

OBTENCIÓN DE LOS PRINCIPALES PARÁMETROS DE LAS SEÑALES DE
TENSIÓN Y CORRIENTE PRESENTES EN UN SISTEMA MONOFÁSICO
UTILIZANDO UN SISTEMA DIGITAL

KEVIN ANDRÉS POLO JARAMILLO
WILSON JAIR RODRÍGUEZ LEÓN

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA

2015

OBTENCIÓN DE LOS PRINCIPALES PARÁMETROS DE LAS SEÑALES DE
TENSIÓN Y CORRIENTE PRESENTES EN UN SISTEMA MONOFÁSICO
UTILIZANDO UN SISTEMA DIGITAL

KEVIN ANDRÉS POLO JARAMILLO
WILSON JAIR RODRÍGUEZ LEÓN

Trabajo de Grado para optar al título de
Ingeniero Electrónico

Director

JAIME GUILLERMO BARRERO PEREZ
Magister en Potencia Eléctrica (MPE)

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA
2015

RESUMEN

TÍTULO: OBTENCIÓN DE LOS PRINCIPALES PARÁMETROS DE LAS SEÑALES DE TENSIÓN Y CORRIENTE PRESENTES EN UN SISTEMA MONOFÁSICO UTILIZANDO UN SISTEMA DIGITAL *

AUTORES: KEVIN ANDRÉS POLO JARAMILLO**
WILSON JAIR RODRÍGUEZ LEÓN

PALABRAS CLAVE: DSPIC, microcontrolador, sistema, parámetro, frecuencia, FFT,

DESCRIPCIÓN:

Los sistemas embebidos se han convertido rápidamente en una solución factible para muchos de los problemas o situaciones que a diario enfrentan estudiantes y profesionales, frecuentemente el mercado presenta grandes innovaciones en los recursos disponibles en una sola tarjeta de desarrollo, por ejemplo la familia dsPIC33FJXXXGPX06A / X08A / X10A utiliza una potente arquitectura de 16 bits que reúne las características de control de un microcontrolador (MCU) con las capacidades computacionales de un procesador digital de señales (DSP). Con este sistema de desarrollo se plantea el diseño de una herramienta portátil de bajo costo con la capacidad de obtener los principales parámetros eléctricos de las señales de tensión y corriente presentes en un sistema monofásico.

En este proyecto se presenta una configuración de hardware, entre varias que podrían escogerse, con el cual se atenúan señales provenientes del generador de señales con hasta 12[V_{pp}] a valores de tensión entre 0 y 3.3 [V]. Se describen los aspectos principales de una rutina con los cuales se puede obtener algunos parámetros eléctricos de las señales muestreadas, como su valor promedio, valor eficaz y valores máximo y mínimo, además se realiza un análisis en frecuencia a partir de la transformada rápida de Fourier (FFT), mediante esta se logra la construcción del espectro en frecuencia de la señal y se encuentra su correspondiente distorsión armónica total (THD).

El resultado final es un dispositivo-prototipo confiable de medición dentro de los estándares planteados, con el cual es posible visualizar en dos canales los parámetros de medición mencionados, y sobre el cual los usuarios de la Universidad Industrial de Santander posean total autonomía para consultar su programación, estudiar la forma en que se realizan los cálculos o adicionar otros parámetros de medida requeridos.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez, Ingeniero Electrónico

ABSTRACT

TITLE: OBTAINING THE MAIN PARAMETERS OF VOLTAGE AND CURRENT SIGNALS PRESENT IN A SINGLE PHASE SYSTEM USING A DIGITAL SYSTEM *

AUTHORS: KEVIN ANDRÉS POLO JARAMILLO**
WILSON JAIR RODRÍGUEZ LEÓN

KEYWORDS: dsPIC, microcontroller, system, parameter, frequency, FFT.

DESCRIPTION:

Embedded systems have quickly become a feasible solution for many of the problems or situations that students and professionals face on a daily basis. Frequently the market presents great innovations in available resources on a single card development, for example the family dsPIC33FJXXXGPX06A / X08A / X10A uses a powerful 16-bit architecture that combines the features of a microcontroller (MCU) control with the computational capabilities of a digital signal processor (DSP). This development system considers the design of a low cost portable tool with the ability to get the main electrical parameters in a single-phase system voltage and current signals.

This project shows a hardware configuration, among several that could be chosen, with which signals coming from the generator of signals are attenuated up to 12 [Vpp] values of voltages between 0 and 3.3 [V]. The main aspects of a routine are described. It is useful to get some electrical parameters of sampled signals, as its average value, RMS and maximum and minimum values. In addition, an analysis in frequency from the fast Fourier transform (FFT) to achieve the signal frequency construction of the spectrum and its corresponding total harmonic distortion (THD) is located.

The final result is a reliable measurement prototyping-tool that meet raised standards, and has the possibility to show in two channels, the mentioned measurement parameters, on which the users of the Universidad Industrial de Santander have total autonomy to check their schedule, see how calculations are made, or having the possibility to add other required measurement parameters.

* Bachelor Thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez, Ingeniero Electrónico

DEDICATORIA

*Dedicado especialmente a la memoria de Delcy Jaramillo y Jacqueline León,
Hermosas mujeres que empezaron este viaje con nosotros,
Atentas y amorosas madres quienes en presencia
Nos dieron su apoyo y aún hoy en ausencia
Nos llenan de motivación para continuar en este camino.*

CONTENIDO

	pág.
INTRODUCCIÓN	17
1. MARCO TEÓRICO	18
1.1. PARAMETROS ELÉCTRICOS CARACTERÍSTICOS	19
1.2. TRANSFORMADA DISCRETA DE FOURIER	22
1.3. TRANSFORMADA RÁPIDA DE FOURIER	25
1.4. TEOREMA DEL MUESTREO	27
2. DESCRIPCIÓN DEL SISTEMA	28
2.1. DSPIC 33FJ256GP710A	28
2.2. SISTEMA DE DESARROLLO	29
2.3. COMPILACIÓN	30
2.4. PROGRAMACIÓN	31
2.5. INTERFAZ GRÁFICA	33
3. HARDWARE	34

3.1. CIRCUITO DE ALIMENTACIÓN Y REGULACIÓN DE TENSIÓN	34
3.2. ACONDICIONAMIENTO DE LA SEÑAL	35
4. SOFTWARE	38
4.1. LECTURA DE DATOS	41
4.2. PROCESAMIENTO DE DATOS EN EL TIEMPO	44
4.3. PROCESAMIENTO DE DATOS EN FRECUENCIA	52
5. ANÁLISIS Y RESULTADOS	54
5.1. PRUEBAS DEL SISTEMA	56
5.1.1. Pruebas para el sistema en el tiempo	56
5.1.2. Pruebas para el sistema en frecuencia	65
6. CONCLUSIONES	68
7. RECOMENDACIONES	69
8. BIBLIOGRAFÍA	70
ANEXOS	72

LISTA DE FIGURAS

	pág.
Figura 1. Esquema del dsPIC 33FJ256GP710A	28
Figura 2. Sistema de desarrollo Mikromedia para dsPIC33®	29
Figura 3. Diagrama de pines del sistema de desarrollo Mikromedia para dsPIC33®	30
Figura 4. Interfaz de mikroC PRO for dsPIC30/33 and PIC24	32
Figura 5. Programador mikroProg™ de MikroElektronika®	32
Figura 6. Interfaz de mikroProg suite™ para PIC®	32
Figura 7. Interfaz de Visual TFT de MikroElektronika®	33
Figura 8. Esquema de fuente de alimentación y reguladores de tensión	34
Figura 9. Esquema general del acondicionamiento de la señal	35
Figura 10. Topología de la etapa de acondicionamiento de señal	36
Figura 11. Respuesta en frecuencia de la salida del filtro pasabaja diseñado	37
Figura 12. Señal de entrada frente a su equivalente acondicionada	38

Figura 13. Rutina general del programa implementado	39
Figura 14. Descripción paso a paso del programa implementado	40
Figura 15. Código para configuración manual ADC	41
Figura 16. Código para configuración de frecuencia de muestreo	42
Figura 17. Ciclo <i>for</i> para toma de muestras	43
Figura 18. Muestras tomadas simultáneamente	44
Figura 19. Código para búsqueda de máximos y mínimos	45
Figura 20. Código para hallar valor medio de la señal.	46
Figura 21. Código para cálculo de parámetros dependientes del periodo	47
Figura 22. Distribución de pantalla para visualización de señal y parámetros	49
Figura 23. Código para impresión en pantalla gráfica	50
Figura 24. Localización del punto (0,0) en la pantalla gráfica	51
Figura 25. Funciones para imprimir en pantalla	52
Figura 26. Código rutina para transformada rápida de Fourier	53
Figura 27. Esquema de la implementación del trabajo de grado	54

Figura 28. Osciloscopio GDS – 2062 de GW Instek	56
Figura 29. Onda tipo seno de 60 [Hz] y amplitud 3 [V] leída por el osciloscopio	57
Figura 30. Visualización de onda seno de 60 [Hz] y amplitud 3 [V] en el sistema	57
Figura 31. Onda cuadrada de 60 [Hz] y amplitud 3 [V] leída por el osciloscopio	60
Figura 32. Visualización de onda cuadrada de 60 [Hz] y 3 [V] de amplitud en sistema	60
Figura 33. Visualización de los dos canales del sistema para cálculo de potencia	62
Figura 34. Visualización de las señales de tensión y corriente en el osciloscopio	63
Figura 35. Parámetros de potencia observadas en el sistema	64
Figura 36. Espectro de una señal cuadrada de 60 [Hz] visualizada en el sistema.	66
Figura 37. Espectro de frecuencia una señal cuadrada de 395 [Hz] visualizada en el sistema.	67

LISTA DE TABLAS

	pág.
Tabla 1. Frecuencias de muestreo implementadas.	43
Tabla 2. Medida de frecuencia para una señal sinusoidal de 60 [Hz] con amplitud variable.	57
Tabla 3. Medida de valor eficaz, máximo y mínimo de una señal sinusoidal de 60 [Hz] con amplitud variable.	58
Tabla 4. Medida de valor promedio de una señal sinusoidal de 60 [Hz] con amplitud variable.	59
Tabla 5. Medida de frecuencia para una señal cuadrada de 60 [Hz] con amplitud variable.	60
Tabla 6. Medida de valor eficaz, máximo y mínimo de una señal cuadrada de 60 [Hz] con amplitud variable.	61
Tabla 7. Medida de valor promedio de una señal sinusoidal de 60 [Hz] con amplitud variable.	62
Tabla 8. Potencia de tensión y corriente a 60 [Hz].	63
Tabla 9. Medidas tomadas a tres señales sinusoidales en diferentes frecuencias	65

Tabla 10. Valores obtenidos del espectro de una señal cuadrada de 60 [Hz] 66

Tabla 11. Valores obtenidos del espectro de una señal cuadrada de 395 [Hz] 67

INTRODUCCIÓN

Actualmente en el mercado existe una gran cantidad de sistemas digitales de capacidad variable, ideales para la solución de muchas de las tareas específicas a la que estudiantes y profesionales se ven enfrentados día a día. Con el pasar del tiempo, los fabricantes han logrado integrar en una sola tarjeta diferentes componentes de gran desempeño, ofreciendo así una herramienta poderosa que puede ser programada con lenguajes de bajo, medio y alto nivel a través de compiladores, además en muchos casos ofrece diversas funciones que permiten al programador interactuar de manera rápida y sencilla con los periféricos disponibles en el sistema.

El presente trabajo de grado se cuenta con un sistema digital, basado en dsPIC, dispositivo que combina las características de un procesador digital de señales (DSP por sus siglas *Digital Signal Processor*) con un microcontrolador, con el que se pretende desarrollar un instrumento capaz de procesar y encontrar los parámetros eléctricos característicos de las señales de tensión y corriente presentes en un sistema monofásico (previa adecuación del usuario para su lectura).

Los parámetros a que se refiere este trabajo corresponde a valores máximos y mínimo, valor promedio, valor eficaz o RMS (*Root Mean Square*), frecuencia, transformada rápida de Fourier o FFT (*Fast Fourier Transform*), THD (*Total Harmonic Distorsion*) entre otros, cuya definición y modelo de implementación es presentado en el capítulo uno.

En el capítulo dos se describe de manera detallada todos los componentes que integran el sistema de desarrollo, las diferentes herramientas en hardware como en

software que resulta necesario su adquisición para la correcta implementación del sistema.

Para definir las características de las señales que el instrumento podrá censar, se diseñó y construyó un hardware que además de integrarse correctamente con la tarjeta, proporciona la energía y la extensión de los periféricos con los que el usuario interactúa, dicho hardware es presentado en el capítulo tres.

Con la adquisición de las herramientas antes mencionadas se procede a crear una rutina donde procesan y visualizan las señales muestreadas, se obtienen los parámetros mencionados y se configuran botones en pantalla para que el usuario pueda mejorar la visualización de la señal muestreada, los detalles de esta rutina son presentados en el capítulo cuatro.

Finalmente, en el quinto y último capítulo se muestra el sistema funcionando, se define el alcance en tensión y frecuencia para distintos tipos de onda, adicionalmente se compara los resultados obtenidos en el sistema con el patrón elegido y valores teóricos con la finalidad de encontrar el porcentaje de error en los cálculos de los parámetros eléctricos de la señal de entrada.

1. MARCO TEÓRICO

1.1. PARAMETROS ELÉCTRICOS CARACTERÍSTICOS

La obtención de los parámetros característicos de las señales eléctricas permite la recolección de información del comportamiento de dichas señales, además de contrastar los valores encontrados teóricamente con los valores observados en la vida real.

Entre los principales parámetros de una señal eléctrica se encuentran la frecuencia, los valores máximos y mínimos que toma la onda, valor medio o *average*, valor eficaz o *rms*, entre otros. A continuación se presenta una breve definición de cada uno de los parámetros nombrados anteriormente:

- Frecuencia: Número de ciclos que se repite la onda en la unidad de tiempo. El ciclo es la parte de una onda en un intervalo de tiempo equivalente a un período.

Cuando una señal es muestreada, el conversor analógico digital toma una lectura de la señal de entrada cada T_s segundos, a este intervalo de tiempo se le conoce como periodo de muestreo, este valor normalmente es conocido por el programador y puede ser modificado cambiando el valor de algunos registros como se verá más adelante.

Para encontrar la frecuencia de una señal muestreada hay que determinar la duración de un ciclo de la señal, o el número de muestras requerido para almacenar un ciclo completo de la señal en el vector, uno de los métodos utilizados para identificar dichos ciclos es el cruce por cero, ya que en cada ciclo las señales sinusoidales, cuadradas y triangulares estándares, interceptan el eje "x" en 3 ocasiones, una vez identificadas la cantidad de muestras en que están albergadas

un ciclo de la señal, la frecuencia de la señal corresponde al inverso del producto entre el número de muestras y el período de muestreo.

