Generador didáctico de códigos de línea

Juan Axel Tenorio Torres
Universidad Autónoma Metropolitana
al2112003006@correo.azc.uam.mx

Mario Reyes Ayala
Universidad Autónoma Metropolitana
mra@correo.azc.uam.mx

Genaro Hernández Valdez
Universidad Autónoma Metropolitana
ghv@correo.azc.uam.mx

Edgar Alejandro Andrade González
Universidad Autónoma Metropolitana
eaag}@correo.azc.uam.mx

Resumen

Un aspecto fundamental en el diseño de sistemas alámbricos de transmisión digital en banda base es la selección del código de línea que se utilizará para codificar la información (representada por una secuencia binaria de unos y ceros lógicos). El presente artículo presenta la integración de los dispositivos eléctricos/electrónicos necesarios para diseñar y construir un generador didáctico de códigos de línea (GDCL). El aparato construido es de uso amigable, con la finalidad de que la atención del estudiante se centre en analizar y comparar las características más relevantes de los diferentes códigos de línea (tales como ancho de banda, deriva en corriente directa, capacidad para proporcionar información de reloj, entre otras). El GDCL es portátil y puede ser usado en un laboratorio en donde se cuente con un osciloscopio (que tenga la opción de calcular la transformada rápida de Fourier) para poder observar y analizar la

secuencia binaria codificada en los dominios del tiempo y de la frecuencia.

Palabra(s) Clave(s): Códigos de línea, instrumento didáctico, transmisión digital en banda base, secuencia binaria.

1. Introducción

Una forma de clasificar a los sistemas de comunicaciones es por su forma de trasmisión: pasa banda o banda base. En la transmisión pasa banda el mensaje modula a una portadora (generalmente senoidal de relativamente alta frecuencia) para trasladar el espectro del mensaje a otras bandas de frecuencia y de esta manera explotar el uso de las mismas. Sin embargo, para nuestros fines nos enfocaremos en la transmisión digital en banda base, en la cual la información (representada por una secuencia binaria) se codifica utilizando algún código de línea y se transmiten directamente en forma de pulsos eléctricos. La aplicabilidad de esta técnica de transmisión se limita, por lo tanto, a medios alámbricos tales como cable coaxial, par trenzado o fibra óptica [1]. El hecho de que los códigos de línea incluyan componentes de baja frecuencia en su espectro al ser transmitidos, es lo que origina el nombre de transmisión digital en banda base [1]. Los códigos de línea presentan ventajas y desventajas unos con respecto a otros al momento de ser aplicados; por lo que se pueden comparar en términos de las siguientes características: ancho de banda que requieren para su transmisión, si presentan o no componente de corriente directa, capacidad para proporcionar información de reloj (útil para mantener la sincronía entre transmisor y receptor), capacidad para detectar errores, costos de implementación, probabilidad de error en presencia de ruido, entre otros. Con esto en mente, en el presente trabajo se reporta el desarrollo de un instrumento electrónico capaz de generar una secuencia binaria codificada utilizando códigos de línea.

La figura 1 muestra el diagrama a bloques de un sistema digital de comunicaciones en banda base. En esta se puede observar cómo los sistemas en banda base hacen uso de bloques tales como la codificación de fuente y codificación de canal. En donde el primer bloque hace referencia a aplicar técnicas para que idealmente se elimine la redundancia producida por la fuente de

información, mientras que el segundo bloque se encarga de agregar redundancia con la finalidad de proteger la información de posibles errores durante la transmisión [2], de esta manera, cuando la señal llegue la GDCD, este pueda codificarla y posteriormente ser transmitida. También podemos observar cómo existe un canal analógico, así como un bloque para la decodificación de canal y otro para la decodificación de fuente. El canal analógico por lo regular se trata de una línea de transmisión cuando se utiliza transmisión utilizando códigos de línea (recuerde que en un sistema en banda base la señal no es compatible con el espacio) y los bloques de decodificación se encargan de realizar los procesos inversos mencionados en la codificación de fuente y de canal.

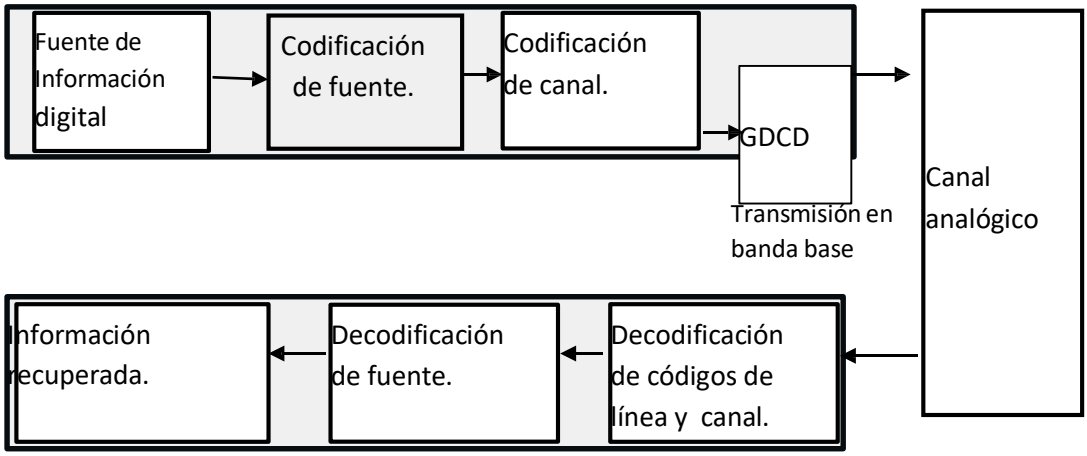

Figura 1 Diagrama simplificado de bloques de sistema digital de transmisión banda base.

2. Códigos de Línea

Los códigos de línea (o formatos de señalización) son técnicas que se aplican sobre alguna señal binaria con la finalidad de hacerla compatible con el medio de transmisión alámbrico [3]. Como se mencionó anteriormente, al no realizarse una modulación, estos códigos son una forma de transmisión en banda base, en donde los '1' y '0' lógicos se pueden representar en diversos formatos de señalización serial de bit. No obstante, cualquiera que sea el formato de señalización, este debe permitir la recuperación de la señal en el receptor. Sin embargo, cómo los unos y ceros son representados con alguna forma de onda específica, esto implica que dicha forma de onda influenciará directamente en

parámetros tales como: la potencia de transmisión, el ancho de banda, facilidad de recuperación del reloj en el receptor, entre otros. Parámetros que deben de tomarse en cuenta al momento de acoplar el formato de la señal con el canal. De esta manera, se puede aplicar algún código de línea para producir una señal serial cuyo espectro concuerde con el ancho de banda del canal [4]. Por ejemplo, el código de línea unipolar, por su sencillez al representar los '1' y '0' lógicos únicamente con dos niveles de voltaje, es uno de los más comunes, muchas veces conocido como secuencia TTL. Cuando se emplean códigos unipolares, se tiene la ventaja de utilizar componentes que requieren únicamente de una fuente de poder para generarlos, por ejemplo, una de 5 Volts para circuitos con tecnología TTL.

