



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SISTEMA DE POSGRADO**

**MAESTRÍA EN TELECOMUNICACIONES**

**TEMA:**

**Diseño de un sintetizador digital directo de frecuencia controlado con  
Arduino nano para un radioreceptor definido por software**

**AUTORA:**

**María Isabel Rosado Cevallos**

**Trabajo de titulación previo a la obtención del grado de  
Magister en Telecomunicaciones**

**TUTOR:**

**MSc. Manuel Romero Paz**

Guayaquil, a los 10 días de noviembre de 2017



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SISTEMA DE POSGRADO**  
**MAESTRÍA EN TELECOMUNICACIONES**

**CERTIFICACIÓN**

Certificamos que el presente trabajo fue realizado en su totalidad María Isabel Rosado Cevallos como requerimiento parcial para la obtención del Título de Magíster en Telecomunicaciones.

TUTOR

---

**MSc. Manuel Romero Paz**

DIRECTOR DEL PROGRAMA

---

**MSc. Manuel Romero Paz**

Guayaquil, a los 10 días de noviembre de 2017



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES**

**DECLARACIÓN DE RESPONSABILIDAD**

YO,

María Isabel Rosado Cevallos

DECLARO QUE:

El trabajo de Titulación “**Diseño de un Sintetizador Digital Directo de frecuencia controlado con Arduino Nano para un radioreceptor definido por *software*.**”, previa a la obtención del Título de **Magíster en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 10 días de noviembre de 2017

LA AUTORA

---

María Isabel Rosado Cevallos



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES**

AUTORIZACIÓN

YO, María Isabel Rosado Cevallos

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación**, en la biblioteca de la institución del Trabajo de Titulación de Titulación, “**Diseño de un Sintetizador Digital Directo de frecuencia controlado con Arduino Nano para un radiorreceptor definido por *software*.**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 10 días de noviembre de 2017

LA AUTORA

---

María Isabel Rosado Cevallos

# REPORTE DE URKUND

**URKUND**

**Documento** [Tesis Sintetizador Digital Directo.docx](#) (D31516118)

**Presentado** 2017-10-20 15:48 (-05:00)

**Presentado por** fernandopm23@hotmail.com

**Recibido** edwin.palacios.ucsg@analysis.orkund.com

**Mensaje** Revisión TT Rosado María [Mostrar el mensaje completo](#)

1% de estas 30 páginas, se componen de texto presente en 1 fuentes.

**Lista de fuentes** Bloques

+	Categoría	Enlace/nombre de archivo	▢
+	>	<a href="#">tesis Marcador electronico edison toapanta.docx</a>	✓
+	<b>Fuentes alternativas</b>		
+	<b>Fuentes no usadas</b>		

Reiniciar Exportar Compartir ?

1 Advertencias

SISTEMA DE POSGRADO

MAESTRÍA EN TELECOMUNICACIONES

TEMA: Diseño de un sintetizador digital directo de frecuencia controlado con Arduino nano para un radioreceptor definido por software

AUTORA: María Isabel Rosado Cevallos

Trabajo de titulación previo a la obtención del grado de Magister en Telecomunicaciones

TUTOR: MSc. Manuel Romero Paz

Guayaquil, a los XX días de XX de XX de 2017

SISTEMA DE POSGRADO MAESTRÍA EN

## **Dedicatoria**

A mi amado hijo Caleb por ser mi fuente de motivación e inspiración para poder superarme cada día más, a mi familia y todas aquellas personas que fueron parte de este sueño que es tan importante para mí, quienes dieron todas sus ayudas, sus palabras motivadoras, sus conocimientos, sus consejos y su dedicación.

## **Agradecimientos**

Quiero agradecer en primer lugar a Jehová, por guiarme y fortalecerme espiritualmente para empezar un camino lleno de éxito.

A mi esposo que con todo el cariño y amor creyó en mi capacidad para alcanzar esta meta gracias por su sacrificio y esfuerzo.

A mis compañeros y amigos presentes y pasados quienes sin esperar nada a cambio compartieron su conocimiento, alegrías y tristezas.

A mi tutor de proyecto, quien con su conocimiento y su guía fue una pieza clave para que pudiera desarrollar cada etapa del trabajo.



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

**SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES**

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

**MSc. Manuel Romero Paz**

TUTOR

f. \_\_\_\_\_

**MSc. Manuel Romero Paz**

DIRECTOR DEL PROGRAMA

f. \_\_\_\_\_

**MSc. Celso Bohórquez Escobar**

REVISOR

f. \_\_\_\_\_

**MSc. Edwin Fernando Palacios Meléndez**

REVISOR



## RESUMEN

En este trabajo se presenta el diseño de un sintetizador de frecuencias basado en la tecnología de Síntesis Digital Directa. Con este fin se analizaron aspectos generales de los distintos métodos de síntesis de frecuencia, profundizando en la teoría de Síntesis Digital Directa (DDS por las siglas en inglés de *Direct Digital Synthesize*). Se realizó la selección de los circuitos integrados más adecuados para satisfacer los requerimientos del diseño y se escogió al Arduino Nano V3 para su empleo en la sección de control. Se diseñó una aplicación en la plataforma Visual Studio para permitir al usuario controlar los cambios de frecuencia del dispositivo. Se realizaron varias mediciones de laboratorio que evidenciaron la precisión y calidad de la señal generada. El trabajo es exploratorio y descriptivo, debido a que pretende explorar las características y evolución de la DDS de frecuencias. Es descriptivo, puesto que permite analizar, diseñar y evaluar las diferentes tecnologías que aplican este método en la generación de señales con frecuencias estables. Es del paradigma Empírico-Analítico con un enfoque Cuantitativo porque utiliza cálculos matemáticos para el diseño electrónico de los diferentes bloques funcionales que conforman el dispositivo para la DDS de frecuencias. Y es Experimental, debido a que se alteran las variables de estudio, para finalmente proceder con el análisis respectivo.

**Palabras clave:** DDS, Arduino Nano V3, DAC, DDS, Sintetizadores, Conversor

## *ABSTRACT*

*In this paper we present the design of a frequency synthesizer based on Direct Digital Synthesis technology. For this purpose we analyzed general aspects of the different methods of frequency synthesis, deepening the theory of Digital Direct Synthesis (DDS). The selection of the most suitable integrated circuits to satisfy the design requirements was made and the Arduino Nano V3 was chosen for use in the control section. An application was designed on the Visual Studio platform to allow the user to control device frequency changes. Several laboratory measurements were performed that demonstrated the accuracy and quality of the signal generated. The work is exploratory and descriptive, because it pretends to explore the characteristics and evolution of the DDS of frequencies. It is descriptive, since it allows to analyze, design and evaluate the different technologies that apply this method in the generation of signals with stable frequencies. It is of the Empirical-Analytical paradigm with a Quantitative approach because it uses mathematical calculations for the electronic design of the different functional blocks that make up the device for the DDS of frequencies. Y is Experimental, because the study variables are altered, to finally proceed with the respective analysis.*

**Keywords:** *DDS, Arduino Nano V3, DAC, DDS, Synthesizers, Converter*

## ÍNDICE GENERAL

ÍNDICE DE FIGURAS .....	XIII
ÍNDICE DE TABLAS .....	XIV
CAPITULO 1 . Descripción del proyecto de intervención. ....	15
1.1 Justificación del problema a investigar. ....	15
1.2 Antecedentes. ....	16
1.3 Definición del problema. ....	16
1.4 Objetivos.....	16
1.4.1 Objetivo General. ....	17
1.4.2 Objetivos específicos.....	17
1.5 Hipótesis. ....	17
1.6 Metodología de investigación. ....	17
CAPITULO 2 . Fundamentación teórica. ....	19
2.1 Sintetizadores de frecuencia indirectos. ....	19
2.2 Sintetizadores de frecuencia directos. ....	20
2.3 Teoría de la síntesis digital directa de frecuencias. ....	21
2.3.1 El acumulador de fase. ....	23
2.3.2 El convertidor fase-amplitud. ....	25
2.3.3 El conversor Digital–Analógico (DAC). ....	26
2.4 Frecuencias espurias generadas en los DDS.....	28
2.4.1 Efecto de la resolución del DAC, error de cuantificación. ....	28
2.4.2 El efecto del truncamiento en el acumulador de fase.....	29
2.4.3 No linealidad del DAC. ....	31
2.4.4 Señales espurias a la salida del DAC. ....	32
2.4.5 Otras fuentes de ruido.....	32
2.5 Ventajas y desventajas de los DDS. ....	33
2.6 Fundamentos de la Radio Definida por <i>Software</i> . ....	34
2.6.1 Origen y desarrollo.....	35
2.7 Definiciones de Radio Definida por <i>Software</i> . ....	36
2.8 Arquitectura de un sistema de Radio Definida por <i>Software</i> . ....	38
CAPITULO 3 . Descripción de la implementación. ....	40
3.1 Generalidades del circuito integrado AD9850.....	41
3.1.1 Descripción de terminales del AD9850.....	43
3.2 Aplicación del AD9850.....	45

3.2.1	Programación del AD9850. ....	47
3.2.2	Configuración externa del AD9850 como generador de reloj. ....	48
3.3	Arduino Nano V3. ....	49
3.3.1	Características técnicas. ....	49
3.3.2	Alimentación: ....	50
3.3.3	Entradas y salidas. ....	50
3.3.4	Comunicación ....	51
3.3.5	Programación ....	52
3.3.6	¿Por qué emplear Arduino y no otra plataforma de desarrollo? ....	52
3.4	Multiplicador de frecuencia con circuito integrado CDC5801. ....	53
3.4.1	Descripciones de terminales del CDC5801. ....	56
3.4.2	Configuración del CDC5801. ....	58
CAPITULO 4 . Diseño de <i>software</i> de control para el DDS y mediciones físicas. ....		59
4.1	<i>Software</i> de control para el cambio de frecuencia del DDS. ....	59
4.1.1	Entorno de desarrollo de aplicaciones Microsoft Visual Studio. ....	60
4.1.2	Comunicación entre el Arduino y el <i>software</i> de control. ....	61
4.1.3	Propuesta de comunicación entre el Arduino y el AD9850 y el CDC 5801. .	65
4.2	Análisis del desempeño del dispositivo mediante mediciones físicas ....	66
CONCLUSIONES Y RECOMENDACIONES ....		70
REFERENCIAS BIBLIOGRÁFICAS ....		72
GLOSARIO DE TÉRMINOS. ....		74
ANEXOS ....		76

## ÍNDICE DE FIGURAS

Figura 2.1. Esquema en bloques de un sintetizador de frecuencias PLL de un solo lazo. ....	20
Figura 2.2. Sintetizador de frecuencia directo de un solo cristal u oscilador. ....	21
Figura 2.3. Ejemplo de esquema básico de DDS y señales generadas. ....	22
Figura 2.4. Rueda de fase digital. ....	23
Figura 2.5. Circuito DAC de 4 bits basado en un circuito escalera R-2R. ....	27
Figura 2.6. Efecto denominado glitch. ....	27
Figura 2.7. Ruido de cuantificación. ....	28
Figura 2.8. Error de truncamiento de fase y la rueda de fase. ....	30
Figura 2.9. Diagrama de bloques funcionales de SDR. ....	38
Figura 3.1. Diagrama en bloques del módulo DDS completo. ....	40
Figura 3.2. Diagrama en bloques funcional del AD9850. ....	42
Figura 3.3. Dimensiones y encapsulado del CI AD9850 en milímetros. ....	43
Figura 3.4. Distribución de terminales del AD9850. ....	43
Figura 3.5. Diagrama funcional del AD9850. ....	46
Figura 3.6. Función básica del AD9850 como generador de reloj. ....	48
Figura 3.7. Vista superior y lateral del Arduino Nano V3. ....	49
Figura 3.8. Distribución de terminales del Arduino Nano V3. ....	51
Figura 3.9. Distribución de terminales del CDD5801. ....	56
Figura 4.1. Esquema en bloques de la comunicación del sistema. ....	59
Figura 4.2. Interfaz gráfica de la aplicación de control. ....	61
Figura 4.3. Señal de salida del oscilador de referencia. ....	66
Figura 4.4. Medición de la señal de onda cuadrada de 5 MHz a la salida del DDS. ....	67
Figura 4.5. Señal de 20 MHz a la salida del multiplicador. ....	68
Figura 4.6. Espectro de frecuencias de la señal sinusoidal de 5 MHz. ....	68
Figura 4.7. Espectro de frecuencias de la señal sinusoidal de 20 MHz. ....	69

## ÍNDICE DE TABLAS

Tabla 3.1.Descripción de terminales del AD9850. ....	45
Tabla 3.2..Códigos de control de pruebas reservados internamente. ....	49
Tabla 3.3.Modos de operación.....	56
Tabla 3.4.Coeficiente de multiplicación.....	57
Tabla 3.5.Coeficiente de división.....	57
Tabla 3.6.Configuración de los terminales del CDC5801.....	58
Tabla 4.1.Peticiones HTTP más usadas en REST.....	64
Tabla 4.2.Terminales usados para la comunicación entre Arduino y los CI .....	67

## **CAPITULO 1 . Descripción del proyecto de intervención.**

En el presente capítulo se justificará y delimitará el problema a investigar, se establecerán los objetivos generales y específicos, la hipótesis y la metodología de investigación.

### **1.1 Justificación del problema a investigar.**

El método de Síntesis Digital Directa es una técnica que se ha extendido en los últimos años gracias a los avances alcanzados en la tecnología de fabricación de los circuitos integrados. Además de proporcionar alta velocidad de generación, alto rendimiento, costo competitivo y pequeño tamaño, los generadores basados en este método se han convertido rápidamente en una alternativa atractiva respecto a los sintetizadores por lazo de enganche de fase (PLL).

Un dispositivo basado en la Síntesis Digital Directa (DDS por sus siglas en inglés) permite generar señales de alta precisión y pureza desde una representación digital. La forma de onda representada digitalmente se reconstruye con un DAC (*Digital-Analog Converter*) o conversor digital analógico, que hoy en día se fabrican con altas velocidades, de modo que han permitido la generación digital directa de señales de radiofrecuencias (RF) con un bajo nivel de ruido, alta resolución y excelente estabilidad.

Las aplicaciones de la Síntesis Digital Directa de frecuencia son muchas, desde un generador de señal para un laboratorio, un oscilador de referencia variable para un PLL en VHF (*Very High Frequency*) o SHF (*Super High Frequency*), hasta un oscilador local para un receptor de Radio Definido por *Software* (SDR por sus siglas en inglés) en la banda de HF (*High Frequency*), que es el caso motivo del presente proyecto.

## **1.2 Antecedentes.**

El avance tecnológico en el campo de las radiocomunicaciones durante las últimas décadas ha sido realmente asombroso. Un factor decisivo fue la introducción de los Sintetizadores Digitales Directos (DDS) para obtener señales de RF de frecuencia variable a partir de una frecuencia de referencia extremadamente estable, generalmente un oscilador a cristal.

Los DDS presentan varias ventajas únicas, como la posibilidad de obtener una resolución de frecuencia del orden de mili-Hertz e incluso de nano-Hertz, una extremada rapidez de conmutación de una frecuencia a otra, además su arquitectura digital elimina la necesidad de que el ajuste de la sintonización del sistema sea manual permitiendo una interfaz de control digital donde los sistemas son operados de forma remota y minuciosamente optimizados, bajo el control de un microcontrolador.

Uno de los dispositivos basados en el método DDS más novedosos y distintivos es el AD9850 del fabricante *Analog Devices*. Este es un circuito integrado de última generación que utiliza la más avanzada tecnología DDS junto a un comparador y un Conversor Digital Analógico de alta velocidad.

Para enviar información de control al AD9850, se puede usar un microcontrolador que maneje el cambio de la frecuencia del dispositivo mediante la comunicación con una aplicación de escritorio diseñada para fácil manejo del usuario del sistema.

## **1.3 Definición del problema.**

Necesidad de diseñar un oscilador local programable digitalmente para un radorreceptor definido por *software* que garantice estabilidad en frecuencia.

## **1.4 Objetivos.**

A continuación, se detalla el objetivo general y los objetivos específicos:



### 1.4.1 Objetivo General.

Diseñar un Sintetizador Digital Directo de frecuencia, usando los circuitos integrados AD9850 y CDC8501 controlados con Arduino Nano, para su uso como oscilador local en un radiorreceptor definido por *software*.

### 1.4.2 Objetivos específicos.

- ✓ Caracterizar los sintetizadores de frecuencias directos.
- ✓ Analizar el método de Síntesis Digital Directa (DDS) de frecuencias, sus ventajas y desventajas.
- ✓ Plantear los requerimientos del diseño.
- ✓ Seleccionar los componentes de acuerdo a los requerimientos del diseño.
- ✓ Caracterizar el Arduino Nano V3 para su empleo en la sección de control.
- ✓ Diseñar la aplicación de usuario o *software* de control.
- ✓ Analizar el desempeño del dispositivo mediante mediciones físicas.

