  if (TMRIF)
  {
    freqc_high++;
  }
  freqc_low=get_timer1();
  freq=make32(freqc_high,freqc_low);
  freq2=freq;
  frecf=freq;
  frecf=frecf/1000;
}
void calibrando()
```

```
{
lcd_putc('\f');
lcd_putc(" calibrando 311");
delay_ms(100);
obtener_freq();
frecal1=freq;
frecal1=frecal1/1000;
output_bit(PIN_E0,1);
delay_ms(100);
obtener_freq();
frecal2=freq;
frecal2=frecal2/1000;
frecal1=frecal1*frecal1;
frecal2=frecal2*frecal2;
output_bit(PIN_E0,0);
capcal=(capacitorfix*frecal2)/(frecal1-frecal2);
inducal=(k1/capacitorfix)*((1/(frecal2))-(1/(frecal1)));
}

void calibrando2()
{
lcd_putc('\f');
lcd_putc("Medidor LC ");
lcd_putc('\n');
lcd_putc("ESPOL V 1.0.3");
delay_ms(100);
```

```
    obtener_freq();
    frecf2=frecf;
}

void main()
{
    delay_ms(10);
    set_tris_a(0xff);
    set_tris_e(0x08);
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_spi(FALSE);
    setup_psp(PSP_DISABLED);
    output_bit(PIN_E0,0);
    output_bit(PIN_E1,1);
    output_bit(PIN_E2,1);
    lcd_init();
    setup_timer_1(T1_EXTERNAL | T1_DIV_BY_1);
    port_b_pullups(FALSE);
    k1=0.02533029;
    k2=1.443;
    capacitorfix=0.00104;
    capacitorfix2=1.00;
    calibrando();
    output_bit(PIN_E1,0);
```

```
output_bit(PIN_E2,0);
delay_ms(25);
calibrando2();
output_bit(PIN_E1,1);
output_bit(PIN_E2,1);
delay_ms(200);
while (true)
{
  disable_interrupts(int_ext);
  disable_interrupts(global);
  output_bit(PIN_E1,1);
  if (input(pin_a1))
  {
    obtener_freq();
    frecf=frecf*frecf;
    capf=(k1/(frecf*inducal))-capcal;
    capacitor=capf*1000000;
    microcapac=capacitor*100;
    freq=capacitor;
    microcap=microcapac;
  }
  else
  {
    obtener_freq();
    frecf=frecf*frecf;
  }
}
```

```
capf=(k1/(frecf*capcal))-inducal;
inductor=capf*1000000;
microinduc=inductor*100;
freq=inductor;
microcap=microinduc;
}
cap=freq;
if (cap>999)
{
if (cap>9999||freq2<1)
{
if (input(pin_a1))
{
do
{
output_bit(PIN_E1,0);
delay_ms(50);
obtener_freq();
resiscal=14.1*frecf;
capf=k2/resiscal;
capacitor2=capf*100000;
if (capacitor2>99999||freq2<1)
{
if (capacitor2>999999||freq2<1)
{
```

```
output_bit(PIN_E2,0);
setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
set_timer1(0x0000);
disable_interrupts(global);
disable_interrupts(int_timer1);
disable_interrupts(int_rda);
disable_interrupts(int_ext);
ext_int_edge(0,L_TO_H);
flagToggleFlanco=0;
enable_interrupts(int_rda);
enable_interrupts(int_ext);
enable_interrupts(global);
do
{
cap=1;
if(flagHayDatos==1)
{
if(t2>t1)
{
tt=t2-t1;
capf=tt;
frecf=capf*40/(260*frecf2);
cap=frecf;
mod1=cap/10;
mod2=cap/100;
```

```
    mod3=cap/1000;
    mod4=cap/10000;
    mod5=cap/100000;
    mod6=cap/1000000;
    decimal1=(cap%10)+48;
    decimal0=(mod1%10)+48;
    digito0=(mod2%10)+48;
    digito1=(mod3%10)+48;
    digito2=(mod4%10)+48;
    lcd_gotoxy(1,1);
    lcd_putc("C =");
    lcd_putc(digito2);
    lcd_putc(digito1);
    lcd_putc(digito0);
    lcd_putc(decimal0);
    lcd_putc(decimal1);
    lcd_putc(" uf");
}
}
}
while (cap<5300&&cap!=0);
disable_interrupts(global);
disable_interrupts(int_timer1);
disable_interrupts(int_rda);
disable_interrupts(int_ext);
```

```
    output_bit(PIN_E2,1);
    output_bit(PIN_E1,1);
}
else
{
    freq=capacitor2/1000;
}
}
else
{
    freq=capacitor2;
}
cap2=freq;
mod1=cap2/10;
mod2=cap2/100;
mod3=cap2/1000;
mod4=cap2/10000;
decimal1=(cap2%10)+48;
decimal0=(mod1%10)+48;
digito0=(mod2%10)+48;
digito1=(mod3%10)+48;
digito2=(mod4%10)+48;
lcd_putc('\f');
lcd_putc("C =");
lcd_putc(digito2);
```



```
lcd_putc(digito1);
lcd_putc(digito0);
lcd_putc(".");
lcd_putc(decimal0);
lcd_putc(decimal1);
if (capacitor2>99999)
{
    lcd_putc(" uf");
}
else
{
    lcd_putc(" nf");}
    output_bit(PIN_E1,0);
    delay_ms(20);
}
while (capacitor2>1000);
output_bit(PIN_E1,1);
}
else
{
    if (cap>399999)
    {
        lcd_putc('\f');
        lcd_putc("Fuera de rango");
    }
}
```

```
else
{
    mod2=cap/100;
    mod3=cap/1000;
    mod4=cap/10000;
    mod5=cap/100000;
    mod6=cap/1000000;
    decimal0=(mod1%10)+48;
    digito0=(mod2%10)+48;
    digito1=(mod3%10)+48;
    digito2=(mod4%10)+48;
    digito3=(mod5%10)+48;
    lcd_putc('\f');
    lcd_putc("L =");
    lcd_putc(digito3);
    lcd_putc(digito2);
    lcd_putc(digito1);
    lcd_putc(".");
    lcd_putc(digito0);
    lcd_putc(decimal0);
    lcd_putc(" mh");
}
}
else
```

```
{
mod2=cap/100;
mod3=cap/1000;
mod4=cap/10000;
mod5=cap/100000;
mod6=cap/1000000;
decimal0=(mod1%10)+48;
digito0=(mod2%10)+48;
digito1=(mod3%10)+48;
digito2=(mod4%10)+48;
digito3=(mod5%10)+48;
lcd_putc('\f');
if (input(pin_a1))
{
lcd_putc("C=");
}
else
{
lcd_putc("L=");
}
lcd_putc(digito3);
lcd_putc(digito2);
lcd_putc(digito1);
lcd_putc(".");
lcd_putc(digito0);
```

```
    lcd_putc(decimal0);
    if (input(pin_a1))
    {
        lcd_putc(" nf");
    }
    else
    {
        lcd_putc(" mh");
    }
}
}
else
{
    if (input(pin_a1))
    {
        mod1=cap/10;
        mod2=cap/100;
        decimal1=(cap%10)+48;
        decimal0=(mod1%10)+48;
        digito0=(mod2%10)+48;
        lcd_putc('\f');
        lcd_putc("C =");lcd_putc(digito0);
        lcd_putc(decimal0);
        lcd_putc(decimal1);
        lcd_putc(" pf");
    }
}
```

```
    }  
else  
    {  
        mod1=microcap/10;  
        mod2=microcap/100;  
        mod3=microcap/1000;  
        mod4=microcap/10000;  
        decimal1=(microcap%10)+48;  
        decimal0=(mod1%10)+48;  
        digito0=(mod2%10)+48;  
        digito1=(mod3%10)+48;  
        digito2=(mod4%10)+48;  
        lcd_putc('\f');  
        lcd_putc("L =");  
        lcd_putc(digito2);  
        lcd_putc(digito1);  
        lcd_putc(digito0);  
        lcd_putc(".");  
        lcd_putc(decimal0);  
        lcd_putc(decimal1);  
        lcd_putc(" uh");  
    }  
}  
}
```

### 3.4.1 Funciones usadas y dinámica del programa

#### 3.4.1.1 Manejo de la interrupción externa

```
void handle_ext_int() {
    if(flagToggleFlanco==0) {
        t1=get_timer1();
        ext_int_edge(0,H_TO_L);
        flagToggleFlanco=1;}
    else {
        t2=get_timer1();
        ext_int_edge(0,L_TO_H);
        flagToggleFlanco=0;
        set_timer1(0);
        if(flagHayDatos==0)
            {flagHayDatos=1;
            }} }
}
```

La primera de las funciones encontradas es la de manejo de la interrupción externa por parte de la pata RB0 la cual es usada en el modo de conteo de tiempo en alto y sirve para determinar el tiempo en alto de la señal emitida por el oscilador 7555, lo que concretamente hace esta función es cambiar el flanco de detección de la interrupción externa.

### 3.4.1.2 La función Obtener frecuencia

