



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO GENERADOR DE
SECUENCIAS DIGITALES PARA EL EQUIPO DREAMCATCHER

AUTOR

EDUARDO JAVIER MERCHÁN TOALOMBO

AÑO

2018



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO GENERADOR DE
SECUENCIAS DIGITALES PARA EL EQUIPO DREAMCATCHER

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Redes y
Telecomunicaciones

Profesor Guía

Mgs. Iván Ricardo Sánchez Salazar

Autor

Eduardo Javier Merchán Toalombo

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido el trabajo, “Diseño e implementación de un prototipo generador de secuencias digitales para el equipo dreamcatcher”, a través de reuniones periódicas con el estudiante Eduardo Javier Merchán Toalombo, en el semestre 2018-2, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Iván Ricardo Sánchez Salazar
Magister en Calidad Seguridad y Ambiente
C.C.: 1803456142

DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, “Diseño e implementación de un prototipo generador de secuencias digitales para el equipo dreamcatcher”, del estudiante Eduardo Javier Merchán Toalombo, en el semestre 2018-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Pablo Geovanny Palacios Játiva
Magister en Ingeniería de Redes de Comunicaciones
C.C.: 0927864454

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

Eduardo Javier Merchán Toalombo

C.C.: 1715226104

AGRADECIMIENTOS

Agradezco a Dios, por permitirme culminar un gran objetivo en mi vida, agradezco a mi gran familia, en especial a mi madre Elvia y mi abuelita Margarita por su incondicional amor, apoyo y motivación. A mis amigos del con los que curse la carrera por su constante ánimo. Y a mí tutor guía Iván por sus valiosos consejos y recomendaciones para la realización de este proyecto.

DEDICATORIA

A mi abuelita Margarita, una luz en mi camino, a mi madre Elvia por todo su infinito amor, dedicación y gran apoyo que nunca ha mermado. A mis queridos hermanos por todo su ánimo, apoyo, consejos constantes que hicieron posible alcanzar esta meta.

RESUMEN

El presente trabajo de investigación tiene como objetivo diseñar e implementar un prototipo que genere secuencias digitales, señal que se ingresará en un módulo de comunicaciones digitales fabricado por la empresa china DreamCatcher, mismo que fue adquirido por la UDLA, para su uso en prácticas de laboratorio de telecomunicaciones.

En el desarrollo de este prototipo se utilizan software, hardware y equipos para medición y accesorios para la implementación y desarrollo de las pruebas respectivas. Los equipos son los siguientes:

- Hardware: Plataforma Arduino Uno R3, módulo DreamCatcher Me1100
- Software: Visual Basic 6.0, Arduino Monitor Serie
- Equipos: Osciloscopio Keysight MSOX3014T, generador de señales
- Accesorios: Cables coaxial SMA, cable USB

En el diseño e implementación de este proyecto se emplean los métodos de estudio inductivo y experimental, ya que se basa en la investigación aplicada a la generación de secuencias binarias. La parte inductiva se refiere a la investigación de las características de la plataforma, el software y hardware necesario, todo esto en conjunto con la tarjeta DreamCatcher RF se usarán en el estudio de secuencias. Con el método experimental se hace referencia al diseño del prototipo y banco de pruebas.

ABSTRACT

The objective of this research work is to design and implement a prototype that generates digital sequences, a signal that entered to digital communications module manufactured by the Chinese company DreamCatcher, which acquired by the UDLA, for use in laboratory practices of telecommunications.

In the development of this prototype software, hardware and equipment for measurement and accessories used for the implementation and development of the respective tests. The teams are the following:

- Hardware: Arduino Uno R3 platform, DreamCatcher Me1100 module
- Software: Visual Basic 6.0, Arduino Monitor Series
- Equipment: Keysight Oscilloscope MSOX3014T, signal generator
- Accessories: SMA coaxial cables, USB cable

In the design and implementation of this project, the methods of inductive and experimental study used, since it is based on research applied to the generation of binary sequences. The inductive part refers to the investigation of the characteristics of the platform, the software and hardware necessary, all this together with the DreamCatcher RF card used in the study of sequences. With experimental method, reference made to the design of the prototype and test bench.

ÍNDICE

1. INTRODUCCIÓN.....	1
1.1 Antecedentes	1
1.2 Objetivo General	1
1.3 Objetivos específicos	1
1.4 Marco teórico	2
1.4.1 Conceptos.....	2
1.4.2 Magnitud analógica.....	2
1.4.3 Magnitud digital.....	2
1.4.4 Nivel alto.....	2
1.4.5 Nivel bajo.....	2
1.4.6 Estado.....	2
1.4.7 Biestables	2
1.4.8 Tiempos de propagación	2
1.5 Sistemas Digitales.....	2
1.5.1 Circuitos lógicos combinacionales	2
1.5.2 Circuitos lógicos secuenciales	3
1.5.3 Cicuitos secuenciales asíncronos.....	4
1.5.4 Cicuitos secuenciales síncronos.....	5
1.5.5 Modelo autómata de Moore	6
1.5.6 Modelo autómata de Mealy.....	6
1.6 Sistemas Secuenciales.....	7
1.7 Aplicaciones con sistemas secuenciales.....	8
1.7.1 Contadores y divisores de frecuencia	8
1.7.2 Registro de desplazamiento	9
1.7.3 Máquinas de estados finitos	9
1.7.4 VHDL	10
1.7.5 PRBS	10
1.7.6 Relación con el proyecto.....	10

2. DESCRIPCIÓN DEL HARDWARE Y SOFTWARE A UTILIZAR	10
2.1 DreamCatcher.....	10
2.1.1 Módulo ME1100.....	11
2.1.2 Especificaciones técnicas.....	12
2.2 Plataforma Arduino	13
2.2.1 Arduino Uno	14
2.2.2 Microcontrolador Atmel ATmega328.....	15
2.3 Análisis FODA.....	17
2.3.1 Analisis de software a utilizar	17
2.3.2 Matlab	17
2.3.3 Visual Basic 6.0.....	18
3. DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO	20
3.1 Descripción del prototipo	20
3.1.1 Diagrama de bloques del prototipo.....	22
3.1.2 Diseño de la interfaz de usuario	24
3.1.2.1 Bloque de caracterización de palabras digitales	26
3.1.2.2 Tipo de envío.....	26
3.1.2.3 Tipo de dato.....	26
3.1.2.4 Número de palabras	26
3.1.2.5 Número de bits	27
3.1.2.6 Tiempo de intervalo de cada bit.....	27
3.1.2.7 Bloque de ingreso por teclado	27
3.1.2.8 Bloque de transmisión	27
3.1.3 Formato de caracterización de palabras digitales	27
3.1.4 Diseño del Hardware	29
3.1.4.1 Descripción de la fase de comunicación serial UART o USART.	30
3.1.4.2 Descripción de la fase del microcontrolador	32
3.1.4.3 Descripción de la fase del puerto Out.....	32

4. PRUEBAS Y RESULTADOS.....	32
4.1 Conexión con GUI desarrollada en Visual Basic 6.0	32
4.2 Filtros del programa.....	33
4.2.1 Pruebas de filtros para envíos de palabras digitales en modo único.....	34
4.2.2 Pruebas de filtros para envíos de palabras digitales en modo continuo.....	37
4.3 Ejemplo de aplicación de la GUI diseñada	38
4.4 Ejemplo de aplicación con monitor serie de Arduino.....	42
4.5 Prueba de envío de palabra única de 4 bits con el módulo ME1100.....	45
5. CONCLUSIONES Y RECOMENDACIONES	49
5.1 Conclusiones.....	49
5.2 Recomendaciones	49
REFERENCIAS	51
ANEXOS	53

1. INTRODUCCIÓN

La utilización de secuencias digitales tiene gran relevancia en el campo de la electrónica y de las telecomunicaciones en formato digital, en sistemas de control y en general en cualquier proceso donde intervenga la información digitalizada y su posterior tratamiento y procesamiento.

1.1 Antecedentes

Observar la naturaleza y tratar de entender cómo funcionan sus fenómenos han llevado al hombre a buscar la forma de replicarlos, es así como se da la necesidad de implementar laboratorios, para poder simular un ambiente cada vez más parecido al medio en el que vivimos y en el que se desarrollan los distintos fenómenos físicos, como es el caso de las telecomunicaciones.

Hay varias aplicaciones como contadores, registros, máquinas de estados, etc., con respecto a estas últimas hay dos tipos bien conocidos: las máquinas de Mealy y la de Moore, que fueron la base para la creación de la criptografía en las telecomunicaciones por los años 50.

Actualmente las secuencias digitales se usan para encriptación en tecnologías celulares como por ejemplo PCM.

1.2 Objetivo General

Diseño e implementación de un prototipo de generación de secuencias digitales binarias, basado en la plataforma Arduino, para su uso con la tarjeta DreamCatcher RF, diseñado con una interfaz gráfica que permita al usuario interactuar con el prototipo.

1.3 Objetivos específicos

- Realizar un estudio técnico de la plataforma Arduino.
- Diseñar un prototipo generador de secuencias digitales.
- Desarrollar una interfaz gráfica que permita al usuario interactuar con el prototipo generador de secuencias digitales.
- Realizar pruebas de validación.

1.4 Marco Teórico

1.4.1 Conceptos

1.4.2 Magnitud analógica. – “Utiliza valores continuos. Es el caso que aparece en la mayoría de los fenómenos naturales. Un ejemplo de esto es la temperatura” (Valdivia, C., 2005, p.10).

1.4.3 Magnitud digital. – “Utiliza valores discretos, de forma que se pueden procesar y almacenar. Para poder procesar una señal análoga hay que muestrear en un intervalo de tiempo” (Valdivia, C., 2005, p.10).

1.4.4 Nivel alto. – “Es o corresponde al nivel de tensión más alto y se le representa por el uno “1” lógico” (Valdivia, C., 2005, p.10).

1.4.5 Nivel bajo. – “Corresponde al nivel de tensión más bajo y se representa por cero “0” lógico” (Valdivia, C., 2005, p.10).

1.4.6 Estado. – “Un circuito secuencial puede estar en determinado momento en un estado o en otro dependiendo del valor que tenga almacenado en los elementos que constituyen su memoria” (Muñoz, J. 2012, p.136).

1.4.7 Biestables. - “Son circuitos que manejan o poseen dos estados que se mantiene estables, por lo que pueden tener 1 bit almacenado” (Muñoz, J. 2012, p.136).

1.4.8 Tiempos de propagación. - “Como en las compuertas lógicas, es un intervalo de tiempo en el que se actualizan las salidas de los biestables luego de sucedido variaciones en las entradas” (Muñoz, J. 2012, p.146).

1.5 Sistemas Digitales

Los sistemas digitales se dividen en dos grandes ramas: los circuitos lógicos combinacionales y los circuitos lógicos secuenciales.

1.5.1 Circuitos lógicos combinacionales. - se caracterizan por que el estado o variables de sus salidas dependen exclusivamente de los estados o variables de las combinaciones de sus entradas. (Floyd, T.L., 2009).

De ahí deriva su nombre y se subdividen en:

- Codificadores
- Decodificadores
- Multiplexores
- Demultiplexores
- Sumadores
- Restadores
- Rom



Figura 1. Diagrama de circuito combinacional

1.5.2 Circuitos lógicos secuenciales. - se caracterizan por que el estado o sus variables de salida depende además de sus estados o variable de entrada, de un estado anterior. (Morris, Mano., 2010). Se subdividen en dos grupos principales: sincrónicos y asincrónicos. Dentro de los mismos podemos citar los siguientes:

- Flip Flops
- Registros
- Contadores
- Memorias

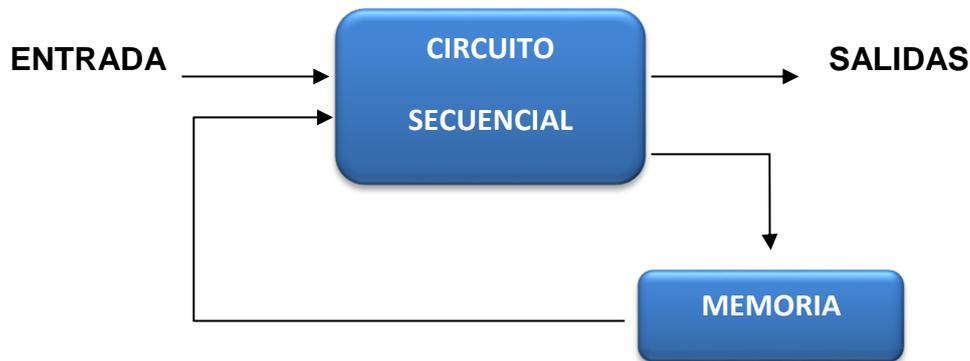


Figura 2. Diagrama de circuito secuencial

En el anterior gráfico se puede notar que en los circuitos combinacionales forman parte de los circuitos secuenciales, con la variación de que presentan una especie de retroalimentación hacia elementos de memoria, los mismos que pueden almacenar datos y retroalimentar la entrada de tal manera que las salidas no son únicamente afectadas por las entradas, sino que las salidas son también afectadas por los estados previos que han quedado guardados en el sistema y se han retroalimentado. Los circuitos secuenciales que como ya lo mencionamos anteriormente se dividen en dos grupos:

- Asíncronos: no dependen de una señal de reloj
- Síncronos: dependen de una señal de reloj

1.5.3 Circuitos secuenciales asíncronos

Los circuitos secuenciales asíncronos en sí no poseen elementos de memoria, ya que utilizan sus propios tiempos de retardo o también llamados de propagación que se presentan en sus compuertas lógicas o circuitos integrados. Esta forma de funcionar presenta ciertos problemas en cómo opera el circuito, ya que el diseñador no tiene bajo control los retardos de propagación. Se destaca entre sus características principales él no permitir que

sus variables de entrada varíen simultáneamente, a fin de prevenir la formación del fenómeno conocido como de carreras críticas en las entradas. En la figura 3 se ve el diagrama.

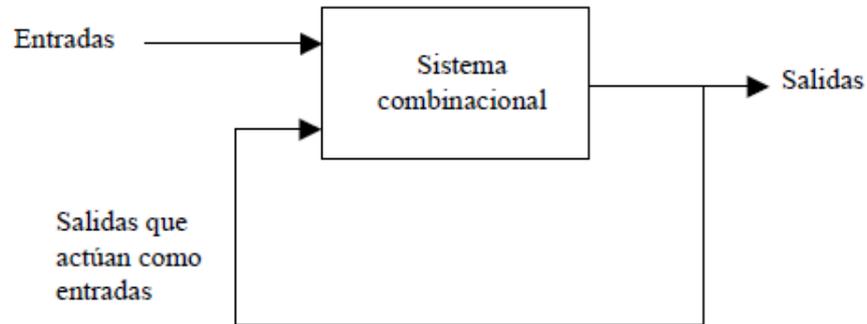


Figura 3. Diagrama de circuito secuencial asíncrono

Tomado de (Roth, C., 2015, p. 75)

1.5.4 Circuitos secuenciales síncronos

Es circuitos se caracterizan por que todo el sistema combinacional cambia de estado a la vez que cambia de flanco la señal de reloj externa. En el siguiente gráfico, figura 4 se presenta un diagrama del funcionamiento descrito.

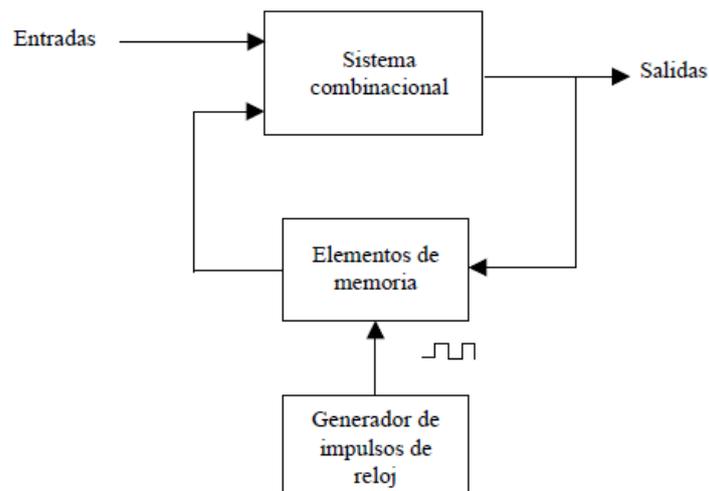


Figura 4. Diagrama de circuito secuencial síncrono.

Tomado de (Roth, C., 2015, p. 67)

Hay dos modelos bien conocidos circuitos secuenciales síncronos, llamados autómatas finitos, que son los siguientes:

- Modelo de Autómata de Moore
- Modelo de Autómata de Mealy

1.5.5 Modelo Autómata de Moore

Se caracterizan por que sus salidas dependen de los estados anteriores del sistema y sus entradas son utilizadas para provocar una evolución en los estados. (Wakerly, 2010). En la figura 5, se presenta un diagrama de bloques del modelo de Moore.

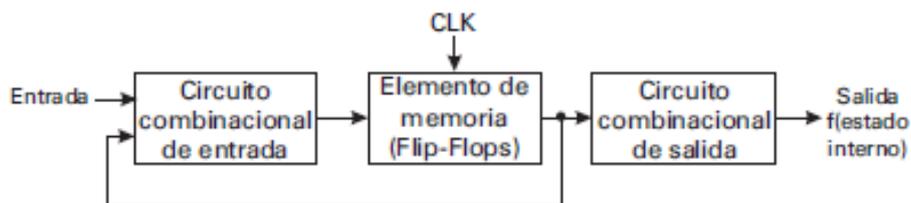


Figura 5. Diagrama de bloques del modelo del autómata de Moore
Tomado de (Valdivia, C., 2005, p. 56).

1.5.6 Modelo Autómata de Mealy

Este modelo se caracteriza porque la salida depende tanto valores adquiridos por las entradas como del estado anterior del sistema. La figura 6 muestra la representación en un diagrama de bloques para entender mejor el funcionamiento, y las etapas que lo constituyen con sus respectivas entrada, salida y realimentación. (Parera, A., 1991)

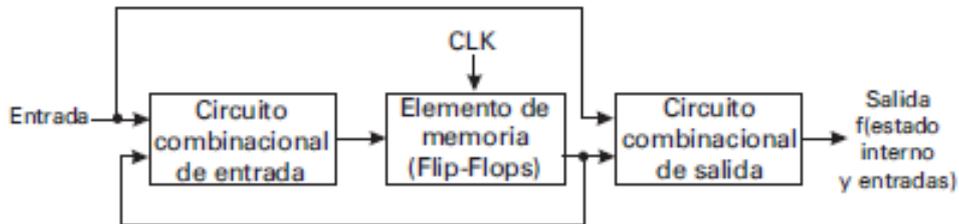


Figura 6. Diagrama de bloques del modelo del autómata de Mealy
Tomado de (Valdivia C., 2005, p. 45)

1.6 Sistemas secuenciales

La lógica secuencial es una forma de diseño de sistema binario que emplea una o más entradas y una o más salidas, cuyos estados están relacionados por reglas definidas que dependen, en parte, de estados previos. Cada una de las entradas y salidas puede alcanzar cualquiera de los dos estados: lógica 0 (baja) o lógica 1 (alta). Los sistemas secuenciales tienen memoria, característica que hace posible que el sistema reaccione a una secuencia de entradas. En este particular caso no es de utilidad la memoria, porque se fija en un valor de uno (1) y no se puede hacer que regrese a un valor de cero (0). De todas maneras, se resalta la función de memoria a través de retroalimentar las salidas. Lo mencionado se lo puede observar en la figura 7, en el que se representa con un diagrama de estados. (Hermosa, A., 2004)

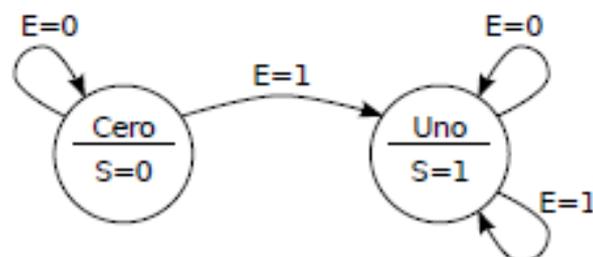


Figura 7. Diagrama de estados
Tomado de (Muñoz, J., 2012, p. 136)

1.7 Aplicaciones con sistemas secuenciales

En los sistemas secuenciales se emplean muchos arreglos de biestables (flip-flops) y circuitos lógicos para poder diseñar otros dispositivos tales como: contadores, divisores de frecuencia, registros de desplazamiento, pbrs, fpga, fsm, etc. (Pallás, R., 1993).

