

UTILIZACIÓN DEL ORDENADOR EN EL LABORATORIO DE TECNOLOGÍA DE LAS COMUNICACIONES

Gómez, Isabel⁽¹⁾; Molina, Alberto⁽¹⁾; Medina, A. Verónica⁽¹⁾ y Esteban, A.
Mariano⁽²⁾

⁽¹⁾Departamento de Tecnología Electrónica, Universidad de Sevilla
Facultad de Informática. Campus Universitario de Reina Mercedes.
41012 Sevilla

tfno: 954552787, fax: 954552764, email: igomez@cica.es

⁽²⁾Guadaltel S.A.

Pastor y Landero 19
41001 Sevilla

tfno: 954562540, email: amer@guadaltel.es

RESUMEN

En este trabajo se describen dos prácticas de laboratorio ideadas para clarificar los conceptos que se imparten en las clases de aula de la asignatura Tecnología de las Comunicaciones. Dicha asignatura, pertenece al tercer curso de la titulación de Informática. La nota común de estos trabajos es la utilización del ordenador junto con un circuito electrónico para la realización de funciones que se requieren en los procesos de transmisión de datos.

1. INTRODUCCIÓN

La asignatura Tecnología de las Comunicaciones está centrada principalmente en los aspectos físicos de las comunicaciones, aunque tiene una parte final dónde se introducen conceptos de diseño de protocolos. Podríamos plantear un laboratorio de prácticas muy adecuado para esta asignatura que consistiría en el montaje de diversos circuitos electrónicos sin necesidad de introducir el ordenador. Sin embargo, hemos comprobado que el hecho de introducirlo supone un incremento en la motivación del alumno a la hora de realizar las prácticas.

2. PRIMERA PRÁCTICA: MODULACIÓN FSK

Esta práctica está diseñada para que el alumno comprenda los conceptos de modulación. Para ello se va a realizar una transmisión de datos entre dos ordenadores utilizando la radio. En el caso de la radio la modulación por desplazamiento de frecuencia [1] es una de las técnicas más empleadas para transformar un valor digital (1,0) en una señal enviable por un transmisor. En nuestra práctica el valor 1 o "mark" se asocia con una frecuencia cercana a los 1445 Hz mientras que al 0 o "space" se le asocia el valor de 1275 Hz.

2.1 Transmisiones FSK, el programa HAMCOMM

En el mundo de la radioafición hay multitud de aplicaciones [2,3,4] para realizar transmisiones de datos tanto de fichero de textos como de imágenes. Entre ellas destaca una realizada por el radioaficionado alemán W.F.Schöder denominado HAMCOMM [5], que con la ayuda de un pequeño circuito electrónico realiza la demodulación y modulación de las señales FSK.

A continuación se va a describir el funcionamiento del programa estudiando los circuitos necesarios para recibir y transmitir, así como algunas funciones especiales del mismo. Para realizar el proceso de demodulación se emplea un circuito electrónico cuya función es la adaptación de niveles de tensión entre la señal de audio proveniente del transceptor y el ordenador. Dicho circuito no necesita alimentación exterior, ya que ésta, se toma del propio puerto serie. La señal senoidal recibida, es convertida en una onda cuadrada de la misma frecuencia, disponible en la patilla DSR del puerto serie. Las transiciones entre ceros y unos generan interrupciones que emplea el programa tanto para determinar la frecuencia de la señal recibida como para poder hacer corresponder esta con un valor binario (1 ó 0).

El proceso de modulación se realiza básicamente por el programa generando una señal cuadrada de la frecuencia correspondiente al bit a transmitir. Esta señal es sometida a un tratamiento de filtrado (filtros paso bajo) y atenuación que permiten acomodar la señal de audio obtenida para atacar la entrada de micrófono del transceptor. Conjuntamente con la generación de ondas de audio existe un proceso de activación del modo transmisión, para lo cual un transistor pone a tierra la señal de PTT (del inglés "Press To Talk"). Esta señal debe ser activada un momento antes de enviar la de audio, para permitir que se establezca la onda portadora antes de ser modulada

2.2 El programa HAMCOMM funcionando

Seguidamente se describen algunas de las pantallas del programa, señalando aquellas que puedan ser interesantes para el análisis de la codificación FSK. Al arrancar el programa aparece la pantalla de la figura 1, donde se observa un menú en la parte superior con los bloques de opciones disponibles. Seleccionando la opción *Mode* se despliega un submenú donde se encuentra la selección de protocolos, la ventana de diálogo (transmisión recepción), así como un conjunto de ventanas útiles para el análisis de la señal recibida por el transceptor.

