

IV Entornos de funcionamiento

En este capítulo vamos a explicar el *modo de funcionamiento* de nuestra aplicación, en los dos entornos en los que ha sido desarrollado, esto es, en *Matlab* y en la consola *MS-DOS de Windows*. Mostraremos algunas gráficas para que se tenga una idea del *entorno gráfico* en el que nos vamos a mover durante la aplicación.

IV.1.- ENTORNO DE MATLAB

TRANSMISOR

Para llevar a cabo la ejecución del proyecto en *Matlab* es necesario abrir dicha aplicación y situarse en el directorio en el cual se ubiquen los ‘*archivos.m*’ que implementen el *Bloque Transmisor*. A continuación debemos teclear ‘*z_main*’ (sin comillas), para llamar a la función principal y arrancar la aplicación transmisora.

Seguidamente, el programa nos preguntará si queremos introducir los valores de los *parámetros de diseño* de nuestro sistema, y que ya conocemos sobradamente, o bien preferimos los que están por defecto incluidos en el código de la función ‘*z_main*’. En caso de que queramos introducir los datos, éstos aparecerán en el siguiente orden:

- Nombre del Fichero a transmitir (si no se encontrase generaría un error).
- Número de símbolos de la constelación.
- Número de muestras por símbolo.
- Número de bytes del campo DATOS por cada trama.
- Frecuencia de muestreo.
- Número de bits de resolución.
- Factor de Roll-off.

Tras introducir estos datos, las tramas se irán procesando y, a medida que esto ocurre, irá apareciendo en pantalla el número de trama que acaba de ser procesada. Posteriormente aparece un mensaje pidiéndonos que pulsemos “*enter*” para transmitir el fichero. Finalmente aparece un mensaje que indica que el fichero ha sido transmitido.

A modo de *ejemplo* ilustrativo, mostraremos un gráfico del entorno de *Matlab* de nuestro proyecto durante la transmisión de un fichero de texto denominado ‘*prueba.txt*’.

Los parámetros usados, como puede verse en la imagen, son los siguientes:

- Nombre del fichero: ‘*prueba.txt*’.
- $M = 256$ símbolos.
- $N = 10$ muestras por símbolo.
- BDT = 70 bytes de datos por trama.
- $F_s = 44100$ hz.
- NB = 16 bits por muestra.
- $\alpha = 0.5$.

Para mostrar el procesamiento en transmisión hemos tenido que crear una ‘megapantalla’ a partir de 3 impresiones gráficas, con la ayuda de las herramientas de Word.

La pantalla que resulta es la siguiente:

```

MATLAB
File Edit View Web Window Help
Current Directory: F:\Documents and Settings\zeni\Mis documentos\Proyecto\zeni\transmisor

          PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACION DE UN MODEM APK MEDIANTE SOUNDBLASTER

AUTOR: José Miguel Moreno Pérez
DIRECTOR: José Ramón Cerquides Bueno

*****
*                TRANSMISOR                *
*****

desea meter usted los datos?[Y/N]: y

Introduzca el fichero que desea transmitir: prueba.txt
Número de niveles de la modulación M-QAM (M=4/8/16/32/64/128/256): 256
Número de muestras por símbolo: 10
Longitud (bytes) del campo de datos de las tramas Ethernet:70
Frecuencia de muestreo (Hz): 44100
Resolución (8 ó 16 bits): 16
Factor de Roll off (entre 0 y 1): 0.5

PROCESANDO FICHERO. POR FAVOR, ESPERE...
Trama: 1
Trama: 2
Trama: 3
Trama: 4
Trama: 5
Trama: 6
Trama: 7
Trama: 8
Trama: 9
Trama: 10
Trama: 11
Trama: 12
Trama: 13
Trama: 14
Trama: 15

PULSE LA TECLA "ENTER" PARA TRANSMITIR EL FICHERO...

*****FICHERO TRANSMITIDO*****

Ready

```

Fig. 113.- Entorno Matlab. Transmisor
 Diseño e implementación de un módem
 José Miguel Moreno Pérez
 APK mediante SoundBlaster

RECEPTOR

Al igual que en el Transmisor, para ejecutar el **Bloque Receptor** del proyecto, deberemos abrir otra sesión de *Matlab* y situarnos en el directorio que contenga la implementación de dicho bloque mediante nuestros *'archivos.m'* de Matlab. A continuación hay que volver a teclear *'z_main'* para llamar al programa principal del Bloque Receptor, y ejecutar la cadena receptora. Acto seguido, el programa nos volverá a preguntar si queremos introducir o no los datos, o bien dejar los que están contenidos por defecto en la función *'z_main'*. Hay que volver a insistir en que, excepto el nombre de fichero destino, el resto de **parámetros de diseño** deben ser **iguales a** los del **Bloque Transmisor**, para una correcta recepción de los datos. Éstos, en caso de que se decida introducirlos manualmente, aparecerán en el siguiente orden:

- Nombre del Fichero destino (si no existiera se crearía).
- Número de símbolos de la constelación.
- Número de muestras por símbolo.
- Número de bytes del campo DATOS por cada trama.
- Frecuencia de muestreo.
- Número de bits de resolución.
- Número de bytes totales del fichero origen.
- Factor de Roll-off.

