

Generador de señales arbitrarias a partir de señales definidas

*José León TI**

*Jhonnatan Torres Suárez***

Resumen

En este artículo se describe el desarrollo de un dispositivo que genera cualquier señal definida bajo las leyes de Fourier, lo que significa que la señal debe ser periódica y continua. El mejor método para construir una señal específica, y diferente a las habituales, es el de Joseph Fourier: con la suma de señales comunes y de acuerdo a ciertos coeficientes es posible obtener señales no tan comunes. El dispositivo requiere ingresar los coeficientes que resultan del análisis matemático respectivo.

Palabras clave: arreglos, Fourier, generador de señales, sumador

Abstract

This article describes the development of a device that generates any signal defined under Fourier's laws, which means that the signal must be periodic and continuous. The best method to construct a specific signal, and different from the usual ones, is that of Joseph Fourier: with the sum of common signals and according to certain coefficients it is possible to obtain signals not so common. The device requires entering the coefficients that result from the respective mathematical analysis.

Keywords: Adder, Arrays, Fourier, Signal Generator

* Docente de la Universidad Católica de Colombia. Contacto: jleon@ucatolica.edu.co

** Unisys Corporation. Contacto: Jhonnatan.Torres@unisys.com

Introducción

La reconstrucción de señales especiales es un campo en el que se concentra el análisis armónico. De toda señal que sea continua y periódica se puede obtener una serie de Fourier compuesta por señales convencionales y con diferentes ganancias. Con estas ganancias se diseñan filtros, a los cuales se le ingresan señales seno o coseno, de acuerdo a la serie obtenida. Después se ingresan al sumador para finalmente obtener la señal a la que se le debe realizar un previo análisis matemático.

En el presente documento se relacionan algunos descubrimientos de una investigación en curso de largo aliento que pretende desarrollar un módulo en cuya entrada están las ganancias o coeficientes de la serie de Fourier y, a la salida, la señal correspondiente a la serie. Se explicarán paso a paso los desarrollos o experimentos que se llevaron a cabo para realizar el módulo, los dispositivos que se utilizaron y los respectivos procesos que conllevan y aplican, además del tratamiento de señales que este módulo exige.

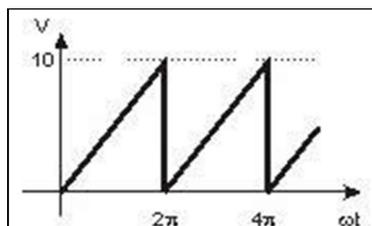
Metodología

Idea general

El primer paso que debe realizar el usuario del módulo es estudiar una señal y obtener la serie de Fourier. Se sugiere el procedimiento descrito

por Valdivia (2007). Por ejemplo, si se utiliza la siguiente señal:

Figura 1. Señal *diente de sierra*



Fuente: Serie de Fourier (2010, s. p.)

Conocida como señal *diente de sierra*, se obtiene la siguiente serie de Fourier:

$$5 - \frac{10}{\pi} \sum_{k=1}^{\infty} \frac{\text{sen}(k\omega)}{k} \quad (\text{Ecuación 1})$$

$$|\omega = (2\pi * f) \quad (\text{Ecuación 2})$$

$$x(t) = 5 - \frac{10}{\pi} \text{sen}(\omega t) - \frac{10}{2\pi} \text{sen}(2\omega t) - \frac{10}{3\pi} \text{sen}(3\omega t) - \dots \quad (\text{Ecuación 3})$$

La cual debe ser transformada a:

$$x(t) = 5 - 3.18309\text{sen}(\omega t) - 1.59154\text{sen}(2\omega t) - 1.06103\text{sen}(3\omega t) - \dots \quad (\text{Ecuación 4})$$

En donde los coeficientes son:

Tabla 1. Coeficientes de trabajo

1	-3,1830
2	-1,5915
3	-1,0610
4	-0,7957
5	-0,6366
6	-0,5355

Fuente: elaboración propia

Con estos datos se puede trabajar con el módulo, ya que estos son los datos de entrada requeridos. Aunque es necesario eliminar el valor DC de la señal, este se reconoce por no estar acompañado de funciones trigonométricas. El caso de la serie mencionada en este documento corresponde a la ecuación número 5 (Smith, 1999).

$$x(t) = 5 - 3.18309\text{sen}(\omega t) - \dots$$

(Ecuación 5)

También existen series cuyos coeficientes A_n y B_n están acompañados de una amplitud. De

Primera idea de desarrollo

La idea primigenia, aquella con la que se planteó el proyecto, era usar un DSP para procesar las señales que se generaban; sin embargo, se reevaluó

hecho, en la serie anteriormente descrita, uno de los coeficientes es igual a 0 (Edwards, 2012).

$$f(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos \frac{n\pi}{T} t + b_n \sin \frac{n\pi}{T} t \right] \quad (\text{Ecuación 6})$$

$$a_n = \frac{1}{T} \int_{-T}^T f(t) \cos \left(\frac{n\pi}{T} t \right) dt \quad (\text{Ecuación 7})$$

El nivel DC está definido por la expresión:

$$a_0 = \frac{1}{T} \int_{-T}^T f(t) dt \quad (\text{Ecuación 8})$$

En estos casos, el resultado es una expresión de la forma: $(1F(\text{Sen}A1a+\text{Cos}A1b))$.

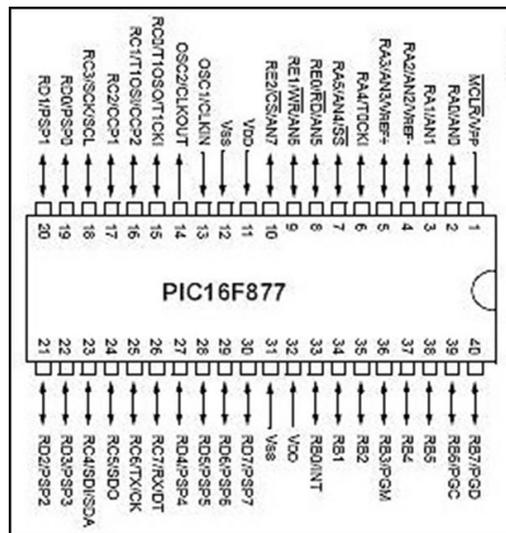
cuando se estudiaron las posibilidades que ofrecen los PSOC, ya que estos pueden ser más útiles gracias a sus bloques análogo-digitales.

Segunda idea de desarrollo

Los coeficientes se ingresaban al sistema mediante un teclado alfa numérico de 16 teclas, conocido en el mercado como de 4 x 4. Los datos se iban a visualizar en una pantalla LCD de

32 caracteres, o de 2 líneas por 16 caracteres. El cerebro encargado de recibir la información y dirigirla era un microcontrolador de la casa Microchip y de referencia 16F877A.

Figura 2. Esquema Microcontrolador 16F877



Fuente: Microchip (2010, s. p.)

Proceso en el PSOC

Proceso de ordenamiento

Para lograr reconstruir el dato fue necesario crear una interrupción para indexar el dato. Por ejemplo, si el dato es positivo:

Datos que envía el Microcontrolador:

Dato Bandera = 9, dato 1, dato 2, dato 3, dato 4.
Si el dato es negativo, el dato bandera cambia a 8.

Con este esquema, los datos eran capturados y se les asignaba una variable para almacenarlos. Finalmente, se organizaban y se conocía el valor introducido por el usuario. Se utilizaron variables formato *float*, que soportan datos con cifras decimales.

