



**TECNOLOGICO DE ESTUDIOS  
SUPERIORES DE ECATEPEC**



**DIVISIÓN DE INGENIERÍA ELECTRÓNICA Y  
TELEMÁTICA**

**PRÁCTICAS DE LABORATORIO**

**ASIGNATURA: INGENIERÍA TELEMÁTICA**

**REALIZÓ:**

**HÉCTOR OSORIO RAMÍREZ**

**SEPTIEMBRE 2009.**

# PRESENTACIÓN

El presente manual de prácticas fue realizado, para la asignatura de Ingeniería Telemática, el cual, intenta proporcionar a los docentes y estudiantes un material de apoyo que facilite el proceso enseñanza-aprendizaje, a través del trabajo en el laboratorio, reforzando de esta manera, la teoría mostrada en el salón de clases.

Las prácticas de este manual, son presentadas para que el estudiante logre un aprendizaje significativo, debido a que están diseñadas de forma que el docente actúe como guía y el docente participe activamente, haciendo experimentos y al mismo tiempo aprendiendo por descubrimiento.

Dicho lo anterior, se justifica el brindar a los alumnos un manual que los encamine a la aplicación de los conceptos teóricos, permitiendo profundizar más en los casos prácticos.

## ÍNDICE

PRÁCTICA 1. Codificación de datos binarios	4
PRÁCTICA 2. Modulación ASK, FSK, PSK	10
PRÁCTICA 3. Transmisión Asíncrona	16
PRÁCTICA 4. Transmisión Síncrona	27
PRÁCTICA 5. Detección y Corrección de Errores (Frame Relay)	31
PRÁCTICA 6. Control de Flujo de Transmisión	38
PRÁCTICA 7. Multiplexaje por División de Frecuencia	41
PRÁCTICA 8. Multiplexaje por División de Tiempo	41
PRÁCTICA 9. Conmutación Digital y Enrutamiento	47
PRÁCTICA 10. Creación de una Red LAN	55

# Práctica 1

## Codificación de datos binarios

### 1. Objetivo

En esta práctica se pretende que el alumno comprenda la dificultad que plantea la codificación binaria de la información. Hoy en día se tiende a almacenar todo tipo de información en formato digital: imágenes, sonido, textos, etc. Todas estas operaciones se realizan sobre ficheros que contienen la información binaria, es necesario un proceso de conversión de la información que queremos almacenar a formato digital.

Se va a realizar manualmente el proceso de conversión de información a formato digital para el caso de imágenes, ya que éste es uno de los que presenta menor dificultad.

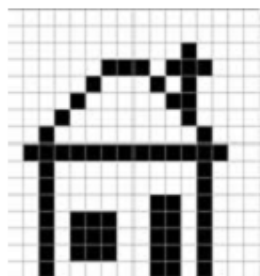
### 2. Desarrollo

El desarrollo de esta práctica consistirá en visualizar y modificar la información digital existente en archivos en formato de imágenes digitales. Para ello se dispone de diversos ficheros con imágenes en diferentes formatos y se utilizarán programas de edición avanzada de ficheros.

#### 2.1. Ejemplo de imagen en blanco y negro

La codificación de la información en formato digital se realiza mediante la obtención de una cadena de unos y ceros, que mediante algún tipo de transformación denominada digitalización.

En la Figura 1 se muestra la información digital necesaria para digitalizar una imagen en blanco y negro. Cada dígito binario (0 ó 1) indica si el correspondiente punto debe estar en negro en blanco. En una imagen digital, el punto, es la unidad mínima de información de que se dispone. Además del punto, para conseguir visualizar bien la figura es necesario saber cual es el número de filas y de columnas de puntos que tendrá la imagen, con el fin de colocar cada punto en su lugar correcto.



```

0000000000000000
0000000000000000
0000000000010000
0000001110111000
0000010001010000
0000100000110000
0001000000010000
0010000000001000
0010000000001000
0111111111101100
0010000000001000
0010000000001000
0010000000001000
0010000001101000
0010111001101000
0010111001101000
0010111001101000
0010000001101000

```

**Figura 1. Codificación binaria de una imagen en blanco y negro**

La imagen mostrada es de 16x16, es decir, tiene 16 columnas de puntos por 16 filas de puntos, y además, cada punto sólo puede tener 2 valores, cero o uno, indicando cada de los valores un color.

## 2.2. Sistema de codificación binario/hexadecimal

Para simplificar la escritura de información digital se utiliza el sistema de codificación hexadecimal. Éste consiste básicamente en representar 4 dígitos binarios por una única letra.

Binario	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

**Tabla 1. Sistema de numeración Hexadecimal**

0000 0000 0000 0000	→	0000
0000 0000 0000 0000	→	0000
0000 0000 0001 0000	→	0010
0000 0011 1011 1000	→	03B8
0000 0100 0101 0000	→	0450
0000 1000 0011 0000	→	0830
0001 0000 0001 0000	→	1010
0010 0000 0000 1000	→	2008
0111 1111 1110 1100	→	7FEC
0010 0000 0000 1000	→	2008
0010 0000 0000 1000	→	2008
0010 0000 0110 1000	→	2068
0010 1110 0110 1000	→	2E68
0010 1110 0110 1000	→	2E68
0010 1110 0110 1000	→	2E68
0010 0000 0110 1000	→	2068

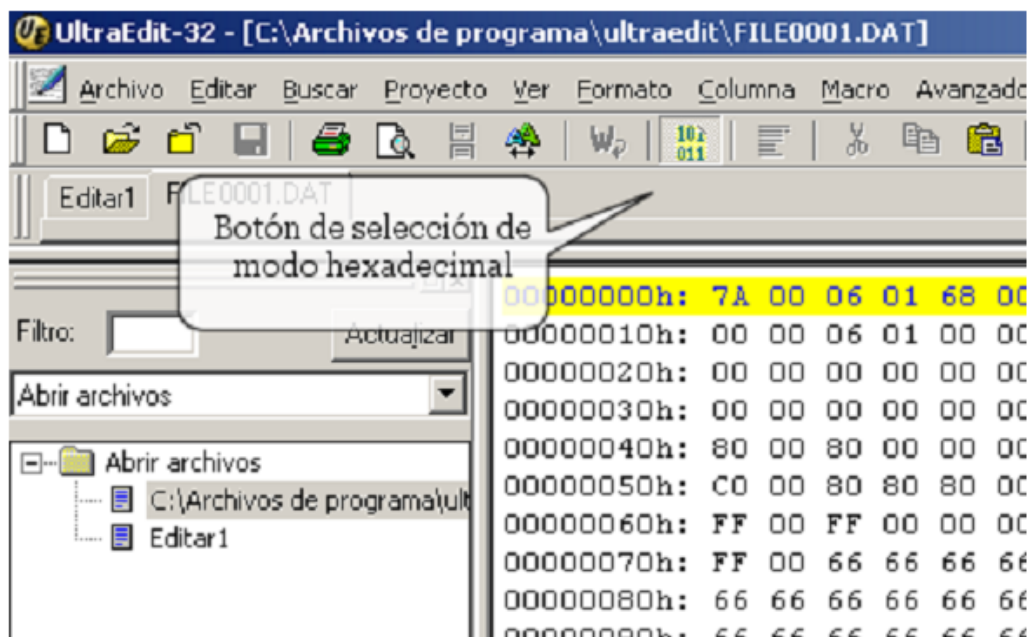
**Tabla 2. Transformación binaria-hexadecimal.**

En la Tabla 1 se muestra una correspondencia de cada grupo de 4 dígitos binarios con símbolo. Si hacemos la conversión de los números binarios a este nuevo código, obtenemos la misma información representada de forma más compacta. Para aplicar la tabla los datos de la Figura 1, hay que agrupar los datos binarios de 4 en 4 y sustituir por valores hexadecimales. Esto se muestra en la Tabla 2.

Este es el procedimiento que seguiremos para conseguir codificar la imagen a formato digital.

## 2.3. Edición de archivos binarios

Para realizar la edición de archivos en modo binario se puede utilizar cualquier editor que admita edición hexadecimal. Para el desarrollo de esta práctica se ha escogido un editor llamado *ultraedit*.