A continuación, el programa nos pedirá que pulsemos la tecla *"enter"* para empezar a grabar. Ya sabemos que grabaremos 3 segundos más de lo necesario, esto es, $44100 \cdot 3 = 132300$ muestras más de lo necesario (para una $F_s = 441000$ hz.), y tener así el tiempo suficiente para poder pulsar y transmitir los datos en el Bloque Transmisor.

Luego se nos vuelve a pedir que pulsemos la tecla *"enter"* para procesar los datos registrados, tras lo cual empiezan a enumerarse, al igual que en la pantalla del Bloque Transmisor, las tramas que van siendo procesadas. Por último, muestra un mensaje para informar de la creación del nuevo fichero, y muestra los resultados, es decir, el número de tramas procesadas, y el número de tramas erróneas.

Para ilustrar todo lo comentado en este bloque mediante un **ejemplo**, volveremos a usar el fichero *'prueba.txt'* que transmitimos anteriormente en el Bloque Transmisor. Es importante destacar, como ya lo hicimos durante el capítulo III.2, que es indiferente el tipo de fichero que quiera transmitirse, pero ya que expusimos el ejemplo de un fichero de texto en transmisión, vamos a mostrar la parte de recepción de dicho fichero, utilizando, obviamente, los mismos parámetros, es decir:

- Nombre del fichero destino: *'pruebarx.txt'*.
- $M = 256$ símbolos.
- $N = 10$ muestras por símbolo.
- $BDT = 70$ bytes de datos por trama.
- $F_s = 44100$ hz.
- $NB = 16$ bits por muestra.
- Tamaño del fichero origen: 1000 bytes.
- $\alpha = 0.5$.

La pantalla que resulta es la siguiente:

.....

```

MATLAB
File Edit View Web Window Help
Current Directory: F:\matlabR12\work\receptor

PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACION DE UN MODEM APK MEDIANTE SOUNDBLASTER

AUTOR: Jose Miguel Moreno Perez
DIRECTOR: Jose Ramon Cerquides Bueno

*****
*                               Receptor                               *
*****

desea meter usted los datos?[Y/N]: y
Introduzca el nombre del fichero que desea crear: pruebarx.txt
Numero de niveles de la modulacion APK (M=4/8/16/32/64/128/256): 256
Numero de muestras por simbolo: 10
Longitud (bytes) del campo de datos de las tramas Ethernet: 70
Frecuencia de muestreo (Hz): 44100
Resolucion (8 o 16 bits): 16
Longitud (bytes) del fichero: 1000
Factor de Roll of (entre 0 y 1): 0.5

PULSE LA TECLA "ENTER" PARA COMENZAR A GRABAR...

PULSE LA TECLA "ENTER" PARA PROCESAR LOS DATOS REGISTRADOS...

PROCESANDO MUESTRAS. POR FAVOR, ESPERE...

Trama: 1
Trama: 2
Trama: 3
Trama: 4
Trama: 5
Trama: 6
Trama: 7
Trama: 8
Trama: 9
Trama: 10
Trama: 11
Trama: 12
Trama: 13
Trama: 14
Trama: 15

Numero de Tramas transmitidas: 15

Numero de Tramas erroneas: 0

creando el nuevo fichero...

*****FICHERO TRANSMITIDO*****
>>
Ready

```

Fig. 114.-Entorno Matlab. Receptor

IV.2.- ENTORNO DE MS-DOS

TRANSMISOR

Para ejecutar nuestra aplicación en la *consola de MS-DOS de Windows*, para el *Bloque Transmisor*, tendremos que ubicarnos en dicha consola, dentro del directorio donde se encuentre el fichero ejecutable generado por el compilador, y escribir su nombre, es decir, '*z_main.exe*', y el Transmisor comenzará a ejecutarse siguiendo los pasos idénticos a su ejecución en Matlab.

Realmente *no es necesario abrir la consola de MS-DOS*, sino que directamente hacemos *dobles clic* sobre el icono del fichero '*z_main.exe*', y la consola se abrirá automáticamente para que el fichero pueda ser ejecutado.