El esquema de programación era similar a:

```

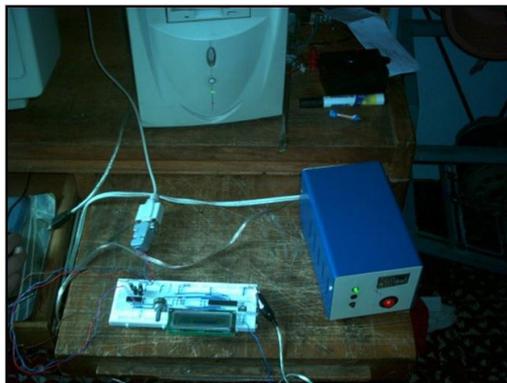
if(data1==8){
float data2 = float data2 * 1;
float data3 = float data3 * (0.1);
float data4 = float data4 * (0.01);
float data5 = float data5 * (0.001);
float final_data = data2+data3+data4+data5;
}
    
```

Figura 3. Pruebas de laboratorio. Transmisión-recepción de datos con protocolo RS-232



Fuente: elaboración propia

Figura 4. Comunicación Protocolo RS 232 en el puerto serie del pc



Fuente: elaboración propia

Proceso de generación de señal

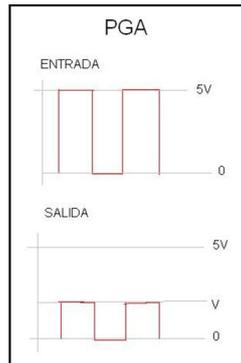
La intención era generar 14 señales sinusoidales o cuadradas, de diferentes frecuencias: 1 kHz, 2 kHz, 3 kHz, etc. Posteriormente, se introducían a los amplificadores de ganancia variable –conocidos como el módulo PGA de los PSOC–, se filtraban

y se sumaban. La ventaja de esta posibilidad es que no se requerían ampliificaciones adicionales y las señales eran de amplitudes mayores (Cajas, Campoverde y Tello, 2012).

La amplitud V es definida por el valor que se ingresó en el microcontrolador y que fue entregado por el protocolo RS-232 al PSOC. Con este valor, y de acuerdo a una tabla de amplitudes

predefinidas, se realizó un algoritmo de análisis y aproximación al valor preestablecido más cercano.

Figura 5. Ejemplo de la señal de salida del módulo PGA del PSOC



Fuente: elaboración propia

Posteriormente, se filtraron las señales cuadradas de distintas amplitudes con ayuda de filtros de ganancia 1 (para no alterar el valor de

amplitud=coeficiente). Finalmente, se sumaron las señales filtradas y se generó la señal.

Proceso final

Las siete señales obtenidas se ingresaron a un sumador análogo, después a una etapa en evaluación, similar a la de los osciloscopios, y

finalmente a un conector BNC para que se pueda adaptar una sonda, conectarla a un osciloscopio y así visualizar la señal.

Inconvenientes

Sin embargo, al realizar las pruebas respectivas de laboratorio con el dispositivo completado, estas no coincidieron con las simulaciones hechas en las aplicaciones matemáticas; apareció una

especie de *efecto espejo* en la señal, lo que hizo reevaluar lo sucedido para realizar más pruebas con los generadores.

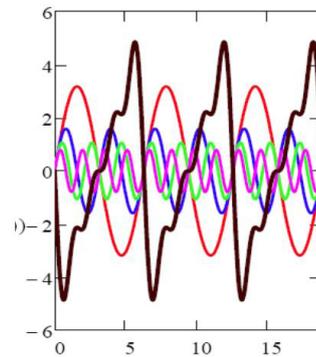
Figura 6. Pruebas con el módulo al ingresar los coeficientes de la onda *diente de sierra*



Fuente: elaboración propia

En la figura 6, se puede visualizar el *efecto espejo* del que se hablaba. Se empieza a notar desde la mitad de la imagen. Está relacionado con los valores de las frecuencias: el error máximo permitido es de 0.1, ya que una variación mayor en las frecuencias empieza a producir el efecto mencionado. En la simulación matemática en MathCad se evidencia el resultado de la suma de las señales, donde es clara la diferencia entre las predicciones matemáticas y experimentales (Comparar figura 6 y figura 7)