En la figura 2(a) se muestra una forma de onda codificada de manera unipolar sin retorno a cero (UPNRZ), en donde previamente se estableció una regla para representar los estados lógicos. Dicha regla se observa en la figura 2(b) y es importante entender que puede cambiar, si así se desea, el papel de unos y ceros, lo que conllevaría a invertir los niveles de voltaje con respecto a los '1' y '0' lógicos. Sin embargo, el código de línea UPNRZ tiene ciertas desventajas, ya que presenta una componente de corriente directa (deriva en C.D.), la cual al ser el promedio de la señal adquiere un comportamiento aleatorio al momento de que la señal es transmitida. Por lo tanto, no hay forma de poder predecir su valor, haciendo que los niveles de voltaje originados en el transmisor sean diferentes a los recibidos en el receptor, algo que es completamente indeseable pues la probabilidad de error aumenta. Por esta razón, surgen los códigos de línea bipolares sin retorno acero (BPNRZ), en donde los niveles de voltaje con los que se representan los '1' y '0' lógicos están representados por niveles positivos y negativos de igual magnitud de voltaje. Con esto se pretende que el promedio de la secuencia codificada sea cero desapareciendo de esta manera la deriva en C.D.

Hasta este punto se han abordado técnicas de codificación en donde se complica la recuperación del reloj en presencia de secuencias largas de unos o de ceros, ya que en estos códigos de línea no se envía ninguna señal

adicional de sincronía. Para hacerle frente a este inconveniente, se definieron los códigos bifase. En la figura 3(a) se muestra uno de los tantos códigos bifase: el código Manchester, cuya regla de codificación se muestra en la figura 3(b). Este código de línea garantiza al menos una transición en la mitad del intervalo de bit, por lo que su capacidad para mantener la sincronía entre transmisor y receptor es alta. Sin embargo, al tener un mayor número de transiciones, el ancho de banda se incrementa aproximadamente al doble respecto al código bipolar convencional.

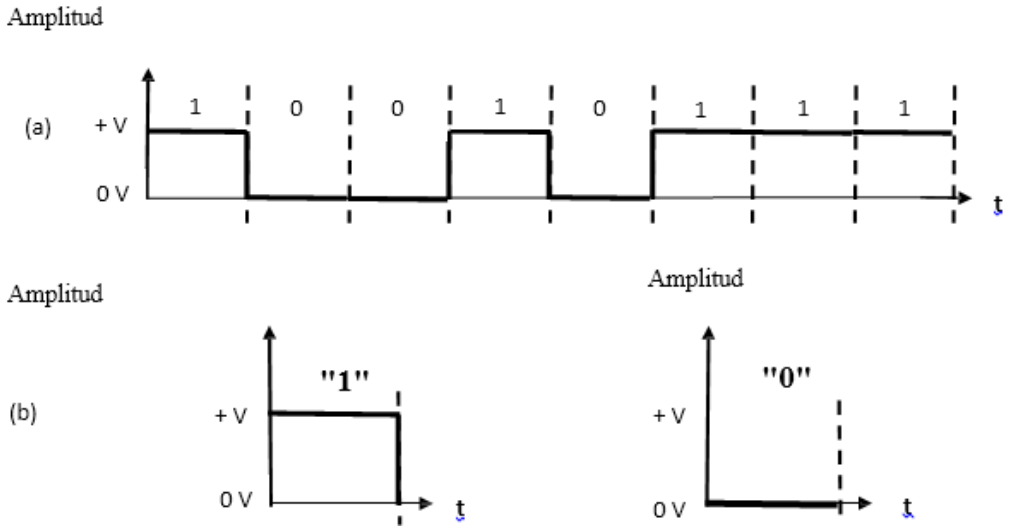

Figura 2 (a) Código unipolar sin retorno a cero. (b) Regla de representación lógica.

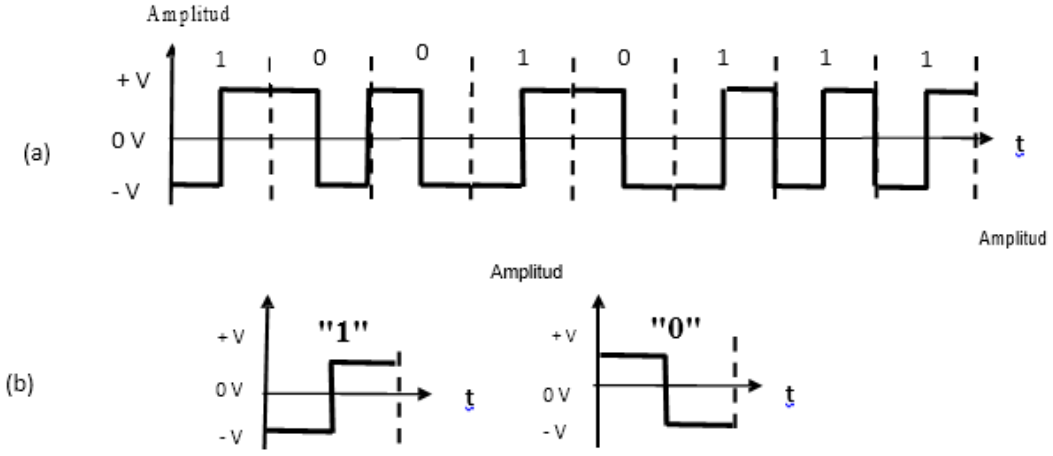

Figura 3 (a) Código Manchester. (b) Regla de representación lógica.

El hecho de que el código Manchester garantice una transición a la mitad del tiempo de bit en beneficio de la sincronización, motivó la propuesta de los códigos de línea con retorno a cero. Los códigos de línea con retorno a cero obedecen las mismas reglas que los códigos que se han explicado hasta el momento, con la diferencia de que a la mitad del intervalo de bit retorna al nivel de 0 Volts. En la figura 4 se ilustra la versión del código bipolar en su versión con retorno a cero (BPRZ).

Por otra parte, tenemos los códigos de línea diferenciales, los cuales surgen por la necesidad de trabajar en ambientes en donde el ruido eléctrico es intenso. Estas técnicas se fundamentan en codificar los cambios del estado lógico (transiciones), de esta manera no importa si los niveles de voltaje son afectados por el ruido externo. Esto es muy útil en sistemas de comunicación que se encuentran principalmente en zonas industriales o en lugares en donde se tengan máquinas de potencia [5][6].

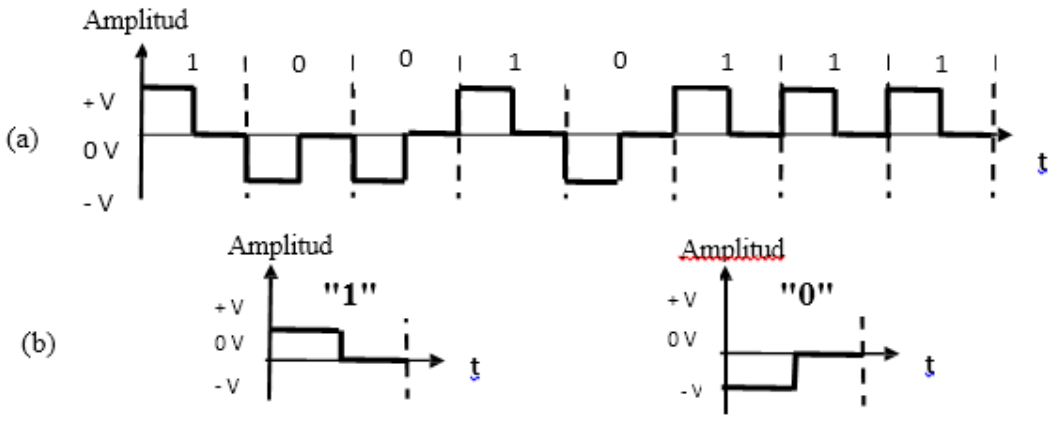

Figura 4 (a) código BPRZ. (b) Regla de representación lógica.