**Figura 2. Editor Binario / Hexadecimal**

En primer lugar iniciaremos el programa y abriremos un archivo. Los archivos que contienen imágenes son aquellos que tienen la extensión “.bmp”. Éstos corresponden a imágenes. Se han preparado diversos ficheros de imagen para trabajar a lo largo de la práctica. Estos ficheros son:

Nombre del fichero	Contenido
16x16x1bit.bmp	Imagen de 16 columnas, 16 filas 2 colores
16x16x4bit.bmp	Imagen de 16 columnas, 16 filas 16 colores
16x16x8bit.bmp	Imagen de 16 columnas, 16 filas 256 colores
16x16x24bit.bmp	Imagen de 16 columnas, 16 filas 16 millones de colores

**Tabla 3. Ficheros usados en la práctica**

Una vez abierto el fichero se mostrará la información en dos diferentes columnas. La de la derecha corresponde a dígitos hexadecimales (agrupaciones de 4 bits) y la de la izquierda el carácter ASCII correspondiente. Es necesario recordar que el código ASCII es un código formado por 8 dígitos binarios a los que hay asociado una letra o símbolo del alfabeto.

```

00000030h: 00 00 10 00 00 00 00 00 00 00 00 00 80 00 00 80 ; .....e..e
00000040h: 00 00 00 80 80 00 80 00 00 00 80 00 80 00 80 80 ; ...ee.e...e.e.ee
00000050h: 00 00 C0 C0 C0 00 80 80 80 00 00 00 FF 00 00 FF ; ..AAA.eee...y..y
00000060h: 00 00 00 FF FF 00 FF 00 00 00 FF 00 FF 00 FF FF ; ...yy.y...y.y.yy
00000070h: 00 00 FF FF FF 01 23 45 67 89 AB CD EF 00 FF FF ; ..yyy.#Eg%<Íi.yy
00000080h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ; yyyyyyyyyyyyyyyy
00000090h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ; yyyyyyyyyyyyyyyy

```

**Figura 3. Edición binaria/hexadecimal del fichero 16x16x4bits.bmp.**

Para comenzar a utilizar el editor se van a realizar unas modificaciones en el fichero "*16x16x4bits.bmp*". Tras abrirlo, debe aparecer la información tal y como se observa en la Figura 3. La línea amarilla indica la línea en la que vamos a realizar modificaciones y el cuadro azul (cursor) indica el lugar exacto donde se realizarán las modificaciones. Es posible mover el cursor entre las diferentes columnas. Si modificamos algún dato en la posición del cursor, se cambiarán simultáneamente la información de las dos columnas, ya que ambas muestran la misma información, la de la derecha en hexadecimal y la de la izquierda en ASCII. La representación hexadecimal son agrupaciones de 4 dígitos binarios, mientras que la representación ASCII son agrupaciones de 8 dígitos binarios.

Es posible averiguar cual el código binario de un carácter (letra o símbolo) sólo con escribirlo en la columna de la izquierda y ver su correspondencia en la columna de la derecha. Por ejemplo, en la columna de la izquierda de la Figura 3 aparece repetido varias veces el símbolo del Euro, si miramos el correspondiente en la columna de datos hexadecimales se obtiene los dígitos hexadecimales "80". Si a estos datos le aplicamos la Tabla 1, se puede obtener el código binario correspondiente: "1000 0000".

#### 2.4. Modificación de los datos de la imagen

El siguiente paso consiste en modificar la imagen *16x16x4bit.bmp* que tenemos abierta en el editor hexadecimal. Esta imagen tiene 16 columnas, 16 filas y cada uno de los puntos está formado por 4 dígitos binarios. Haremos los siguientes pasos:

1. Situamos el cursor azul en la columna hexadecimal y en la fila 0000070h y en la columna II tal y como se muestra en la Figura 3.
2. Tras esto cambiamos los dígitos existentes "FF FF FF FF" por "01 23 45 67 89 AB CD F0" (se muestra el cambio en la Figura 3)
3. Guardamos el fichero en el disco y lo abrimos con el explorador de Windows y vemos la imagen creada.

Cada uno de los números que hemos introducido es un color diferente que muestra la imagen, es decir, habría que rellenar una tabla con los 16 colores de la imagen, correspondiente cada uno a un dígito hexadecimal.

Numero Hexadecimal	Color
0	Negro
1	Granate
2	Verde oscuro
3	Amarillo oscuro
4	Azul oscuro
5	Fucsia
6	Azul marino
7	Gris claro
8	Gris oscuro
9	Rojo
A	Verde claro
B	...
C	...
D	...
E	...
F	...

**Tabla 4. Tabla de colores de la imagen 16x16x4bits.bmp**

## 2.5. Creación de nuestra propia imagen

Para poner puntos de indeterminado color sólo hay que escribir el número correspondiente al color (mostrado en la Tabla 4) con el que queremos pintar en algún lugar de la columna de edición hexadecimal pero siempre **a partir de la posición 117**. El número de la posición se puede ver en la denominada barra de estado del programa que se encuentra en la parte inferior de la ventana. Se muestra en la Figura 4.



Figura 4. Barra de estado del editor binario.

Cada uno de los dígitos que vamos a escribir corresponde a un único punto. Por ejemplo, si escribimos "FF FF FF FF FF FF" escribiríamos unos 12 puntos en color blanco seguidos, por tanto, corresponde a una línea en blanco.

Como ejercicio se propone intentar conseguir diferentes líneas de diferentes colores. También se debería intentar obtener alguna línea vertical.

## 2.6. Edición de otras imágenes

En la Tabla 3 se indican otros ficheros de imágenes que representan diferente número de colores. Se puede calcular para cada uno de ellos cuantos colores es capaz de representar la imagen y cuantos dígitos hexadecimales corresponden con cada punto de la imagen.

El procedimiento consiste ver el número de bits que tiene la imagen por punto mediante potencia de 2:

Número de colores	Bits	Dígitos hexadecimales
2	1	---
16 ( $2^4$ )	4	1
256 ( $2^8$ )	8	2
16Millones ( $2^{24}$ )	24	6

Tabla 5. Relación entre número de colores y dígitos hexadecimales

Utilizando estos datos se podría intentar modificar las imágenes que tienen mayor número de colores.



### 3. Cuestiones

1. Indique cual es el código binario de 4 letras del alfabeto y de 4 símbolos del teclado utilizando para ello el editor hexadecimal.
2. Terminar de rellenar la Tabla 4 con los colores que faltan, realizando para ello modificaciones en el fichero desde el editor binario.
3. Modifique el fichero 16x16x8bit.bmp e intente pintar alguna línea en algún color. ¿Cuál es el color que ha utilizado? ¿A que valor binario o hexadecimal corresponde?
4. Indique cuales serían los dígitos hexadecimales que forman la letra 'U' en una imagen de 4 columnas, 4 filas y de 16 colores.
5. Cree un archivo con el editor de textos Microsoft Word y escriba algún texto fácilmente reconocible. Guarde el fichero con algún nombre y abra luego el mismo fichero pero con el editor binario.
  - 5.1. ¿Reconoce en algún sitio del fichero el texto que escribió? ¿en que línea?
  - 5.2. Haga una modificación en el texto con el editor binario y guarde los cambios en el fichero. Tras abrir el archivo modificado con Word ¿Se ve reflejado el cambio que realizó?
6. Indique en los dígitos hexadecimales necesarios para pintar la imagen de la Figura 1 con 16 colores y con el contorno de la casa pintada en amarillo. ¿Qué habría que hacer para pintar el cielo de la figura de azul?
7. *(Opcional)* Intente modificar las otras imágenes que hay en el disco: 16x16x8bits.bmp y 16x16x24bits.bmp. ¿Se puede dibujar fácilmente en estos ficheros? ¿Qué dificultades encuentra?