### Hipótesis.

¿Es posible diseñar un Sintetizador Digital Directo de frecuencia usando los circuitos integrados AD9850 y CDC8501 controlados con Arduino Nano que satisfaga los requerimientos de estabilidad para su uso como oscilador local en un SDR?.

### 1.5 Metodología de investigación.

El actual trabajo investigativo previo al grado académico de Magister en Telecomunicaciones, es exploratorio y descriptivo. Se dice “**EXPLORATORIO**”, debido a que pretende explorar las características y evolución de la Síntesis Digital Directa (DDS) de frecuencias . Es “**DESCRIPTIVO**”, puesto que permite analizar, diseñar y evaluar las diferentes tecnologías que aplican este método en la generación de señales con frecuencias estables.

Además, esta investigación es del paradigma “**EMPÍRICO-ANALÍTICO**” con un enfoque “**CUANTITATIVO**” porque se utilizan cálculos matemáticos para el diseño electrónico de los diferentes bloques funcionales que conforman el dispositivo para la Síntesis Digital Directa de frecuencias. Estos cálculos se comprueban posteriormente a través de mediciones realizadas en el laboratorio.

Finalmente el diseño del presente trabajo investigativo es “**EXPERIMENTAL**”, debido a que se alteran las variables de estudio, para finalmente proceder con el análisis respectivo.

## **CAPITULO 2 . Fundamentación teórica.**

En décadas anteriores las técnicas de síntesis de frecuencia eran consideradas una novedad y sólo se utilizaban en equipos sofisticados y complejos. En nuestros días, esta técnica se emplea en prácticamente todas las esferas de la electrónica.

Según su funcionamiento los sintetizadores se dividen en Sintetizador Directo Analógico, Sintetizador Digital Directo y Sintetizador Indirecto con PLL (*Phase Locked Loop*). Las dos últimas técnicas son las más utilizadas en la actualidad para la síntesis de frecuencia y se explican a continuación.

### **2.1 Sintetizadores de frecuencia indirectos.**

En la síntesis indirecta de frecuencias se usa un divisor-multiplicador controlado por retroalimentación (como por ejemplo un PLL), para generar varias frecuencias de salida. La síntesis de frecuencia indirecta es más lenta y susceptible al ruido, sin embargo, es menos costosa y requiere filtros menos complicados, que la síntesis directa de frecuencias.

- **Sintetizadores de frecuencias con lazo de fase cerrada.**

En la figura 2.1 se muestra un diagrama de bloques para un sintetizador de frecuencias PLL de un solo lazo que trabaja el rango de 1MHz a 10 MHz.

Se parte de una referencia de frecuencia estable constituida por un oscilador controlado por cristal. El intervalo de frecuencias generadas, y la resolución, dependen de la red divisora y de la ganancia de lazo abierto. El divisor de frecuencias es un circuito que divide por  $n$ , donde  $n$  es cualquier número entero. La forma más simple de circuito divisor es un contador digital de subida y bajada con frecuencia de salida  $f_c=f_0/n$  donde  $f_0$  es la frecuencia de salida del VCO (Oscilador Controlado por Voltaje).

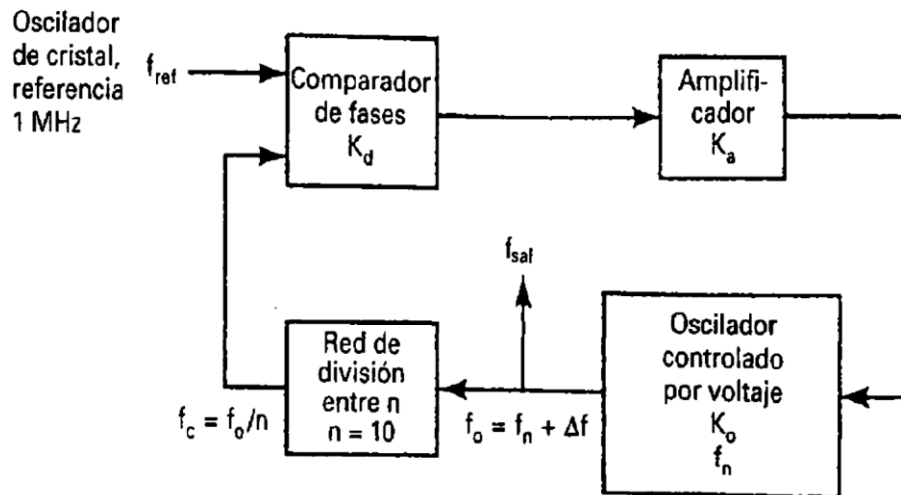


Figura 2.1. Esquema en bloques de un sintetizador de frecuencias PLL de un solo lazo.  
Fuente: W. Tomasi, 2003

Como todo PLL, el dispositivo ha alcanzado el enganche cuando  $f_c = f_{ref}$ . Entonces la señal de salida  $f_{sal}$  tiene una frecuencia dada por  $f_o = n * f_{ref}$ . De esta forma, este sintetizador indirecto es esencialmente un multiplicador de frecuencia por  $n$ . (W. Tomasi, 2003)

## 2.2 Sintetizadores de frecuencia directos.

En la síntesis directa de frecuencias la señal de salida se obtiene por sucesivas multiplicaciones, divisiones y mezcla de frecuencias que se generan a partir del oscilador de referencia. Este conjunto de elementos proporciona una elevada estabilidad y rapidez en el cambio de frecuencia de salida. Dentro de este método de síntesis existen dos variantes generales: los sintetizadores de frecuencia de cristal múltiple y los de un solo cristal. En la figura 2.2 se muestra un ejemplo de este último.

La resolución y rango de frecuencias están en función del número y tipo de circuitos que lo componen. Los mezcladores que se usan son moduladores balanceados, con filtros de salida que se sintonizan a la suma de las dos frecuencias de entrada.

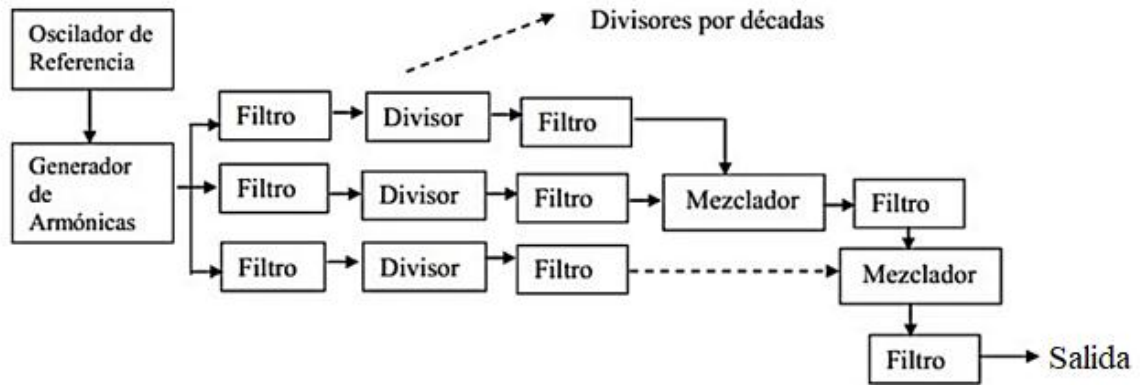


Figura 2.2. Sintetizador de frecuencia directa de un solo cristal u oscilador.

Fuente: La autora

La Síntesis Digital Directa de frecuencias, se puede definir básicamente como la forma de generar señales de alta precisión y pureza desde una representación digital. La forma representada digitalmente se reconstruye con un convertidor digital-analógico de alta velocidad. El convertidor proporciona una señal de salida analógica, generalmente sinusoidal.

En el DDS, la señal se genera y manipula digitalmente desde “cero”. Por lo tanto, las muestras que luego conformarán la señal se generan digitalmente en forma numérica y no a partir del muestreo de una señal analógica proveniente de un oscilador. Una vez terminados todos los procesos sobre las muestras digitales generadas, éstas se convierten a una señal analógica a través de un conversor digital analógico. Esto denota una diferencia fundamental entre la generación del DDS y las otras técnicas de síntesis de frecuencia. (R.F Guilló, C.M Puliafito, P.G Aliquó, J.L Puliafito, 2012)

### 2.3 Teoría de la síntesis digital directa de frecuencias.

Existe una gran variedad de implementaciones de DDS, la más frecuente consiste en cuatro bloques fundamentales como puede apreciarse en la figura 2.3. Todo el sistema trabaja con un reloj de referencia que genera la frecuencia empleada en el muestreo y para la sincronización de las operaciones que

realiza todo el sistema. La teoría de los DDS está basada en la acumulación de cambios de fase y su reproducción en una forma de onda digitalizada.

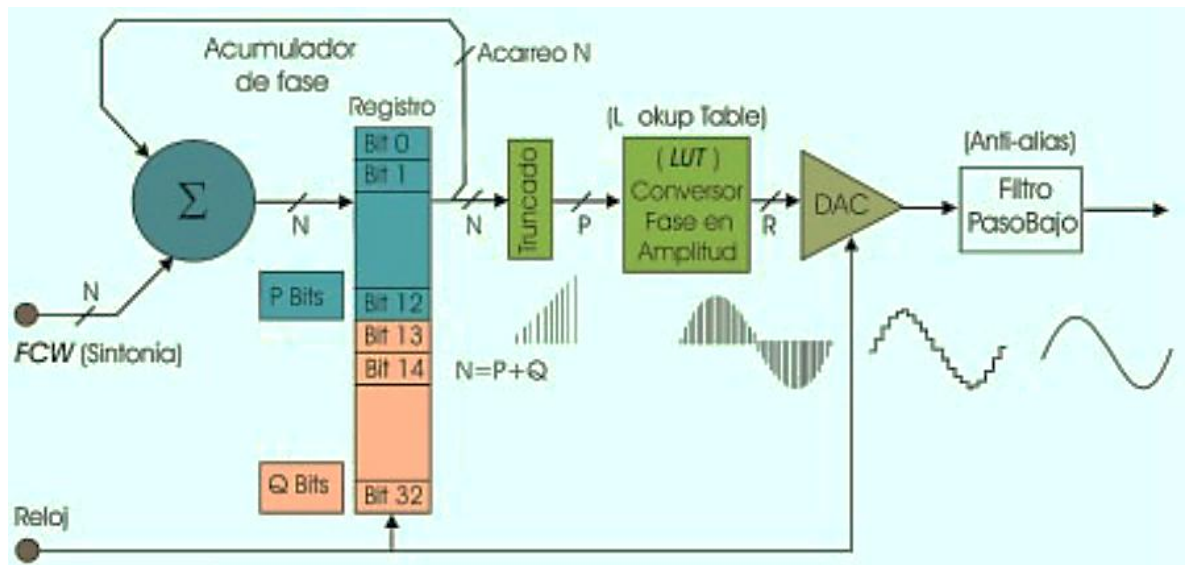


Figura 2.3. Ejemplo de esquema básico de DDS y señales generadas.

Fuente: S. Pagel, 2007

En la figura 2.3 se aprecian los siguientes bloques:

1. Un reloj (CLK) que genera la frecuencia empleada en el muestreo y para sincronizar las operaciones a ser realizadas por el sistema.
2. Un acumulador de fase que compara la frecuencia del reloj con la frecuencia que se desea generar, impresa en la palabra binaria de sintonía, se trata de Un dispositivo que realiza operaciones aritméticas sencillas.
3. La denominada “lookup table” (LUT) que suele ser una “sin/cos ROM” donde se realizan las operaciones de conversión de fase en amplitud.
4. El convertor digital-analógico (DAC) para convertir la onda muestreada, a la salida de la memoria, en una onda escalonada analógica.
5. Finalmente, un filtro paso bajo anti-alias (LPF) que provee un suavizado del escalonado de la senoide que ahora queda lista para ser utilizada.

A continuación, se describen a profundidad cada uno de los bloques que se aprecian en la figura 2.3.

### 2.3.1 El acumulador de fase.

Para sintonizar la frecuencia deseada, a la entrada del acumulador de fase se utiliza una palabra binaria impresa en un registro de entrada al dispositivo DDS que se denomina palabra de control de frecuencia o palabra de sintonía (*FCW*). Se trata de una palabra de  $N$  bits (40 en el caso del circuito integrado AD9850) que determina la resolución del dispositivo DDS. El acumulador de fase es el componente principal del sistema y su resolución puede ir desde los 24 hasta los 64 bits.

La naturaleza de una señal periódica continua, como la senoide, presenta una fase que varía entre  $0$  y  $2\pi$ , fuera de este rango de valores las fases se repiten de forma periódica. En el caso de la realización del DDS el acarreo permite explicar el proceso de creación de una senoide basada en una rueda de fase con valores discretos como se ve en la figura 2.4. (Analog Devices,1999).

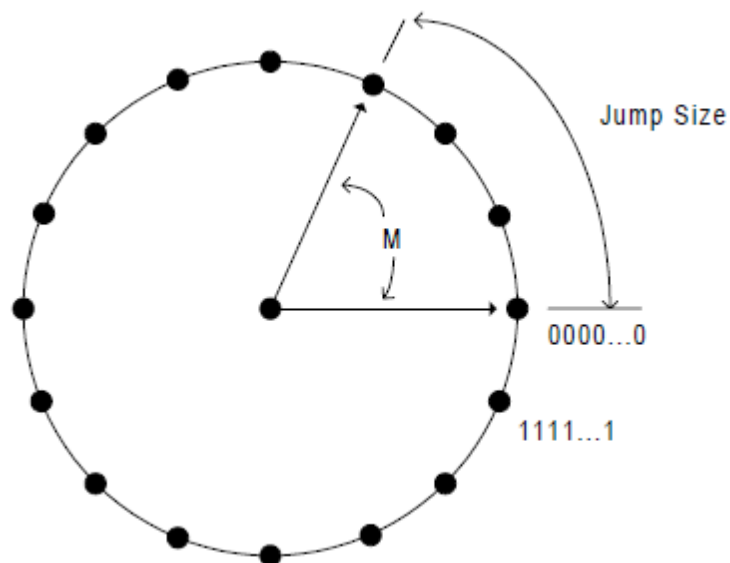


Figura 2.4. Rueda de fase digital.  
Fuente: Analog Devices, 1999

Para entender esta función básica, se debe visualizar la onda sinusoidal como un fasor alrededor del círculo de fase donde cada punto del círculo se corresponde con un punto de la senoide, en el caso continuo, hay infinitos

puntos y una rotación del fasor a una velocidad constante, equivale a un ciclo completo de la senoide generada a la salida. El número de puntos de fase discreta contenida en la "rueda" está determinado por la resolución,  $N$ , del acumulador de fase. En el caso discreto que se va a tratar en este trabajo el número de puntos equidistantes sobre el círculo es exactamente  $2^{32}$ , es decir, 4.294'967.296 puntos, correspondiente a los 32 bits de resolución.

El acumulador de fase se utiliza para proporcionar el equivalente de rotación lineal del vector alrededor de la rueda de fase. El contenido del acumulador de fase corresponde a los puntos en el ciclo de la onda sinusoidal de salida. La salida del acumulador de fase es lineal y no se utiliza directamente para generar una onda sinusoidal o cualquier otra forma de onda excepto una rampa. El acumulador de fase es un contador de módulo  $2^n$ , que incrementa su número almacenado cada vez que recibe un impulso de reloj. La magnitud del incremento del acumulador sobre la rueda de fase se determina por la palabra de sintonización que es una palabra codificada de  $N$  bits que se pone a la entrada del DDS. En concreto, esta indica la magnitud del salto de fase que es necesario imponer al acumulador para generar una determinada frecuencia desde el valor mínimo, fijado por la resolución, hasta un valor máximo, dependiendo del sistema implementado. (R.F Guilló, C.M Puliafito, P.G Aliquó, J.L Puliafito, 2012)

Para una resolución de  $2^n$  y una frecuencia de reloj dada, la frecuencia de salida " $f_0$ " estará determinada por la palabra de sintonización, de acuerdo con la ecuación (1.1) de sintonía del DDS:

$$f_0 = \frac{M}{(2^n)} f_{clk} \quad (1.1)$$

Dónde:

$f_0$ : frecuencia de salida.

$M$ : palabra de sintonía.

$f_{clk}$ : frecuencia del reloj interno de referencia.



Los cambios en el valor de M en la arquitectura de los DDS resultan en cambios inmediatos y continuos en la fase de la frecuencia de salida. En la aplicación práctica, el valor M o una palabra de sintonía por frecuencia, se carga en un registro interno de forma serie que precede al registro de fase. Esto se hace generalmente para minimizar el número de terminales del dispositivo DDS a utilizar. Una vez que se desborda el acumulador de fase con la palabra de sintonía comienza un nuevo ciclo de la senoide.

