COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

BACHELOR'S DEGREE IN INDUSTRIAL ENGINEERING

BACHELOR'S DEGREE FINAL THESIS

DEVELOPMENT OF A DIRECT DIGITAL SYNTHESIS BASED GENERATOR

Author: Teresa Corchado Arozarena

Director: Csaba Szombathy

Collaborating entity: Budapest University of Technology and Economics

Madrid

July 2019

AUTHORIZATION FOR DIGITALIZATION, STORAGE AND DISSEMINATION IN THE NETWORK OF END-OF-DEGREE PROJECTS, MASTER PROJECTS, DISSERTATIONS OR BACHILLERATO REPORTS

1. Declaration of authorship and accreditation thereof.

The author Mr. /Ms. Teresa Corchado Arozarena

HEREBY DECLARES that he/she owns the intellectual property rights regarding the piece of work: Development of a direct digital synthesis based generator
that this is an original piece of work, and that he/she holds the status of author, in the sense granted by the Intellectual Property Law.

2. Subject matter and purpose of this assignment.

With the aim of disseminating the aforementioned piece of work as widely as possible using the University's Institutional Repository the author hereby **GRANTS** Comillas Pontifical University, on a royalty-free and non-exclusive basis, for the maximum legal term and with universal scope, the digitization, archiving, reproduction, distribution and public communication rights, including the right to make it electronically available, as described in the Intellectual Property Law. Transformation rights are assigned solely for the purposes described in a) of the following section.

3. Transfer and access terms

Without prejudice to the ownership of the work, which remains with its author, the transfer of rights covered by this license enables:

a) Transform it in order to adapt it to any technology suitable for sharing it online, as well as including metadata to register the piece of work and include "watermarks" or any other security or protection system.

b) Reproduce it in any digital medium in order to be included on an electronic database, including the right to reproduce and store the work on servers for the purposes of guaranteeing its security, maintaining it and preserving its format.

c) Communicate it, by default, by means of an institutional open archive, which has open and cost-free online access.

d) Any other way of access (restricted, embargoed, closed) shall be explicitly requested and requires that good cause be demonstrated.

e) Assign these pieces of work a Creative Commons license by default.

f) Assign these pieces of work a HANDLE (*persistent* URL). by default.

4. Copyright.

The author, as the owner of a piece of work, has the right to:

a) Have his/her name clearly identified by the University as the author

b) Communicate and publish the work in the version assigned and in other subsequent versions using any medium.

c) Request that the work be withdrawn from the repository for just cause.

d) Receive reliable communication of any claims third parties may make in relation to the work and, in particular, any claims relating to its intellectual property rights.

5. Duties of the author.

The author agrees to:

a) Guarantee that the commitment undertaken by means of this official document does not infringe any third party rights, regardless of whether they relate to industrial or intellectual property or any other type.

b) Guarantee that the content of the work does not infringe any third party honor, privacy or image rights.

c) Take responsibility for all claims and liability, including compensation for any damages, which may be brought against the University by third parties who believe that their rights and interests have been infringed by the assignment.

d) Take responsibility in the event that the institutions are found guilty of a rights infringement regarding the work subject to assignment.

6. Institutional Repository purposes and functioning.

The work shall be made available to the users so that they may use it in a fair and respectful way with regards to the copyright, according to the allowances given in the relevant legislation, and for study or research purposes, or any other legal use. With this aim in mind, the University undertakes the following duties and reserves the following powers:

a) The University shall inform the archive users of the permitted uses; however, it shall not guarantee or take any responsibility for any other subsequent ways the work may be used by users, which are non-compliant with the legislation in force. Any subsequent use, beyond private copying, shall require the source to be cited and authorship to be recognized, as well as the guarantee not to use it to gain commercial profit or carry out any derivative works.

b) The University shall not review the content of the works, which shall at all times fall under the exclusive responsibility of the author and it shall not be obligated to take part in lawsuits on behalf of the author in the event of any infringement of intellectual property rights deriving from storing and archiving the works. The author hereby waives any claim against the University due to any way the users may use the works that is not in keeping with the legislation in force.

c) The University shall adopt the necessary measures to safeguard the work in the future.

d) The University reserves the right to withdraw the work, after notifying the author, in sufficiently justified cases, or in the event of third party claims.

Madrid, on …17….. of …July……………………..., ……….

HEREBY ACCEPTS

Signed………………………………………………

Reasons for requesting the restricted, closed or embargoed access to the work in the Institution's Repository

I, hereby, declare that I am the only author of the project report with title:

Development of a Direct Digital Synthesis based generator

which has been submitted to ICAI School of Engineering of Comillas Pontifical University in the academic year 2018/19. This project is original, has not been submitted before for any other purpose and has not been copied from any other source either fully or partially. All information sources used have been rightly acknowledged.

Fdo.: Date: 17/07/2018

I authorize the submission of this project

PROJECT SUPERVISOR

Fdo.: Csaba Szombathy MSc. Date: 19 July 2019

DEVELOPMENT OF A DIRECT DIGITAL SYNTHESIS BASED GENERATOR

Author: Teresa Corchado Arozarena

Director: Csaba Szombathy

Collaborating Entity: Budapest University of Technology and Economics

1. Introduction

Direct digital synthesis (DDS) is a method of producing, by means of digital processing blocks, an analog waveform from a reference clock frequency. The output signal is typically a sine wave, but it is also possible to generate square and triangle waves.

DDS is the solution for applications that demand precise frequency generation or frequency agility, key requirement to several industries. A DDS device is digitally programmable, which means easy controllability, and have a stable performance. Communication systems including modulation or direct RF transmissions can be highlighted among the wide variety of applications of DDS.

The aim of the present project is to design and implement a sine wave generator intended to be used as a local oscillator for RF transmissions. It will be part of a DRM30 modulator system, that will be broadcasting RF radio signals from the university.

2. Methodology

In order to achieve the objectives of the project, the following methodology has been applied: in-depth research about fundamentals of DDS and its theory of operation, programming the DDS device so it could generate a sine wave at an established frequency and writing a code so the frequency could be set externally. Finally, the evaluation of results is done, and possible improvements are considered.

3. DDS theory of operation

A complete DDS device includes a Numerically Controlled Oscillator (NCO) and a D/A converter into a single chip. The NCO consists on the phase accumulator and the sine lookup table. The system is shown in figure 1.

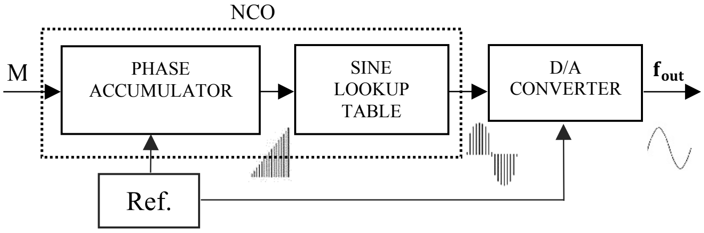


Figure 1: Components of a direct digital synthesizer. Source: Own elaboration.

The input to the phase accumulator is called the tuning word (M). It specifies the desired output frequency in a binary number of 24 to 48 bits. Every clock cycle, M is added to the value contained in the phase accumulator, until it overflows and starts again. The phase accumulator computes the phase values of one cycle of the sinewave, before overflowing [1].

In order to understand the mode of operation of the phase accumulator, the phase wheel showed in figure 2 can be used. The wheel represents a whole cycle of a sine wave from 0º to 360º, being each point of the circle the equivalent phase point of the output wave. The number of points in the wheel is equal to 2^n.

M represents the the jump size the phase accumulator does every clock cycle along the wheel. When the tuning word is big, the phase accumulator register value will take less time in overflowing. This makes that M and the output frequency are directly related. The higher the tuning word, the higher the output frequency and vice versa [1].

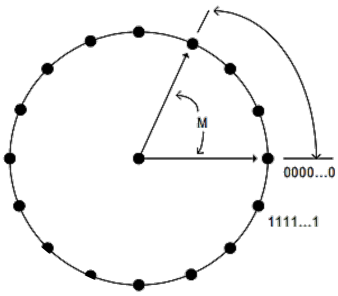


Figure 2: Phase wheel. Source: Analog devices [1]

The sine lookup table contains all the digital amplitude values corresponding to a whole cycle of a sine wave. The output of the phase accumulator serves as an address for the sine lookup table acting as a phase-to-amplitude converter.

Finally, the DAC converts the digital amplitude into the corresponding value of analog voltage or current. [1]

The basic tuning equation of the DDS relates the output frequency as a function of, the tuning word, the clock frequency and the number of bits of the phase accumulator (n).

$$f_{out} = M \times \frac{fclk}{2^n} \quad [1]$$

4. Implementation

The system, showed in figure 3, consists on a DDS device that outputs a sinewave at the desired frequency. The frequency can be set externally with some pushbuttons and it is showed in a display.

The DDS device is an AD9834 type, built-in in a module board. The phase accumulator is implemented in 28 bits and the clock frequency is 75 MHz [2]. The Nyquist criteria states that at least two samples are required in order to reconstruct the output waveform [3]. In other words, the clock frequency must be as twice as the desired frequency. Thus, the theoretical frequency limitation is $f_{clk}/2$ = 37.5 MHz.

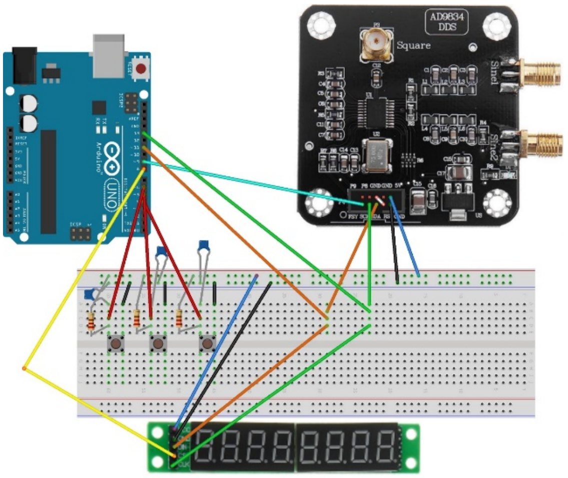


Figure 3: Schematics of the system. Source: Fritzing

The DDS is controlled by an Arduino Uno, ATmega328P microcontroller, via Serial Peripheral Interface (SPI). The microcontroller is also connected into a display, type MAX7219, also via SPI.

Both the DDS and the display are programmed by loading to the internal registers a 16-bit word. The AD9834 includes a control register, that allows the user to set the device, and two frequency registers to specify the output frequency [2].

5. Results

In the first evaluation of results, after minor problems are solved, some different frequencies are set. When the frequency is increased, the signal gets too noisy and negligible. This is the result of passing the signal through a filter included in the module, with a non-appropriate cutoff frequency.

In order to avoid the negative effect of the filter, the signal is measured in the output of the AD9834. The output signal from 1 MHz to 30 MHz is correctly generated and read in the panel meter. The results are considerably better obtaining a functional prototype DDS based generator.

However, radio signals transmitters applications require a high quality of the signal. In order to evaluate it, SFDR is analyzed. Output plots of the signal, at two different frequencies, and its spectrum are shown in figures 4 and 5. It can be observed how the SFDR gets worse as the frequency increases.

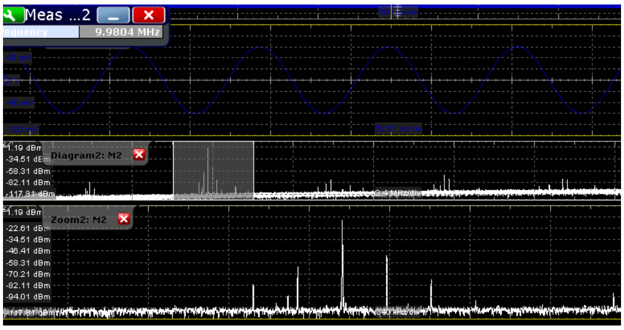


Figure 4: Output signal at 10MHz

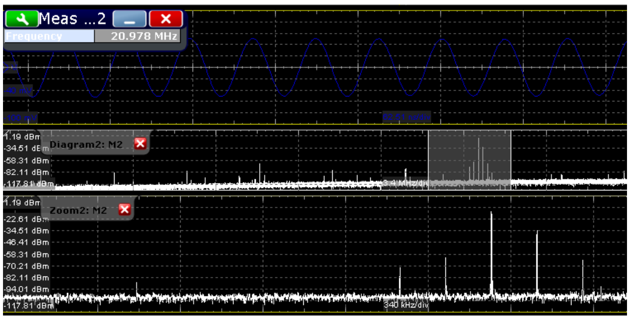


Figure 2: Output signal at 20 MHz

Some research about causes of unwanted spurs have been made. Truncation is a primal source of spectral impurity. In order to reduce the size of the sine lookup table, the output of the phase accumulator is truncated. This generates a truncation error signal that creates discrete spurs in the frequency domain.

Another reason of spectral impurity is the nonlinearity of the DAC. There will always be a difference between the theoretical output of the DAC and the actual output signal. This difference results to be not perfectly linear, so the output of the DAC will be distorted [4]. The output will be the expected signal plus harmonics, that are called quantizer Non-Linearity Spurs.

6. Conclusions

The aim of the present thesis is to discuss about Direct Digital Synthesis (DDS) and implement a sinewave generator based on this technology.

The contributions of the project include the construction of a functional prototype which can be used for educational purposes, among others. Regarding the primary motivation of the project, work as a local oscillator for RF transmissions, some improvements should be made.