# PRÁCTICA 2

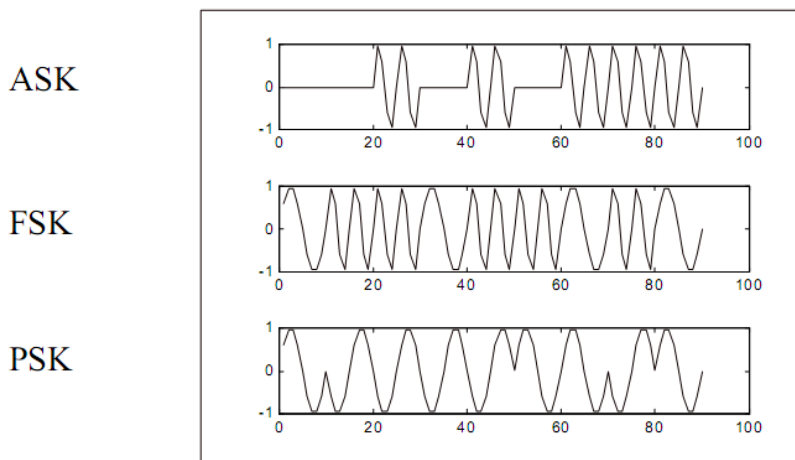
## Modulación ASK, FSK, PSK

### OBJETIVOS

- Visualizar tres tipos de modulaciones digitales: ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying), y PSK (Phase Shift Keying) en el dominio del tiempo y de la frecuencia.
- Ampliar el manejo de *MatLab*, en particular el uso de funciones programadas.

### INTRODUCCIÓN

Los tres esquemas de modulación que se verán en esta práctica son las tres formas más sencillas de modulación digital y son las analizadas en el curso de Sistemas de Comunicación II. Estas son análogas a las modulaciones AM, FM, y PM que fueron analizadas en el curso anterior, Sistemas de Comunicación I, con la diferencia que la señal moduladora no es de tipo analógico sino digital. Así, el desarrollo de esta práctica comenzará con la generación de una señal digital, y su transformación a cada uno de los esquemas propuestos.



En particular en esta práctica se utilizarán funciones diseñadas por los alumnos. Estas funciones son las que dan a *MatLab* su gran potencial para resolver problemas. Algunas

son intrínsecas al programa o integradas al procesador, otras son parte de alguna librería y algunas otras se van generando por cada usuario según sus necesidades, a estas funciones se les llaman *M-files*. En general no son más que una serie de comandos normales de *MatLab* que se graban en un archivo tipo texto con la extensión *\*.m*, de donde viene su nombre. Estos archivos pueden llamar dentro de ellos a otros archivos o inclusive utilizar recursividad, y pueden ser de dos tipos: *Script* o *Function*. Cuando se utiliza un *Script*, se escribe el nombre en la ventana de comando de *MatLab* y el procesador ejecuta todos los comandos que encuentra en el archivo, las variables utilizadas quedan dentro del espacio de trabajo de *MatLab*. Si el archivo comienza con la palabra *Function*, en lugar de ejecutarse como un *Script*, las variables utilizadas son locales y no están disponibles en el espacio de trabajo de *MatLab*. Al ejecutarse desde *MatLab*, se pueden tener valores de entrada y salida como si fuera una subrutina de programación. En los Demos del programa se encuentran algunos ejemplos de aplicación. Utilizar la ayuda para estas dos instrucciones:

*help script*                      *help function*

## DESARROLLO

### 1.- ASK

En primer lugar se simulará un esquema de modulación ASK, para el cual es necesario contar con una serie de datos binarios los cuales se pueden generar con:

```
data = round(rand(1,nbits));
```

donde *nbits* es la longitud de los datos. Esta puede elegirse entre 6 y 12 para poder visualizar los datos en el tiempo y verificar que la modulación haya sido programada de manera correcta, y posteriormente más grande, 64 a 128, para que se pueda asegurar que es una secuencia aleatoria. Posteriormente se toman muestras de esta señal, para esto se debe determinar el número de muestras que se utilizarán, el cual puede ser entre 10 y 20. Es conveniente utilizar una función muestreo que se debe grabar como un archivo de texto *muestreo.m*:

```
function mdata=muestreo (data, nbits, muestras)
% muestro de los datos
mdata=[ ];
for i=1:nbits
    if data(i)==1
        mdata=[mdata ones(1,muestras)];
    else
        mdata=[mdata zeros(1,muestras)];
    end;
end;
end;
```

Para muestrear la señal *data* simplemente se teclea en la ventana principal de *MatLab*:

```
data_m=muestreo(data,128,10);
```

Con base en lo anterior, escribe una función para modular la señal *data\_m* en ASK. Para esto se debe generar una señal senoidal de la misma longitud que *data\_m*, y efectuar la multiplicación término a término.

```
t=1/muestras:1/muestras:length(data);  
seno=sin(2*pi*t);  
askt=data_m.*seno;
```

Una vez hecho esto, graficar los primeros 40 valores tanto los datos digitales como la señal ASK:

```
figure(1);  
subplot(2 1 1);  
plot(data_m(1:40));  
title ('datos digitales');  
subplot(2 1 2);  
plot(askt(1:40));  
title ('Señal ASK');
```

Para analizar el espectro de las señales se puede utilizar una función *potencia.m* que obtenga la señal espectral y su potencia:

```
function potaskw=potencia (askt)  
askw=fft(askt);  
n=length(askw);  
potaskw=askw(1:n/2).*conj(askw(1:n/2));  
end;
```

La función anterior entrega la potencia de la señal ASK en forma espectral, ¿por qué se utiliza solamente la mitad de los datos? La escala de frecuencia dependerá de la tasa de transmisión de los datos digitales en banda base (supóngase 1200 bps) y del número de muestras que se tome. Este debe ser corregido y graficado en el eje *x*:

```
j=1:length(potaskw);  
freq=j*1200/128;
```

Observar la gráfica de la potencia en escala lineal y escala logarítmica.

```

figure(2);
subplot(211);
plot(freq,potaskw);
subplot(212);
semilogy(freq,potaskw);

```

Correr varias veces la modulación para ver como varía el espectro dependiendo de los datos digitales originales. Para evitar las variaciones en la gráfica del espectro, modificar el programa para obtener el promedio de varios bloques de datos en forma aleatoria. Una vez diseñado el sistema de modulación, grabar el programa principal como un *Script*. Comentar todo el proceso.

## 2.- FSK

El proceso a seguir para FSK debe ser análogo al de ASK, con la diferencia de que en este se contará con dos frecuencias. Las funciones creadas anteriormente pueden ser utilizadas nuevamente, lo cual simplifica el trabajo. La modulación ASK tiene la gran desventaja de depender de la amplitud, lo cual lo hace más vulnerable a ruido externo, de la misma manera que AM está sujeta a ruido externo. La modulación en FSK permite que la información esté contenida en la frecuencia de la señal con lo que es más inmune a ruido. ¿Qué desventaja tiene la modulación en frecuencia sobre la modulación en amplitud?

Para tener dos frecuencias según los datos digitales se deben de tener dos funciones diferentes:

```

seno1=sin(4*pi*t);
seno2=sin(2*pi*t);

```

Generar una función diferente para generar FSK, se puede hacer una comparación término a término:

```

for i=1:length(data_m)
    if data_m(i)==1
        fskt=[fskt seno1(i)];
    else
        fskt=[fskt seno2(i)];
    end
end
end

```

o bien generar dos señales ASK de diferente frecuencia, una para los ceros y otra para los unos y sumarlas. Seguir los pasos de la sección de ASK para obtener las mismas tres

gráficas; FSK en el tiempo, en el espectro lineal y en el espectro logarítmico. ¿Se ajustan los resultados a lo esperado?

### 3.- PSK

La modulación en frecuencia no es el esquema más efectivo para modular datos digitales, la modulación en fase PSK, ofrece una buena alternativa. En esta cada bit 0 ó 1 va a alterar la fase 180 grados. Repetir los pasos de las modulaciones anteriores para generar una función que module en PSK una serie de datos digitales, y que muestre la señal en los dominios del tiempo y de la frecuencia. La señal puede cambiar de fase con un simple cambio de signo.

```
if data_m(i)==1
    pskt=[pskt seno(i)];
else
    pskt=[pskt -seno(i)];
end
```

Registrar las tres gráficas observadas y comentar.