Figura 7. Simulación matemática en MathCad



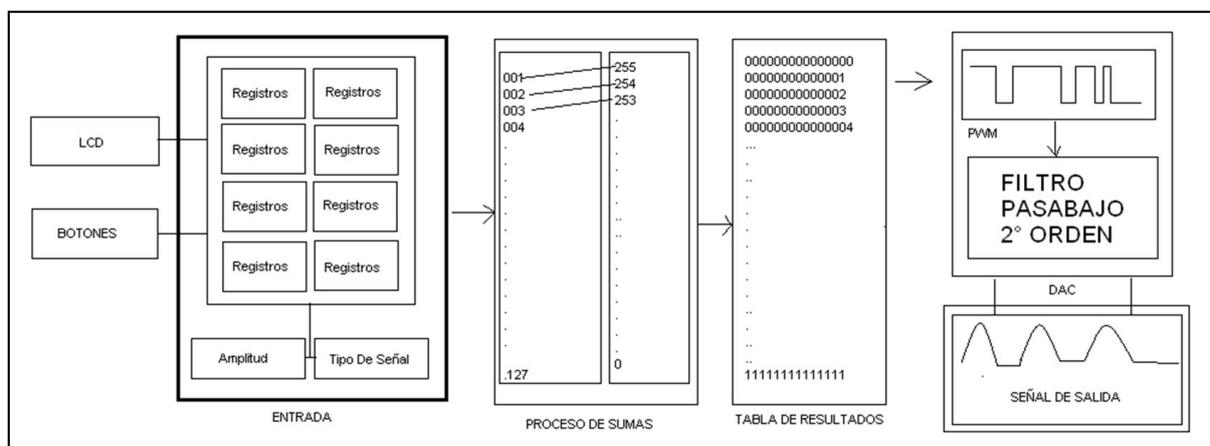
Fuente: elaboración propia

Idea final de desarrollo

Debido a los inconvenientes encontrados con las dos anteriores ideas, fue necesario integrar el dispositivo a la mayor escala. En un foro de Motorola, una persona expuso un proyecto que generaba una onda seno con ayuda de un microcontrolador Motorola, un módulo PWM y un filtro pasabajos. Se trataba del microcontrolador de referencia MC68HC908GP32CB, conocido comúnmente como GP32 (Microchip, 2012).