```

void obtener_freq()
{
    ciclos8=0;
    ciclos=0;
    freqc_high=0;
    TMRIF=0;
    set_timer1(0);
    setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
    while (ciclos!=0xFF)
    {
        while (ciclos8!=0xFF)
        {
            if (TMRIF)
            {
                TMRIF=0;freqc_high++;
            }
            else
            {
                delay_cycles(5);
            }
            delay_cycles(62);
            ciclos8++;
        }
        delay_cycles(216);
        ciclos++;
    }
    delay_cycles(211);
    setup_timer_1(T1_DISABLED);
    if (TMRIF)
    {
        freqc_high++;
    }
    freqc_low=get_timer1();
    freq=make32(freqc_high,freqc_low);
    freq2=freq;
    frecf=freq;
    frecf=frecf/1000;
}

```

La función obtener frecuencia es la base de todo el programa pues calcula la frecuencia en base al contador del timer 1 misma que está gobernada por un lazo

de un segundo de duración entregando la frecuencia en hercios para freq y en kilohercios para frecv.

### 3.4.1.3 La función Calibrando

```
void calibrando()
{
  lcd_putc('\f');
  lcd_putc("  calibrando 311");
  delay_ms(100);
  obtener_freq();
  frecv1=frecv;
  frecv1=frecv1/1000;
  output_bit(PIN_E0, 1);
  delay_ms(100);
  obtener_freq();
  frecv2=frecv;
  frecv2=frecv2/1000;
  frecv1=frecv1*frecv1;
  frecv2=frecv2*frecv2;
  output_bit(PIN_E0, 0);
  capcal=(capacitorfix*frecv2)/(frecv1-frecv2);
  inducal=(k1/capacitorfix)*((1/(frecv2))-1/(frecv1));
}
```

Esta función es la que calcula dos frecuencias, una de oscilación inicial con los componentes L y C un inductor y un capacitor de tolerancia aceptable y luego otra, puesta en paralelo con un capacitor de alta calidad que sirve para realizar los cálculos para obtener los valores de L y C que se usaran posteriormente en el programa inicial.



### 3.4.1.4 La función Calibrando 2

```
void calibrando2 ()
{
  lcd_putc ('\f');
  lcd_putc ("Medidor LC ");
  lcd_putc ('\n');
  lcd_putc ("ESPOL V 1.0.3");
  delay_ms (100);
  obtener_freq();
  frecf2=frecf;
}
```

Esta función logra calcular con precisión el oscilador que da la cuenta de tiempo en alto del oscilador 7555 cuando se usa en modo interrupción externa es decir ajusta el reloj de 1 Khz en caso de que no fuera así guardando este valor en frecf2.

## 3.5 El programa principal

Vamos a centrarnos en analizar las partes más importantes del programa principal expuesto anteriormente en orden de simplificar su comprensión y dado que arriba ya está expuesto vamos a resumir rutinas con nombres.

```

void main()
{
  inicializacion de variables y puertos;
  calibrando();
  calibrando2();
  while (true)
  {
    output_bit(PIN_E1,1);
    if (input(pin_a1))
    {
      obtener_freq();
      calculo capacitancia;
    }
    else
    {
      calculo inductancia;
    }
    cap=freq;
    if (cap>999)
    {
      if (cap>9999||freq2<1)
      {
        if (input(pin_a1))
        {
          do
          {
            conmuto al 555 y calculo capacitancia en decimos de nF;
            if (capacitor2>99999||freq2<1)
            {
              if (capacitor2>999999||freq2<1)
              {

```

Como vemos al inicio del programa se llama a las funciones de calibración y se realiza la inicialización de las interrupciones y de los puertos, es entonces cuando se llama a la función de obtener frecuencia, con esta frecuencia de oscilación y en función únicamente del pin de selección L/C se calcula bien la capacitancia o la inductancia del circuito ahora bien si este valor llega a ser mayor que 999 es obviamente ahí donde comienza nuestro primer autorango pues luego de esto ya pasa al rango de los nanofaradios o microhenrios según se dé el caso así que comienza nuestro primer lazo; si, luego de esto y tomando en consideración que con un capacitor de 20 nf el oscilador LM311 empieza a

presentar problemas de falta de oscilación conmutamos el capacitor al circuito oscilador 7555 que se comporta de mejor manera con capacitores más grandes.

Este cambio de circuito se logra gracias al relé conectado a la salida de la pata E1 el mismo que conmuta el valor. Ahora bien, nuevamente nos topamos con un cálculo de capacitancia pues este es un nuevo oscilador por lo que llamamos a obtener frecuencia nuevamente y calculamos con la nueva fórmula, esta nos lo entrega en décimas de nanofaradio, pues bien si este valor supera los 999.9 nanofaradios tenemos que llamar al segundo rango y pasamos a presentar en microfaradios.

```

commuto al calculo de tiempo en alto;
do
{
cap=1;
if(flagHayDatos==1)
{
if(t2>t1)
{
calculo y presento capacitancia en uf;
}
}
}
while (cap<5300&&cap!=0);
regreso al modo LM311
}
else
{
freq=capacitor2/1000;
}
}
else
{
{
freq=capacitor2;
}
presento capacitancia en;;
if (capacitor2>99999)
{
lcd_putc(" uf");
}
}
else
{

```

```

else
{
    lcd_putc(" nf");
    regreso al modo LM311
}
while (capacitor2>1000);
regreso al modo LM311
}
else
{
    if (cap>399999)
    {
        lcd_putc('\f');
        lcd_putc("Fuera de rango");
    }
    else
    {
        presento datos en mH;
    }
}
}
}

```

Pues bien resulta que el cálculo de la función obtener frecuencia es válido hasta que la frecuencia sea mayor o cercana a 1 hercio luego de lo cual esta se vuelve completamente inútil para calcular capacitancia, pues no calcula décimas ni centésimas de hercio, si bien el oscilador 7555 no ha parado de oscilar simplemente su frecuencia se ha vuelto indetectable para la función por eso al llegar al valor crítico de 10 uF o 9999.9 nanofaradios llamamos al tercer modo de cálculo.

El nuevo cálculo no realiza conmutación de circuito para el capacitor mas sí lo realiza para el timer 1 que ahora recoge la frecuencia de 1 Khz que da la temporización para que el contador del programa pueda calcular el tiempo en alto por medio de la interrupción externa, esto lo hará hasta un máximo teórico de 5300 uF, si excede este valor o si se vuelve más pequeño que 10 uF volverá al modo principal de oscilación por el LM311, luego de esto se presentarán los

datos calculados en el formato que corresponda a si son capacitores y de que rango de eso se encargan los lazos finales.