- Valor medio: Corresponde al promedio de la señal de tensión o corriente a través del tiempo. Se define como la relación entre la suma de los valores instantáneos de la señal y el período de la onda.

$$V_{\text{avg}} = \frac{1}{T} \int_0^T v_o(t) dt [V] \quad \text{Ecuación 1.}$$

Para la correcta implementación de esta ecuación en el sistema digital es necesario discretizar la integral, interpretando está como una suma finita de términos, dichos términos corresponde al producto entre una muestra y el periodo de muestreo de la señal, este periodo de muestreo representa el “espacio” o “distancia” en tiempo que hay entre las muestras.

$$V_{\text{avg}} = \frac{1}{T} * \sum_0^T V_o[n] * \Delta t [V] \quad \text{Ecuación 2.}$$

$$V_{\text{avg}} = \frac{\Delta t}{T} * \sum_0^T V_o[n] [V] \quad \text{Ecuación 3.}$$

$$T = N * \Delta t [\text{Segundos}] \quad \text{Ecuación 4.}$$

Donde Δt corresponde al período de muestreo, T al período de la señal y N corresponde al número de muestras que contienen un período de la señal.

$$V_{\text{avg}} = \frac{1}{N} * \sum_0^T V_o[n] \text{ [V]} \quad \text{Ecuación 5.}$$

- Valor medio cuadrático (*rms*): Corresponde al valor de tensión o corriente que produce igual potencia que el nivel equivalente de cd. Se define como la raíz cuadrada de la suma de los cuadrados de los valores instantáneos de la señal en un período.

$$V_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T v_o^2(t) dt} \text{ [V]} \quad \text{Ecuación 6.}$$

Al igual que con el valor promedio es necesario realizar la discretización de la integral esta vez teniendo cuidado con el cuadrado de las muestras, y manteniendo la sumatoria dentro de la raíz cuadrada.

$$V_{\text{rms}} = \sqrt{\frac{1}{T} * \sum_0^T V_o^2[n] * \Delta t} \text{ [V]} \quad \text{Ecuación 7.}$$

$$V_{\text{rms}} = \sqrt{\frac{\Delta t}{T} * \sum_0^T V_o^2[n]} \text{ [V]} \quad \text{Ecuación 8.}$$

$$V_{\text{rms}} = \sqrt{\frac{1}{N} * \sum_0^T V_o^2[n]} \text{ [V]} \quad \text{Ecuación 9.}$$

- Potencia eficaz: Hace referencia al producto de los valores eficaces de tensión y corriente en la salida.

$$S = V_{\text{rms}} * I_{\text{rms}} \text{ [VA]} \quad \text{Ecuación 10.}$$

- Potencia promedio: calculada como el promedio del producto entre la tensión y corriente instantánea en un periodo de la señal.

$$P_{\text{avg}} = \frac{1}{T} \int_0^T v_o(t) * i_o(t) dt \text{ [W]} \quad \text{Ecuación 11.}$$

$$P_{\text{avg}} = \frac{1}{T} \sum_0^T V_o[n] * I_o[n] * \Delta t \text{ [W]} \quad \text{Ecuación 12.}$$

$$P_{\text{avg}} = \frac{1}{N} \sum_0^T V_o[n] * I_o[n] \text{ [W]} \quad \text{Ecuación 13.}$$

- Factor de potencia: definida como la proporción entre las potencias real o promedio (activa) y la potencia aparente. [1]

$$f. p = \frac{S}{P_{\text{avg}}} \quad \text{Ecuación 14.}$$

1.2. TRANSFORMADA DISCRETA DE FOURIER

La transformada discreta de Fourier mejor conocida como DFT (*Discrete Fourier Transform*) se trata de una herramienta importante para conocer y estudiar el espectro de frecuencia de una señal muestreada.

$$x[n] = \frac{1}{2\pi} \int_{2\pi} X(e^{jw}) e^{jwn} dw \quad \text{Ecuación 15.}$$

$$X(e^{jw}) = \sum_{n=-\infty}^{+\infty} x[n] e^{-jwn} \quad \text{Ecuación 16.}$$

La función $X(e^{jw})$ se conoce como la transformada de Fourier en tiempo discreto. Las ecuaciones descritas anteriormente son conocidas como el par de transformada de Fourier estas corresponden a la ecuación de síntesis y la ecuación de análisis respectivamente [2].

La ecuación de síntesis es una representación de la serie $x[n]$ como una combinación lineal de exponenciales complejas cercanas en frecuencia y con amplitudes de $X(e^{jw})(dw/2\pi)$. Por tal motivo a menudo se hace referencia a la transformada discreta de Fourier como el espectro de $x[n]$, debido a que proporciona la información relacionada de cómo $x[n]$ está compuesta de valores de tipo exponencial complejas a frecuencias distintas.

Debido a que la sumatoria es infinita, resulta inconveniente a la hora de implementar una rutina de código de la DFT, así que se elige cierta tasa para muestrear la señal, de modo que la forma de onda en el tiempo pasa a ser una señal discreta, se puede hallar la transformada discreta de Fourier a través de:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{nk(-j2\pi/N)} \quad \text{Ecuación 17.}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{-nk(-j2\pi/N)} \quad \text{Ecuación 18.}$$

Las variables k y n varían entre cero y $N - 1$, N corresponde al número de muestras que se toma de la señal. Para tener una ecuación más compacta y conveniente, el término de la sumatoria se reemplaza por 'W', también conocida como factor *twiddle*:

$$W = e^{-j2\pi/N} \quad \text{Ecuación 19.}$$

Teniendo como resultado el par de ecuaciones de la DFT

$$X(k) = \sum_{n=0}^{N-1} x[n] W^{nk} \quad \text{Ecuación 20.}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W^{-nk} \quad \text{Ecuación 21.}$$

Cabe decir que el cálculo de la DFT requiere una cantidad elevada de operaciones de sumas y productos donde el factor que influye directamente es el número de muestras con que se elige trabajar, en la medida que N sea un valor grande, el tiempo que se espera en la salida y la cantidad de operaciones (N^2) aumenta considerablemente. Por tal razón se buscan algoritmos que ofrezcan mayor rapidez en el cálculo de la DFT, llegando de este modo a la transformada rápida de Fourier (FFT por sus siglas en *Fast Fourier Transform*).

1.3. TRANSFORMADA RÁPIDA DE FOURIER

La Transformada Rápida de Fourier (*FFT*) se considera una herramienta esencial en el tratamiento digital de señales. En 1965 J.W.Cooley y J.W Tukey estudiaron por primera vez el problema en programación de un algoritmo capaz de calcular series complejas.

La FFT consiste en un algoritmo para el cálculo de la Transformada Discreta de Fourier (DFT). Este algoritmo adquiere importancia gracias a que elimina una gran cantidad de cálculos repetitivos presentados en la DFT, logrando de esta forma más rapidez en el cálculo.

En la Transformada Discreta de Fourier obtener $X(k)$ para un 'k' específico requiere aproximadamente N sumas complejas y N productos complejos, ya que:

$$X(k) = \sum_{n=0}^{N-1} x[n] W^{nk} \quad \text{Ecuación 22.}$$

$$X(k) = x[0] + x[1]W^k + x[2]W^{2k} + \dots + x[N - 1]W^{(N-1)k} \quad \text{Ecuación 23.}$$

Donde $k = 0, 1, 2, \dots, N - 1$. Para obtener la serie $X(0), X(1), X(2), \dots, X(N - 1)$, se necesitan un total de N^2 sumas complejas y N^2 productos complejos. Esto se traduce en requerimientos computacionales excesivos y cálculos tediosos fundamentalmente cuando se tiene el número de muestras 'N' grande, influyendo directamente en el tiempo de respuesta en la salida.

En el trabajo de reducir tiempos, la FFT aprovecha la periodicidad y simetría del factor *twiddle* 'W' para el cálculo del Transformada Discreta de Fourier. La periodicidad de 'W' y la simetría implican respectivamente:

$$W^k = W^{k+N} \quad \text{Ecuación 24.}$$

$$W^k = -W^{k+N/2} \quad \text{Ecuación 25.}$$

La FFT se encarga de descomponer la DFT de N puntos en transformadas más pequeñas. Una DFT de N puntos es descompuesta en dos DFT's de (N/2) puntos, estas a su vez son descompuestas en DFT's de (N/4) puntos, obteniendo un total de 4 DFT's, en pasos sucesivos hasta llegar a tener (N/2) DFT's de dos puntos cada una. Por tal motivo N debe corresponder a una potencia de dos.

Puede demostrarse que cualquier forma de onda periódica puede representarse como la sumatoria de una serie de ondas sinusoidales de distintas frecuencias y fase, conformando el espectro armónico de la onda más el valor medio. La sección correspondiente a la DFT y FFT está basada en [3] y [4].

La frecuencia de la onda predominante se conoce como frecuencia fundamental, y las frecuencias restantes son llamadas armónicos, correspondientes a múltiplos enteros de la frecuencia fundamental. De ahí que se considere una herramienta importante en el desarrollo de este proyecto, ya que conociendo el espectro de frecuencia, se puede encontrar la distorsión armónica total, también llamada THD de la señal de entrada,

$$\%THD = \sqrt{\frac{H_2^2 + H_3^2 + \dots + H_n^2}{H_1^2}} * 100 \quad \text{Ecuación 26.}$$

H_1 corresponde al armónico fundamental y H_2, H_3, \dots, H_n , son los armónicos que conforman el espectro de la señal. Para el caso particular de una onda cuadrada de amplitud A, el valor de sus armónicos corresponde a:

$$H_n = \frac{4 * A}{n * \pi}$$

Ecuación 27.

1.4. TEOREMA DE MUESTREO

Oppenheim presenta la definición del teorema de muestreo como se presenta a continuación [5]:

Sea $x(t)$ una señal de banda limitada con $X(j\omega) = 0$ para $|\omega| > \omega_M$. Entonces $x(t)$ se determina unívocamente mediante sus muestras $x(nT)$, $n = 0, \pm 1, \pm 2, \dots$, si

$$\omega_s > 2\omega_M$$

Ecuación 28.

Donde,

$$\omega_s = \frac{2\pi}{T}$$

Ecuación 29.

Dadas estas muestras podemos reconstruir $x(t)$ generando un tren de impulsos periódicos en el cual los impulsos sucesivos tengan amplitudes que correspondan a valores de muestras sucesivas. Este tren de impulsos es entonces procesado a través de un filtro paso bajas ideal con ganancia T cuya frecuencia de corte sea mayor que ω_M y menor que $\omega_s - \omega_M$. La señal de salida resultante será exactamente igual a $x(t)$.

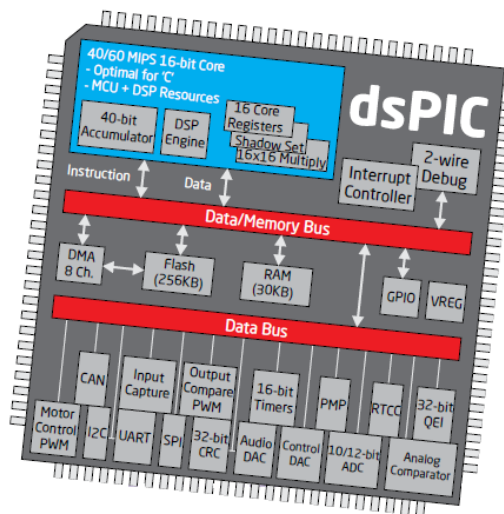
En resumen, la frecuencia de muestreo se define como el número de muestras tomado de una señal continua por unidad de tiempo. Para tener una reconstrucción aproximada de la señal, la frecuencia de muestreo debe exceder dos veces la frecuencia máxima de la señal que se quiere muestrear.

2. DESCRIPCIÓN DEL SISTEMA

2.1. DSPIC 33FJ256GP710A

La familia dsPIC33FJXXXGPX06A / X08A / X10A utiliza una potente arquitectura de 16 bits que combina las características de control de un microcontrolador (MCU) con las capacidades computacionales de un procesador digital de señales (DSP). Este tipo de dispositivos se utilizan en el control, aplicaciones que dependen de alta velocidad y los cálculos iterativos y repetitivos.

Figura 1. Esquema del dsPIC 33FJ256GP710A



Fuente: Manual de Mikromedia para dsPIC33, p. 10.

Las características principales del dsPIC fabricado por Microchip® son:

- Hasta 40 MIPS de operación.
- Arquitectura de 16 bits.
- 100 pines.

- 256 KB de memoria de programa.
- 30.720 Bytes de RAM.
- 85 pins I / O.
- Oscilador interno 7,37 MHz, 512kHz.
- “Timer” de 16 bits.
- 2 ADC, 32 canales.

2.2. SISTEMA DE DESARROLLO

Mikromedia para dsPIC33® es un sistema portátil con una gran cantidad de periféricos, que permiten el desarrollo de dispositivos con contenidos multimedia. La parte central del sistema es un microcontrolador dsPIC33FJ256GP710A de 16 bits. Mikromedia es compacto y delgado, se adapta perfectamente a la palma de la mano (figura 2), por lo que lo hace conveniente para los dispositivos portátiles.

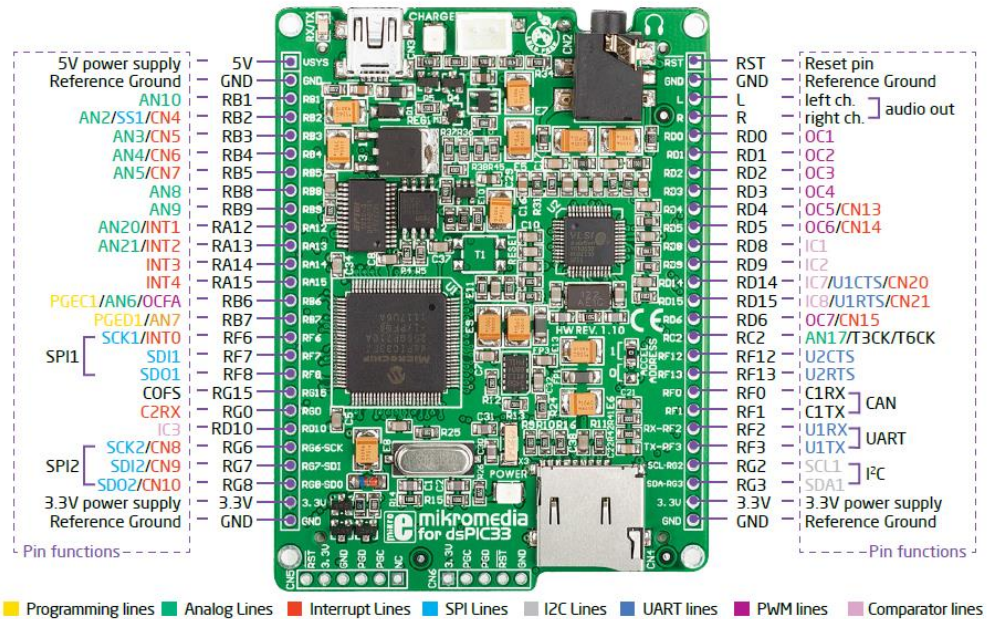
Figura 2. Sistema de desarrollo Mikromedia para dsPIC33®



Fuente: Manual de Mikromedia para dsPIC33, p. 4.

En la figura 3, se observa el diagrama de pines de Mikromedia para dsPIC33®.

Figura 3. Diagrama de pines del sistema de desarrollo Mikromedia para dsPIC33®



Fuente: Manual de Mikromedia para dsPIC33, p. 39.

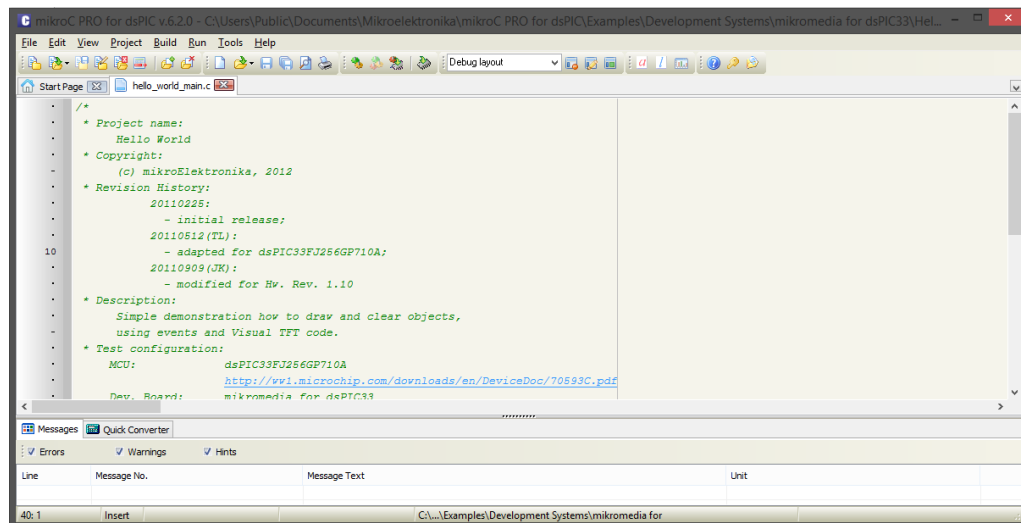
2.3. COMPILACIÓN

La compilación de la rutina de código se realiza mediante mikroC PRO para dsPIC30/33 and PIC24, adquirido de forma legal directamente con MikroElektronika®. MikroC PRO consiste en una poderosa herramienta de desarrollo con un alto contenido de funciones para estos microcontroladores. Entre sus principales características se encuentran:

- Se escribe el código fuente en el editor de código incorporado (que incluye asistente de código y parámetros, plegado de código, resaltado de sintaxis, autocorrección, plantillas de código, y más).

- Incluye bibliotecas para acelerar el desarrollo de: la adquisición de datos, la memoria, pantallas, las conversiones, la comunicación, etc.
- Se puede utilizar el mikroICD integrado (depurador en circuito) herramienta de depuración en tiempo real para supervisar la ejecución del programa en el nivel de hardware.
- Ofrece una cantidad elevada de ejemplos para ampliar, desarrollar y utilizar como guía para la construcción de proyectos.

Figura 4. Interfaz de mikroC PRO for dsPIC30/33 and PIC24



2.4. PROGRAMACIÓN

La programación se lleva a cabo a través de mikroProg™ de MikroElektronika®, rápido programador USB 2.0 con hardware mikroICD™, su inteligente ingeniería permite soportar PIC10®, PIC12®, PIC16®, PIC18®, dsPIC30 / 33®, PIC24® y PIC32® en un solo programador.