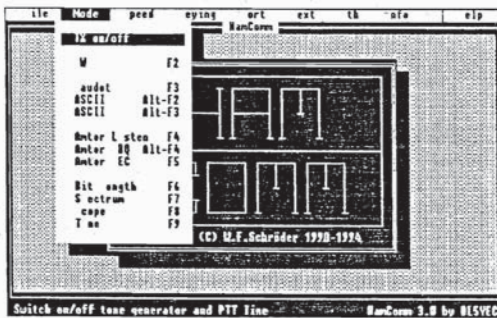


Figura 1: Pantalla inicial del programa

Seguidamente se hace un breve repaso de estas opciones. La primera opción corresponde al paso de modo transmisión a recepción y viceversa, permitiendo la conmutación del PTT de

forma manual, esto será necesario en los modos CW (morse), BAUDOT, ASCII-7, y ASCII-8. En los modos Amtor la conmutación es automática. A continuación vienen cuatro opciones muy interesantes, la primera denominada *Bit Length* (longitud de bit) permite determinar la velocidad de transmisión de la señal que se está recibiendo. En las dos figuras siguientes se presenta el espectro de frecuencia y la traducción a señales digitales de un fragmento de transmisión digital en el formato BAUDOT. En la figura 2, obtenida con la función Spectrum, se puede observar como existen zonas donde se concentran las frecuencias de las señales, correspondiente a 1300 Hz ("space") y 1470 Hz ("mark") aproximadamente.

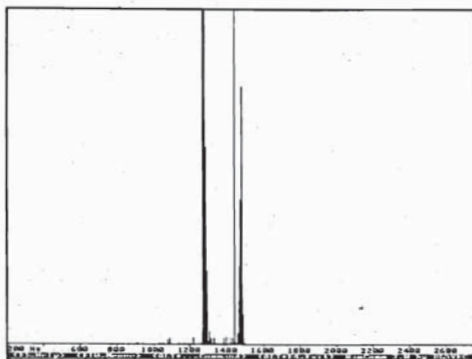


Figura 2: Espectro de los datos

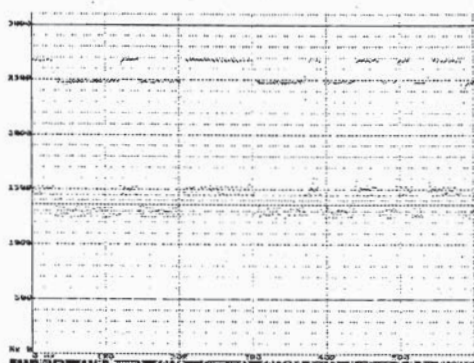


Figura 3: Señal de audio decodificada

En la figura 3, obtenida con la opción *Scope*, se presenta la señal de audio una vez decodificada por el programa en la parte central. En esta misma pantalla, en la parte superior, se representa la señal anterior una vez sometida a un filtro paso bajo, de esta forma se puede ver claramente la señal digital que se está recibiendo. Esta señal es posteriormente convertida según el protocolo utilizado en un conjunto de caracteres representados en la pantalla del ordenador.

Por último se tiene la opción *Tune* que permite desplazar la frecuencia umbral para que quede centrada entre los tonos que representan los bits. Así mismo la aplicación incluye numerosas opciones para variar la velocidad de transmisión, el valor de desplazamiento entre tonos, de tal forma que se puede acomodar la transmisión a las condiciones de propagación existente, niveles de ruido, desplazamiento de frecuencia portadora, etc.