```

else
{
presento datos en:
if (input(pin_a1))
{
lcd_putc("C =");
}
else
{
lcd_putc("L =");
}
if (input(pin_a1))
{
lcd_putc(" nf");
}
else
{
lcd_putc(" mh");
}
}
}
else
{
if (input(pin_a1))
{
presento datos en pf;
}
else
{
presento datos en nf;
}
}

```

### 3.6 Operación del instrumento

El instrumento consta de dos interruptores y un botón.

Para encender el instrumento el interruptor marcado como encendido tiene que estar hacia la derecha, el botón etiquetado como 3 pondrá al circuito a calibrar

teniendo que encenderse siempre en el modo C del interruptor L/C o en caso contrario si el interruptor se encuentra en el modo L los lagartos de medición deben estar cortocircuitados para poder ser calibrado.



**Figura 8: Interruptores de selección del circuito**

Para proceder con la autocalibración del circuito nuevamente se presiona el botón marcado como calibración botón 3 que pondrá el circuito a calibrarse tomando en consideración que el interruptor LC debe ser manejado como en el encendido.

## **CAPÍTULO 4**

### **4 SIMULACIÓN, PRUEBAS Y DATOS EXPERIMENTALES**

#### **4.1 Implementación y mediciones cálculo de error**

Tal como se puede observar en el anexo 7 la tabla nos muestra un comportamiento relativamente estable de nuestro medidor tanto para capacitores grandes como pequeños, lo que contrasta con el medidor Fluke el cual sólo es preciso para altos valores y el medidor BK Precision que lo es para los valores del orden desde 1 uF hasta 1000 uF. Lo que podemos concluir de estos resultados de la simulación es que fue un éxito la implementación del circuito; observamos como en capacitores de picofaradios las medidas están dentro de la tolerancia del componente lo que supondría tasa de error cero en la medición en contraste con los otros medidores.

## 4.2 Simulación con PROTEUS

Para poder realizar la simulación se debe tomar en cuenta los siguientes factores primero que el oscilador LM311 no podía ser simulado, se usó su contraparte mejorada el LM111; segundo, que al ser modelado el PIC más el comparador hacía la simulación demasiado lenta para los valores 1000 pF y 100 uH en la simulación, por lo que se optó por presentar los mismos multiplicados por diez para que así concluyera en un par de horas la simulación caso contrario hubiera tomado diez veces ese tiempo.