En la figura 5 se ilustra la versión diferencial del código bipolar sin retorno a cero (BPNRZ Dif), donde se nota que hay estados lógicos iguales en donde se tienen distintos niveles de voltaje. En estos códigos de línea se establece previamente una regla, en donde se indique en qué estado lógico realizar un cambio. Por ejemplo, en la figura 5 se observa que en presencia de un '1' binario el nivel de voltaje cambia (ocurre una transición) y cuando se tiene un '0'

binario el nivel de voltaje permanece igual al estado lógico anterior. Esto último indica que se necesita de una condición inicial para que el sistema sepa cómo responder ante el primer dígito binario, pues se debe de especificar el nivel de voltaje a tomar con respecto al estado lógica anterior.

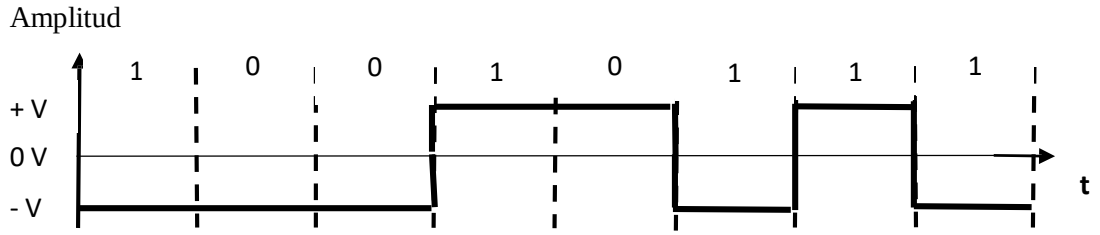

Figura 5 Código bipolar diferencial sin retorno a cero (BPNRZ Dif.).

Es importante notar cómo el dígito binario que hace que cambie el nivel del voltaje depende de cómo se defina el sistema con antelación (condición inicial), de esta manera podemos tener variantes de todos los códigos de línea (a excepción de los multinivel), respetando las características con las que cuenta cada uno, solo que ahora haciéndolo inmune al ruido eléctrico.

Los códigos de línea que se abordan en el generador didáctico de códigos de línea son los siguientes:

- Unipolar sin retorno a cero (UPNRZ) y con retorno a cero (UPRZ).
- Bipolar sin retorno a cero (BPNRZ) y con retorno a cero (BPRZ).
- Código Manchester.
- Código de pulsos alternados en presencia de unos lógicos y nivel cero en presencia de ceros lógicos (conocido como código AMI).
- Códigos multinivel.

También se consideran las versiones diferenciales de los tres primeros códigos de la lista anterior.

3. Componentes del Generador didáctico de códigos de línea

El desarrollo del GDCL constó principalmente de dos etapas, en la primera se realizó el diseño del hardware y en la segunda se desarrolló e implementó el

software para realizar las diferentes codificaciones. Debido a que en esta sección nos enfocamos en los componentes utilizados para llevar a cabo la construcción el GDCD, en la figura 6 se observa el diagrama a bloques general del prototipo mencionado, así como de los componentes utilizados.

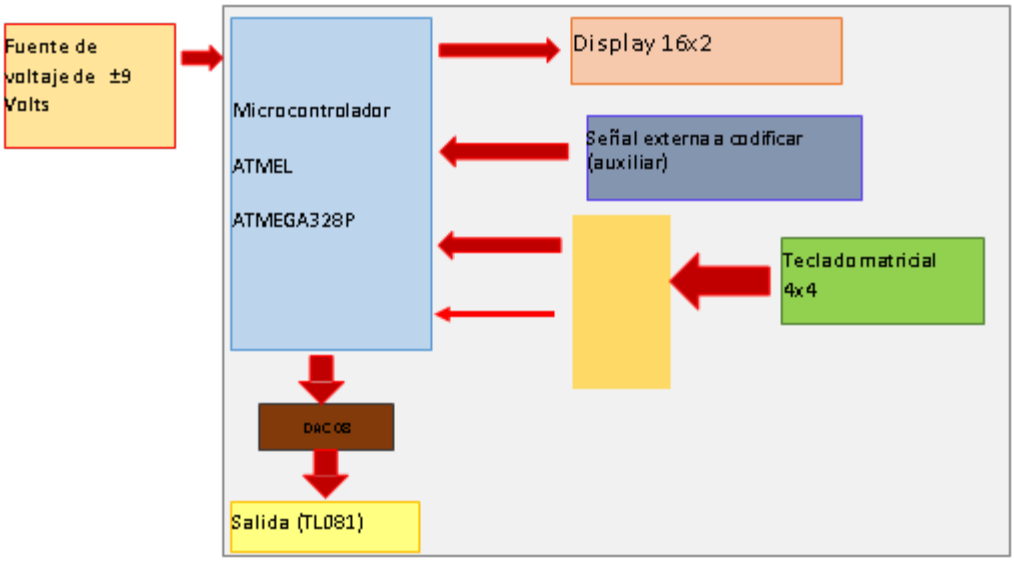

Figura 6 Diagrama general a bloques del generador didáctico de códigos de línea.

El *microcontrolador ATMEGA328P*, perteneciente a la familia ATMEL, es la unidad de control y procesamiento de todo el sistema, el cual se encarga de procesar y generar en banda base la señal a codificar, según haya sido el tipo de código de línea seleccionado por el usuario. Entre sus principales características están las de tener una memoria flash de 32 kB, así como también una RAM estática de 2 kB y una EEPROM de datos de 1 kB, suficiente para cumplir con nuestro objetivo y codificar señales usando los códigos de línea de mayor importancia en los sistemas de comunicaciones actuales. Cuenta también con tres puertos (puerto B, C y D), de los cuales sólo los puertos B y D pueden ser usados como entradas o salidas; el primero, cuenta con 6 pines, mientras que el segundo está conformado por 8 pines, de los cuales dos pueden ser ocupados para generar interrupciones; el puerto C es un puerto analógico, diseñado para que por medio de los 6 pines que lo conforman solamente se lean señales. La manera en cómo fueron utilizados los puertos en este proyecto fue la siguiente:

- *Puerto B*: programado como salida, este puerto manda al convertidor digital a analógico (DAC) una señal de 6 bits que contiene la información de la señal ya codificada.
- Puerto C: En este puerto se ocuparon solamente los pines 2,3,4 y 5, ya que es el encargado de leer una señal de solo 4 bits proveniente del decodificador de teclado y la cual contiene información de la tecla que se presionó.
- Puerto D: Este puerto fue configurado para controlar la pantalla de cristal líquido (LCD) por los pines 0,1,4,5,6 y 7; mientras que los pines 2 y 3 fueron usados como interrupciones, una para avisar que se presionó una tecla y leer el puerto C y la otra para leer una señal binaria generada por alguna fuente externa, respectivamente.

La figura 7 ilustra de manera global el funcionamiento del microcontrolador, visto como una caja negra que procesa la información y que genera salidas con respecto a lo que se tiene a la entrada.