1.7.1 Contadores y divisores de frecuencia

Un contador está formado por un arreglo de biestables, que progresa de estado en estado, en respuesta a un suceso, donde un suceso puede ser un ciclo de señal de reloj. El número de estados que atraviesa el contador antes de regresar al estado de inicio se llama Módulo (m), un contador con m estados se llama un contador con módulo m , los dichos se muestra en la figura 8.

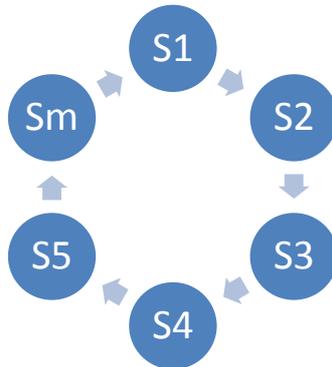


Figura 8. Diagrama de estados de un contador

Probablemente el contador binario de n bits es el más utilizado, el cual requiere de n biestables para su construcción, y 2^n estados. La secuencia será: $0, 1, 2, \dots, 2^n - 1, 0, 1, \dots$, tal contador puede ser diseñado de modo que pasen o no por todos sus posibles estados. (Zetina, A. y Zetina, A., 2004).

Se puede afirmar que todo contador de módulo m , es un divisor de frecuencia f/m , donde f es la frecuencia de reloj y $m=2^n$ es el número de estados.

Por ejemplo, un contador $m=16$ podría denominarse asimismo contador divisor por 16.

1.7.2 Registro de desplazamiento

Un FF puede almacenar, guardar o recordar; es decir, registrar un bit. Entonces, un arreglo de estos dispositivos es lo que se conoce como registro, en el que se puede escribir una palabra y del que se puede leer la palabra almacenada cuando se requiera. Es así, que los registros son el papel de anotaciones en el que se pueden escribir palabras binarias para futuras referencias y consultas.

La mayor parte de las operaciones en una computadora u otros procesadores digitales no son más que transferencias de palabras de un registro a otro. Una característica útil que se puede incorporar a los registros es la posibilidad de que estos pueden realizar operaciones de desplazamiento de la información almacenada en ellos hacia la izquierda o hacia la derecha, característica que los ha denominado registros de desplazamiento.

Los registros de desplazamiento son registros de n bits que tienen la característica de desplazar sus bits almacenados. Un arreglo de biestables es la configuración lógica de un registro de desplazamiento el cual constituye un sistema sincrónico, ya que cada biestable se activa mediante la misma señal de reloj.

1.7.3 Máquinas de estados finitos

Este tipo de máquinas, nombradas comúnmente como autómatas finitos o de simplemente de estados; hacen que se pueda diseñar sistemas secuenciales de mayor complejidad, dotándoles de una propiedad o característica de toma de decisiones, que depende tanto del estado en que se encuentre el circuito como de qué valor exista en cada una de sus entradas. Los dos modelos de

máquinas de estados finitas son las de Moore y la de Mealy, mismas que ya se describió anteriormente.

1.7.4 VHDL

Es un lenguaje de descripción de circuitos electrónicos que se encuentra constituido por módulos los cuales son identificados a través del proceso de reserva una palabra única de este lenguaje. En cada uno de sus módulos hay un conjunto ordenado y consecutivo de instrucciones que hacen posible que se solucione los sistemas digitales, a través de los procesos de detallar, comprender y tomar decisiones. (Maxinez, D. 2012)

1.7.5 Secuencia binaria pseudoaleatoria (PRBS)

Una PRBS, es un flujo de bits matemáticamente aleatorio para que la información sea balanceada. Se emplea no solo en pruebas funcionales digitales puras sino también en pruebas de señal mixta como datos integrados en una señal de modulación (Advantest Corporation., 2013)

1.7.6 Relación con el proyecto

Los sistemas secuenciales al generar señales binarias digitales poseen una gran importancia para el presente proyecto, ya que estas señales con las que se necesita para inyectar en los módulos 1100 y 1110 de DreamCatcher que son los que realizan el procesamiento de las señales digitales, como es el caso de la modulación I/Q.

2. DESCRIPCIÓN DEL HARDWARE Y SOFTWARE A UTILIZAR

2.1 DreamCatcher

DreamCatcher es una empresa que se fundó en 2002 para proporcionar capacitación técnica a la industria eléctrica y electrónica. Con un crecimiento rentable y un equipo altamente experimentado de profesionales técnicos, desde entonces se ha expandido a otras industrias como Telecomunicaciones,

Petróleo y Gas, y Ciencias. En especial lo que nos interesa de esta empresa son los kits de enseñanza didáctica que fábrica para cursos de pregrado universitaria como el Kit ME1100 orientado a estudiantes de segundo y tercer nivel. Este paquete incluye un módulo de baja y alta frecuencia para modulación I/Q, presentaciones en power point y guías de laboratorios para realizar prácticas de modulación digital básica.

2.1.1 Módulo ME1100

La figura 9 a continuación muestra el diagrama de bloques del kit de entrenamiento ME1100 que se utiliza en las prácticas de laboratorio de comunicaciones digitales de RF, en el cual se aprecia sus principales componentes claramente.

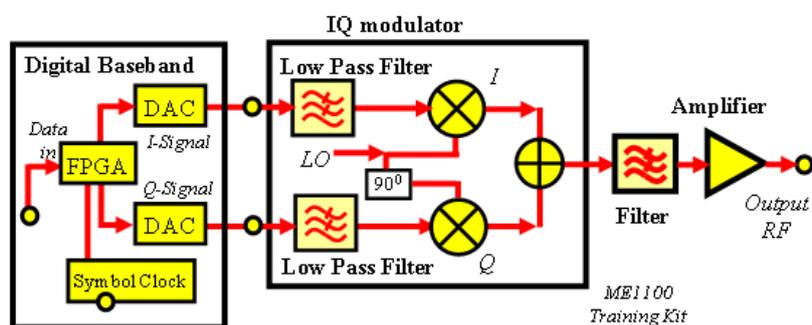


Figura 9. Diagrama de Bloques DreamCatcher ME1100

Tomado de (Slides teaching, 2017, p. 4)

Se puede ver 4 bloques que componen el módulo, como son: la banda base digital, modulador IQ, filtro y amplificador. El bloque banda base está compuesto por 2 conversores digital / análogo (DAC), el símbolo de reloj, un FPGA a través del cual se ingresa la señal de entrada (Datos). A la salida de los DAC se tiene la señal I y la señal Q, que ingresaran respectivamente al siguiente bloque el modular I/Q, en donde pasan por filtros pasa bajos.

Las señales que salen del filtro pasa bajos ingresan a unos mezcladores de señales mismos que están conectados a un oscilador local que se encarga de desfasar las señales I/Q 90° una respecto de la otra, a su salida la dos señales son sumadas obteniéndose una sola que pasa nuevamente por otro filtro pasa bajos y finalmente es amplificada para ser transmitida por la salida RF.

La figura 10 a continuación muestra la distribución de las distintas etapas que componen el módulo de entrenamiento ME1100, identificado en la placa misma.

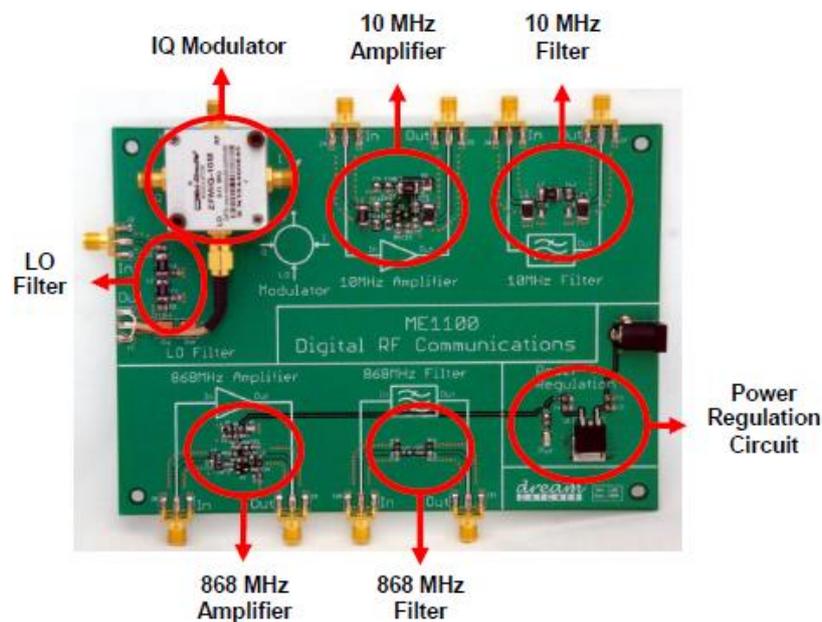


Figura 10. Módulo DreamCatcher ME1100.

Tomado de (Slides teaching, 2017, p. 11)

2.1.2 Especificaciones Técnicas

En la hoja de especificaciones técnicas del fabricante, se detalla las características técnicas del módulo, las mismas son la base para el diseño del generador de secuencias digitales prototipado. En la tabla 1 se resume las principales características eléctricas y mecánicas que posee el módulo ME1100 para comunicaciones digitales RF. Además de las especificaciones técnicas

citadas, el fabricante recomienda usar la plataforma Arduino para realizar las respectivas prácticas en el laboratorio.

Tabla 1.

Características eléctricas - mecánicas

	Módulo Baja-Frecuencia	Módulo Alta-Frecuencia
RF		
IQ modulator conversión loss	< 7.0 dB	
IQ modulator DC offset	< 0.09 mv	
Filtro pasabanda (3dB)	5.4 MHz a 13.3 MHz	794 MHz a 1233 MHz
General		
Voltaje de entrada	4.5 V (min), 5.5 V (max)	
Corriente de entrada	22 mA	
Conectores	12 SMA (f)	

Adaptado de (Dreamcatcher, 2017)

Tomando en cuenta el voltaje de entrada y frecuencia de operación que se puede ver en la tabla 1, se va a utilizar el módulo de baja frecuencia para conectar la placa Arduino.

2.2 Plataforma Arduino

Arduino es una plataforma de electrónica que está basada en hardware y software *open source* fundamentada en una placa de circuito impreso que cabe en la palma de la mano. Posee un microcontrolador *Atmel AVR* (los más usados son *Atmega168*, *Atmega328*, *Atmega1280*, *Atmega8*) y conexiones de entrada y salida, permitiendo interactuar con toda clase de dispositivos como son: *leds*, *displays*, motores eléctricos, servos, sensores infrarrojos, conexiones de red, flashes de cámaras fotográficas, potenciómetros, botones etc. La idea

detrás de Arduino es expandir esas posibilidades de interacción lo máximo posible. (Torrente, 2013, pp. 63-64)

Arduino se puede utilizar para desarrollar prototipos interactivos autónomos o puede ser conectado al software del ordenador como, por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data. (Processing.s.f.)

La placa Arduino, es en la cual se implementará la programación y los diferentes elementos que ayudarán a la comunicación con el módulo dreamcatcher y con el ordenador. Para este proyecto se utilizará la placa Arduino UNO R3 figura 11.

2.2.1 Arduino Uno



Figura 11. Arduino UNO R3
Tomado de (Arduino, 2017)

El Arduino Uno básicamente es una placa electrónica que difiere de muchas otras por no necesitar un chip conversor de puerto USB a puerto serial, pero en su lugar posee el ATmega16U2 programado ya como conversor de USB a

serie. También tiene una serie de conexiones para facilitar la comunicación ya sea con un ordenador, otra placa Arduino u otros microcontroladores. (Di Justo, P. y Gertz, E., 2013).

2.2.2 Microcontrolador ATMEL ATmega328

ATMEL es la empresa fabricante de una amplia gama de microcontroladores al igual que su principal competencia Microchip, con quienes comparten el primer lugar en el mercado de fabricación y comercialización de este tipo de dispositivos electrónicos de uso masivo en todo el mundo. Una de sus principales características y que marca su diferencia con otros similares es su velocidad de procesamiento. Los microcontroladores Atmega328, utilizan su señal de reloj completa sin hacer divisiones como lo hacen los PIC's, lo que hace que sean mucho más rápidos para ser más precisos en un factor de 4. La distribución de pines se puede observar claramente en la figura 12. (Atmel, 2017).

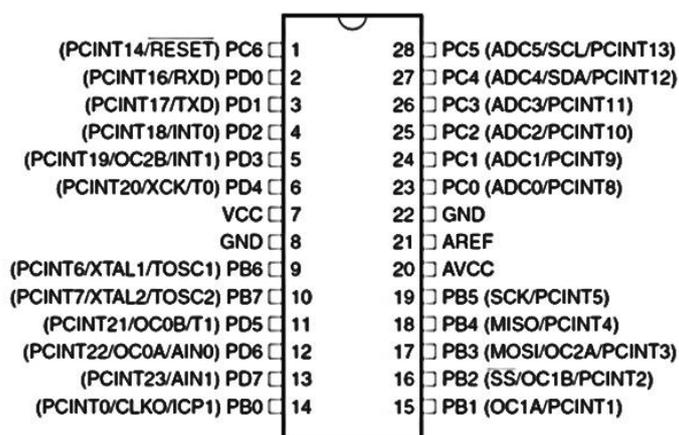


Figura 12. Asignación de pines

Tomado de (Atmel, 2017)

En la figura 12 se puede apreciar el microcontrolador ATmega328, que es la esencia de la plataforma Arduino UNO, tal como la que se observa en la utilizada por el presente proyecto.

El encapsulado del microcontrolador ATmega328, para uso en zócalo se lo puede apreciar en la figura 13.

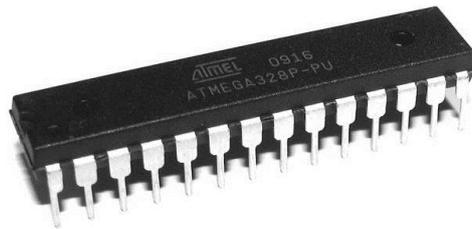


Figura 13. Microcontrolador ATmega328

Tomado de (Atmel, 2017)

El software de Arduino (IDE) incluye un monitor de serie que permite que los datos simples de texto se envíen hacia la placa del Arduino Uno gracias a una biblioteca de Software serial. Las terminales RX, TX y LED (pin 13), parpadean en el tablero cuando se está transmitiendo datos a través del circuito integrado de que convierte la comunicación serial a USB y luego hasta el ordenador. (Barrett, S., 2013).

Se decidió utilizar esta plataforma por la facilidad de programación en el lenguaje C++ de su interfaz y su carga al microcontrolador por medio de un puerto USB. También tiene una mayor compatibilidad con otros tipos de módulos y complementos. Además de tener una asignación de puertos de entrada y salida más simple.

Al ser requerida una constante comunicación entre una PC y el prototipo se ha considerado que el Arduino UNO es el adecuado para el proyecto al no tener que agregar o implementar una rutina de comunicación entre el microcontrolador y el módulo al que se envía la señal digital. En la tabla 2 se muestra las características mecánicas y eléctricas de la plataforma Arduino.

Tabla 2.

Resumen características eléctricas y mecánicas del Arduino UNO

Microcontrolador	Atmega328
Tensión de funcionamiento	5V
Pines digitales I / O	14 (6 pines proveen PWM)
Pines de entrada analógica	6
Corriente DC en Pin 3.3V	50mA
Memoria Flash	32 KB (0.5 KB utiliza el boot manager)
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Adaptado de (Arduino, 2017)

2.3 Análisis FODA

El análisis FODA (Fortalezas, Oportunidades, Debilidades, Amenazas), es una herramienta que permite conformar un cuadro de la situación actual del objeto de estudio (persona, empresa u organización, etc) permitiendo de esta manera obtener un diagnóstico preciso que permite, en función de ello, tomar decisiones acordes con los objetivos y políticas formulados.

2.3.1 Análisis Software a utilizar

Se procederá a realizar un análisis FODA del software matlab y visual basic 6.0 para elegir el que se utilizará para diseñar la interface para generar las secuencias digitales.

2.3.2 Matlab

Es un entorno cuya característica más relevante es una matriz en la que se basa de ahí que derive su nombre, que hace que se sea muy interactivo y pueda resolver complejos, especialmente en el campo del cálculo numérico problemas numéricos en mucho menos tiempo de lo que le lleva a otros lenguajes de programación. Entre sus principales fortalezas están:

- Programación sencilla.

- Biblioteca de matemática amplia
- Abundantes herramientas gráficas
- Continuidad entre valores enteros, reales y complejos

Dado que en el ámbito educativo universitario ha tenido gran acogida para la enseñanza en el campo de las matemáticas avanzadas, algunos de sus usos típicos son los siguientes:

- Cálculos matemáticos
- Desarrollo de sus algoritmos
- Adquisición de datos
- Modelación y simulado
- Análisis de datos, exploración y visualización
- Gráficos científicos y de ingeniería
- Desarrollo de aplicaciones, incluyendo interfaz gráfica de usuario

2.3.3 Visual Basic 6.0

Visual Basic es un lenguaje de programación de Microsoft que utiliza un entorno gráfico para poder visualizar el programa que se está desarrollando, ahorrando una gran cantidad de tareas al no tener que escribir o digitar código. Entre las principales fortalezas de este lenguaje destacan las siguientes:

- Facilidad de su uso en plataforma Windows
- Posee soporte de empaquetamiento y distribución
- Gran cantidad de librerías DLL
- Ambiente de trabajo visual IDE, de fácil manejo
- Creación de archivos ejecutables (.exe)
- Facilidad de integración con otros programas de Microsoft
- Generación de datos automáticamente
- Utilización de formularios
- Utiliza ventanas en su entorno gráfico
- Asistentes para corrección de errores

Estas son algunas de las muchas características que posee la versión 6.0 de Visual Basic., que posteriormente fue sucedido por Visual.Net.

La aplicación de este lenguaje abarca una gran variedad de campos, entre los cuales se cita los siguientes:

- Ingeniería de control
- Proyectos electrónicos con microcontroladores
- Integración con bases de datos
- Automatización y Domótica
- Redes Scada
- Comunicaciones digitales

Este lenguaje también posee ciertas debilidades como son:

- Deficiencia en la programación orientada a objetos.
- Requieren adaptaciones para instalar en sistemas operativos actuales.

Para la evaluación de la matriz FODA se procederá a elaborar una tabla de ponderación para elegir el software más adecuado en base a las características revisadas. En la tabla 3 se muestra lo planteado.

Tabla 3.

Tabla de ponderación

Detalle	Calificación
Excelente	4
Muy bueno	3
Bueno	2
malo	1

Para el análisis FODA del software que se muestra en la tabla 4, se utilizó la tabla de ponderación (tabla 3) para facilitar el análisis comparativo.

Tabla 4.

Análisis comparativo de software

Matriz FODA	Características prototipo	VB6	Matlab
Fortalezas	Lectura de puertos seriales	4	4
	Manejo ventanas Windows	4	3
	Creación de objetos	4	4
Oportunidades	Empaquetamiento	4	3
	Creación de ejecutables	4	4
	Editor IDE	4	3
Amenazas	Accesibilidad de adquisición	4	4
	Seguridad	4	3
	Administración de posible BD	4	3
Debilidades	Importación de librerías	4	4
	Aplicación en otros SO	3	4
	Soporte de fabricante	4	4
		47	43

Al realizar el análisis de comparación de los dos softwares visto en la tabla 4, se puede apreciar que el lenguaje de programación que se procede a elegir es el Visual Basic 6.0, ya que se puede apreciar que el editor IDE es más más amigable que el que tiene Matlab, también a nivel de manejo de ventanas e importación de librerías es mucho más rápido y fácil realizarlo en Visual Basic.

3. DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

3.1 Descripción del prototipo

El prototipo de generación de secuencias digitales está basado en el diseño de secuencias de un número determinado de palabras digitales conformadas a su

vez por un número determinado de bits, los cuales se observarán en la pantalla de un osciloscopio digital.