### 4.- Comparación de modulaciones.

Las modulaciones propuestas difieren tanto en el tiempo como en el espectro. Graficar en una sola imagen los espectros logarítmicos de las tres modulaciones. La instrucción subplot permite esto:

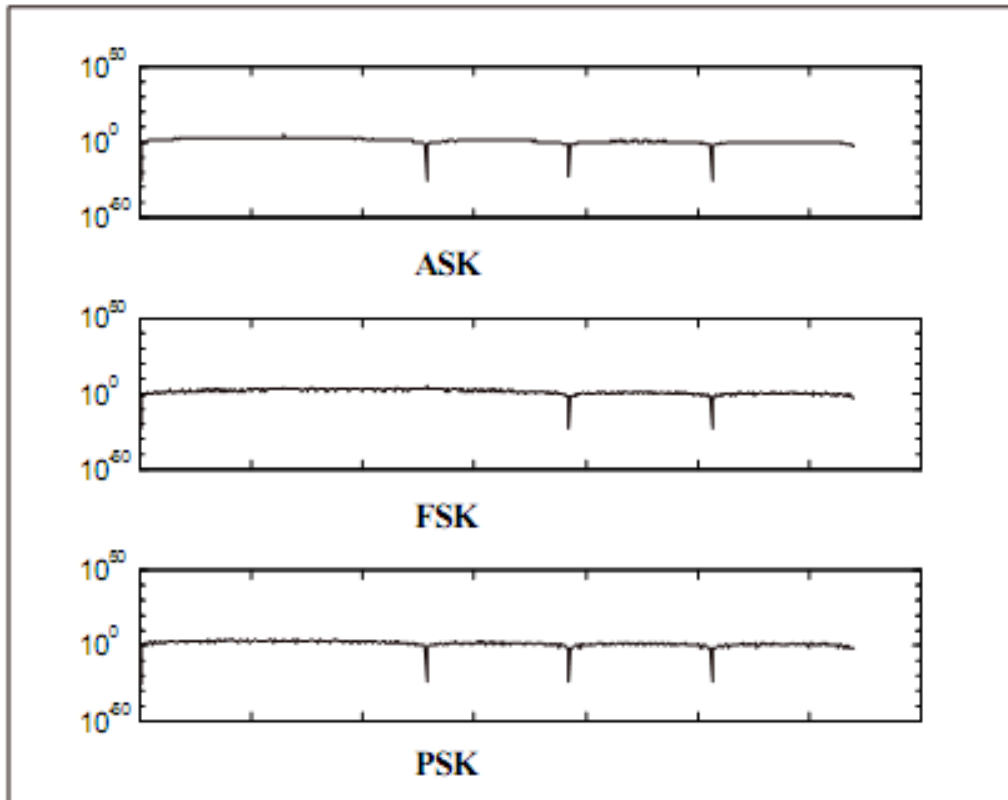
```
figure(1)
subplot(311)
semilogy(freq,potaskw);
```

Esto permite comparar el ancho de banda de las tres señales. Nótese que existen varios lóbulos para las señales de los cuales solamente el lóbulo principal es necesario para transmitir la información por lo que se puede filtrar los otros lóbulos. ¿Que modulación utiliza mayor ancho de banda? ¿Cual el menor? Con base en lo anterior ¿qué esquema de modulación propondría y por que?

### 5.- Otros esquemas de modulación. (Investigación)

¿Existen esquemas que utilicen de manera más eficiente el ancho de banda? Investigar otras modulaciones (TFM, MSK, QAM, ...) que utilicen un ancho de banda menor.

Preparar un reporte en el que se respondan todas las preguntas formuladas, las gráficas observadas para cada modulación ( reducir para no ocupar demasiadas hojas) y comentarios y conclusiones.



## BIBLIOGRAFÍA

Sistemas de Comunicación  
Stremier F. G.  
Addison Wesley  
Tercera Edición



# PRÁCTICA 3 TRANSMISIÓN ASÍNCRONA

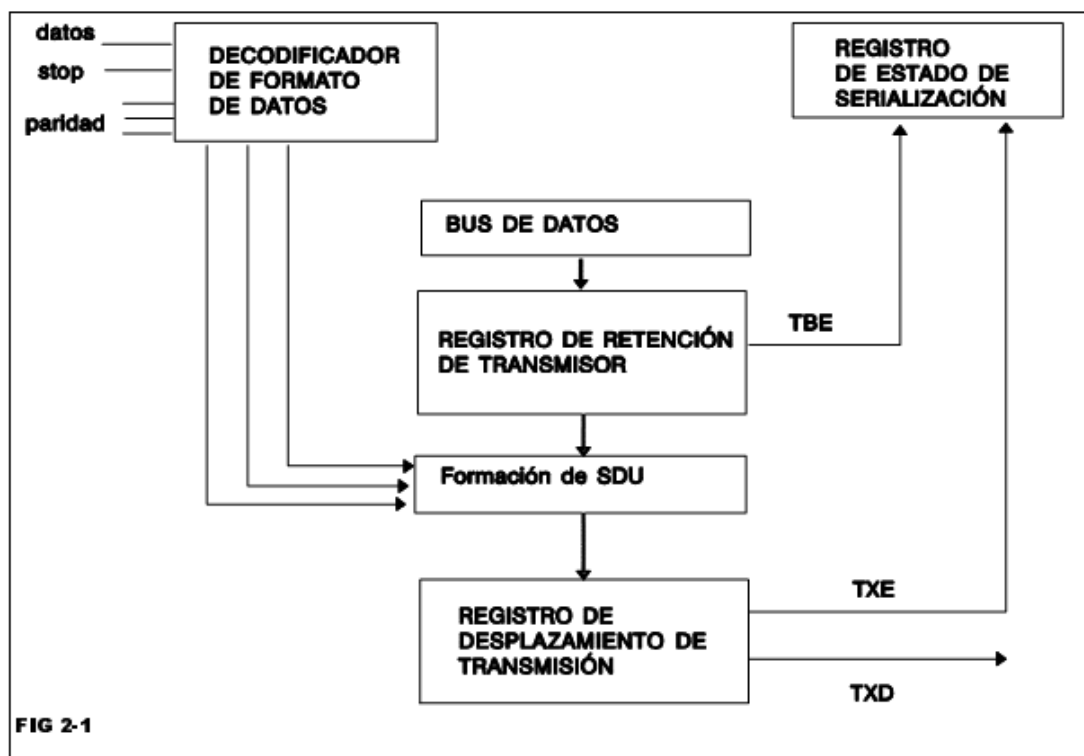
## 1.-Objetivos

- Realizar varios programas para transmitir datos con diferentes formatos asíncronos y diferentes velocidades, comprobando su correcto funcionamiento en el laboratorio.
- Representar y visualizar la señal que se transmite en el dominio temporal, mediante el osciloscopio, comprobando su velocidad binaria y tiempo de bit.
- Comprobar los niveles de señal que facilita el interfaz eléctrico RS-232 C y razonar que ventajas ofrecen el variar los niveles de tensión en una comunicación de DATOS.

## 2.-Estudio Teórico

### • Descripción de la transmisión serie asíncrona

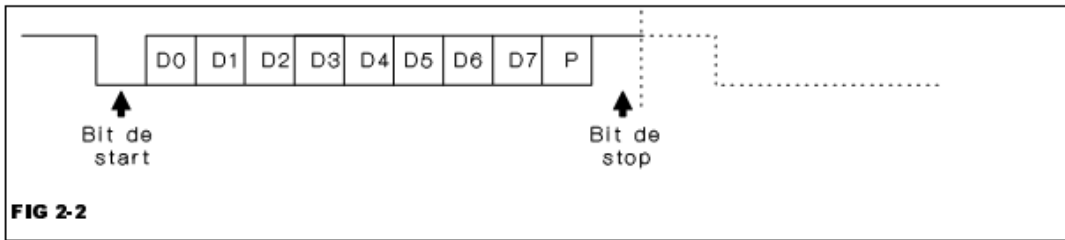
Un transmisor serie asíncrono debe ser capaz de recibir los datos del procesador a través de un bus y enviarlos de forma serie al exterior. Para ello el componente esencial de un transmisor serie es un registro de desplazamiento. Este registro no es accesible directamente. El procesador carga el byte a transmitir en un registro llamado Registro de Mantenimiento de la Transmisión (THRE), que está conectado a las entradas del registro de desplazamiento (TSR). La carga del dato en el registro de



desplazamiento se realiza de forma automática cuando éste se queda vacío (figura 2-1).