Cabe señalar que al disminuir el número de muestras por ciclo tomadas sobre la rueda de fase la frecuencia de salida aumenta y recíprocamente, al aumentar el número de muestras, por cada revolución del fasor, la frecuencia de salida disminuye. La fase presenta un comportamiento lineal con el tiempo, de manera que, al generar una frecuencia constante, la salida del acumulador de fase es una rampa de valores discretos.

### **2.3.2 El convertidor fase-amplitud.**

El paso siguiente en el desarrollo de la descripción de los DDS es la conversión fase-amplitud que se realiza en la denominada LUT (*Lookup Table*). Se trata en realidad de un “correspondedor” entre la fase y la amplitud de una senoide. Es en este bloque donde cada uno de los valores discretos de fase se le hace corresponder un valor de amplitud de una senoide. (S. Pagel, 2007)

Cabe destacar que si la LUT, una memoria ROM (*Read Only Memory*), tuviera que manejar los  $2^n$  valores (recordar que  $2^{32}$  son 4 294 967 296 puntos) se produciría un aumento exponencial en las dimensiones del circuito integrado, habría que aumentar la superficie para mejorar la disipación o produciría una ralentización, entre otros efectos. Por esta razón la salida del acumulador de fase debe truncarse y, de los N bits que salen del acumulador, la LUT sólo utilizará P bits ( $P < N$ ), los de mayor peso o más significativos, razón por la que sólo habrá una palabra de  $2^P$  entradas en la LUT.

Los valores de los bits truncados dependen del criterio de diseño adoptado por el fabricante del circuito integrado que se vaya a emplear. El truncado de

referencia suele ser una de las causas de la generación de líneas espectrales espurias producidas en el espectro de salida. (S. Pagel, 2007)

Para reducir la carga binaria de la LUT y agilizar su manejo se han propuesto varios métodos. A continuación, se describen dos de ellos:

Método 1: Consiste en desarrollar sólo ángulos de fase correspondientes a un cuarto de ciclo de la senoide muestreada, es decir se almacenan los valores de magnitud entre 0 y  $\pi/2$  y se aprovecha la simetría inherente a la onda sinusoidal para regenerar el resto, esta tarea también se realiza en la LUT. Para reconstruir la onda completa entre 0 y  $2\pi$  un método comúnmente empleado consiste en utilizar los dos primeros bits de mayor peso, o sea, los dos MSB (*More Significant Bit*) del acumulador de fase, el primero de ellos, para fijar el signo, mientras que el siguiente MSB resuelve si la amplitud del seno va a ser creciente o decreciente. En esta aproximación se utilizan sólo  $2^{P-2}$  entradas de la LUT hecho que conduce a una compresión de 4:1.

Método 2: Otro método para ganar espacio en la LUT, utilizado por algunos fabricantes, consiste en aplicar algoritmos de aproximación donde se descompone la fase en sus diferentes componentes aplicando identidades trigonométricas para llegar al resultado. Este método requiere del uso de multiplicaciones que vuelven a ralentizar el proceso. (S. Pagel, 2007)

### **2.3.3 El conversor Digital–Analógico (DAC).**

Para convertir los niveles discretos obtenidos a la salida del acumulador de fase en una senoide, se utiliza el denominado DAC. La estructura de los DAC dependerá del tipo de integración, en el caso del CI AD9850 se utiliza tecnología CMOS (*Complementary Metal Oxide Semiconductor*). En la figura 2.5 se muestra uno de los circuitos clásicos, un DAC de 4 bits. El circuito escalera R-2R está constituido por sólo dos valores de resistencias lo que facilita la integración en poli-silicio minimizándose así el problema de la tolerancia.

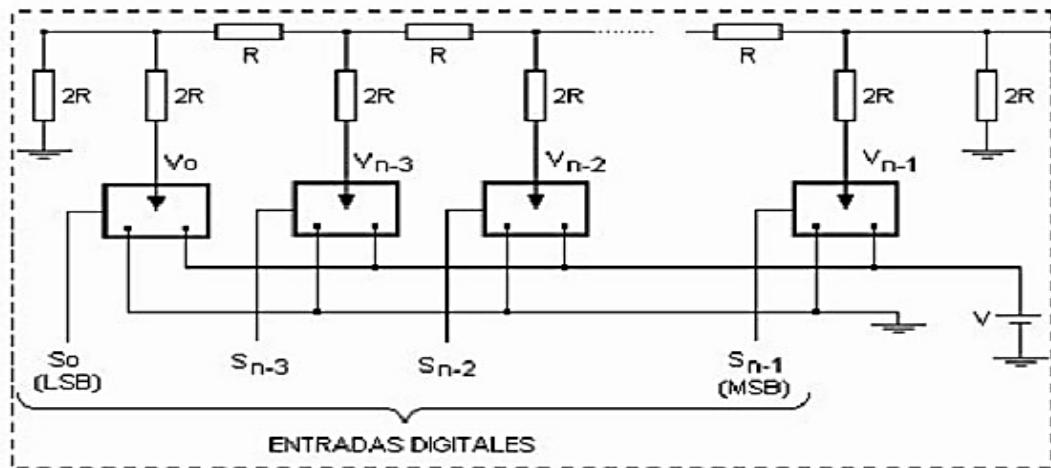


Figura 2.5. Circuito DAC de 4 bits basado en un circuito escalera R-2R.  
Fuente: la autora

Uno de los inconvenientes que se producen debido al uso de DAC es en las transiciones donde aparecen señales anómalas como consecuencia del proceso de conmutación. Se trata de una subida repentina e inesperada de energía (*glitch*), una especie de sobre impulso, que se produce en el instante en que la matriz de conmutación conmuta las fuentes de corriente de un estado a otro. En la figura 2.6 se ha destacado este fenómeno sólo en uno de los pulsos.

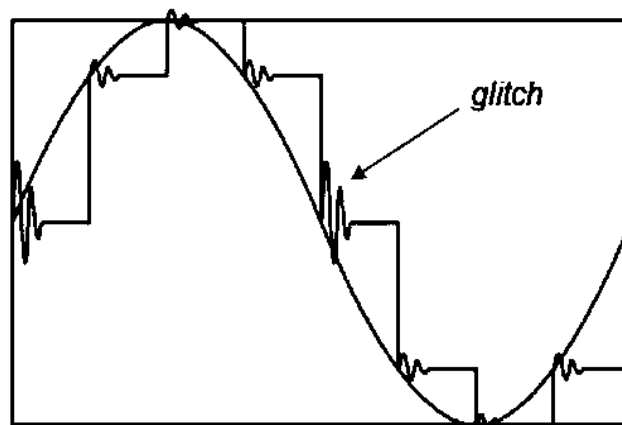


Figura 2.6. Efecto denominado glitch.  
Fuente: la autora

Una posible solución sería realizar la conmutación una vez que el DAC se haya establecido en el nuevo nivel, en la parte plana o valle de la onda escalonada. El DAC es el eslabón más débil de la cadena del DDS en lo referente a ruido y

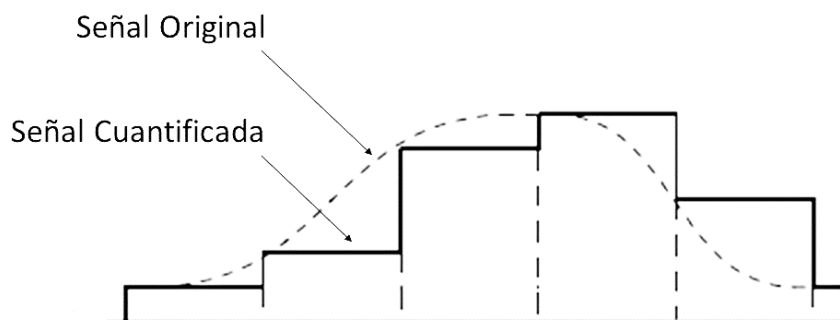
linealidad. Los sintetizadores que emplean esta tecnología casi siempre presentan este problema, lo importante es que estén acotados y su magnitud se encuentre muy por debajo de la señal portadora.

## 2.4 Frecuencias espurias generadas en los DDS.

Una conclusión del análisis funcional del DDS es la presencia de frecuencias imágenes deducibles del criterio de Nyquist. Desafortunadamente no son las únicas frecuencias espurias generadas, ya que hay otros fenómenos que afectan la pureza espectral del sintetizador. A continuación, se describen algunas de las causas de generación de este tipo de señales.

### 2.4.1 Efecto de la resolución del DAC, error de cuantificación.

La resolución de un DAC se especifica por el número de sus bits de entrada. Por ejemplo, la resolución de un DAC de 10 bits de entrada, se dice que tiene "resolución de 10 bits". El impacto de la resolución del DAC es más fácilmente entendido mediante la visualización de la reconstrucción de una onda sinusoidal como se muestra en la figura 2.7.



*Figura 2.7. Ruido de cuantificación.  
Fuente: la autora*

Las líneas verticales son marcadores de tiempo e identifican los instantes en el que la salida del DAC se actualiza a un nuevo valor. Por lo tanto, la distancia horizontal entre las líneas verticales representa el período de la muestra. Se puede observar con absoluta claridad la desviación entre la salida de la señal del DAC y la onda sinusoidal perfecta. La distancia vertical entre las dos trazas en el instante del muestreo es el error introducido por el DAC como resultado

de su resolución finita. Este error es conocido como error de cuantificación y da lugar a un efecto conocido como distorsión de cuantificación.

Para entender la naturaleza de la distorsión de cuantificación, hay que tener en cuenta los cambios abruptos en forma de escalón en la salida del DAC. Estos implican la presencia de componentes de alta frecuencia superpuestos en la frecuencia fundamental. Es en estos componentes de alta frecuencia en donde se presenta la distorsión de cuantificación. En el dominio de la frecuencia, los errores de distorsión de cuantificación son armónicos dentro de la banda de Nyquist y aparecen como señales espurias discretas en el espectro de salida del DAC.

Como la resolución del DAC aumenta, la distorsión de cuantificación disminuye, es decir, el contenido de señales espurias en el espectro de salida del DAC disminuye. Esto tiene sentido, ya que un aumento en la resolución resulta en una disminución en el error de cuantificación. Esto, a su vez, se traduce en menos errores en la reconstrucción de la onda sinusoidal. (Analog Devices,1999)

#### **2.4.2 El efecto del truncamiento en el acumulador de fase.**

El truncamiento de fase es un aspecto importante de la arquitectura de los DDS. Considere un DDS de 32 bits, para convertir directamente esos 32 bits de fase a una amplitud correspondiente sería requerir  $2^{32}$  entradas en la LUT. Serían 4 294 967 296 entradas, si cada entrada se almacenara con 8 bits de precisión, entonces se requeriría de 4 gigabytes de memoria en la tabla de búsqueda por lo que es poco práctico implementar un diseño de este tipo. La solución es utilizar una fracción de los bits más significativos de la salida del acumulador para proporcionar la información de fase. Por ejemplo, en un diseño de DDS de 32 bits, sólo los 12 bits superiores se podrían utilizar para la información de fase. Los 20 bits más bajos serían ignorados (truncados) en este caso.

Para entender las implicaciones de truncar la salida del acumulador de fase es útil utilizar el concepto visto anteriormente de la rueda de fase digital. Considerando una arquitectura sencilla que utilice 8 bits en el acumulador de

fase de los cuales sólo los 5 bits superiores se utilizarán para la información de fase. La representación de la rueda de fase de este modelo en particular, se muestra en la figura 2.8. Con un acumulador de 8 bits, la resolución de fase asociado con el acumulador es de  $(1/256)$  parte de un círculo completo o 1.41 grados  $(360/2^8)$ .

En la figura 2.8, la resolución del acumulador de fase se identifica por los puntos de la circunferencia externa. Si solo se utilizan los 5 bits más significativos del acumulador para transmitir información de fase, entonces la resolución se convierte en la  $(1/32)$  parte del círculo completo o 11,25 grados  $(360/2^5)$  suponiendo que se utiliza un valor de palabra de sintonía de 6, es decir, el acumulador va a contar por incrementos de 6. En la figura 2.8 se muestran los primeros cuatro ángulos de fase correspondientes a los pasos del conteo (6) del acumulador.

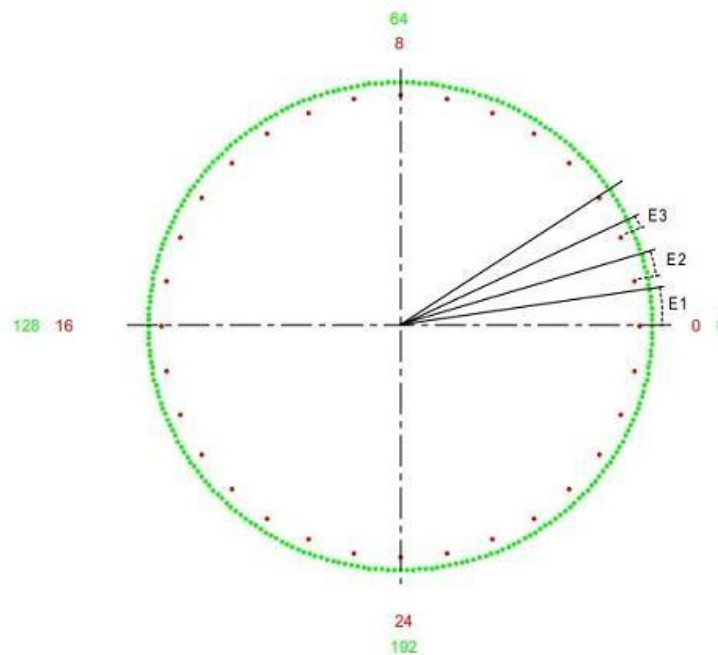


Figura 2.8. Error de truncamiento de fase y la rueda de fase.  
Fuente: Analog Devices, 1999

Es necesario tener en cuenta que la primera etapa de la fase (6 conteos en el círculo exterior) se queda corto del primer punto interior marcado. Así, surge una discrepancia entre la fase del acumulador (el círculo exterior) y la fase determinada por la resolución de 5 bits (el círculo interior). Esta incoherencia

resulta en un error de fase de  $8,46^\circ$  ( $6 \times 1,41^\circ$ ), representada por el arco E1 en la figura. En la segunda etapa de fase del acumulador (6 conteos más en el círculo exterior) la fase del acumulador debe residir entre el primer y segundo punto de la circunferencia interna. Una vez más, existe una discrepancia entre la fase del acumulador y la fase determinada por los 5 bits de resolución.

El error en este caso es de  $5,64^\circ$  ( $4 \times 1,41^\circ$ ) representado por el arco E2 en la figura 2.7. Del mismo modo, en el tercer paso de fase del acumulador el error es de  $2,82^\circ$  ( $2 \times 1,41^\circ$ ). Sin embargo, en la 4ta etapa de fase, la fase del acumulador y la fase de resolución de los 5 bits coinciden resultando en que no existe error de fase. Este patrón continúa cada vez que el acumulador realiza los incrementos de 6 conteos en el círculo exterior.

Los errores de fase introducidos por truncar el acumulador darán lugar a errores en la amplitud durante el proceso de conversión de fase a amplitud inherente en el sistema de los DDS. La diferencia entre el valor ideal de amplitud y el truncado presenta un error variable de forma periódica, el cual es independiente de la palabra de sintonización elegida. Puesto que estos errores de amplitud son periódicos en el dominio del tiempo, aparecen como líneas espectrales en el dominio de la frecuencia, lo que se conoce como frecuencias espurias por truncamiento de fase.

### **2.4.3 No linealidad del DAC.**

Esta es una consecuencia de la incapacidad para diseñar un DAC perfecto. Siempre habrá un error con el nivel de salida esperado asociado al DAC para un código de entrada y el nivel de salida real. Los fabricantes expresan este error como DNL (*Differential NonLinearity*) e INL (*Integral Non-Linearity*).

El resultado principal del DNL e INL es que la relación entre la salida esperada del DAC y su salida real no es perfectamente lineal. Esto significa que una señal de entrada se puede transformar a través de algún proceso no lineal antes de aparecer en la salida. Si una onda sinusoidal digital perfecta se introduce en el DAC, el proceso no lineal causa que la salida contenga la onda sinusoidal, más armónicos de la misma. Así, una onda sinusoidal distorsionada

se producirá en la salida del DAC. Esta forma de error se conoce como distorsión armónica.

El resultado son señales espurias relacionadas armónicamente en el espectro de salida. En ocasiones se utilizan pues pueden ser útiles si se desea generar frecuencias altas, superiores al reloj de los DDS. Generalmente afectan la calidad de las señales moduladas digitalmente en amplitud.