Existen muchas otras alternativas a la mostrada con el programa HAMCOMM, sólo se ha pretendido dar una introducción a las posibilidades existentes, así como presentar un programa fácil de utilizar para realizar experimentos y montajes de transmisiones digitales, y a su vez permite conocer los fundamentos básicos de la tecnología de las comunicaciones.

3. SEGUNDA PRÁCTICA: GENERADOR/DETECTOR DE CRC

En esta práctica se utiliza el ordenador de dos formas: para manejar una tarjeta que permite la generación y detección de códigos de redundancia cíclica (CRC) y para realizar un programa

que realice la misma función anterior pero mediante un procedimiento exclusivamente software.

De esta forma se aprende a utilizar el ordenador para controlar un circuito y se permite que puedan contrastarse un procedimiento hardware y uno software para realizar la misma función.

3.1. Procedimiento hardware

Los códigos de redundancia cíclica o CRC son una técnica muy poderosa y de fácil implementación que se utiliza para detectar errores. El procedimiento puede ser explicado como sigue: dado un mensaje de k bits, el transmisor genera una secuencia de n bits, denominada secuencia de chequeo, que se añade al mensaje a transmitir formando una trama de $k+n$ bits que debe ser divisible por un número determinado. Cuando la trama llega al receptor, se divide por dicho número y si el resto es 0 supone que no hay error.[1]

Para generar la secuencia de n bits que se añade al mensaje, se utiliza lo que se denomina el polinomio generador. A partir de dicho polinomio obtendremos un número binario que tendrá un 1 en la posición i si el término x^i aparece en el polinomio, en caso contrario tendrá un 0. Existe un circuito digital muy sencillo (figura 4) que puede utilizarse para generar el CRC. Dicho circuito se basa en un registro de desplazamiento y se implementa como sigue:

1. El registro tiene n bits correspondientes al número de bits de la trama de chequeo.
2. Hay un máximo de n puertas exclusive OR
3. La presencia o ausencia de una puerta coincide con la presencia, o no, de un término en el polinomio generador.

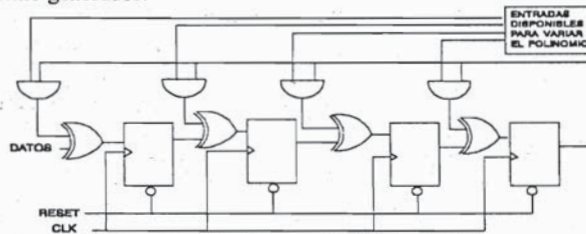


Figura 4: Circuito generador/detector de CRC

Mediante las entradas de las puertas AND, se anula o no el efecto de la puerta XOR en cada una de las etapas del registro. Por lo que se pueden programar distintos polinomios generadores (siempre que sean de grado 4). A la salida de cada uno de los biestables, se ha colocado un diodo LED que indica al alumno cual es el estado de los mismos a lo largo de todo el proceso. El circuito se controla mediante un programa en el lenguaje C que manda los valores adecuados al puerto paralelo del ordenador, cuyas salidas controlan las distintas funciones de la tarjeta generadora/detectora de CRC [6,7].

Una vez suministrados todos los datos, en los biestables queda almacenado el valor del CRC que debe de añadirse a las tramas. Se deben calcular los CRC de distintos tipos de datos y para varios polinomios generadores.

El circuito también se utiliza como detector, para ello se introducen por la línea de datos las tramas anteriores con los CRC calculados, se debe observar que todos los biestables permanecen a cero. Por último, se provocan errores en las tramas, en este caso, en la mayoría de las pruebas, al final del proceso siempre queda algún LED encendido.

3.2. Procedimiento software

El procedimiento software emula el funcionamiento del generador/detector hardware. Por un lado, se define una variable denominada "resultado", cuyo contenido se corresponde con los valores almacenados en los biestables y la entrada DATOS. Por otro lado, definiremos la variable "polinomio", que contiene los bits correspondientes al polinomio generador.