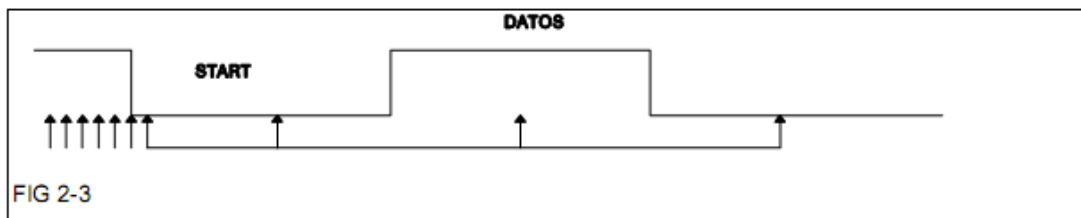


El protocolo asíncrono está orientado al carácter, cada byte que se transmite se rodea de un grupo de bits necesarios para la comunicación y que describiremos a continuación. Cuando no hay datos que transmitir la línea o salida del transmisor se encuentra en el estado lógico 1 "MARK". Si existe transmisión de algún carácter la línea podrá tener la forma que se representa en la figura 2-2.



Previo a la transmisión del byte, la línea se pone a 0 lógico durante el tiempo de un bit, esto es el bit de start. Esto se utiliza para avisar y sincronizar al receptor de la existencia de un dato. A continuación van los bits del carácter comenzando por el menos significativo, el bit de paridad (opcional) cuyo objetivo es el control de errores y, en último lugar, el bit de stop que marca el final de un carácter.

En la UART se emplean dos señales de reloj separadas: una para controlar las operaciones internas y otra para controlar las operaciones de desplazamiento en la secciones de transmisión y recepción. La última, dependerá de la velocidad a la que se quiera realizar la transmisión. La señal de reloj interna está diseñada para que su velocidad sea varias veces superior a la velocidad de transmisión (de 16 a 64 veces superior). El detector del bit de START se encuentra muestreando a la frecuencia interna del reloj, cuando detecta un cambio a 0, en el caso de que la velocidad de transmisión sea 16 veces inferior, volverá a comprobar la entrada 8 ticks de reloj más tarde, si la línea se encuentra a 1, esto implica que se trata de ruido, en caso contrario es el bit de start. En tal situación, y a partir de este momento, se muestrea



cada 16 ticks de reloj, lo cual hace que se tomen los valores de entrada en el punto medio de cada bit (figura 2-3).

La estructura general del receptor UART se muestra en la figura 2-4. Cuando todos los bits que constituyen la trama han sido cargados en el registro de desplazamiento, se construye el byte de datos válido y se pasa al buffer de recepción, activándose el flag RxRDY, que indica que el dato ya está listo. En el proceso de recepción pueden producirse errores (lo veremos más adelante en este documento), de los que se informa al registro de serialización.

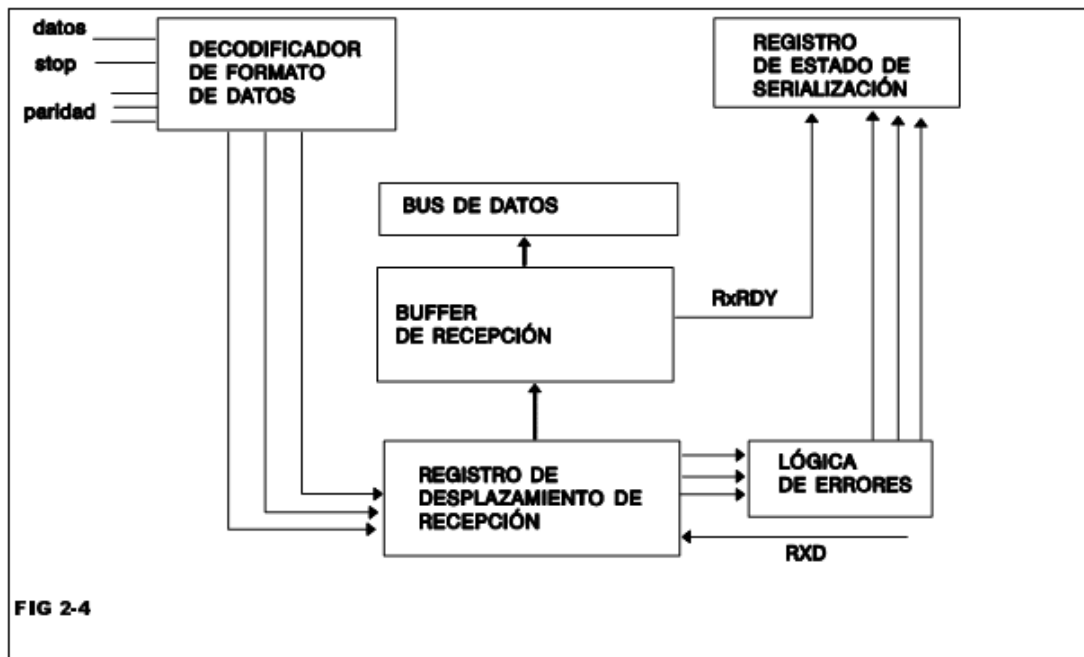


FIG 2-4

- UART (Descripción funcional interna)

El nombre de UART corresponde a las siglas de Universal Asynchronous Receiver Transceiver que quiere decir Transmisor Receptor Asíncrono Universal. En los PC XT y AT este dispositivo lo constituye el circuito integrado 8250 (o el actual 16C450) que estudiaremos a continuación. Para el resto de los modelos, '386 y '486, se suelen utilizar otros integrados, pero manteniendo la compatibilidad con el anterior, lo cual justifica que nos centremos en el estudio del 8250. Este circuito contiene un conjunto de registros que gobiernan su funcionamiento y que tienen asignados una dirección específica en el espacio de entrada/salida del procesador. Esto implica que el control sobre la UART y la transmisión de datos, desde el punto de vista del programador, solo será escribir o leer en posiciones de I/O determinadas. Está claro que deberemos estudiar las funciones de cada uno de estos registros y su localización en el mapa de I/O; esta es la tarea que se desarrolla a continuación.

El PC, por defecto, posee dos puertos para la comunicación serie: el COM1, y el COM2. En la siguiente tabla se muestran los registros internos de la UART y su dirección correspondiente al COM1 y al COM2

COM1	COM2	ACCESO	REGISTROS	DLAB
0X3F8	0X2F8	W	TX BUFFER	0
0X3F8	0X2F8	R	RX BUFFER	0
0X3F8	0X2F8	R/W	DIVISOR LSB	1
0X3F9	0X2F9	R/W	DIVISOR MSB	1
0X3F9	0X2F9	R/W	IER	0
0X3FA	0X2FA	R/W	IIR	X
0X3FB	0X2FB	R/W	LCR	X
0X3FC	0X2FC	R/W	MCR	X
0X3FD	0X2FD	R/W	LSR	X
0X3FE	0X2FE	R/W	MSR	x

• **Registro de Control de Línea(LCR)**

Este registro gobierna el formato de la comunicación asincrónica.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DLAB	BRK	SP	EPS	PEN	STB	WLS1	WLS0

Bit 0 y 1.- Estos dos bits especifican el número de bits en cada carácter recibido o transmitido. El número de bits dependerá de WLS1 y WLS0 según muestra la siguiente tabla

WLS1	WLS0	
0	0	5 BITS
0	1	6 BITS
1	0	7 BITS
1	1	8 BITS

Bit 2.- Este bit especifica el número de bits de stop que serán transmitidos o comprobados en la recepción. Si STB=0, se envía o se comprueba en recepción un bit de stop. Si STB=1 y el tamaño del carácter es de 5bits, se envían y comprueba 1'5 bits de stop, en caso contrario 2 bits.  
Este es el bit de habilitación de paridad. Si PEN= 1 lógico se generará el bit de paridad para los datos a transmitir y se chequeará para los datos recibidos.