Ya establecidos los parámetros principales a obtener, se procede a definir el número de palabras y número de bits que las conforman, para enviarlos en las secuencias digitales caracterizadas previamente. Como se va a utilizar el lenguaje de programación Visual Basic 6.0, se toma en cuenta el tipo y tamaño de fuente, el número de palabras y el número de bits que pueden ser fácilmente visualizadas por el ojo humano desde un monitor de laptop de 14 pulgadas hasta uno de mayores dimensiones.

Tabla 5.

Criterio de selección de cantidad de palabras y bits

Visualización	Selección
Tipo de fuente	Sans serif
Tamaño de fuente	24
Nro. Palabras	8
Nro. bits	16

Las secuencias caracterizadas con el criterio de la tabla 5, serán enviadas a una placa controladora que es el Arduino Uno, para que sean procesadas y luego se obtenga una señal cuadrática a la salida de uno de sus pines digitales, y esta a su vez ingrese en puerto de entrada de un módulo fabricado por DreamCatcher que para nuestro caso será el ME1100 y finalmente observado en un osciloscopio.

Para describir de mejor manera todo el proceso que involucra la construcción del prototipo de generación de secuencias digitales, se ha dividido este capítulo en tres partes en las que se detallará los diagramas de bloques del prototipo, el diseño de la interfaz de gráfica de usuario, el diseño del hardware.

3.1.1 Diagrama de bloques del prototipo

El prototipo de generación de secuencias digitales contiene cinco bloques que son los siguientes:

1. **Bloque de Caracterización:** este bloque es administrado por el usuario y está conectado directamente al bloque de procesamiento. (Bloque 1)
2. **Bloque de Conexión I:** este bloque es el medio por el que se conectan los bloques de caracterización y de procesamiento. (Bloque 4)
3. **Bloque de Procesamiento:** el bloque de procesamiento es la parte central y en el cual está concentrado todas las conexiones, aquí se conecta directamente el bloque de caracterización y el bloque de pruebas. (Bloque 2)
4. **Bloque de Conexión II:** este bloque se conecta directamente a los bloques de procesamiento y de pruebas. (Bloque 5)
5. **Bloque de pruebas:** este bloque interactúa directamente con el bloque de procesamiento. (Bloque 3)
6. **Bloque de visualización:** este bloque se conecta directamente al bloque de pruebas. (Bloque 6)

En la figura 14, se muestra los bloques que componen el prototipo, junto con una ilustración de los equipos y dispositivos a utilizar y el proceso desde donde interviene el usuario a través de la interfaz gráfica hasta el equipo donde se ingresa la secuencia digital generada.

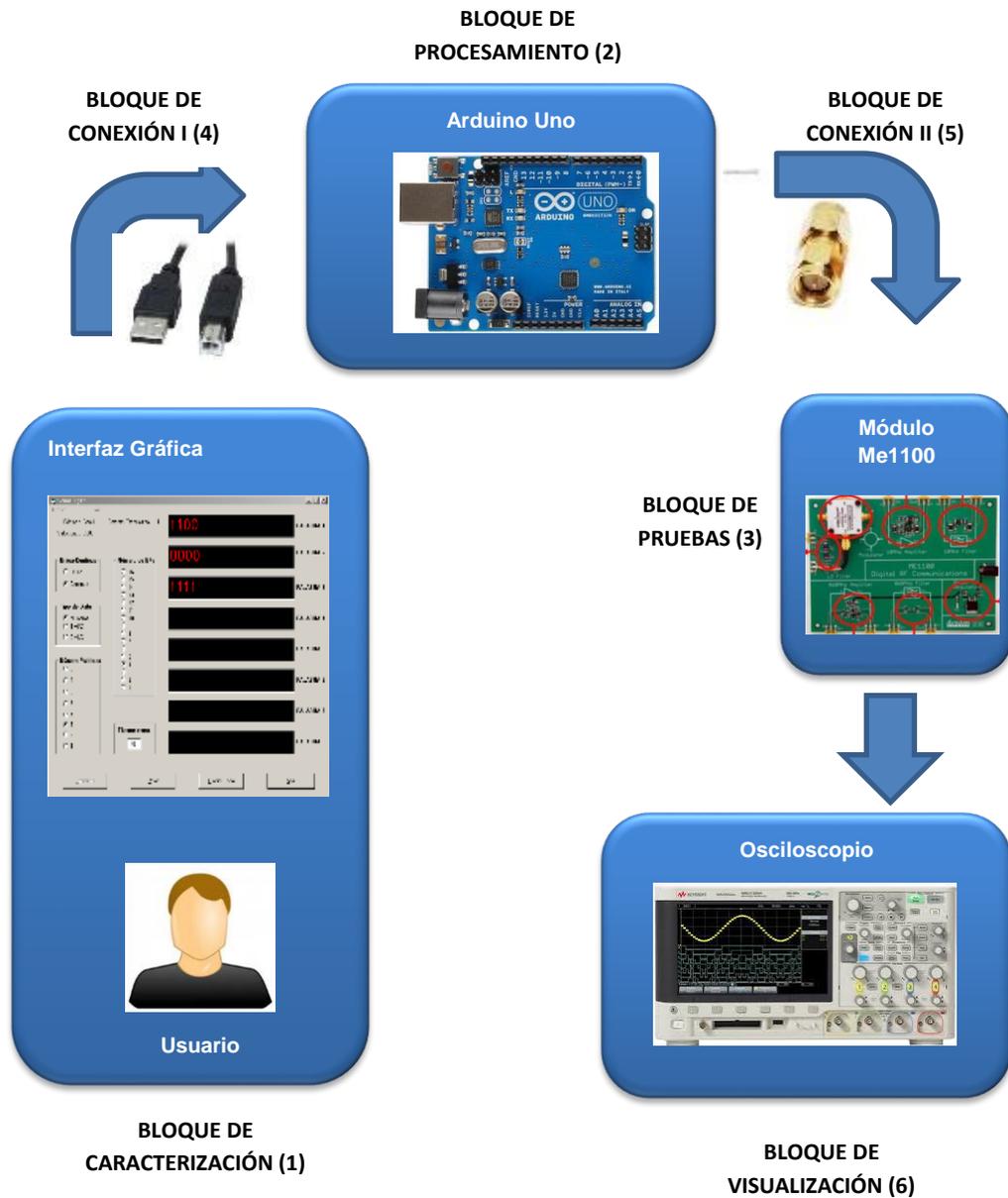


Figura 14. Diagrama de bloques del generador de secuencias digitales.

El Bloque 1 será en el cual el usuario caracteriza las secuencias, a través de la interfaz gráfica instalada en el computador, este a su vez se conecta directamente con el bloque 2, mediante el bloque de conexión I.

El Bloque 2 es el central, ya que está conformado por todo el hardware y está conectado a su vez con el bloque 1, el bloque 3, a través de los bloques adyacentes, el bloque 4 y el bloque 5. Este bloque estará comandado por el bloque 1 que es el cual le envía las secuencias que el usuario desea transmitir.

El Bloque 3 está conformado por los diseños digitales que se deseen probar en este caso el módulo de dreamcatcher que recepta la información se envía al bloque 1 para que la información sea visualizada por el usuario. Este bloque está directamente conectado al bloque 2.

El Bloque 4 es el responsable de la conexión del bloque 1 y el bloque 2 a través del protocolo USB.

El Bloque 5 es el responsable de la conexión del bloque 2 y el bloque 3 a través de un cable coaxial y conector SMA.

El Bloque 6 es el que permite mostrar visualmente las secuencias recibidas desde el bloque de 5 a través de un cable coaxial.

3.1.2 Diseño de la interfaz de usuario

Para realizar nuestro diseño se toma como base el criterio de selección de la tabla 5, sobre la cual vamos a seguir agregando otros elementos que se considera importantes para que el usuario caracterice de mejor manera la o las secuencias que desea enviar.

Esta fase se divide en tres bloques para el diseño de la interfaz que serán de los que se nombra a continuación:

- **Bloque de caracterización:** este bloque esquematiza el formato de las secuencias digitales que estará lista para que usuario ingrese los datos en el bloque de ingreso a través del teclado.
- **Bloque de ingreso por teclado:** este bloque es el encargado de facilitar al usuario el ingreso de las palabras que decida para ser enviadas por el

bloque de transmisión, es el bloque que conecta el bloque de caracterización con el bloque de transmisión.

- **Bloque de transmisión:** este bloque conecta con el bloque de ingreso por teclado, y envía la o las palabras caracterizadas por el usuario previamente.

En la figura 15, se muestra los bloques que componen la interfaz gráfica.

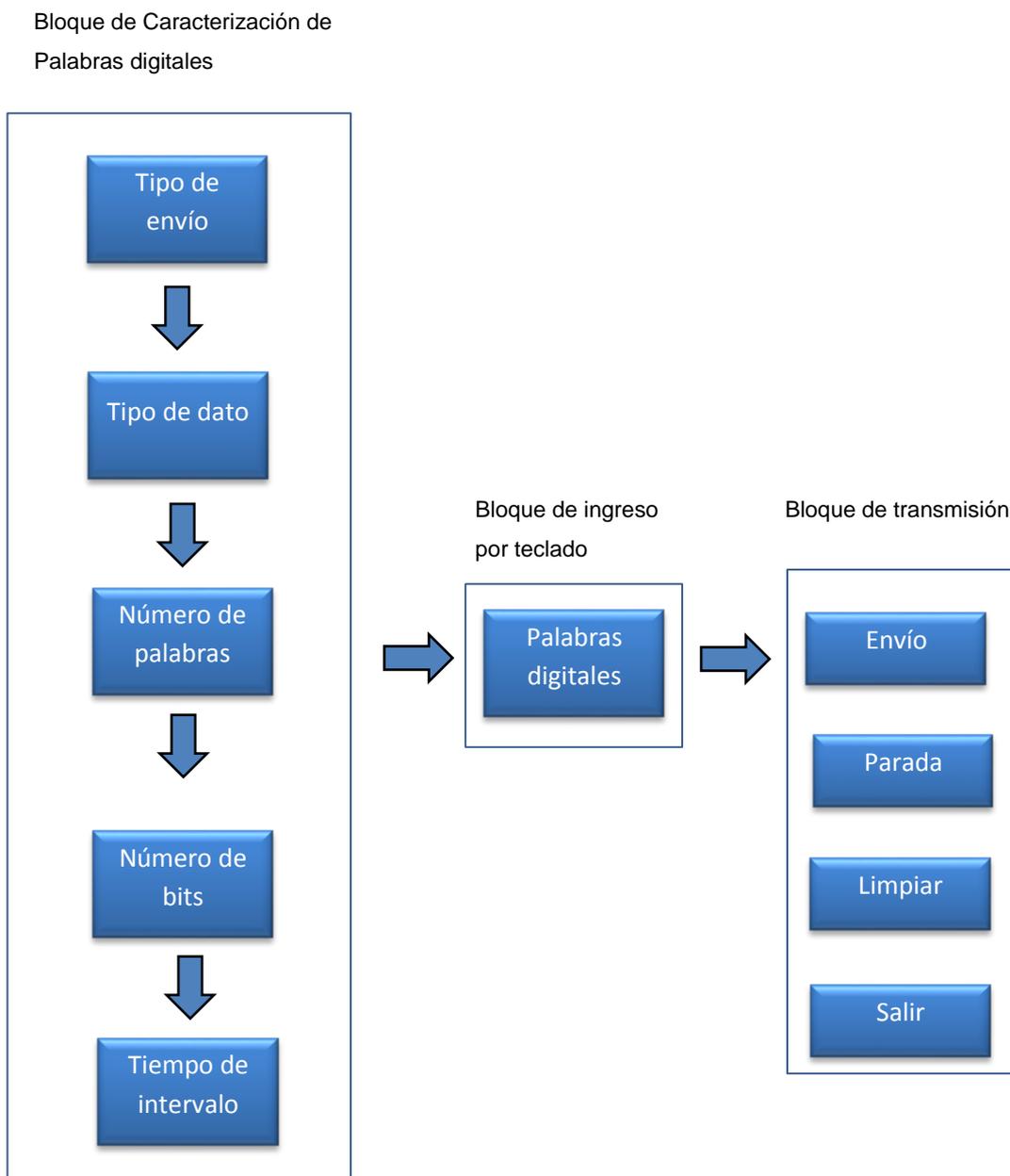


Figura 15. Diagrama de bloques de la interfaz gráfica

3.1.2.1 Bloque de caracterización de palabras digitales

El primer subsistema del generador de secuencias digitales es la interfaz la cual es una aplicación para computadora programada en Visual Basic 6.0, la cual posee un menú gráficamente amigable al usuario, para que pueda seleccionar los parámetros e ingresar las palabras digitales que se quiere obtener en la salida del sistema generador. Mediante el programa visto en el ordenador el usuario podrá escoger en opciones del menú. Las opciones son:

- Tipo de envío (único o continuo).
- Tipo de dato (binario o alfanumérico de máximo 16 bits).
- Numero de palabras, entre 1 y 8.
- Número de bits, entre 1 y 16.
- Tiempo de intervalo de cada bit, entre 00 y 99 ms.

3.1.2.2 Tipo de envío

Hay dos tipos envío que podemos seleccionar al escoger cualquiera de las dos opciones disponibles en la interfaz. Estas opciones son las siguientes:

- **Único.** – En este modo se envía una sola secuencia de la palabra o palabras digitales
- **Continuo.** – En este modo se envía continuamente secuencias de la palabra o pablaras digitales

3.1.2.3 Tipo de dato

Esta opción permite escoger entre caracteres alfanuméricos sean estos números o caracteres 1 ASCII de 8 bits y 2 ASCII de 16 bits en cada caso.

3.1.2.4 Número de palabras

Está opción permite seleccionar el número de palabras a caracterizar entre uno (1) y ocho (8) palabras digitales.

3.1.2.5 Número de bits

Esta opción permite escoger el número de bits de cada palabra a ser caracterizada para el envío, en el diseño hay la posibilidad de elegir entre 1 y 16 bits, tamaño que será para todas las palabras digitales seleccionadas en la opción números de palabras.

3.1.2.6 Tiempo de intervalo de cada bit

En esta opción hay la posibilidad de elegir el tiempo de intervalo de bit a bit, desde cero (0) milisegundos (ms) a 99 milisegundos (ms).

3.1.2.7 Bloque de ingreso por teclado

Esta fase se caracteriza por que se ingresa las palabras digitales por medio del teclado, está confirmado por ocho (8) bloques de texto en los que puede ingresar hasta 16 bits por bloque. Cada bloque está diseñado para no permitir el envío erróneo de algún bit de más o bit de menos en la palabra previamente caracterizada.

3.1.2.8 Bloque de transmisión

Este bloque o fase se caracteriza porque está formado por cuatro botones, los dos primeros se encargan de que controlar la transmisión y parada de las palabras caracterizadas listas para ser enviadas al hardware de nuestro circuito. El tercer botón se encarga de limpiar las cajas de texto que contienen las palabras caracterizadas, para poder ingresar las nuevas palabras que el usuario decida ingresar para sus respectivas pruebas. El cuarto botón se encarga de cerrar la ventana de la interfaz gráfica y salir de nuestra aplicación.

A continuación, se describe con detalle el formato con el que se caracteriza las palabras que conforman las secuencias digitales.

3.1.3 Formato de caracterización de palabras digitales

La disposición en la que está el formato de caracterización de las palabras enviadas se compone de la siguiente manera; empezando por el bit menos

significativo (LSB): 3 bits para indicar el número de palabras digitales a enviar, 4 bits para indicar el número de bits de cada palabra digital, 1 bit que indica el tipo de dato y 1 bit que indica el tipo de envío único o continuo.

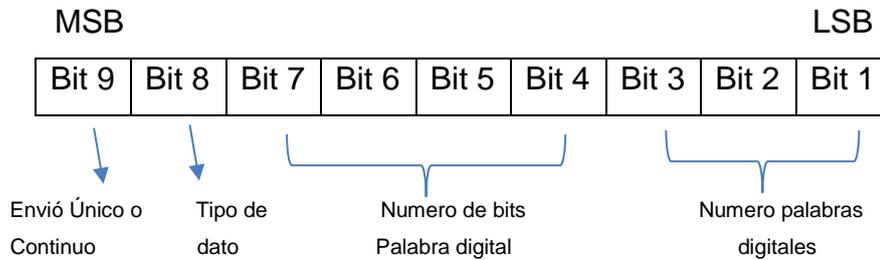


Figura 16. Formato de disposición de bits

El bit 9 corresponde al segundo byte, lo que indica que se envía 2 bytes en total, siendo el bit 9 el menos significativo (LSB) del segundo byte. Para que el segundo byte este completo se pone ceros (0), como se lo representa en la figura 17 que se muestra a continuación.

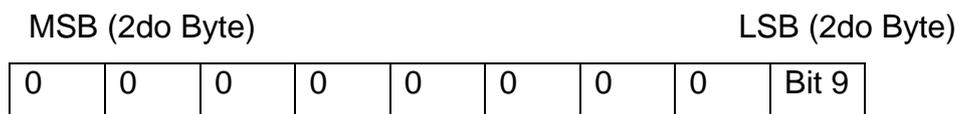


Figura 17. Formato de disposición de bits del 2do byte

En total se realiza 18 envíos de byte con sus respectivos bits de inicio y parada. Estos realizados (18) llevan contenida la información para la generación correcta de las palabras caracterizadas digitalmente que el sistema va a entregar. Los datos que se envían por el puerto serial del ordenador se transmiten a una velocidad de 9600 bps dando un tiempo por bit de:

$$1/9600 = 104 \mu\text{s},$$

A este tiempo se debe adicionar el tiempo de intervalo entre bit y bit de la trama que puede ser entre 1 a 100 μs , dando un tiempo de transmisión máxima de:

$$104 \mu\text{s} + 1 \mu\text{s} = 105 \mu\text{s},$$

Y un tiempo de transmisión mínima de: $104 \mu\text{s} + 100 \mu\text{s} = 204 \mu\text{s}$

3.1.4 Diseño del hardware

En esta parte del diseño de nuestro prototipo se describe los elementos que conforman el hardware, que es la parte central donde se procesa los datos de nuestro generador de secuencias digitales. Para explicar nuestro diseño se divide a esta fase en tres bloques que son los siguientes:

- **Bloque de comunicación:** este bloque se encarga de conectar la interfaz gráfica con el bloque procesador.
- **Bloque procesador:** este bloque se conecta con el bloque de comunicación que receipta los datos de la interfaz gráfica y los procesa. También conecta directamente con el bloque de salida.
- **Bloque de salida:** en este bloque que conecta con el bloque procesador se obtiene la salida de la secuencia digital.

En la figura 18, se muestra los bloques que componen el hardware, donde se poder ver la arquitectura interna en forma general del bloque procesador y los puertos que forman parte de los bloques de comunicación y de salida respectivamente.