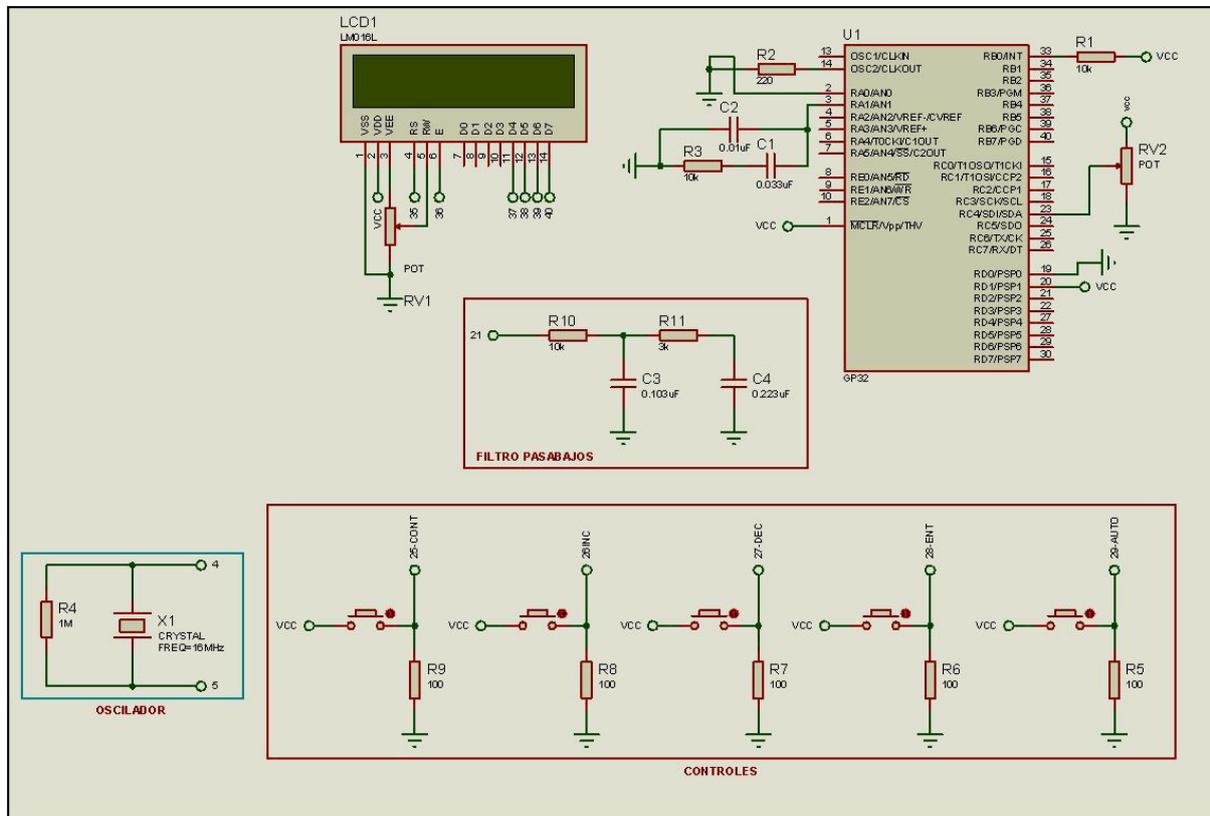
La idea era guardar registros a manera de arreglos en la memoria RAM del microcontrolador. Por ejemplo, para la onda seno eran 256 registros: desde 127 a 255, de 255 a 0 y de 0 a 127. Después se activó el puerto PWM del micro y se adaptó un filtro pasabajos RC de segundo orden. Se utilizó el diseño que se muestra a continuación (figura 8):

Figura 8. Diagrama de bloques del dispositivo



Fuente: elaboración propia

Figura 9. Esquema electrónico del dispositivo



Fuente: elaboración propia

Al adoptar esta idea, se generaron 28 arreglos con registros base de las señales:

- 1F Sen, 2F Sen...7F Sen.
- 1F -Sen, 2F-Sen....7F-Sen.
- 1F Cos, 2F Cos...7FCos
- 1F -Cos, 2F-Cos...7F-Cos

Estas tablas se generaron con ayuda de una función especial del *software* matemático Matlab. Se definieron 10 amplitudes, la posibilidad de

seleccionar la función y un valor de registro para asignarlo a la tabla correspondiente. También se agregó una función especial de autoajuste, que calcula las amplitudes mayores de la señal y se autoescala para generar una señal más grande. Estos valores en las tablas son multiplicados por la amplitud ingresada y cada uno de ellos se suma para indicar los valores obtenidos al módulo PWM. Por ejemplo, si la amplitud ingresada es 3 para 2 señales, el proceso del módulo es el siguiente:

Tabla 2. Proceso del módulo

Posición	Señal 1	Amplitud 1	V total 1
0	127	3	381
1	128	3	384
2	129	3	387
Posición	Señal 2	Amplitud 2	V total 2
0	0	3	0
1	1	3	3
2	2	3	6

Fuente: elaboración propia

Con un contador se hace el rastreo de las posiciones: contador 0 = valor 0. En la primera tabla es 381 y en la segunda es 0; el valor resultante es = 381. A continuación, se presenta la suma de los valores totales:

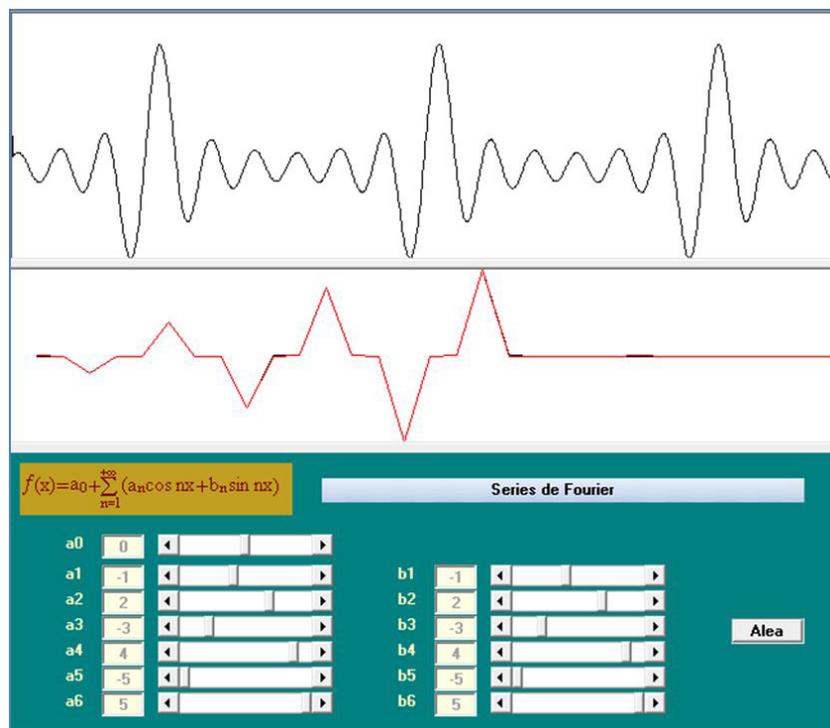
Tabla 3. Suma de valores totales

Posición	V Total 3
0	381
1	387
2	393

Fuente: elaboración propia

Como se evidencia en las tablas anteriores, la señal resultante es muy similar a la señal 2. Este método también es conocido como *suma punto a punto*. Estos conceptos se programaron en un *software* conocido como Microgrades, el cual trabaja de manera gráfica y se enfoca en programar microcontroladores Motorola. El resultado de la simulación es el siguiente (figura 10):

Figura 10. Simulación, espectro y coeficientes señal 1



Fuente: elaboración propia

Microgrades es un entorno que simplifica el desarrollo de soluciones tecnológicas mediante la computación embebida. Esta última implica la resolución de problemas a través del modelamiento basado en conocimientos aplicables:

1. Control lógico y secuencial
2. Adquisición de datos comunicaciones seriales
3. Algoritmia e informática

Resultados

- Las pruebas fueron un éxito: no apareció el *efecto espejo* de los experimentos anteriores y el módulo quedó mucho más reducido.
- Las amplitudes se dan en valores enteros: si la amplitud real contiene decimales, estos se