Bit 4.- Este es el bit de selección de tipo de paridad. Si PEN = 1 lógico y este bit está a 0 lógico, se genera y chequea el bit de paridad impar; si EPS=1 se genera y chequea el bit de paridad par.

Bit 5.- Este bit permite transmitir el bit de paridad siempre a 1 o a 0. Si SP=1 y PEN=1, se transmite o chequea un 0 si EPS=1 o un 1 si EPS=0.

SP	EPS	PEN	OPERACIÓN
X	X	0	Sin operación
1	1	1	Siempre 0
1	0	1	Siempre 1
0	0	1	Paridad impar
0	1	1	Paridad par

Bit 6.- Este es el bit de control de Break(ruptura). Cuando está a 1 lógico la salida de transmisión se pone a 0 lógico independientemente de cualquier otra actividad. Si el bit de 6 se pone a 0 lógico desaparecerá la condición de Break. La utilidad de esto reside en la posibilidad que tiene el procesador emisor de alertar al sistema remoto.

Bit 7.- Este es el bit de acceso al latch divisor. Debe ser 1 para acceder a los latches divisores del generador de velocidad. Debe ser 0 para acceder al registro de recepción en operación de lectura o al registro de transmisión en operación de escritura.

**• Generador de velocidad programable**

El 8250 posee un generador de velocidad programable capaz de dividir la frecuencia del reloj de entrada (1,8432 Mhz ) mediante un divisor, desde 1 a 216 - 1. La frecuencia de salida del generador es de 16 x velocidad en b.p.s. escogida. Por tanto el contenido del latch divisor será:

$$\text{divisor} = \frac{\text{frecuencia}(1,8432\text{Mhz})}{16 * \text{velocidad}}$$

Existen dos latches de 8 bits para almacenar el contenido del divisor. Para poder acceder al latch más significativo tendremos que poner a 1 el bit DLAB del registro de control de línea y leer o escribir en la dirección 0x2F9, para el latch de menor peso haremos lo mismo pero en la dirección 0x2F8.

Para generar una velocidad de transmisión de 9600 baudios, tendremos que escribir en el latch divisor

DIVISOR LATCH LSB = 0x0C  
 DIVISOR LATCH MSB = 0x00

• **Registro de estado de línea (LSR)**

Este registro informa al procesador sobre el estado de la comunicación.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	TSRE	THRE	BI	FE	PE	OR	DR

**Bit 0.- (Data Ready).** Este bit informa al procesador de la existencia de un dato en el buffer de recepción si está a 1 lógico. Este bit se borra cuando se realiza una operación de lectura en el registro de recepción o cuando se escribe expresamente un 0 en esta posición.

**Bit 1.- (Overrun error).** Este bit informa de la sobre escritura de un dato almacenado en el buffer o registro de recepción. Se borra cuando el procesador lee el registro LSR.

**Bit 2.- (Parity error).** Este bit se pone a 1 lógico cuando el carácter recibido no tiene el bit de paridad correcto en función de lo programado en el LCR. Este bit se pone a 0 cuando se lee este registro.

**Bit 3.- (Framming error).** Este bit se pone a 1 lógico cuando el número de bits de stop del carácter recibido no corresponde con el esperado. Este bit se pone a 0 cuando se lee este registro.

**Bit 4.- (Break indication).** Este bit informa de la recepción de una condición de Break del transmisor. Recordamos que la condición de break se genera cuando la línea se encuentra a 0 lógico durante un tiempo superior al tiempo de transmisión del carácter incluido los bits del protocolo asíncrono; esto es, bit start + carácter + bit paridad + bit de stop.

**Bit 5.-** Este bit es el indicador de buffer de transmisión vacío. Cuando se encuentra a 1 lógico el procesador puede cargar datos en él para su posterior transmisión.

**Bit 6.-** Este es el indicador de registro de desplazamiento de transmisión vacío. Si está a 1 lógico indica que este registro está en reposo. Se pone a 0 lógico en una transferencia desde el registro de transmisión.

**Bit 7.-** Siempre a 0.

• **Registro de Control de Modem (MCR)**

Este registro permite el control con un modem o algún dispositivo periférico que emule a un modem.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	LOOP	OUT2	OUT1	RTS	DTR

**Bit 0.- (Data Terminal Ready).** Este bit controla la salida /DTR. Si está a 0 lógico la salida será un 0 lógico, si es un 1 la salida corresponderá a un 1.

**Bit 1.- (Request to Send).** Este bit controla la salida /RTS del mismo modo que el descrito anteriormente para RTS.

**Bit 2,3.-** Controlan las salidas /OUT 1 y /OUT2, respectivamente, del mismo modo que los bits anteriores.

**Bit 4.- (Loop).** Este bit provee la posibilidad de autodiagnóstico del circuito. Opera de la siguiente manera: la salida de datos serie del circuito se desconecta. Internamente la salida del registro de Tx se conecta con la entrada del registro de recepción. Asimismo las 4 entradas de control del modem (/CTS,/DTS,/RISD, y /RI) son desconectadas; las 4 salidas del MCR son conectadas a las cuatro entradas

de control de modem anteriores.



• **Registro de Estado de Modem (MSR)**

Este registro contiene 4 bits correspondientes al estado de las líneas de control del modem al procesador. Además de estas líneas existen otros 4 bits que muestran información sobre el cambio de las líneas anteriores. Estos bits se ponen a 1 cuando las líneas de entrada cambian de valor y se ponen a 0 lógico cuando el procesador accede a este registro.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CTS	DSR	RI	RLD	DCTS	DDSR	TER	DRLD

Bit 0.- Este bit es el *Delta Clear to Send indicator*. Este bit indica que la entrada CTS ha cambiado de estado desde la última vez que el procesador accedió a este registro.

Bit 1.- Este bit es el *Delta Data Set Ready indicator*. Este bit indica que la entrada DSR ha cambiado de estado desde la última vez que el procesador accedió a este registro.

Bit 2.- Este bit es el *Trailing Edge of Ring Indicator*. Este bit indica que la entrada RI ha cambiado de 1 lógico (ON) a 0 lógico (OFF).

Bit 3.- Este bit es el *Delta Received Line Signal Detector*. Este bit indica que la entrada RLSD ha cambiado de estado.

NOTA: Cuando los bits 0,1,2 o 3 están a 1 lógico se genera una interrupción de estado del modem.

Bit 4.- Este bit es el complemento de la entrada /CTS. Si el bit 4 del MCR está puesto a 1 lógico, este bit es equivalente a RTS en MCR.

Bit 5.- Este bit es el complemento de la entrada /DSR. Si el bit 4 deL MCR está a 1 lógico este bit es equivalente a DTR en MCR.

Bit 6.- Este bit es el complemento de la entrada /RI. Si el bit 4 del MCR está puesto a 1 lógico, este bit es equivalente a OUT1 en MCR.

Bit 7.- Este bit es el complemento de la entrada /RLSD. Si el bit 4 deL MCR está a 1 lógico este bit es equivalente a OUT2 en MCR.

• **Registro de habilitación de interrupciones (IER)**

Este registro habilita los cuatro tipos de interrupción que permite el 8250 y que generan la activación de la línea INTRPT del dispositivo. Es posible inhabilitar todo el sistema de interrupción o seleccionar los tipos de interrupción deseados. El contenido del registro de interrupción es:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	EMSI	ERLS	ETI	EDAI

Bit 0.- Este bit habilita, cuando está a 1 lógico, la interrupción de dato disponible.

Bit 1.- Este bit habilita, cuando está a 1 lógico, la interrupción de registro de transmisión vacío.