Al encender el circuito y luego de transcurrido el primer segundo observamos como la frecuencia corresponde a una oscilación de L y C aún no calculados, mas como en Proteus se cuenta con valores exactos podemos decir que es de 1 nF y 100 uH lo que nos da una frecuencia teórica de 503 Khz, pero por la transición de inicio se ubica en 493 Khz en el contador de frecuencia y en el osciloscopio podemos calcular a modo grueso 500 Khz por las divisiones del osciloscopio así como observamos la frecuencia de oscilación del pic 12f629.



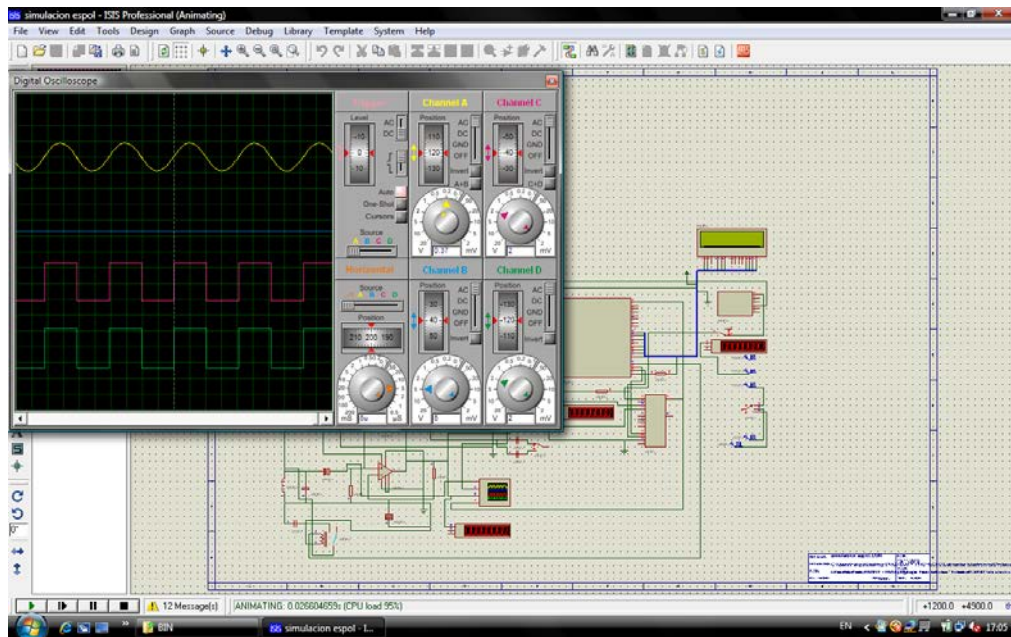


Figura 9: Obtención de frecuencia mostrada en el osciloscopio

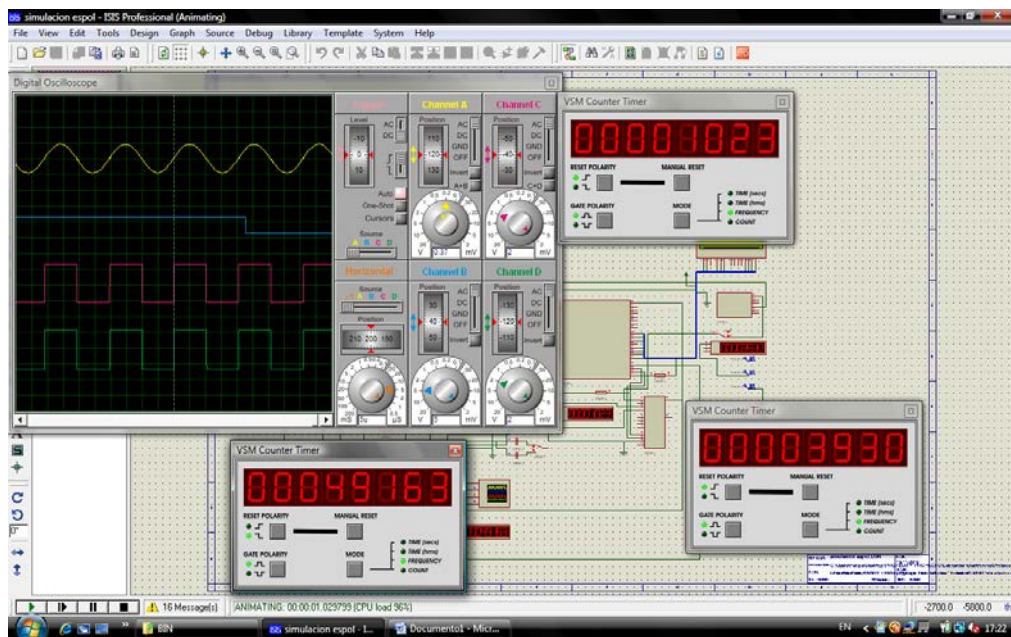
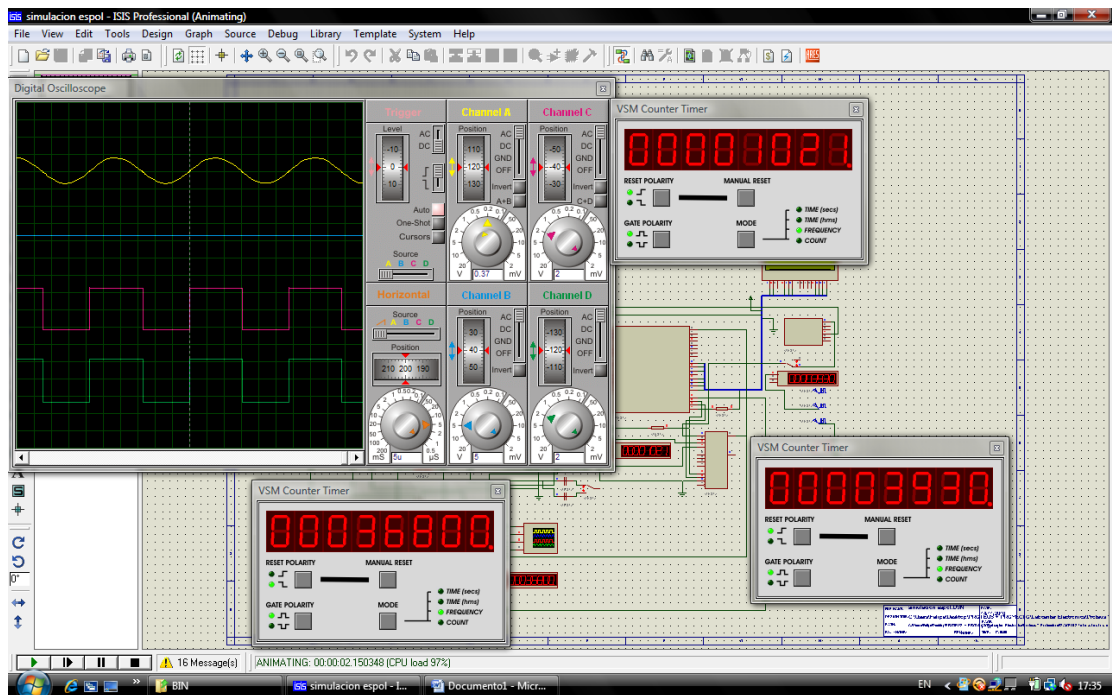


Figura 10: Frecuencia visualizada en osciloscopio y digitalmente

Luego de transcurrido 1,15 segundos vemos como se cierra el relé de calibración y de esta manera el circuito cambia de frecuencia.



**Figura 11: Frecuencia de calibración del circuito**

La frecuencia real debería ser de 355 Khz y que calculamos por ser 5,8 divisiones de 2 us en 3,44 Khz lo que corresponde con nuestra calibración.

Luego de este proceso el circuito calibra el oscilador del PIC 12F629 y realiza este proceso cambiando el 74153 hacia el primer puerto, esto también conmuta el relé de modo mas no es usado todavía.

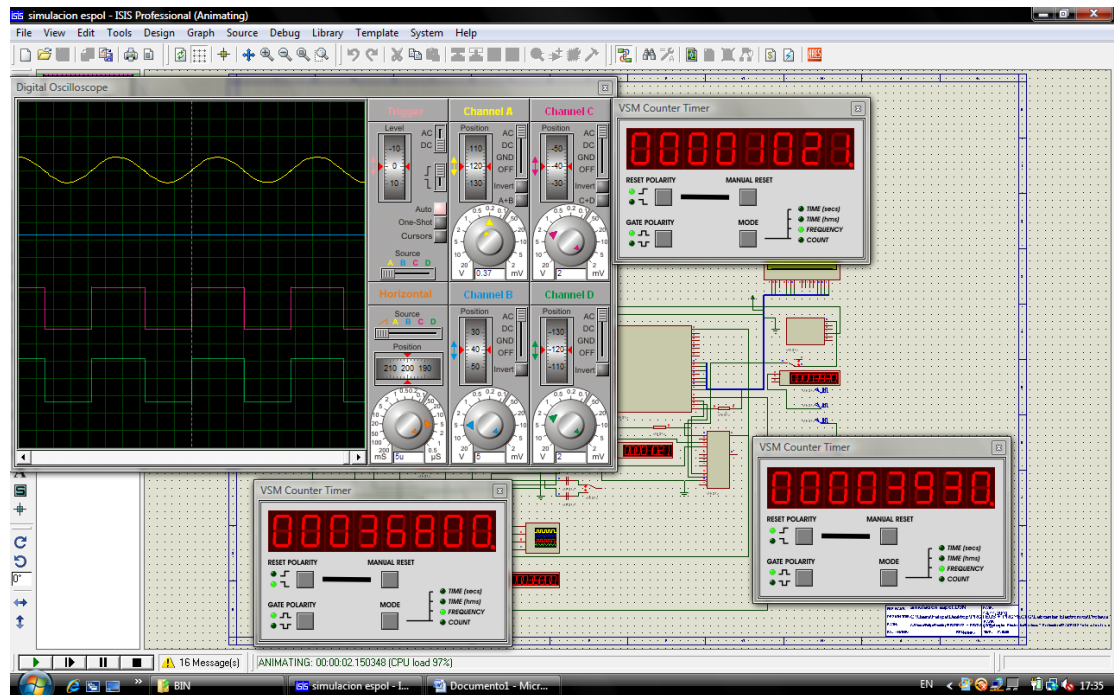


Figura 12: Frecuencia de 1 KHz emitida por el PIC

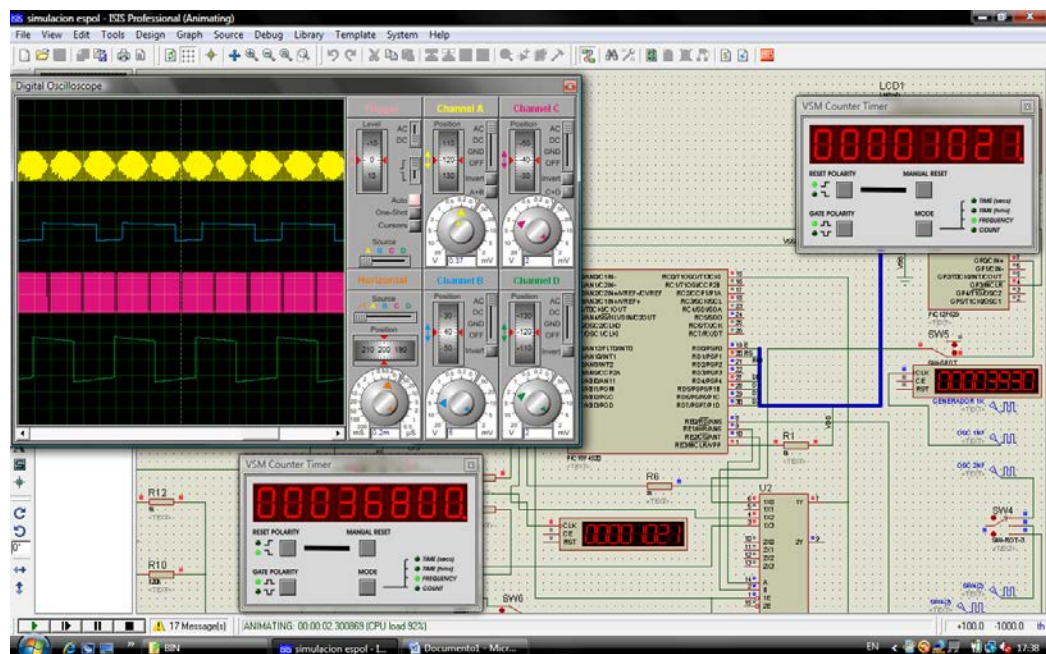


Figura 13: Frecuencia de 1 KHz emitida por el PIC

Claramente se observa la frecuencia de 1 KHz emitida por el Pic; este ha terminado su proceso de calibración y vuelve a la frecuencia inicial mostrando ahora esta como una medición de 0 pF.

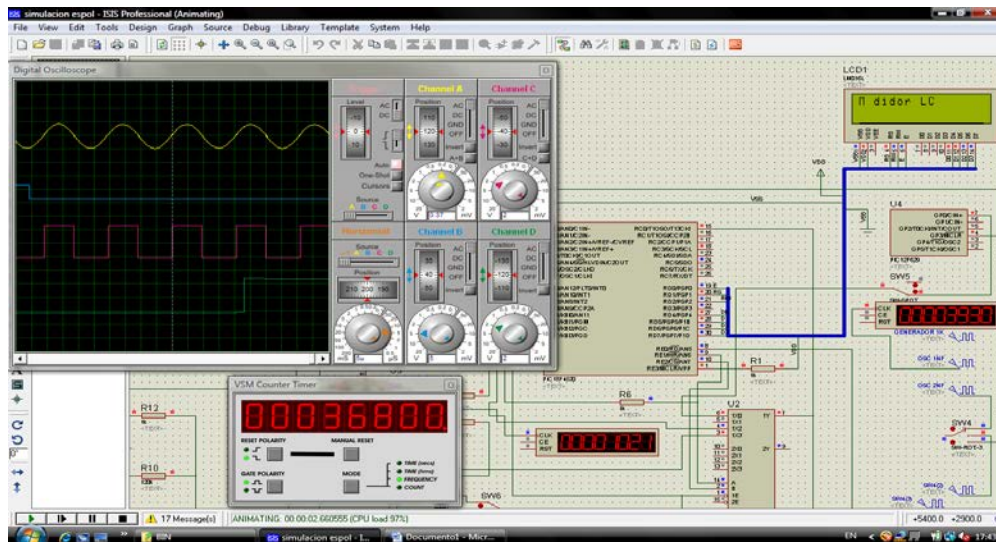


Figura 14: Finalización de calibración inicial del circuito

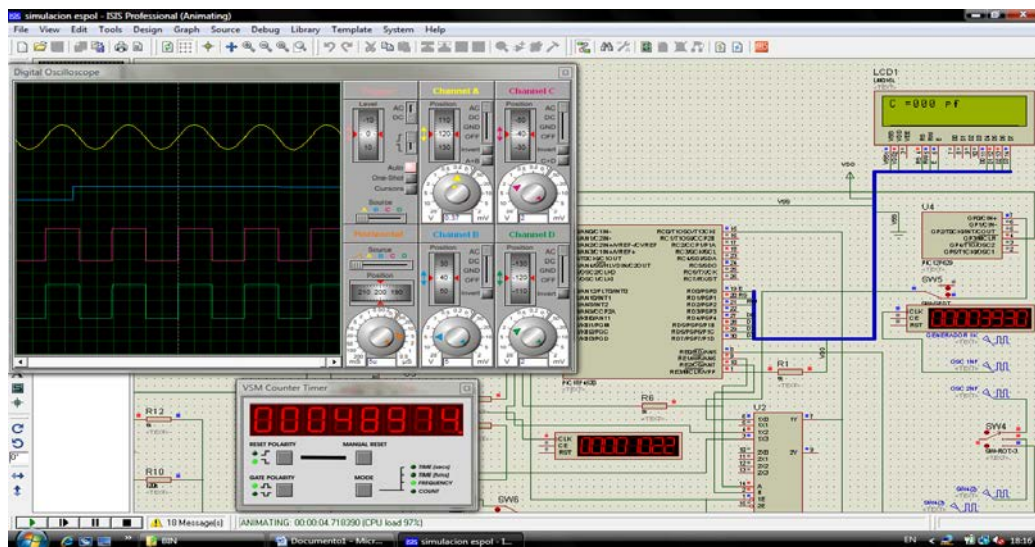