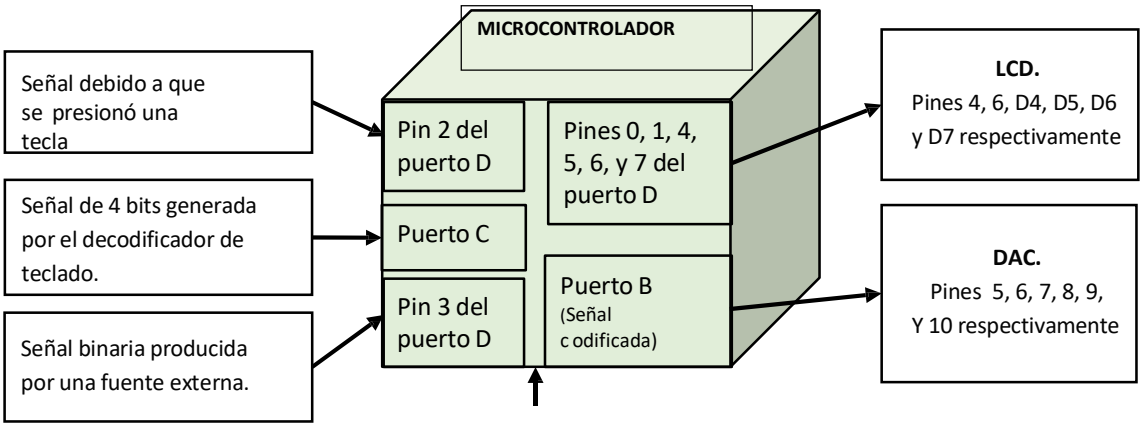

Figura 7 Vista general de cómo el ATMEGA328P-PU está configurado para atender las peticiones y mandar las salidas.

Como se observa en la figura 7, el microcontrolador cuenta con un módulo extra que le permite funcionar, este es la señal de reloj. Para este fin, se utilizó un oscilador de cristal de 16 MHz, conectado en paralelo a una resistencia de 1MΩ y

a los pines 9 y 10 (XTAL1 y XTAL2) del ATMEGA328P junto con dos capacitores de 15 pf, cada uno entre un pin y tierra.

Para el bloque del *decodificador de teclado*, se utilizó el dispositivo 74C922, el cual es el encargado de eliminar los rebotes generados al momento de presionar un botón del teclado matricial. Su modo de trabajo es el siguiente: Cuando el decodificador detecta que se ha presionado alguna tecla, este solo hace caso a la primera señal que es la que contiene la información deseada, despreciando a las que puedan generarse de manera indeseada por las vibraciones del botón. Esto lo hace gracias a los capacitores de 100 nf y 1 µf, cuidadosamente seleccionados, conectados a los pines 5 y 6 respectivamente del decodificador. Una vez que ya se tiene la señal de 8 bits del teclado, el 74C922, a través del pin 12, genera una señal, la cual avisa al microcontrolador que hay una petición por el usuario; dicha petición es de 4 bits y se manda al *ATMEGA328P-PU,* quien ya se encuentra a espera de dicha señal. Haciendo uso de interrupciones, el microcontrolador lee la señal generada por el bloque en el que estamos. Esta señal lleva información de la tecla que fue presionada, ante este evento, el controlador procesa la información y responder ante la petición del usuario. En la figura 8, se ilustra el diagrama a bloques del decodificador de teclado.

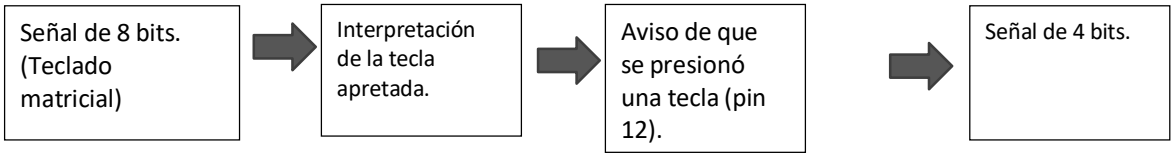

Figura 8 Diagrama a bloques del funcionamiento del 74C922.

Por otra parte, l*a pantalla de cristal líquido 16x2 (LCD 16X2),* es un dispositivo que solamente figura como esclavo, pues no genera ninguna señal hacia el microcontrolador, ya que es el encargado de mantener una interfaz gráfica con el usuario. A través de esta pantalla se mostrarán tanto las opciones de códigos de línea que se pueden implementar cómo las opciones de la secuencia binaria a codificar: i) secuencia generada de forma externa, ii) secuencia generada internamente o iii) el usuario define la señal a codificar por medio del teclado.

Para el bloque del *convertidor digital analógico (DAC)*, se utilizó el DAC08, el cual

es un convertidor de 8 bits, con un tiempo de respuesta de 100 ns, un consumo de 33 mW a ±5 Volts, entre otras características. Por lo que estas lo hacen un dispositivo adecuado para nuestros fines; ya que el tiempo máximo de bit generado por el codificador didáctico es de 1 ms, siendo lo suficientemente lento como para que el DAC responda ante la señal de 6 bits proveniente del microcontrolador. Las dos terminales restantes (el DAC es de 8 bits) simplemente se mandan a tierra. Entonces, este bloque es el encargado de darle una interpretación analógica a la señal digital codificada proveniente del microcontrolador y la cual permitirá tener flexibilidad en los niveles de voltaje de los códigos de línea generados. En la figura 9 se muestra el diagrama a bloques del funcionamiento del DAC08.

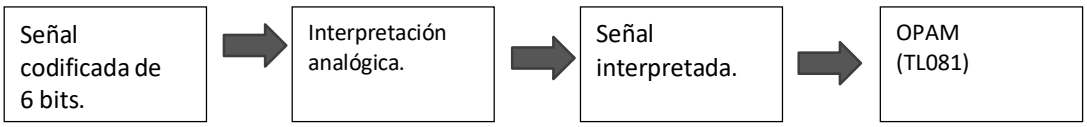

Figura 9 Diagrama a bloques del funcionamiento del DAC08.

En la figura 6 se observa que existe un bloque dedicado a leer una señal externa, este es señalado como auxiliar debido a que puede o no intervenir en el sistema. Esto es en caso de contar con un generador de funciones o bien, con alguna fuente digital que entregue una secuencia binaria a codificar. Sin embargo, el generador didáctico de códigos de línea tiene la capacidad de poder generar una señal de manera interna y así no tener la necesidad de tomar alguna muestra externa para que el dispositivo pueda trabajar.

En el bloque de salida se utilizó el *amplificador operacional (OPAM)* TL081, debido a su bajo consumo de potencia, el cual tiene un ancho de banda de 3 MHz y entre sus características más importantes se encuentra que al ser un amplificador con tecnología JFET, tiene una alta impedancia de entrada, así como también cuenta con una tasa de crecimiento (“slew rate”) alta. Este dispositivo se utilizó con la finalidad de tener a la salida una señal simétrica binaria; ya que a la salida de este se puede observar la señal resultante. Es gracias a este bloque que se pueden hacer mediciones y comparaciones de

manera visual con los distintos tipos de códigos de línea generados, los cuales se usan en transmisión digital en banda base en diferentes sistemas alámbricos de comunicaciones [7]. Una parte importante de todo el sistema es el bloque dedicado al *teclado matricial 4x4*, debido a que por medio de éste, el usuario puede hacer la selección de lo que se desea realizar. Haciendo uso de botones, se puede acceder a generar un código de línea específico o simplemente permite que la persona pueda cancelar la operación, así como también cambiarla por otra de las opciones que el sistema ofrece. Este teclado genera una señal de 8 bits, la cual es procesada por el decodificador 74C922 para después poder mandarla al microcontrolador, tal y como se ilustra en la figura 8.