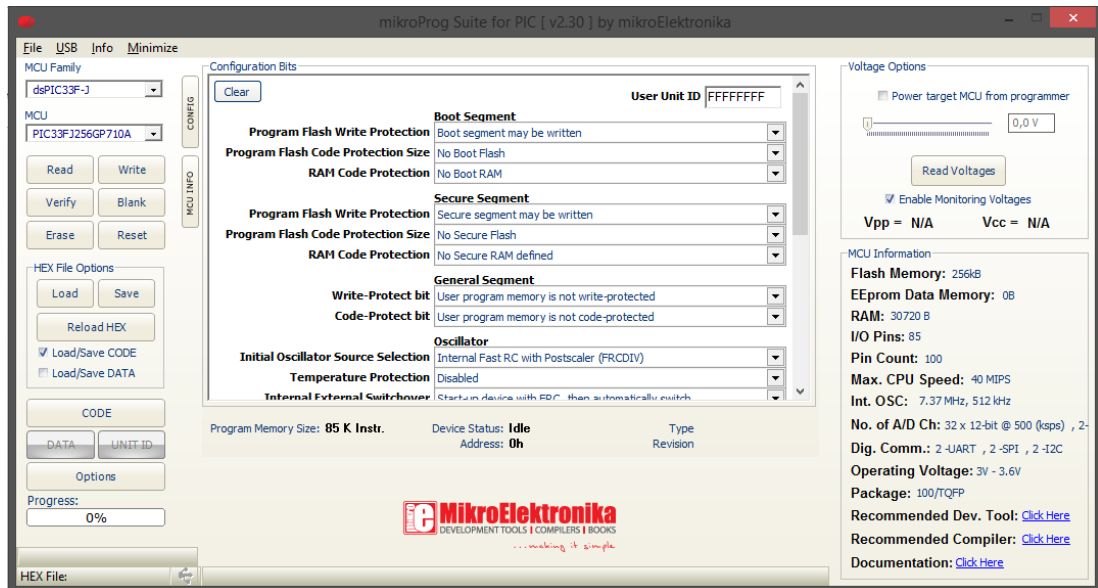
Figura 5. Programador mikroProg™ de MikroElektronika®



Fuente: Manual de Mikromedia para dsPIC33, p. 18.

Para llevar a cabo la programación de archivos, mikroProg™ necesita un software especial llamado mikroProg suite™ para PIC®, El software tiene una interfaz intuitiva (figura 6), que facilita la programación de todas las familias de microcontroladores de Microchip®.

Figura 6. Interfaz de mikroProg suite™ para PIC®



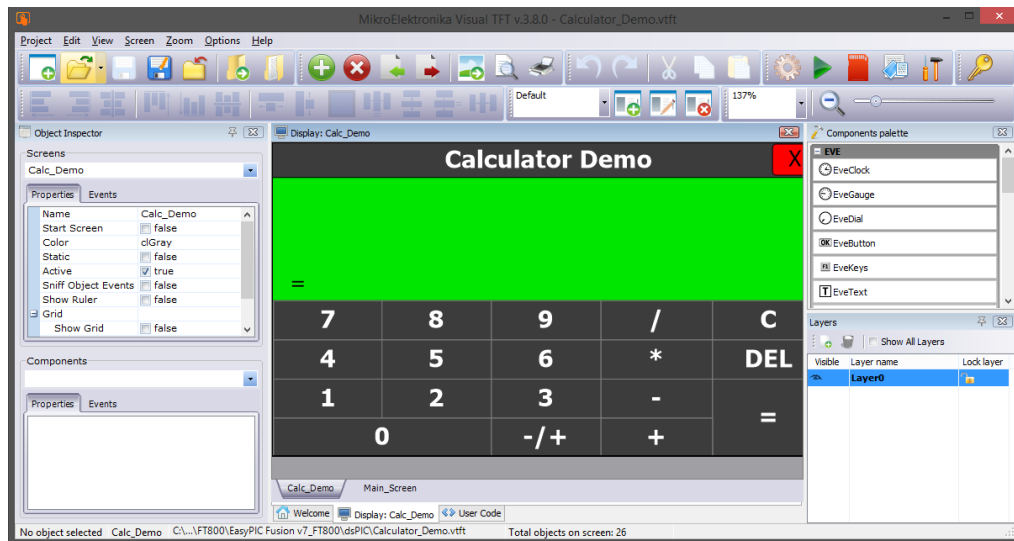
Fuente: Manual de Mikromedia para dsPIC33, p. 19.

2.5. INTERFAZ GRÁFICA

La interfaz gráfica es el medio que permite la comunicación de dos sistemas independientes. Se define como un grupo de formas y métodos que hacen posible la interacción de un sistema con los usuarios utilizando formas gráficas (botones, íconos, ventanas, fuentes, etc. los cuales representan funciones, acciones e información) e imágenes.

Referente al diseño de la interfaz gráfica, se realiza con ayuda de la herramienta Visual TFT en su versión 3.8.0, con licencia legal adquirida directamente con MikroElektronika®, dicha herramienta consiste en un software con el desarrollo gráfico para el diseño de contenidos visuales que se quieren mostrar en los TFT's.

Figura 7. Interfaz de Visual TFT de MikroElektronika®



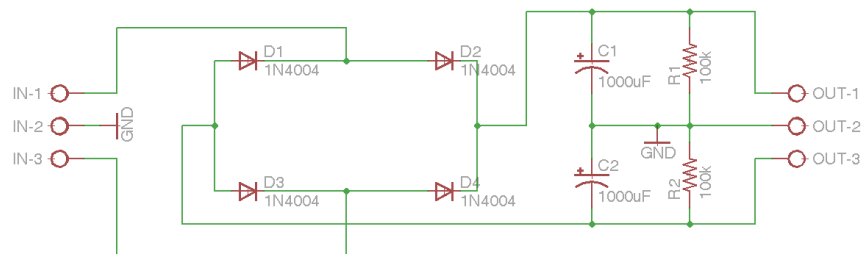
Luego de completar el diseño, esta herramienta genera un código compatible con el compilador. En el caso que se ocupa este proyecto se tiene en cuenta que el sistema de desarrollo cuenta con una pantalla TFT de 320x240 cubierta con un panel táctil resistivo que permite que la interacción con la pantalla.

3. HARDWARE

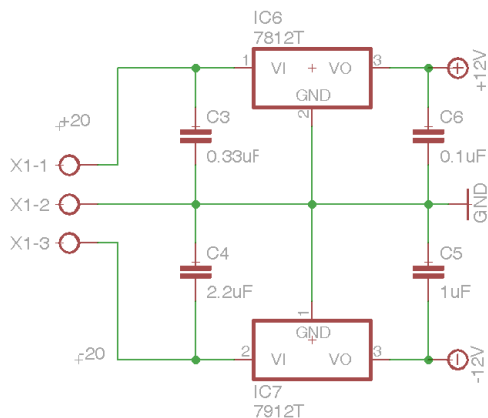
3.1. CIRCUITO DE ALIMENTACIÓN Y REGULACIÓN DE TENSIÓN

En primer lugar se diseña una fuente de alimentación propia para el circuito, con el fin de reducir costos en el desarrollo del proyecto, y proporcionar alimentación dual a los dispositivos que componen la parte externa del proyecto.

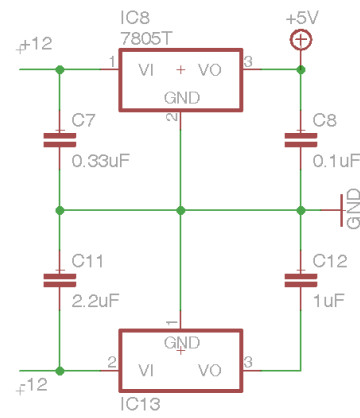
Figura 8. Esquema de fuente de alimentación y reguladores de tensión



a. Fuente de alimentación.



b. Regulación de tensión de 12 [V]



c. Regulación de tensión de 5 [V]

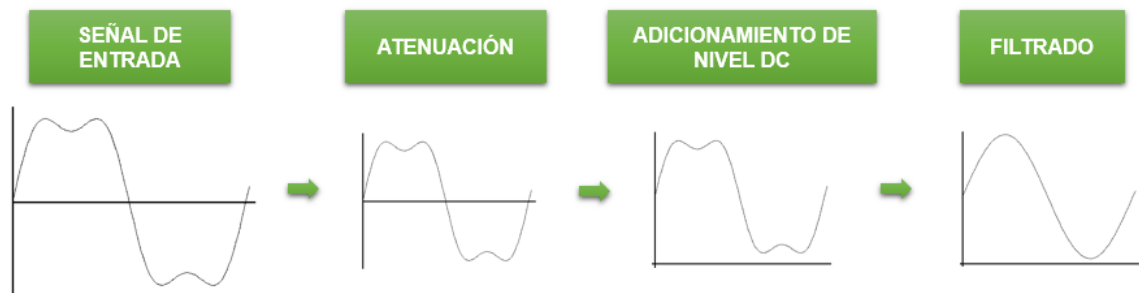
Se planteó y construyó dos circuitos de regulación, cada uno con distinta función de acuerdo a las necesidades presentadas en las configuraciones diseñadas, las configuraciones se muestran en la figura 8.b, y 8.c.

La regulación de ± 12 [V] tiene como finalidad la alimentación de todos los amplificadores operacionales, además de ser la entrada para la regulación de 5 [V], mientras que la regulación de ± 5 [V] tiene como principal función la alimentación de la pantalla LCD TFT de Mikromedia para dsPIC33®.

3.2. ACONDICIONAMIENTO DE LA SEÑAL

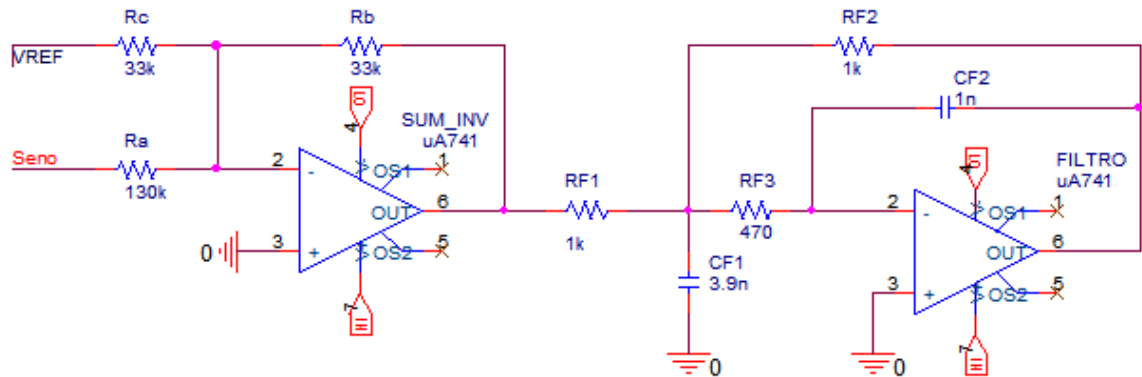
Para obtener una correcta lectura de los puertos del conversor analógico digital (ADC) se debe cumplir que señal debe estar en el rango comprendido entre 0 [V] y 3.3 [V], de modo que se hace necesario hacer un acondicionamiento de la señal, antes de entrar al sistema. En la figura 9, se observa el esquema implementado.

Figura 9. Esquema general del acondicionamiento de la señal



Mediante la herramienta de simulación Capture de Orcad®, se procede a realizar la simulación teniendo en cuenta la topología mostrada en la figura 10. Cabe aclarar la licencia del programa es legal y es propiedad de la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones,

Figura 10. Topología de la etapa de acondicionamiento de señal.



La atenuación de la señal y posterior adición de nivel de DC, se realiza mediante la implementación de un sumador inversor. La tensión de salida de esta etapa se muestra en la ecuación 30.

$$V_o = R_b * \left(\frac{V_{in}}{R_a} + \frac{V_{REF}}{R_c} \right) \text{ [V]} \quad \text{Ecuación 30.}$$

Teniendo en cuenta que R_b es igual a R_c se tiene finalmente:

$$V_o = \frac{R_b}{R_a} * V_{in} + V_{REF} \text{ [V]} \quad \text{Ecuación 31.}$$

El factor de atenuación está dado por:

$$F_a = \frac{R_b}{R_a} = \frac{33k}{130k}$$

De esta forma el factor corresponde a 0.2538, teniendo en cuenta esto, si la tensión de entrada tiene una amplitud de 6 [V], la tensión de salida corresponde a 1.523 [V].

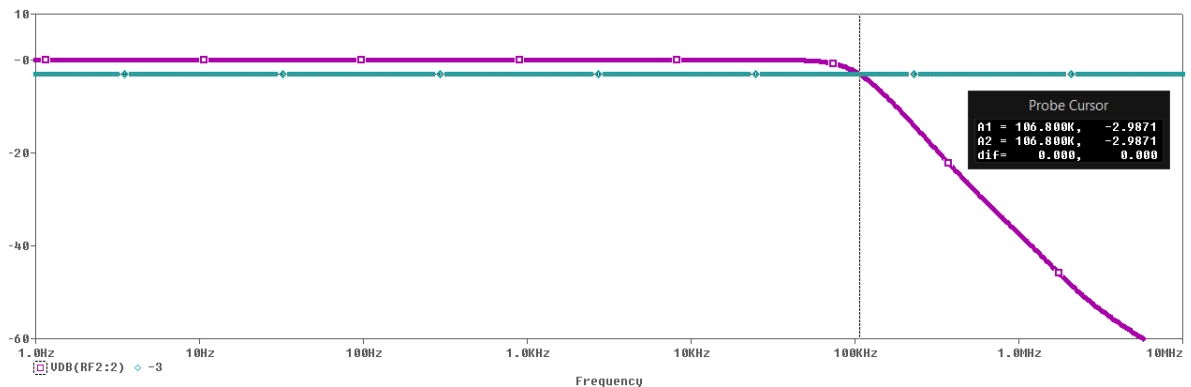
Luego de tener la señal en el rango deseado, se diseña una etapa de filtrado con el fin de eliminar componentes en alta frecuencia, que pueden ser posibles fuentes de error a la hora de hacer el tratamiento digital a la señal.

Texas Instruments, Inc ofrece un software gratuito para el diseño de filtros analógicos llamado FilterPro Desktop. FilterPro solicita ciertos parámetros tales como tipo de filtro, especificaciones del filtro (Ganancia, frecuencia de corte, orden del filtro, etc.), respuesta del filtro y topología del filtro.

De esta manera se diseña un filtro pasabaja de segundo orden de ganancia unitaria con frecuencia de corte $F_c = 100$ [KHz], y filtro Butterworth, debido a su respuesta plana en el ancho de banda deseado.

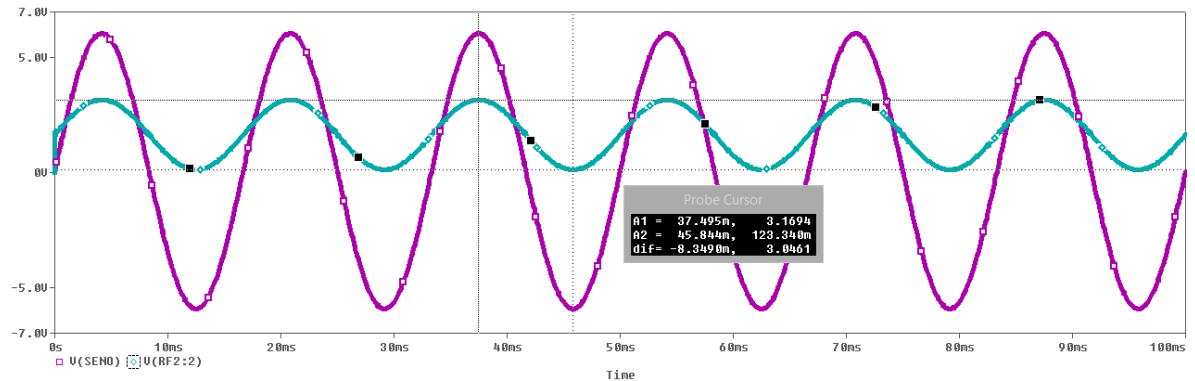
Se realiza un análisis en frecuencia para observar las características del filtro, desde 1 [Hz] hasta 1 [MHz], con paso de 101 puntos por década para observar claramente la respuesta del filtro e identificar la frecuencia de corte. La frecuencia de corte corresponde a 106,800 [KHz] como se muestra en la figura 11.