### **Señales espurias a la salida del DAC.**

Muchos diseños de señales mixtas incluyen uno o más circuitos de reloj de alta frecuencia en el circuito integrado. No es raro que aparezcan estas señales de reloj en la salida del DAC por medio de acoplamiento capacitivo o inductivo. Cualquier acoplamiento de una señal de reloj en la salida del DAC se traducirá en una línea espectral a la frecuencia fundamental del espectro de salida de la señal. Otra posibilidad es que la señal del reloj de referencia se encuentre acoplada al reloj de muestreo del DAC.

Esto provoca que la señal de salida del DAC sea modulada por la señal de reloj, teniendo como consecuencia la existencia de señales espurias simétricas alrededor de la frecuencia de la señal de salida. Las técnicas apropiadas de diseño y fabricación son el único seguro contra esta forma de contaminación.

#### **2.4.4 Otras fuentes de ruido.**

La pureza espectral máxima alcanzable de una onda sinusoidal sintetizada es en última instancia relacionada con la pureza del reloj del sistema utilizado para manejar el DDS. Esto es debido al hecho de que en un sistema muestreado se espera que el intervalo de tiempo entre las muestras sea constante. Las limitaciones prácticas, sin embargo, hacen que estos intervalos de muestreo perfectamente uniformes sean imposibles de lograr. Siempre hay cierta variabilidad en el tiempo entre muestras que conducen a desviaciones del intervalo de muestreo deseado. Estas variaciones instantáneas que se traducen en corrimientos en la frecuencia de salida y generación de ruido se denominan *jitter*. Las causas principales del *jitter* son dos:



- Ruido térmico: Denominado en la física como ruido de Johnson, se produce por el movimiento aleatorio de los electrones en los circuitos eléctricos. Cualquier dispositivo posee una resistencia eléctrica que sirve como una fuente de ruido térmico. Puesto que el ruido térmico es movimiento, su espectro de frecuencia es aleatorio e infinito. De hecho, en cualquier ancho de banda dado, la cantidad de potencia de ruido térmico producido por una resistencia dada es constante. Este hecho conduce a una expresión de la tensión de ruido  $V_r$  que está dado por la ecuación (1.2).

$$V_r = \sqrt{4kTRB} \quad (1.2)$$

Donde  $V_r$  es el valor RMS de la tensión de ruido de Johnson,  $k$  es la constante de Boltzmann,  $T$  es la temperatura en grados Kelvin,  $R$  es la resistencia en ohmios y  $B$  es el ancho de banda analizado. La implicación aquí es que independientemente del circuito a utilizar para generar el reloj del sistema, siempre habrá una cierta cantidad finita de saltos temporales debido al ruido térmico. Por lo tanto, el ruido térmico es el factor limitante cuando se trata de minimizar los saltos temporales. (E. M. Lorca, 2010)

- Acoplamiento a fuentes externas: El ruido acoplado puede ser en forma de ruido acoplado localmente causado por la diafonía y/o bucles de tierra dentro o adyacente a la zona inmediata del circuito. También puede ser introducido por fuentes alejadas del circuito. La interferencia que se introduce en el circuito proveniente del medio ambiente circundante se conoce como EMI (interferencia electromagnética). Las fuentes de EMI pueden incluir líneas eléctricas cercanas, radio, transmisores de TV y motores eléctricos, sólo por nombrar unos pocos.

## 2.5 Ventajas y desventajas de los DDS.

Algunas de las ventajas que poseen los DDS son:

- La frecuencia de la señal sinusoidal se sintoniza digitalmente, típicamente con resoluciones de menos de un 1Hz.
- La fase de la onda sinusoidal es digitalmente ajustable, con sólo un ligero aumento en la complejidad del circuito.
- Debido a que el DDS es digital, la frecuencia y la fase son determinadas numéricamente, no hay errores de la señal por el cambio de temperatura o por el envejecimiento de los componentes.
- El salto de frecuencia a frecuencia, o de fase es extremadamente rápido.
- La interfaz del control digital de la arquitectura de los DDS facilita el medio donde este sistema puede ser controlado remotamente y optimizado con una alta resolución bajo el control del procesador o microcontrolador.
- Control preciso de la frecuencia de salida sin afectar la fase.
- Rápidos cambios arbitrarios tanto como en frecuencia, como en fase.
- Modulación precisa.

Algunas desventajas que poseen los DDS son:

- La frecuencia de salida debe ser menor o igual a la mitad de la fuente del reloj de referencia.
- La amplitud de la señal sinusoidal es fija, esto debido a la naturaleza del DAC interno del DDS. Esta amplitud puede ser modificada con circuitos externos adicionales.
- Debido a que la señal sinusoidal es generada digitalmente por técnicas de muestreo, el usuario debe aceptar una pequeña cantidad de distorsión, esto es que la señal no es 100% pura. Esto se resuelve con un filtro pasa bajos de reconstrucción. (A. Torres, 2009)

## 2.6 Fundamentos de la Radio Definida por *Software*.

Radio Definida por *Software* (SDR por sus siglas en inglés) es una técnica de radio muy utilizada actualmente tanto en el ámbito comercial como en el

amateur en la que, varios de los circuitos que componen un receptor o transmisor de radio convencional son implementados mediante *software*. Son equipos de radio en los que la parte *hardware* (circuitería) es mínima, y la mayor parte de las funciones se ejecutan a través de programas en una computadora.

### **2.6.1 Origen y desarrollo.**

La concepción basada en *software* de los SDR minimiza la necesidad de realizar modificaciones de *hardware* durante actualizaciones tecnológicas. Para mantener criterios de interoperabilidad, el sistema de radio debe estar construido preferentemente sobre una arquitectura bien definida, estandarizada y abierta. Contar con estas bases garantiza la escalabilidad del proyecto, así como permite que los componentes de un sistema se desarrollen independientemente, a la vez que se garantiza la interconexión posterior de las partes.

Uno de los primeros radios definidos por *software* fue el sistema *Integrated Communications Navigation and Identification Avionics (ICNIA)* de la fuerza aérea de los Estados Unidos, desarrollado a finales de la década de 1970. El sistema utilizaba un modem basado en un DSP, que era reprogramable para diferentes plataformas. Basado en esa tecnología se desarrollaron otros radios para uso militar. A finales de la década de 1980, GEC desarrolló el primer Radio Programable Digital (PDR, por sus siglas en inglés) descrito en la literatura. El radio estaba compuesto por dos extremos, uno “negro” (encriptado) y uno “rojo” que incluía una CPU, un procesador de entrada/salida y la fuente de energía. Poco después, *ITT Corporation* desarrolló un radio muy similar al PDR, con elementos de alto nivel, como procesadores Intel 486 y FPGAs.

Desde el punto de vista de arquitectura SDR, uno de los primeros ejemplos fue el sistema *SPEAK easy*, concebido por la Fuerza Aérea de los Estados Unidos, y convertido eventualmente en un esfuerzo conjunto de varias ramas del ejército norteamericano. A principios de los años 1990, la Fase I de *SPEAK easy* buscaba crear formas de onda de empleo militar como prueba de

concepto, que incluían un sistema de alta velocidad y cuatro canales, con técnicas de espectro ensanchado. El objetivo era crear un modem donde la mayoría del procesamiento de señales ocurriese en un DSP, el cual fue cumplido. La Fase II llevó a una arquitectura de sistema modular, abierta, reprogramable basada esta vez en componentes comerciales en la segunda mitad de los 1990. El sistema estaba limitado al intervalo de frecuencias de 4 a 400 MHz, aunque de acuerdo a los objetivos iniciales de diseño se pretendía que cubriera de 2 MHz a 2 GHz y no tenía soporte a formas de ondas de uso civil. (T. Walter, 2002)

A finales de la década de los 1990 existían en el Departamento de Defensa de los Estados Unidos más de 200 proyectos relacionados con tecnologías de radiocomunicaciones. La SCA (*Software Communications Architecture*) surge como producto del esfuerzo de estandarización de varios actores del Departamento de Defensa de los Estados Unidos (DoD) responsabilizados con los proyectos de *Joint Tactical Radio System* (JTRS), encabezados por la *Joint Program Executive Office* (JPEO).

El desarrollo de la tecnología SDR ha generado un gran campo de trabajo y desarrollo en las telecomunicaciones en el ámbito mundial. La explotación de esta tecnología en las diferentes ramas de las telecomunicaciones ha devenido en un gran desafío para todas las ramas del saber implicadas en la implementación de sistemas de este tipo. Existen consideraciones de diseño que deben tomarse en cuenta en el desarrollo de cada una de las etapas que componen los SDR, motivado por las tecnologías empleadas en la implementación de cada una de ellas. En el caso particular del *software* para SDR, el desconocimiento y la complejidad de las soluciones más flexibles y genéricas hacen que se sigan produciendo sistemas rígidos e incompatibles.

## **2.7 Definiciones de Radio Definida por Software.**

Joseph Motola fue quien acuñó en 1991, el término SDR. Lo hizo refiriéndose en aquel momento, a una clase de radio reprogramable y reconfigurable. Sin embargo, la definición exacta de Radio Definida por *Software* es controversial.

A continuación, se muestran algunas de las definiciones más difundidas:

- Equipo de radio que define mediante *software* aspectos tales como modulación/demodulación, corrección de errores, procesos de encriptación, que además controle el *hardware* encargado de la etapa de radio frecuencia y permite ser reprogramado. (T. Walter, 2002)
- Un sistema de radio cuya funcionalidad se define sustancialmente en *software*, pudiendo ser alterado el comportamiento de su capa física mediante cambios en ese *software*. (J. Reed, 2002)
- Equipo de radio que, mediante la programación de su *software*, permite que una misma pieza de *hardware*, realice diferentes funciones en diferentes momentos. (T. Walter, 2002)
- Según el SDR Fórum son elementos de los circuitos inalámbricos cuyos parámetros y modos de operación pueden ser modificados vía *software*.
- Según la Comisión Federal de Comunicaciones (FCC, *Federal Communications Commission*) es un sistema transmisor en el cual los parámetros operativos relativos al rango de frecuencia, tipo de modulación o máxima potencia de salida pueden ser alterados mediante la manipulación del *software*, sin realizar ningún cambio en los componentes del *hardware* que afecten las emisiones de radio frecuencia.
- Citando a Eric Blossom, fundador del proyecto GNU Radio, SDR puede definirse así: “*Software* Radio es la técnica de acercar el código a la antena tanto como sea posible. SDR convierte los problemas de *hardware* de radio en problemas de *software*”. (E. Blossom, 2004)
- Es aquella radio que tiene casi todos sus "componentes" definidos y funcionando en forma de programas en una computadora, a excepción de un mínimo de componentes físicos necesarios, externos al mismo, que no pueden ser definidos por *software*. Modificando o reemplazando

ese *software*, se consigue modificar sus funcionalidades, añadiendo o mejorando sus prestaciones. (B. Fette, 2006)

## 2.8 Arquitectura de un sistema de Radio Definida por Software.

El concepto SDR ha ido evolucionando con los años, pero los avances conseguidos han partido esencialmente de la misma configuración básica que se muestra en la figura 2.9. Se compone de tres bloques funcionales: sección de RF, sección de IF (*Intermediate Frequency*) y sección Banda Base, de las cuales RF (*Radio Frequency*) casi siempre trabaja con *hardware* analógico mientras que las secciones de IF y Banda Base se implementan con módulos de *hardware* digitales. ([roboticslab@uc3m.es](mailto:roboticslab@uc3m.es))

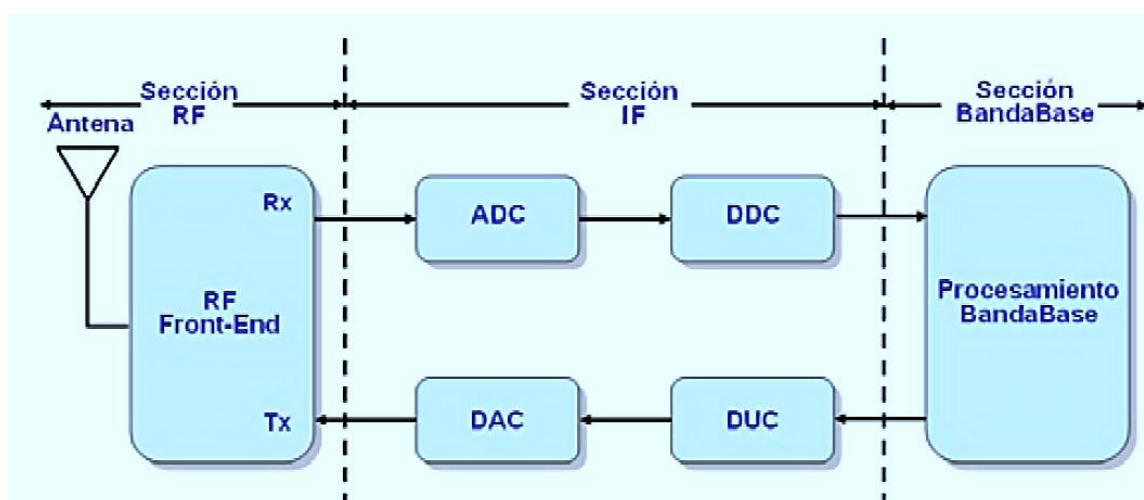


Figura 2.9. Diagrama de bloques funcionales de SDR.

Fuente: [roboticslab@uc3m.es](mailto:roboticslab@uc3m.es)

La sección de IF es la encargada de pasar la señal de IF a banda base y digitalizarla (en el caso de la recepción) o pasar la señal de banda base a IF y hacer la conversión digital-analógica de la señal (en el caso de la transmisión). Los encargados de la conversión analógica-digital o digital-analógica de la señal son los módulos ADC/DAC (*Analog to Digital Conversion / Digital to Analog Conversion*). Los módulos DDC/DUC (*Digital Down Conversion / Digital*

*Up Conversion*) son los encargados de bajar digitalmente la señal de IF a Banda Base o subir de banda base a IF respectivamente.

La sección de Banda Base es la encargada de todo el procesamiento en banda base de la señal (como *frequency hopping*, establecimiento de sesión, ecualización, manejo de tiempos de bit, entre otros) y en algunos casos de la implementación de protocolos del nivel de enlace del modelo OSI (*Open Standard Interconnection*).

En esta configuración de SDR los módulos DDC/DUC y la sección de banda base son los que más MIPS (*Million of Instructions Per Second*) necesitan, motivo por el cual normalmente son implementados en dispositivos de propósito general ([roboticslab@uc3m.es](mailto:roboticslab@uc3m.es)).

## CAPITULO 3 . Descripción de la implementación.

Una vez analizados los fundamentos teóricos en que se sustentan los DDS se tienen los elementos necesarios para realizar la propuesta de diseño del *hardware* para este proyecto.

Se requiere que el diseño garantice las siguientes especificaciones:

- Generación de señales en un amplio rango de frecuencias.
- Rapidez y precisión en la generación de frecuencias.
- Buena estabilidad y bajo nivel de ruido de la señal de salida del dispositivo.
- Configuración reprogramable.

En la figura 3.1 se muestra el diagrama en bloques del módulo DDS completo que se propone para cumplir lo antes planteado.

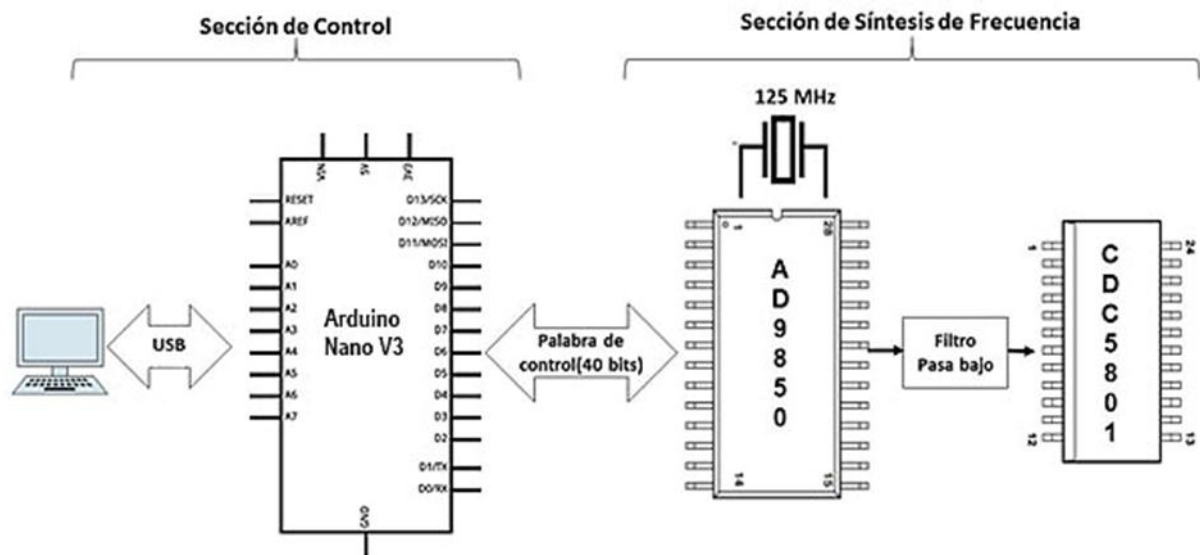


Figura 3.1. Diagrama en bloques del módulo DDS completo.