Por último se cuenta con el bloque de *fuente de alimentación*, el cual es el encargado de proveer energía a todo el sistema, tal y como se ilustra en la figura 6. Como lo que se desea es que el generador didáctico de códigos de línea sea portátil, este bloque está conformado por dos baterías de 9 volts o bien, tiene la opción de conectarse a una fuente de corriente directa, para de esta manera suministrar la suficiente energía a todo el sistema.

4. Diseño y desarrollo del software

Las diferentes funciones con las que consta el generado didáctico, fueron programadas en el ATMEGA328P, haciendo uso de la plataforma de Arduino [8]. A continuación, se explican las librerías que se utilizaron en este proyecto. Mediante diagramas de flujo, se ilustra el procedimiento a seguir para generar algunas de las subrutinas (los códigos de línea) que integran a este prototipo. En la figura 10 y de manera global se muestra el diagrama de flujo de todo el programa, en donde se puede observar cómo es la secuencia en que el microcontrolador procesa y genera las peticiones demandadas por el usuario, así como también las configuraciones que se le hicieron para poder atender a los periféricos.

Librerías

Las librerías que se utilizaron fueron *TimerOne.h* y *LiquidCrystal.h*; la primera nos

ayuda a generar los pulsos, lo que significa que dentro de esta interrupción que se procesa cada determinado tiempo (desde 1ms hasta 1s) se codifican los bits. Dentro del programa, esta interrupción es llamada *pulso* y el usuario tiene la posibilidad de cambiar el periodo con el que se codifica un bit de la señal binaria, modificando de esta manera el tiempo de bit mediante la instrucción *Timer1.initialize(A)*, en donde A es el tiempo en microsegundos que tarda en entrar a la interrupción *pulso*. El nombre de esta interrupción puede variar, esto se hace desde un inicio pues este campo solo se configura una sola vez en todo el programa; en la figura 11 se muestra el diagrama de flujo de esta primera librería, figurando el procedimiento que se sigue para generar los códigos de línea.

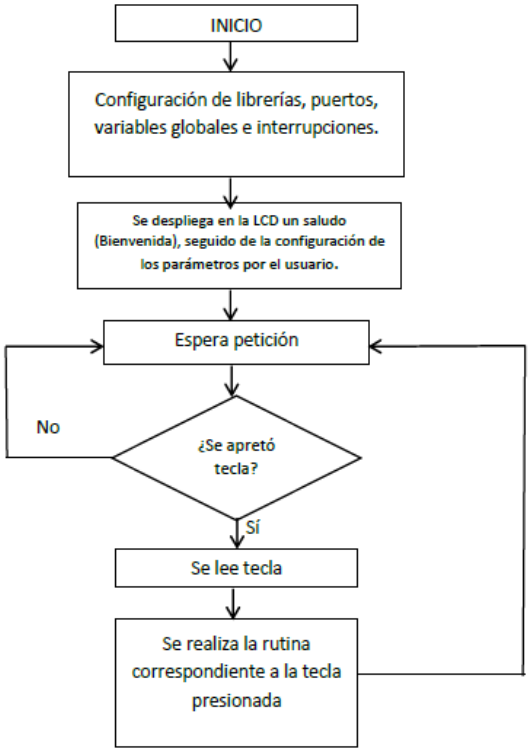

Figura 10 Diagrama de flujo de todo el sistema visto de manera global.

La segunda librería mencionada, solo nos ayuda a configurar los puertos para poder controlar la LCD; ya que por medio de las instrucciones *lcd.setCursor(A,B), lcd.print("mensaje")* y *lcd.clear()* respectivamente, se puede colocar el cursor en alguna parte de la pantalla (siendo A la columna y B la fila) o mandar a imprimir algún mensaje (siendo mensaje la frase a imprimir) o

también borrar todo lo que se encuentre impreso en la LCD. Es importante mencionar que en la sección en donde se configuran los parámetros de la LCD, se debe de señalar bajo la instrucción *lcd.begin(X,Y)* el tipo de pantalla que se va a utilizar, en donde *X* son las columnas y *Y* los renglones con los que el dispositivo cuenta. Por lo tanto, cómo para este proyecto se usó una LCD de 16x2, la instrucción se escribió como *lcd.begin(16,2).*

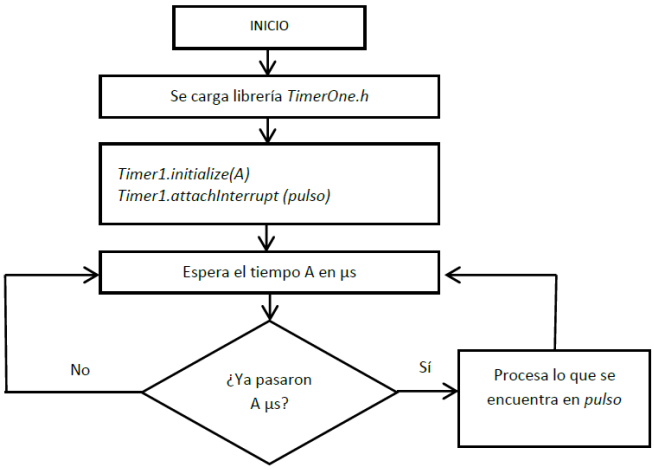

Figura 11 Diagrama de flujo de la librería TimerOne.h.

Códigos de línea

Antes de pasar a explicar la manera de codificar las señales, es importante tener en cuenta las siguientes variables principales:

- *amplitud_positiva*: es una variable de 6 bits, la cual contiene información del nivel de voltaje positivo que se debe de tener a la salida (ver figura 9).
- *amplitud_negativa*: en esta variable de 6 bits se tiene la información del nivel de voltaje negativo que se debe de presentar a la salida.
- *indicador_pulso*: Es la variable que contiene la información de cual tipo de código aplicar, gracias a esta, se puede tener accesos a las diferentes secciones de la interrupción pulso.
- *valor*: Contiene información del tipo de bit a codificar, es decir, si es un '1' o '0' lógico.
- *PORTB*: Es una instrucción que asigna un nivel de voltaje a cada pin del puerto B con respecto a un número binario.

Código UPNRZ y BPNRZ

Estos dos códigos son muy similares en el sentido de programación, debido a que solamente se modifica el voltaje negativo. Por ejemplo, en UPNRZ el voltaje negativo es cero, mientras que en BPNRZ es -5 Volts.

El diagrama de flujo que se muestra en la figura 12, ilustra la secuencia del código de programación para codificar señales unipolares o bipolares sin retorno a cero.

Recordar que para haber llegado a este punto, se necesitó que el usuario indicara que se desean aplicar códigos unipolares o bipolares y esto en el programa se traduce como haber señalado por medio de una variable que se desea leer una secuencia dentro de la interrupción "*pulso*".