Figura 11. Respuesta en frecuencia de la salida del filtro pasabaja diseñado



Finalmente en la figura 12 se muestra una señal sinusoidal de entrada, proveniente de un generador de señales, de amplitud equivalente a 6 [V], frecuencia de 60 [Hz], frente a la señal acondicionada,

Figura 12. Señal de entrada frente a su equivalente acondicionada



4. SOFTWARE

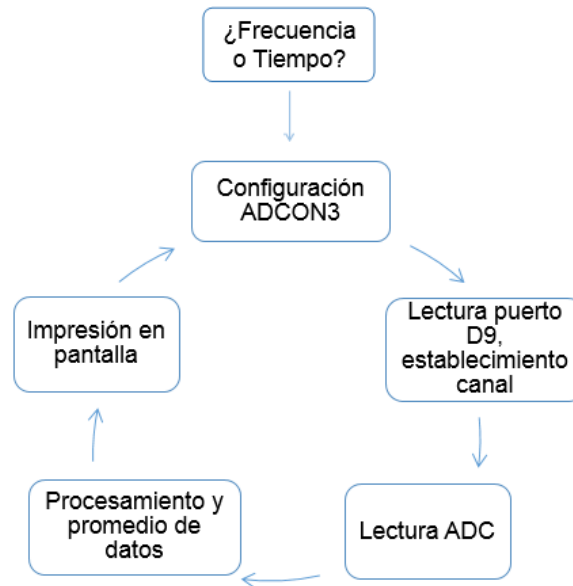
Se cuenta con un sistema de desarrollo, un compilador y una programadora. La dinámica para la creación de un proyecto consiste en construir una rutina de código en el compilador, dicho código está formado por funciones claves creadas por el desarrollador del software, aunque el programador también posee la facultad de crear sus propias funciones.

Creada una rutina, el compilador revisa el código en busca de errores, en caso de encontrarlos envía una mensaje de alerta con la posible localización del error de lo contrario procede a crear un archivo con extensión “.HEX” el cual contiene la traducción del código al lenguaje de máquina que será interpretado por el dsPIC. Se utiliza la programadora mikroProg™ para cargar el archivo en el sistema de desarrollo, la próxima vez que inicie el sistema, dicho archivo será ejecutado mostrando la interfaz y rutina pretendida.

En la figura 13 se muestra el diagrama de flujo general de la rutina descrita para el presente trabajo, en ella se observa, que en primera instancia el usuario debe definir si desea cargar el programa para análisis en tiempo o en frecuencia, luego de ello

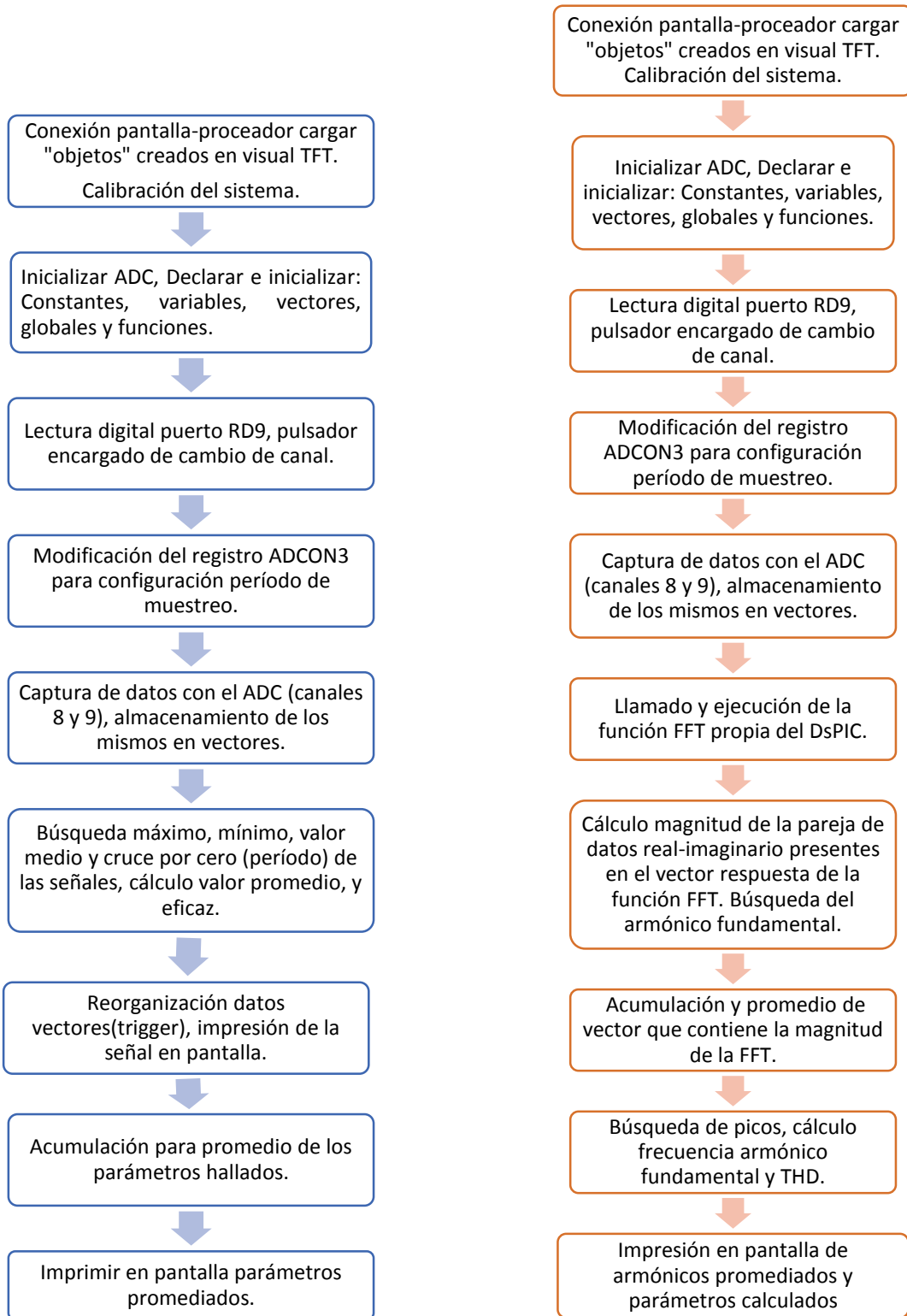
se nota un ciclo que no se ve interrumpido por ninguna acción diferente al reinicio del hardware.

Figura 13. Rutina general del programa implementado.



En la figura 14 se muestra en detalle cada uno de los procesos que conforman dicho ciclo, en azul podremos los procesos referentes al análisis en tiempo y en naranja los que hacen referencia al análisis en frecuencia.

Figura 14. Descripción paso a paso del programa implementado



4.1. LECTURA DE DATOS

El sistema de desarrollo dispone de conversores analógicos digitales (ADC) de 10 y 12 bits, con 16 canales disponibles para la conversión. En primer lugar se define el canal donde se realiza la lectura, debido a que se desea implementar la visualización de dos canales se ha seleccionado los canales 8 y 9 para las señales de tensión y corriente respectivamente.

Seleccionados los canales se inicializa el conversor analógico digital, la inicialización puede hacerse de forma automática o manual. La manera automática se realiza con los comandos “ADCx_Init” o “ADCx_Advanced” (donde la “x” debe ser reemplazada por el número del conversor)*.

La inicialización también puede hacerse de forma manual con la modificación del registro ADxCON1 tal como se muestra a continuación.

Figura 15. Código para configuración manual del ADC

```
          //**** Registros AD1CON1 ****//
AD1CON1bits.ADON = 1; // Enciende ADC
AD1CON1bits.FORM = 3; // Formato datos de salida
AD1CON1bits.SSRC = 7; // Autoconvertir
AD1CON1bits.ASAM = 1; // ADC Sample Control: Sampling begins immediately after conversion
AD1CON1bits.AD12B = 0; // Resolución 10-bit
```

Es necesario determinar una serie de parámetros además de tener ciertos cuidados descritos a continuación para que los datos digitalizados correspondan a la señal que está siendo muestreada. El parámetro más importante a definir es la frecuencia de muestreo, para ello es obligatorio la inclusión del teorema de muestreo detallado en el capítulo 1.

* Para mayor información acerca de estos comandos se recomienda visitar el menú ayuda del compilador.

Dado que en muchos casos la frecuencia de la señal entrante es desconocida se recomienda en principio muestrear a la frecuencia más alta posible. Ésta afecta directamente en la visualización de la señal en pantalla debido a que entre mayor sea, menos periodos de una misma señal podrán visualizarse si se considera que existe un numero de muestras fijas. Para seleccionar una frecuencia de muestreo debe referirse al registro ADxCON3.

Figura 16. Código para configuración de frecuencia de muestreo

```
        //**** Registros AD1CON3 ****//  
AD1CON3bits.ADRC = 1; // Reloj interno del ADC: 7.37 MHz  
AD1CON3bits.ADCS = 31; // ADC Conversion clock  $T_{ad}=T_{cy}*(ADCS+1)$   
AD1CON3bits.SAMC = 31; // Tiempo automatico de muestreo: 31  $T_{ad}$ 
```

Dicho registro se divide en 3 registros de diferentes bits:

- AD1CON3bits.ADRC: Se utiliza para seleccionar uno de los relojes internos del sistema.
- AD1CON3bits.ADCS: corresponde la variable que completa la formula $T_{ad}=T_{cy}*(ADCS+1)$ donde T_{ad} es el periodo fundamental de muestreo.
- AD1CON3bits.SAMC: selecciona la cantidad de periodos fundamentales que conformaran la frecuencia automática de muestreo.

Luego del establecimiento de la frecuencia de muestreo, se procede a decidir el número de muestras que se desea tomar, debe tenerse en cuenta que un número muy grande de muestras, p. ej. 2048, puede hacer muy lento el procesamiento de los datos mientras que un número muy pequeño, p. ej. 16, quizás no contenga la información necesaria para la correcta caracterización de las señales.

Para el desarrollo de este trabajo de grado se ha decidido tomar 128 muestras, correspondiente a una potencia de dos, debido a que la función que calcula la FFT trabaja con esa condición, más adelante se abordará el tema con más detalle.

Con el fin de poder visualizar un número mayor de señales de diferentes periodos, se han configurado varias frecuencias de muestreo que el usuario podrá aumentar o disminuir según lo requiera, éstas se relacionan en la tabla 1.

Tabla 1. Frecuencias de muestreo implementadas

	FS [KHz]				
TIEMPO	1.55	6.95	20.83	101.00	281.70
FFT	1.22	13.88	86.00	-	-

Previo a la toma de datos se ha creado un vector que posee tamaño equivalente al número de muestras, luego se configura un ciclo “*For*” donde se utiliza una función de lectura de canal como se muestra en figura 17.

Figura 17. Ciclo *for* para toma de muestras

```

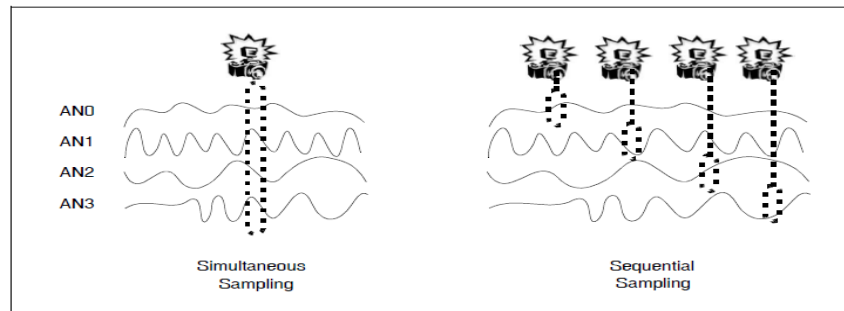
For (i=0; i<Nn; i++) {
    signalch1 [i]=Adc1_get_sample(8);
    signalch2 [i]=Adc1_get_sample(9);
    PORTA=~PORTA;
}

```

Al configurar de esta manera la toma de datos se garantiza que las muestras se tomaron de forma casi simultánea posible, que al tratarse de un lenguaje secuencial ambas lecturas no podrán ocurrir al mismo tiempo sin embargo el tiempo entre cada una de ellas es corto (equivalente a un periodo del ciclo de reloj del sistema 7.37 [MHz] que equivaldrían aproximadamente a 135 [ns]), esto es importante dado que

para hallar los parámetros de potencia ambas señales tanto tensión como corriente deben ser examinada en un mismo periodo o intervalo de tiempo.

Figura 18. Muestras tomadas simultáneamente.



Fuente: *dsPIC33F Family Reference Manual. Section 16. p. 15*

Donde, “signalch1” es el vector donde se desea guardar las muestras del canal 1 y “signalch2” el vector para el canal 2, “Nn” corresponde la cantidad de muestras a tomar e “i”, equivale a la variable de incremento.

También se puede visualizar el comando de lectura “Adcx_Get_Sample”, el cual conserva las modificaciones o configuraciones que se hayan realizado a los registros ADCON. Para la correcta implementación de esta función debe especificarse primero el convertor analógico digital que se desea usar y entre el paréntesis el canal donde se hace la lectura.

4.2. PROCESAMIENTO DE DATOS EN EL TIEMPO

Al finalizar el ciclo “for” los vectores “signalch1” y “signalch2” contendrá “Nn” valores enteros (128 muestras), la magnitud de los elementos de estos vectores estará en el rango de 0 a 1023, debido a la elección de trabajar con una resolución en el ADC de 10 bits, estos elementos deberán ser normalizados en diferentes oportunidades según el propósito del programador.

No se debe olvidar que las señales provienen de una etapa de acondicionamiento, de modo que se considera importante “quitar” el nivel de tensión DC que en dicha etapa fue añadido, esto se lleva acabo al sustraer de cada dato de “signalch1” y “signalch2” el valor digitalizado de la tensión DC, equivalente al cociente entre producto del nivel DC de referencia por el número máximo que se puede representar con 10 bits (1023), y la tensión de referencia del ADC (3,3[V]).

Una vez realizado este proceso, en el siguiente paso se calculan los parámetros característicos de la señal empezando por los valores máximo y mínimo de cada vector, para ello se toma una muestra aleatoria del mismo y se guarda en una variable auxiliar, posteriormente se compara todos los elementos del vector con esta muestra, en caso de encontrar un número mayor (o menor si se busca el mínimo), la variable auxiliar tomará este valor y seguirá la comparación hasta haber examinado las 128 muestras.

Figura 19. Código para búsqueda de máximos y mínimos

```
aux=todo[50];
aux2=todo[50];

for (r=0; r<Nn ;r++) {
  if (todo[r]<aux){
    aux=todo[r];
  }
  If (todo[r]>aux2){
    aux2=todo[r];
  }
}
```

Una vez encontrados, estos valores se asignan a las variables “vmin2” y “vmax2” respectivamente para almacenarlas hasta el momento de la impresión y poder usar las auxiliares en otras operaciones. Con el valor máximo y mínimo de la señal es posible encontrar el valor medio de la siguiente manera:

$$\text{Valor medio} = \frac{\text{maximo} + \text{minimo}}{2}$$

Ecuación 32.

Este valor es importante dado que las señales que se procesen pueden contener nivel de continua, por lo cual el cruce por cero no será una constante para todas, así podremos indagar sobre el paso de la señal por su valor medio y determinar el número de muestras que conforman un periodo de la señal.

Figura 20. Código para hallar valor medio de la señal

```
auxm=aux+(aux2-aux)/2;
r1=0;
for (bb=0; bb<Nn-1; bb++) {
if(todo[bb]>=auxm && todo[bb+1]<=auxm){
    Gpos[r1]=bb;
    r1=r1+1;
}
}
```

Una vez hecha la operación para encontrar el valor medio, dicho resultado se almacena en la variable “auxm”. Idealmente debería preguntarse cuál de las 128 muestras toma el valor medio de la señal, pero esto correspondería a una condición demasiado estricta ya que no puede garantizarse que este punto fue muestreado, causando errores en el procesamiento, por lo que preferiblemente se procede de la siguiente manera:

Se recorre el vector muestra por muestra, se establecen dos condiciones, la primera hace referencia si el valor de la muestra es mayor o igual al valor medio, mientras la segunda pregunta si el valor siguiente es menor o igual al valor medio, si se cumplen simultáneamente se toma la posición donde se encuentra la muestra y se almacena en el vector “Gpos”, en este vector entonces quedaran almacenadas todas las posiciones en la que la señal tuvo un cruce por su valor medio. Con estos

valores es posible calcular la frecuencia de la señal tal como se menciona en el capítulo uno.

Hallada la frecuencia se realiza el cálculo de parámetros como son: el valor promedio, el eficaz y la potencia promedio. Para el caso de la potencia promedio previamente se debió realizar un ciclo *for* en donde se multipliquen elemento a elemento el vector “signalch1” y “signalch2” dando resultado a un nuevo vector que en este proyecto fue denominado “Poten”*, además en las variables “welta1” y “welta2” se almacena la cantidad de muestras en las que está contenido un periodo de cada señal†.