En primer lugar, el algoritmo carga los bits del carácter en la variable resultado y realiza un desplazamiento a la derecha de tres bits. Con esto, tenemos en resultado los cinco primeros bits del carácter. A continuación, entramos en un bucle dónde se procede a la división en módulo 2. Se determina si el bit 4 de resultado es un uno, en cuyo caso se realiza la operación XOR con los bits de polinomio, o es un cero, en cuyo caso no se realiza ninguna operación. Posteriormente, se desplaza hacia la izquierda el contenido de resultado y se introduce el siguiente bit de la trama en la posición menos significativa de esta variable. El bucle finaliza cuando se ha repetido el proceso con todos los bits de la trama (incluidos los bits del CRC). La función `calculo_crc`, coloca el resultado del crc en los cuatro bits siguientes a los datos. Este programa sirve tanto para la generación como para la detección.

Al final del programa queda almacenado en resultado el valor del CRC. Dicha variable será cero, si junto con los bits del carácter se introduce el CRC correcto. En la tabla 1 se muestra el programa que deben de diseñar los alumnos.

3.3 Comparación de ambos métodos

Como en casi todos los procesos, el hecho de poseer un hardware específico para su realización tiene la ventaja de aumentar la velocidad del proceso y la desventaja del coste de dicho hardware.

En nuestro caso, el proceso de cálculo del CRC en el procedimiento hardware está muy unido a la transmisión de los datos al canal de comunicaciones. A medida que los datos se van transmitiendo bit a bit para pasar a un canal de comunicaciones serie, se va haciendo pasar por un registro de desplazamiento que corresponde al que realiza el cálculo del CRC. Una vez transmitidos todos los datos, se transmite el CRC de forma inmediata.

En el procedimiento software, hay que disponer de todos los bits del dato que se desea transmitir para introducirlos en el programa. Tras la ejecución del programa, hay que obtener los datos correspondientes al CRC, añadirlos a la trama a transmitir y proceder a la transmisión. Todos estos pasos hacen que el proceso se relentece.

Igual ocurre en el proceso de la recepción para detectar el CRC. En el hardware, a medida que los bits se van recibiendo, se van haciendo pasar por un registro de desplazamiento. Si cuando ya han pasado todos los bits, el registro se queda a cero, es que no se han producido errores.

<pre> #include <stdio.h> int pol(int); unsigned char calculo_crc(unsigned char *tabla, int tamaño) { unsigned char resultado; unsigned char polinomio=0x15; int numero_bits=tamaño*8; int i=0; resultado=tabla[0]; /*cargamos los bits del caracter en resultado*/ resultado>>=3; /*se desplaza a la derecha 3 bits para tener el mismo número de bits efectivo que polinomio*/ for(i=5; i<numero_bits+4; i++) { if(resultado&0x10) resultado^=polinomio; /*si primer bit del caracter es 1 se realiza la operación XOR con los bits del polinomio*/ resultado<<=1; if(tabla[i/8]&pol((7-(i%8)))) resultado++; /*se añade siguiente bit del caracter*/ } if(resultado&0x10) resultado^=polinomio; </pre>	<pre> tabla[tamaño]=resultado; tabla[tamaño]<<=4; /*con esto el CRC queda almacenado al final de los bits del dato y dispuesto para ser enviado*/ return resultado; } int pol(int dato) { if(dato==0) return 1; else return 2*pol(dato-1); } void main (void) { char tabla[3]; tabla[0]=0x80; tabla[1]=0; tabla[2]=0; printf("%s", calculo_CRC(tabla,2)); } </pre>
---	--

Tabla 1: Programa correspondiente al procedimiento software

4. Bibliografía

- [1] Stallings W.. "Comunicaciones y Redes de Computadores". Ed. Prentice-Hall. Quinta Edición. 1997.
- [2] RadioRaft life v 2.12. <http://ourworld.compuserve.com/homepage/F6FLT/>
- [3] WINTNC. <http://www.g7jjf.demon.co.uk>
- [4] Bavarian Packet Radio Group. <http://www.baycom.de>
- [5] Programa Hamcomm. <http://ftp.uni-heidelberg.de/inline/xtrct/ftp/pub/msdos/simtel/hamradio/hamcom22.zip>
- [6] Campbell J.. "Comunicaciones Serie. Guía de Referencia del programador en C". Ed. Anaya. 1989.
- [7] IBM. "Technical Reference". 1983.