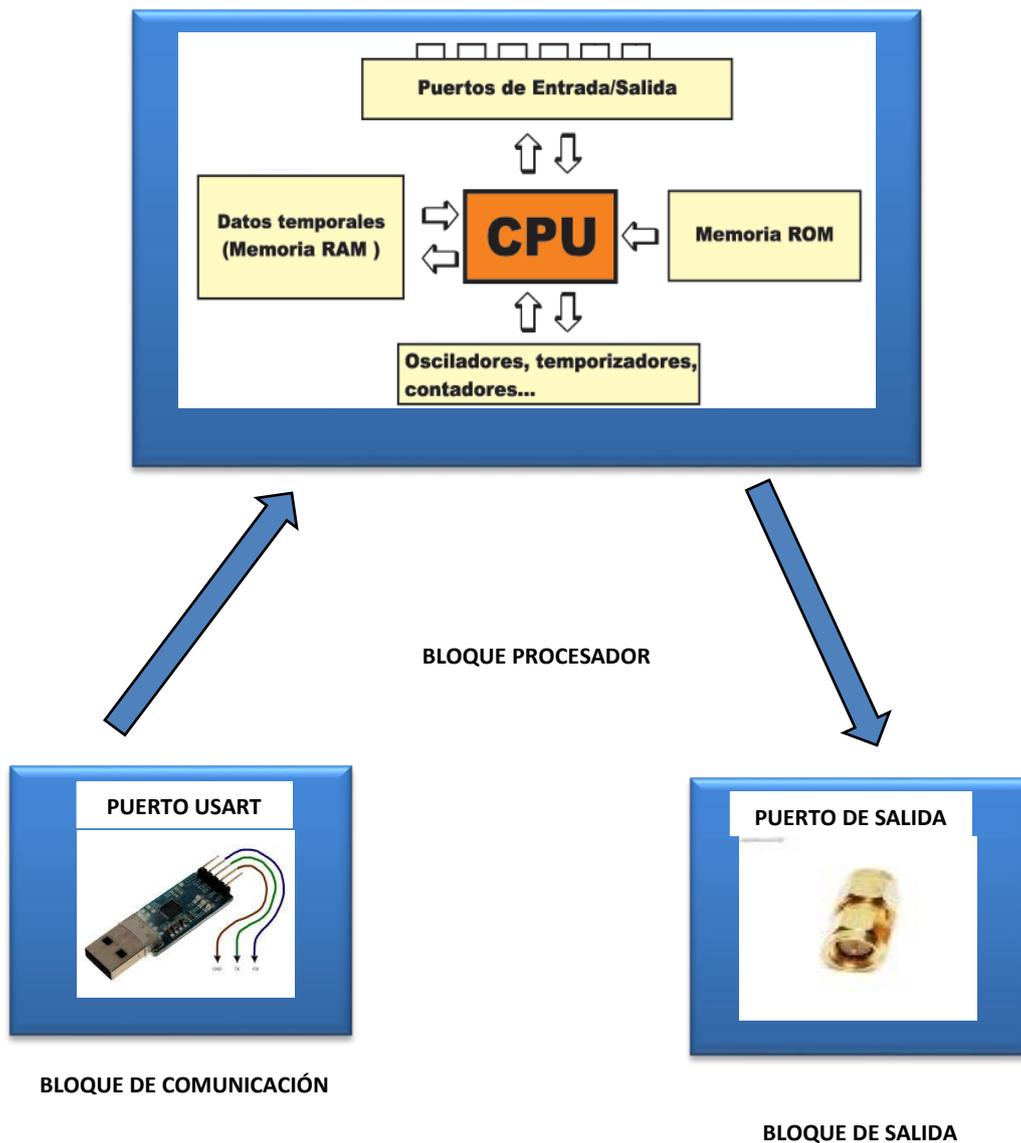


Figura 18. Diagrama de bloques del hardware

3.1.4.1 Descripción de la fase de comunicación serial UART o USART

Esta fase está constituida por el puerto de comunicación serial, que realiza la transmisión y recepción tanto sincrónica como asincrónica, entre el ordenador u otros dispositivos y la plataforma Arduino, debido a estas características se le designa por las siglas UART o USART.

Para realizar la comunicación se usan los pines de recepción Rx y de transmisión Tx, correspondiendo el cero lógico OL a la recepción, y el uno lógico 1L a la transmisión. Estos pines que se mencionaron se comunican con el ordenador a través de su puerto USB que disponen todas las mainboard y muchos dispositivos hoy en día.

Generalmente los niveles de voltaje que se utiliza son de 3.3V o 5V dependiendo de la placa Arduino que se utilice. Para utilizar la comunicación serial Arduino tiene su propio software open source llamado monitor serie que es un entorno integrado de desarrollo IDE, que es muy fácil de utilizar, simplemente con seleccionar la misma velocidad medida en baudios que se muestra en la llamada being(). (Arduino, 2017)

En la figura 19 se aprecia la conexión de la pc o laptop con el Arduino Uno, a través de la interfaz serial USB.

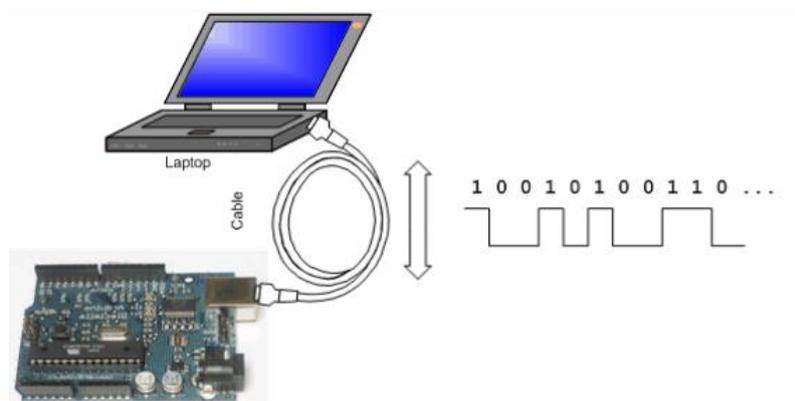


Figura 19. Diagrama del sistema generador de secuencias digitales.

Tomado de (Arduino, 2017)

La velocidad de transferencia por defecto del puerto serial es de 9600 bps o baudios, pero el Arduino Uno soporta las siguientes velocidades: 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1843200 y 3686400, en bits por segundo.

3.1.4.2 Descripción la fase del microcontrolador

Esta fase se caracteriza porque utiliza la parte principal de la placa Arduino Uno, que es el microcontrolador ATmega328, dispositivo del cual ya se habla con detenimiento en el segundo capítulo del presente trabajo. Aquí se almacena la programación previamente cargada desde nuestro computador utilizando la interfaz propia de Arduino, además procesa y ejecuta las palabras digitales recibidas desde la interfaz gráfica de nuestro sistema generador de secuencias que fueron caracterizadas anteriormente.

3.1.4.3 Descripción de la fase del puerto Out

Fase en la que la o las palabras digitales procesadas en la fase del microcontrolador son enviadas y recibidas por el módulo ME1100 de DreamCatcher. El puerto digital de salida que se utiliza en nuestro diseño es el pin 13, mismo al que irá conectado un cable coaxial con un conector SMA macho que se acoplará con el conector SMA hembra del módulo.

4. PRUEBAS Y RESULTADOS

4.1 Conexión con GUI desarrollada en Visual Basic 6.0

Primero verificamos el puerto serial virtual (**COM**), en el que aparece conectado el generador, verificándolo en administrador de dispositivos del S.O. Windows 7. Una vez identificado se procede a configurar el mismo puerto en la interfaz, la velocidad está seteada por defecto en 9600 bps, pero se puede habilitar para cambiar a otra velocidad, si así se lo requiriera.

Una vez configurados estos parámetros, se le da un click con el mouse del ordenador sobre el botón OK y se establece la conexión entre la GUI y la plataforma Arduino. En la figura 20 se muestra el proceso indicado.

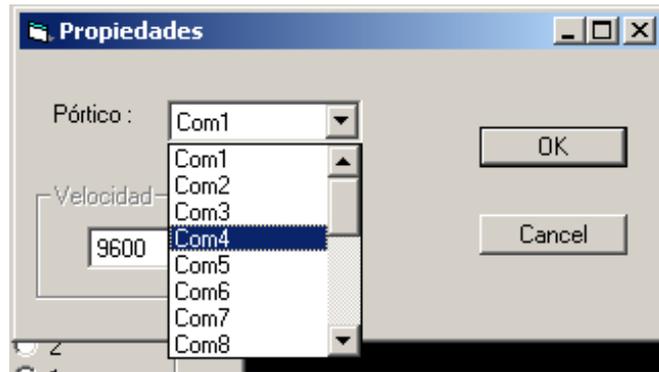


Figura 20. Ventana para configurar parámetros de conexión

4.2 Filtros del programa

Esta prueba sirve para comunicar al usuario si los datos enviados desde el ordenador son correctos y corresponden al número de palabras digitales y número de bits seleccionados previamente, mostrando un mensaje de alerta comunicando el error cometido como se muestra en la figura 21.

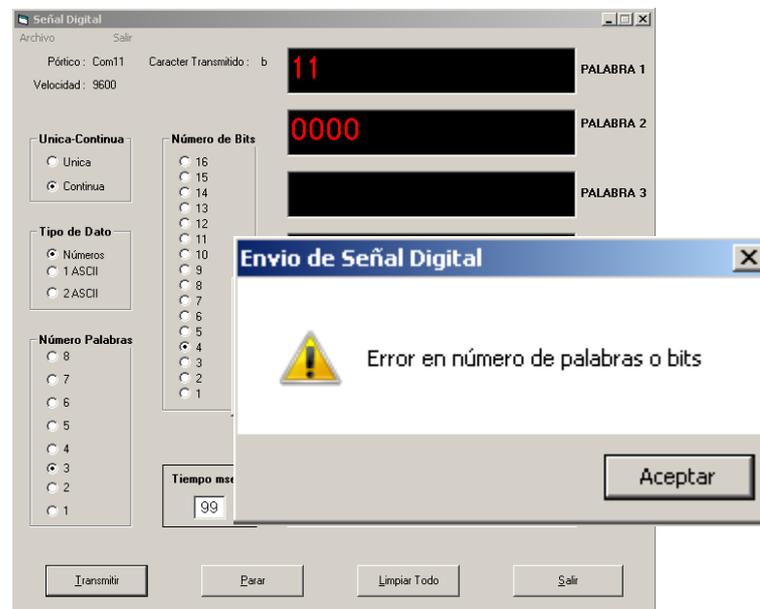


Figura 21. Mensaje de alerta mostrando error en la caracterización

En la figura anterior se puede ver claramente el error cometido al caracterizar la trama a ser enviada, ya que no cumple con los parámetros seleccionados de números de palabra=3, y números de bits=4. En la primera palabra, faltan 2 bits por completar y no existe la tercera palabra que debe ser ingresada. De la misma manera la interfaz muestra a través de un mensaje la correcta caracterización y el pro siguiente envío de datos, tal como la figura 22 muestra.

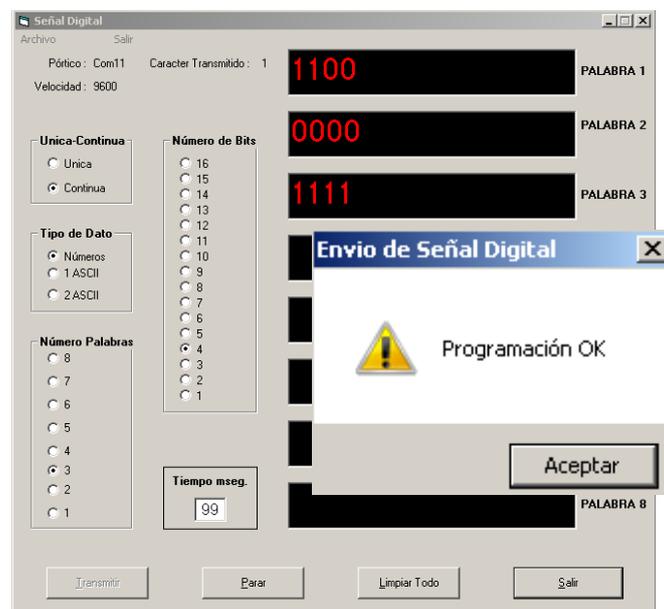


Figura 22. Mensaje de alerta mostrando correcta caracterización y envío de datos

4.2.1 Pruebas de filtros para envíos de palabras digitales en modo único

En la figura 23 se observa el procedimiento para la generación de 2 palabras de 4 bits cada una, que son enviadas seleccionando el tipo de envío Único. Caracterizadas en todas sus fases y visualizadas en las cajas texto de fondo negro y fuente color rojo.

Como se puede apreciar las palabras enviadas son:

Palabra1: 1010

Palabra 2: 1101

Y a la secuencia obtenida en el osciloscopio mostrada e color amarillo es:

10101101

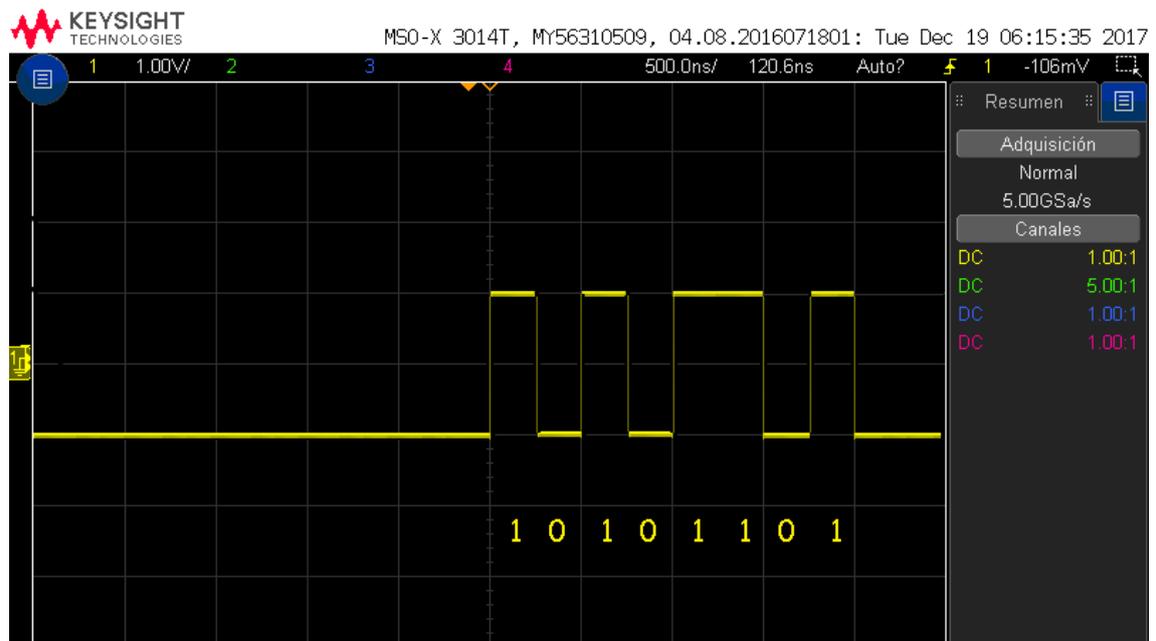
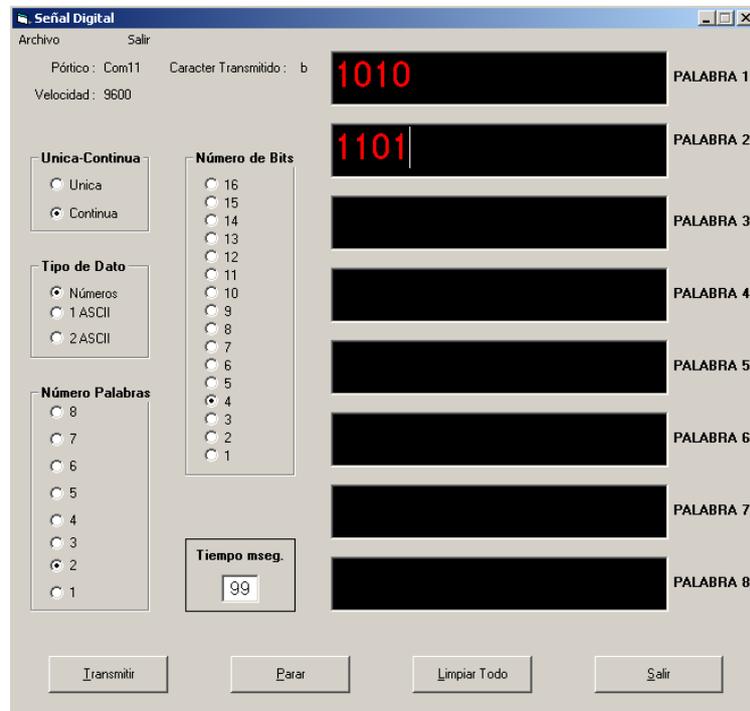


Figura 23. Palabras digitales (2) de 4 bits enviadas en modo único

En la figura 24, se observa a continuación el procedimiento para la generación de 5 palabras de 5 bits cada una, que son enviadas seleccionando el tipo de envío Único.

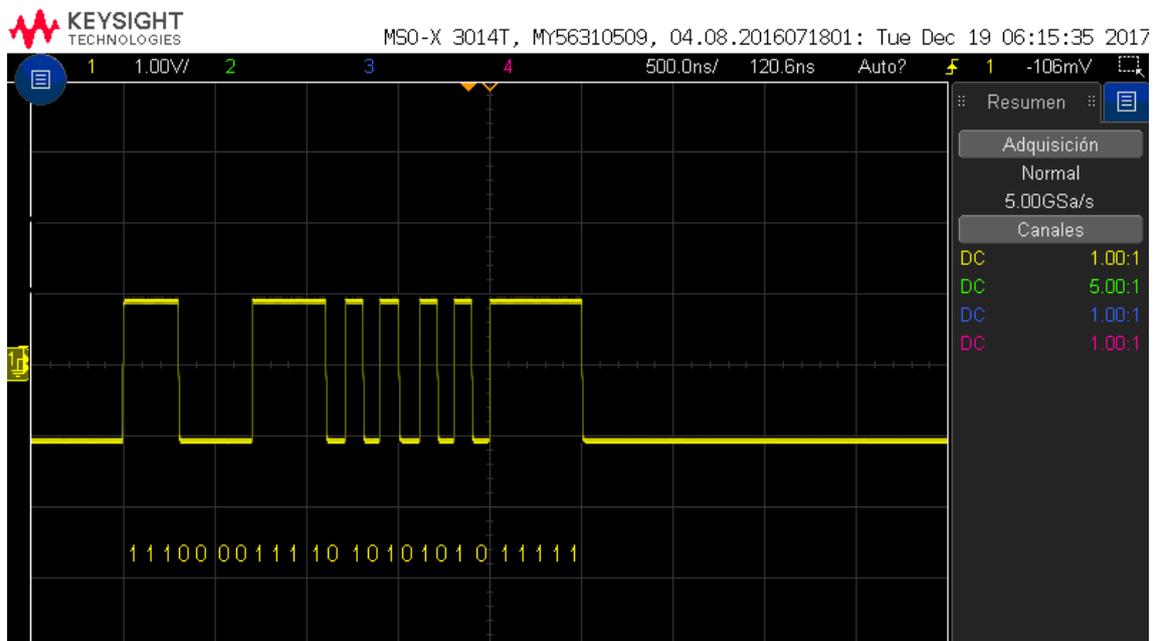
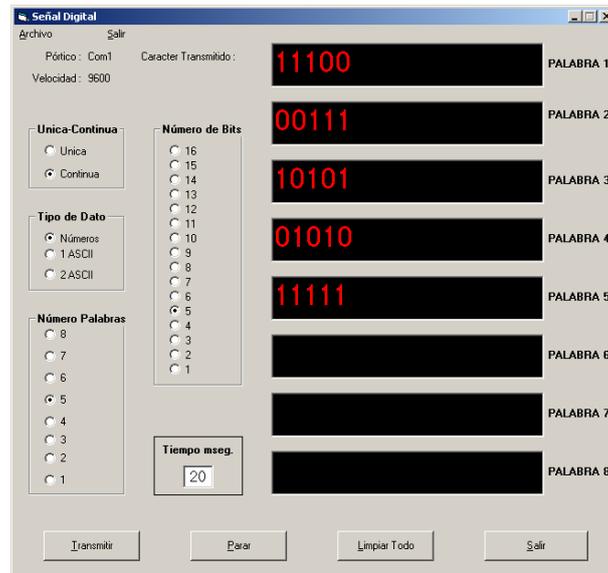


Figura 24. Palabras digitales (5) de 5 bits enviadas en modo único

De las figuras 23 y 24 se puede observar que la forma de onda obtenida en el osciloscopio corresponde a los bits enviados en cada palabra digital.

4.2.2 Pruebas de filtros para envíos de palabras digitales en modo continuo

En la figura 25, se observa el procedimiento para la generación de 3 palabras de 3 bits cada una, que son enviadas seleccionando el tipo de envío continuo.