Figura 21. Código para cálculo de parámetros dependientes del periodo

```
for(i=0;i<Nn;i++){
    todo2[i]=(signalch2[i]-507);
    signalp2[i]=(signalch2[i]);
    Poten[i]=(float)todo[i]*todo2[i];
}

welta1=Gpos[2]-Gpos[0];
welta2=Gpos2[2]-Gpos2[0];

for(i=Gpos2[0];i<Gpos2[2];i++){
    Vavg2=(float)Vavg2+todo2[i];
    acu=(float)todo2[i]*todo2[i];
    vrms2=(float)vrms2+acu;
    Potenciapro=Potenciapro+Poten[i];
}

Vavg2=(float)((Vavg2/welta2)*ampl);
vrms2=(float)(sqrt(vrms2/welta2)*ampl);
Potenciapro=(float)(Potenciapro/(welta2*100.))*ampl*ampl;
```

* La multiplicación se realizó con los vectores a los cuales ya se les había sustraído el nivel DC, adicionado a las señales en la etapa de adecuación por hardware.

† El cálculo de estas muestras se realiza cuando se determina la frecuencia de la señal, tal como se explica en el capítulo 1.

El vector que contiene la señal sin el nivel de tensión continua añadido por el hardware se denomina "todo2", por lo que se procede a recorrer el vector y sumar todas las muestras en un periodo, dicha suma se acumula en la variable "Vavg2", la cual es de tipo flotante para evitar desbordamiento, inmediatamente se calcula el cuadrado de la muestra, guardando la misma en la variable "acu" que también es de tipo flotante, del mismo modo se acumula la suma de los cuadrados en la variable "vrms2", para la potencia se realiza el mismo procedimiento que con la tensión promedio esta vez con la variable "potenciapro" y el vector "Poten".

Una vez realizado este procedimiento se divide el acumulado de la tensión y la potencia entre el número de muestras que conforman el periodo de la señal, lo mismo ocurre con la tensión eficaz (vrms2) pero posteriormente se le debe sacar la raíz cuadrada a este valor.

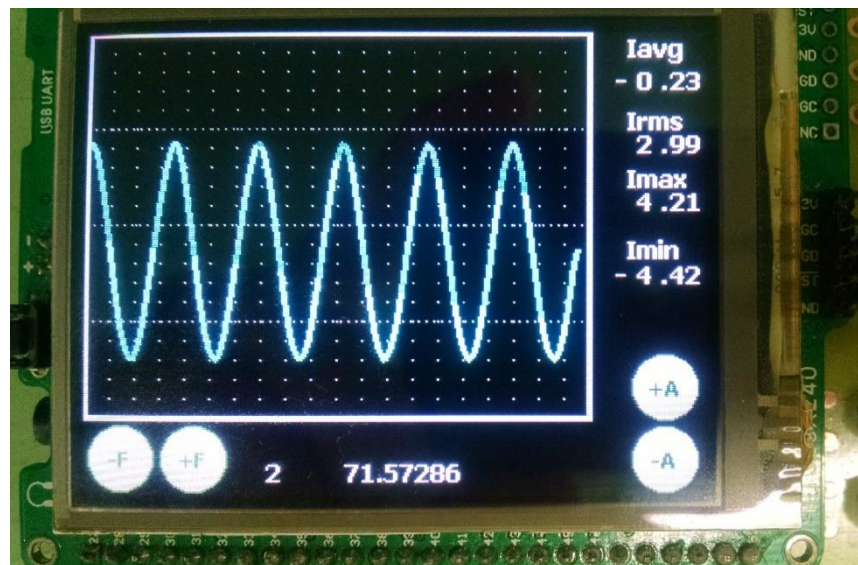
Así en las variables "vmin2", "vmax2", "Vavg2", "vrms2" y "potenciapro" quedan almacenados los valores correspondientes a los parámetros de la señal presente en el segundo canal, el siguiente paso consiste en acumular la sumatoria de estas variables tal como se hizo con los valores promedios, al cumplir un número determinado de ciclos, veinte para este proyecto, se procede a dividir entre el número de ciclos y posteriormente se imprimen los resultados. Este procedimiento es el mismo para las señales presentes en ambos canales sin importar si el usuario está visualizando solo un canal.

Para imprimir la señal en pantalla se realiza un proceso de reorganización del vector, con lo cual se pretende que dicha señal en pantalla permanezca estática, el objetivo de este proceso consiste en buscar un flanco negativo de la señal presente en el vector "signalch1" o "signalch2" y a partir, de la muestra que contenga este flanco empezar a imprimir la señal. Es evidente que las muestras que estén previas al flanco se perderán por lo que es recomendable tomar un número mayor de muestras, en este proyecto el número de muestras tomadas para el programa en

tiempo es de 192, y la búsqueda del flanco se realiza en las primeras 64 muestras, dejando así por lo 128 datos para la visualización.

En este proyecto se decidió tomar los 320 píxeles para el eje horizontal (tiempo) y 240 píxeles para el eje vertical (magnitud), sin embargo es necesario dejar espacio para la visualización de los parámetros, por lo que la impresión de la señal se realiza sobre un rectángulo de 260 x 200 píxeles que a su vez contendrá una grilla espaciada uniformemente.

Figura 22. Distribución de pantalla para visualización de señal y parámetros



Una vez definido el espacio de impresión, el vector que contiene las señales debe ser adecuado para que la muestra de mayor magnitud ocupe los 200 píxeles destinados en el rectángulo, en el diseño realizado para este proyecto el usuario además podrá realizar *zoom* en amplitud sobre la señal por lo que la adecuación del vector se realiza con variables que cambiarán según la interacción del usuario con la pantalla.

Figura 23. Código para impresión en pantalla gráfica

```
TFT_Set_Brush(1,cl_black,0, LEFT_TO_RIGHT, CL_WHITE, CL_WHITE);
    TFT_Set_Pen(CL_silver, 3);
    TFT_Rectangle(0,0,260,200);

    for(i=0;i<127;i++){
        coorY[0]=liminf-(signalp[i]/atenua);
        coorY[1]=liminf-(signalp[i+1]/atenua);
        coorX[0]=i*4;
        coorX[1]=(i+1)*4;
        TFT_Line(coorX[0],coorY[0],coorX[1],coorY[1]);
    }

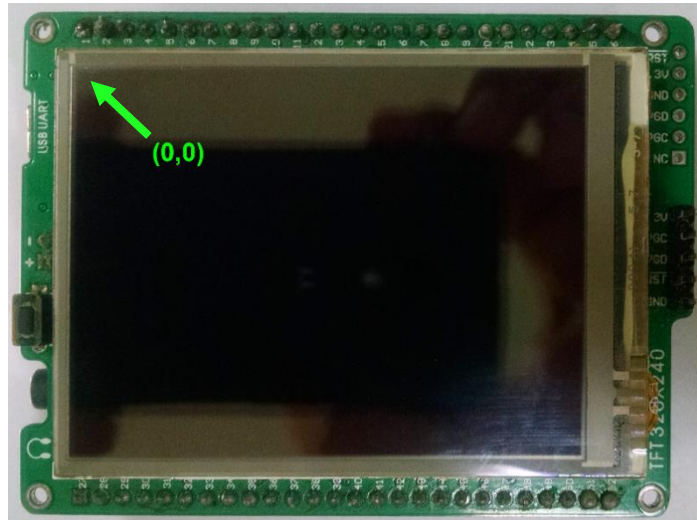
    /*******//
    // Event code, funciones de los botones
    /*******//

    void AmpMasOnClick() {
        atenua=atenua-1;
        if(atenua<1)
            atenua=1;
    }

    void AmpMenosOnClick() {
        atenua=atenua+1;
        if(atenua>10)
            atenua=10;
    }
```

En el anterior código se puede apreciar la configuración del ciclo encargado de imprimir la señal, la función principal, "TFT_LINE()" propia del compilador traza una línea entre dos puntos especificados dentro de los paréntesis, para ello se ha definido dos vectores de dos posiciones denominados "coorX" y "coorY" los cuales contendrán las coordenadas entre los que se desea trazar dicha línea, para la correcta implementación de esta función es importante conocer cuál es el eje de referencia de la pantalla, es decir donde está localizado el punto (0,0).

Figura 24. Localización del punto (0,0) en la pantalla gráfica



Una vez identificado este punto en la esquina superior izquierda de la pantalla, se entiende que a medida que la coordenada crece, el punto se desplaza hacia la parte inferior de la pantalla, por lo cual se crea la variable “liminf” con el fin de invertir la dirección en la coordenada “Y”, esta variable tomara el valor doscientos (200) cuando se desee imprimir una sola señal.

Otras de las variables presentes en el código es “atenua” la cual aumenta o disminuye respecto de la cantidad de veces que el usuario pulse los botones “AmpMas” y “AmpMenos” esto es fácilmente apreciable en el código donde se definen las funciones “AmpMasOnClick” y “AmpMenosOnClick”. Respecto a las coordenadas en “X”, simplemente corresponde a la posición del vector de la muestra que se esté imprimiendo, dado que existen 128 muestras y el espacio para la impresión es 260 pixeles, se decide imprimir una muestra cada 4 pixeles, ocupando así todo el espacio horizontal previsto para la impresión.

Para imprimir los valores de los parámetros almacenados previamente en variables, se construye la función “trunprint”, la cual redondea los números tipo flotantes a sus dos primeras cifras decimales luego de la coma, lo que facilita una mejor

visualización de estos valores. De igual manera se hace uso de la función “tft_write_text()” para imprimir los rótulos de cada parámetro, y la función “TFT_H_LINE” dibuja una línea de color negro, sobre el lugar donde se imprimió el valor del parámetro, esto con el fin de borrar cada valor antiguo y escribir el nuevo valor.

Figura 25. Funciones para imprimir en pantalla

```
TFT_H_LINE(265,320,25);  
  
tft_write_text("Vavg",280,0);  
trunprint(VavgP,275,17);
```

4.3. PROCESAMIENTO DE DATOS EN FRECUENCIA

Luego de tomar los datos, se trasladan al espacio de memoria RAM denominado “ydata” con el fin de que la función “FFT” del compilador pueda acceder más rápido a dichos datos, mejorando así el tiempo de procesamiento de la función.

Figura 26. Código rutina para transformada rápida de Fourier

```
        k=0;  
        while(globali<(NUM_OF_SAMPLES*2)-1){  
Samples[globali++] = todochf1[k++]-508; // Re  
        Samples[globali++] = 0;        // Im  
        }  
        if (globali >= (NUM_OF_SAMPLES*2)-1)  
        { globali = 0;  
        //do not overwrite non processed  
        memcpy(InputSamples, Samples,256);  
        }  
  
// Twiddle factors are taken from the <TwiddleFactors.dpas> unit  
        FFT(7, TwiddleCoeff_128, InputSamples);  
        BitReverseComplex(7, InputSamples);
```

Desde el principio debe ser reservado el espacio en la memoria RAM, en este proyecto se declara un vector de 256 posiciones en la cabecera del programa de la siguiente manera “ydata int inputsamples[256]”, además de reservar dicho espacio, se especifica que los datos que serán almacenados son de tipo entero.

En el vector “todochf1”, de 128 muestras, se encuentra los datos leídos por el ADC, cabe recordar que se debe restar el nivel DC adicionado en la etapa de acondicionamiento de la señal. Al realizar este proceso, las muestras se almacenan en el vector “Samples”, incluyendo ceros entre muestras con el fin de representar la parte imaginaria.

En seguida, con la función “memcpy()” se reemplazan los primeros 256 valores del vector “Samples” en las primeras 256 posiciones del vector “InputSamples”, siendo la entrada de la función “FFT” del compilador, que se encarga de realizar la transformada discreta de Fourier. El resultado se escribe sobre el mismo vector de entrada, conteniendo la parte real e imaginaria.

En primera instancia, se procede a realizar el cálculo de la magnitud, mientras se busca el armónico fundamental, este proceso es muy similar a los descritos anteriormente para el procesamiento en el tiempo por lo que se recomienda al lector repasar el aparte anterior y revisar el código disponible en los anexos.

Una vez identificado el armónico fundamental se guarda la posición en la cual se encontró, y con la misma se calcula la frecuencia del armónico esto se hace multiplicando la frecuencia de muestreo por el número de muestras reales tomadas y dividiendo entre el número de la muestra de dicho armónico.

Para las señales que presenten diferentes armónicos se ha construido una grilla con base en la magnitud y frecuencia máxima que puede mostrarse en pantalla según

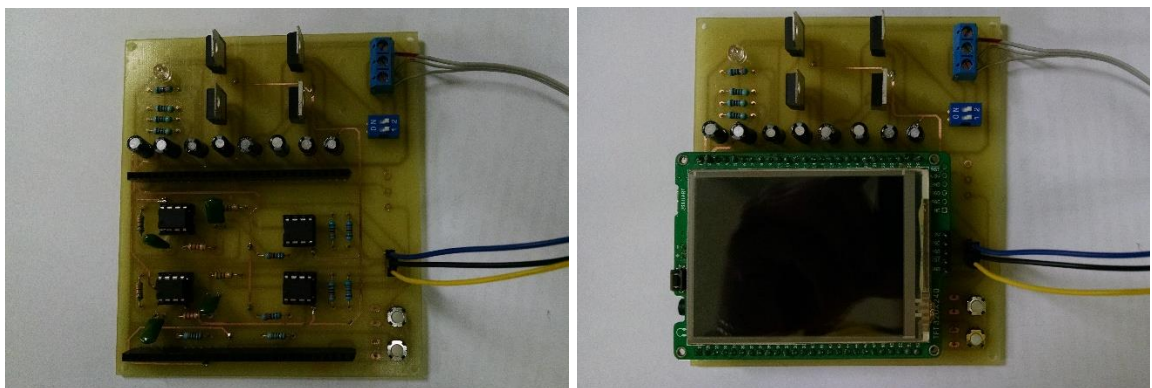
configuración de ADC, los diferenciales de las magnitudes mencionadas son presentados en pantalla para facilidad del usuario.

Para este aparte del programa el usuario cuenta con la opción de aumentar o disminuir la frecuencia de muestreo tal como se explicó anteriormente para el tiempo, las frecuencias de muestreo disponibles se relacionan en la tabla 1. De igual manera también se configuran botones para modificar el *zoom* en amplitud con el fin de que el usuario pueda visualizar mejor los armónicos más pequeños de las ondas, por supuesto a realizar una modificación en la amplitud el delta de la grilla cambia.

5. ANÁLISIS Y RESULTADOS

En el presente proyecto se tiene un dispositivo que utiliza el conversor analógico digital para la medida de la caída de tensión que posteriormente es leída por el ADC y procesada por el dsPIC, que se encarga de realizar los cálculos necesarios para presentar los parámetros característicos de las señales en una pantalla gráfica.

Figura 27. Esquema de la implementación del trabajo de grado



El intervalo de medición se especifica con señales con amplitud entre -6 [V] hasta 6 [V], en frecuencia, al conjunto de formas de onda comprendida entre 60 [Hz] y 1 [KHz] para señales de tipo cuadrada, mientras que en señales sinusoidales y triangulares se amplía el límite hasta 10 [KHz].

A pesar de que la tensión de referencia del ADC es de 0 a 3.3 [V], con las adecuaciones a la señal, el rango de lectura del dispositivo se localiza entre -6 [V] y 6 [V], aunque esto permite muestrear una mayor cantidad de señales, aumenta el error en las medidas, dado que la resolución de la lectura disminuye.

La resolución se encuentra dividiendo el valor pico a pico de la señal máxima entrante entre el número máximo que se puede expresar con la cantidad de bits del ADC (1023). De igual manera es necesario realizar la cuantificación de los valores de tensión leídos, sabiendo que el mayor valor pico a pico que se puede leer corresponde a 1023, en este caso 12 [V] pico a pico corresponde a 1023.

Para hallar cualquier valor asignado a las tensiones leídas basta con aplicar una sencilla regla de tres. Con ello se encontró que al cuantificar un voltio el valor asignado a esta tensión es 78 y al cuantificar 0.01 [V] el valor correspondiente es 0.78 pero dado que el ADC convierte solo a valores enteros, éste le asignara uno o cero, por lo que la sensibilidad de este dispositivo disminuye significativamente a partir de la segunda cifra decimal. Provocando errores sobre todo cuando se quiere hacer cálculos que involucran magnitudes pequeñas o en las que se espera como resultado valores muy pequeños.