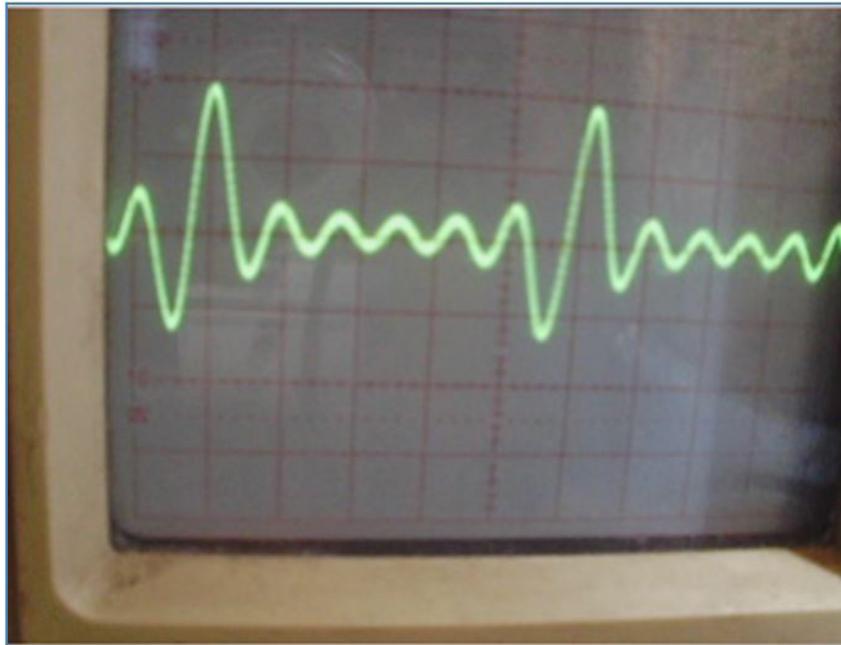
4. Manejo de periféricos

En este *software* se programaron las tablas con los registros, las acciones de acuerdo a los pulsadores y se activó el módulo PWM. Este ejercicio arrojó un resultado más integrado, que consta de una LCD, 4 pulsadores y un filtro RC, a diferencia de los anteriores diseños, para los que se contabilizaron más de 10 chips o pastillas integradas.

aproximan al valor entero más cercano por arriba; por ejemplo $1,6 = 2$.

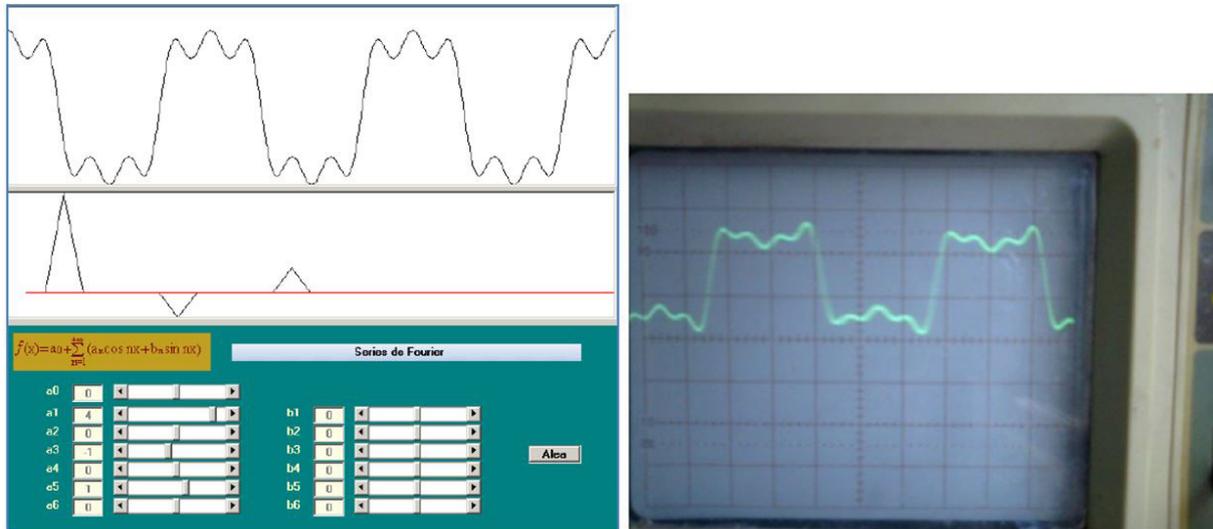
- El valor de las amplitudes no puede ser mayor a 10 (en el módulo), y en medida equivale a 5 voltios.