The first improvement to be made is to replace the actual module filter, by a better one, with higher cut off frequency and slope. The filter will act as an antialiasing filter, attenuating the images response that appear in the spectrum at fclk ± fout.

Furthermore, some simulations demonstrate that quantization noise have an important role in the quality of the signal. In order to diminish it, accuracy should be increased. It depends on the clock frequency and the number of bits of the phase accumulator. In order to increase the accuracy, the DDS device should be changed. The resolution of the AD9834 is calculated as follows:

$$\frac{f_{clk}}{2^n} = \frac{75\ MHz}{2^{28}} = 0.28\ Hz$$

7. References

[1] E. Murphy and . C. Slattery , “All about direct digital synthesis,” *Analogue dialogue,* 2004.

[2] Datasheet, “20 mW Power, 2.3 V to 5.5 V, 75 MHz complete DDS,,” 2003-2014. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD9834.pdf. [Accessed July 2019].

[3] “Fundamentals of direct digital synthesis,” *Analogue devices, Inc., Tech. rep,* 2017.

[4] K. Gentile and . R. Cushing, *A Technical Tutorial on Digital Signal Synthesis,* Analogue devices, Inc., Tech. rep, 1999.

DESARROLLO DE UN GENERADOR BASADO EN LA SÍNTESIS DIGITAL DIRECTA

Autor: Teresa Corchado Arozarena

Director: Csaba Szombathy

Entidad colaboradora: Budapest University of Technology and Economics

1. Introducción

La síntesis digital directa (DDS) es un método para producir, mediante bloques de procesamiento digital, una forma de onda analógica a partir de una frecuencia de reloj como referencia. La señal de salida es típicamente una onda sinusoidal, pero también es posible generar ondas cuadradas y triangulares.

DDS es la solución para aplicaciones que exigen una precisa generación de frecuencia y agilidad, requisito clave para varias industrias. Un dispositivo DDS es digitalmente programable, lo que significa que es fácil de controlar y es siempre estable. Los sistemas de comunicación, incluyendo la modulación o las transmisiones directas de RF, pueden destacarse entre la amplia variedad de aplicaciones de DDS.

El objetivo de este proyecto es diseñar e implementar un generador de onda sinusoidal, destinado a ser utilizado como oscilador local para transmisiones de RF. Formará parte de un sistema modulador DRM30, que emitirá señales de radiofrecuencia desde la universidad.

2. Metodología

Para alcanzar los objetivos del proyecto, se ha aplicado la siguiente metodología: investigación en profundidad sobre los fundamentos de DDS y su modo de funcionamiento, programación del dispositivo DDS para que genere una onda sinusoidal a una frecuencia establecida y escritura de un programa en Arduino, para que la frecuencia pueda ser seleccionada externamente.

Finalmente, se realiza la evaluación de los resultados y se consideran las posibles mejoras.

3. Teoría del funcionamiento

Se entiende por un dispositivo DDS completo, aquel que incluye un oscilador controlado numéricamente (NCO) y un convertidor D/A en un solo chip. El NCO consiste en el acumulador de fase y la tabla de búsqueda de asignación sinusoidal. El sistema se muestra en la figura 1.

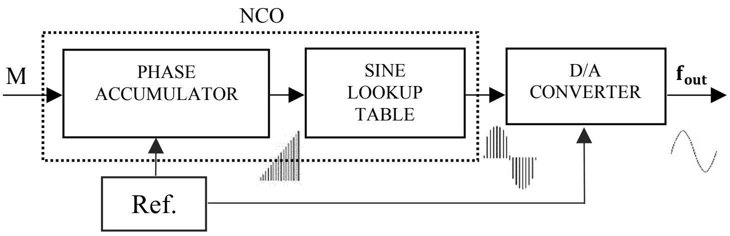


Figura 3: Componentes de un sintetizador directo digital. Fuente: Elaboración propia.

La entrada al acumulador de fase se denomina palabra crucial (M). Especifica la frecuencia de salida deseada en un número binario de 24 a 48 bits. Cada ciclo de reloj, M se suma al valor contenido en el acumulador de fase, hasta que se desborda y vuelve a empezar. El acumulador de fase calcula los valores de fase de un ciclo completo de la onda senoidal, antes de desbordarse [1].

Para entender el modo de funcionamiento del acumulador de fase, se puede utilizar la rueda de fases mostrada en la figura 2. La rueda representa un ciclo completo de una onda sinusoidal de 0º a 360º, siendo cada punto del círculo el punto de fase equivalente de la onda de salida. El número de puntos en la rueda es igual a 2^n.

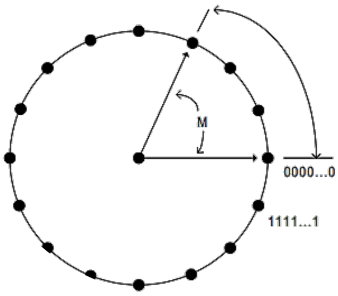


Figure 2: Rueda de fases. Fuente: Analog devices, 2004 [1]

M representa el tamaño de salto que el acumulador de fase realiza cada ciclo de reloj a lo largo de la rueda. Cuando la palabra de sintonía es grande, el valor del acumulador de fase tardará menos tiempo en desbordarse. Esto hace que M y la frecuencia de salida estén directamente relacionadas. Cuanto mayor sea la palabra de sintonía, mayor será la frecuencia de salida y viceversa [1].

La tabla de búsqueda sinusoidal contiene todos los valores de amplitud digital correspondientes a un ciclo completo de una onda sinusoidal. La salida del acumulador de fase sirve como dirección para la tabla de búsqueda sinusoidal que actúa como convertidor de fase a amplitud.

Finalmente, el DAC convierte la amplitud digital en el valor correspondiente de tensión o corriente. [1]

La ecuación básica del DDS relaciona la frecuencia de salida en función de la palabra crucial, la frecuencia del reloj y el número de bits del acumulador de fase. (n).

$$f_{out} = M \times \frac{fclk}{2^n} \quad [1]$$

4. Implementación

El sistema, mostrado en la figura 3, consiste en un dispositivo DDS que emite una onda senoidal a la frecuencia deseada. La frecuencia se puede ajustar externamente con algunos botones y además se muestra en una pantalla.

El dispositivo DDS es del tipo AD9834, incorporado en un módulo. El acumulador de fase está implementado en 28 bits y la frecuencia de reloj es de 75 MHz [2]. El criterio de Nyquist establece que se necesitan al menos dos muestras para reconstruir correctamente la onda de salida [3]. En otras palabras, la frecuencia del reloj debe ser el doble de la frecuencia deseada. Así, la limitación de frecuencia teórica es $f_{clk}/2 = 37.5$ MHz.

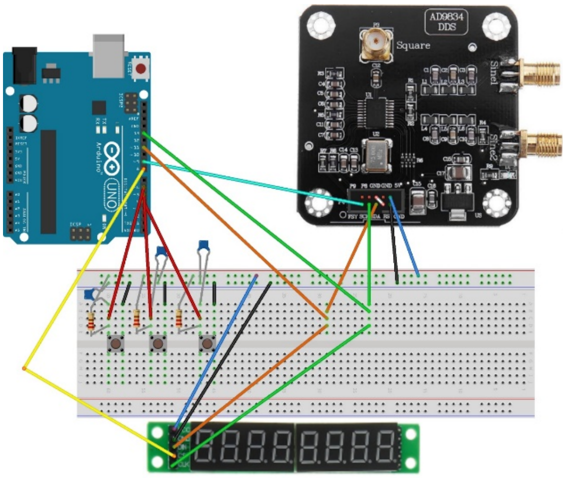


Figure 3: Esquema del circuito. Fuente:Fritzing, 2019

El DDS está controlado por un microcontrolador Arduino Uno, ATmega328P, a través del bus de interfaz de periférico serie (SPI). El microcontrolador también está conectado a una pantalla, tipo MAX7219, también por SPI.

Tanto el DDS como la pantalla se programan cargando en los registros internos una palabra de 16 bits. El AD9834 incluye un registro de control, que permite al usuario configurar el dispositivo, y dos registros de frecuencia para especificar la frecuencia de salida [2].

5. Resultados

En la primera evaluación de resultados, después de resolver problemas menores, se establecen frecuencias de salida diferentes. Cuando la frecuencia aumenta, la señal se vuelve demasiado ruidosa y poco estable. Esto resulta ser el efecto de pasar la señal a través de un filtro incluido en el módulo, con una frecuencia de corte inadecuada.

Para evitar el efecto negativo del filtro, la señal se mide en la salida del AD9834, antes de la salida del módulo. La señal de salida se genera correctamente de 1 MHz a 30 MHz y se lee en el medidor del panel. Los resultados son considerablemente mejores obteniendo un prototipo funcional basado en un generador DDS.

Sin embargo, las aplicaciones de transmisores de señales de radio requieren una alta calidad de la señal. Para evaluarla, se analiza la SFDR. En las figuras 4 y 5 se muestran los diagramas de salida de la señal, a dos frecuencias diferentes, y su espectro. Se puede observar cómo la SFDR empeora a medida que aumenta la frecuencia.

Se han realizado algunas investigaciones sobre las causas de los espurios no deseados. El truncamiento es una fuente primaria de impurezas espectrales. Para reducir el tamaño de la tabla de búsqueda de asignación sinusoidal, la salida del acumulador de fase se trunca. Esto genera una señal de error de truncamiento que crea espurios discretos en el dominio de la frecuencia.

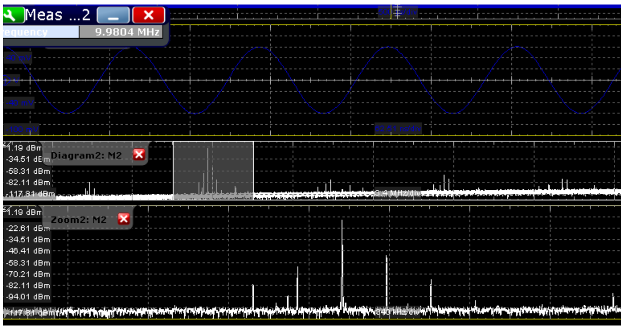


Figure 4: Output signal at 10MHz

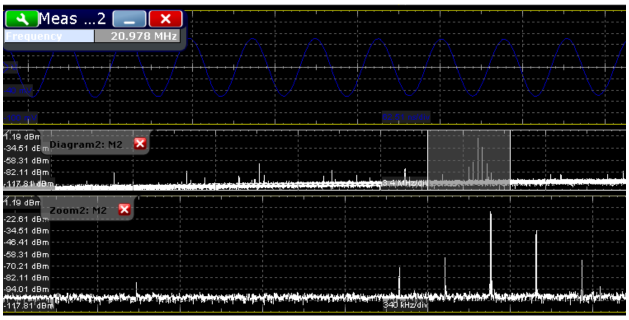


Figure 4: Output signal at 20 MHz

Otra razón de la impureza espectral es la no linealidad del convertidor analógico a digital. Siempre habrá una diferencia entre la salida teórica del DAC y la señal de salida actual. Esta diferencia resulta no ser perfectamente lineal, por lo que la salida del DAC se distorsionará [4]. La salida será la señal esperada más los armónicos.

6. Conclusiones

El objetivo del presente trabajo es discutir sobre la Síntesis Digital Directa (DDS) e implementar un generador de ondas senoidales basado en esta tecnología.

Las aportaciones del proyecto incluyen la construcción de un prototipo funcional que puede ser utilizado con fines educativos, entre otros. En cuanto a la motivación principal del proyecto, el trabajo como oscilador local para las transmisiones de RF, se deben hacer algunas mejoras.

La primera mejora que se debe hacer es reemplazar el filtro del módulo actual, por uno mejor, con mayor frecuencia de corte y pendiente. El filtro atenuará la respuesta de las imágenes que aparecen en el espectro a fclk ± fout.

Además, ciertas simulaciones demuestran que el ruido de cuantificación tiene un papel importante en la calidad de la señal. Para disminuirlo habría que aumentar la resolución cambiando el tipo de DDS empleado. La resolución del AD9834 se calcula de la siguiente manera:

$$\frac{f_{clk}}{2^n} = \frac{75\ MHz}{2^{28}} = 0.28\ Hz$$

7. Referencias

[1] E. Murphy and . C. Slattery , “All about direct digital synthesis,” *Analogue dialogue,* 2004.

[2] Datasheet, «20 mW Power, 2.3 V to 5.5 V, 75 MHz complete DDS,,» 2003-2014. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD9834.pdf. [Último acceso: July 2019].

[3] «Fundamentals of direct digital synthesis,» *Analogue devices, Inc., Tech. rep,* 2017.

[4] K. Gentile y . R. Cushing, *A Technical Tutorial on Digital Signal Synthesis,* Analogue devices, Inc., Tech. rep, 1999.

BACHELOR'S DEGREE IN INDUSTRIAL ENGINEERING

BACHELOR'S DEGREE FINAL THESIS

DEVELOPMENT OF A DIRECT DIGITAL SYNTHESIS BASED GENERATOR

DESCRIPTIVE MEMORY

Author: Teresa Corchado Arozarena

Director: Csaba Szombathy

Collaborating entity: Budapest University of Technology and Economics

Madrid

July 2019

INDEX

List of figures

1. Introduction

1.1. Approach and motivation

Direct digital synthesis (DDS) is a method of producing, by means of digital processing blocks, an analog digital waveform from a reference clock frequency. The output signal is typically a sine wave, but it is also possible to generate square and triangle waves. The input of the system is a binary number that specifies the output frequency called tuning word (M). A complete DDS modulator integrates DDS technology, Numerically-controlled oscillator (NCO), and a D/A converter into a single chip. Figure 1 shows a basic diagram of the system.