Para mostrar un *ejemplo* ilustrativo del entorno de la *consola de MS-DOS*, vamos a transmitir un fichero gráfico escogido al azar, y que se denomina '*Pica.gif*', cuya imagen es la siguiente:



Fig.115.- Pica.gif

Vamos a tratar de transmitir la imagen de este lindo ratoncito a través de la tarjeta de sonido. El resultado es el siguiente (los parámetros usados se reflejan en la gráfica):

```

F:\WINNT\System32\cmd.exe
F:\Documents and Settings\zemi\Mis documentos\Proyecto\zemi\transmisor\lge C>z_main

          PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACION DE UN MODEM APK MEDIANTE SOUNDBLASTER

AUTOR: Jose Miguel Moreno Perez
DIRECTOR: Jose Ramon Cerquides Bueno

*****
*                               *
*                   TRANSMISOR   *
*                               *
*****

desea meter usted los datos?[Y/N]: y

Introduzca el fichero que desea transmitir: Pica.gif
Numero de niveles de la modulacion M-QAM (M=4/8/16/32/64/128/256): 8
Numero de muestras por simbolo: 10
Longitud (bytes) del campo de datos de las tramas Ethernet:70
Frecuencia de muestreo (Hz): 44100
Resolucion (8 o 16 bits): 16
Factor de Roll of (entre 0 y 1): 0.5

PROCESANDO FICHERO. POR FAVOR, ESPERE...
Trama: 1
Trama: 2
Trama: 3
Trama: 4
Trama: 5
Trama: 6
Trama: 7
Trama: 8
Trama: 9
Trama: 10
Trama: 11
Trama: 12
Trama: 13
Trama: 14
Trama: 15

PULSE LA TECLA "ENTER" PARA TRANSMITIR EL FICHERO...

*****FICHERO TRANSMITIDO*****

F:\Documents and Settings\zemi\Mis documentos\Proyecto\zemi\transmisor\lge C>_

```

Fig. 116.-Entorno MS-Dos. Transmisor

RECEPTOR

Para llevar a cabo la ejecución en el *Bloque Receptor*, volvemos a repetir los mismos pasos que en el Bloque Transmisor, de manera que, o bien hacemos *doble clic* en el fichero '*z_main.exe*' correspondiente al archivo ejecutable del bloque receptor, o bien *abrimos la consola de MS-DOS*, nos situamos en el directorio de dicho fichero, y tecleamos su nombre.

El resultado será, en ambos casos, la captura y el procesamiento de las muestras. Lógicamente, los pasos que se siguen en este extremo de la comunicación son los mismos que se dieron durante la ejecución del Bloque Receptor en el entorno *Matlab*.

Como *ejemplo* gráfico, vamos a recibir el archivo gráfico que transmitimos en el Bloque Transmisor, y vamos a comprobar si efectivamente "el ratón llega sano y salvo a recepción":

```

F:\WINNT\System32\cmd.exe
F:\Documents and Settings\zemi\Mis documentos\Proyecto\zemi\receptor\lge C>z_main

          PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACION DE UN MODEM APK MEDIANTE SOUNDBLASTER

AUTOR: Jose Miguel Moreno Perez
DIRECTOR: Jose Ramon Cerquides Bueno

*****
*              Receptor              *
*****

desea meter usted los datos?[Y/N]: y
Introduzca el nombre del fichero que desea crear: picarx.gif
Numero de niveles de la modulacion APK (M=4/8/16/32/64/128/256): 8
Numero de muestras por simbolo: 10
Longitud (bytes) del campo de datos de las tramas Ethernet: 70
Frecuencia de muestreo (Hz): 44100
Resolucion (8 o 16 bits): 16
Longitud (bytes) del fichero: 1017
Factor de Roll of (entre 0 y 1): 0.5

PULSE LA TECLA "ENTER" PARA COMENZAR A GRABAR...

PULSE LA TECLA "ENTER" PARA PROCESAR LOS DATOS REGISTRADOS...

PROCESANDO MUESTRAS. POR FAVOR, ESPERE...

Trama: 1
Trama: 2
Trama: 3
Trama: 4
Trama: 5
Trama: 6
Trama: 7
Trama: 8
Trama: 9
Trama: 10
Trama: 11
Trama: 12
Trama: 13
Trama: 14
Trama: 15

Numero de Tramas transmitidas: 15

Numero de Tramas erroneas: 0

creando el nuevo fichero...

*****FICHERO TRANSMITIDO*****
F:\Documents and Settings\zemi\Mis documentos\Proyecto\zemi\receptor\lge C>_

```

Fig. 117.-Entorno MS-Dos. Receptor

A juzgar por los resultados (ninguna trama errónea), parece ser que el fichero *'Pica.gif'* es idéntico al fichero generado en destino, *'picarx.gif'*. De todos modos, vamos a mostrar un dibujo de dicho fichero *'picarx.gif'*, para verificar los óptimos resultados:



Fig.118.- picarx.gif

Nota: Volvemos a insistir que si queremos transportar el archivo *'z_main.exe'*, además de dicho fichero hacen falta los ficheros *playsnd.dll*, y *rcsnd.dll* (encargados de la transmisión y recepción de las muestras, y que situaremos en el mismo directorio de su ejecutable correspondiente como ya explicamos), y las librerías dinámicas (*'run-time libraries'*), que se encuentran, como buenamente sabemos, en el archivo instalador de dichas librerías denominado *mglinstaller.exe*.