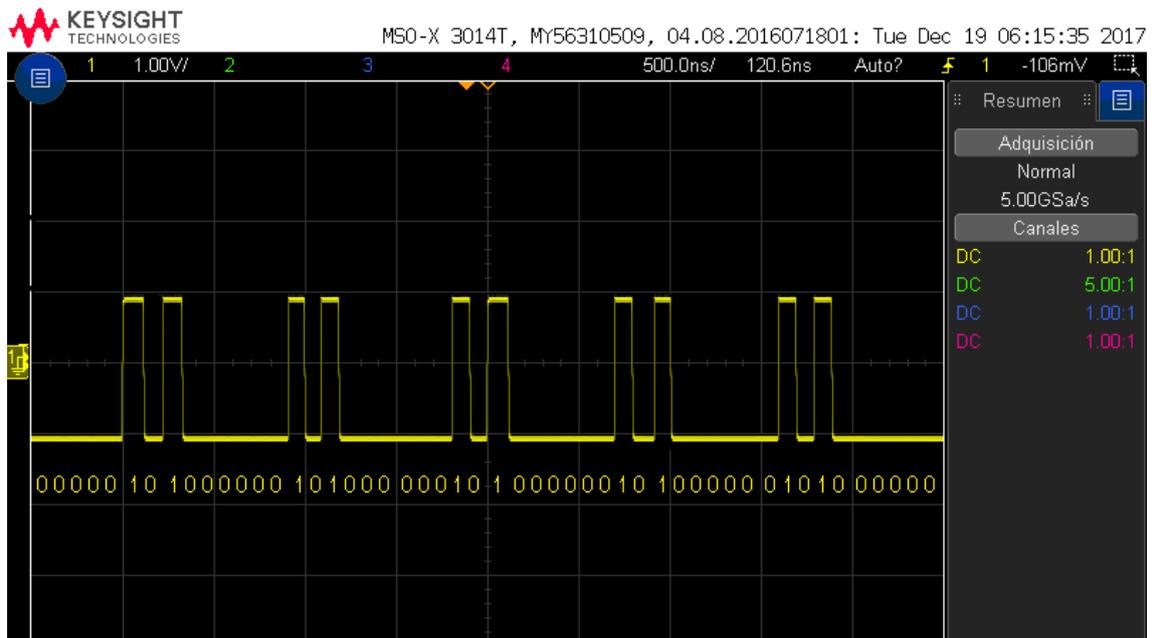
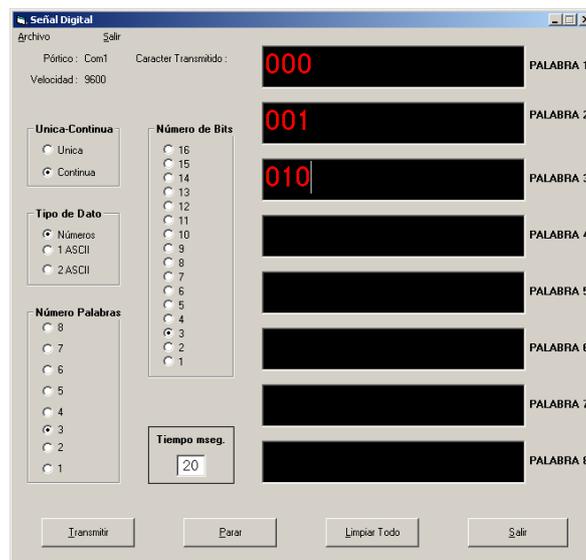


Figura 25. Palabras digitales (3) de 3 bits enviadas en modo continuo

En la figura 26, se observa el procedimiento para la generación de 4 palabras de 2 bits cada una, que son enviadas seleccionando el tipo de envío continuo.

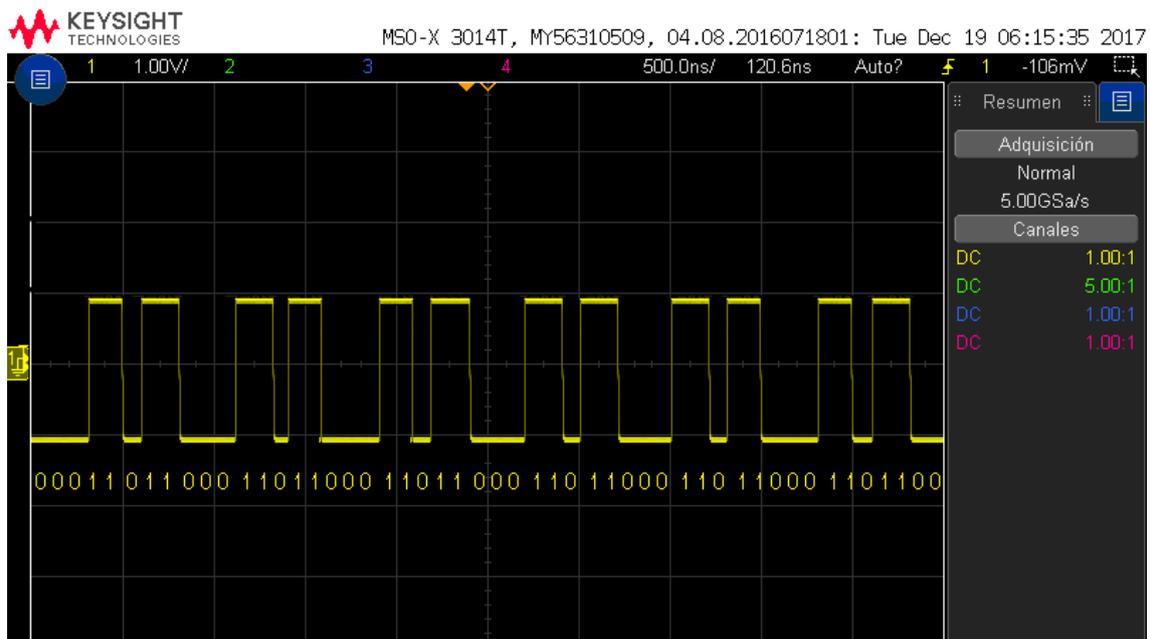
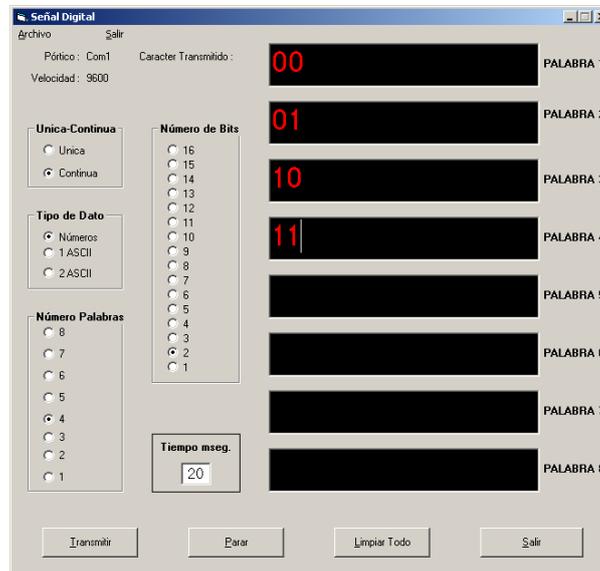


Figura 26. Palabras digitales (4) de 2 bits enviadas en modo continuo

4.3 Ejemplo de aplicación de la GUI diseñada

La interfaz gráfica de usuario es como se presenta a continuación desde donde se caracteriza a la palabra o palabras para ser enviados al generador: Como se

aprecia en la figura 27, la utilización de la interfaz es muy amigable al usuario, la mayoría de las opciones permiten ser seleccionadas con un clic, únicamente la palabra a enviar se debe digitar en cada caja de texto. Para una mejor visualización las cajas de texto se diseñaron con fondo negro y las palabras tienen color rojo para mayor contraste y apreciación.

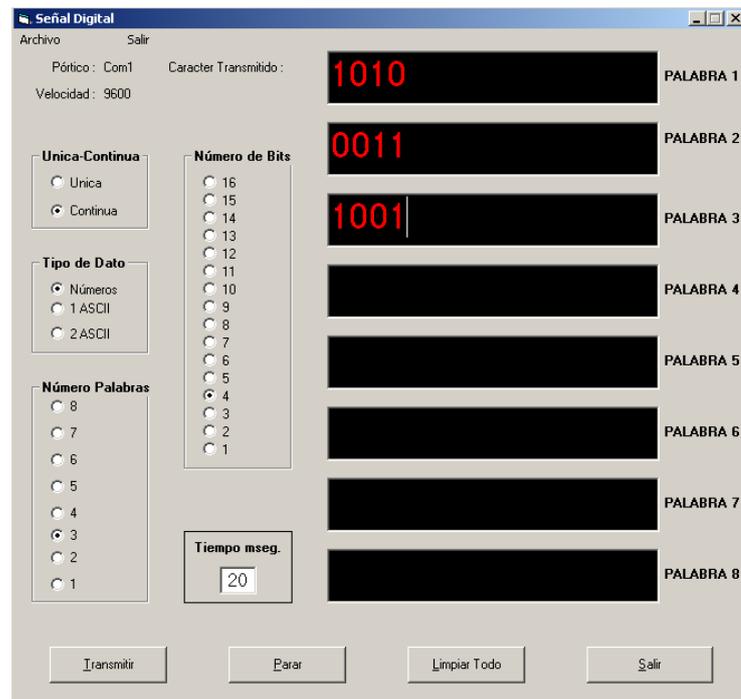


Figura 27. Interfaz GUI del generador de secuencias digitales.

Seleccionados e ingresados los datos caracterizados con los parámetros para las palabras que serán generados por el prototipo, son transmitidos por el puerto USB del ordenador. La transmisión de datos desde la computadora al hardware es bit por bit, donde va contenida la información de las palabras caracterizadas previamente en el menú de la interfaz.

El generador está programado para enviar un total de 8 palabras digitales de 16 bits, haciendo 128 bits en total, que se envían al Arduino en envíos únicos o continuos de 16 bytes entregados por el sistema, información contenida en cada palabra de la secuencia digital. Por el puerto USB del computador los datos son transmitidos a 9600 bps. En lo referente a los niveles de tensión que

se manejan tanto en el puerto USB del computador como en el puerto USB del hardware (Arduino) son iguales a 5V, por lo tanto, no hay inconvenientes ni es necesario usar un conversor de voltajes. Además, la interfaz está diseñada para que una vez seleccionado el número de palabras, solo esas cajas de texto están activas para poder digitar los caracteres, por ejemplo, si selecciono en número de palabras la opción 4, en las cajas de texto solamente se puede escribir en las 3 primeras de nombre: palabra 1, palabra 2 y palabra 3.

De igual manera sucede con el número de bits o el tipo de dato, así, si el usuario ha seleccionado 6 bits, por ejemplo, podrá únicamente digitar 6 bits, si intenta por error digitar un bit mas no se escribirá y se reproducirá un sonido predeterminado de Windows notificando que el usuario ha errado. Con el parámetro tipo de dato, de forma similar ocurre que, si se selecciona números, solo se podrá digitar en binario, ósea unos y ceros (1's y 0's). Así en la figura 28 se muestra la selección de números de palabras y bits.

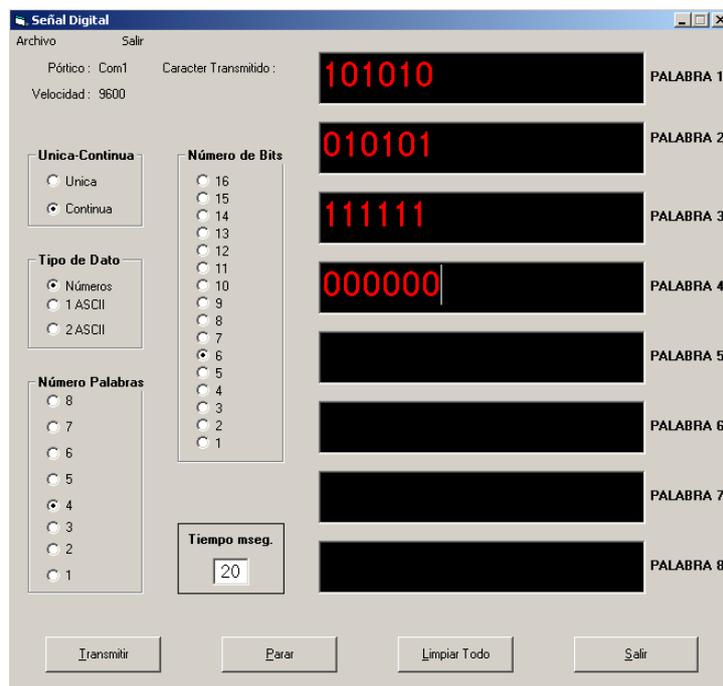


Figura 28. Selección de Numero de palabras y numero de bits.

Si se selecciona 1 ASCII se puede digitar solo un carácter alfanumérico y/o caracteres especiales de 8 bits, que se setea automáticamente por el

programa, al digitar otro carácter no se escribe nada y en su lugar se reproduce un sonido predeterminado de Windows para notificar el error cometido. Al seleccionar 2 ASCII, la interfaz setea automáticamente 16 bits, y se puede digitar hasta 2 caracteres alfanuméricos y/o especiales, cada uno de 8 bits, dando un total de 1 byte a ser transmitido, y si se digita un tercer dígito el programa no acepta el ingreso de ese dígito e inmediatamente reproduce un sonido predeterminado de Windows notificando el error cometido al instante. En la figura 29 y 30, se muestra las selecciones del tipo de dato.

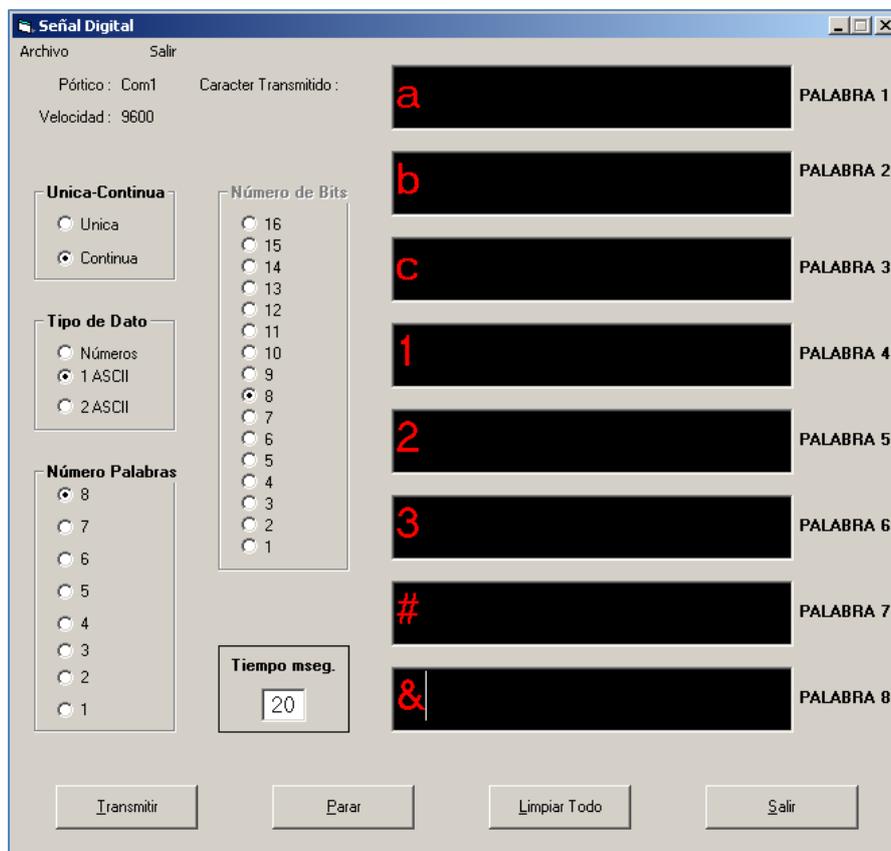


Figura 29. Selección de tipo de dato 1 ASCII.

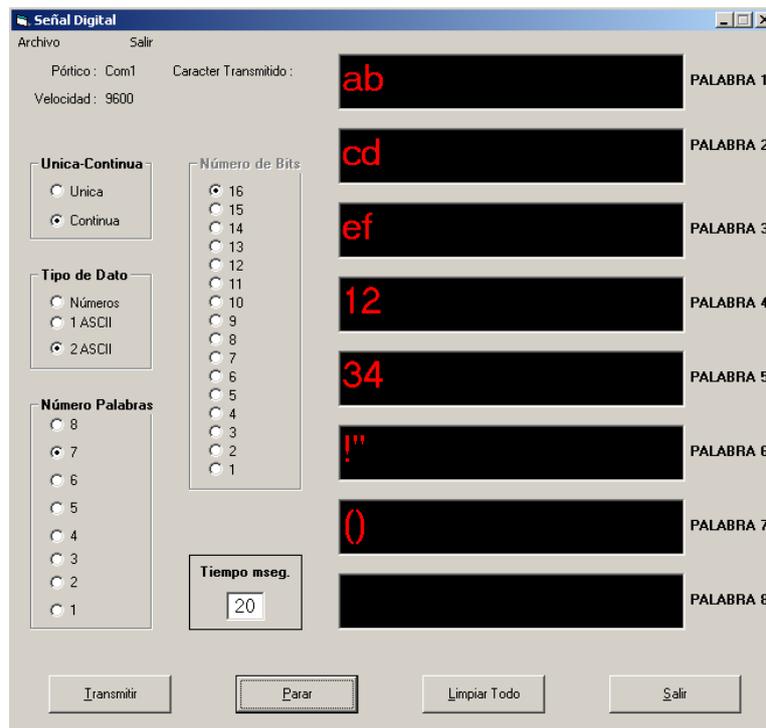


Figura 30. Selección de tipo de dato 2 ASCII

4.4 Ejemplo de aplicación con monitor serie de Arduino

El hardware en su totalidad lo compone la plataforma Arduino UNO, misma que ya describimos anteriormente, es esta parte del subsistema se realiza la programación que receipta la información caracterizada por parte de la interfaz GUI ejecutada desde el ordenador y enviada por medio del puerto USB.

Al igual que la información caracterizada la programación cargada al microcontrolador ATmega328 también es capaz de enviar secuencias digitales utilizando para este cometido el monitor serie mostrado en la figura 31, que se usa para programar, compilar y subir el programa a la plataforma Arduino.

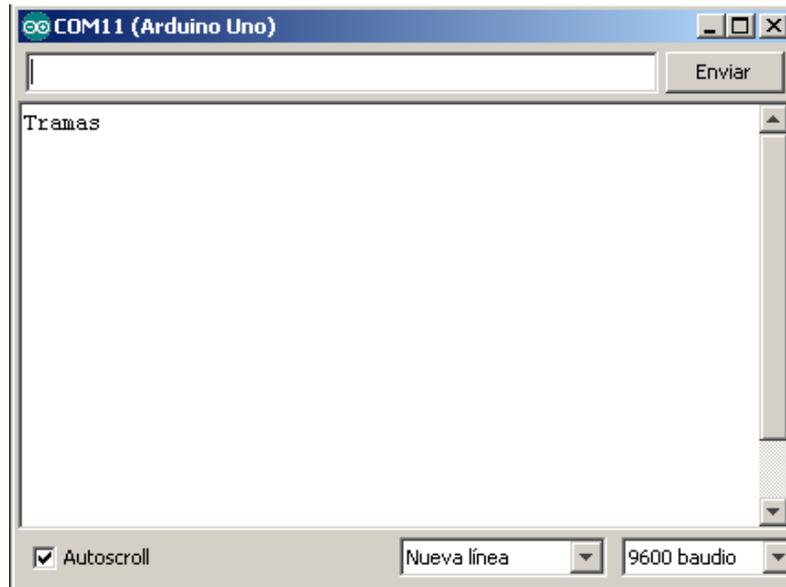


Figura 31. Monitor serie de software Arduino

Para enviar una trama por medio del monitor serie ingresamos por teclado caracteres alfanuméricos de la siguiente manera:

- Enviar/parar datos
 - a → enviar
 - b → parar
- Modo de envío
 - 1 → envío continuo
 - 0 → envío único
- Número de palabras
 - Del 01 al 08
- Número de bits
 - Del 00 al 16
- Tiempo (ms)
 - Del 00 al 99

Así, por ejemplo, si queremos enviar la siguiente secuencia de 3 palabras de 4 bits: 1100, 0011, 1100, se escribe a130420110000111100, tal como se puede ver en la figura 32:

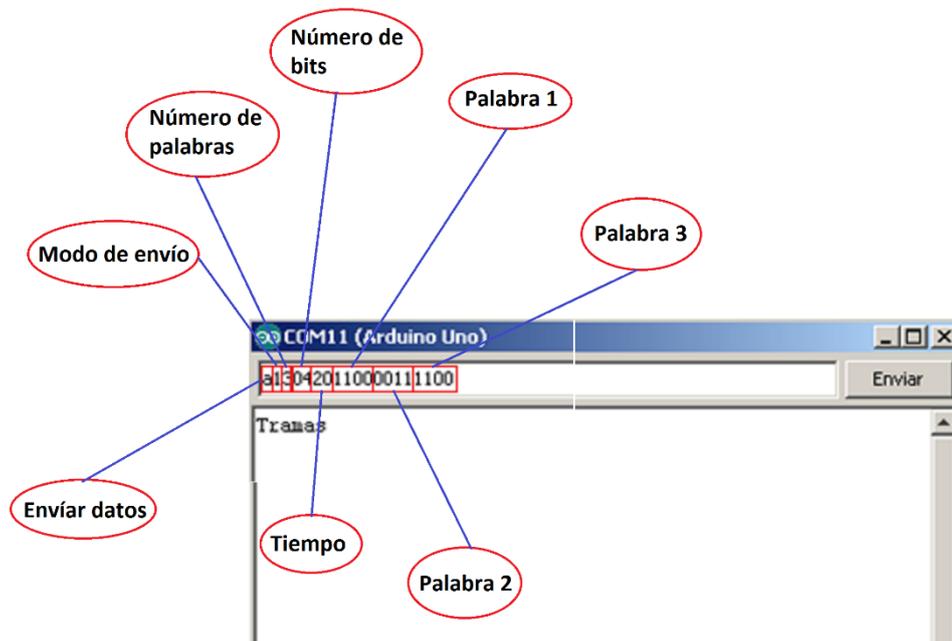


Figura 32. Secuencia caracterizada por el monitor serie

Luego de escribir la secuencia damos clic en enviar para que el generador transmita por el pin13. En la figura 33 se observa la secuencia 1100, 0011,1100, gráficamente en el osciloscopio ingresada a través del canal 1, y como se puede apreciar la señal obtenida es la misma que previamente fue caracterizada utilizando el monitor serie del Arduino.