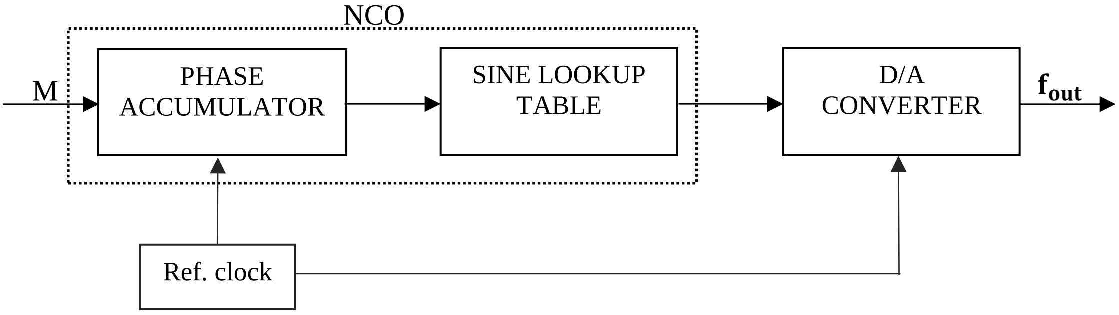


Figure 1: Components of a direct digital synthesizer. Source: Own elaboration, 2019.

The aim of the thesis is to develop a DDS-based function generator for RF transmissions. The DDS, an integrated circuit type AD9834, will be programmed via a microcontroller type ATMEGA328P. The output frequency will be set externally using buttons and showed on an 8 digits display type MAX7219. The DDS and the display will be connected to the microcontroller in an SPI communication.

Furthermore, the contents of the project extend to several areas of industrial electronic engineering. Knowledge in subjects as electronics, digital electronics, microprocessors and embedded systems must be applied.

1.2. State of the art

Waveforms generators are extensively used in many different fields as industry, medicine or communications. Several methods have been developed over the years to improve basic performance characteristics. Direct digital synthesis is the solution for applications that demand precise frequency generation or frequency agility, key requirement to several industries. For this reason, modern waveform generators generally use DDS technology, high resolutions are required that analog generators can't provide [1].

DDS is used in a broad range of applications; the following stand out [2]:

- Communication systems including modulation or direct RF transmissions.
- Generating a frequency stimulus in industrial or biomedical test equipment applications.
- Filter characterization.
- Component testing.

Today two dominant technologies of frequency generation exist on the market: DDS and Phase locked loops (PLLs). The many benefits that DDS technology offers make these devices an interesting alternative to other methods, including PPLs.

PLLs are a negative-feedback system used for generating a stable output signal at a reference frequency that is the input of the system. The system compares the difference between the phase of the input signal and that of the output (error signal) and reduces it to zero in steady state. Figure 2 shows a simple Block Diagram of PLL. [1]

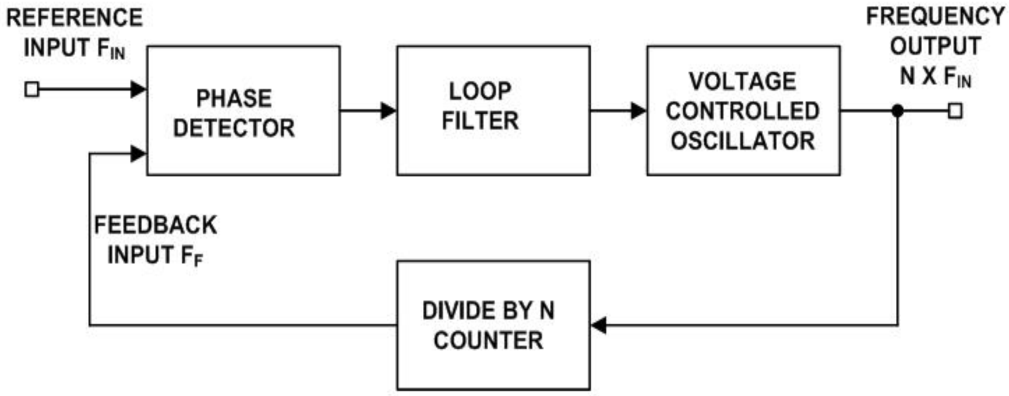


Figure 2: PLL Block Diagram. Source: Teledyne LeCroy, 2013 [3].

The main advantage of PPLs is the high quality of the output signal. They have low phase noise, as DDS, and high spurious-free dynamic range (SFDR), while DDS devices offer a lower spectral purity. The spectrum of a signal is the representation of the amplitude of the signal vs the frequency. SFDR is calculated as the ratio between the amplitude of the fundamental frequency being generated and the amplitude of the next largest harmonic. In theory, an ideal signal generator will generate frequency content only at the desired frequency. However, there are always going to be harmonics, called spurs, at other frequencies. Nevertheless, their difficulties to tune the output frequency and the slow response, make DDS more suitable for many applications, including the RF modulator, that require fast switching frequencies [4].

Benefits of DDS can be summarized in the following three points:

- It provides good frequency resolution and stability. Accuracy is limited by the crystal oscillator. As DDS implementation is always stable, unlike analog devices, there is no need of a gain control.
- It is digitally programable which leads to rapidly and easily modify the output frequency without changing any external component. In addition, it can operate over a wide range of frequencies.
- With DDS-based circuits, it is possible to perform advanced capabilities, as a modern function generator, at a very low cost. Furthermore, these devices have a surprisingly low power consumption and a small package.

1.3. Objectives and motivation

The aim of the present project is to design and implement a sine wave generator intended to be used as a local oscillator for RF transmissions. It will be part of a DRM30 modulator system, showed in figure 3 that will be broadcasting RF radio signals from the university. The IQ modulator is being developed by other students while this project is focused on the carrier generator. It will consist on a DDS based oscillator dedicated solely to generate sine waves for the modulator. The output frequency will be in a range from 1 to 30 MHz.

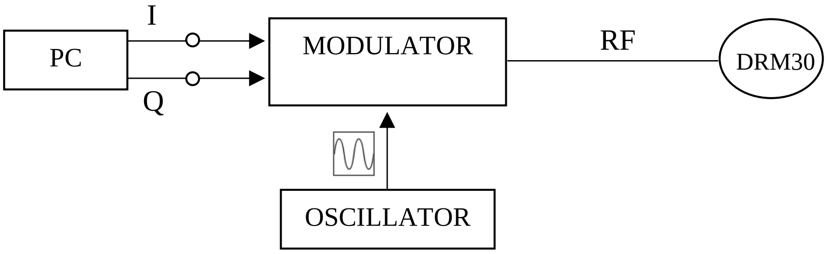


Figure 3: DRM30 modulator system. Source: Own elaboration, 2019.

The other major motivation is the use of the project for educational purposes. Signals generators are frequently used in laboratories and this one has the benefit of being a low-power device and it only needs a 5 V power supply voltage.

Furthermore, it can be used, besides for frequency generation and RF communication systems, for frequency phase tuning and modulation.

The objectives of the project are the following ones:

1. Investigate and understand DDS technology.
2. Gain experience in the design of mixed signals circuits.

3. Program the microcontroller to communicate via SPI (Serial Peripheral Interface) with other devices.

4. Connect and program the registers of the DDS.

5. Connect and program the registers of the display.

6. Design and implement a circuit connecting the pushbuttons to the Arduino Board in order to set the frequency.

1.4. Methodology

In order to achieve the objectives of the project mentioned above, the following methodology has been applied:

1. In-depth research about fundamentals of DDS and its theory of operation.

2. Selection of the microcontroller and implementation of an schematic design.

3. Programming the DDS device, through the microcontroller, so it could generate a sine wave at an established frequency.

4. Writing a code so any frequency could be generated.

5. Add a display and some pushbuttons to the circuit, so the frequency could be set externally and showed in the display.

6. Finally, evaluation of results and possible improvements.

7. Writing the descriptive memory.

The chart showed in figure 4 describes the organization of the project according to the methodology just described.

	November					January				February				March				April				May				June				July			
1	🟨	🟨	🟨	🟨		🟨	🟨																										
2								🟦	🟦																								
3										🟩	🟩	🟩	🟩																				
4														🟧	🟧	🟧	🟧	🟧															
5																		🟩	🟩	🟩													
6																					🟪	🟪	🟪	🟪									
7																								🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Figure 4: Organizational chart. Source: Own elaboration, 2019.

1.5. Thesis organization

The descriptive memory of the project aims to be organized in a way, so it describes, as clear as possible, the process followed. The different sections of the memory are mostly organized in chronological order.

First, Direct Digital Synthesis technology is explained, in detail, in chapter 2. The theory of operation of its components is explained in subsections of the chapter. Then, DDS technology is applied to the DDS device, AD9834, that it is used for the project. General parameters and specifications of DDS are specified for the device.

Chapter 4 includes the software of the project. The way of programming each device is explained, the last subsection includes the explanation in detail of the code. Chapter 5 includes a description of each device and the schematics of the whole circuit.

Finally, results and conclusions are discussed in chapter 6 and 7. Future work, including possible solutions in order to improve the quality of the output signal, is also discussed.

2. Direct Digital Synthesis Fundamentals

2.1. Introduction

Before talking about the theory of operation, let's review about the basic concepts of sinusoidal signals. They belong to the class of periodic complex exponentials signals, which have the form:

$$x(t) = \beta(t)e^{j\omega t}$$

β(t) is a function of time (complex or real) or either a constant. The angular frequency is denoted by ω. Using Euler's identity, x(t) can also be write as:

$$x(t) = \beta(t)[cos(\omega t) + jsin(\omega t)]$$

Extracting the real part of it, the form of a sinusoidal signal is obtained:

$$x(t) = Acos(\omega t)$$

Sine waves can be defined as time varying signals with a repetitive angular phase range of 0 to 2π. The phase is linear as shown in figure 5. The angular frequency is related to the natural frequency, f, by ω = 2πf [5].

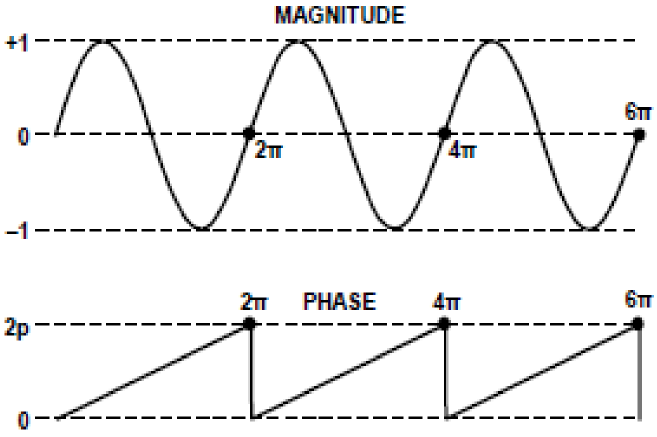


Figure 5: Linearity of sinusoidal signals phase. Source: Analog devices, 2014 [6]

2.2. Theory of operation

A complete DDS system includes two main blocks: the numerically-controlled oscillator (NCO), and a D/A converter. The NCO consists of the phase accumulator, heart of the system, and the sine look up table.

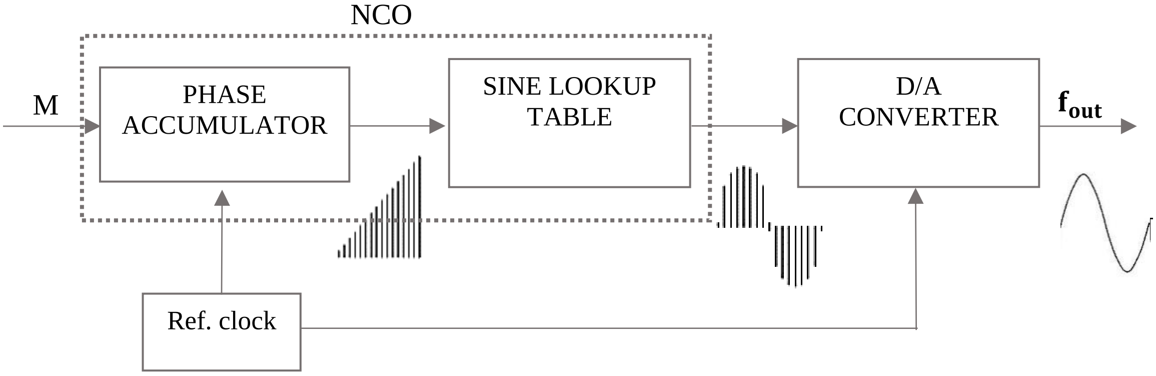


Figure 6: Signal flow through the DDS. Source: Own Elaboration, 2019.