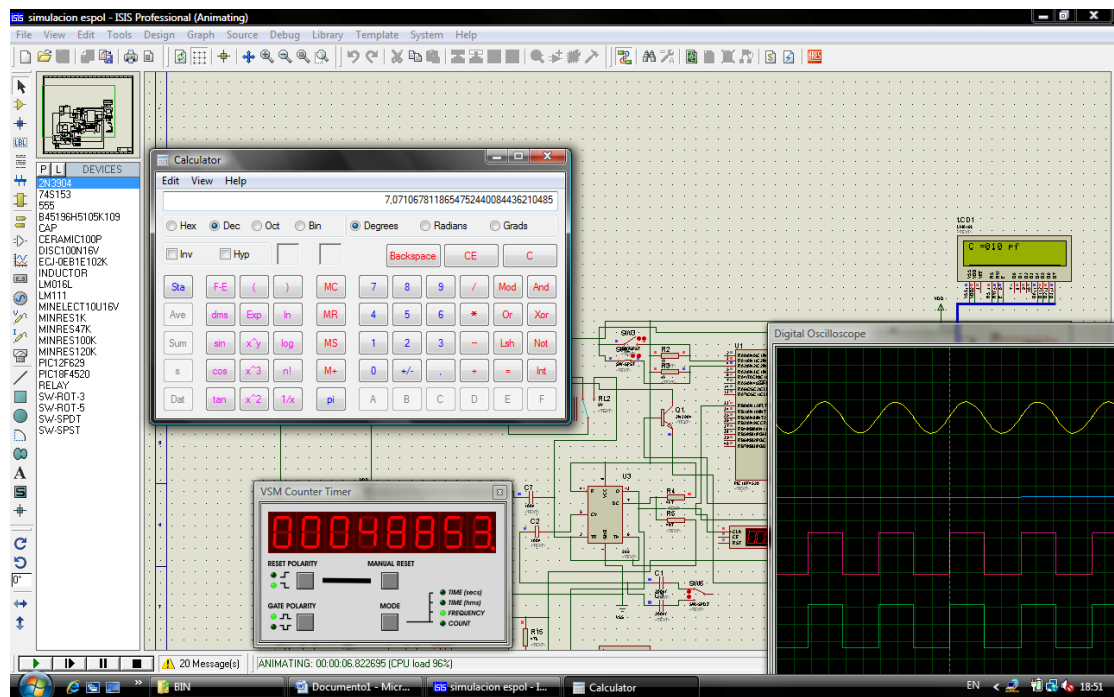


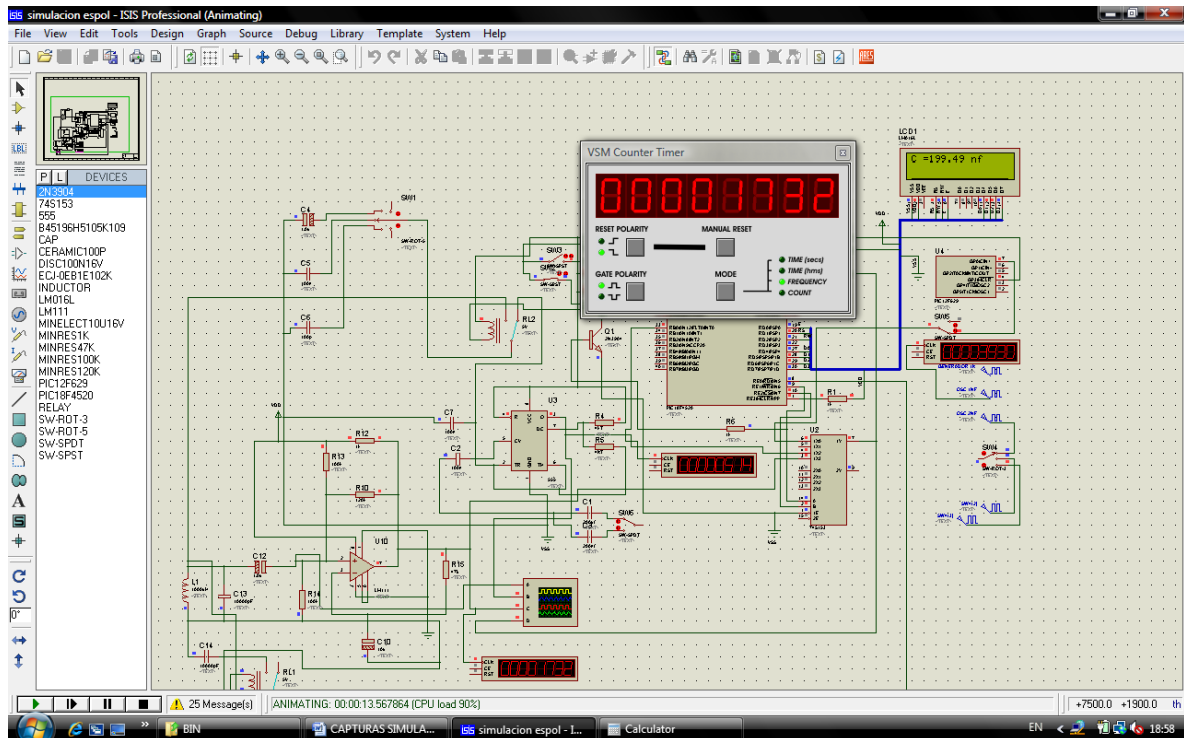
Figura 15: Circuito calibrado y listo para realizar mediciones

Ahora procedemos a conmutar un capacitor de 10 pF para poder ver el cambio en la frecuencia la misma que corresponde a 488 Khz según los cálculos lo que corresponde a la figura ahora bien,



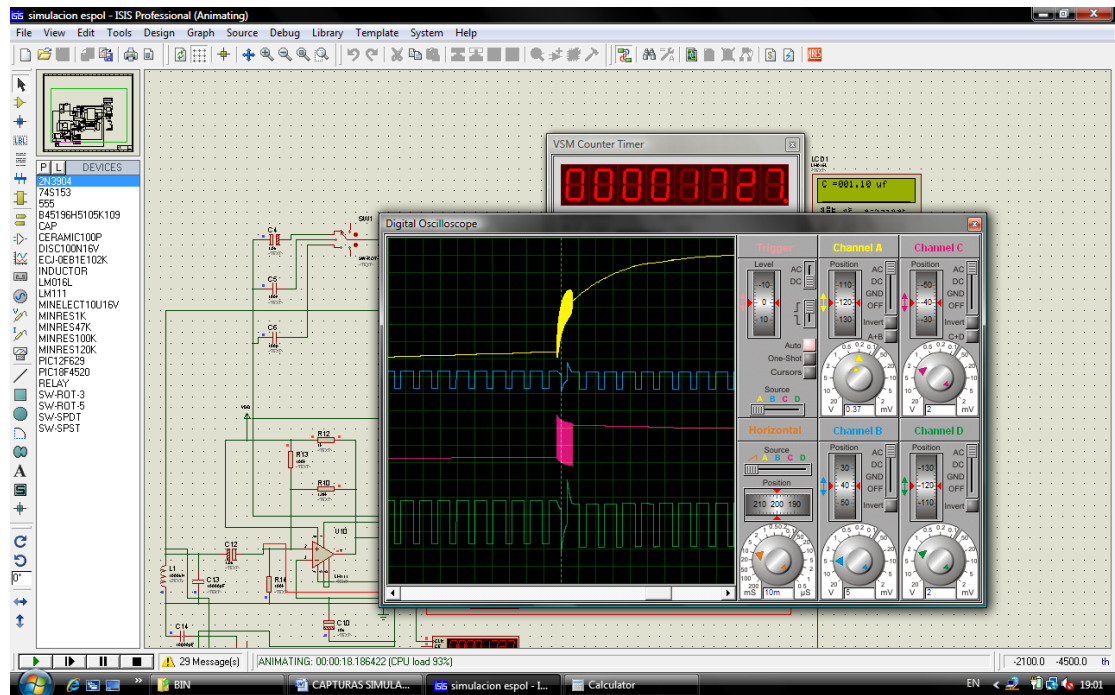
**Figura 16: Cálculo de capacitancia de 10 pF ( modo LM311)**

A continuación probamos la conmutación de circuito al poner un capacitor de 200 nF en línea con el medidor el cual nos conmuta el relé y procede a calcular la oscilación del circuito 7555 y finalmente da el valor de la frecuencia correspondiente a ese oscilador:



**Figura 17: Cálculo de una capacitancia de 200 nF (modo 7555 en modo oscilador astable)**

Y finalmente probamos la conmutación de circuito al último modo al poner un capacitor de 10 uF en línea con el medidor el cual nos conmuta el relé y procede a calcular la oscilación del circuito 7555 y finalmente da el valor de la frecuencia correspondiente a ese oscilador:



**Figura 18: Cálculo de una capacitancia de 10 uF (modo 7555 en modo contador de tiempo en alto de pulso)**

## CONCLUSIONES Y RECOMENDACIONES

Mediante la implementación física del medidor de capacitancia e inductancia podemos concluir:

1. El rango efectivo de medición del instrumento en modo capacitancia con 1 pF de resolución es desde 1 pF hasta un valor máximo de 3300 uF.
2. El rango efectivo de medición del instrumento en modo inductancia con 10 nH de resolución es desde 10 nH hasta un valor máximo de 40 mH.
3. La precisión del medidor está claramente por debajo del 5% en capacitores e inductores de precisión (1-2%) y por debajo del 10% en capacitores electrolíticos de precisión, lo que está dentro de los valores correspondientes a equipos comerciales, mas si se toma en cuenta la tolerancia de los elementos la medición tiene 0% de error.
4. Hemos ampliado notoriamente el rango del medidor LC que oferta la compañía DIY Electronics y mejorado la precisión en baja capacitancia de los modelos comerciales Fluke 867 y BK Precision 875 B.