Bit 2.- Este bit habilita, cuando está a 1 lógico, la interrupción de estado de la línea de recepción.

Bit 3.- Este bit habilita, cuando está a 1 lógico, la interrupción de estado del modem.

### • Registro de identificación de interrupción (IIR)

El 8250 prioriza interrupciones en cuatro niveles: estado de línea del receptor (prioridad 1), dato recibido disponible (prioridad 2), registro de transmisión vacío (prioridad 3) y estado del modem (prioridad 4). El IIR, cuando es accedido libera la interrupción pendiente de mayor prioridad; ninguna otra es reconocida hasta que la interrupción particular sea servida por el procesador. El contenido del IIR se muestra a continuación.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	ID1	ID0	IP

**Bit 0.**- Este bit se usa para indicar si existe una interrupción pendiente cuando está a 0 lógico. Cuando esto ocurra, el contenido del IIR puede ser utilizado como puntero a la rutina de interrupción apropiada. Cuando este bit se encuentra a 1 lógico no existen interrupciones pendientes.

**Bit 1 y 2.**- Estos dos bits se utilizan para identificar la interrupción pendiente de mayor prioridad. A continuación se muestra la relación de interrupciones en función del contenido de este registro, las causas que las provocan y los modos de limpiarlas.

Si [IR]=1 no existe ninguna interrupción pendiente y por tanto no se genera la activación de la línea INTRP.

Si [IR]=6 Existe la interrupción de mayor prioridad que es la generada por el estado de la línea del receptor. Las fuentes de esta interrupción son: Error de overrun, error de paridad, error de trama o condición break. Leyendo el registro LSR se elimina la fuente de interrupción.

Si [IR]=4 Existe la interrupción de segundo nivel de prioridad que es provocada por la recepción de datos en el registro de recepción. Leyendo el buffer de recepción se elimina esta interrupción.

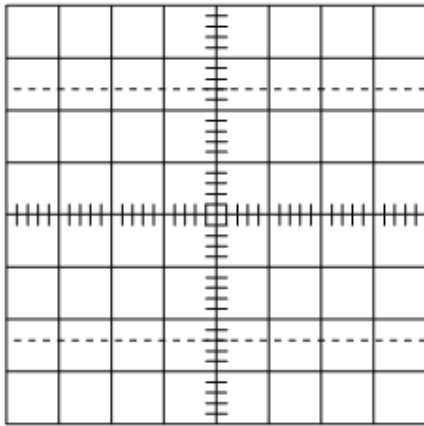
Si [IR]=2 Existe la interrupción de tercer nivel de prioridad que es provocada al quedarse vacío el registro de transmisión. Para eliminar la interrupción hay que mandar un dato al registro de transmisión o bien leer el registro IIR.

Si [IR]=0 Existe la interrupción de menor prioridad que es provocada por el cambio en las líneas de entrada de control del modem. Se limpia la interrupción leyendo el MSR.

## 3.-Estudio previo

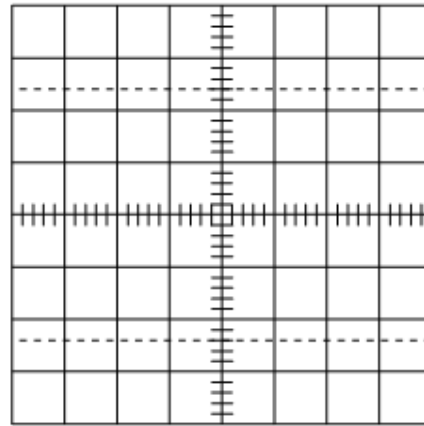
1.- Una determinada UART de un PC está configurada a 600bps, 7 bit, paridad impar, 2 bits de stop y transmitiendo de forma continua el carácter 'A'. Representa en la siguiente pantalla de osciloscopio lo que se vería a la salida del puerto si se conecta la sonda de dicho aparato en él y teniendo en cuenta que un 0 lógico es 12V y un 1 lógico -12V. Determina las unidades para el eje vertical y el horizontal. Realiza la misma operación para velocidades de 1200bps, 2400bps y 4800bps.

Escala eje Y \_\_\_\_\_ V/div



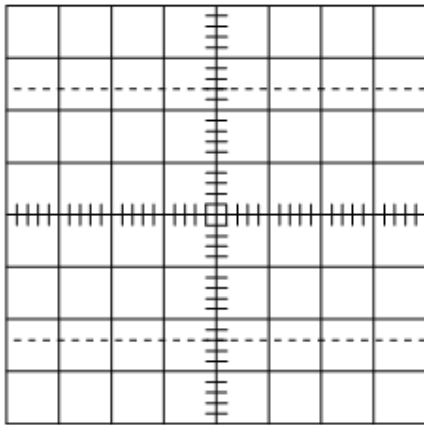
Escala eje X \_\_\_\_\_ s/div

Escala eje Y \_\_\_\_\_ V/div



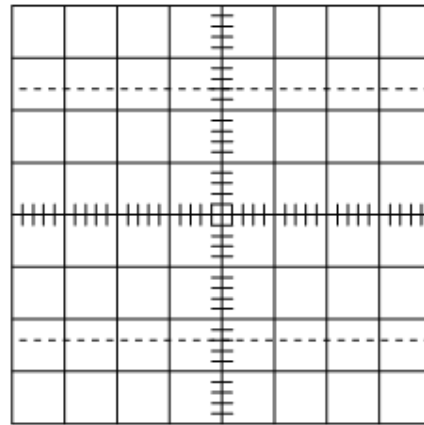
Escala eje X \_\_\_\_\_ s/div

Escala eje Y \_\_\_\_\_ V/div



Escala eje X \_\_\_\_\_ s/div

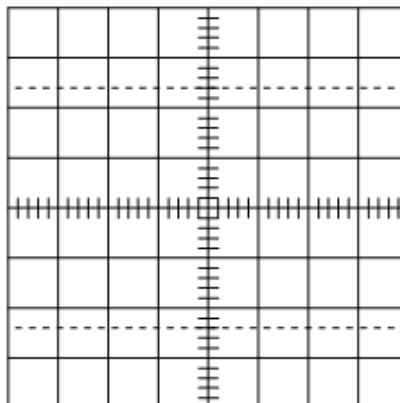
Escala eje Y \_\_\_\_\_ V/div



Escala eje X \_\_\_\_\_ s/div

2.- Repite el apartado anterior para 1200 bps, 7 bits de datos, paridad par y 1 bit de stop y transmitiendo de forma continua el carácter 'R'.

Escala eje Y \_\_\_\_\_ V/div



Escala eje X \_\_\_\_\_ s/div



3.- Realiza una función en lenguaje de programación C, con la configuración del ejercicio 1, para que transmita y visualice el carácter recibido en pantalla, sabiendo que está en bucle los datos transmitidos con los recibidos en el conector de salida.

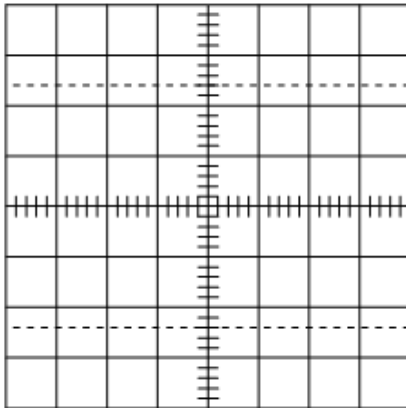
4.- Repite el apartado anterior, con la configuración del ejercicio 2.

#### 4.- Realización práctica

1. Ejecuta el programa propuesto en el punto 3 del estudio previo y comprueba, utilizando un

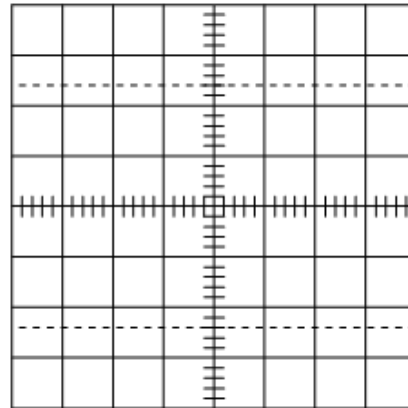
osciloscopio, la salida de la UART para los casos prácticos del apartado 1 del estudio anterior. Además dibuje en las plantillas siguientes, la salida mostrada por el osciloscopio.