The input to the phase accumulator is the tuning word, binary number of 24 to 48 bits, that specifies the desired output frequency. With this digital word, the phase accumulator computes a phase value. The sine lookup table contains all the digital amplitude values corresponding to a whole cycle of a sine wave. The output of the phase accumulator serves as an address for the sine lookup table acting as a phase-to-amplitude converter. When generating a constant frequency, the data in the phase accumulator increases linearly so it generates a ramp. Finally, the DAC converts the digital amplitude into the corresponding value of analog voltage or current. [2]

2.2.1. Phase accumulator

In order to understand the mode of operation of the phase accumulator, the phase wheel showed in figure 5 can be used. The generation of the sine wave can be compared to a vector rotating around the wheel at a constant velocity. The wheel represents a whole cycle of a sine wave from 0º to 360º, being each point of the circle the equivalent phase point of the output wave. The number of points in the wheel is equal to 2^n , being n the number of bits of the phase accumulator. This number will determine the resolution of the DDS [2].

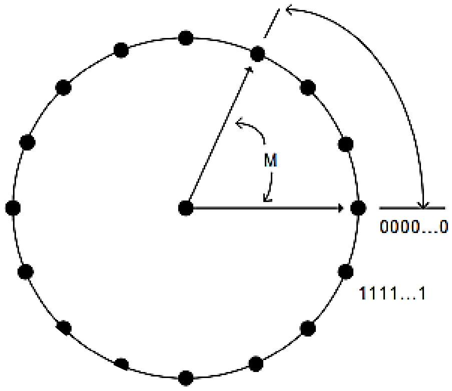


Figure 7: Phase wheel. Source: Analog dialogue, 2004 [4]

The data in the phase accumulator is uploaded each clock pulse. The binary number M, contained in the delta phase register, is added to the value of the phase accumulator every clock cycle, until it overflows and restarts again. The phase accumulator steps through every point of the phase wheel before the register value overflows. As a result of this, M will determine how many clock cycles needs the phase accumulator to overflow and restart. For instance, when M is 0…01, the phase accumulator register value is going to be updated by 1 each clock pulse. This means that is going to overflow after 2^{n} clock cycles. When the tuning word is bigger, the phase accumulator register value will take less time in overflowing. In other words, M represents the jump size the phase accumulator does every clock cycle along the wheel. This makes that M and the output frequency are directly related. The higher the tuning word, the higher the output frequency and vice versa. The output frequency does not depend only in the tuning word but also in the clock frequency [5].

2.2.2. Basic tuning equation

The basic tuning equation relates the output frequency as a function of the tuning word, the clock frequency and the number of bits of the phase accumulator.

During a time interval (Δt) the phase rotation can be calculated.

$$\Delta Phase = \omega \Delta t$$

Substituting ω by the natural frequency and the time period by the clock frequency (Δt=1/fclk):

$$f = \Delta Phase \times \frac{fclk}{2\pi}$$

Sine waves have a phase range of 0 to 2π that represents a whole cycle of the signal. The digital implementation is based on the linearity of the phase. The accumulator scales the range of phase numbers into the binary number 2^{n}. Considering that $0 < \Delta\mathrm{Phase} < 2\pi$, this relation is deducted: $0 < \Delta\mathrm{Phase} < 2^{\mathrm{n}} - 1$. Applying the equality $2\pi = 2^{\mathrm{n}}$:

$$f = \Delta Phase \times \frac{fclk}{2^{n}}$$

Introducing the tuning word as the ΔPhase, the basic tuning equation for DDS technology is obtained [6].

$$f = M \times \frac{fclk}{2^{n}}$$

2.2.3. Sine look up table

The output of the phase accumulator is a ramp that represents the phase values of a cycle of a sine wave. In order to generate the digital signal, a phase-to-amplitude conversion must be made. The sine lookup table is the one that performs the operation. The digital output of the phase accumulator serves as an address to the sine lookup table that will output the equivalent amplitude value.

In a practical DDS system, the output of the phase accumulator, which is n-bit size, is truncated. This way, only the first 13 to 15 MSBs are used as an address to the sine lookup table. The size of the sine lookup table is considerably reduced, no longer needing 2^n entries. The truncation does not affect the resolution of the output signal because the errors are smaller than the resolution of the DAC. The resolution of the sine lookup table is usually 2 or 4 bits more that the DAC. The phase truncation only adds a small but acceptable amount of phase noise to the final output [7].

In addition, it is only necessary the data from the phase accumulator of one quarter of the sine wave. The rest of the signal is synthesized using symmetry.

2.2.4. DAC

The final step of the generation of the waveform is to convert the digital output of the sine lookup table into an analog signal. This is done by the digital-to-analog converter. The purity of the output spectrum is mainly determined by the DAC because it always add some noise to the output. The reconstruction of a sine wave made by a converter is shown in figure 8.

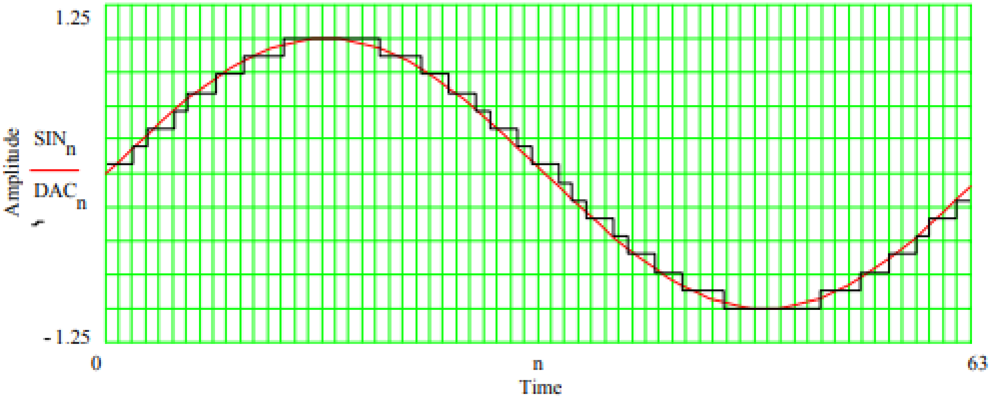


Figure 8: DAC reconstruction of a sine wave Source: Analog devices, 1999 [5].

The vertical lines are the instants in time at which the DAC output value is updated. The distance between them represent the sample period. As the converter output value is not a series of zero-width pulses, but rectangular pulses, there is some deviation between the output and the real signal. The vertical distance between both signals is the error introduced by the DAC, due to its finite resolution. This error is called the quantization error [5].

Finite resolution of the converter is one of the causes of spurs presence in the output. Higher DAC resolutions leads to an improvement of the output signal, decreasing the spurs content in the output spectrum.

2.3. AD9834 device

For this project, the DDS type AD9834 is used. The internal circuitry consists of the following main sections: a phase accumulator, frequency and phase modulators, SIN ROM, a DAC, a comparator, and a regulator [6]. The general characteristics of DDS are specified for the AD9834 in the following sections.

2.3.1. Phase accumulator implementation

The main component of the NCO, the phase accumulator, is implemented with 28 bits. The number of points of the phase wheel will be $2^{28} = 268435456$. It is the number of points the phase accumulator has to step through before overflowing. The phase numbers are scaled in the range $0 < \Delta\text{Phase} < 2^{28} - 1$ [6]. The DDS is provided with a 75 MHz clock rate resulting the following tuning equation:

$$f_{out} = M \times \frac{75\,MHz}{2^{28}}$$

The frequency resolution of the DDS system is calculated as follows:

$$\frac{f_{clk}}{2^n} = \frac{75\,MHz}{2^{28}} = 0.28\,Hz$$

The AD9834 includes a 10-bit DAC that converts the output of the sine lookup table into the output analog signal. As previously mentioned, the phase accumulator output is truncated in order to reduce the entries of the sine lookup table. Truncation errors must be lower than quantization errors introduced by the DAC, so the sine lookup table needs to have two bits more of phase resolution [6]. That is why the output of the phase accumulator is truncated into 12 bits.

2.3.2. Output frequency limitation

The frequency of the output signal is limited. The Nyquist criteria states that at least two samples are required in order to reconstruct the output waveform. In other words, the clock frequency must be as twice as the desired frequency [7]. Thus, the theoretical frequency limitation is f_{clk}/2 = 37.5 MHz.

However, this limitation is just theoretical. Many applications in DDS include a low pass filter after the output in order to eliminate image responses. They appear in the spectrum at $f_{clk} \pm f_{out}$. Since the low pass filter is not ideal, it is not going to have a flat response. That is why the practical limitation is around 40% of the clock frequency [5].

$$f_{limit} = 0.4 * 75\ MHz = 30 \text{ MHz}$$

3. Firmware

In this chapter, it is going to be described how each of the components of the system is programed. The last subsection includes an explanation of the whole code in Arduino.

3.1. Serial peripheral interface

The communication between the microcontroller, type ATMEGA328P, and the DDS, just as with the display, is via SPI. It is a bus data communication system between two devices named Master and Slave. The SPI basic system communication is shown in figure 9. Several slaves can be connected to one Master device.

The microcontroller is the master that programs both peripherals, DDS and display. In this project, the data is only needed to go one way, from the master to the slave, so three wires are required:

- Serial Clock (SCK): Synchronizes the system.
- Serial Data Output (SDO): Output data from the master to the slave.
- Slave select (SS): Enables the communication and chooses the slave to communicate with.

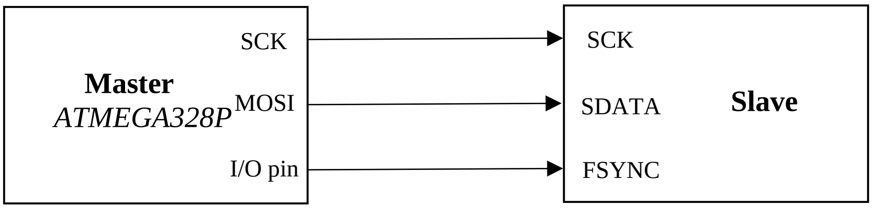


Figure 9: SPI communication diagram. Source: Own Elaboration, 2019.

3.2. AD9834 Registers

The DDS is programmed through the microcontroller via SPI. The microcontroller programs the device by writing to the registers. The following image, figure 10, shows the functional block diagram of the DDS. In the image, the main blocks of a direct digital synthesizer can be appreciated, including the phase accumulator, the sine lookup table and the digital to analog converter, which in this case is a 10-bit DAC. The block diagram also includes the different registers that can be programmed and the different input and output pins of the device.

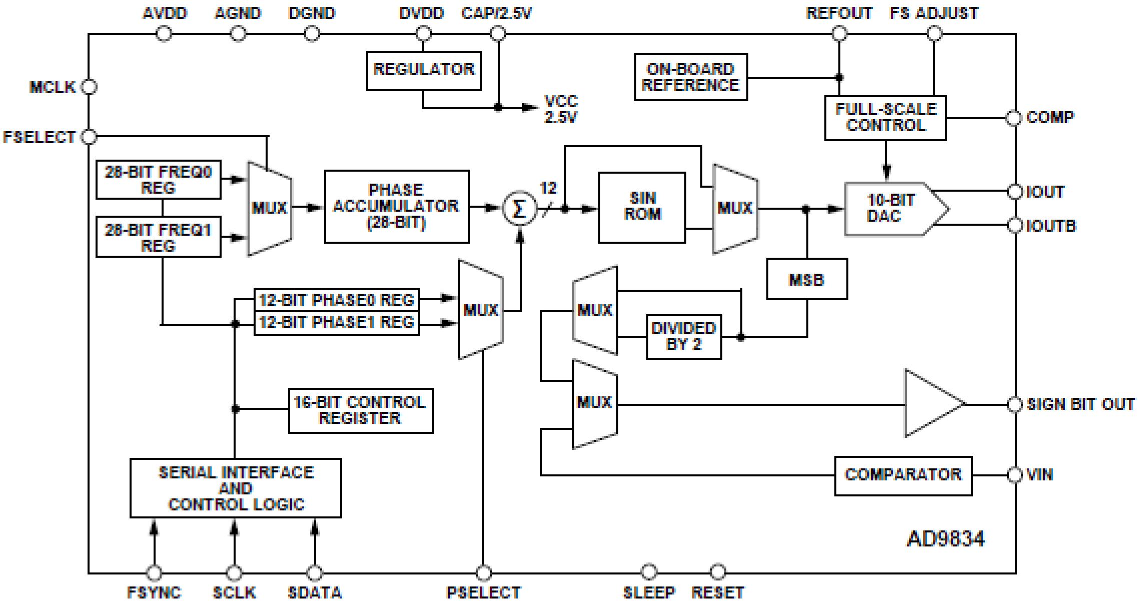


Figure 10: Functional block diagram of the DDS. Source: Analog devices, 2014 [6].

The DDS is programmed by loading to the registers a 16-bit word. It includes a control register, that allows the user to set the device, and two frequency registers (FREQ0 and FREQ1) to specify the output frequency. The frequency register is 28-bit wide, so two loads of 14 bits are required to modify the entire content of the frequency register. The two most significant bits of the word (DB15 and DB14) indicate the direction of the register, as shown in table 1 [6].

Table 1: Registers addresses

	DB15	DB14
Control register	0	0
Frequency register (FREQ0)	0	1
Frequency register (FREQ1)	1	0

The AD9834 also includes two 12-bits phase register for phase modulation. The content of the register is added to the output of the phase accumulator. This configuration supports Phase-shift keying modulation (PSK) and Frequency-shift keying modulation (FSK). In this project they are not going to be used, just the frequency must be controlled.

3.2.1. Control register

The control register is the first one to be written, since the mode of operation of the device is defined. Each bit of the control register, table 2, is described below [6].