5. Logramos desarrollar la función de autocalibración del circuito que puede ser usada en general para calcular cualquier circuito tanque de oscilación de otro proyecto similar al nuestro, y mejora considerablemente el desempeño del instrumento, pues sólo con conseguir un elemento de precisión, en este caso el capacitor de 1000 pF, se puede usar cualquier otro tipo de elemento sin importar sus características.



6. Tomamos lo mejor de dos proyectos distintos: el rango del capacímetro con el temporizador 7555, y la precisión con pequeños elementos del oscilador LM311 y los integramos en un solo instrumento con rango automático.
7. Se cumplió con los objetivos planteados en el proyecto, los cuales eran entregar una herramienta de precisión y bajo costo para estudiantes y radiotécnicos, a la par de medidores de marcas de prestigio con elementos que fácilmente se encuentran en el mercado local y de fácil reemplazo, además de contar con una interfaz gráfica sencilla y vistosa.
8. El medidor de tiempo en alto usado para determinar la capacitancia de capacitores de más de 10  $\mu\text{F}$  es de muy buena precisión y puede ser empleado en otras aplicaciones que requieran de medir el tiempo en alto de un pulso.
9. El consumo del aparato es normalmente 50 mA, en su mayor parte consumidos por la pantalla LCD y el relé de conmutación de modo, pues en el modo de baja capacitancia su consumo es de tan solo 20 mA.

Y recomendamos lo siguiente:

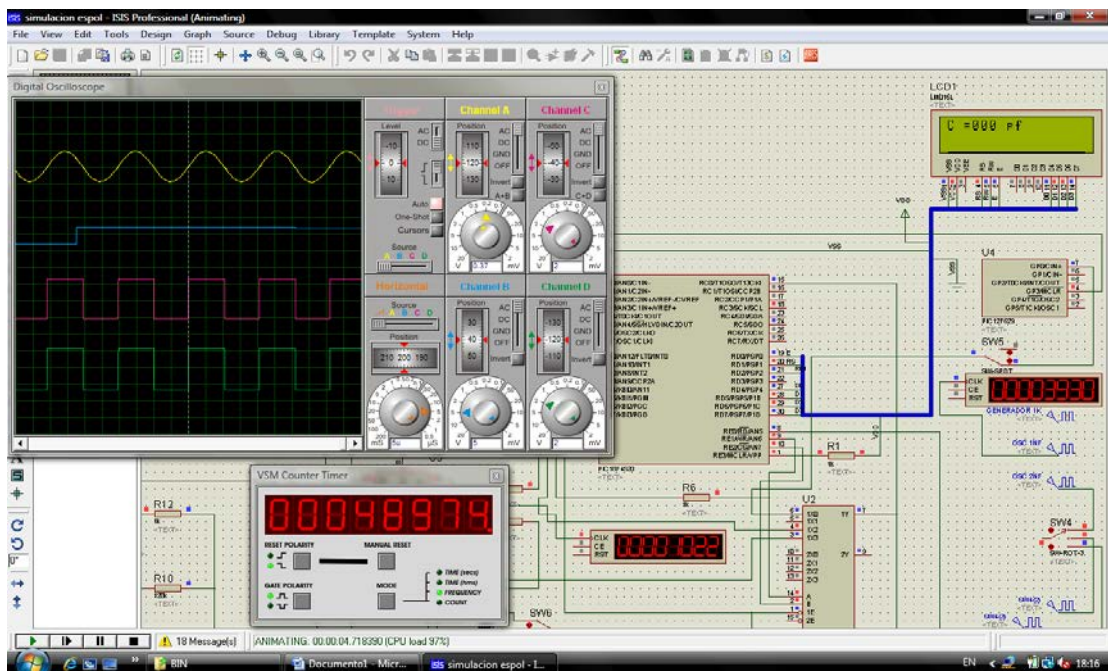
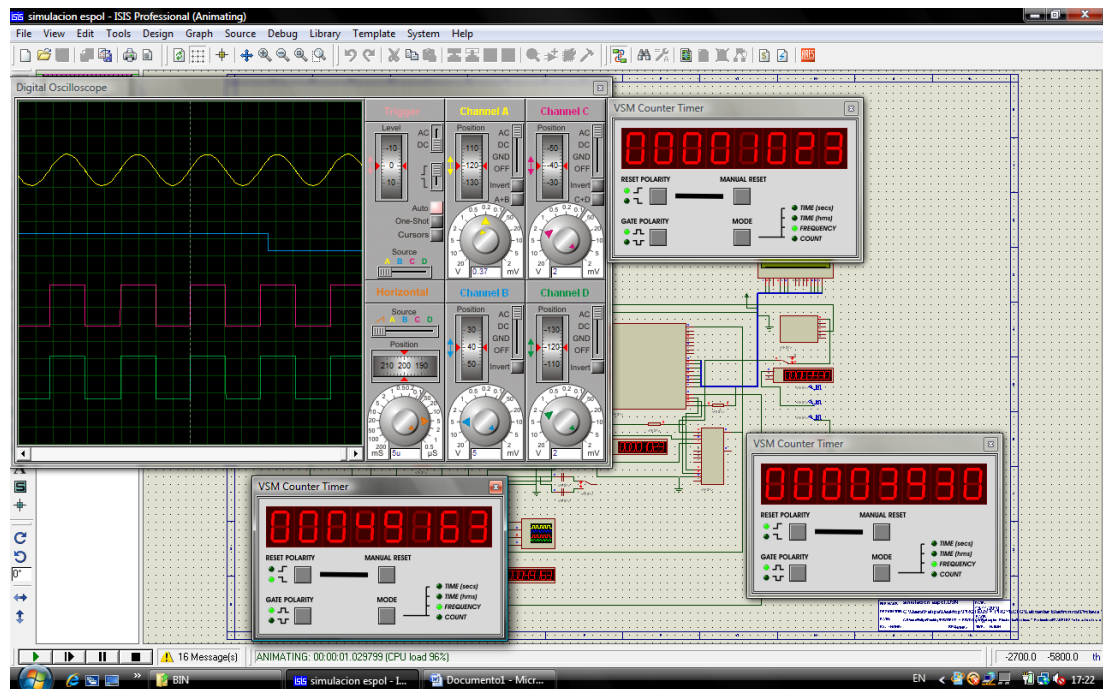
1. Usar un capacitor de muy alta calidad para la calibración es la parte clave en la precisión del circuito.
2. Recordar encender el circuito en el modo C pues de otra manera calibrará de forma errónea.
3. En caso de que el modo de capacitancia llegue a marcar algún valor distinto a 0 pF basta con cortocircuitar las tenazas de medición sin que esto represente un error o un cortocircuito que dañe el aparato.
4. En caso de obtener una medición errónea debe considerarse cambiar el capacitor pues este seguramente se encuentre deteriorado o la bobina se encuentre abierta.