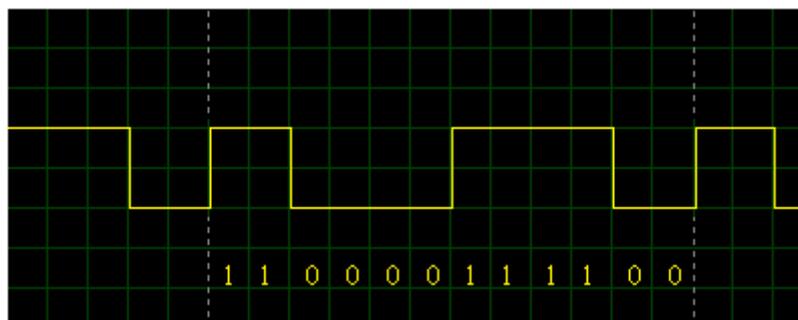


Figura 33. Secuencia digital vista en osciloscopio

Para la realización de las pruebas y verificar el adecuado funcionamiento del **generador de secuencias digitales binarias**, se utilizó el osciloscopio Keysight, adquirido por parte de la UDLA, para su uso en el laboratorio de electrónica y telecomunicaciones. También se utilizó, un generador de funciones, el kit Me1100 que es parte del estudio del presente proyecto y una laptop con Windows 7 donde se ejecuta el programa con la GUI.

Para analizar los resultados del generador de secuencias digitales se procedió a realizar las siguientes pruebas:

- Conexión con GUI desarrollada en Visual Basic 6.0
- Filtros del programa
- Pruebas de filtros para envíos de palabras digitales en modo único.
- Pruebas de filtros para envíos de palabras digitales en modo continuo.
- Pruebas de filtros para envíos de palabra digital con el kit ME1100 y el generador de señales.

4.5 Prueba de envío de palabra única de 4 bits con el módulo ME1100

Para realizar esta prueba utilizamos el módulo ME1100, para lo cual conectamos la salida digital de nuestro generador de secuencias a la entrada digital del módulo, luego conectamos las dos canales digitales de un generador de señales a las entradas I y Q del módulo respectivamente. Finalmente, la señal de salida la conectamos al osciloscopio para poder observar la señal obtenida.

Todo lo mencionado se muestra en la figura 34 a continuación, donde se ve la conexión de los equipos y dispositivos utilizados en esta prueba, como son la laptop, generador de secuencias, generador de señales, el módulo dreamcatcher y finalmente el osciloscopio.

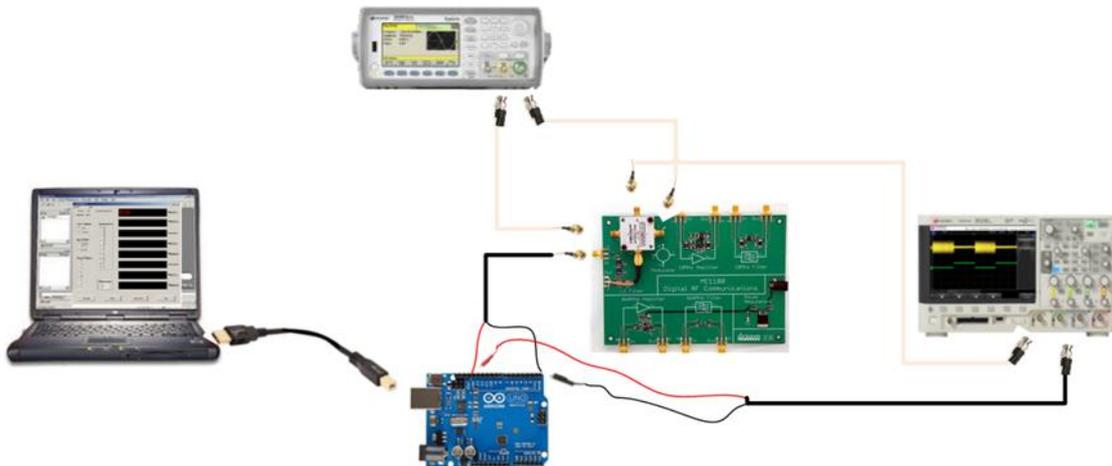


Figura 34. Sistema generador de secuencias digitales, con generador de señales, módulo dreamcatcher ME1100 y osciloscopio.

En el módulo ME1100 también ingresamos dos (2) señales cuadráticas utilizando un generador de señales, a las mismas que configuramos con una amplitud de 5 Vpp, una frecuencia de 1000 Mhz y un desfase de 90° entre una señal y otra. Las señales cuadráticas configuradas con los parámetros mencionados los podemos observar en las s figuras 35 y 36 respectivamente.

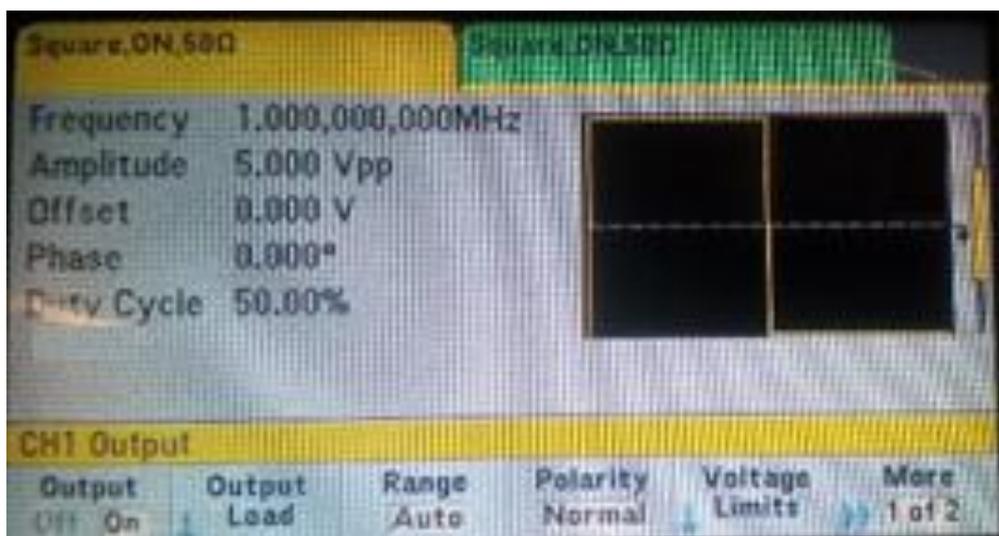


Figura 35. Generador de señales con parámetros configurados en el canal 1.

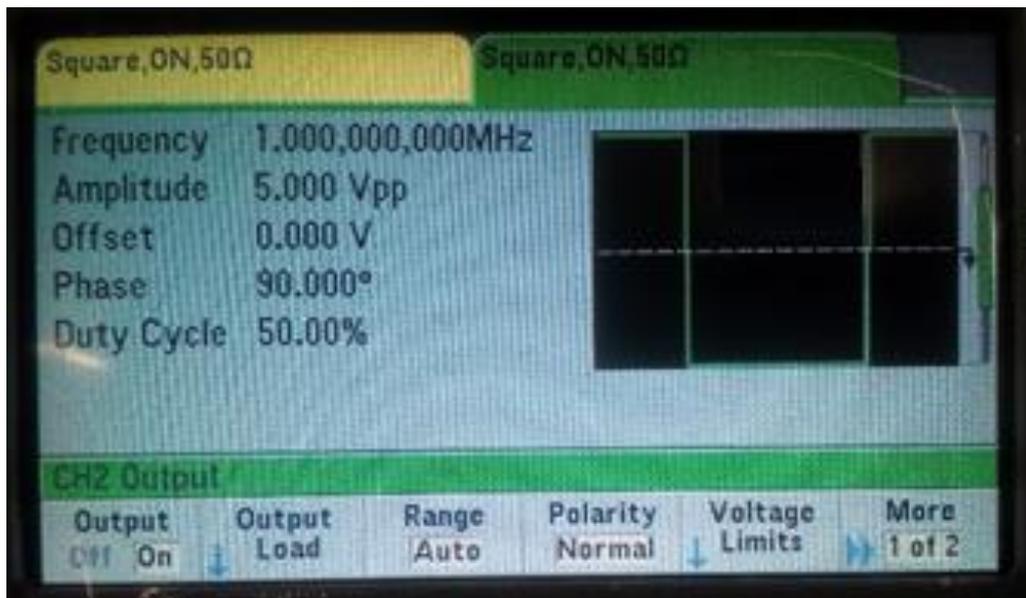


Figura 36. Generador de señales con parámetros configurados con desfase de 90° en el canal 2.

Estas señales configuradas previamente en cada canal del generador de señales se ingresan en las entradas I y Q del módulo Me 1100 respectivamente. En la salida del generador de secuencias digitales que está conectado en el canal 2 de color verde del osciloscopio, se observa la secuencia digital de 4 bits: *0101*, y la señal de salida del módulo Me 1100 está conectada en el canal 1 de color amarillo del osciloscopio.

A continuación, en la figura 37, se puede ver una señal digital modulada que tiene coherencia con los bits que contiene la secuencia caracterizada anteriormente y luego ingresada desde el generador de secuencias digitales hacia el módulo.

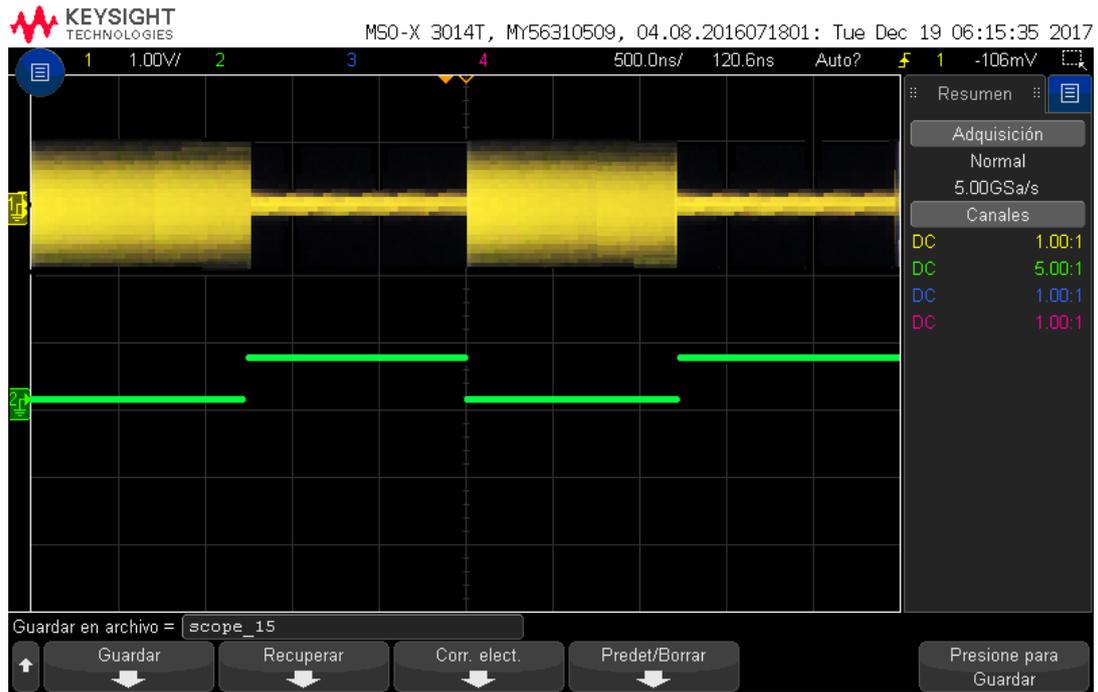


Figura 37. Palabra digital única de 4 bits enviadas al módulo ME1100.

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Este prototipo es una buena herramienta para poder probar diseños digitales, ya que el generador de secuencias digitales provee de diversas señales para probar diseños implementados por estudiantes.
- Arduino es una plataforma que presenta mejoras para desarrollar este tipo de proyectos al ser de fácil instalación, fácil configuración y programación de código abierto, también posee una interfaz muy amigable con el usuario y accesible para cualquier persona que desee emprender en este tipo de proyectos.
- Al contar con una interfaz gráfica, el usuario puede seleccionar las palabras digitales a ser enviadas en las secuencias, de una manera sencilla y ordenada.
- El generador proporciona muchas posibilidades de poder generar diferentes secuencias, ya que cada una de las ocho palabras digitales puede estar formada desde uno a dieciséis bits.
- La forma de envíos que tiene el generador secuencial en modo único y continuo, hacen que sea fácil la verificación de las palabras ingresadas y la señal presentada a la salida del generador.
- Durante las pruebas realizadas se comprueba el funcionamiento de cada uno de los filtros que posee la interfaz gráfica, al caracterizar diferentes palabras digitales y enviar utilizando los dos tipos de envío a diferentes velocidades de intervalo para luego observar las secuencias digitales obtenidas, en el osciloscopio.

5.2 Recomendaciones

- Para una mejor visualización de las secuencias en el osciloscopio se recomienda escoger el tipo de envío en modo continuo, ya que en ese modo seleccionado no se interrumpe la secuencia y se puede realizar los ajustes necesarios en el osciloscopio.

- Para obtener una mejor calidad en la forma de onda de las secuencias se recomienda utilizar una plataforma Arduino de mayor frecuencia de operación como por ejemplo el Arduino Due que es un modelo más avanzado que el utilizado para la implementación de nuestro prototipo.
- El prototipo de generación de secuencias digitales puede ser mejorado utilizando más salidas digitales de la plataforma Arduino, para lo cual se utilizaría una sonda digital para conectar el Arduino al osciloscopio.
- Para proveer de otro tipo de envío se podría añadir a la fase de caracterización en la interfaz gráfica, para realizarla en forma paralela utilizando más salidas digitales mismas que posee el Arduino Due en su hardware.

REFERENCIAS

- Arduino IDE. (s.f.) Arduino software IDE. Recuperado el 14 de octubre de 2017 de <http://arduino.cc/en/Guide/Environment>
- Arduino. (s.f.). Arduino Board UNO. Recuperado el 05 de octubre de 2017 de <http://arduino.cc/en/Main/arduinoBoardUno>
- Barrett, S. (2013). *Arduino Microcontroller Processing for Everyone*. (3.^a ed.). USA: Morgan & Claypool Publishers.
- Di Justo, P. y Gertz, E. (2013). *Atmopheric Monitoring with Arduino*. USA: O'Reilly Media. Inc.
- Dreamcatcher. (s.f.). Módulo de comunicaciones digitales ME1100. Recuperado el 05 de octubre de 2017 de <http://dreamcatcher.asia/cw>
- Enríquez, R. (2009). *Guía de Usuario de Arduino*. San Francisco, USA: Creative Commons.
- Floyd, T.L. (2009). *Fundamentos de sistemas digitales*. (9.^a ed.). México: Prentice Hall
- Hermosa, A. (2004). *Electrónica Digital fundamental*. (3.^a ed.). Barcelona, España: MARCOMBO S.A
- Maxinez, D. (2014). *Programación de Sistemas Digitales con VHDL*. Castilla, España: Grupo Editorial Patria
- Morris, Mano. (2010). *Diseño Digital*. (3.^a ed.). México: Prentice Hall
- Muñoz, J. (2012). *Sistemas Digitales*. México D.F.: Grupo Noriega Editores
- Pallás, R. (1993). *Adquisición y distribución de Señales*. Barcelona, España: MARCOMBO S.A.
- Parera, A. (1991). *Electrónica Básica*. Barcelona, España: MARCOMBO S.A.
- PBRS. (s.f.). PBRS Waveform, Bits and Edges. Recuperado el 15 de septiembre de 2017 de

<https://www.advantest.com/documents/11348/3e95df23-22f5-441e-8598-f1d99c2382cb>

Processing. (s.f.). *Software Processing*. Recuperado el 05 de noviembre de 2017 de <https://processing.org>

Roth, C. (2005). Fundamentos de diseño lógico. (5.^a ed.). México D.F., México: Cengage Learning

Torrente, O. (2013). Curso Práctico de Formación. Madrid, España: RC LIBROS.

Valdivia, C. (2005). Arquitectura de Equipos y Sistemas Informáticos. Madrid, España: Paraninfo, S.A.

Wakerly. (2010). Diseño Digital Principios y Prácticas. (3.^a ed.). México: Prentice Hall

Zetina, A. y Zetina, A. (2004). Electrónica Básica. Balderas, México D.F.: Grupo Noriega Editores.

ANEXOS

Anexo 1. Especificaciones técnicas del módulo ME1100 de DreamCatcher.