Table 2: Control register

DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8
0	0	B28	HLB	FSEL	PSEL	PIN/SW	RESET
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
SLEEP1	SLEEP12	OPBITEN	SIGN/PIB	DIV2	0	MODE	0

- **DB15** and **DB14** must be equal to 0 in order to indicate that the content of the control register will be modified.
- **DB13** set to 1 allows to write the 28 bits of the frequency register. It is written in two consecutive loads, sending first 14 LSBs and then the 14 MSBs. In the case of DB13=0, the 28-bit register acts as two separate registers. The 14 MSBs or 14 LSBs can be modified independently of one another. As it is needed a 28-bit resolution, DB13=1.
- **DB12** is only significant when the previous bit is set to 0. DB12=0 allows to write the 14 LSBs and DB12=1 the MSBs. Since DB13=1, this bit in meaningful.
- **DB11** control the access of the phase accumulator to the frequency register, either FREQ0 or FREQ1. As the registered used is FREQ0, DB11=0.
- **DB10** control the access to the phase register which is not used.
- **DB9** decides whether the bits (DB9=0) or the pins (DB9=1) control the functions. Since the DDS is digitally programmed, DB9=0.
- **DB8** resets the internal registers, when is set to 1, but not the control, frequency or phase registers. When the DDS is turned on, it must be reset.
- **DB7** and **DB6** can be set to 1 in order to reduce power consumption powering down not used elements. DB7 turn off the MCLK and DB6 the DAC. Both functions are needed to be active so both bits are set to 0.
- **DB5** enables the digital outputs from the SIGN BIT OUT pin which are not needed.
- **DB4** and **DB3** are only relevant if the OPBITEN bit (DB5) is set to 1.
- **DB2** must be 0 in all cases.
- **DB1** controls the output in IOUT pin/IOUT B pin. The output of the DDS can be a sinusoidal signal but also a triangle wave. MODE= 0 results in a sinusoidal wave while MODE= 1 results in a triangle output.
- **DB0** must be 0 in all cases.

3.2.2. Frequency register

The AD9834 includes two 28-bit frequency registers but only the FREQ0 register is used. The address of the register is determined by bits DB15 and DB14, which are set to 0 and 1 respectively, corresponding to FREQ0. The bits left, DB13…DB0, specify the frequency, containing the tuning word. The first load to the frequency register will include the 14 LSBs while the latter the 14 MSBs (Table 3) [6].

Table 3: Frequency register bits

DB15	DB14	DB13…DB0
0	1	Tuning word 14 LSBs
0	1	Tuning word 14 MSBs

3.3. Microcontroller

Microcontrollers are integrated circuits that contains programmable input/output peripherals, processor, and memory. They are used in automatic control systems and they have multitude of applications.

The microcontroller, type ATMEGA328P, acts as the master of the SPI communication. It contains an 8-bit shift register where the data is loaded before being sent to the slave. The communication starts taking low the slave select. When data is loaded into the shift register, the SPI clock generator starts and the 8 bits are shifted into the slave. After sending one byte, the clock generator stops and the end of transmission flag (SPIF) is set [8].

3.3.1. SPI microcontroller registers

The microcontroller contains different registers besides the 8-bit shift register already explained. SPCR is a control register where each bit controls an SPI setting. SPDR contain the data to be sent to the slave. Finally, the status register SPSR is an only read register (except for bit SPI2X which is not used) that inform of the state of the transmission.

The control register bits of the ATMEGA328P are shown in the following table.

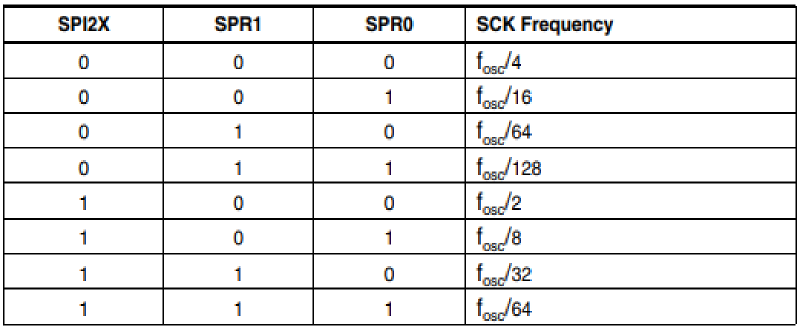
Table 4: SPI control register, SPCR

BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPCR	SPI	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

- **Bit 7** (SPIE: SPI Interrupt Enable): interrupts are not used.

- **Bit 6** (SPE: SPI Enable): it must be set up to 1 to enable SPI operation.
- **Bit 5** (DORD: Data Order): it decides whether the LSB or MSB of the data word is transmitted first. This bit is set to 0 so the MSB is transmitted first.
- **Bit 4** (MSTR: Master/Slave Select): this bit is set to 0 so the device is configured as a Master.
- **Bit 3** (CPOL: Clock Polarity): the clock polarity is defined depending on the characteristics of the slave. When CPOL=1, SCK is high when idle. When CPOL=0, SCK is low when idle.
- **Bit 2** (CPHA: Clock Phase): it decides whether the data is sampled on the leading edge (first edge of the pulse), CPHA = 0, or on the trailing edge (last edge of the pulse), CPHA = 1.
- **Bits 1, 0** (SPR1, SPR0: SPI Clock Rate Select 1 and 0): it controls the relationship between SCK and the oscillator frequency as showed in table 5.

Table 5: Relationship Between SCK and the Oscillator Frequency



SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

The data register, SPDR, described in table 6, writes to the 8-bit shift register to initiate data transmission. It is noted that the DDS load registers are 16-bit wide and the microcontroller send 8-bit data words. That is why there is a function in the code, that divides the data word, to send two consecutive loads of 8 bits [8].

Table 6: Data register, SPDR

BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPDR	MSB			DATA				LSB

3.4. Display MAX7219

The output frequency is shown in an 8-digit led display that is also programmed via SPI by the microcontroller. The mode of operation is similar to the one of the DDS but the data is sampled in the rising edge of the clock instead of in the falling edge. The system is shown in the following figure.

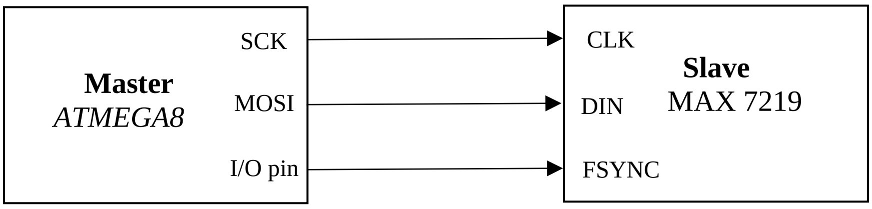


Figure 11: SPI display communication. Source: Own Elaboration, 2019.

The display contains a 16-bit register showed in table 8. D11-D8 bits contain the address of the register and D7-D0 contains the data. D15-D12 are don’t care bits [9].

Table 7: Display 16-bit register

D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
X	X	X	X	ADDRESS			
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
MSB			DATA				LSB

3.4.1. Initialization

In order to set up the display, some control registers are programmed in the initialization part. The corresponding addresses of the control registers are shown in the following table [9].

Table 8: Control register addresses

ADDRESS					HEX CODE	REGISTER
D15-D12	D11	D10	D9	D8		
X	1	1	0	0	0xXC	Shutdown mode
X	1	0	1	1	0xXB	Scan-limit
X	1	0	0	1	0xX9	Decode mode
X	1	0	1	0	0xXA	Intensity

When the display is turned on, all registers are reset and the display enters in shutdown mode. Shutdown register has two operation modes, shown in table 9. In order to initialize the device, normal operation is set.

Table 9: Shutdown register modes

MODE	ADDRESS CODE (HEX)	REGISTER DATA		
		D7-D1	D0	HEX CODE
Shutdown mode	0xXC	X	0	0xX0
Normal operation	0xXC	X	1	0xX1

The scan-limit register sets how many digits are displayed. Since the device can show from 1 to 8 digits, the maximum is chosen, in order to have more accuracy.

Table 10: Scan-limit register

MODE	ADDRESS CODE (HEX)	REGISTER DATA								
		D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
8 digits	0xXB	X	X	X	X	X	1	1	1	0xX7

Display brightness is also digitally programmable. It is provided by an internal pulse-width modulation where the lower nibble is controlled. The range is from 31/32 down to 1/32. After trying different values, an intermediate one (15/32) is selected. The following table includes the register code.

Table 11: Intensity register

MODE	ADDRESS CODE (HEX)	REGISTER DATA								
		D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
15/32	0xXA	X	X	X	X	X	1	1	1	0xX7

Finally, the decode mode register is programmed. Decode mode allows to set BCD code to each digit. When no-decode mode is selected, data bits refer to the segment lines. In this case, decode mode is set for all digits. The register data is showed in table 12.

Table 12: Decode mode register

MODE	ADDRESS CODE (HEX)	REGISTER DATA								
		D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
Code BCD digits 7-0	0xX9	1	1	1	1	1	1	1	1	0xFF

3.4.2. Digit registers

Digits are updated individually, so each digit has its address, showed in table 13. Since decode mode to BCD is set for all digits, only 4 bits are needed, so only D3-D0 are considered. Bits D6-D4 are disregarded and D7 sets the decimal point, which is not going to be used [9].

Table 13: Digit registers

ADDRESS					HEX CODE	REGISTER
D15-D12	D15-D12	D15-D12	D15-D12	D15-D12		
X	0	0	0	0	0xX0	No operation
X	0	0	0	1	0xX1	Digit 0
X	0	0	1	0	0xX2	Digit 1
X	0	0	1	1	0xX3	Digit 2
X	0	1	0	0	0xX4	Digit 3
X	0	1	0	1	0xX5	Digit 4
X	0	1	1	0	0xX6	Digit 5
X	0	1	1	1	0xX7	Digit 6
X	1	0	0	0	0xX8	Digit 7

3.5. Code

The code is explained by going through the functions, involving DDS, the display and finally the buttons. Once the functions are explained, the initialization part and main function are described.

3.5.1. Functions involving communication with the DDS and display

The first step to communicate with the DDS, just like with the display, is the function **send_frequency**. It sets the control register of the microcontroller (SPCR) for both peripheral and calls the respective functions for setting the frequency. The function receives the frequency in Hz.

It is important to consider the slave transmission characteristics in order to set the clock phase (bit 2) and clock polarity (bit 3) of the control register.

In the DDS, the clock is high when idle as shows figure 12, so CPOL=1. The data is sampled on the falling edge of the clock, which is the leading edge, so CPHA = 0.

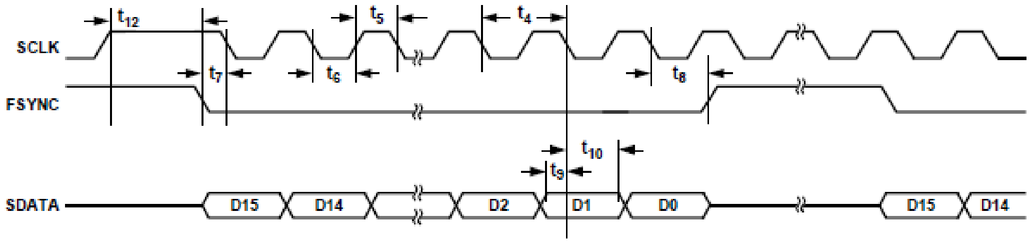


Figure 12: DDS timing characteristics. Source: AD9834 Datasheet, 2014 [6].

In the display, the clock is low when idle unlike the DDS, so CPOL=0. The clock phase bit is the same as in the DDS, the data is sampled in the rising edge of the clock which is the leading edge in this case.

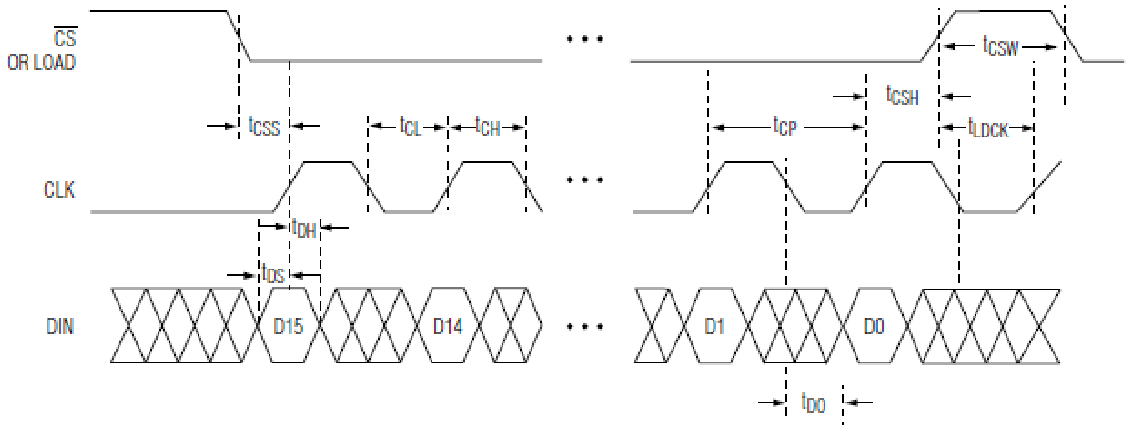


Figure 13: Max7219 timing characteristics. Source: Datasheet, 2003 [9].