Fuente: la autora

La sección de síntesis de frecuencia está compuesta por 4 partes principales: el sintetizador de frecuencias integrado AD9850, el reloj de referencia, el filtro pasa bajo o filtro suavizador y el multiplicador de frecuencias CDC5801.



La sección de control tiene como componente fundamental el Arduino Nano V3 que maneja al AD9850, garantiza la alimentación del módulo completo y permite la comunicación vía USB con una aplicación de escritorio en la computadora personal con el objetivo de que el usuario realice los cambios de frecuencia que estime convenientes.

A continuación, se describen cada una de las partes analizadas con anterioridad y se expone la configuración propuesta para el diseño del presente trabajo.

### **3.1 Generalidades del circuito integrado AD9850.**

El AD9850 es un dispositivo de alta escala de integración que utiliza las tecnologías más avanzadas en la síntesis de frecuencias acopladas a un comparador y un convertor digital-analógico (DAC) de alto rendimiento y velocidad para formar un completo sintetizador de frecuencias digitalmente programable (véase figura 3.2). Este dispositivo, usando una fuente de reloj precisa, genera a la salida una onda sinusoidal analógica programable en frecuencia y fase, con una excelente pureza espectral.

La señal digital conformada es convertida a analógica mediante un convertor digital-analógico interno de 10 bits de resolución. Un comparador de alta velocidad permite la conversión de la señal sinusoidal a una señal cuadrada con bajo nivel de ruido para aplicaciones de reloj de alta velocidad. Emplea tecnología avanzada CMOS para proveer gran nivel de funcionalidad y rendimiento con sólo 380mW de disipación de potencia (5V de alimentación) ([www.analog.com](http://www.analog.com)).

El núcleo DDS rápido e innovador del AD9850 utiliza una palabra de 32 bits para la sintonización de la frecuencia, lo que permite una resolución de sintonía de 0,0291 Hz con una frecuencia de reloj de referencia de 125 MHz. Su arquitectura permite generar frecuencias hasta la mitad de la frecuencia de reloj que se utilice, es decir, hasta 62,5 MHz si el reloj es de 125 MHz y la frecuencia de salida puede ser cambiada digitalmente (asincrónicamente) en un rango de hasta 23 millones de frecuencias nuevas cada segundo. El dispositivo también provee de 5 bits de

modulación de fase controlados digitalmente, lo cual permite saltos de fase en la salida que pueden incrementarse en 180°, 90°, 45°, 22.5°, 11.25° o cualquier combinación de estos.

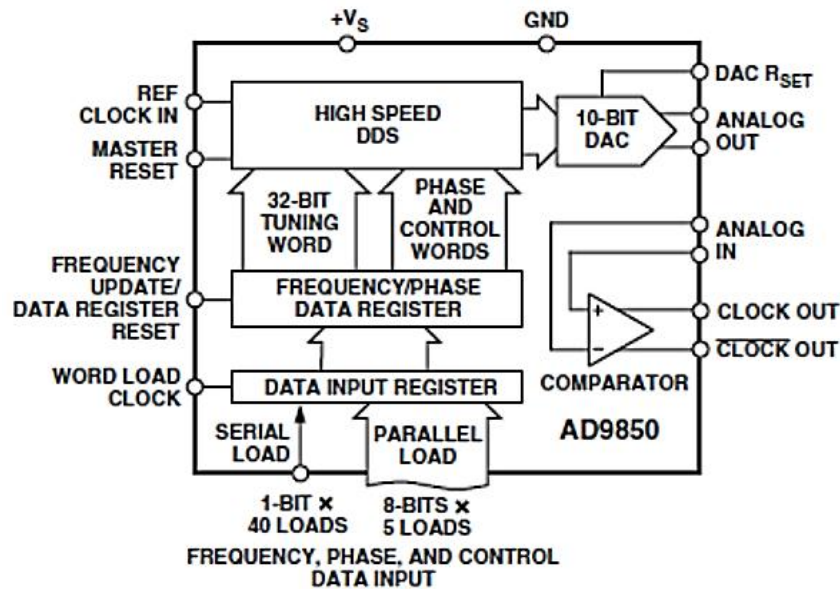


Figura 3.2. Diagrama en bloques funcional del AD9850

Fuente: [www.analog.com](http://www.analog.com)

Las palabras de control, de sintonía de frecuencia y modulación de fase pueden ser cargadas al AD9850 de forma paralela o serie. La carga paralela consiste en hacer cinco cargas iterativas de 8 bits (1 byte). El primer byte especifica la modulación de fase, la habilitación del modo *power-down* y el formato de carga. Los bytes del 2 al 5 comprenden los 32 bits para la sintonización de la frecuencia. La carga serie se realiza mediante un flujo de 40 bits de datos aplicados a un solo terminal.

La frecuencia de reloj de referencia debe ser como mínimo de 1MHz. El dispositivo tiene una circuitería interna que funciona como sensor, de tal manera que, una vez que la frecuencia de referencia excede el umbral, el dispositivo entra en el modo *power-down* y permanece en él hasta que la frecuencia de entrada supere nuevamente el valor de 1MHz. Este modo previene el paso de una excesiva corriente a través de los registros dinámicos del dispositivo.

El encapsulado del integrado AD9850 utilizado es del tipo SSOP (*Shrink Small Outline Package*) de montaje superficial de 28 contactos y tamaño ultra miniatura. La figura 3.3 exponen las dimensiones en milímetros de este tipo de encapsulado.

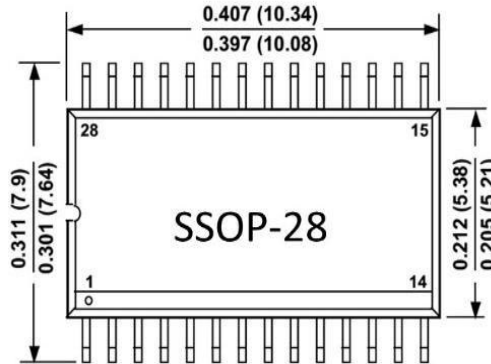


Figura 3.3. Dimensiones y encapsulado del CI AD9850 en milímetros

Fuente: [www.analog.com](http://www.analog.com)

Una precaución a tener en cuenta es que el AD9850 es sensible a las descargas eléctricas producidas por la estática del cuerpo humano.

### 3.1.1 Descripción de terminales del AD9850.

En la figura 3.4 se observa la distribución de terminales del AD9850 y en la tabla 3.1 las descripciones de las funciones de dichos terminales.

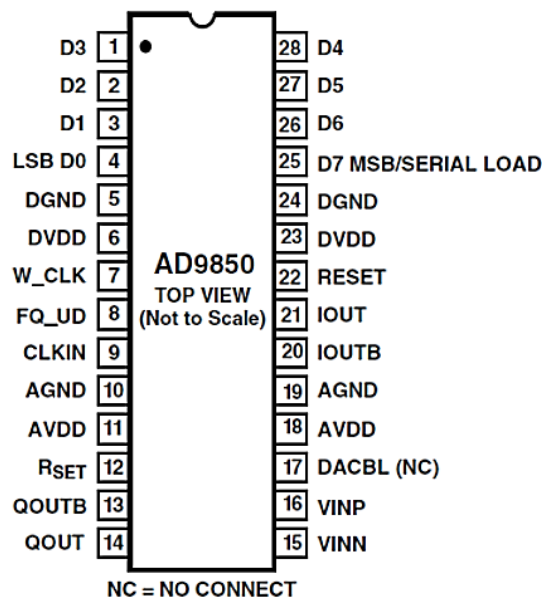


Figura 3.4. Distribución de terminales del AD9850.

Fuente: [www.analog.com](http://www.analog.com)

Tabla 3.1. Descripción de terminales del AD9850.

Número de terminal.	Nombre	Función
De 1 a 4, 25 a 28	D0 a D7	Entrada de datos de 8 bits. Este es el puerto de datos de 8 bits para la lectura de los 32 bits de la palabra de frecuencia y los 8 de la palabra fase y de control. D7 = MSB ( <i>Most Significant Bit</i> ); D0 = LSB ( <i>Least Significant Bit</i> ).
5, 24	DGND	Tierra digital. Esta es la tierra para la sección digital del circuito.
6, 23	DVDD	Fuente de voltaje para el circuito digital.
7	W_CLK	Palabra de carga del reloj. Este reloj es usado para cargar las palabras de fase, frecuencia y control de forma serie o paralelo.
8	FQ_UD	Actualización de frecuencia. Cada vez que ocurre un pulso de reloj, el DDS actualiza la frecuencia (o fase) de los datos cargados en el registro de entrada y entonces resetea el puntero de la palabra a 0.
9	CLKIN	Entrada de reloj de referencia. Esta puede ser un tren de pulsos continuos de nivel CMOS o una señal proveniente de un oscilador.
0, 19	AGND	Tierra analógica. Para la parte analógica del circuito (DAC y el comparador).
11, 18	AVDD	Fuente de voltaje para el circuito analógico (DAC y el comparador).

12	Rset	Conexión externa de la Rset del DAC. Este valor de resistor pone la corriente de salida del DAC en escala completa. Para aplicaciones normales, su valor es de 3,9 K $\Omega$ conectada a tierra.
13	QOUTB	Salida complementaria. Esta es la salida complementaria del comparador.
14	QOUT	Salida verdadera. Esta es la salida verdadera del comparador.
15	VINN	Entrada de voltaje inversora. Esta es la entrada inversora del comparador.
16	VINP	Entrada de voltaje no inversora. Esta es la entrada no inversora del comparador.
17	DACBL (NC)	DAC <i>Baseline</i> , normalmente no se conecta para tener un rendimiento óptimo.
20	IOUTB	Salida analógica complementaria del DAC.
21	IOUT	Salida analógica verdadera del DAC.
22	RESET	Esta es la función de <i>reset</i> , cuando está en nivel alto, limpia todos los registros (excepto el registro de entrada)

Fuente: [www.analog.com](http://www.analog.com)

### 3.2 Aplicación del AD9850.

En la figura 3.5 se muestra el diagrama en bloques funcional del AD9850 funcionando como generador de señal de reloj, función para la cual va a ser utilizado en este proyecto.

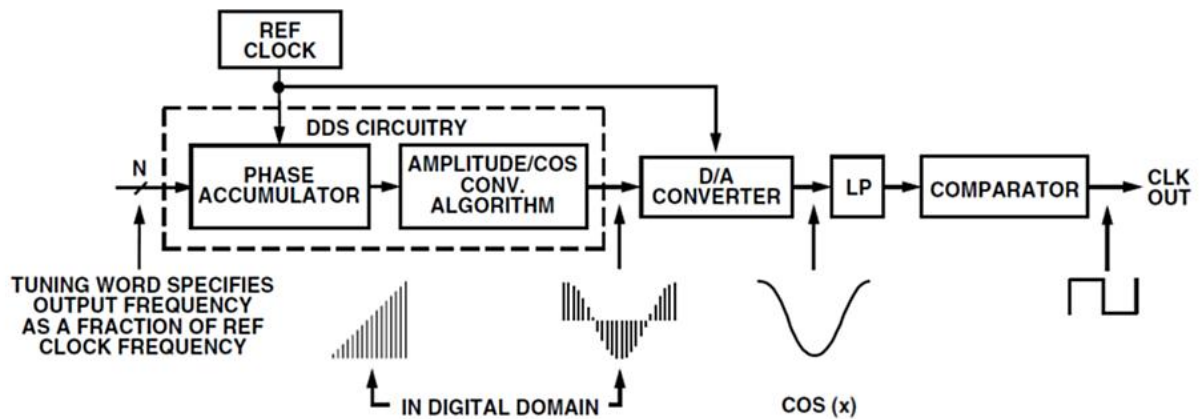


Figura 3.5. Diagrama funcional del AD9850.

Fuente: [www.analog.com](http://www.analog.com)

El sistema DDS o *DDS Circuitry* es básicamente un divisor de frecuencia digital donde el incremento de su resolución viene dado por la frecuencia del reloj de referencia dividida por  $2^N$  el número de bits en la palabra de sintonía. El acumulador de fase es un contador que incrementa el valor almacenado por cada pulso de reloj.

La palabra de sintonía de la frecuencia, fija el valor del contador con lo que se determina el tamaño del incremento de fase que es añadido al valor del acumulador de fase en el siguiente pulso de reloj. Entre mayor el incremento añadido sea, más rápido se desbordará el acumulador, lo que resulta en una frecuencia de salida mayor.

La relación entre la frecuencia de salida, el reloj de referencia y la palabra de sintonía del AD9850 viene dada por la ecuación:

$$f_{OUT} = (\Delta F_{ase} \times CLKIN) / 2^{32}$$

Dónde:

$\Delta F_{ase}$  es el valor de la palabra de sintonía de 32 bits.

CLKIN es la frecuencia de entrada del reloj de referencia en MHz.

$f_{OUT}$  es la frecuencia de la señal de salida en MHz.

La señal sinusoidal digital de salida del bloque DDS es transformada por el conversor Digital/Analógico de alta velocidad de 10 bits para obtener una señal analógica con bajo ruido.

### **3.2.1 Programación del AD9850.**

El AD9850 contiene un registro de 40 bits que se emplea para programar los modos de control. Este registro se carga de forma serie o paralelo de la siguiente manera:

- En el modo paralelo, el registro se carga mediante un bus de 8 bits, la palabra completa de 40 bits requiere de cinco iteraciones de este bus. Las señales *W\_CLK* (*Word Clock*) y *FQ\_UD* (*Frequency Updated*) para direccionar y cargar los registros.
- La carga de la palabra de control en el modo serie se efectúa enviando 40 bits al terminal 25 (D7) del bit de menor peso al bit de mayor peso, después de cada uno de los bits se envía un pulso alto a *W\_CLK* y al final de cada carga se envía un pulso alto a *FQ\_UD* para actualizar la frecuencia de salida del DDS. Cada movimiento de frecuencia implica una nueva carga de 40 bits. Este tipo que carga es la que será empleada en este proyecto.

El AD9850 admite los datos de control a una velocidad muy alta, los pulsos necesitan una duración mínima de tan solo unos pocos nanosegundos, esto permite realizar cambios de frecuencias tan rápidos que resultan inapreciables en tiempo real.

Es preciso aclarar que existen códigos de control específicos utilizados para propósitos de prueba en las fábricas, que pueden dejar temporalmente inoperable el CI AD9850. El usuario debe evitar la entrada de datos en los códigos registrados en la tabla 3.2.

Tabla 3.2. Códigos de control de pruebas reservados internamente.

Formato de carga	Códigos reservados
Paralelo	W0 = XXXXXX10 W0 = XXXXXX01
Serie	W32 = 1; W33 = 0 W32 = 0; W33 = 1 W32 = 1; W33 = 1

(Fuente: [www.analog.com](http://www.analog.com))

### 3.2.2 Configuración externa del AD9850 como generador de reloj.

En la figura 3.6 se propone una configuración externa del AD9850 como generador de señal de reloj para su uso el diseño planteado.

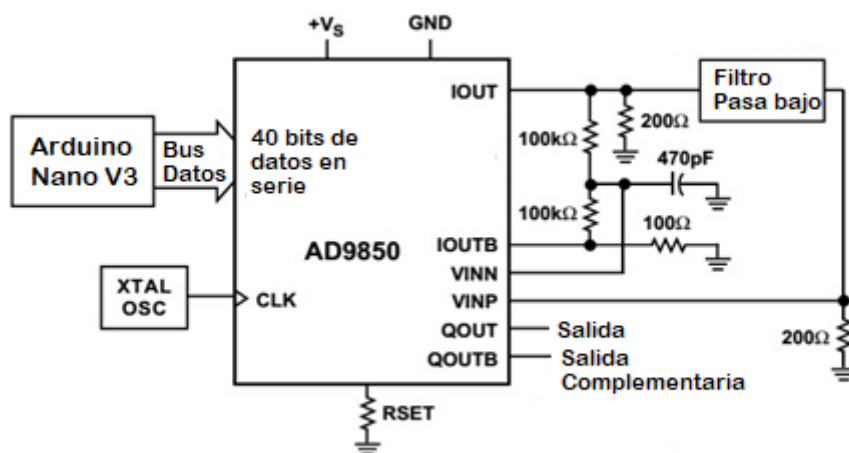


Figura 3.6. Función básica del AD9850 como generador de reloj.