868 MHz Filter		Min	Typical	Max	Unit	Remarks / Descriptions
Center frequency		-	901	-	MHz	
Bandwidth		-	439	-	MHz	
Stopband frequency	Low	-	794	-	MHz	Loss > 3 dB
	High	-	1233	-		
	Low	-	725	-	MHz	Loss > 10 dB
	High	-	1330	-		
	Low	-	650	-	MHz	Loss > 20 dB
	High	-	1487	-		
Insertion loss		-	12.0	-	dB	@ 868 MHz
VSWR (:1)	In	-	1.80	-		@ 868 MHz
	Out	-	1.80	-		
Power Requirements		Min	Typical	Max	Unit	Remarks / Descriptions
Input voltage		4.5	5	5.5	V	Power will be regulated to 5 V
Input current		-	22	-	mA	
Frequency range	Low	-	12.0	-	MHz	
	High	-	1050.0	-		
Input reflection, S ₁₁		4.70	-5.40	-	dB	Minimum @ 648 MHz; Typical @ 868 MHz
Output reflection, S ₂₂		-35.70	-7.80	-	dB	Minimum @ 588 MHz; Typical @ 868 MHz
Input transmission, S ₂₁		6.70	6.80	-	dB	Minimum @ 900 MHz; Maximum @ 100 MHz
Output transmission, S ₁₂		-	-18.70	-	dB	@ 868 MHz
Max. power output @ 1 dB compression		-	-2.60	-	dBm	@ 868 MHz
3rd order intercept power, IP3		-	6.20	-	dBm	Two tones, 867 MHz and 869 MHz
VSWR (:1)	In	-	3.30	-		@ 868 MHz
	Out	-	7.90	-		

IQ Modulator		Min	Typical	Max	Unit	Remarks / Descriptions
Conversion loss		-	5.8	7.0	dB	
Sideband rejection		30	40	-	dBc	
Carrier rejection		30	41	-	dBc	
3rd harmonic suppression		45	58	-	dBc	
5th harmonic suppression		58	68	-	dBc	
DC offset		-	0.08	0.09	mV	
10 MHz Amplifier		Min	Typical	Max	Unit	Remarks / Descriptions
Power supply voltage		3.2	3.3	3.4	V	
Gain		-	25.70	-	dB	@ 10 MHz
Frequency range	Low	-	3	-	MHz	@ approximate gain < 10 dB
	High	-	18	-		
Input reflection, S ₁₁		1.41	1.44	-	dB	Minimum @ 5.5 MHz; Typical @ 10 MHz
Output reflection, S ₂₂		24.20	4.8	-	dB	Minimum @ 11.5 MHz; Typical @ 10 MHz
Input transmission, S ₂₁		26.07	26.25	-	dB	Minimum @ 3 MHz; Maximum @ 11 MHz
Output transmission, S ₁₂		-	-49.00	-	dB	@ 10 MHz
Max. power output @ 1 dB compression		-	5.20	-	dBm	@ 10 MHz
3rd order intercept power, IP3		-	14.10	-	dBm	Two tones, 10.1 MHz and 9.9 MHz
VSWR (:1)	In	-	9.60	-		@ 10 MHz
	Out	-	3.80	-		

Anexo 2. Código Fuente realizado en Visual Basic 6.0

Form 1:

VERSION 5.00

Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0";
"MSCOMM32.OCX"

Begin VB.Form Form1

Caption = "Señal Digital"
ClientHeight = 8790
ClientLeft = 2775
ClientTop = 1320
ClientWidth = 9825
LinkTopic = "Form1"
MaxButton = 0 'False
ScaleHeight = 8790
ScaleWidth = 9825
StartPosition = 2 'CenterScreen

Begin VB.Frame Frame4

Caption = "Tipo de Dato"
BeginProperty Font
Name = "MS Sans Serif"
Size = 8.25
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

EndProperty

Height = 1335
Left = 240
TabIndex = 57
Top = 2760
Width = 1575

Begin VB.OptionButton **Option29**

Caption = "2 ASCII"
Height = 195
Left = 240
TabIndex = 60
Top = 960
Width = 1095

End

Begin VB.OptionButton **Option28**

Caption = "1 ASCII"
Height = 255
Left = 240
TabIndex = 59
Top = 600

Width = 1095

End

Begin VB.OptionButton Option27

Caption = "Números"

Height = 195

Left = 240

TabIndex = 58

Top = 360

Value = -1 'True

Width = 975

End

End

Begin VB.CommandButton CmdLimpiar

Caption = "&Limpiar Todo"

Height = 495

Left = 5280

TabIndex = 48

Top = 8040

Width = 1575

End

Begin VB.CommandButton CmdParar

Caption = "&Parar"

Height = 495

Left = 2880

TabIndex = 47

Top = 8040

Width = 1575

End

Begin VB.TextBox Text1

BackColor = &H00000000&

BeginProperty Font

Name = "MS Sans Serif"

Size = 24

Charset = 0

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

ForeColor = &H000000FF&

Height = 735

Index = 7

Left = 4200

MaxLength = 16

TabIndex = 46

Top = 6720

Width = 4455

End

```
Begin VB.TextBox Text1
  BackColor = &H00000000&
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 24
    Charset = 0
    Weight = 400
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  ForeColor = &H000000FF&
  Height = 735
  Index = 6
  Left = 4200
  MaxLength = 16
  TabIndex = 45
  Top = 5760
  Width = 4455
```

End

```
Begin VB.TextBox Text1
  BackColor = &H00000000&
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 24
    Charset = 0
    Weight = 400
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  ForeColor = &H000000FF&
  Height = 735
  Index = 5
  Left = 4200
  MaxLength = 16
  TabIndex = 44
  Top = 4800
  Width = 4455
```

End

```
Begin VB.TextBox Text1
  BackColor = &H00000000&
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 24
    Charset = 0
    Weight = 400
    Underline = 0 'False
```

```
    Italic      = 0 'False
    Strikethrough = 0 'False
EndProperty
ForeColor     = &H000000FF&
Height       = 735
Index        = 4
Left         = 4200
MaxLength    = 16
TabIndex     = 43
Top         = 3840
Width       = 4455
```

End

Begin VB.TextBox Text1

```
BackColor     = &H00000000&
BeginProperty Font
    Name       = "MS Sans Serif"
    Size      = 24
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
ForeColor     = &H000000FF&
Height       = 735
Index        = 3
Left         = 4200
MaxLength    = 16
TabIndex     = 42
Top         = 2880
Width       = 4455
```

End

Begin VB.TextBox Text1

```
BackColor     = &H00000000&
BeginProperty Font
    Name       = "MS Sans Serif"
    Size      = 24
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
ForeColor     = &H000000FF&
Height       = 735
Index        = 2
Left         = 4200
MaxLength    = 16
```

TabIndex = 41
Top = 1920
Width = 4455

End

Begin VB.TextBox Text1

BackColor = &H00000000&
BeginProperty Font
Name = "MS Sans Serif"
Size = 24
Charset = 0
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
ForeColor = &H000000FF&
Height = 735
Index = 1
Left = 4200
MaxLength = 16
TabIndex = 40
Top = 960
Width = 4455

End

Begin VB.TextBox Text9

Alignment = 2 'Center
BeginProperty Font
Name = "MS Sans Serif"
Size = 12
Charset = 0
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 375
Left = 2760
MaxLength = 2
TabIndex = 38
Text = "20"
Top = 6960
Width = 495

End

Begin VB.Frame Frame3

Caption = "Número de Bits"
BeginProperty Font
Name = "MS Sans Serif"
Size = 8.25

```
Charset      = 0
Weight       = 700
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False
EndProperty
Height       = 4335
Left         = 2280
TabIndex     = 11
Top          = 1320
Width        = 1455
Begin VB.OptionButton Option26
    Caption    = "16"
    Height     = 195
    Left       = 240
    TabIndex   = 37
    Top        = 360
    Width      = 855
End
Begin VB.OptionButton Option25
    Caption    = "15"
    Height     = 195
    Left       = 240
    TabIndex   = 36
    Top        = 600
    Width      = 855
End
Begin VB.OptionButton Option24
    Caption    = "14"
    Height     = 195
    Left       = 240
    TabIndex   = 35
    Top        = 840
    Width      = 855
End
Begin VB.OptionButton Option23
    Caption    = "13"
    Height     = 195
    Left       = 240
    TabIndex   = 34
    Top        = 1080
    Width      = 855
End
Begin VB.OptionButton Option22
    Caption    = "12"
    Height     = 195
    Left       = 240
    TabIndex   = 33
```

Top = 1320
Width = 855

End

Begin VB.OptionButton [Option21](#)

Caption = "11"
Height = 195
Left = 240
TabIndex = 32
Top = 1560
Width = 735

End

Begin VB.OptionButton [Option20](#)

Caption = "10"
Height = 195
Left = 240
TabIndex = 31
Top = 1800
Width = 855

End

Begin VB.OptionButton [Option19](#)

Caption = "9"
Height = 195
Left = 240
TabIndex = 30
Top = 2040
Width = 735

End

Begin VB.OptionButton [Option18](#)

Caption = "8"
Height = 195
Left = 240
TabIndex = 29
Top = 2280
Width = 735

End

Begin VB.OptionButton [Option17](#)

Caption = "7"
Height = 195
Left = 240
TabIndex = 28
Top = 2520
Width = 855

End

Begin VB.OptionButton [Option16](#)

Caption = "6"
Height = 195
Left = 240
TabIndex = 27

Top = 2760
Width = 735

End

Begin VB.OptionButton Option15

Caption = "5"
Height = 195
Left = 240
TabIndex = 26
Top = 3000
Width = 615

End

Begin VB.OptionButton Option14

Caption = "4"
Height = 195
Left = 240
TabIndex = 25
Top = 3240
Value = -1 'True
Width = 735

End

Begin VB.OptionButton Option13

Caption = "3"
Height = 195
Left = 240
TabIndex = 24
Top = 3480
Width = 735

End

Begin VB.OptionButton Option12

Caption = "2"
Height = 195
Left = 240
TabIndex = 23
Top = 3720
Width = 735

End

Begin VB.OptionButton Option11

Caption = "1"
Height = 195
Left = 240
TabIndex = 22
Top = 3960
Width = 735

End

End

Begin VB.Frame Frame2

Caption = "Unica-Continua"
BeginProperty Font

```
Name      = "MS Sans Serif"
Size      = 8.25
Charset   = 0
Weight    = 700
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
Height    = 1095
Left     = 240
TabIndex = 10
Top      = 1320
Width    = 1575
Begin VB.OptionButton Option2
  Caption = "Unica"
  Height  = 195
  Left    = 240
 TabIndex = 13
  Top     = 360
  Width   = 975
End
Begin VB.OptionButton Option1
  Caption = "Continua"
  Height  = 255
  Left    = 240
 TabIndex = 12
  Top     = 720
  Value   = -1 'True
  Width   = 1095
End
Begin VB.Frame Frame1
  Caption = "Número Palabras"
  BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 8.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height    = 3015
  Left     = 240
 TabIndex  = 9
  Top      = 4440
  Width    = 1575
  Begin VB.OptionButton Option10
```

Caption = "8"
Height = 255
Left = 240
TabIndex = 21
Top = 240
Width = 1215

End

Begin VB.OptionButton [Option9](#)

Caption = "7"
Height = 255
Left = 240
TabIndex = 20
Top = 600
Width = 1095

End

Begin VB.OptionButton [Option8](#)

Caption = "6"
Height = 255
Left = 240
TabIndex = 19
Top = 960
Width = 1095

End

Begin VB.OptionButton [Option7](#)

Caption = "5"
Height = 255
Left = 240
TabIndex = 18
Top = 1320
Width = 1095

End

Begin VB.OptionButton [Option6](#)

Caption = "4"
Height = 255
Left = 240
TabIndex = 17
Top = 1680
Width = 1095

End

Begin VB.OptionButton [Option5](#)

Caption = "3"
Height = 375
Left = 240
TabIndex = 16
Top = 1920
Value = -1 'True
Width = 1095

End

Begin VB.OptionButton Option4

Caption = "2"
Height = 255
Left = 240
TabIndex = 15
Top = 2280
Width = 1095

End

Begin VB.OptionButton Option3

Caption = "1"
Height = 255
Left = 240
TabIndex = 14
Top = 2640
Width = 1215

End

End

Begin VB.CommandButton CmdTrans

Caption = "&Transmitir"
Height = 495
Left = 480
TabIndex = 1
Top = 8040
Width = 1575

End

Begin VB.CommandButton CmdSalir

Caption = "&Salir"
Height = 495
Left = 7680
TabIndex = 2
Top = 8040
Width = 1695

End

Begin VB.TextBox Text1

BackColor = &H00000000&
BeginProperty Font
Name = "MS Sans Serif"
Size = 24
Charset = 0
Weight = 400
Underline = 0 'False'
Italic = 0 'False'
Strikethrough = 0 'False'
EndProperty
ForeColor = &H000000FF&
Height = 735
Index = 0
Left = 4200

MaxLength = 16
TabIndex = 0
Top = 0
Width = 4455

End

Begin MSCommLib.MSComm MSComm1

Left = 2160
Top = 7920
_ExtentX = 1005
_ExtentY = 1005
_Version = 393216
DTREnable = 0 'False
OutBufferSize = 1024

End

Begin VB.Label Label16

Caption = "Label16"
Height = 255
Index = 7
Left = 6960
TabIndex = 68
Top = 9720
Width = 2055

End

Begin VB.Label Label16

Caption = "Label16"
Height = 255
Index = 6
Left = 4680
TabIndex = 67
Top = 9720
Width = 2055

End

Begin VB.Label Label16

Caption = "Label16"
Height = 255
Index = 5
Left = 2400
TabIndex = 66
Top = 9720
Width = 2055

End

Begin VB.Label Label16

Caption = "Label16"
Height = 255
Index = 4
Left = 120
TabIndex = 65
Top = 9720

```
Width      = 2055
End
Begin VB.Label Label16
Caption    = "Label16"
Height    = 255
Index     = 3
Left      = 6720
TabIndex  = 64
Top       = 9360
Width     = 2055
```

```
End
Begin VB.Label Label16
Caption    = "Label16"
Height    = 255
Index     = 2
Left      = 4560
TabIndex  = 63
Top       = 9360
Width     = 2055
```

```
End
Begin VB.Label Label16
Caption    = "Label16"
Height    = 255
Index     = 1
Left      = 2400
TabIndex  = 62
Top       = 9360
Width     = 2055
```

```
End
Begin VB.Label Label16
Caption    = "Label16"
Height    = 255
Index     = 0
Left      = 0
TabIndex  = 61
Top       = 9360
Width     = 2055
```

```
End
Begin VB.Line Line4
X1        = 2280
X2        = 3720
Y1        = 7440
Y2        = 7440
```

```
End
Begin VB.Line Line3
X1        = 3720
X2        = 3720
Y1        = 7440
```

```

Y2          = 6480
End
Begin VB.Line Line2
X1          = 3720
X2          = 2280
Y1          = 6480
Y2          = 6480
End
Begin VB.Line Line1
X1          = 2280
X2          = 2280
Y1          = 6480
Y2          = 7440
End
Begin VB.Label Label15
Alignment   = 1 'Right Justify
Caption     = "PALABRA 8"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 8.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height     = 255
Left       = 8640
TabIndex   = 56
Top        = 6960
Width      = 1095
End
Begin VB.Label Label14
Alignment   = 1 'Right Justify
Caption     = "PALABRA 7"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 8.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height     = 255
Left       = 8640
TabIndex   = 55
Top        = 6000

```

```
Width      = 1095
End
Begin VB.Label Label13
Alignment  = 1 'Right Justify
Caption    = "PALABRA 6"
BeginProperty Font
    Name     = "MS Sans Serif"
    Size     = 8.25
    Charset  = 0
    Weight   = 700
    Underline = 0 'False
    Italic   = 0 'False
    Strikethrough = 0 'False
EndProperty
Height     = 255
Left       = 8640
TabIndex   = 54
Top        = 5040
Width      = 1095
```

```
End
Begin VB.Label Label12
Alignment  = 1 'Right Justify
Caption    = "PALABRA 5"
BeginProperty Font
    Name     = "MS Sans Serif"
    Size     = 8.25
    Charset  = 0
    Weight   = 700
    Underline = 0 'False
    Italic   = 0 'False
    Strikethrough = 0 'False
EndProperty
Height     = 255
Left       = 8640
TabIndex   = 53
Top        = 4080
Width      = 1095
```

```
End
Begin VB.Label Label11
Alignment  = 1 'Right Justify
Caption    = "PALABRA 4"
BeginProperty Font
    Name     = "MS Sans Serif"
    Size     = 8.25
    Charset  = 0
    Weight   = 700
    Underline = 0 'False
    Italic   = 0 'False
```

```

    Strikethrough = 0 'False
EndProperty
Height      = 255
Left       = 8640
TabIndex  = 52
Top       = 3120
Width     = 1095
End
Begin VB.Label Label8
Alignment  = 1 'Right Justify
Caption    = "PALABRA 3"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 8.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height     = 255
Left      = 8640
TabIndex  = 51
Top       = 2160
Width     = 1095
End
Begin VB.Label Label7
Alignment  = 1 'Right Justify
Caption    = "PALABRA 2"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 8.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height     = 255
Left      = 8640
TabIndex  = 50
Top       = 1080
Width     = 1095
End
Begin VB.Label Label5
Alignment  = 1 'Right Justify
Caption    = "PALABRA 1"
BeginProperty Font

```

```
Name      = "MS Sans Serif"
Size      = 8.25
Charset   = 0
Weight    = 700
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
Height    = 255
Left     = 8640
TabIndex = 49
Top      = 240
Width    = 1095
End
Begin VB.Label Label3
Alignment = 2 'Center
Caption   = "Tiempo mseg."
BeginProperty Font
Name      = "MS Sans Serif"
Size      = 8.25
Charset   = 0
Weight    = 700
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
Height    = 255
Left     = 2400
TabIndex = 39
Top      = 6600
Width    = 1215
End
Begin VB.Label lblvelocidad
Height    = 255
Left     = 1200
TabIndex = 8
Top      = 480
Width    = 495
End
Begin VB.Label lblportico
Height    = 255
Left     = 1200
TabIndex = 7
Top      = 120
Width    = 495
End
Begin VB.Label Label2
Alignment = 1 'Right Justify
```

```

Caption      = "Velocidad :"
Height      = 255
Left        = 240
TabIndex    = 6
Top         = 480
Width       = 855
End
Begin VB.Label Label1
Alignment   = 1 'Right Justify
Caption     = "Pórtico :"
Height     = 255
Left       = 360
TabIndex   = 5
Top        = 120
Width      = 735
End
Begin VB.Label Label4
Alignment   = 2 'Center
Height     = 255
Left       = 3720
TabIndex   = 4
Top        = 120
Width      = 255
End
Begin VB.Label Label6
Alignment   = 1 'Right Justify
Caption     = "Caracter Transmitido :"
Height     = 255
Left       = 1920
TabIndex   = 3
Top        = 120
Width      = 1695
End
Begin VB.Menu MnuArchivo
Caption     = "&Archivo"
Begin VB.Menu Mnulinea
Caption     = "-"
End
Begin VB.Menu MnuPropiedades
Caption     = "Propiedades"
End
Begin VB.Menu Menulinea
Caption     = "-"
End
Begin VB.Menu MnuSalir1
Caption     = "&Salir"
Begin VB.Menu MnuAcerca

```

```

        Caption      = "A&cerca de ..."
    End
    Begin VB.Menu MnuSalir
        Caption      = "&Salir"
    End
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Dim Totalcom As Integer
Dim Totalcoms As String * 3
Dim Totalc(3) As Byte
Dim Totals(3) As String * 3
Dim dato1 As String * 1
Dim dato2 As String * 2
Dim dato3 As String * 3
Dim Cuenta As Integer
Dim error1 As Boolean

Sub Pausams()
    e = Timer
    Do
        f = Timer - e
    Loop While f < 0.02
End Sub

Private Sub CmdLimpiar_Click()
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

Private Sub CmdParar_Click()
Dim OpenFlag
    CmdParar.Enabled = False
    OpenFlag = Form1.MSComm1.PortOpen
    If OpenFlag Then
    Else
        Form1.MSComm1.PortOpen = Not Form1.MSComm1.PortOpen
    End If
    Form1.MSComm1.OutBufferCount = 0
    Form1.MSComm1.InBufferCount = 0
    '-----
    ' Caracter Paro
    Form1.MSComm1.Output = "b"

```

```

Form1.Label4.Caption = "b"
Do
    ret = DoEvents()
Loop Until Form1.MSComm1.OutBufferCount = 0
Call Pausams
CmdParar.Enabled = True
End Sub

Private Sub CmdSalir_Click()
    If Form1.MSComm1.PortOpen Then Form1.MSComm1.PortOpen = False
    End
End Sub

Private Sub CmdTrans_Click()
Dim Mensaje, Estilo, Título, Respuesta
Dim OpenFlag
'Dim dato1 As String * 1
Dim dato1 As String
Dim dato2 As String
Dim DatoAscii
Dim DatoBinario() As String
Dim DatoText1 As String
Dim DatoText2 As String
Dim DatoText3 As String
Dim DatoText4() As String
Dim c As Integer
Dim x As Byte
Cuenta = 0

    If Option3.Value = True Then palabras = "1"
    If Option4.Value = True Then palabras = "2"
    If Option5.Value = True Then palabras = "3"
    If Option6.Value = True Then palabras = "4"
    If Option7.Value = True Then palabras = "5"
    If Option8.Value = True Then palabras = "6"
    If Option9.Value = True Then palabras = "7"
    If Option10.Value = True Then palabras = "8"

    If Option27.Value = True Then tipodato = "1"
    If Option28.Value = True Then tipodato = "2"
    If Option29.Value = True Then tipodato = "3"

    If Option11.Value = True Then bits = "01"
    If Option12.Value = True Then bits = "02"
    If Option13.Value = True Then bits = "03"
    If Option14.Value = True Then bits = "04"
    If Option15.Value = True Then bits = "05"
    If Option16.Value = True Then bits = "06"