The function sets the control register, in order to communicate with the DDS first, and calls *set_dds_freq* function. Then, the control register is set again and *set_display_digits* function is called.

```
        // Send_frequency function
    void send_frequency(uint32_t freq_in_hz){
      SPCR = (1<<6)|(1<<4)|(1<<3)|(1<<1)|(1<<0);
       set_dds_freq(freq_in_hz);
     SPCR = (1<<6)|(1<<4)|(1<<1)|(1<<0);
       set_display_digits(freq_in_hz);
    }
```

Set_dds_freq function receives the desired output frequency in Hz and calculates the tuning word, using the tuning equation of DDS. The resulting tuning word (28 bits) is divided in two parts. The 14 LSBs of the tuning word are sent first (DB13-DB0), with the direction of the frequency register DB15 = 0 and DB14 = 1. The 14 MSBs are sent in the same way.

```
        // Set_dds_freq function
    void set_dds_freq(uint64_t freq_in_hz){
     uint64_t freq = ( freq_in_hz << 28 ) / 75000000;
```

```
uint32_t freq1 = freq & 0b00000000000000000011111111111111;
freq1 = freq1 | 0b00000000000000000100000000000000;
uint32_t freq2 = freq >> 14;
freq2 = freq2 & 0b00000000000000000011111111111111;
freq2 = freq2 | 0b00000000000000000100000000000000;
send_data_to_dds(freq1);
send_data_to_dds(freq2);
}
```

Once the content of the frequency register is set with the previous function, **send_data_to_dds** function is called. It receives the 16-bit register and sends it in two parts through the function *spi_tranceiver*. This division is done because the microcontroller can only send 8 bits through SPI. In order to initiate the communication, the slave select (PB1) is taken low and, after the transmission, it is taken high again.

```
      // Send_data_to_dds function
void send_data_to_dds(uint16_t reg){
  PORTB = PORTB & (0b1111101);
  spi_tranceiver(reg >> 8);
  spi_tranceiver(0x00FF & reg);
  PORTB = PORTB | (0x02);
  }
```

Spi_tranceiver function sends the data directly to the DDS, it is the final step of the transmission. The microcontroller can only send an 8-bit word so it is called twice in order to send the whole tuning word which is 16-bit. SPDR loads the data into the buffer and the program waits until the transmission is completed, checking when the transmission flag (SPIF bit) is set.

```
      // Spi_tranceiver function
unsigned char spi_tranceiver (uint8_t data){
   SPDR = data;
   while(!(SPSR & (1<<SPIF) ));
   return(SPDR);
  }
```

Each digit is sent to a different address register, so the frequency has to be divided in individual bits. **Set_display_digits** function receives the desired output frequency in Hz and separates the frequency in individual bits. Then, they are sent individually to the display with the function *send_data_to_display(dig)*.

```
        // Set_display_digits function
void set_display_digits(uint32_t freq_in_hz){
  uint32_t num = freq_in_hz;
  for(uint32_t i=1; i<9 ; i++){
    uint32_t dig = num % 10;
    num = num / 10;
    dig = dig | (i<<8);
    send_data_to_display(dig);
  }
}
```

The operation of the next function, **Send_data_to_display**, is mainly the same as *send_data_to_dds*, but taking the display select bit (PB0) down, instead of taking down the DDS select bit.

```
        // Send_data_to_display function
void send_data_to_display(uint16_t reg){
   PORTB = PORTB & (0b1111110);
   spi_tranceiver(reg >> 8);
   spi_tranceiver(0x00FF & reg);
   PORTB = PORTB | (0x01);
  }
```

3.5.2. Initialization

In the setup part of the program, the outputs and inputs are defined, and the initialization of both peripherals is done.

The outputs of the microcontroller are the SPI wires to the DDS and the display. The inputs are the pushbuttons that set the output frequency. See figure 14.

The initialization of the DDS consists of three steps:

1. Send the control register, setting the reset bit (DB8) to 1.
2. Write to the frequency register with an initial frequency. The initial frequency is the default frequency, set as 1MHz.
3. Send the control register, setting the reset bit to 0 again

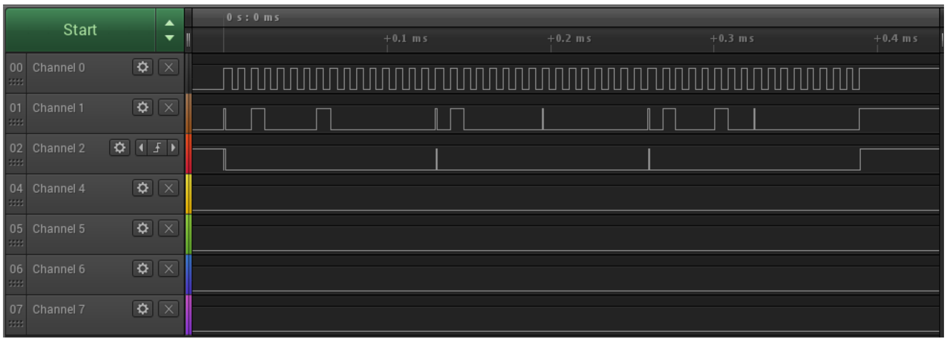


Figure 14: Logic analyzer output signals for the initialization part. Source: Own elaboration, 2019

. Table 14: Initial control register of the DDS

DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8
0	0	1	0	0	0	0	1
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0

The initialization of the display consists on sending the four different control registers explained before: shutdown mode, scan-limit, decode mode and intensity.

```
void setup() {
  DDRB = (1<<5)|(1<<3)|(1<<1)|(1<<2)|(1<<0);      //MOSI, FSYNC and SCLK as OUTPUT
  PORTB = 0b00101111;
  PORTD |= (1<<PB6)|(1<<PB5)|(1<<PB4); //define pull-ups of Inputs

  SPCR = (1<<6)|(1<<4)|(1<<3)|(1<<1)|(1<<0);   //talking to the DDS
  send_data_to_dds(0x2100);
  set_dds_freq(DEFAULT_FREQ);
  send_data_to_dds(0x2000);


  SPCR = (1<<6)|(1<<4)|(1<<1)|(1<<0);    //talking to the display module
  send_data_to_display(0x0C01);    //Turns the display on
  send_data_to_display(0x0B07);    //choose to display 8 bits
  send_data_to_display(0x09FF);    //Decode mode to BCD
  send_data_to_display(0x0A07);    //Brightness
  set_display_digits(DEFAULT_FREQ);
}
```

3.5.3. Pushbuttons

The system includes three pushbuttons, so the frequency can be selected externally by the user. One of the buttons moves along the 8 digits, and the others change the value of the digit, one goes up and the other goes down.

The following three functions, one for each button, avoid detecting several pulses while the buttons are pressed. They return 1 when the button is released.

```
    uint8_t
if_move_button_pressed(){
      if((PIND&0b00010000)      ==
0){

while((PIND&0b00010000)==
0){}
         return 1;
      }else{
```

```
uint8_t if_down_button_pressed(){
  if((PIND&0b01000000) == 0){
    while((PIND&0b01000000)== 0){}
    return 1;
  }else{
     return 0; }
}
```

```
uint8_t if_up_button_pressed(){
  if((PIND&0b00100000) == 0){
    while((PIND&0b00100000) == 0){}
    return 1;
  }else{
     return 0; }
}
```

Finally, the frequency is changed depending on the state of the buttons.

```
void loop() {
  if(if_move_button_pressed()){
    if(pos<10000000){
      pos*=10;
}else{
      pos = 1;
}
  }
```

```
if(if_up_button_pressed()){
    freq_in_hz += pos;
     send_frequency(freq_in_hz);
  }
  if(if_down_button_pressed()){
    freq_in_hz -= pos;
     send_frequency(freq_in_hz);
  }
}
```

4. Implementation

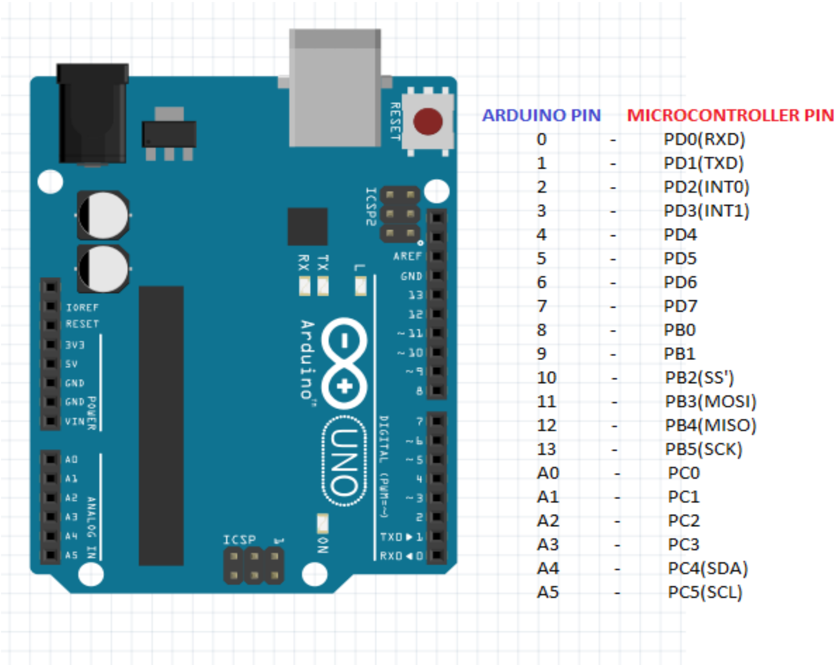
The above section explained how the registers of each of the components of the system are programed while in this section, the hardware is described. The three components of the model are described individually, specifying the pins that are connected and their functionality.

4.1. Arduino Uno board

The microcontroller is built-in in an Arduino Uno board. The pin mapping is shown in figure 15 and the used pins are the ones included in table 15. The microcontroller is connected to two different peripherals. It controls the DDS and the display via SPI, so it has one slave select for each one. In addition, the buttons are connected to input pins.

Table 15: Arduino pin description

Arduino pin	Microcontroller pin	Pin description
Pin 8	PB0	Display select
Pin 9	PB1	DDS select
Pin 10	PB2	Always defined as an output when using SPI. [8]
Pin 11	PB3	Master output data
Pin 13	PB5	SCK
Pin 4	PD4	Bottom move
Pin 5	PD5	Bottom up
Pin 6	PD6	Bottom down



ARDUINO PIN		MICROCONTROLLER PIN
0	-	PD0(RXD)
1	-	PD1(TXD)
2	-	PD2(INT0)
3	-	PD3(INT1)
4	-	PD4
5	-	PD5
6	-	PD6
7	-	PD7
8	-	PB0
9	-	PB1
10	-	PB2(SS')
11	-	PB3(MOSI)
12	-	PB4(MISO)
13	-	PB5(SCK)
A0	-	PC0
A1	-	PC1
A2	-	PC2
A3	-	PC3
A4	-	PC4(SDA)
A5	-	PC5(SCL)

Figure 15: Arduino pin mapping. Source: iCircuit, 2014 [11]

4.2.DDS module board

The DDS device, AD9834, it is built-in in a module, showed in figure 16. It keeps the AD9834 characteristics [6]:

- Power supply of +5V.
- Output frequency up to 37.5 MHz.
- Low power consumption.
- Sine and triangular output.
- Comparator that support square wave generation.
- Temperature range: -40℃ to +105℃.

The available pins of the module are shown in figure 17 and their functions are described below:

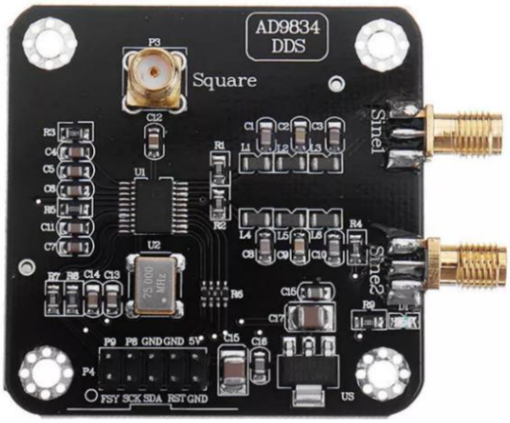


Figure 17: DDS module. Source: Banggood, 2019 [16].

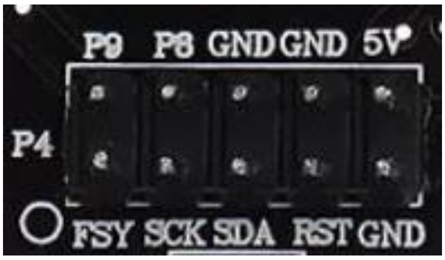


Figure 17: DDS module pins. Source: Banggood, 2019 [16].

- FSYNC acts as a chip enable. When it is taken low, the communication between the master and the DDS starts. After FSYNC goes low, the data is shifted into the device in each falling edge of the clock. At the end of the data transfer FSYNC is taken high.
- SDA is the serial data input.
- SCK is the serial clock input.
- 5V pin is the power supply.
- RST is the reset pin. It is grounded, however it is digitally-controlled.

4.3. MAX729

The MAX7219 is a display driver that interface microprocessors to 7-segment numeric LED displays of up to 8 digits, bar-graph displays, or 64 individual LEDs [9]. The device is connected to a 7-segment numeric LED as shown in figure 18 in order to act as a panel meter (figure 19).