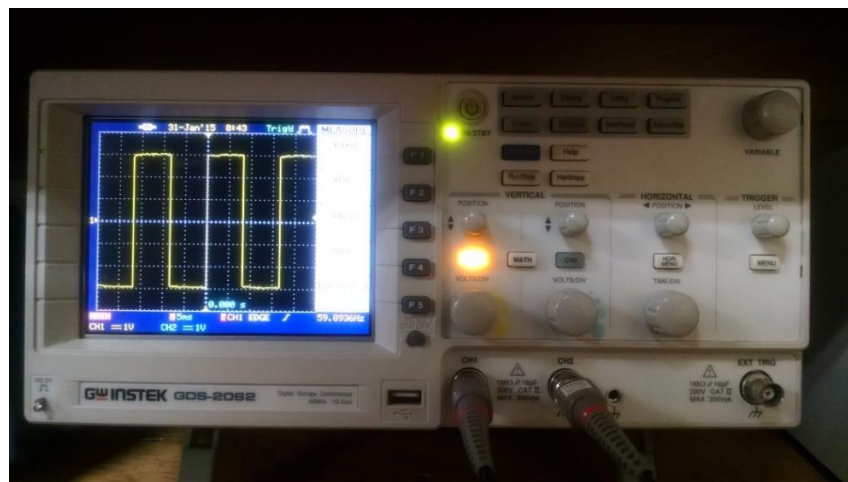
5.1. PRUEBAS DEL SISTEMA

5.1.1. Pruebas para el sistema en el tiempo

Las mediciones del sistema desarrollado se han comparado con las medidas proporcionadas por el osciloscopio GDS – 2062 construido por GW Instek, este equipo se muestra en la figura 28, las medidas del osciloscopio mencionado son tomadas como patrón con el fin de encontrar un error para validar los cálculos proporcionados por este sistema.

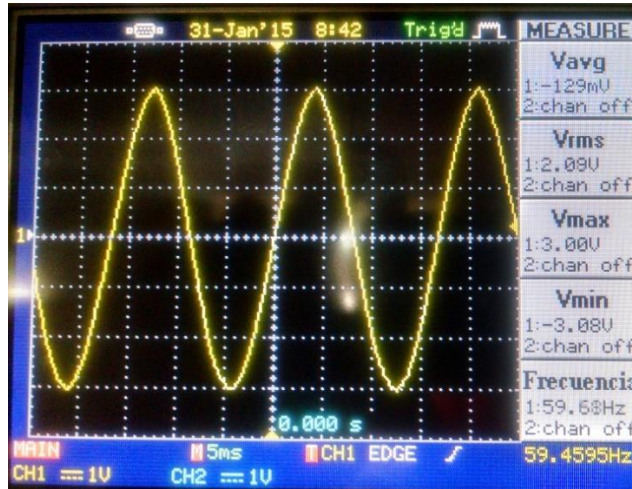
A partir del generador BK Precision® 4040A, se tomaron datos para tensiones iguales o mayores al 50 % del valor máximo que lee el conversor analógico digital (6 [V]) para los dos canales disponibles. Se toman datos para dos tipos de onda: sinusoidal y cuadrada.

Figura 28. Osciloscopio GDS – 2062 de GW Instek



En la onda tipo seno se trabaja con frecuencia equivalente a 60 [Hz], tomándose cuatro datos en cada parámetro (valor eficaz, valor máximo y valor mínimo). En la figura 29 se observa una onda sinusoidal de 60 [Hz] con amplitud equivalente a 3 [V], leída en el osciloscopio junto a las medidas encontradas por este equipo.

Figura 29. Onda tipo seno de 60 [Hz] y amplitud 3 [V] leída por el osciloscopio



En la figura 30 se observa la forma de onda leída por el ADC, junto a los parámetros calculados por el sistema desarrollado para cada uno de los canales disponibles.

Figura 30. Visualización de onda seno de 60 [Hz] y amplitud 3 [V] en el sistema

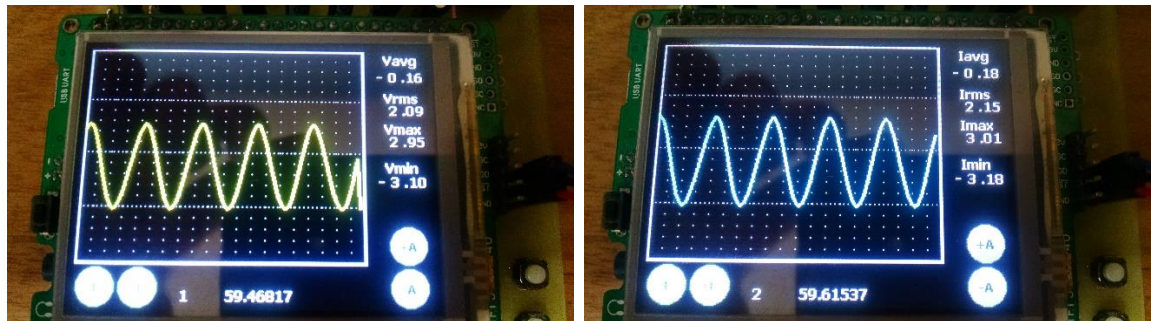


Tabla 2. Medida de frecuencia para una señal sinusoidal de 60 [Hz] con amplitud variable.

Frecuencia				
Osciloscopio	Sistema CH1	Sistema CH2	% Error CH1	% Error CH2
[Hz]	[Hz]	[Hz]		
59,46	59,46	59,61	0,00	0,25

Tabla 3. Medida de valor eficaz, máximo y mínimo de una señal sinusoidal de 60 [Hz] con amplitud variable.

Valor Eficaz				
Osciloscopio [V]	Sistema CH1 [V]	Sistema CH2 [V]	% Error CH1	% Error CH2
2,09	2,09	2,15	0,00	2,87
2,83	2,83	2,87	0,00	1,41
3,53	3,53	3,58	0,00	1,42
4,16	4,15	4,21	0,24	1,20
		Promedio	0,06	1,73

Valor Máximo				
3,00	2,95	3,01	1,67	0,33
4,08	4,01	4,06	1,72	0,49
5,04	5,00	5,07	0,79	0,60
6,00	5,92	6,00	1,33	0,00
		Promedio	1,38	0,35

Valor Mínimo				
-3,08	-3,10	-3,18	0,65	3,25
-4,16	-4,17	-4,25	0,24	2,16
-5,20	-5,20	-5,28	0,00	1,54
-6,08	-6,12	-6,23	0,66	2,47
		Promedio	0,39	2,35

Como se evidencia en la tabla 2, la medida de frecuencia tiene un porcentaje de error menor al 1 % respecto a la medida dada por el osciloscopio. En la tabla 3, para los parámetros calculados, el error es menor al 3% en los dos canales.

La elección del rango especificado de tensiones se debe a que el osciloscopio presenta valores grandes en el valor promedio, cercano a 100 [mV] cuando se espera que este valor para formas de onda distintas sea un valor cercano a cero.

En el momento que son comparados con el sistema desarrollado hace que el error calculado tome valores mayores al 30 % ya que pequeñas variaciones en el nivel de tensión producen errores grandes, de modo que se decide no comparar la lectura del sistema con la del osciloscopio, se opta por hacerlo con la medida dada por un multímetro digital.

Tabla 4. Medida de valor promedio de una señal sinusoidal de 60 [Hz] con amplitud variable.

Valor promedio							
Osciloscopio [V]	Multímetro [V]	Sistema CH1 [V]	Sistema CH2 [V]	% Error CH1	% Error CH2	% Error CH1	% Error CH2
-0,13	-0,15	-0,16	-0,18	23,08	38,46	6,67	20,00
-0,15	-0,19	-0,20	-0,21	33,33	40,00	5,26	10,53
-0,17	-0,23	-0,25	-0,27	47,06	58,82	8,70	17,39
-0,21	-0,27	-0,28	-0,29	33,33	38,10	3,70	7,41
Promedio				34,20	43,85	6,08	13,83

En la tabla 4 se muestra que el error calculado para el sistema comparado con la tensión dada por el osciloscopio, corresponde a valores grandes, mayores al 30 %. Mientras que el error dado por la comparación con el multímetro es menor al 15 %. Del mismo modo, se toman los datos correspondientes a una onda cuadrada de amplitud variable, con frecuencia de equivalente a 60 [Hz]. En la figura 31 y 32, se observa la medición obtenida en el osciloscopio y los canales del sistema.

Figura 31. Onda cuadrada de 60 [Hz] y amplitud 3 [V] leída por el osciloscopio

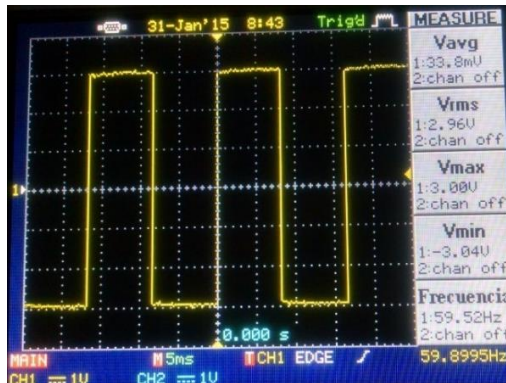


Figura 32. Visualización de onda cuadrada de 60 [Hz] y 3 [V] de amplitud en sistema

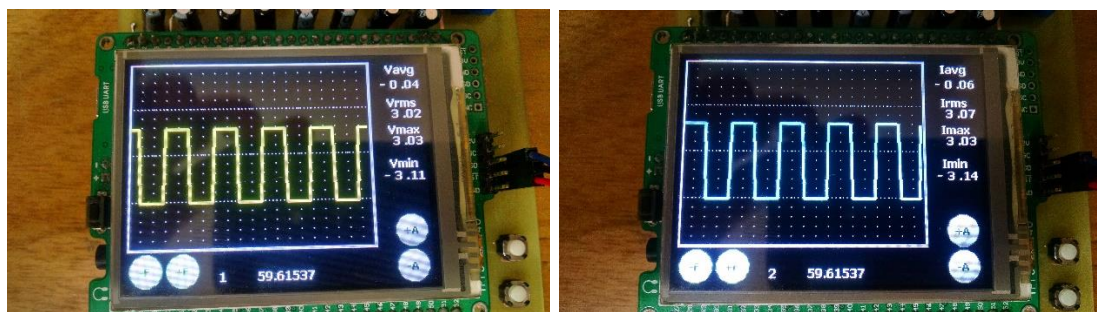


Tabla 5. Medida de frecuencia para una señal cuadrada de 60 [Hz] con amplitud variable.

Frecuencia				
Osciloscopio [Hz]	Sistema CH1 [Hz]	Sistema CH2 [Hz]	% Error CH1	% Error CH2
59,53	59,61	59,61	0,13	0,13

Tabla 6. Medida de valor eficaz, máximo y mínimo de una señal cuadrada de 60 [Hz] con amplitud variable.

Valor Eficaz				
Osciloscopio [V]	Sistema CH1 [V]	Sistema CH2 [V]	% Error CH1	% Error CH2
2,96	3,02	3,07	2,03	3,72
3,96	3,95	4,02	0,25	1,52
4,96	4,96	5,05	0,00	1,81
5,90	5,94	6,03	0,68	2,20
Promedio			0,74	2,31

Valor Máximo				
3,00	3,03	3,03	1,00	1,00
4,08	4,00	4,01	1,96	1,72
5,04	5,07	5,02	0,60	0,40
6,00	6,01	6,09	0,17	1,50
Promedio			0,93	1,15

Valor Mínimo				
-3,04	-3,11	-3,14	2,30	3,29
-4,08	-4,08	-4,10	0,00	0,49
-5,12	-5,17	-5,12	0,98	0,00
-6,08	-6,03	-6,19	0,82	1,81
Promedio			1,03	1,40

Con base en las tablas 5 y 6, correspondientes a los parámetros del valor eficaz máximo y mínimo, el error presente en estos parámetros es menor al 3 % para los dos canales del sistema desarrollado.

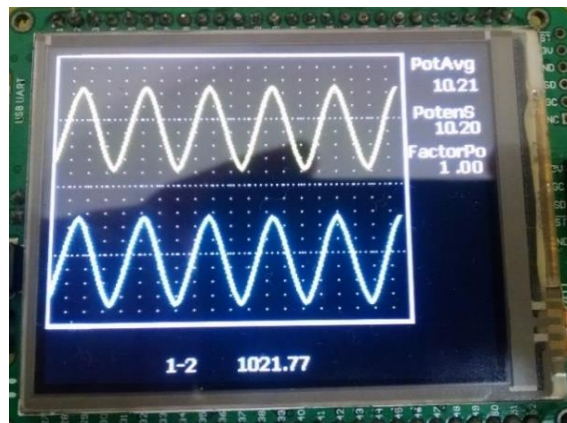
En el caso de la onda cuadrada se evidencia que en el valor promedio la variación de decenas de milivolts resulta crítico para valores muy pequeños. De esta forma toma fuerza la idea de que la comparación con la medida dada por el osciloscopio para este valor en particular no aplica.

Tabla 7. Medida de valor promedio de una señal sinusoidal de 60 [Hz] con amplitud variable.

Valor promedio							
Osciloscopio [V]	Multímetro [V]	Sistema CH1 [V]	Sistema CH2 [V]	% Error CH1	% Error CH2	% Error CH1	% Error CH2
0,03	-0,04	-0,04	-0,04	233,33	233,33	0,00	0,00
0,03	-0,05	-0,07	-0,06	333,33	300,00	40,00	20,00
0,04	-0,05	-0,05	-0,06	225,00	250,00	0,00	20,00
0,06	-0,06	-0,06	-0,06	200,00	200,00	0,00	0,00
Promedio				247,92	245,83	10,00	10,00

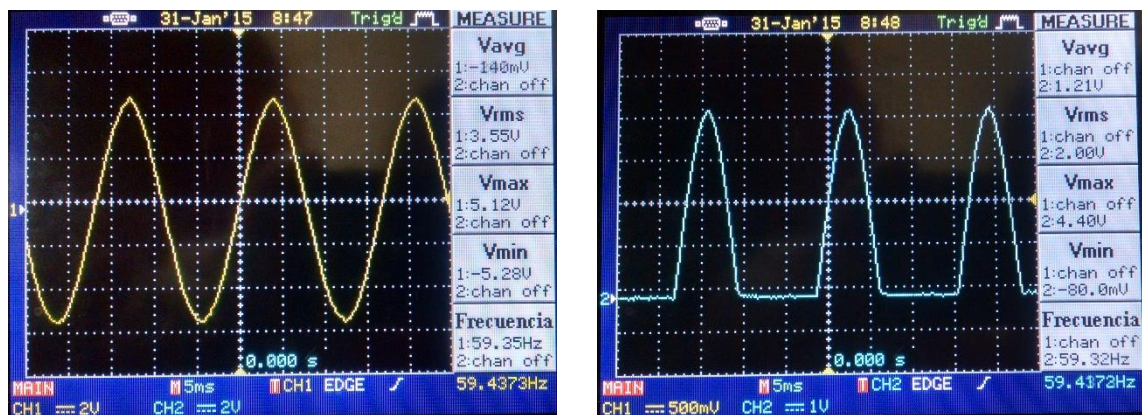
Cuando se observan simultáneamente los dos canales, el sistema calcula la potencia promedio, potencia eficaz y el factor de potencia, teniendo en cuenta que el canal uno corresponde a la tensión y el canal dos a la señal de corriente.

Figura 33. Visualización de los dos canales del sistema para cálculo de potencia.



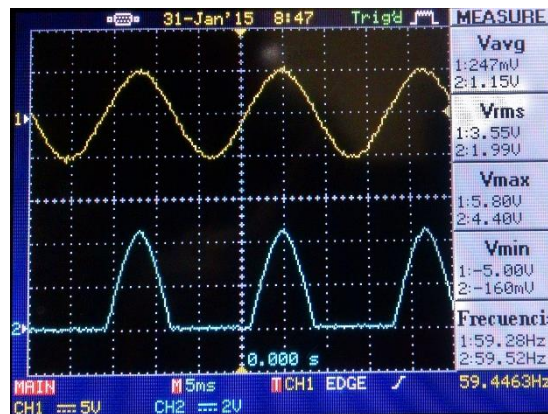
Cuando las señales que lee los dos canales provienen directamente del generador con la misma forma de onda, el factor de potencia equivale a uno ya que no se encuentran desfasadas. Con el fin de comprobar la veracidad del factor de potencia, en el canal de tensión se coloca una señal sinusoidal, mientras que en el canal de corriente se toma la salida de una configuración de rectificador de media onda, de modo que se pueda comprobar con un valor, distinto a la unidad.

Figura 34. Visualización de las señales de tensión y corriente en el osciloscopio



a. Señal de tensión

b. Señal de corriente

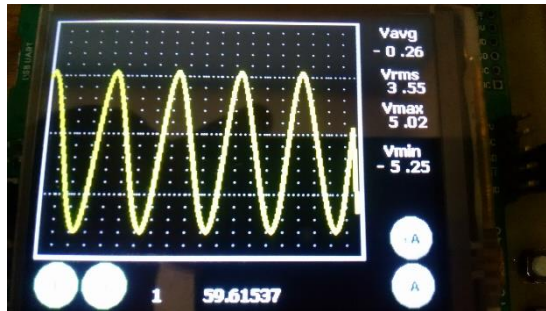


c. Señales de tensión y corriente observadas en el osciloscopio

En la figura 34 se observa las lecturas del osciloscopio de tensión y corriente, base para los datos teóricos que se incluyen en la tabla 6. Los valores teóricos de

potencia promedio, eficaz y factor de potencia se encuentran con Matlab®, y se comparan con los datos mostrados en la figura 35.