Fuente: la autora

La salida de señal sinusoidal se conecta a dos resistencias de un mismo valor, así se logra un divisor de voltaje que pone la salida de señal cuadrada a la mitad de potencial de este terminal. Se coloca un condensador del terminal 13 a tierra



con el objetivo de estabilizar el voltaje presente en este terminal. El filtro pasa bajo según especifica el fabricante debe estar entre orden 5 y orden 7 generalmente.

### 3.3 Arduino Nano V3.

Arduino Nano V3 es una pequeña y completa placa basada en el ATmega328 (véase figura 3.7). Tiene funcionalidad similar al Arduino UNO, pero con una presentación diferente. No posee conector para alimentación externa, y funciona con un cable USB Mini-B en vez del cable estándar. (<https://www.arduino.cc/en/Guide/ArduinoNano> )

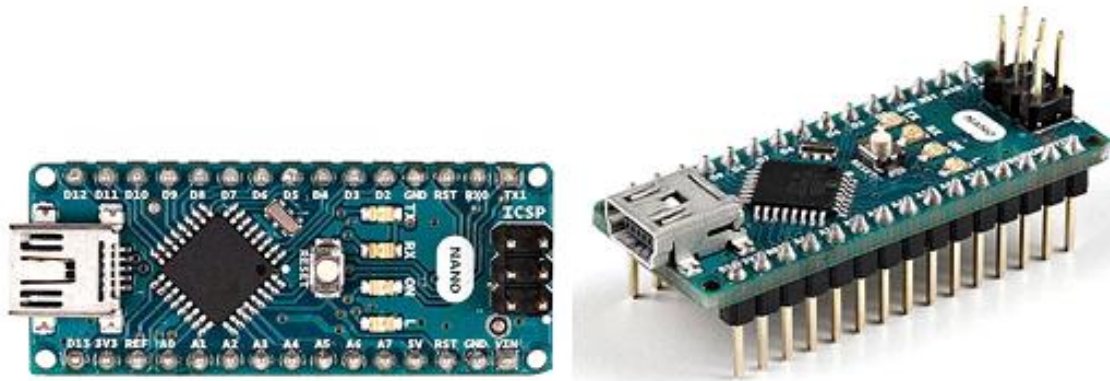


Figura 3.7. Vista superior y lateral del Arduino Nano V3.

Fuente: la autora

#### 3.3.1 Características técnicas.

Microcontrolador: ATmega328.

Tensión de Operación (nivel lógico): 5 V.

Tensión de Entrada (recomendado): 7-12 V.

Tensión de Entrada (límites): 6-20 V.

Terminales E/S Digitales: 14 (de los cuales 6 proveen de salida PWM).

Entradas Analógicas: 8.

Corriente máxima por cada terminal de Entrada/Salida: 40mA.

Memoria Flash: 32 KB (ATmega328) de los cuales 2KB son usados por el *bootloader*.

SRAM: 2 KB (ATmega328).

EEPROM: 1 KB (ATmega328).

Frecuencia de reloj: 16 MHz.

Dimensiones: 18.5mm x 43.2mm

### 3.3.2 Alimentación:

El Arduino Nano posee selección automática de la fuente de alimentación y puede ser alimentado a través de:

- Una conexión Mini-B USB.
- Una fuente de alimentación no regulada de 6-20V (terminal 30).
- Una fuente de alimentación regulada de 5V (terminal 27)

Al alimentar el Arduino a través del Mini USB, se obtiene una salida de 3.3V en el terminal 16 de la placa. Por ende, cuando se conecta a una fuente externa (no USB), los 3.3V no se encuentran disponibles.

### 3.3.3 Entradas y salidas

Cada uno de los 14 terminales digitales del Nano operan a 5V y pueden ser usados como entradas o salidas. Cada terminal puede proveer o recibir un máximo de 40mA y poseen una resistencia de *pull-up* (desconectada por defecto) de 20 a 50kΩ. Algunos terminales poseen funciones especializadas (véase figura):

- Serial: 0 (RX) y 1 (TX). (RX) usado para recibir y (TX) usado para transmitir datos TTL vía serie.
- Interrupciones Externas: terminales 2 y 3. Estos terminales pueden ser configurados para activar una interrupción por flanco de bajada o flanco de subida, o por cambio de nivel.
- PWM: terminales 3, 5, 6, 9, 10, y 11. Proveen de una salida PWM de 8-bits.
- SPI: terminales 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos terminales soportan la comunicación SPI.
- LED: Terminal 13. Existe un LED conectado al terminal digital 13. Cuando el terminal se encuentra en nivel alto, el LED está encendido, cuando el terminal está a nivel bajo, el LED estará apagado.

- El Nano posee 8 entradas analógicas, cada una de ellas provee de 10 bits de resolución (1024 valores diferentes). También, algunos de estos terminales poseen funciones especiales:
  - ❖ I2C: Terminales 4 (SDA) y 5 (SCL). Soporta comunicación I2C (TWI).
  - ❖ AREF. Tensión de referencia por las entradas analógicas.
- Reset. Puesto a nivel bajo se resetea el microcontrolador.

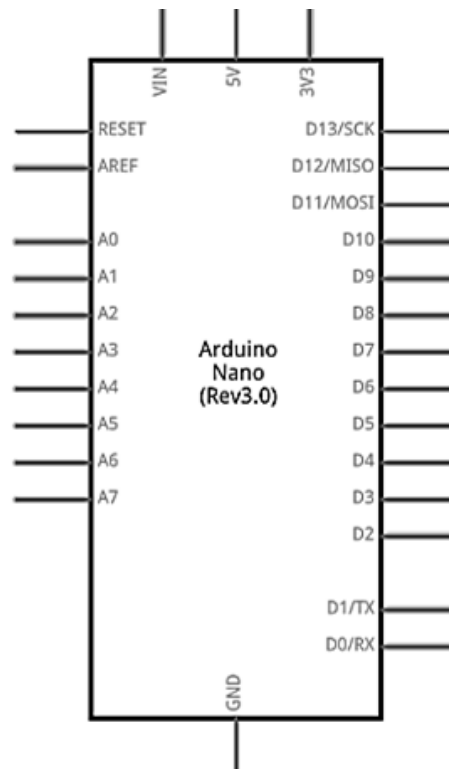


Figura 3.8. Distribución de terminales del Arduino Nano V3.

Fuente: la autora

### 3.3.4 Comunicación

El Arduino Nano tiene algunos métodos para la comunicación con una computadora, otro Arduino, u otros microcontroladores. El ATmega328 posee un módulo UART TTL compatible el cual permite una comunicación serie disponible usando los terminales 0 (RX) y 1 (TX).

El chip FTDI FT232RL en la placa hace de puente a través del USB para la comunicación serial y los controladores FTDI (incluidos con el software de Arduino) proveen a la PC de un puerto COM virtual para el software en la computadora.

El software Arduino incluye un monitor serial que permite visualizar en forma de texto los datos enviados desde y hacia la placa Arduino. Los leds RX y TX en la placa parpadearán cuando los datos se estén enviando a través del chip FTDI y la conexión USB con la computadora.

### **3.3.5 Programación**

El ATmega328 del Arduino Nano viene pre-programado con un bootloader que te permite ingresar código al Arduino sin la necesidad de un programador externo. Se comunica usando el protocolo STK500 original.

También se puede programar el microcontrolador usando un programador ICSP (*In-Circuit Serial Programming*).

### **3.3.6 ¿Por qué emplear Arduino y no otra plataforma de desarrollo?**

Existen diversos microprocesadores o microcontroladores en el mercado que brindan funcionalidades similares a Arduino, pero éste presenta una serie de ventajas que lo hacen ideal para desarrollar un diseño electrónico. Algunas de estas son:

- *Hardware* libre y extensible: El término *hardware* libre se refiere a la posibilidad de acceso público de las especificaciones, diagramas esquemáticos y otras informaciones de cualquier dispositivo, ya sea de forma gratuita o bajo algún tipo de pago. Ofreciendo así la libertad de uso, estudio, modificación y distribución del equipo, además de la redistribución de las versiones modificadas.
- *Software* libre y extensible: De igual forma el entorno de desarrollo, así como las bibliotecas son libres y de código abierto. Cualquier programador puede participar en el desarrollo del mismo.
- Su entorno de programación es multiplataforma: se puede instalar y ejecutar en sistemas Windows, Mac OS X y Linux. Esto no ocurre con el *software* de muchas otras placas.

- Gran popularidad: Día a día la comunidad Arduino crece, y con ella aumentan el número de bibliotecas, ejemplos, tutoriales y recursos disponibles para el mismo.
- Baratas: Las placas Arduino son relativamente baratas frente a otras de otros fabricantes.
- Reutilizables y versátiles: reutilizables porque se puede aprovechar la misma placa para varios proyectos (ya que es muy fácil de desconectarla, reconectarla y reprogramarla), y versátiles porque las placas Arduino proveen varios tipos diferentes de entradas y salidas de datos, los cuales permiten capturar información de sensores y enviar señales a actuadores de múltiples formas.

### **3.4 Multiplicador de frecuencia con circuito integrado CDC5801.**

La multiplicación de frecuencia es una técnica utilizada para el aumento del rango de frecuencias cuando los dispositivos generadores no lo pueden hacer. El CI CDC5801 es un multiplicador/divisor (depende de la configuración) con alineamiento de fase, de moderna generación muy utilizado en la actualidad. A partir de dichas operaciones el circuito es capaz de generar frecuencias en el rango de 12.5MHz a 500MHz con una entrada de reloj de referencia en el rango de 19MHz a 125MHz.

Dependiendo del modo de operación que se escoja, terminales P0, P1 y P2, puede trabajar de 5 modos de acuerdo a la tabla 3.3.

Tabla 3.3. Modos de operación.

Modo	P0	P1	P2	CLKOUT/CLKOUTB
Multiplicador con demora programable y alineamiento de fase	0	0	0	La señal REFCLK multiplicada de acuerdo a la tabla 3.4. Las salidas demoradas o avanzadas de acuerdo a los terminales de configuración DLYCTRL y LEADLAG.
Divisor con demora programable y alineamiento de fase	0	0	1	La señal REFCLK dividida de acuerdo a la tabla 3.5. Las salidas demoradas o avanzadas de acuerdo a los terminales de configuración DLYCTRL y LEADLAG
Solamente en modo multiplicación.	1	0	0	En este modo solo puede multiplicar como dice la tabla 3.4.
Modo de prueba	1	1	0	El PLL y alineamiento de fase quedan obviados.  REFCLK va directo al canal de salida.
Alta impedancia	0	1	x	Alta impedancia.

Fuente: Texas Instruments, CDC5801 Datasheet

El control de la multiplicación y división se seleccionan por los terminales (MULT 0/1 o DIV 0/1) y puede multiplicar por 4, 6 y 8; y puede dividir por 2, 3 y 4, en dependencia de como este configurado. En las tablas 3.4 y 3.5 se observa los valores que se ponen en los terminales de control para fijar los valores de multiplicación o división y el rango de la entrada de referencia (REFCLK) para obtener una frecuencia determinada a la salida. Este dispositivo es caracterizado por operar en un rango de temperaturas de -40°C a 85°C.

Tabla 3.4. Coeficiente de multiplicación.

<b>MULT0</b>	<b>MULT1</b>	<b>REFCLK (MHz)</b>	<b>Rango de Multiplicación</b>	<b>BUSCLK (MHz)</b>
0	0	38 - 125	4	152 - 500
0	1	25 - 83.3	6	150 - 500
1	1	19 - 62.5	8	152 - 500

Fuente: Texas Instruments, CDC5801 Datasheet

Tabla 3.5. Coeficiente de división.

<b>DIV0</b>	<b>DIV1</b>	<b>REFCLK (MHz)</b>	<b>Rango de división</b>	<b>BUSCLK (MHz)</b>
0	0	100 - 125	2	50 – 62.5
1	0	75- 93	3	25 - 31
1	1	50 - 62	4	12.5 – 15.5

Fuente: Texas Instruments, CDC5801 Datasheet

El encapsulado del integrado CDC5801 utilizado es del tipo QSOP (*Small Outline Package*) de montaje superficial de 24 contactos y tamaño ultra miniatura.

### 3.4.1 Descripciones de terminales del CDC5801.

En la figura 3.9 se observa la configuración de terminales del CDC5801 y en la tabla 3.6 se enuncian las descripciones principales de los terminales de este circuito.

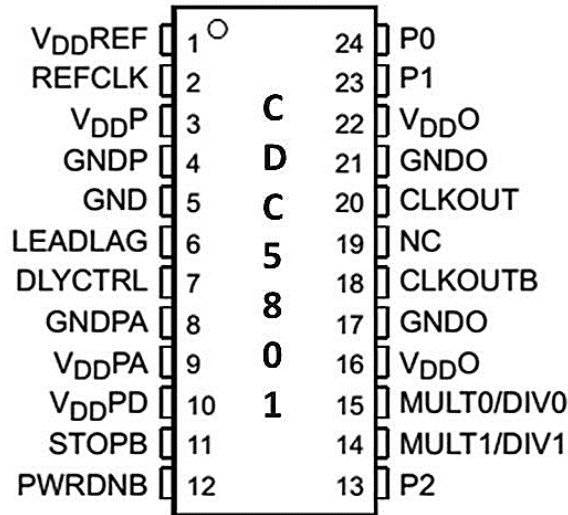


Figura 3.9. Distribución de terminales del CDC5801.  
Fuente: Texas Instruments, CDC5801 Datasheet

Tabla 3.6. Configuración de los terminales del CDC5801.

Nombre del terminal	No.	I/O	Descripción
CLKOUT	20	O	Reloj de salida
CLKOUTB	18	O	Reloj de salida complementario
DLYCTRL	7	I	Cada pulso en este terminal demora o avanza la señal CLKOUT/CLKOUTB
GND	5		GND para VDDREF y VDDPD.
GNDO	17, 21		GND para terminales de reloj de salida (CLKOUT, CLKOUTB).
GNDDP	4		GND para PLL.



GNDPA	8		GND para alineación de fase.
LEADLAG	6	I	Decide si la salida de reloj es demorada o adelantada con respecto a REFCLK.
MULT0/DIV0	15	I	Selección de multiplicación o división.
MULT1/DIV1	14	I	Selección de multiplicación o división.
NC	19	I	No conexión
PWRDNB	12	I	Cuando se pone un nivel bajo, CLKOUT y CLKOUTB van a un nivel bajo
P0	24	I	Modo de control
P1	23	I	Modo de control
P2	13	I	Modo de control
REFCLK	2	I	Entrada de reloj de referencia
STOPB	11	I	Activa en bajo las salidas deshabilitadas
VDDPA	9	I	Alimentación de voltaje para alineación de fase
VDDPD	10	I	Voltaje de referencia para los terminales DLYCTRL, LEADLAG y función STOPB.
VDDREF	1	I	Voltaje de referencia para REFCLK
VDDO	16, 22	I	Voltaje de referencia para los terminales de salida (CLKOUT, CLKOUTB)
VDDP	3	I	Alimentación de voltaje para PLL.

Fuente: Texas Instruments, CDC5801 Datasheet

### 3.4.2 Configuración del CDC5801.

En este proyecto se necesita que el circuito funcione como multiplicador de frecuencias para aumentar el rango de frecuencias del sintetizador hasta 500MHz. Para esto se necesita que opere en el modo multiplicador (**P0 = 1**, **P1 = 0** y **P2 = 0**) y el valor a multiplicar se fija en 8 (**MULT0=1**, **MULT1=1**) para un rango máximo de 500MHz cuando el AD9850 genera su frecuencia máxima a la salida (62.5MHz para un reloj de referencia de 125MHz). Los terminales para definir el modo son controlados por el microcontrolador, de esta manera el usuario determina si desea o no extender el rango. En caso de no requerir la multiplicación se lleva el CDC5801 al modo de prueba (**P0 = 1**, **P1 = 1** y **P2 = 0**) en el cual toda señal a la entrada pasa a la salida sin realizársele procesamiento alguno. El circuito integrado CDC5801 necesita una fuente de alimentación de 3.3V para su funcionamiento. En este proyecto se prevé que el Arduino Nano V3 va a suministrar los 3.3V a través del terminal 16 de la placa.

## CAPITULO 4 . Diseño de *software* de control para el DDS y mediciones físicas.

En este capítulo se expone el diseño del *software* de control del DDS, el cual permite al usuario cambiar fácilmente la frecuencia del dispositivo desde una aplicación de escritorio amigable e intuitiva. Se analiza también la comunicación entre dicho *software* y el Arduino para que este, a su vez, envíe datos de configuración al AD9850 y al CDC5801.



Figura 4.1. Esquema en bloques de la comunicación del sistema.

Fuente: la autora

A continuación, se describe cada una de las partes que intervienen en la comunicación del sistema.