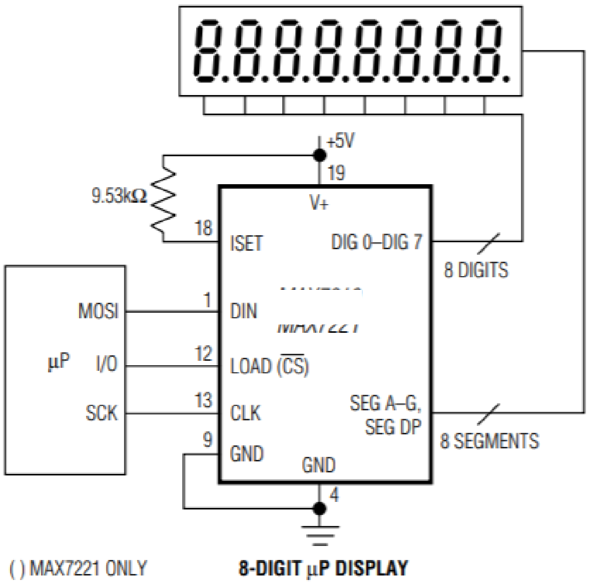


Figure 18: MAX7219 display circuit. Source: Maxim integrated, 2003 [9]

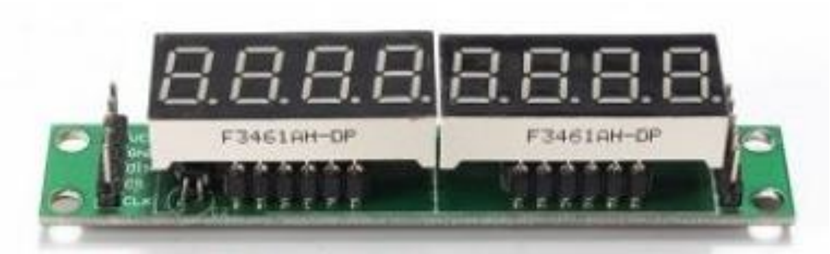


Figure 19: MAX7219 display. Source: Arduino learning, 2019 [13]

The display has five output pins to be connected showed in figure 20. DOUT, LOAD and CLK are connected to the microcontroller for the SPI communication.

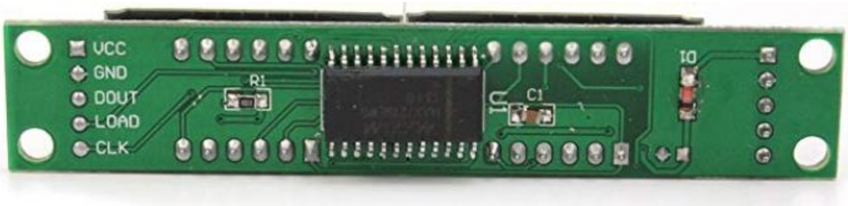


Figure 20: MAX7219 display pin connections. Source: Banggood, 2019 [15]

4.4. Additional circuit

The pushbuttons are connected to the microcontroller using a protoboard. When pressing a button, several pulses are detected before the signal gets stable, see figure 21. It is necessary to debounce the signal, so a single press doesn't appear like multiple presses. The problem can be solved by hardware or by software, in this case it is done by hardware.

The debouncer circuit is shown in figure 22. It consists in adding a capacitor, so it filters out the bouncings. R and C must be selected considering that the result of the product R*C is the time you would like to debounce the signal.

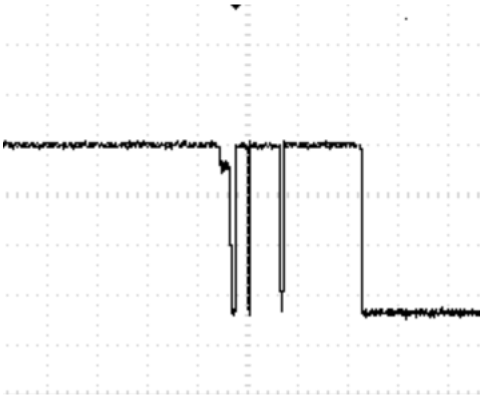


Figure 21: Bounce produced on a switch signal. Source: The lab book pages, 2010 [14].

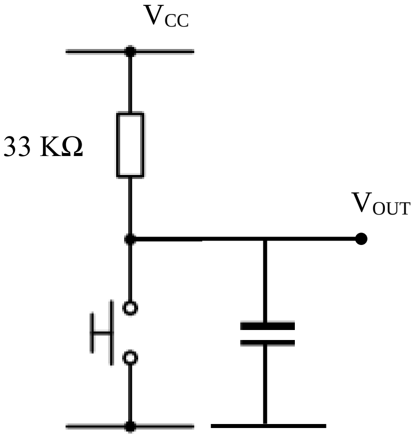


Figure 22: Debouncing circuit. Source: Own elaboration, 2019.

4.5. Schematics of the system

The following image (figure 23), created with the program *fritzing*, shows the schematics of the whole system including the microcontroller, the AD9834 and the display

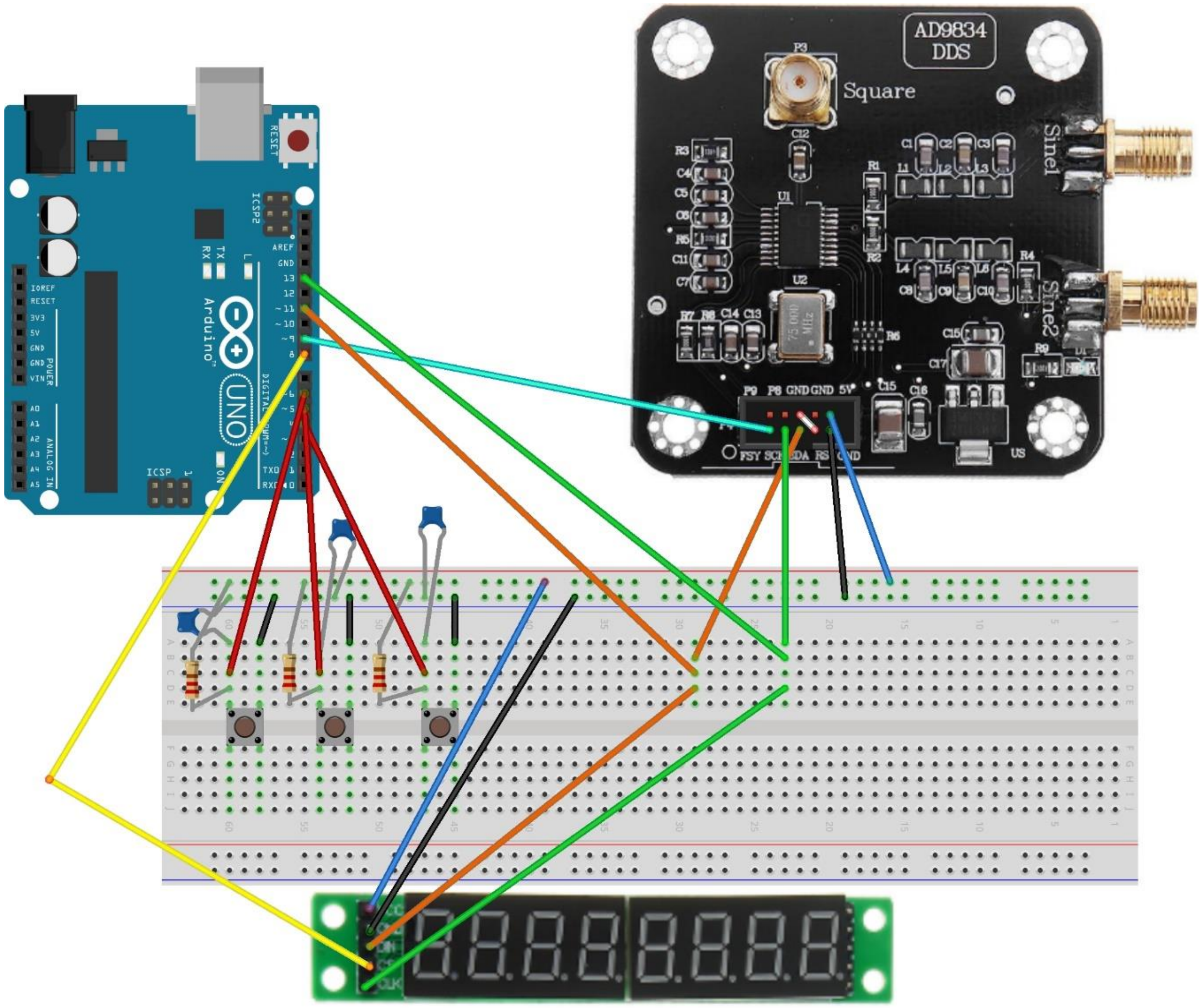


Figure 23: Schematics of the system. Source: Own Elaboration (fritzing), 2019.

5. Results

In this section, the results are described and then analyzed. Some images of the output signal at different frequencies are included. Finally, some conclusions are done talking about the limitations of the model and future work in order to improve the proposed model.

5.1. Assembly

The physical assembly of the system is shown in figure 24. The circuit follows the schematics included in the previous section (figure 23). This configuration is provisional, when the model is completely done and ready to be used, it is going to be put into housing. All the wires will be welded and only the display and the buttons will be visible.

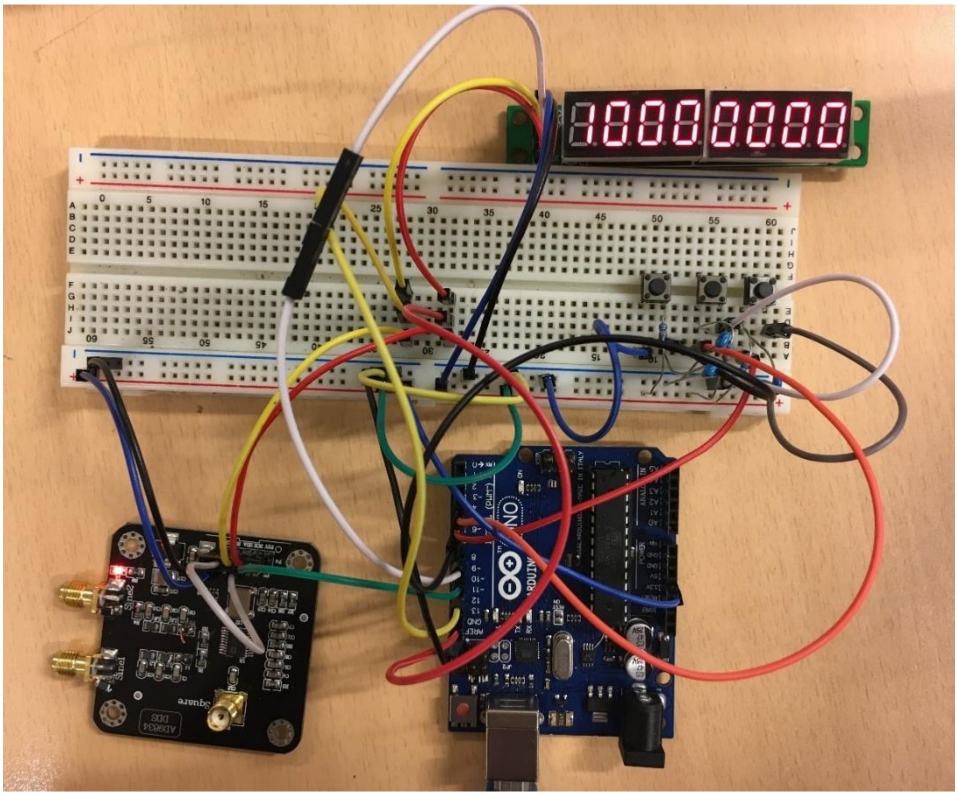


Figure 24: DDS based generator system. Source: Own elaboration, 2019.

5.2. Analysis results

Once the software and hardware are done, and minor errors are solved, the output signal is plot so it can be analyzed. As the frequency increases, the signal becomes too noisy and negligible. After some functional tests and research, the problem turns out to be the internal configuration of the AD9834 module board. The output of the AD9834 is pass through a filter which is cutting off the signal when high frequencies are set.

As already mentioned in section 2.3.2, many applications of DDS include a low pass filter, in order to attenuate the image responses that appears at fclk ± fout in the spectrum.

If the filter is not the appropriate, the cutoff requirements of the filter may affect the signal. This means that the sacrificed bandwidth, showed in figure 25, affects an important part of the Nyquist bandwidth. For this reason, when the frequency is increased, the cut off frequency of the filter cannot be big enough, resulting in an attenuation of the signal.

An ideal filter will have a flat response over the Nyquist band, and it will suppress the images responses. The filter, as it cannot be ideal, it is always going to affect a part of the Nyquist bandwidth. The optimal solution is that the sacrificed bandwidth, as shown in figure 25, is the minimum so it does not affect the signal at its maximum frequency.