```

```

If Option17.Value = True Then bits = "07"
If Option18.Value = True Then bits = "08"
If Option19.Value = True Then bits = "09"
If Option20.Value = True Then bits = "10"
If Option21.Value = True Then bits = "11"
If Option22.Value = True Then bits = "12"
If Option23.Value = True Then bits = "13"
If Option24.Value = True Then bits = "14"
If Option25.Value = True Then bits = "15"
If Option26.Value = True Then bits = "16"
flag1 = 0
If Option27.Value = True Then
    For X1 = 0 To palabras - 1
        If Len(Text1(X1).Text) = bits Then
        Else
            flag1 = 1
        End If
    Next X1
End If
If Option28.Value = True Then
    For X1 = 0 To palabras - 1
        If Len(Text1(X1).Text) = 1 Then
        Else
            flag1 = 1
        End If
    Next X1
End If
If Option29.Value = True Then
    For X1 = 0 To palabras - 1
        If Len(Text1(X1).Text) = 2 Then
        Else
            flag1 = 1
        End If
    Next X1
End If

If flag1 = 1 Then
    Mensaje = " Error en número de palabras o bits " ' Define el mensaje.
    Estilo = vbOKOnly + vbExclamation ' Define los botones.
    Título = "Envío de Señal Digital" ' Define el título.
    Respuesta = MsgBox(Mensaje, Estilo, Título, Ayuda, Ctxt)
    Exit Sub
End If

OpenFlag = Form1.MSComm1.PortOpen
If OpenFlag Then
Else
    Form1.MSComm1.PortOpen = Not Form1.MSComm1.PortOpen

```

```
End If
Form1.MSComm1.OutBufferCount = 0
Form1.MSComm1.InBufferCount = 0
CmdTrans.Enabled = False
```

' Caracter de Inicio

```
Form1.MSComm1.Output = "a"
Form1.Label4.Caption = "a"
Do
    ret = DoEvents()
Loop Until Form1.MSComm1.OutBufferCount = 0
Call Pausams
```

' Unica-Continua

```
If Option1.Value = True Then unica_continua = "1"
If Option2.Value = True Then unica_continua = "0"
Form1.Label4.Caption = unica_continua
Form1.MSComm1.Output = unica_continua
Do
    ret = DoEvents()
Loop Until Form1.MSComm1.OutBufferCount = 0
Call Pausams
```

' Palabras

```
If Option3.Value = True Then palabras = "1"
If Option4.Value = True Then palabras = "2"
If Option5.Value = True Then palabras = "3"
If Option6.Value = True Then palabras = "4"
If Option7.Value = True Then palabras = "5"
If Option8.Value = True Then palabras = "6"
If Option9.Value = True Then palabras = "7"
If Option10.Value = True Then palabras = "8"
```

```
Form1.Label4.Caption = palabras
Form1.MSComm1.Output = palabras
Do
    ret = DoEvents()
Loop Until Form1.MSComm1.OutBufferCount = 0
Call Pausams
```

' Bits

```
If Option11.Value = True Then bits = "01"
If Option12.Value = True Then bits = "02"
If Option13.Value = True Then bits = "03"
If Option14.Value = True Then bits = "04"
If Option15.Value = True Then bits = "05"
If Option16.Value = True Then bits = "06"
If Option17.Value = True Then bits = "07"
```

```
If Option18.Value = True Then bits = "08"  
If Option19.Value = True Then bits = "09"  
If Option20.Value = True Then bits = "10"  
If Option21.Value = True Then bits = "11"  
If Option22.Value = True Then bits = "12"  
If Option23.Value = True Then bits = "13"  
If Option24.Value = True Then bits = "14"  
If Option25.Value = True Then bits = "15"  
If Option26.Value = True Then bits = "16"
```

```
For c = 1 To 2  
    d = Mid(bits, c, 1)  
    Form1.Label4.Caption = d  
    Form1.MSComm1.Output = d  
    Form1.Label4.Caption = d  
    Do  
        ret = DoEvents()  
    Loop Until Form1.MSComm1.OutBufferCount = 0  
    Call Pausams  
Next c  
Do  
    ret = DoEvents()  
Loop Until Form1.MSComm1.OutBufferCount = 0  
Call Pausams
```

'-----
' Retardo

```
For c = 1 To 2  
    d = Mid(Text9.Text, c, 1)  
    Form1.Label4.Caption = d  
    Form1.MSComm1.Output = d  
    Form1.Label4.Caption = d  
    Do  
        ret = DoEvents()  
    Loop Until Form1.MSComm1.OutBufferCount = 0  
    Call Pausams  
Next c
```

'-----
' Envio Datos

```
If Option27.Value = True Then  
    For c = 0 To (palabras - 1)  
        For e = 1 To bits  
            d = Mid(Text1(c).Text, e, 1)  
            Form1.Label4.Caption = d  
            Form1.MSComm1.Output = d  
            Form1.Label4.Caption = d  
        Do  
            ret = DoEvents()
```

```

        Loop Until Form1.MSComm1.OutBufferCount = 0
        Call Pausams
        'MsgBox d
    Next e
Next c
End If

If Option28.Value = True Then
    For f = 1 To palabras
        DatoText1 = Hex2Bin(Left(Hex(Asc(Text1(f - 1).Text)), 1)) &
Hex2Bin(Right(Hex(Asc(Text1(f - 1).Text)), 1))
        Label16(f - 1).Caption = DatoText1
    Next f

    For c = 0 To (palabras - 1)
        For e = 1 To bits
            d = Mid(Label16(c).Caption, e, 1)
            Form1.Label4.Caption = d
            Form1.MSComm1.Output = d
            Form1.Label4.Caption = d
            Do
                ret = DoEvents()
            Loop Until Form1.MSComm1.OutBufferCount = 0
            Call Pausams
            'MsgBox d
        Next e
    Next c
End If

If Option29.Value = True Then
    For c = 0 To (palabras - 1)
        dato1 = Mid(Text1(c).Text, 1, 1)
        DatoText1 = Hex2Bin(Left(Hex(Asc(dato1)), 1)) &
Hex2Bin(Right(Hex(Asc(dato1)), 1))
        dato2 = Mid(Text1(c).Text, 2, 1)
        DatoText2 = Hex2Bin(Left(Hex(Asc(dato2)), 1)) &
Hex2Bin(Right(Hex(Asc(dato2)), 1))
        DatoText3 = DatoText1 + DatoText2
        Label16(c).Caption = DatoText3
    Next c
    For c = 0 To (palabras - 1)
        For e = 1 To bits
            d = Mid(Label16(c).Caption, e, 1)
            Form1.Label4.Caption = d
            Form1.MSComm1.Output = d
            Form1.Label4.Caption = d
            Do
                ret = DoEvents()
            Loop Until Form1.MSComm1.OutBufferCount = 0

```

```

        Call Pausams
        'MsgBox d
    Next e
Next c
End If
'-----
Mensaje = " Programación OK " ' Define el mensaje.
Estilo = vbOKOnly + vbExclamation ' Define los botones.
Título = "Envío de Señal Digital" ' Define el título.
Respuesta = MsgBox(Mensaje, Estilo, Título, Ayuda, Ctxt)
Form1.Label4.Caption = ""
Form1.MSComm1.InBufferCount = 0
Form1.MSComm1.OutBufferCount = 0
Form1.MSComm1.PortOpen = False
CmdTrans.Enabled = True
End Sub

```

```

Private Function Hex2Bin(DatoHex As String) As String
    Select Case LCase(DatoHex)
        Case "0"
            Hex2Bin = "0000"
        Case "1"
            Hex2Bin = "0001"
        Case "2"
            Hex2Bin = "0010"
        Case "3"
            Hex2Bin = "0011"
        Case "4"
            Hex2Bin = "0100"
        Case "5"
            Hex2Bin = "0101"
        Case "6"
            Hex2Bin = "0110"
        Case "7"
            Hex2Bin = "0111"
        Case "8"
            Hex2Bin = "1000"
        Case "9"
            Hex2Bin = "1001"
        Case "a"
            Hex2Bin = "1010"
        Case "b"
            Hex2Bin = "1011"
        Case "c"
            Hex2Bin = "1100"
        Case "d"
            Hex2Bin = "1101"
        Case "e"

```

```

        Hex2Bin = "1110"
    Case "f"
        Hex2Bin = "1111"
    Case Else
        MsgBox "Valor no admitido!"
    End Select
End Function

Private Sub Form_Load()
Dim i As Integer
    Form1.lblPortico.Caption = "Com" & Form1.MSComm1.CommPort
    Form1.lblvelocidad.Caption = Left$(Form1.MSComm1.Settings,
Len(Form1.MSComm1.Settings) - 6)
    Option5_Click
    Option14_Click
'   CmdTrans.Enabled = False
End Sub

Private Sub MnuAcerca_Click()
    frmAbout.Show vbModal
End Sub

Private Sub MnuPropiedades_Click()
    Form2.Show vbModal
End Sub

Private Sub MnuSalir_Click()
    CmdSalir_Click
End Sub

Private Sub Option11_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 1
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

Private Sub Option12_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 2
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

```

```
Private Sub Option13_Click()  
    For y = 0 To 7  
        Text1(y).MaxLength = 3  
    Next y  
    For x = 0 To 7  
        Text1(x).Text = ""  
    Next x  
End Sub
```

```
Private Sub Option14_Click()  
    For y = 0 To 7  
        Text1(y).MaxLength = 4  
    Next y  
    For x = 0 To 7  
        Text1(x).Text = ""  
    Next x  
End Sub
```

```
Private Sub Option15_Click()  
    For y = 0 To 7  
        Text1(y).MaxLength = 5  
    Next y  
    For x = 0 To 7  
        Text1(x).Text = ""  
    Next x  
End Sub
```

```
Private Sub Option16_Click()  
    For y = 0 To 7  
        Text1(y).MaxLength = 6  
    Next y  
    For x = 0 To 7  
        Text1(x).Text = ""  
    Next x  
End Sub
```

```
Private Sub Option17_Click()  
    For y = 0 To 7  
        Text1(y).MaxLength = 7  
    Next y  
    For x = 0 To 7  
        Text1(x).Text = ""  
    Next x  
End Sub
```

```
Private Sub Option18_Click()  
    For y = 0 To 7  
        Text1(y).MaxLength = 8
```

```
Next y
For x = 0 To 7
    Text1(x).Text = ""
Next x
End Sub

Private Sub Option19_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 9
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

Private Sub Option20_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 10
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

Private Sub Option21_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 11
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

Private Sub Option22_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 12
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub

Private Sub Option23_Click()
    For y = 0 To 7
        Text1(y).MaxLength = 13
    Next y
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub
```

```
Next x  
End Sub
```

```
Private Sub Option24_Click()  
For y = 0 To 7  
Text1(y).MaxLength = 14  
Next y  
For x = 0 To 7  
Text1(x).Text = ""  
Next x  
End Sub
```

```
Private Sub Option25_Click()  
For y = 0 To 7  
Text1(y).MaxLength = 15  
Next y  
For x = 0 To 7  
Text1(x).Text = ""  
Next x  
End Sub
```

```
Private Sub Option26_Click()  
For y = 0 To 7  
Text1(y).MaxLength = 16  
Next y  
For x = 0 To 7  
Text1(x).Text = ""  
Next x  
End Sub
```

```
Private Sub Option27_Click()  
Option5_Click  
Option14_Click  
Frame3.Enabled = True  
Option14.Value = True  
End Sub
```

```
Private Sub Option28_Click()  
Option18.Value = True  
For y = 0 To 7  
Text1(y).MaxLength = 1  
Next y  
Frame3.Enabled = False  
End Sub
```

```
Private Sub Option29_Click()  
Option26.Value = True  
For y = 0 To 7
```

```
    Text1(y).MaxLength = 2
Next y
Frame3.Enabled = False
End Sub
```

```
Private Sub Option3_Click()
    Text1(0).Enabled = True
    Text1(1).Enabled = False
    Text1(2).Enabled = False
    Text1(3).Enabled = False
    Text1(4).Enabled = False
    Text1(5).Enabled = False
    Text1(6).Enabled = False
    Text1(7).Enabled = False
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub
```

```
Private Sub Option4_Click()
    Text1(0).Enabled = True
    Text1(1).Enabled = True
    Text1(2).Enabled = False
    Text1(3).Enabled = False
    Text1(4).Enabled = False
    Text1(5).Enabled = False
    Text1(6).Enabled = False
    Text1(7).Enabled = False
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub
```

```
Private Sub Option5_Click()
    Text1(0).Enabled = True
    Text1(1).Enabled = True
    Text1(2).Enabled = True
    Text1(3).Enabled = False
    Text1(4).Enabled = False
    Text1(5).Enabled = False
    Text1(6).Enabled = False
    Text1(7).Enabled = False
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub
```

```
Private Sub Option6_Click()
```

```
Text1(0).Enabled = True
Text1(1).Enabled = True
Text1(2).Enabled = True
Text1(3).Enabled = True
Text1(4).Enabled = False
Text1(5).Enabled = False
Text1(6).Enabled = False
Text1(7).Enabled = False
For x = 0 To 7
    Text1(x).Text = ""
Next x
End Sub
```

```
Private Sub Option7_Click()
Text1(0).Enabled = True
Text1(1).Enabled = True
Text1(2).Enabled = True
Text1(3).Enabled = True
Text1(4).Enabled = True
Text1(5).Enabled = False
Text1(6).Enabled = False
Text1(7).Enabled = False
For x = 0 To 7
    Text1(x).Text = ""
Next x
End Sub
```

```
Private Sub Option8_Click()
Text1(0).Enabled = True
Text1(1).Enabled = True
Text1(2).Enabled = True
Text1(3).Enabled = True
Text1(4).Enabled = True
Text1(5).Enabled = True
Text1(6).Enabled = False
Text1(7).Enabled = False
For x = 0 To 7
    Text1(x).Text = ""
Next x
End Sub
```

```
Private Sub Option9_Click()
Text1(0).Enabled = True
Text1(1).Enabled = True
Text1(2).Enabled = True
Text1(3).Enabled = True
Text1(4).Enabled = True
Text1(5).Enabled = True
```

```
Text1(6).Enabled = True
Text1(7).Enabled = False
For x = 0 To 7
    Text1(x).Text = ""
Next x
End Sub
```

```
Private Sub Option10_Click()
    Text1(0).Enabled = True
    Text1(1).Enabled = True
    Text1(2).Enabled = True
    Text1(3).Enabled = True
    Text1(4).Enabled = True
    Text1(5).Enabled = True
    Text1(6).Enabled = True
    Text1(7).Enabled = True
    For x = 0 To 7
        Text1(x).Text = ""
    Next x
End Sub
```

```
Private Sub Text1_Change(Index As Integer)
' If Len(Text1(Index).Text) = 0 Then
'     CmdTrans.Enabled = False
' Else
'     CmdTrans.Enabled = True
' End If
' If (Len(Text1(Index).Text) >= Text1(Index).MaxLength) Then
'     Mensaje = " Máximo numero de caracteres 16"
'     Estilo = vbOKOnly + vbCritical ' Define los botones.
'     Título = " MENSAJE " ' Define el título.
'     Respuesta = MsgBox(Mensaje, Estilo, Título, Ayuda, Ctxt)
' End If
End Sub
```

```
Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer)
    If Option28.Value = True Or Option29.Value = True Then
        Else
            KeyAscii = SoloNumeros(KeyAscii)
        End If
    End Sub
```

```
Private Sub Text9_Change()
    If Len(Text9.Text) = 2 Then
        'Command1.Enabled = True
    Else
        'Command1.Enabled = False
    End If
```

End Sub

```
Private Sub Text9_KeyPress(KeyAscii As Integer)
    KeyAscii = SoloNumeros1(KeyAscii)
End Sub
```

```
Function SoloNumeros(ByVal KeyAscii As Integer) As Integer
'permite que solo sean ingresados los numeros, el ENTER y el RETROCESO
    If InStr("01", Chr(KeyAscii)) = 0 Then
        SoloNumeros = 0
    Else
        SoloNumeros = KeyAscii
    End If
    ' teclas especiales permitidas
    If KeyAscii = 8 Then SoloNumeros = KeyAscii ' borrado atras
End Function
```

```
Function SoloNumeros1(ByVal KeyAscii As Integer) As Integer
'permite que solo sean ingresados los numeros, el ENTER y el RETROCESO
    If InStr("0123456789", Chr(KeyAscii)) = 0 Then
        SoloNumeros1 = 0
    Else
        SoloNumeros1 = KeyAscii
    End If
    ' teclas especiales permitidas
    If KeyAscii = 8 Then SoloNumeros1 = KeyAscii ' borrado atras
End Function
```

Form 2

```
VERSION 5.00
Begin VB.Form Form2
    Caption       = "Propiedades"
    ClientHeight  = 2085
    ClientLeft    = 60
    ClientTop     = 450
    ClientWidth   = 4680
    LinkTopic     = "Form2"
    ScaleHeight   = 2085
    ScaleWidth    = 4680
    StartupPosition = 2 'CenterScreen
```

Begin VB.CommandButton cmdCancel

Caption = "Cancel"
Height = 300
Left = 3360
TabIndex = 4
Top = 1200
Width = 1080

End

Begin VB.Frame Frame1

Caption = "Velocidad"
Enabled = 0 'False
Height = 870
Left = 120
TabIndex = 2
Top = 960
Width = 2340

Begin VB.ComboBox cboVelocidad

Height = 315
Left = 375
Style = 2 'Dropdown List
TabIndex = 3
Top = 330
Width = 1695

End

End

Begin VB.ComboBox cboPortico

Height = 315
Left = 1080
Style = 2 'Dropdown List
TabIndex = 1
Top = 360
Width = 1425

End

Begin VB.CommandButton cmdOK

Caption = "OK"
Default = -1 'True
Height = 300
Left = 3360
MaskColor = &H00000000&
TabIndex = 0
Top = 555
Width = 1080

End

Begin VB.Label lblPortico

Caption = "Pórtico :"
Height = 315
Left = 240
TabIndex = 5

```

    Top      = 360
    Width    = 615
End
End
Attribute VB_Name = "Form2"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Declare Function CreateFile _
    Lib "kernel32.dll" Alias "CreateFileA" ( _
    ByVal lpFileName As String, _
    ByVal dwDesiredAccess As Long, _
    ByVal dwShareMode As Long, _
    lpSecurityAttributes As SECURITY_ATTRIBUTES, _
    ByVal dwCreationDisposition As Long, _
    ByVal dwFlagsAndAttributes As Long, _
    ByVal hTemplateFile As Long) As Long

Private Declare Function CloseHandle _
    Lib "kernel32.dll" ( _
    ByVal hObject As Long) As Long

Private Type SECURITY_ATTRIBUTES
    nLength As Long
    lpSecurityDescriptor As Long
    bInheritHandle As Long
End Type

Private Const FILE_SHARE_READ = &H1
Private Const FILE_SHARE_WRITE = &H2
Private Const OPEN_EXISTING = 3
Private Const FILE_ATTRIBUTE_NORMAL = &H80

Public Function COMAvailable(iPortNum As Integer) As Boolean
    Dim hCOM As Long
    Dim ret As Long
    Dim sec As SECURITY_ATTRIBUTES

    'Abrir el puerto COMM
    hCOM = CreateFile("COM" & iPortNum & "", 0, FILE_SHARE_READ +
    FILE_SHARE_WRITE, _
    sec, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0)

    If hCOM = -1 Then
        COMAvailable = False
    Else 'Disponible
        COMAvailable = True
    End If
End Function

```

```

        'Cierra el puerto
        ret = CloseHandle(hCOM)
    End If
End Function

Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
Form1.MSComm1.CommPort = cboPortico.ListIndex + 1
Form1.MSComm1.Settings = Trim$(cboVelocidad.Text) & "," & "n" & "," & "8" &
"," & "1"
Form1.lblPortico.Caption = "Com" & Form1.MSComm1.CommPort
Form1.lblvelocidad.Caption = Left$(Form1.MSComm1.Settings,
Len(Form1.MSComm1.Settings) - 6)
Unload Me
End Sub

Private Sub Form_Load()
Dim i As Integer
'For i = 1 To 30
'    If COMAvailable(i) Then
'        cboPortico.AddItem "Com" & Trim$(Str$(i))
'    End If
'Next
For i = 1 To 20
    cboPortico.AddItem "Com" & Trim$(Str$(i))
Next i
cboPortico.ListIndex = Form1.MSComm1.CommPort - 1

cboVelocidad.AddItem "110"
cboVelocidad.AddItem "300"
cboVelocidad.AddItem "600"
cboVelocidad.AddItem "1200"
cboVelocidad.AddItem "2400"
cboVelocidad.AddItem "4800"
cboVelocidad.AddItem "9600"
cboVelocidad.AddItem "14400"
cboVelocidad.AddItem "19200"
cboVelocidad.AddItem "28800"
cboVelocidad.AddItem "38400"
cboVelocidad.Text = Left$(Form1.MSComm1.Settings,
Len(Form1.MSComm1.Settings) - 6)
End Sub

```