### 4.1 *Software* de control para el cambio de frecuencia del DDS.

El *software* de control es el encargado de brindar una interfaz sencilla y amigable para el usuario, que le permita interactuar con el sintetizador y el multiplicador de frecuencia. Posee un entorno visual, que permite al usuario familiarizarse de forma rápida y realizar la acción deseada exitosamente.

Este programa es el que permite al usuario cambiar la frecuencia generada por el circuito desde la computadora. Para la comunicación entre este y el circuito diseñado se emplea el puerto serie virtual que el driver de la placa Arduino crea sobre el puerto USB.

#### **4.1.1 Entorno de desarrollo de aplicaciones *Microsoft Visual Studio*.**

Para desarrollar la aplicación de control se empleó el *software Microsoft Visual Studio* el cual es un entorno de desarrollo integrado, en inglés *Integrated Development Environment (IDE)* para sistemas operativos *Windows*. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic, F#, Python, etcétera. Permite a los desarrolladores crear aplicaciones, sitios y aplicaciones WEB diseñados para ejecutarse sobre la plataforma .NET. Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas WEB, dispositivos móviles, dispositivos embebidos y consolas. *Visual Studio* es un *software* profesional para el desarrollo de aplicaciones, la versión empleada en este trabajo es la *2017 Community Edition* la cual es gratis para estudiantes universitarios y pequeñas empresas.

El lenguaje de programación escogido es el C# y para esta elección se tuvo en cuenta que este es un lenguaje que brinda facilidades para el desarrollo de aplicaciones con interfaces gráficas.

La aplicación creada se utiliza para comunicar, a través de un puerto serie virtual, una computadora con la placa Arduino Nano V3 (conectada físicamente por USB).

El *software* fue diseñado para que tuviera un aspecto similar al de un equipo real. Es por esto que posee un control rotatorio que permite seleccionar la frecuencia a generarse y un conjunto de teclas de selección que modifican la escala de la misma. En caso de que la frecuencia exceda el valor máximo posible de generación (62 MHz con el multiplicador CDC5801 desactivado y 500 MHz activado) se muestra el valor de la frecuencia en color rojo y se desactiva el botón “cambio de frecuencia”.

Al programa también se le añadió un botón que busca y selecciona el puerto serie al que se conecta el Arduino de forma automática, de esta forma el usuario no tiene la necesidad de seleccionar manualmente este parámetro.

En la figura 4.2 se observa el entorno visual de la aplicación creada, la cual brinda, además de lo antes dicho, el estado de conexión del dispositivo.

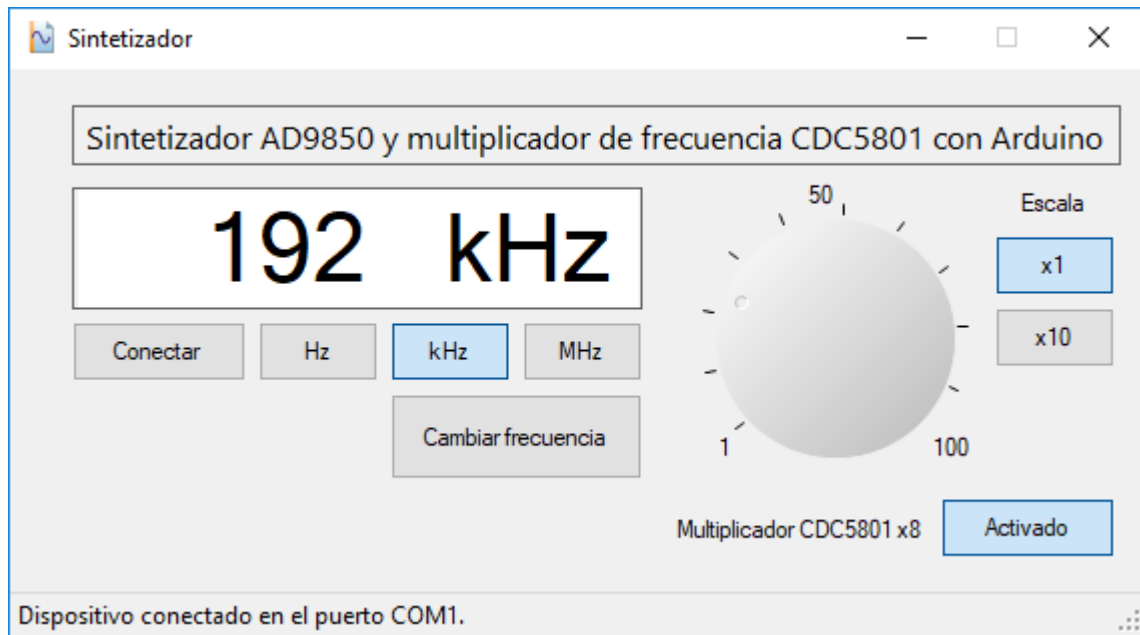


Figura 4.2. Interfaz gráfica de la aplicación de control.

Fuente: la autora

#### 4.1.2 Comunicación entre el Arduino y el *software* de control.

Como ya se explicó anteriormente, la tarjeta Arduino se conecta a la computadora a través de un puerto USB, y el *driver* de la misma permite establecer comunicación empleando un puerto serie virtual, el cual simplifica esta comunicación con respecto al USB pues no es necesario crear un driver específico para el *hardware* que se diseñó.

Aun así, la comunicación a través del puerto serie tiene los inconvenientes de que es necesario predefinir un protocolo de comunicación para asegurar de que todos los comandos enviados en ambas direcciones sean recibidos, y en caso de pérdidas puedan ser retransmitidos. Es por esto que en este trabajo se empleó

*REST*, protocolo ya existente que facilita el intercambio de mensajes entre el *software* de control y la tarjeta Arduino.

El término *REST* se refiere para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc.) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes.

En REST la información se denomina recursos, y a cada uno de estos le corresponde un identificador único dentro de cada servidor denominada URI o identificador de recursos uniforme, y la forma de acceder, ya sea para leer o modificar dichos recursos, es a través de peticiones HTTP, las mismas que se emplean cuando se navega la WEB.

Las peticiones HTTP más usadas en REST se recogen en la tabla 4.1.

*Tabla 4.1. Peticiones HTTP más usadas en REST.*

Petición HTTP	Acción
GET	Devuelve un recurso (leer)
PUT	Crea o actualiza un recurso (crear o modificar)
POST	Actualiza un recurso (modificar)
DELETE	Elimina un recurso

*Fuente: la autora*

Los recursos se pueden representar de múltiples formas cuando se emplea REST, estos pueden estar almacenados en una base de datos o en variables distintas en el servidor, pero si se realiza la petición en un formato soportado por REST este es devuelto de esa forma.

Actualmente el formato más empleado es el denominado JSON, acrónimo de *JavaScript Object Notation*. Es un formato de texto ligero para el intercambio de datos que resulta muy cómodo de trabajar en cualquier lenguaje de programación y es el empleado en este trabajo.

El formato JSON requiere que cada recurso este encerrado entre llaves { }, los campos se encuentran separados por coma, siguiendo la notación “nombre de

campo”: valor, en el caso de que el valor sea un texto también se encierra entre llaves, no siendo necesario para los valores numéricos.

Existen múltiples formas de comunicarse con REST dado que los mensajes del mismo se basan en texto plano, pudiendo emplearse desde un navegador web, hasta una aplicación desarrollada para cualquier sistema operativo o dispositivo móvil.

La plataforma Arduino no implementa de forma nativa la comunicación con REST, pero existen bibliotecas desarrolladas por terceros para este fin, la más destacada es la llamada aREST, disponible en la dirección <http://arest.io>, desarrollada por Marco Schwartz y distribuida con la licencia MIT de carácter abierto y disponible para aplicaciones comerciales sin necesidad de pago de licencias.

Esta biblioteca permite convertir a las placas Arduino en servidores que responden a peticiones REST ya sea a través de la interfaz de red Ethernet, Wifi o del puerto serie simulado en la conexión USB. Esta última es la que se utilizará en este trabajo. Luego de descargar la biblioteca de la página anteriormente referenciada se debe incluir la misma en el código del Arduino de la siguiente forma:

```
// Bibliotecas  
  
#include <aREST.h>
```

Y luego crear un objeto del tipo aREST, el cual permite implementar el protocolo REST.

```
// Objeto aREST  
  
aREST rest = aREST();
```

De la biblioteca aREST se empleó el recurso que permite mandar a ejecutar funciones directamente en el Arduino, para esto se debe crear la función en el código y luego referenciarla para que pueda ser accedida desde el *software* de control:

*// Función para cambiar la frecuencia accesible por peticiones REST*

```
int changeFrequency(String command) {  
  
    // Get state from command  
    freq = command.toInt();  
    sendFrequency(freq);  
  
    return 1;  
}
```

La función declarada para ser accedida por las peticiones debe devolver una variable del tipo entero y recibir como argumento una cadena de texto o *string* según indica la documentación de la biblioteca aREST. Luego de declarada se debe referenciar por la biblioteca de la siguiente forma:

```
// Funciones accesible por las peticiones REST  
  
rest.function("AD9850",changeFrequency);
```

En este caso el URI de la función es / AD9850, al pasar argumento el identificador completo sería:

**/ AD9850?params=1000** , donde la expresión “**?params=**” señala el argumento a ser pasado en la petición, en este caso el valor 1000. Nótese que en la declaración de la función el argumento es del tipo *string* o cadena de texto.

En el *software* de control basta con enviar a través del puerto serie una cadena de texto que contenga el URI anteriormente enunciado.

También se puede acceder a información descriptiva del programa que corre en el Arduino, para esto el URI es /id, este recurso devuelve la información almacenada previamente por las funciones:

```
// Identificador de la tarjeta  
  
rest.set_id("1");  
rest.set_name("DDS_ad9850");
```



Siendo la primera un código que permite diferenciar las tarjetas Arduino en caso de contar con varias y la segunda variable es el nombre que se desee dar al programa.

#### 4.1.3 Propuesta de comunicación entre el Arduino y el AD9850 y el CDC 5801.

En la tabla se observa la correspondencia entre los terminales de los circuitos integrados y los terminales del Arduino.

*Tabla 4.2. Terminales usados para la comunicación entre Arduino y los CI.*

Terminales AD9850/ CDC5801	Terminales Arduino Nano V3
W_CLK	7
FQ_UD	6
DATA (D7)	5
RESET	4
P0, P1, P2 (Modos CDC5801)	8,9,10

*Fuente: La autora*

Como se observa al AD9850 se le introduce la palabra de control en modo serie a través del terminal 5 del Arduino que está conectado al terminal D7 del AD9850. Por cada bit enviado al AD9850 por el terminal W\_CLK del mismo se da un pulso a nivel alto para cargar el dato. Una vez recibida la palabra de control se da un pulso a nivel alto para actualizar la frecuencia del DDS. Si el usuario por *software* activa la multiplicación por 8 para extender el rango de frecuencias, el Arduino envía el dato de control 100 por los terminales P0, P1 y P2 respectivamente para poner al CDC5801 en modo multiplicador. El coeficiente de multiplicación está fijado en 8 por *hardware*. Cuando no está seleccionado el modo de multiplicación el CDC5801 opera en modo de prueba o modo de puente (toda señal a la entrada pasa a la salida sin realizársele operación alguna).

El código de configuración del Arduino se expone en el Anexo 1 del presente trabajo.

## 4.2 Análisis del desempeño del dispositivo mediante mediciones físicas.

En todo proyecto e ingeniería es necesario realizar pruebas y verificar si los resultados obtenidos corresponden con los teóricos.

La primera medición buscó verificar el buen funcionamiento del *hardware* del DDS. El primer paso fue medir la salida del oscilador de referencia del equipo para constatar que se encuentre en 125 MHz y esté alimentada correctamente la entrada de reloj del integrado AD9850. Este proceso se realizó con los 5V que ofrece el Arduino Nano V3. Para ello, utilizando una punta de prueba y el osciloscopio digital HAMEG, se midió la señal en el terminal de salida del oscilador. Como resultado de la medición, se obtuvo en la pantalla del osciloscopio una senoide oscilando a 125 MHz con una amplitud de 976mV pico a pico, como se muestra en la figura 4.3.

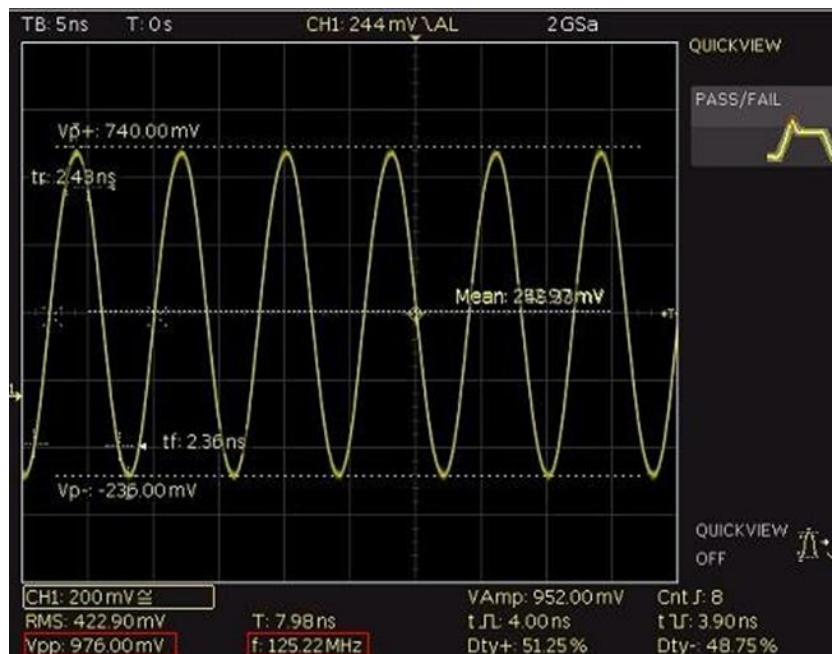


Figura 4.3. Señal de salida del oscilador de referencia.

Fuente: la autora

Luego de garantizar que el reloj de referencia estuviera funcionando correctamente se les realizaron medidas a las señales de salida del DDS y del multiplicador para comprobar el correcto funcionamiento del dispositivo. Las

mediciones se realizaron con el osciloscopio digital HAMEG serie HMO2024. En la figura 4.4 se muestra una señal de 5MHz a la salida del DDS.

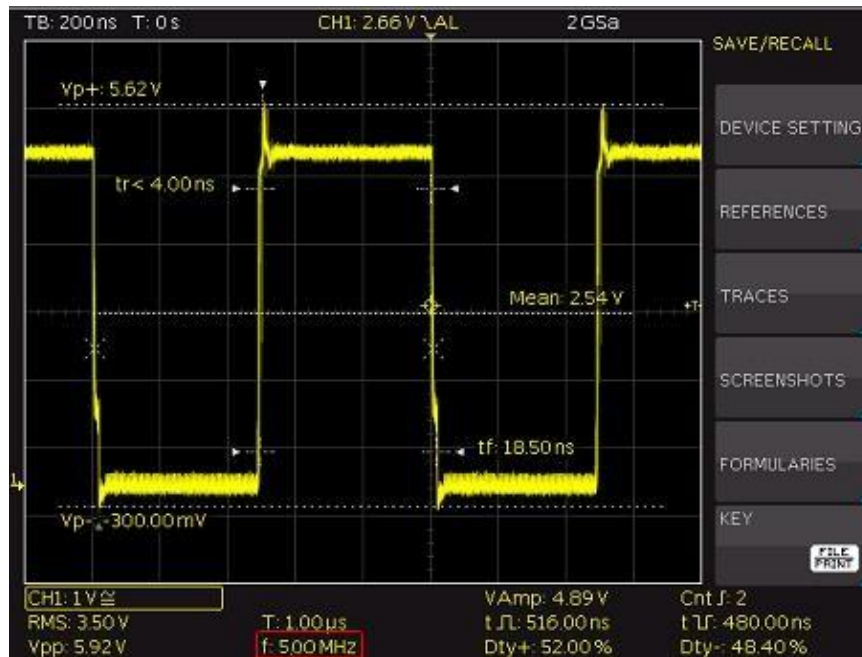


Figura 4.4. Medición de la señal de onda cuadrada de 5 MHz a la salida del DDS.

Fuente: la autora

Si el DDS tiene a su salida una señal de 2.5MHz y se le acopla el multiplicador de frecuencia, con un factor de multiplicación de 8, el resultado que se obtiene a la salida se muestra en la figura 4.5.

En la señal de onda cuadrada a la salida del multiplicador de frecuencia y del DDS se observa que presentan un rizado debido a armónicos de alta frecuencia que requieren de un mejor filtrado. Otro de los inconvenientes presentes en la señal de salida es el *glitch* producido por la conversión de la señal sinusoidal a onda cuadrada por parte del DAC. Este fenómeno es casi imposible de eliminar ya que depende de la lógica interna de conmutación presente en el DAC.