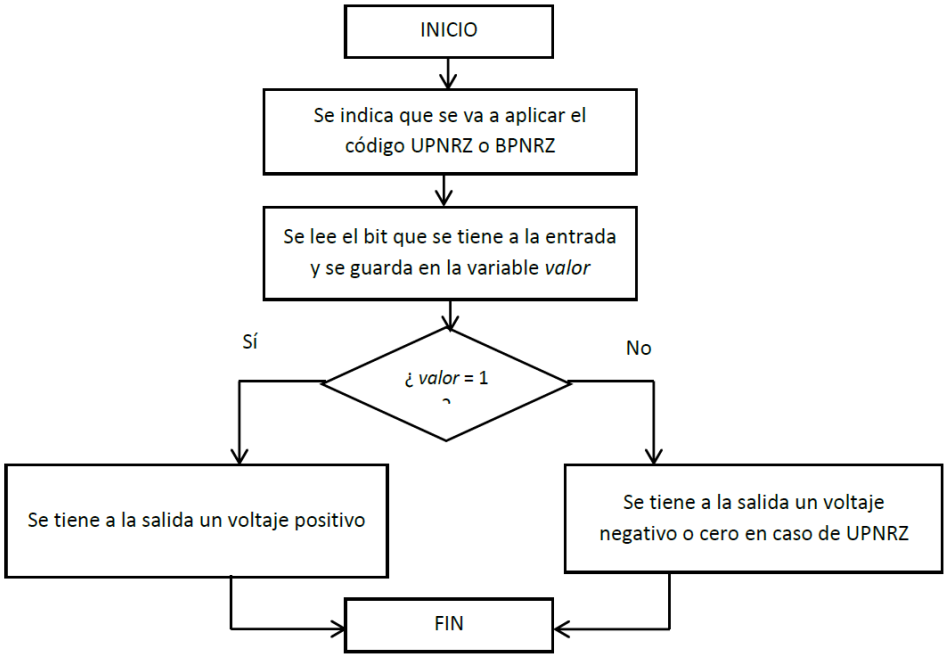

Figura 12 Diagrama de flujo de los códigos unipolar y bipolar sin retorno a cero.

Código Manchester o Bifase:

La manera en la que el programa genera los pulsos ya viene precargada, estos es, si a la entrada tenemos un '1' lógico, en la primera mitad del intervalo de bit se observará una amplitud positiva y durante el resto del intervalo de bit se observará una amplitud negativa; caso contrario para cuando se tiene un '0' lógico. Sin embargo, estos parámetros no son fijos, ya que el generador didáctico de

código de línea permite modificarlos de acuerdo a alguna conveniencia en particular. En la figura 13 se observa el diagrama de flujo para generar los códigos Manchester.

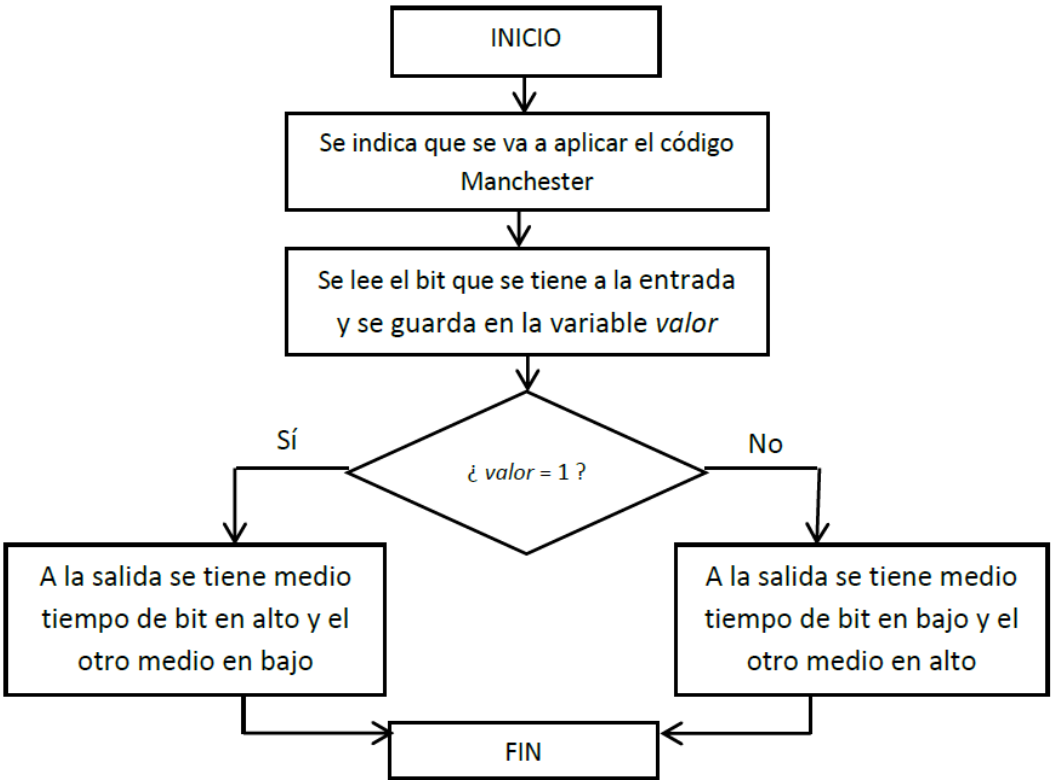

Figura 13 Diagrama de flujo del código Manchester.

Código UPNRZ-Dif y BPNRZ-Dif

Para generar los códigos de línea diferenciales, el usuario no solo debe de dar la instrucción, sino que también especificar la condición inicial. Una vez proporcionado el estado anterior, el programa empezará a leer bit a bit la secuencia que se tiene a la entrada, de esta manera realizará la codificación en tiempo real.

Por omisión, esta sección está diseñada para que se realice un cambio de pulso cuando a la entrada se presente un '1' lógico, en cambio, cuando se lea un '0' lógico la señal permanecerá igual, lo que significa que no se presentará un cambio de fase. En La figura 14 se muestra el diagrama de flujo para generar los códigos UPNRZ-Dif y BPNRZ-Dif.

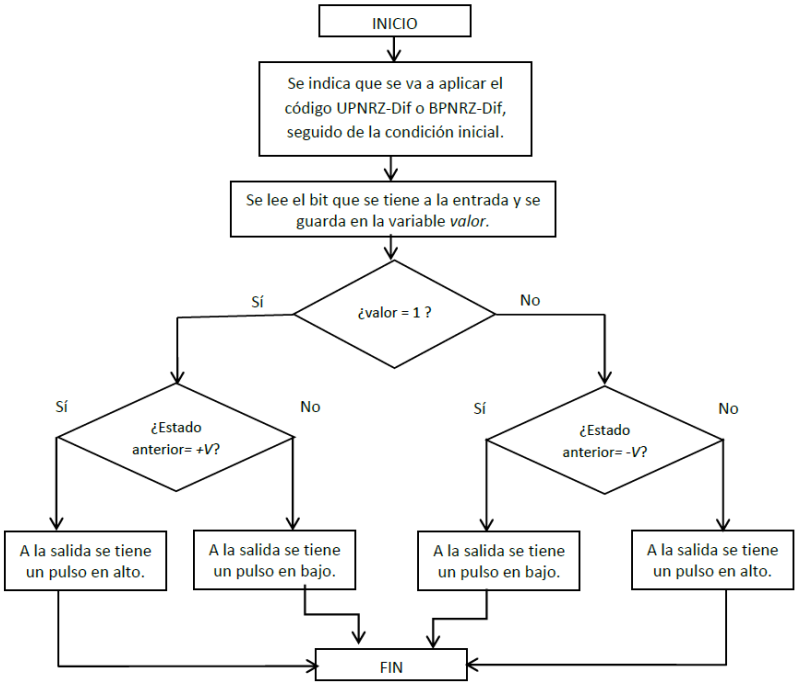

Figura 14 Diagrama de flujo del código UPNRZ-Dif y BPNRZ-Dif, en donde cambia la condición del pulso cuando se tiene un '1' lógico a la entrada.