Figura 11. Resultado práctico



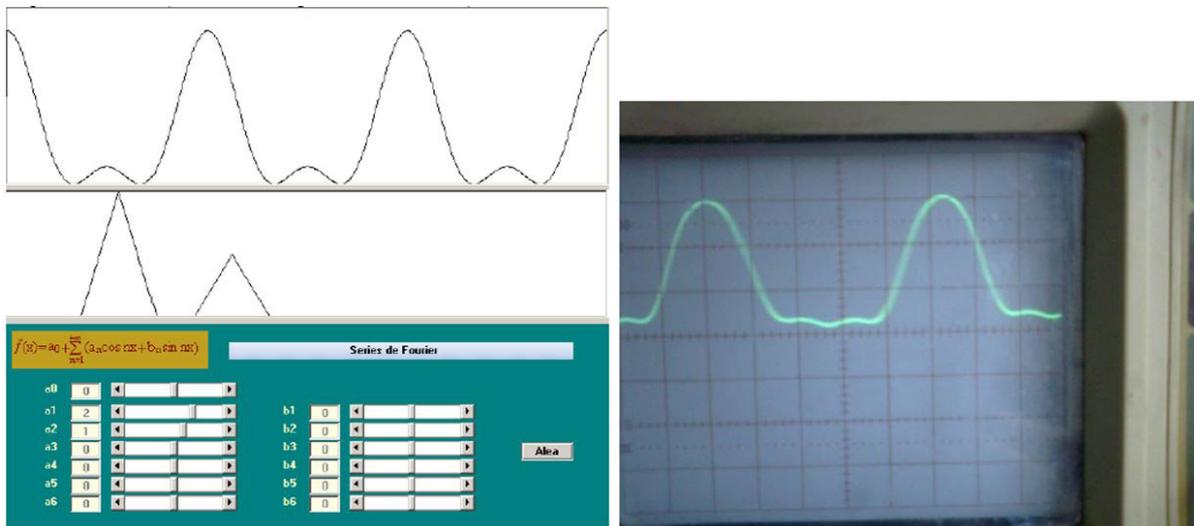
Fuente: elaboración propia

Figura 12. Resultado señal 2



Fuente: elaboración propia

Figura 13. Resultado señal 3



Fuente: elaboración propia

Conclusiones

- Para evitar cualquier tipo de distorsión en la señal es necesario integrar los módulos a la mayor escala posible, así como diseñar los módulos dentro de un mismo circuito integrado o microcontrolador.

- Debido a que se integró el módulo a la mayor escala posible, el resultado fue un dispositivo mucho más reducido a los anteriores.
 - La idea de Fourier es general, ya sea en un simulador matemático, en una hoja de Excel o en un Applet de Java: siempre es el mismo proceso.
- Se puede mejorar el método de uso si se crea una base de datos con las señales más utilizadas, pues se transformaría una serie matemática con sus símbolos y funciones trigonométricas a una tabla con 7 números que definen la señal.

Referencias

- Cajas, R. y Campoverde, F. (2012). *Diseño e implementación de una tarjeta de adquisición de datos de alta velocidad y su aplicación en el diseño de laboratorios académicos basados en software y hardware libre* (trabajo de grado). Universidad Politécnica Salesiana, Cuenca, Ecuador. Recuperado de <https://docplayer.es/28121050-Facultad-de-ingenierias.html>
- Edwards, R. (2012). *Fourier Series: A Modern Introduction* (vol. 2). Nueva York: Springer Science & Business Media. ISBN 978-1-4613-8156-3
- Microchip. (2012). *Using PWM to Generate Analog Output* (recurso en línea). Recuperado de <http://ww1.microchip.com/downloads/en/appnotes/00538c.pdf>
- Microchip. (2010). *PIC 16F877 Datasheet* (recurso en línea). Recuperado de <https://www.microchip.com/wwwproducts/en/PIC16F877>
- Serie de Fourier. (2010). *Serie de Fourier de tiempo continuo* (recurso en línea). Recuperado de <http://www.seriedefourier.com.ar/>
- Smith, S. (1999). *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego: California Technical Publishing. ISBN: 0966017633
- Valdivia, C. (2007). Jean Baptiste Fourier: la armonía en la historia de la ciencia. *Revista Digital Acta*, 45, 55-60. Recuperado de https://www.acta.es/medios/articulos/biografias_y_personajes/045055.pdf