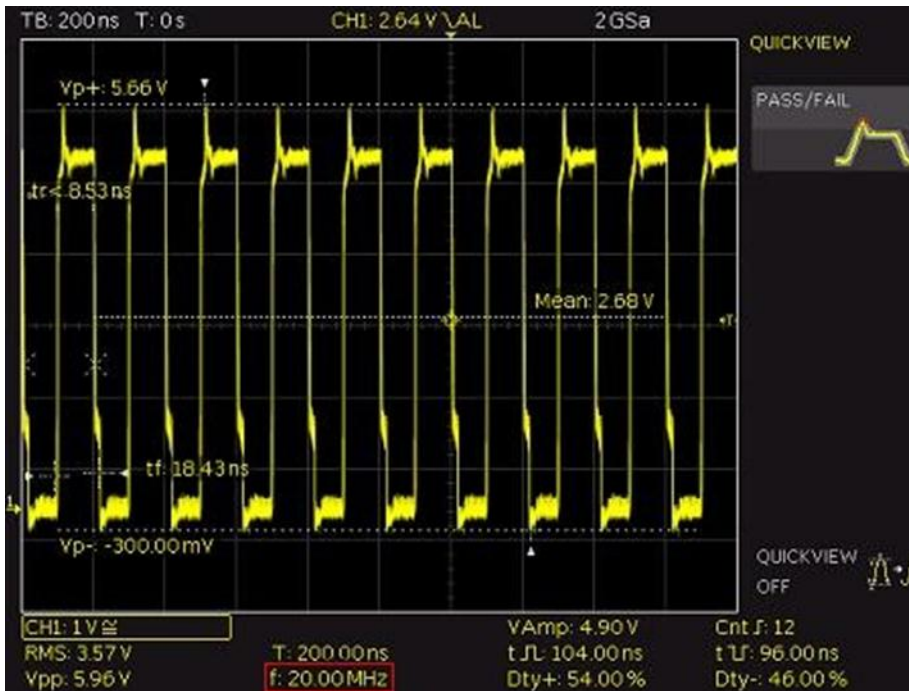


Figura 4.5. Señal de 20 MHz a la salida del multiplicador.

Fuente: la autora

Además de comprobar la exactitud de la señal se efectuaron pruebas para observar el espectro de la señal y analizar la distorsión armónica de la señal de salida. En la figura 4.6 y figura 4.7 se observa el espectro de frecuencia a 5 MHz y 20 MHz respectivamente.

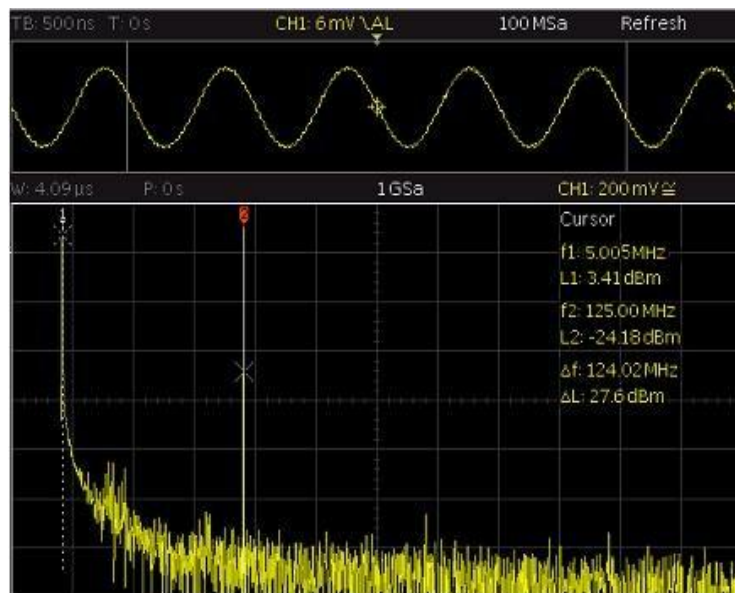


Figura 4.6. Espectro de frecuencias de la señal sinusoidal de 5 MHz.

Fuente: la autora

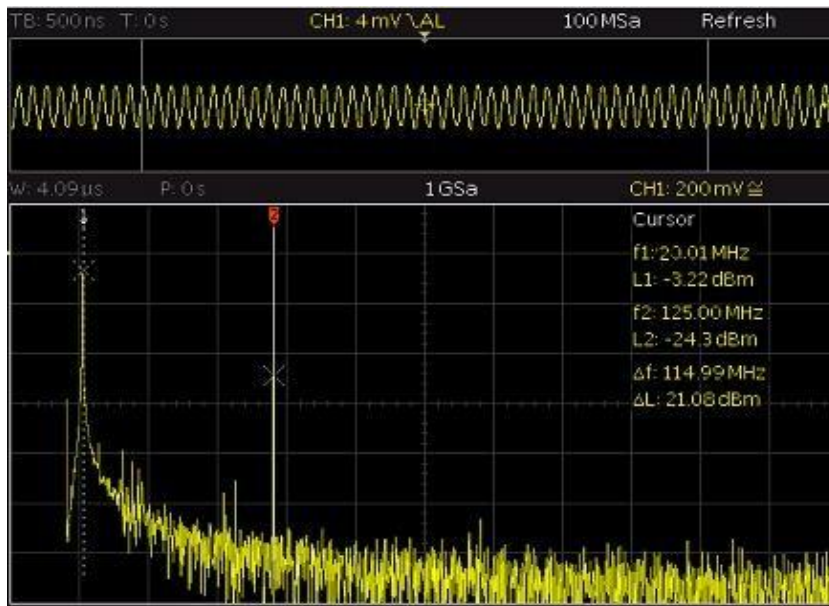


Figura 4.7. Espectro de frecuencias de la señal sinusoidal de 20 MHz.

Fuente: la autora

Ambos resultados muestran la señal portadora acompañada de señales espurias a 125 MHz (reloj de referencia) y múltiplos de éste. Una de las soluciones para lograr una mayor atenuación de la señal de reloj presente en la salida es la construcción de un filtro de mayor orden y el montaje de más capacitores de desacople. Es importante aclarar que esta señal espuria dominante (125 MHz) siempre se encuentra presente en la salida del DDS, por lo que el sintetizador debe coexistir con este problema y lograr atenuar su amplitud lo máximo posible.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- El estudio y caracterización del método de Síntesis Digital Directa de frecuencias permitió, escoger los recursos de *hardware* más adecuados para satisfacer los requerimientos del diseño propuesto.
- Se demostró que la plataforma Arduino puede emplearse satisfactoriamente para el control del dispositivo DDS, dotándolo de *hardware* y *software* de código abierto.
- Se logró un diseño modular, tanto en *software* como en *hardware*, que de acuerdo a la selección que realice el usuario, permite modificar la frecuencia de la señal que genera el dispositivo hasta un valor máximo de 500MHz.
- El diseño de una aplicación de escritorio, en lenguaje de alto nivel, para este dispositivo, le facilitó al usuario el control del sistema.
- Las mediciones físicas realizadas en el laboratorio permitieron comprobar el desempeño correcto del dispositivo.
- Se puede concluir que en este proyecto se ha diseñado un sintetizador digital directo de frecuencias que cumple con todos los requerimientos técnicos para su uso como oscilador local en un radioreceptor definido por *software*.

## **Recomendaciones**

- Realizar la implementación física del diseño completo, realizando para ello el diseño del circuito impreso correspondiente.
- Continuar efectuando mediciones físicas una vez implementado todo el circuito para evaluar su comportamiento de manera prolongada.

## REFERENCIAS BIBLIOGRÁFICAS

- W. Tomasi, Sistema de Comunicaciones Electrónicas. Cuarta Edición. Pearson Educación, 2003.
- R.F Guilló, C.M Puliafito, P.G Aliquó, J.L Puliafito, Implementación de un sintetizador de banda ancha. Proyecciones, Publicación de la Facultad Regional de Buenos Aires. Vol. X, no. 2., octubre 2012.
- S. Pagel, "Síntesis digital Directa de Frecuencias, DDFS", Revista Española de Electrónica, pp. 66-76, marzo 2007.
- Analog Devices, "A Technical Tutorial on Digital Signal Synthesis", 1999.
- E. M. Lorca, "Diseño y Construcción de un Transmisor de Frecuencia Modulada, para Radiodifusión Sonora, con tecnología DDS", Santiago de Chile, 2010.
- A. Torres, Trazador de gráficas de Bode usando síntesis digital directa (DDS), Obregón, 2009.
- E. Blossom, Exploring GNU Radio, 2004. [Online]. Disponible en: <http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html> [Consultado: 4 julio 2016].
- T. Walter, "Software Defined Radio: Origins, Drivers and Perspectives", John Wiley & Sons, 2002.
- J. Reed, "A Modern Approach to Radio Engineering, Prentice Hall Communications Engineering", 2002.
- B. Fette, "Cognitive Radio Technology", Elsevier, 2006.
- "SDR: La alternativa para la evolución inalámbrica a nivel físico". Universidad Carlos III, Madrid. [Online] Disponible en: <https://www.uc3m.es> [Consultado: enero 2017].
- Analog Devices, "125 MHz Complete DDS Synthesizer AD9850". [Online]. Disponible en: [www.analog.com](http://www.analog.com) [Consultado: agosto 2017].



- Página Oficial de Arduino, disponible en: <https://www.arduino.cc/en/Guide/ArduinoNano> [Online]. [Consultado marzo 2017]
- Página Oficial del protocolo aREST, disponible en: <http://arest.io> [Online]. [Consultado septiembre 2017]

## GLOSARIO DE TÉRMINOS

<b>Términos</b>	<b>Significado</b>
BNC	Bayonet Neill-Concellman
CI	Circuito Integrado
CMOS	Complementary Metal Oxide Semiconductor
DAC	Digital Analog Converter
DDS	Direct Digital Synthesizer
DNL	Differential NonLinearity
DSP	Digital Signal Processor
FQ_UD	Frequency Updated
HF	High Frequency
INL	Integral Non-Linearity
LUT	Lookup Table
MSB	More Significant Bit
PC	Personal Computer
PLL	Phase Locked Loop

USB	Universal Serial Bus
-----	----------------------

## ANEXOS

### Anexo I: Programación del Arduino Nano V3.

```
/*
  Control del sintetizador AD9850
*/

// Terminales de conexión con el AD9850
#define W_CLK 7
#define FQ_UD 6
#define DATA 5 // Terminal D7 del AD9850
#define RESET 4

// Terminales del CDC5801
#define P0 8
#define P1 9
#define P2 10

// Bibliotecas
#include <aREST.h>

// Objeto aREST
aREST rest = aREST();

// Frecuencia del módulo AD9850
#define AD9850_CLOCK 125000000 // Cristal de 125 MHz

// Variables
long unsigned int freq = 1000; // Frecuencia inicial

// Función que genera un pulso de nivel alto en un terminal determinado
#define pulseHigh(pin) {digitalWrite(pin, HIGH); digitalWrite(pin, LOW); }
```

// Esta función transmite un byte, bit a bit, empezando por el LSB al AD9850 de forma serie

```
void tfr_byte(byte data) {
    for (int i = 0; i < 8; i++, data >>= 1) {
        digitalWrite(DATA, data & 0x01);
        pulseHigh(W_CLK); //después de cada bit, se manda un pulso alto por
W_CLK
    }
}
```

// Esta función cambia la frecuencia generada por el AD9850

```
void sendFrequency(double frequency) {
    int32_t freq1 = frequency * 4294967295/AD9850_CLOCK; // Empleando un
cristal de 125 Mhz
    for (int b = 0; b < 4; b++, freq1 >>= 8) {
        tfr_byte(freq1 & 0xFF);
    }
    tfr_byte(0x000); // Palabra final de control
    pulseHigh(FQ_UD);
}
```

// Esta función cambia el modo del CDC5801

```
void sendMode(int mode)
{
    switch (mode)
    {
        case 0: // Multiplicador solo
            digitalWrite(P0,HIGH);
            digitalWrite(P1,LOW);
            digitalWrite(P2,LOW);
            break;
        case 1: // Test mode (Salida igual a entrada)
            digitalWrite(P0,HIGH);
```

```

    digitalWrite(P1,HIGH);
    digitalWrite(P2,LOW);
    break;
}
}

void setup(void)
{
    // Se configura el puerto serie
    Serial.begin(115200);

    pinMode(FQ_UD, OUTPUT);
    pinMode(W_CLK, OUTPUT);
    pinMode(DATA, OUTPUT);
    pinMode(RESET, OUTPUT);

    // Funciones accesible por las peticiones REST
    rest.function("AD9850",changeFrequency);
    rest.function("CDC5801",changeMode);

    // Identificador de la tarjeta
    rest.set_id("1");
    rest.set_name("DDS_ad9850");

    // Configuración inicial del AD9850
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // Habilita el modo serie

}

void loop() {

```

```
// Manejo de las peticiones REST a través del puerto serie
rest.handle(Serial);

}

// Función para cambiar el modo del CDC5801
int changeMode(String command) {

    // Get state from command
    int mode = command.toInt();
    sendMode(mode);

    return 1;
}

// Función para cambiar la frecuencia accesible por peticiones REST
int changeFrequency(String command) {

    // Get state from command
    freq = command.toInt();
    sendFrequency(freq);

    return 1;
}
```



**Presidencia  
de la República  
del Ecuador**



**Plan Nacional  
de Ciencia, Tecnología,  
Innovación y Saberes**



**SENESCYT**  
Secretaría Nacional de Educación Superior,  
Ciencia, Tecnología e Innovación

## **DECLARACIÓN Y AUTORIZACIÓN**

Yo, **Rosado Cevallos Maria Isabel**, con C.C: # **1206293506** autor/a del trabajo de titulación: **Diseño de un sintetizador digital directo de frecuencia controlado con Arduino nano para un radioreceptor definido por software**, previo a la obtención del título de **Magíster en Telecomunicaciones** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, a los 10 días de noviembre de 2017

f. \_\_\_\_\_

Nombre: **Rosado Cevallos María Isabel**

C.C: **1206293506**



<b>REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA</b>			
<b>FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN</b>			
<b>TÍTULO Y SUBTÍTULO:</b>	Diseño de un sintetizador digital directo de frecuencia controlado con Arduino nano para un radioreceptor definido por software.		
<b>AUTOR(ES)</b>	Rosado Cevallos Maria Isabel		
<b>REVISOR(ES)/TUTOR</b>	MSc. Edwin Palacios Meléndez; MSc. Celso Bohórquez Escobar / MSc. Manuel Romero Paz		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Sistema de Posgrado		
<b>PROGRAMA:</b>	Maestría en Telecomunicaciones		
<b>TÍTULO OBTENIDO:</b>	Magister en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	<b>Guayaquil, a los 10 días de noviembre de 2017</b>	<b>No. DE PÁGINAS:</b>	<b>80</b>
<b>ÁREAS TEMÁTICAS:</b>	Sintetizadores, Conversores. Acumuladores, Frecuencias espurias, No linealidad		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	DDS, Arduino Nano V3, DAC, DDS, Sintetizadores, Conversor.		
<b>RESUMEN/ABSTRACT:</b>	<p>En este trabajo se presenta el diseño de un sintetizador de frecuencias basado en la tecnología de Síntesis Digital Directa. Con este fin se analizaron aspectos generales de los distintos métodos de síntesis de frecuencia, profundizando en la teoría de Síntesis Digital Directa (DDS por las siglas en inglés de Direct Digital Synthesize). Se realizó la selección de los circuitos integrados más adecuados para satisfacer los requerimientos del diseño y se escogió al Arduino Nano V3 para su empleo en la sección de control. Se diseñó una aplicación en la plataforma Visual Studio para permitir al usuario controlar los cambios de frecuencia del dispositivo. Se realizaron varias mediciones de laboratorio que evidenciaron la precisión y calidad de la señal generada. El trabajo es exploratorio y descriptivo, debido a que pretende explorar las características y evolución de la DDS de frecuencias. Es descriptivo, puesto que permite analizar, diseñar y evaluar las diferentes tecnologías que aplican este método en la generación de señales con frecuencias estables. Es del paradigma Empírico-Analítico con un enfoque Cuantitativo porque utiliza cálculos matemáticos para el diseño electrónico de los diferentes bloques funcionales que conforman el dispositivo para la DDS de frecuencias. Y es Experimental, debido a que se alteran las variables de estudio, para finalmente proceder con el análisis respectivo.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-996914082	<b>E-mail:</b> yzza_1203@hotmail.com	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):::</b>	<b>Nombre:</b> Romero Paz Manuel de Jesús		
	<b>Teléfono:</b> +593-994606932		
	<b>E-mail:</b> manuel.romero@cu.ucsg.edu.ec		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			