5. Resultados

La implementación del hardware junto con el software descritos en la sección anterior, dieron como resultado el generador didáctico de códigos de línea que se muestra en la figura 15, donde se observa que ya se tiene un prototipo completo para ser usado en laboratorios y realizar pruebas de codificación sobre diferentes secuencias binarias.

Figura 15 Diferentes ángulos del Generador didáctico de códigos de línea terminado.

En la figura 16, se ilustra a todo el equipo trabajando en conjunto, en donde se observa que se está haciendo uso de una fuente de voltaje de ±9Volts, con la cual se alimentó al generador didáctico de códigos de línea y de un osciloscopio (el

cual tiene la opción de obtener la transformada rápida de Fourier del código de línea presente a la salida del generador didáctico) para observar la señal binaria codificada en tiempo y en frecuencia.

Figura 16 Generados didáctico de códigos de línea trabajando en conjunto con el equipo de laboratorio.

Puesta a prueba

Para comprobar que el generador didáctico de códigos de línea está trabajando de manera adecuada, se sometió a probar que de manera interna genere una señal binaria y que a la salida entregue dicha señal codificada. En las siguientes figuras se muestran los resultados de haber codificado las secuencias binarias 1010101010101010 y 10010111 utilizando diferentes códigos de línea. Se pueden observar resultados en el dominio del tiempo y en el dominio de la frecuencia, en los que se utilizó un intervalo de bit de 1 ms. Los cursores que se llegan a apreciar en el dominio del tiempo, marcan el inicio y el fin de un periodo de la señal generada, mientras que los cursores que se llegan a apreciar en el dominio de la frecuencia, indican la frecuencia fundamental y el primer armónico de la señal generada.

En la figura 17 se observa la transformada rápida de Fourier de la señal generada cuando la secuencia de unos y ceros lógicos es codificada con el código bipolar sin retorno a cero (BPNRZ). La señal resultante es una señal cuadrada periódica con periodo fundamental de 2 ms (frecuencia fundamental de 500 Hz). En la figura 17 se aprecia claramente cómo la frecuencia fundamental

se encuentra a 500 Hz y como la señal codificada es una señal cuadrada, la siguiente componente frecuencial aparece a los 1500 Hz. Es importante notar cómo en estos códigos aparece una componente de C.D. vista al inicio de la gráfica.

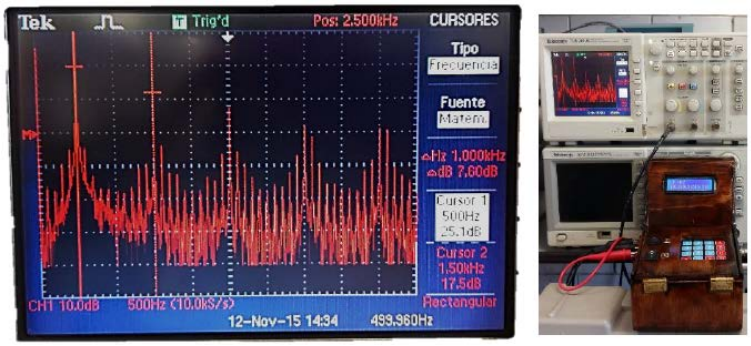

Figura 17 Señal TTL vista en el dominio de la frecuencia, en donde su componente fundamental se encuentra a de 500 Hz.

En la figura 18 se muestra un código Manchester aplicado a una secuencia de la figura 19 muestra la forma de onda en el dominio del tiempo y espectro de frecuencias del código Manchester aplicado ahora a la secuencia binaria 10010111. Debido a que dicha señal es periódica con periodo fundamental igual a 8 ms (el intervalo de bit es igual a 1ms), esta tiene una frecuencia fundamental de 125 Hz, lo que se puede corroborar en la gráfica de la transformada rápida de Fourier del código Manchester, la cual se ilustra en la parte superior izquierda de la figura 20.

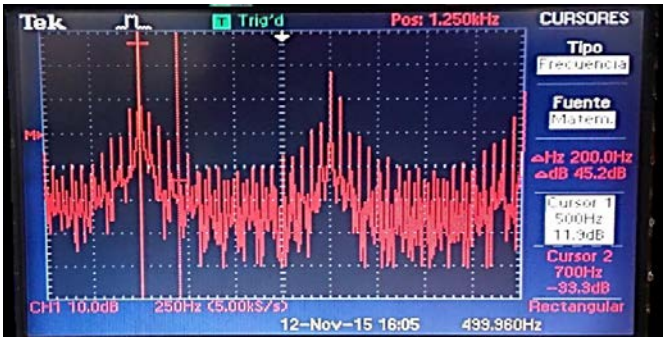

Figura 18 Vista en el osciloscopio y en el dominio de la frecuencia de un tren de pulsos codificado con el código de línea Manchester.

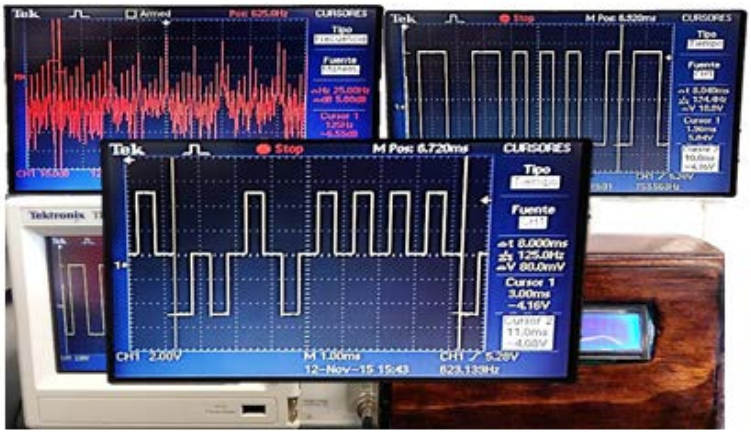

Figura 19 Osciloscopio de la secuencia binaria 10010111 generada por generador didáctico de códigos de línea.

Figura 20 Dominio del tiempo de señal 10010111 codificada con código de línea BP-RZ

Otras secuencias

Para que la señal codificada sea de fácil reconocimiento, se usó ahora una secuencia binaria de 5 bits, dicha secuencia tiene la forma lógica 11000, la cual fue generada de manera interna por el generador didáctico de códigos de línea. En este caso la señal generada es periódica con periodo fundamental igual a 5 ms (el intervalo de bit es igual a 1ms), por lo tanto, tiene una frecuencia fundamental de 200 Hz. A continuación se muestran las imágenes tomadas del osciloscopio de la secuencia binaria codificada utilizando los siguientes códigos de línea: UPRZ y BPRZ (figura 21), BPNRZ-Dif y Manchester-DIF (figura 22), multinivel de 4 y 8 niveles (figura 23) y código AMI (figura 24).