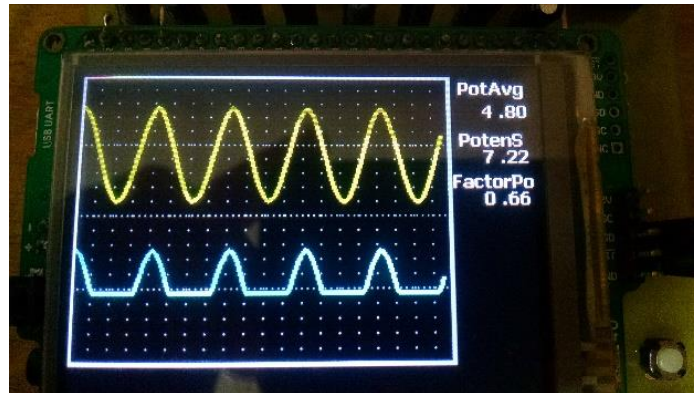
Figura 35. Parámetros de potencia observados en el sistema



a. Señal de tensión en el sistema



b. Señal de corriente en el sistema



c. Parámetros de potencia de las señales de entrada al sistema

Tabla 8. Potencia de tensión y corriente a 60 [Hz].

Frecuencia [Hz]	Potencia Promedio		
	Teórico [W]	Sistema [W]	% Error
60	5,46	4,80	12,09
	Potencia Eficaz		
	7,73	7,22	6,60
	Factor de potencia		
	0,71	0,66	6,56

En la tabla 8, se muestra que los porcentajes de error son menores al 10 % en la potencia eficaz y factor de potencia, respecto a la potencia promedio se ligeramente mayor al 10 %. Los errores presentados se justifican en la forma que se toman los datos y los errores que vienen de cálculos anteriores.

5.1.2. Pruebas para el sistema en frecuencia

En frecuencia se comparan respecto al patrón, tres parámetros importantes: la frecuencia del armónico fundamental, el valor de la amplitud de los armónicos propios de la señal, y finalmente la distorsión armónica total. En la tabla 9 se observa las medidas tomadas, tanto en el osciloscopio como en el sistema, a tres señales sinusoidales de frecuencias 60, 395 y 1000 [Hz], con amplitud variable, mayor al 50 % de la tensión máxima permitida, en pasos de un voltio, se incluye el cálculo del porcentaje de error y la THD.

Tabla 9. Medidas tomadas a tres señales sinusoidales en diferentes frecuencias

Frecuencia			Amplitud			THD
Osciloscopio [Hz]	Sistema [Hz]	% de Error	Osciloscopio [V]	Sistema [V]	% de Error	
60,20	54,00	10,30	3,00	2,90	3,33	0,00
			4,00	3,91	2,25	
			4,96	4,94	0,40	
395,00	432,00	9,37	2,96	2,84	4,05	0,00
			4,00	3,90	2,50	
			4,96	4,80	3,23	
1001	972	2,90	3,00	2,90	3,33	0,00
			4,08	3,95	3,19	
			4,96	4,89	1,41	

Figura 36. Espectro de una señal cuadrada de 60 [Hz] visualizada en el sistema.



En la figura 36 y 37, se observa el espectro de frecuencia calculado para una señal cuadrada de frecuencia equivalente a 60 [Hz] y 395 [Hz] respectivamente, las dos señales con amplitud igual a 5,34 [V]. Los valores teóricos que se expresan en la tabla 10 y 11, correspondientes a la amplitud de los armónicos de una onda cuadrada y la THD, se calculan mediante las ecuaciones 26 y 27, descritas en el capítulo uno.

Tabla 10. Valores obtenidos del espectro de una señal cuadrada de 60 [Hz]

H	Frecuencia			Amplitud			THD		
	Oscilosc. [Hz]	Sistema [Hz]	% Error	Teórico [V]	Sistema [V]	% Error	% Teórico	% Sistema	% Error
1	60,00	54,00	10,00	6,80	6,36	6,46	42,88	44,00	6,46
3	180,00	173,60	3,56	2,27	2,11	6,90			
5	300,00	299,60	0,13	1,36	1,47	8,10			
7	420,00	420,00	0,00	0,97	1,06	8,62			
9	540,00	534,80	0,96	0,76	0,80	6,13			

Figura 37. Espectro de frecuencia una señal cuadrada de 395 [Hz] visualizada en el sistema.



Tabla 11. Valores obtenidos del espectro de una señal cuadrada de 395 [Hz]

H	Frecuencia			Amplitud			THD		
	Oscilosc. [Hz]	Sistema [Hz]	% Error	Teórico [V]	Sistema [V]	% Error	% Teórico	% Sistema	% Error
1	395,00	430,00	8,86	6,80	6,31	7,19	41,41	46,00	7,19
3	1185,00	1202,50	1,48	2,27	2,32	2,41			
5	1975,00	1950,00	1,27	1,36	1,48	8,62			
7	2765,00	2665,00	3,62	0,97	0,87	10,93			

Los porcentajes de error correspondiente a las mediciones obtenidas por el sistema para el análisis de frecuencia son menores al 10 %, respecto a los valores observados en el osciloscopio y teóricos, encontrados mediante ecuaciones descritas en el marco teórico.

6. CONCLUSIONES

Durante el presente trabajo se adquirieron diferentes sistemas de desarrollo de distintas empresas (Arduino, Texas Instruments, Microchip®) con el fin de familiarizarse con ellos y su lenguaje de programación, además de evaluar las variables influyentes en la rapidez de cálculos la transformada rápida de Fourier, se eligieron Mikromedia para dsPIC33® debido a su capacidad para realizar cálculos de manera rápida como la FFT, junto con la facilidad que brinda para la visualización de los resultados en su pantalla gráfica convirtiéndose de esta manera, en una poderosa herramienta para aplicaciones de medida y visualización con una baja inversión respecto a los equipos profesionales.

A lo largo del proyecto se estudiaron y aplicaron conceptos de tratamiento de señales, métodos numéricos, programación en C++ y electrónica analógica por lo cual, se concluye que este tipo de proyectos donde se involucra el uso de un sistema digital constituyen un desafío para los estudiantes de ingeniería electrónica para validar y fortalecer los conceptos estudiados durante el pregrado. De igual manera, la investigación y estudio de otros aspectos del proyecto como la arquitectura del sistema y la creación de su interfaz gráfica forman al estudiante y lo capacitan para enfrentar problemas similares en diversas circunstancias de su futura vida laboral.

La configuración de diferentes tensiones de referencia en el ADC constituiría una mejora importante, dado que al mantener una sola tensión fija y leer señales considerablemente pequeñas respecto a esta tensión se desperdicia un amplio rango de cuantificación, al configurar el cambio de esta tensión de referencia a una tensión menor se amplía dicho rango y se mejora la resolución, permitiendo la disminución de errores en el análisis de señales de baja tensión.

7. RECOMENDACIONES

A pesar del esfuerzo que se realizó para visualizar correctamente el espectro en frecuencia de las señales provenientes del generador, aun son visibles en pantalla algunos armónicos menores no correspondientes a la señal, por lo que se recomienda a quien desee continuar este proyecto la implementación de métodos como el ventaneo que permitan mejorar los resultados de la transformada.

Aunque los botones en pantalla son de gran utilidad para la configuración de los parámetros pretendidos, disponer de un gran número de botones aumentaría la cantidad de llamados que el sistema debe realizar, limitando así las funciones que el desarrollador puede crear, por lo que se recomienda no abusar de la cantidad *screens* y botones a la hora de realizar el diseño con el visual TFT.

En la actualidad en la escuela no se tiene un patrón confiable para la confrontación de las mediciones que se realizan por parte del sistema, de modo que se recomienda la adquisición de un patrón que sirva de referencia para evaluar la exactitud de los sistemas de mediciones construidos por parte de los miembros de la escuela.

8. BIBLIOGRAFIA

[1] HAYT William, KEMMERLY Jack, DURBIN Steven. Análisis de circuitos en ingeniería. Mc Graw Hill, séptima edición 2007, p. 438.

[2] OPPENHEIM Alan, WILLSKY, Alan. Señales y sistemas. Pearson Educación, 1998, p. 361.

[3] POSADAS Juan Luis. Transformada Rápida de Fourier (FFT) e Interpolación en Tiempo Real. Valencia, 1998. Universidad Politécnica de Valencia. Departamento de informática de sistemas y computadores.

[4] TÍJARO ROJAS Omar Javier y RUIDÍAZ PALOMINO Yair de Jesús. . Analizador de espectros portátil utilizando la familia DSP5680X de Motorola. Bucaramanga, 2005. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad Físico Mecánica. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

[5] OPPENHEIM Alan, WILLSKY, Alan. Señales y sistemas. Pearson Educación, 1998, p. 518.

ACEVEDO CASTRO Javier Eduardo y ROJAS RAMOS Rodrigo. Obtención y visualización en tiempo real de los parámetros de rendimiento de un rectificador trifásico basado en tecnología CMOS usando un DSP. Bucaramanga, 2009. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad Físico Mecánica. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

CLAVIJO MENDOZA Juan Ricardo. Diseño y simulación de sistemas microcontrolados en lenguaje C: Programación con el compilador MikroC PRO y simulación en Proteus ISIS. 2011.

IBRAHIM, Dogan. Using LEDs, LCDs and GLCDs in microcontroller projects. John Wiley & Sons, Ltd . 2012

Microchip Technology Inc. dsPIC33FJXXXGPX06A/X08A/X10A Data Sheet. [En línea] [01 de febrero de 2015] disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/70593B.pdf>

MikroElektronika. Mikromedia for dsPIC33 User Manual. [En línea] [01 de febrero de 2015] disponible en: http://www.mikroe.com/downloads/get/1414/mikromedia_dspic33_manual_v110c.pdf

MikroElektronika. MikroC PRO for dsPIC33 User Manual. [En línea] [01 de febrero de 2015] disponible en: http://www.mikroe.com/downloads/get/386/mikroc_pro_for_dspic_manual_v100.pdf

MikroElektronika. MikroProg User Manual. [En línea] [01 de febrero de 2015] disponible en: http://www.mikroe.com/downloads/get/1683/mikroprog_pic_manual_v221b.pdf

MikroElektronika. MikroProg Suite User Manual. [En línea] [01 de febrero de 2015] disponible en: http://www.mikroe.com/downloads/get/1265/mikroprog_suite_pic_manual_v100a.pdf

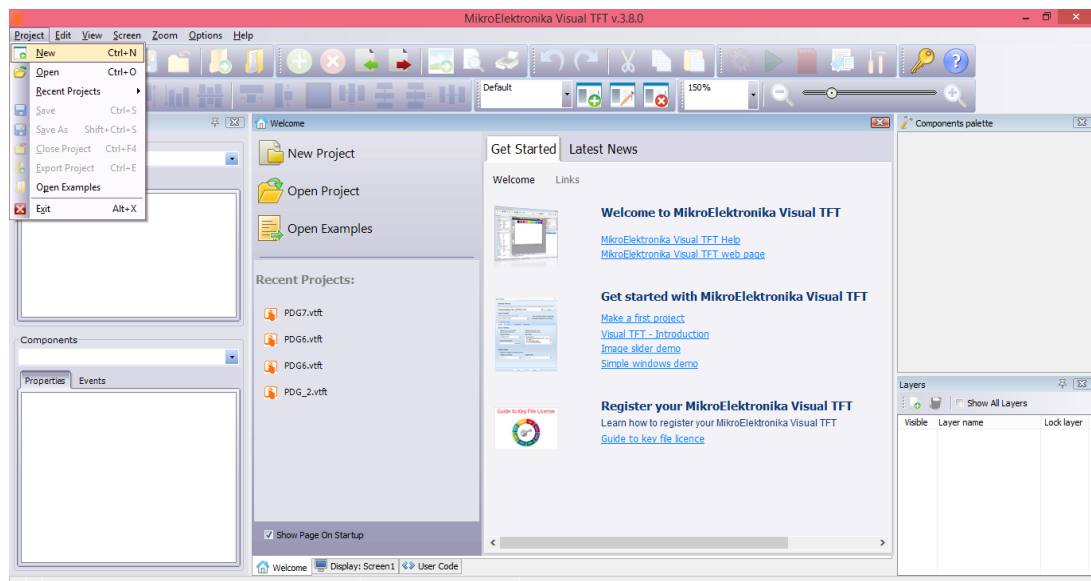
ANEXOS

ANEXO 1. CREACIÓN DE UN PROYECTO.

El primer paso para la implementación de un proyecto que se pretende cargar al sistema de desarrollo “Mikromedia for Dspic33” y en el cual se quiera integrar su pantalla táctil es la creación de la interfaz gráfica con la que el usuario podrá interactuar, con ayuda de Visual TFT de MikroElektronika® en su versión 3.8.0., el programador podrá crear diferentes *screens* y objetos que más tarde podrá manipular con cualquiera de los compiladores de Mikromedia.

Para empezar una vez abierto el programa, se debe hacer clic a la pestaña “Project”, y nuevamente hacerlo sobre la opción *New*.

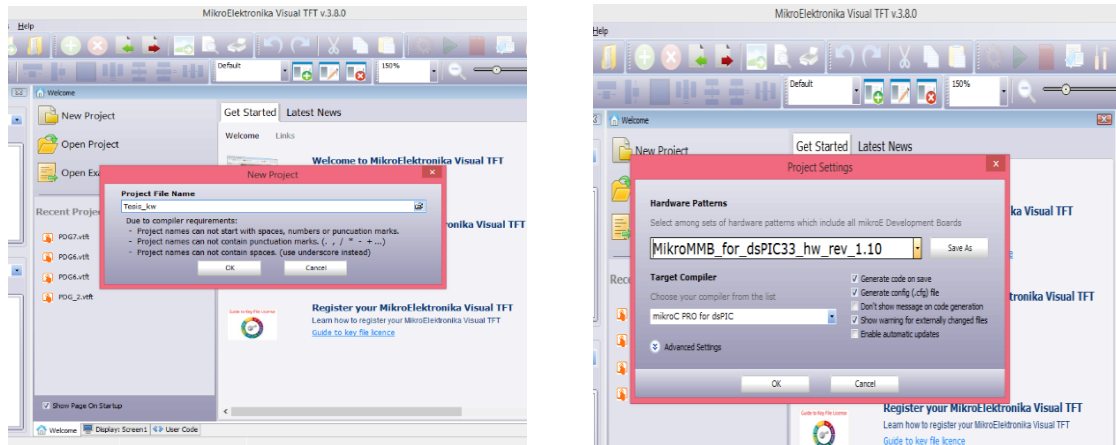
Figura 1. Página principal Visual TFT.



A continuación se debe indicar el nombre del proyecto y la ubicación donde se desea guardar. Definido esto, aparece una nueva ventana donde se realiza la elección del sistema de desarrollo a trabajar, para este caso específico, el sistema elegido corresponde a “MikroMMB_for_dsPIC33_hw_rev_1.10”, es necesario

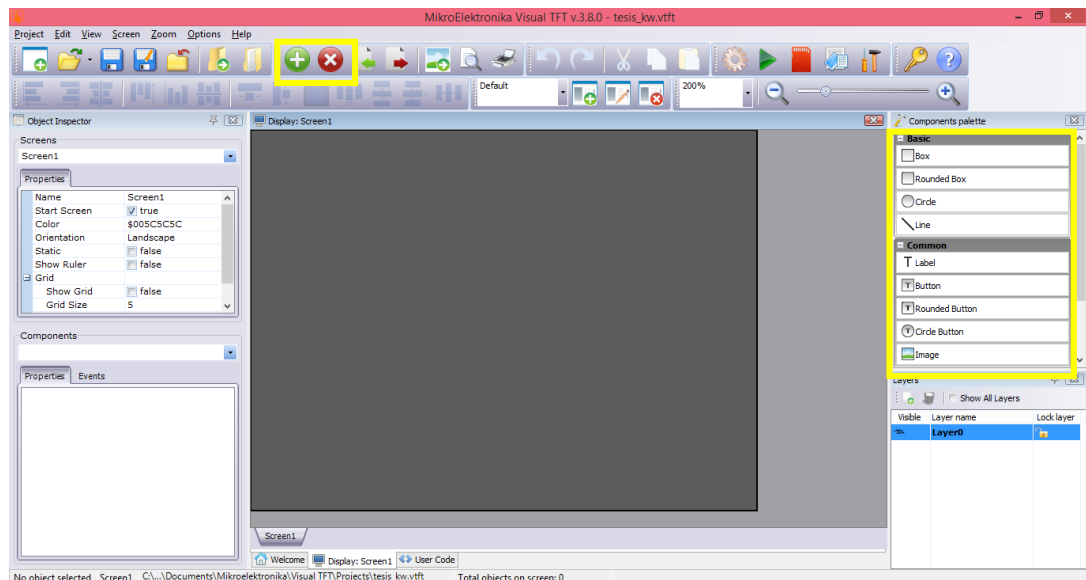
señalar que el compilador con el que se seguirá trabajando en el proyecto es “mikroC PRO for dsPIC”*.