5. Usar pantallas LCD de 16 caracteres que ya vengan preparados para 5 voltios, pues los normales de 4,7 voltios consumen demasiada corriente. Generalmente las pantallas azules son de 5 Voltios.
6. Si se quiere mejorar la medición del circuito en su modo de oscilador 7555 se pueden utilizar resistencias de precisión con el 1 o 2 % de tolerancia.
7. En lo posible en la construcción de otro dispositivo similar debe emplearse reed relés por su bajo consumo de corriente para alargar la vida útil de la batería, así como el uso de la función sleep.

8. Los cables de conexión entre el LM311 y las terminales de entrada deben ser lo más corto posibles para mantener las capacitancias parásitas al mínimo y asegurar la más alta precisión.

# ANEXOS

## ANEXO 1

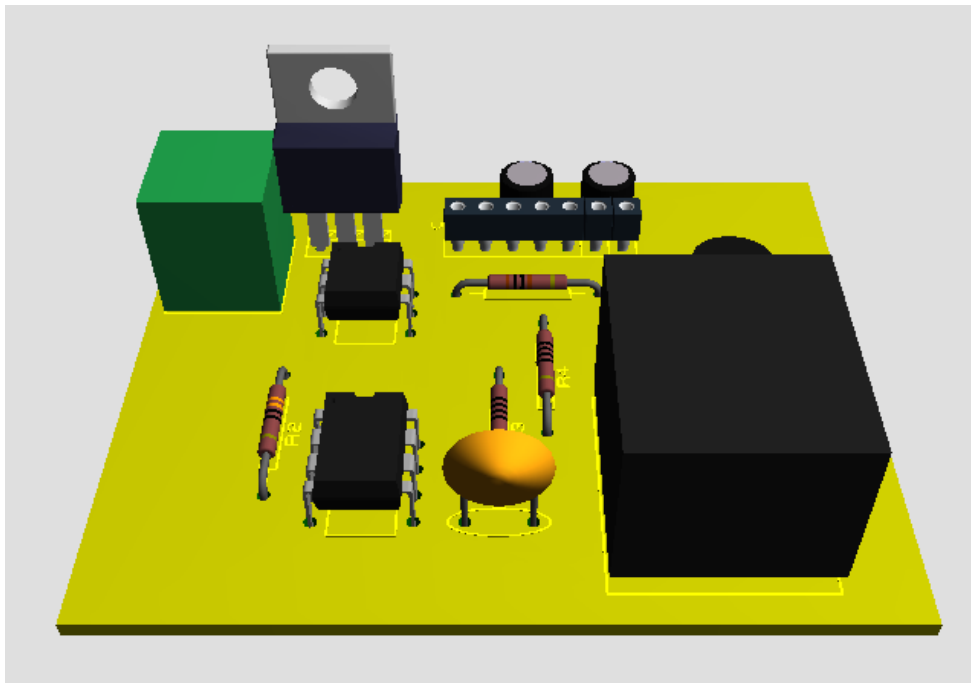
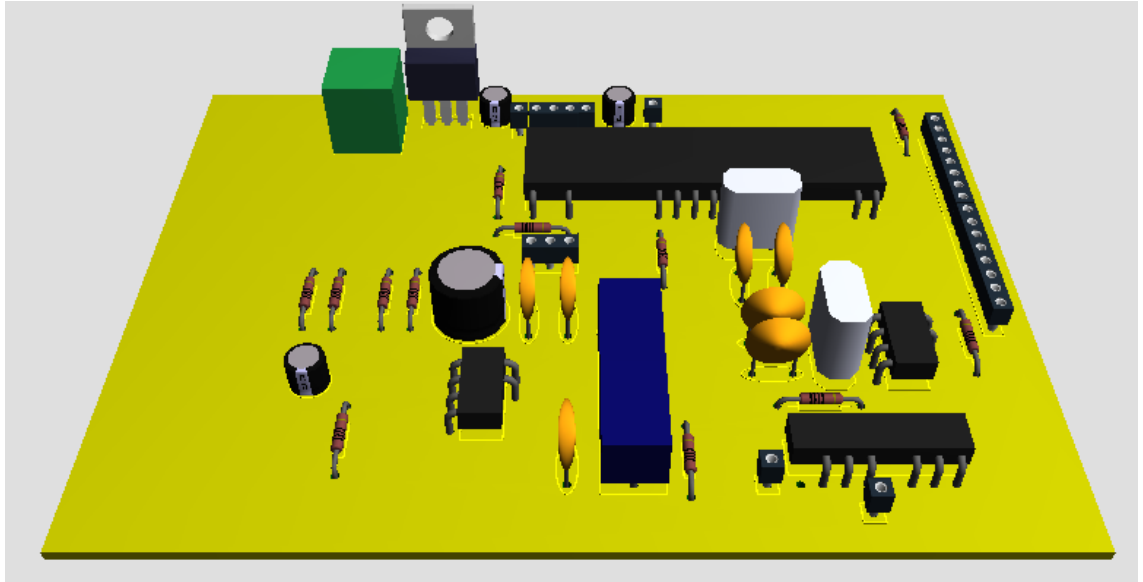
### SIMULACIÓN DEL CIRCUITO CON PROTEUS

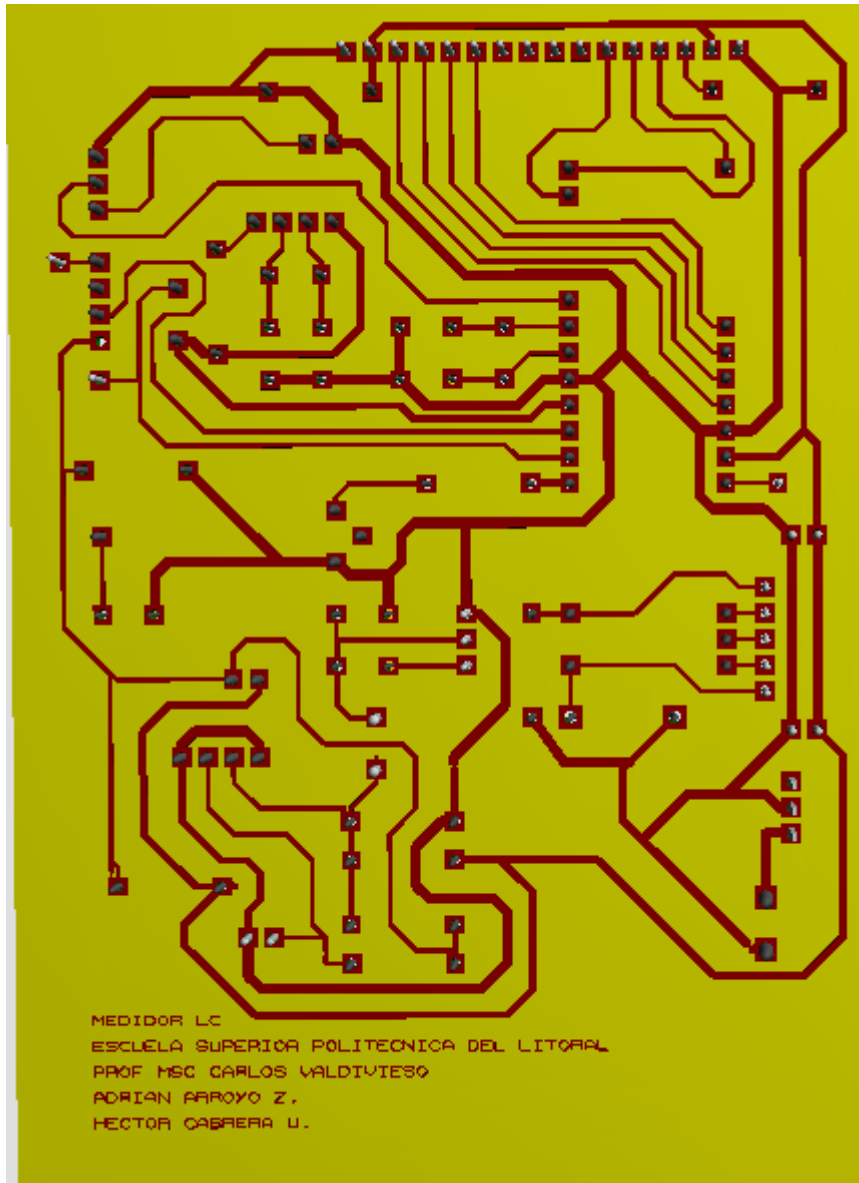


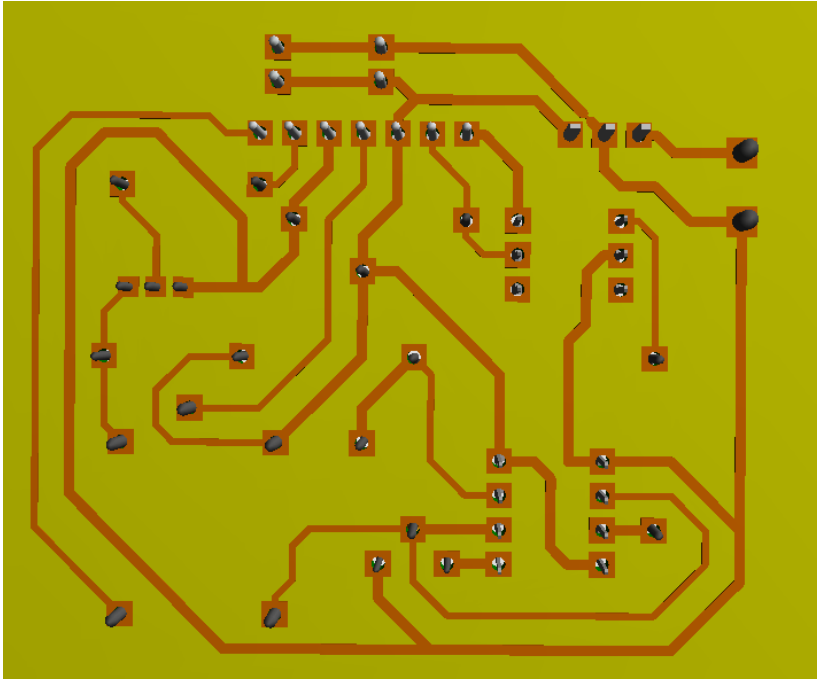
## ANEXO 2

### LISTA DE COMPONENTES

CANTIDAD	COMPONENTE	VALOR
1	MICROCONTROLADOR	PIC 18F4520
1	MICROCONTROLADOR	PIC 12F629
1	COMPARADOR	LM 311
1	OSCILADOR	NE555
1	DISPLAY	16X2
2	REGULADOR	LM7805
1	RELAY	952 A05
1	RELAY	JZC 4123 5V
1	CRISTAL	4 MHZ
1	CRISTAL	10 MHZ
1	CAPACITOR	1000PF
1	CAPACITOR	1 nf
1	INDUCTOR	100 uH
3	CAPACITOR	10 UF TANTALIO
4	CAPACITOR	10 pF
2	CAPACITOR	100 nf
3	RESISTOR	4,75 Kohm 1%
1	RESISTOR	100 ohm
1	RESISTOR	5,6 Kohm
2	RESISTOR	1 kohm
1	RESISTOR	3,3 Kohm
1	RESISTOR	47 Kohm
2	RESISTOR	2,88 Kohm
3	RESISTOR	100 Kohm

**ANEXO 3****PLACA DE PCB PROTEUS ARES VISTA 3D**

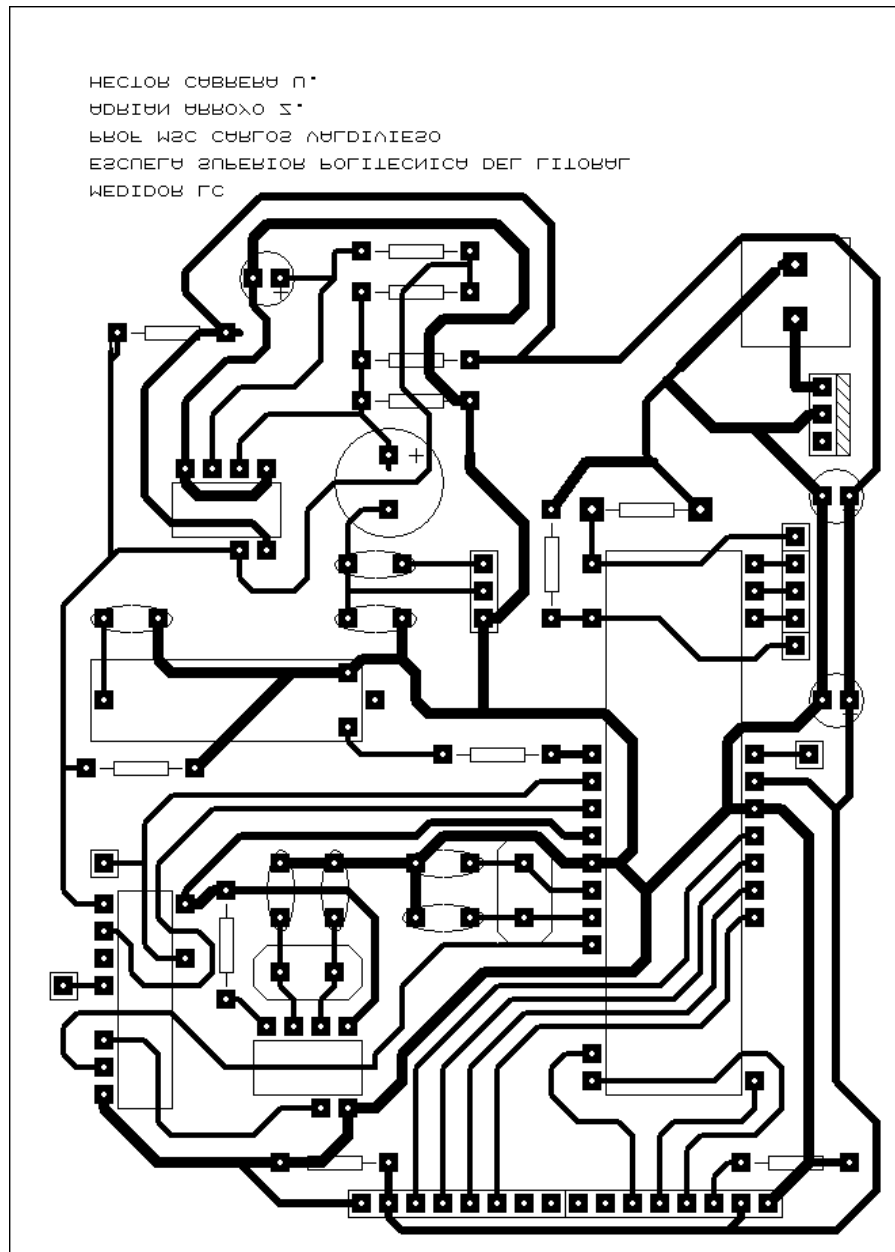
**ANEXO 4****PLACA DE PCB PROTEUS ARES ARTE EN COBRE**

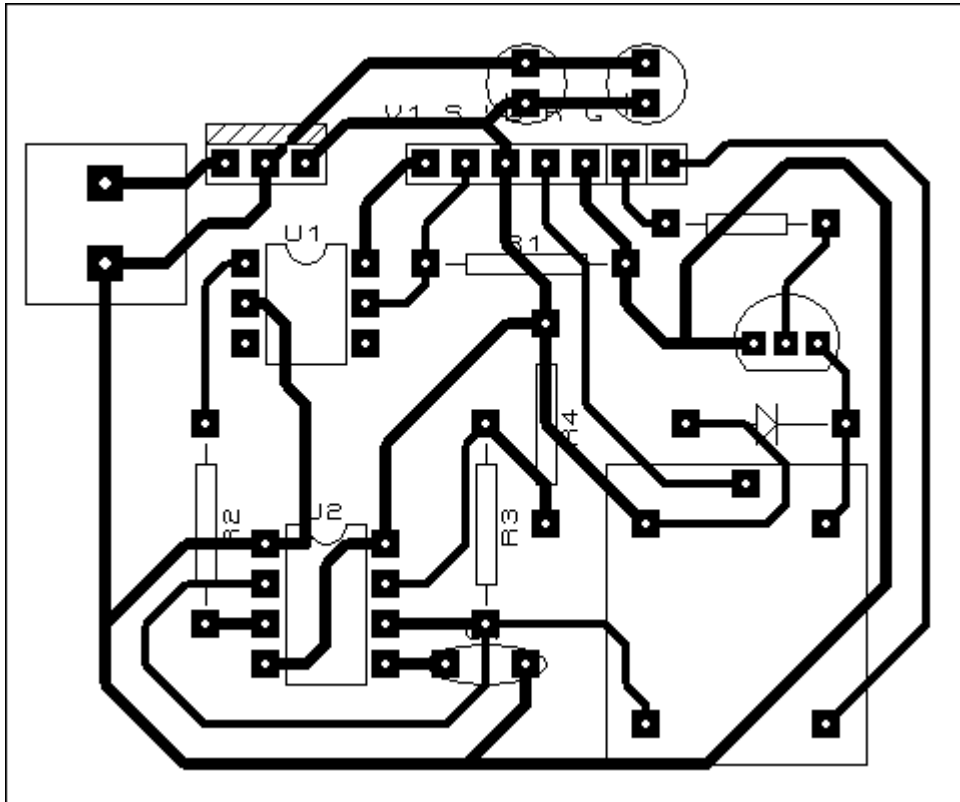




## ANEXO 5

## PLACA DE CIRCUITO IMPRESO PROTEUS ARES







## ANEXO 7

## TABLA DE RESULTADOS Y CÁLCULOS EXPERIMENTALES

capacitor	tolerancia	Frecuencia	medicion	medicion Fluke	Medicion Bk precision	% error dentro de tolerancia	% error fuera tolerancia	
40 pf	2%		41	190	21,2	2%	0%	
820 pf	2%		836	980	624	2%	0%	
470nf	10%		471	478	286	0%	0%	