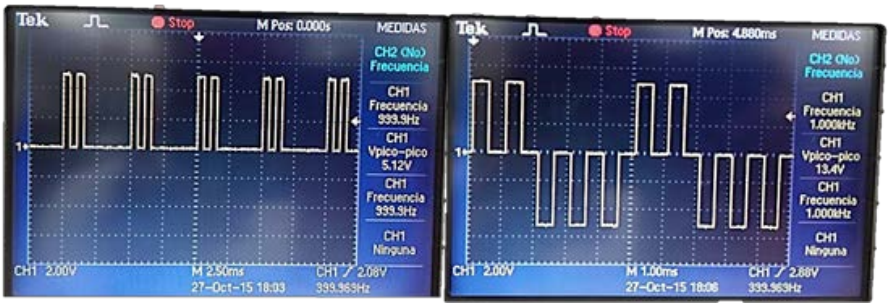

Figura 21 Osciloscopio para la secuencia binaria 11000, codificada con los códigos UP-RZ y BP-RZ respectivamente.

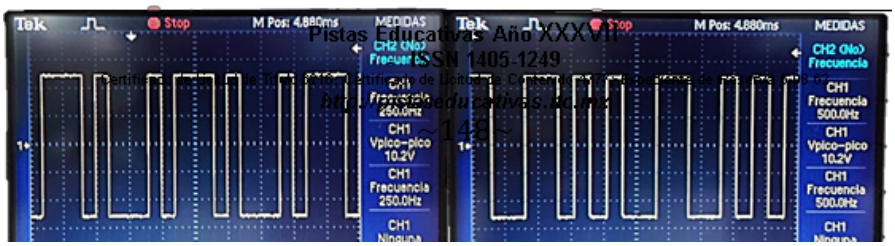

Figura 22 Osciloscopio para la secuencia binaria 11000, codificada con los códigos BPNRZ-Dif y Manchester-DIF respectivamente.

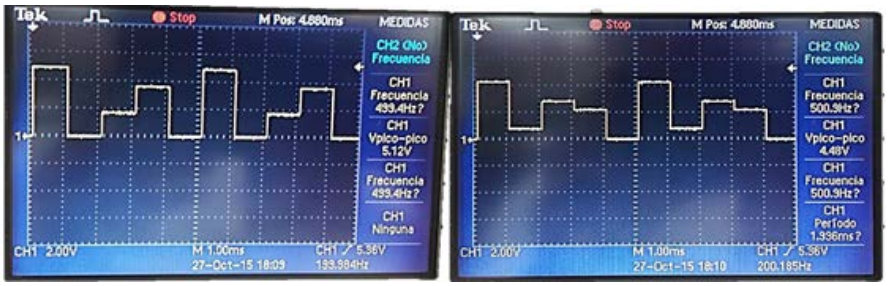

Figura 23 Osciloscopio para la secuencia binaria 11000, codificada con los códigos Multinivel de 4 y 8 niveles, respectivamente.

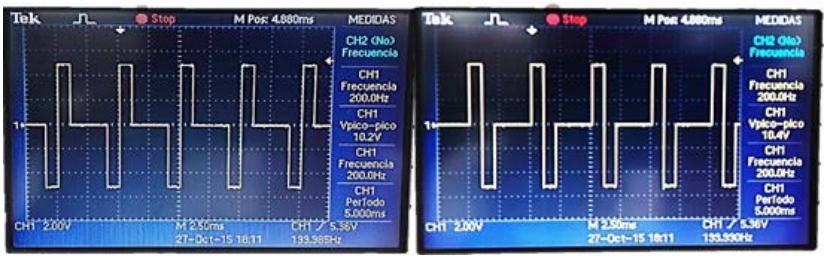

Figura 24 Osciloscopio para la secuencia binaria 11000, codificada con los códigos AMI con condición inicial de +V y –V, respectivamente. Donde cambia de polaridad en presencia de un ‘1’ lógico.

6. Discusión

La intención didáctica del prototipo desarrollado en el presente trabajo se centra en que el alumno observe, analice y compare de manera gráfica y en tiempo real las características (en los dominios del tiempo y de la frecuencia) con las que cuentan los diferentes códigos de línea. El prototipo desarrollado permite que una secuencia binaria de especial interés (por ejemplo, secuencias de puros ceros, unos y ceros alternados, puros unos, entre otras) definida por el usuario a través del teclado matricial, sea codificada con un determinado código de línea. El dispositivo también permite codificar una secuencia binaria proveniente de una fuente de información digital externa. Dentro de las pruebas realizadas, se observó cómo es que una señal unipolar presenta una componente de C.D., mientras que un código Manchester no, esto debido a que el promedio de la señal es la que define dicha deriva y en los códigos bifase, este promedio es cero. También se pudo comparar el contenido espectral de diversos códigos de línea (unipolar con y sin retorno a cero, bipolar con y sin retorno a cero, Manchester, AMI, entre otros) y comprobar experimentalmente que los códigos bifase requieren de un mayor ancho de banda para su transmisión con respecto a los códigos unipolares y bipolares).

Las secuencias binarias utilizadas son periódicas, por lo que el espectro de la señal codificada resultó ser discreto y determinístico con componentes en la frecuencia fundamental y en sus armónicos, lo cual se comprobó en las figuras 17, 18 y 19. Sin embargo, en dichas figuras se observaron componentes espectrales (de amplitud no despreciable) localizadas entre las diferentes componentes armónicas del espectro de la señal generada. Estas componentes se deben a que los pulsos generados no son exactamente cuadrados, pues poseen tiempos de subida y bajada finitos y diferentes de cero, además de que la cresta de los pulsos no es exactamente plana.

Dada la importancia de los códigos de línea para codificar secuencias binarias para su transmisión en medios alámbricos, es de suma importancia que el alumno pueda estudiar de manera analítica y experimental las limitaciones y bondades con las que cuentan, ya que los códigos de línea se aplican dependiendo del tipo

de aplicación, longitud del enlace de comunicación así como de las condiciones de ruido e interferencia en las que va a operar el sistema de comunicación.

En la práctica, con la finalidad de disminuir el ancho de banda requerido para transmitir la señal codificada, se utilizan pulsos con formas diferentes a la cuadrada [1]. Esto se logra a través de un filtro conformador de pulsos (por ejemplo el tipo coseno elevado), el cual permite representar los diferentes códigos de línea con pulsos cuyas transiciones son más suaves (pulsos tipo "sampling") [6]. Como trabajo a corto plazo, se pretende incluir la opción de utilizar un conformador de pulsos al prototipo desarrollado en el presente trabajo.

7. Referencias

[1] Bellamy J. C. (2000). Digital Telephony. 3rd Edition, Editorial JOHN WILEY & SONS.

[2] ANDERSON J. B. and MOHAN, S. Source and Channel Coding: An Algotrithmic Approach. Kluwer Academic Press. Boston, 1991.

[3] Jayant, N.S. and Noll, P. (1984). Digital Coding of waveforms: Principles and applications to speech and video. PRENTICE HALL, Englewood Cliffs, N.J.

[4] Clark, Jr. G.C. and Cain, J. B. (1981) Error-correction Coding for Digital Communications. PLENUM PUBLISHERS, New York.

[5] (CNSO), C. N. (2000). Fundamentos básicos de las telecomunicaciones. Madrid, España.

[6] Leon W. Couch, I. (2008). Sistemas de comunicación digitales y analógicos. 7ª Edición. México: pearson education.

[7] Floyd, T. L. (2008). Dispositivos electrónicos. 8ª Edición. En *Dispositivos electrónicos* (págs. 593-629). México: pearson educación.

[8] Torrente, Ó. (2013). *ARDUINO. Curso práctico de formación.* 1ª Edición. México: Alfaomega.