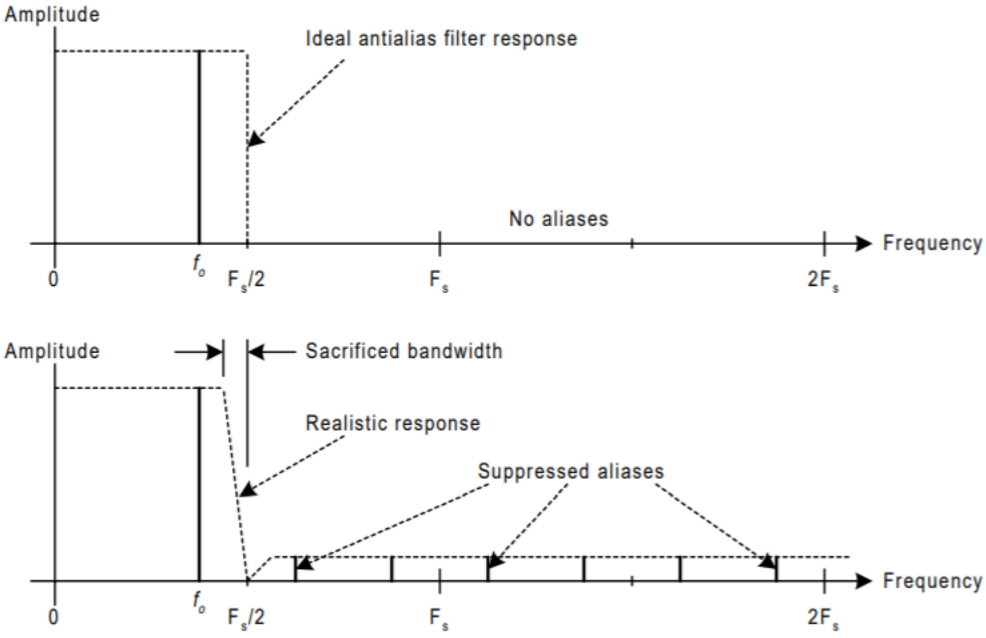
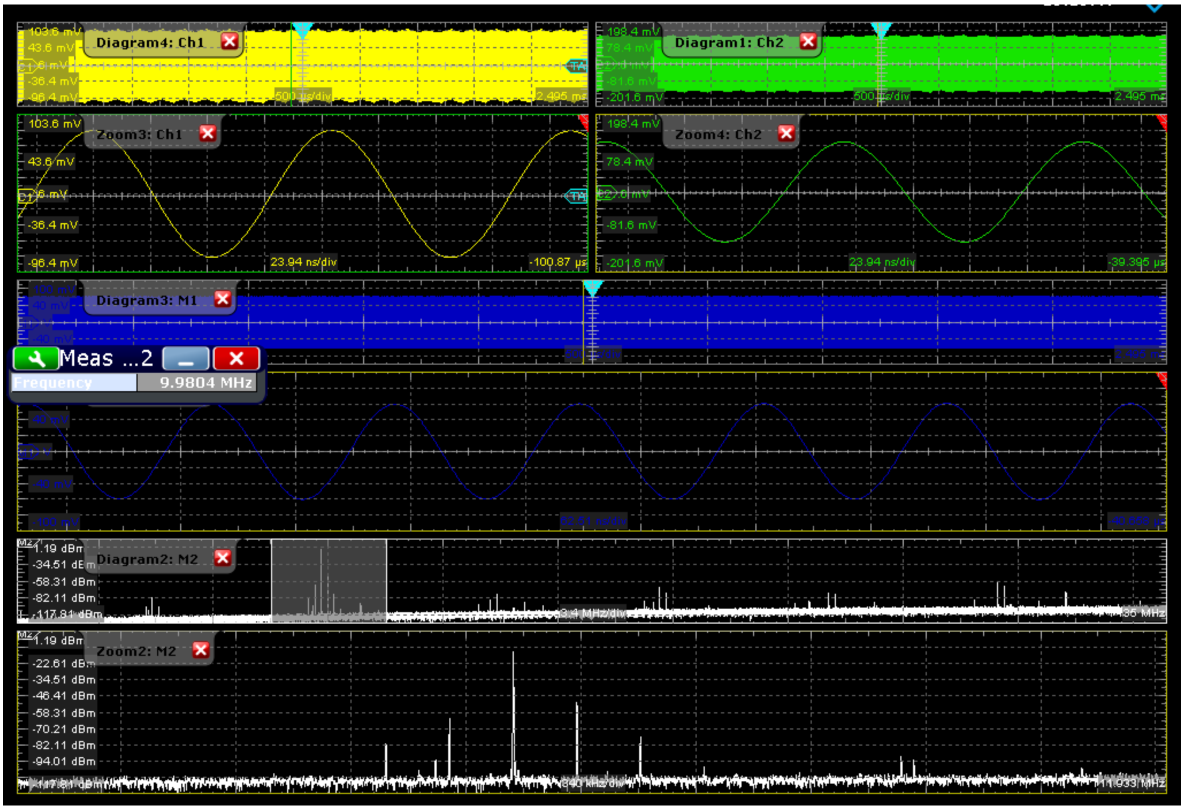


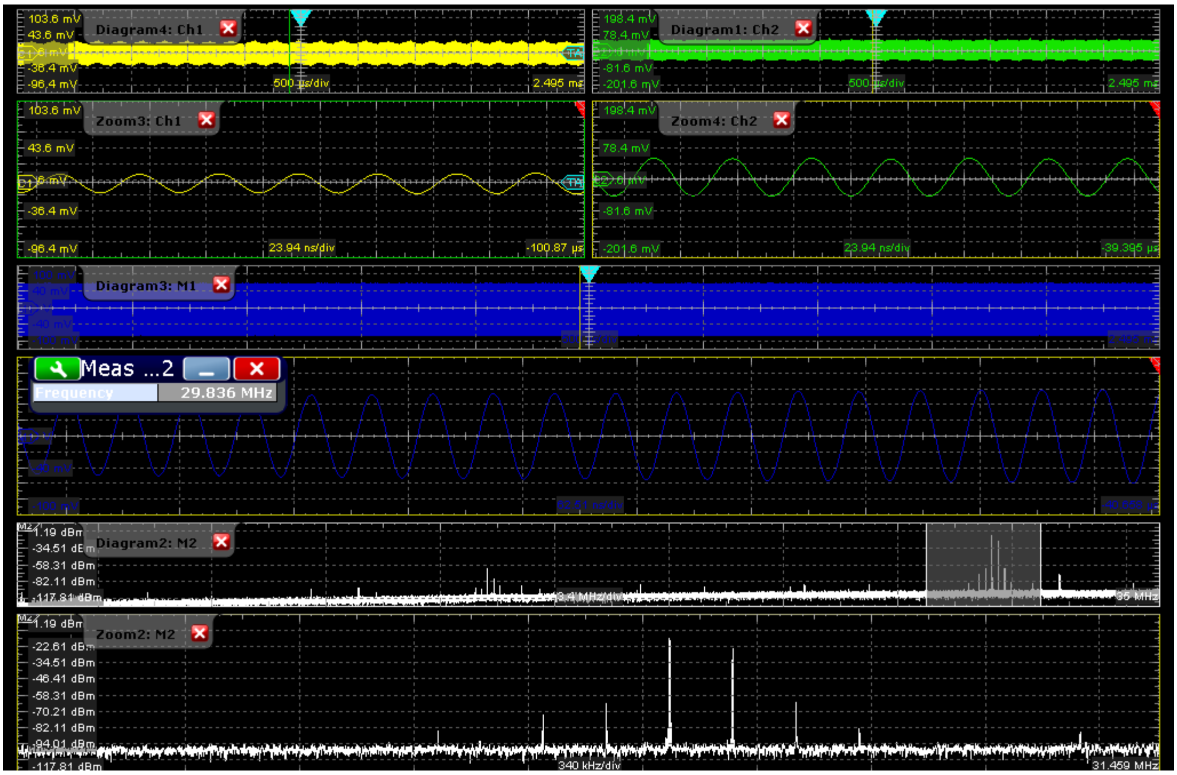
Figure 25: Images response attenuation. Source: Analog devices, 1999 [5].

In order to avoid the negative effect of the filter, the output signal is measured just after the DAC, that is, before the filter. The functional block diagram of the AD9834, included in a previous section (figure 10), shows that there are two current outputs: IOUT and IOUTB. Both signals are connected to the oscilloscope and the actual output signal is calculated as the difference between them. The frequency is changed from 1 MHz to 30 MHz resulting in different sinusoidal signals at the corresponding frequency and a correct reading of the frequency in the display. Some output signals at different frequencies are shown in the following figures. The signals at the top of the image represent the measured signals in both current outputs (IOUT and IOUTB), and the blue one the final output signal. The spectrum is also represented.



Meas ...2
Frequency 9.9804 MHz

Figure 26: Output signal at 10 MHZ. Source: Own elaboration, 2019



Meas ...2
Frequency 29.836 MHz

Figure 27: Output signal at 30 MHz. Source: Own elaboration, 2019

Accuracy and resolution are some of the reasons why DDS is used for this project. The results, although being much better, they are not completely valid for being used as a local oscillator for RF transmissions. The Spurious-Free Dynamic Range (SFDR) is an important parameter that should be analyzed. It refers to the ratio (dB) between the highest level of the fundamental signal and the highest level of any other harmonic in the spectrum. The spectral purity is significantly low because of the presence of unwanted spurs.

There are several reasons that might influence in the spectral purity of the output. Among them, two sources of unwanted spurs are highlighted: truncation and DAC nonlinearity.

Truncation is a primal source of spectral impurity in the output signal of the DDS. In order to reduce the size of the sine lookup table, the output of the phase accumulator is truncated. This generates a truncation error signal that creates discrete spurs in the frequency domain [10].

The resolution of a DAC is defined by its number of bits. However, even an ideal N-bit DAC will generate harmonics in the output spectrum. This is because there will always be a difference between the theoretical output of the DAC and the actual output signal. This difference results to be not perfectly linear, so the output of the DAC will be distorted [5]. The output will be the expected signal plus harmonics, that are called quantizer Non-Linearity Spur.

6. Conclusions

The aim of the present thesis is to discuss about Direct Digital Synthesis (DDS) and implement a sinewave generator based on this technology. The basics of the topic are introduced, and its theory of operation is described in detail.

The contributions of the project include the construction of a DDS based sinewave generator functional prototype, which can be used for educational purposes, among others. It is programmed to generate a sine wave, whose frequency can be set externally, in a range of 1MHz to 30MHz. As already discussed above, spectral purity gets worse as the frequency increases. Regarding the primary motivation of the project, work as a local oscillator for RF transmissions, some improvements should be made.

The first improvement that should be done is to replace the actual filter, included in the module board, by a better one. The new filter must have higher cut off frequency and slope. The filter will act as an antialiasing filter, attenuating the images response that appear in the spectrum at fclk ± fout. Furthermore, some simulations demonstrate that quantization noise have an important role in the quality of the signal. In order to diminish it, accuracy should be increased. It depends on the clock frequency and the number of bits of the phase accumulator. In order to increase the accuracy, the DDS device should be changed looking for a higher clock frequency. The resolution of the AD9834 is calculated as follows:

$$\frac{f_{clk}}{2^n} = \frac{75\ MHz}{2^{28}} = 0.28\ Hz$$

7. References

[1] B. Cronin, «DDS Devices Generate High Quality Waveforms Simply, Efficiently, and Flexibly,» *Analogue Dialogue,* vol. 46, 2012.

[2] E. Murphy and . C. Slattery , “All about direct digital synthesis,” *Analogue dialogue,* 2004.

[3] «Dynamic Measurements of Phase Lock Loop Transient Response,» *Teledyne LeCroy,* 2013.

[4] P. O’Brien, Curtin y Mike, «Phase-Locked Loops for High Frequency Receivers and Transmitters,» *Analogue Dialogue,* vol. 33, 1999.

[5] K. Gentile y . R. Cushing, *A Technical Tutorial on Digital Signal Synthesis,* Analogue devices, Inc., Tech. rep, 1999.

[6] Datasheet, "AD9834," 2014. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD9834.pdf.

[7] «Fundamentals of direct digital synthesis,» *Analogue devices, Inc., Tech. rep,* 2017.

[8] Datasheet, "ATmega328P," 2015. Available: https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf.

[9] Datasheet, "MAX7219," 2003.

Available: https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf.

[10] K. R. A. Patel, «DIRECT DIGITAL SYNTHESIS AND SPUR REDUCTION USING METHOD OF DITHERING,» University of Illinois Urbana-Champaign, 2012.

[11] S. Cheppali, «Arduino Boards-Pin mapping,» icircuit, 30 September 2014.

Available: https://icircuit.net/arduino-boards-pin-mapping/141.

[12] C. J. a. L. L. Y. YuanWang, «A Novel DDS Array with Low Phase Noise and spurs,» Institute of Electrical and Electronics Engineers, Chengdu, 2011.

[13] «Arduino MAX7219 7 segment display module example,» 2019.

Available: http://arduinolearning.com/code/arduino-max7219-7-segment-display-module-example.php.

[14] D. A. Greensted, «Switch Debouncing,» 2010. Available: http://www.labbookpages.co.uk/electronics/debounce.html.

[15] Banggood, «MAX7219 Red 8 Bit Digital Tube LED Pantalla Módulo para Arduino MCU,» 2019.

Available:https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjfnbeQprzjAhWREBQKHQB4CSIQjhx6BAgBEAM&url=https%3A%2F%2Fes.banggood.com%2FMAX7219-Red-8-Bit-Digital-Tube-LED-Display-Module-For-Arduino-MCU-p-907849.html&psig=AOvVaw2kLI67NrnOWHif.

[16] Banggood, «AD9834 DC 5V DDS Signal Generator Module Board Sine/Triangle/Square Wave Waveform Controllability,» 2019.

Available: https://www.banggood.com/AD9834-DC-5V-DDS-Signal-Generator-Module-Board-SineTriangleSquare-Wave-Waveform-Controllability-p-1340632.html?cur_warehouse=CN.