100 nf	10%		96,87	99	89	4%	0%	
12 uf	10%		11	11,8	11,8	9%	0%	
4,7 uf	10%		4,65	4,63	4,6	0,50%	0%	
220 uf	10%		220	2226	214	0%	0%	
1000 uf	10%		985	983	1002	1,50%	0%	
3300 uf	10%		3364	3300	2950	2%	0%	

## BIBLIOGRAFÍA

### SITIOS WEB

1. Medidor LC - NANO1985;  
<http://www.todopic.com.ar/foros/index.php?topic=26840.0>; **Fecha de consulta:** 20 de Agosto de 2010
2. Teoría Medidor LC - MICROS DESIGNS;  
<http://www.micros-designs.com.ar/medidor-lc/>; **Fecha de consulta:** 21 de Agosto de 2010
3. Construcción Medidor LC - AMATEUR RADIO;  
<http://ironbark.bendigo.latrobe.edu.au/~rice/lc/>; Fecha de consulta: 1 de Septiembre de 2010
4. Manual PICKit3 – PIC LINUX;  
<http://pic-linux.foroactivo.net/programadores-f12/pickit3-tutorial-espanol-t172.htm>; **Fecha de consulta:** 2 de Septiembre de 2010
5. Manual de usuario PICKit3 – MICROCHIP  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en538340](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en538340); **Fecha de consulta:** 2 de Septiembre de 2010
6. Manual de referencia de CCS – CUSTOM COMPUTER SERVICES  
[http://www.ccsinfo.com/downloads/ccs\\_c\\_manual.pdf](http://www.ccsinfo.com/downloads/ccs_c_manual.pdf); **Fecha de consulta:** 2 de Septiembre de 2010

7. Técnicas de medición de capacitancias – UNIVERSIDAD DE GUADALAJARA  
[http://148.202.148.5/cursos/17721/Modulo\\_1/1p3/1p3p2/tema1p3p2.htm](http://148.202.148.5/cursos/17721/Modulo_1/1p3/1p3p2/tema1p3p2.htm); **Fecha de consulta:** 3 de Septiembre de 2010
  
8. Osciladores LC – OSCAR SANTA CRUZ;  
<http://www.profesores.frc.utn.edu.ar/electronica/ElectronicaAplicadaIII/Aplicada/Cap01Osciladores2parte.pdf>; **Fecha de consulta:** 2 de Septiembre de 2010
  
9. Medición de tiempo en alto de pulso utilizando intext – TODOPIC;  
<http://www.todopic.com.ar/foros/index.php?topic=25070.0>; **Fecha de consulta:** 10 de Septiembre de 2010
  
10. Medición de ancho de pulso – PETER H. ANDERSON;  
[http://www.phanderson.com/PIC/PICC/CCS\\_PCM/pulse.htm](http://www.phanderson.com/PIC/PICC/CCS_PCM/pulse.htm); **Fecha de consulta:** 10 de Septiembre de 2010