Figura 2. Definir nombre del proyecto y ubicación.



En la pantalla principal se puede agregar *screens*, botones, texto, algunas figuras geométricas disponibles e incluso cargar imágenes.

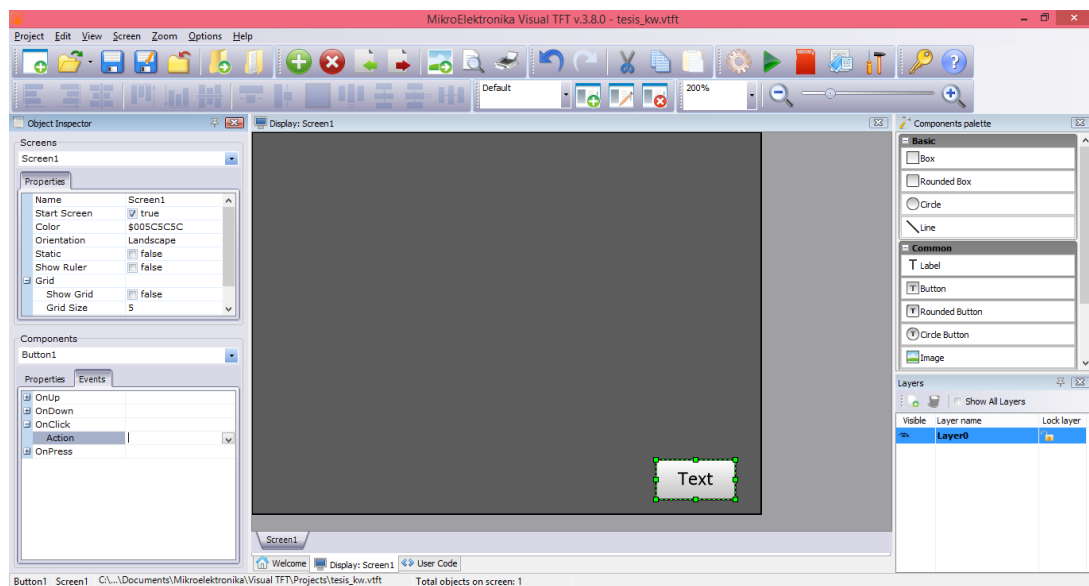
Figura 3. Herramientas principales del software



* En caso de haber adquirido otro software para la compilación del proyecto revisar las demás opciones disponibles en esta pestaña.

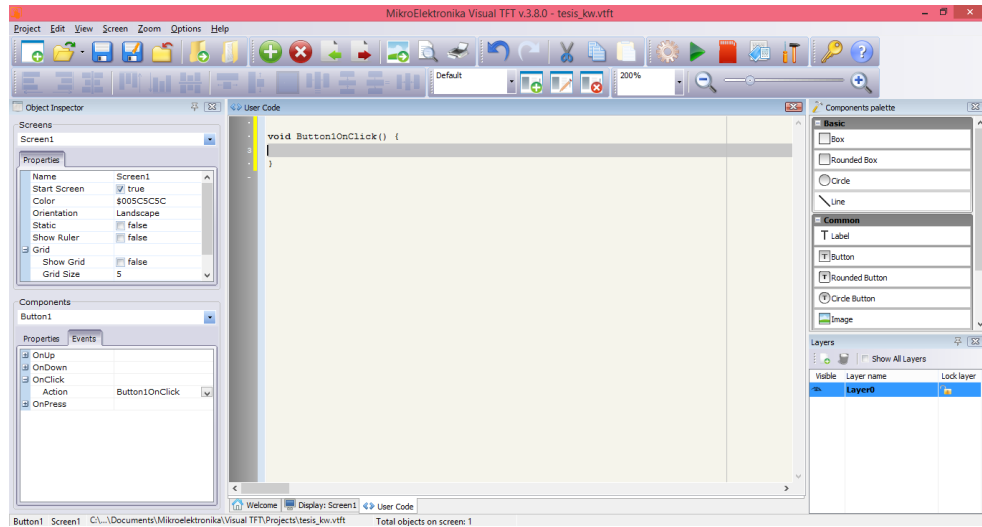
Existe la opción de añadir una acción a cada objeto, dicha acción la realizará cuando sea “presionado” en la pantalla. Luego de añadido el objeto, por ejemplo un botón, en la pestaña *Events* situada en la parte inferior izquierda aparecen cuatro opciones, referentes a los diferentes gestos que desencadena la acción, al seleccionar uno de ellos se despliega una nueva pestaña con el nombre *action*, aquí debe darse doble clic y se crea una función para la acción deseada.

Figura 4. Añadir un botón al *screen*



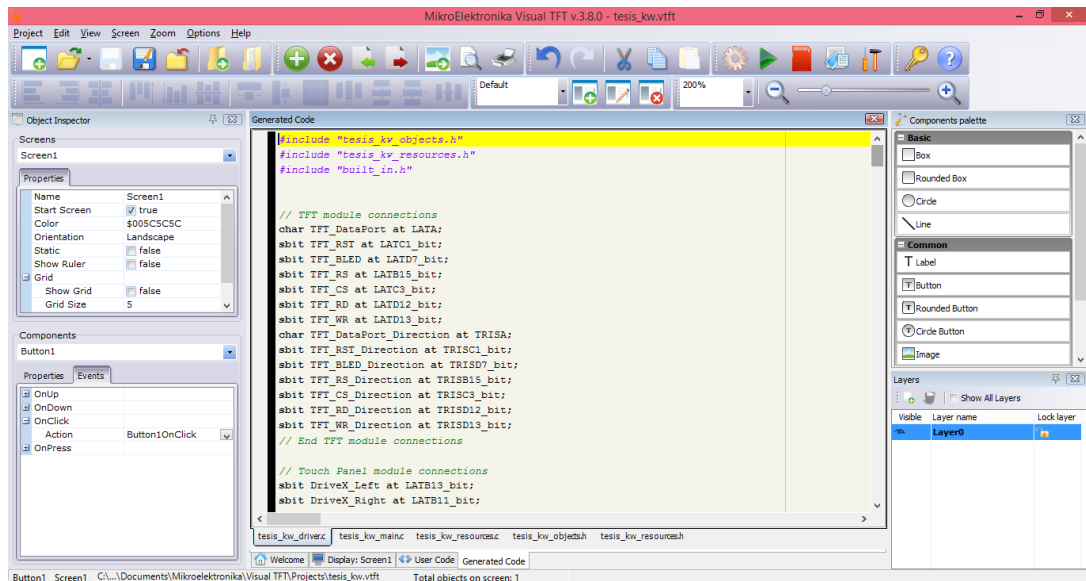
En la pestaña inferior *User Code* aparece el nombre de la función, conformada por el nombre del objeto y la opción que la desencadena, precedido de la palabra clave *void* y seguido por llaves entre las cuales debe describirse el código.

Figura 5. Generar función para el botón



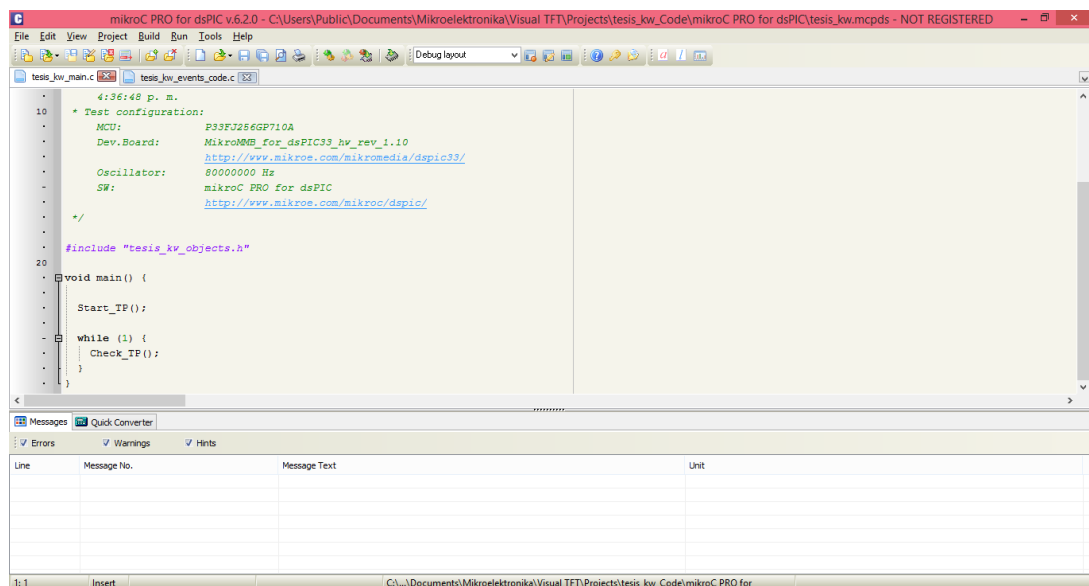
Una vez se ha añadido la cantidad de objetos y *screens* deseadas el usuario puede hacer clic en *Generate Code* cuyo icono está representado por un engrane, en la parte superior derecha.

Figura 6. Generación del *event code*



En la parte inferior de la pantalla aparece una nueva pestaña llamada *Generated Code* que contiene todos los archivos que han sido creados, al hacer clic en el icono *start compiled* representado con un triángulo verde, Visual TFT de MikroElektronika® ejecuta inmediatamente el compilador “mikroC PRO for dsPIC”, este programa mantiene abierto el proyecto que se está desarrollando y muestra dos archivos importantes, “main.c” y “even_code.c”.

Figura 7. Archivo *main* en mikroC for Dspic

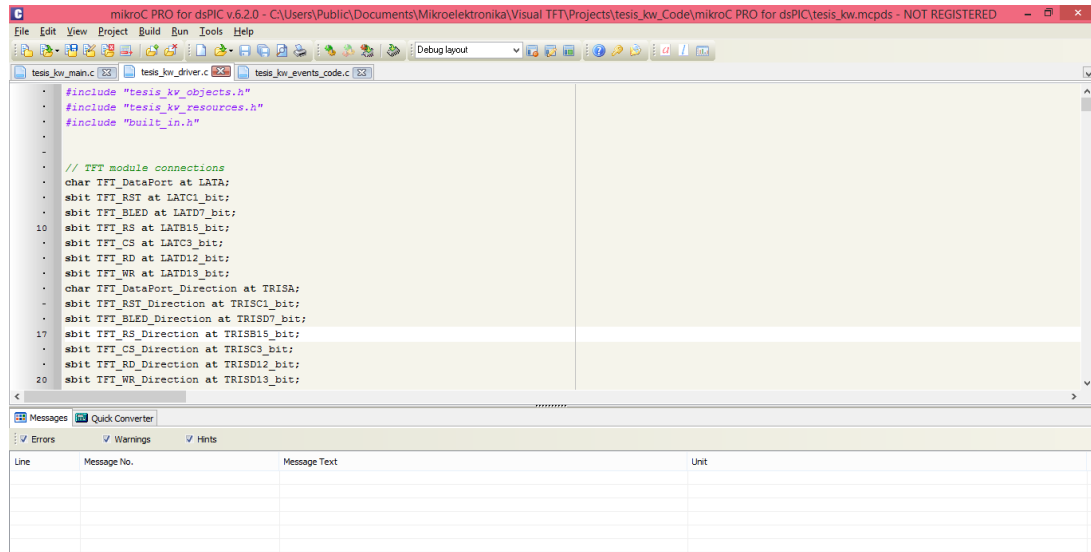


```
4:36:48 p. m.
* Test configuration:
*
* MCU: P33F3256GP710A
* Dev.Board: MikroM8B_for_dsPIC33_hw_rev_1.10
*           http://www.mikroe.com/mikromedia/dspic33/
* Oscillator: 8000000 Hz
* SW: mikroC PRO for dsPIC
*           http://www.mikroe.com/mikroc/dspic/
*/
#include "tesis_kv_objects.h"

void main() {
    Start_TP();
    while (1) {
        Check_TP();
    }
}
```

El archivo “main.c” contiene la función principal *main*. En su interior se puede observar dos funciones *Start_TP* y *Check_TP*, para saber el contenido de estas funciones es necesario abrir otro archivo llamado “driver.c”.

Figura 8. Archivo *driver* en mikroC for Dspic

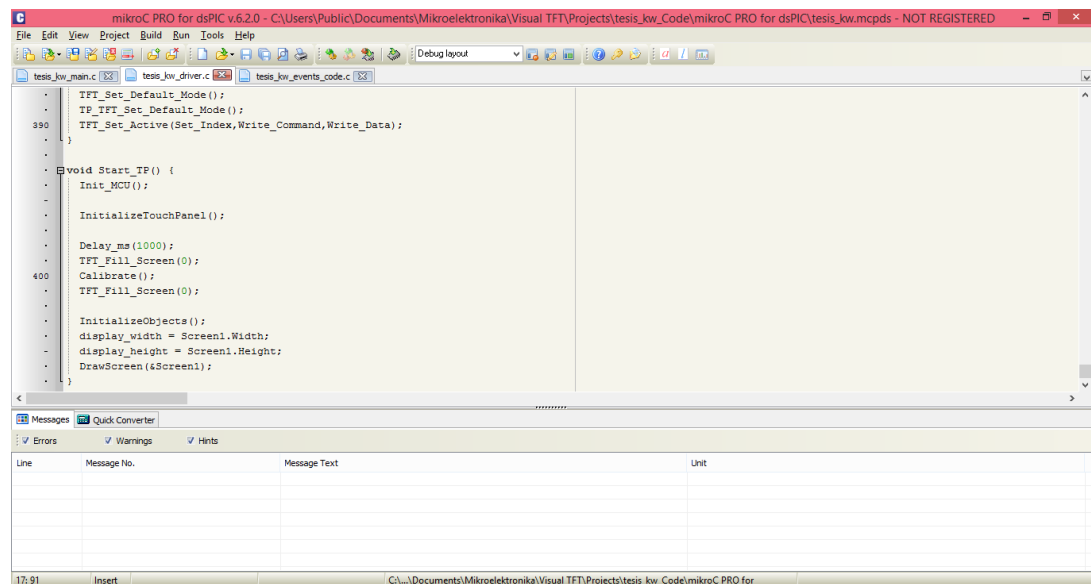


```
#include "tesis_kv_objects.h"
#include "tesis_kv_resources.h"
#include "built_in.h"

// TFT module connections
char TFT_DataPort at LATA;
sbit TFT_RST at LATC1_bit;
sbit TFT_BLED at LATD7_bit;
sbit TFT_RS at LATB15_bit;
sbit TFT_CS at LATC9_bit;
sbit TFT_RD at LATD12_bit;
sbit TFT_WR at LATD13_bit;
char TFT_DataPort_Direction at TRISA;
sbit TFT_RST_Direction at TRISC1_bit;
sbit TFT_BLED_Direction at TRISD7_bit;
sbit TFT_RS_Direction at TRISB15_bit;
sbit TFT_CS_Direction at TRISC9_bit;
sbit TFT_RD_Direction at TRISD12_bit;
sbit TFT_WR_Direction at TRISD13_bit;
```

Este archivo es fundamental para el funcionamiento del sistema de desarrollo dado que aquí están descritas las funciones de inicialización del sistema y otras funciones relacionadas con el entorno gráfico de la pantalla, se recomienda la revisión cuidadosa de este archivo, al final del mismo se encuentra el *Start_TP*.

Figura 9. Función *check_TP*



```
TFT_Set_Default_Mode();
TP_TFT_Set_Default_Mode();
TFT_Set_Active(Set_Index,Write_Command,Write_Data);

void Start_TP() {
    Init_MCU();
    InitializeTouchPanel();
    Delay_ms(1000);
    TFT_Fill_Screen(0);
    Calibrate();
    TFT_Fill_Screen(0);
    InitializeObjects();
    display_width = Screen1.Width;
    display_height = Screen1.Height;
    DrawScreen(&Screen1);
}
```

La función *Check_TP* se encuentra dentro de un bucle infinito dado que se encarga de detectar cualquier interacción del usuario con la pantalla. De este modo si el usuario presiona un botón el cual tiene una acción descrita, *Check_TP* detecta esta interacción y ejecuta la función correspondiente al botón. Si se quiere añadir un código distinto a las acciones cuando se presione alguno de los botones presentes en la pantalla, en el archivo *event_code.c* existe el espacio en donde se puede añadir dicho código.

Figura 10. Añadir código en *event code*