Escala eje Y \_\_\_\_\_ V/div



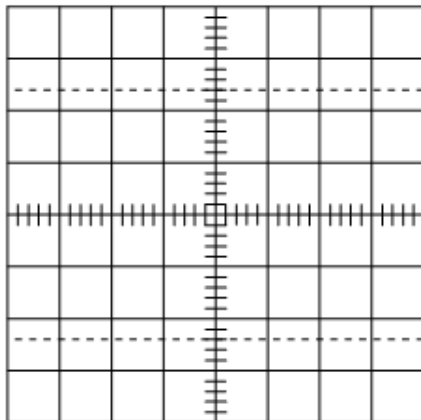
Escala eje X \_\_\_\_\_ s/div

Escala eje Y \_\_\_\_\_ V/div



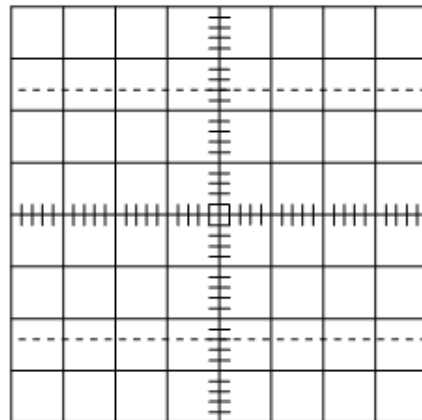
Escala eje X \_\_\_\_\_ s/div

Escala eje Y \_\_\_\_\_ V/div



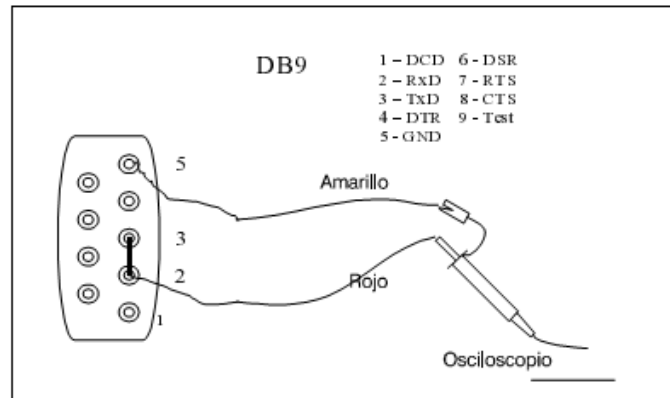
Escala eje X \_\_\_\_\_ s/div

Escala eje Y \_\_\_\_\_ V/div



Escala eje X \_\_\_\_\_ s/div

2. Ejecuta el programa propuesto en el punto 4 del estudio previo y comprueba, utilizando un osciloscopio, la salida de la UART para los casos prácticos del apartado 2 del estudio anterior.



El cable de la figura, realiza un bucle entre el pin de transmisión, TX, y el de recepción, RX. Esto permite a la UART recibir lo que envía.

## 5.- Cuestiones

1.- Si transmitimos a la velocidad binaria de 1200 bits/seg, ¿cuál es la velocidad efectiva en el caso de no haber errores (no hay retransmisiones)?, ¿cómo se puede mejorar la velocidad efectiva en la transmisión serie?

2.- A la salida de la UART (niveles TTL), se coloca el interfaz eléctrico. ¿qué niveles de señal has medido para los "0" y los "1" en el apartado 3 de la práctica? ¿qué consigues mejorar con el interfaz eléctrico?

3.- ¿Para que sirve el bit de paridad en la transmisión asíncrona?

4.- ¿Cómo se sincroniza el bit (temporización de reloj) y cómo se sincroniza el carácter?

# PRÁCTICA 4 TRANSMISIÓN SÍNCRONA

## Objetivos:

- Observar y estudiar los diversos tipos de modulación digital
- Analizar los diferentes tipos de señales en el dominio del tiempo y de la frecuencia
- Observar la diferencia entre detección síncrona (Coherente) y asíncrona (No-coherente).

## Materiales:

- Osciloscopio con 3 puntas
- Entrenador de Comunicaciones
- Tarjeta de Comunicaciones Digitales 2

## Preinforme:

Para la completa comprensión de la práctica el estudiante debe tener claros los siguientes conceptos y exponerlos claramente en el preinforme:

- Modulación ASK
- Modulación FSK
- Modulación PSK
- ¿Cómo se harían estas modulaciones para caso binario?
- Detección (demodulación) síncrona y asíncrona
- ¿Que es Baudio?
- Códigos Manchester, NRZ, RZ

## Procedimiento

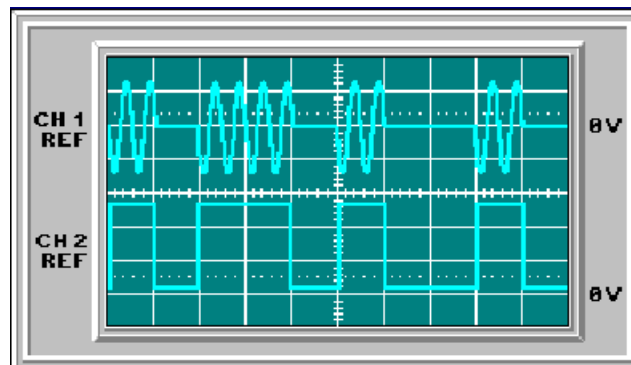
Durante toda la práctica, se debe programar el trigger del osciloscopio en modo externo, con la señal sinc del codificador.

### 1. Modulación ASK

En el modulador ASK/PSK inserte un conector de dos postes para seleccionar el modulador ASK. Como entrada seleccione una señal NRZ. Ubique los potenciómetros DESNIV y EQUIL en la posición de la mitad.

Con el canal 1 del osciloscopio observe la señal en el punto (-) del modulador, y con el canal 2, la señal NRZ. ¿Qué sucede con la señal NRZ a medida que se mueven los potenciómetros DESNIV y EQUIL en sentido horario?

Mueva el potenciómetro DESNIV hasta observar señales de la siguiente forma:

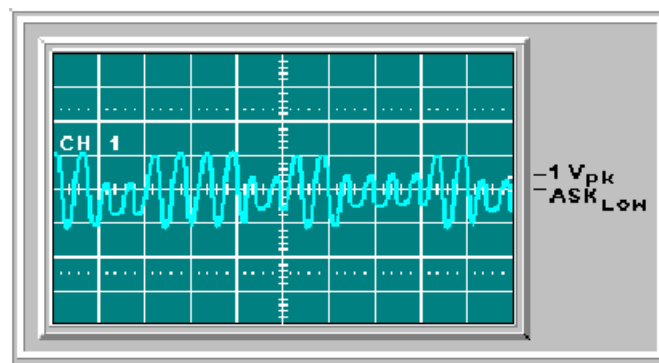


¿Qué tipo de señal se observa a la salida del modulador?

Compare la señal (+) del modulador con la señal a la salida de éste. ¿Qué diferencias y/o semejanzas observa?

## 2. Recepción de señales ASK

Obtenga del modulador una señal de la siguiente forma (nivel bajo 1 V pico).



Conecte la salida del modulador a la entrada del detector asíncrono. Observe los diferentes bloques del proceso de detección, y la salida de cada uno de ellos. Con base en sus observaciones, explique cómo se da la recepción de señales ASK. En el informe, debe estructurar el diagrama de bloques y dibujar la salida en cada punto.

## 3. Modulación PSK

Coloque un conector de dos postes entre la señal NRZ y la entrada al modulador ASK/PSK, y con otro de estos conectores seleccione la modulación PSK (punto P en el modulador). Gire el potenciómetro DESNIV en sentido completamente antihorario, y el EQUIL ubíquelo en la posición de la mitad.

En el osciloscopio, observe las señales NRZ y la salida del modulador. Con base en esa gráfica, explique cómo se da el proceso de modulación PSK.

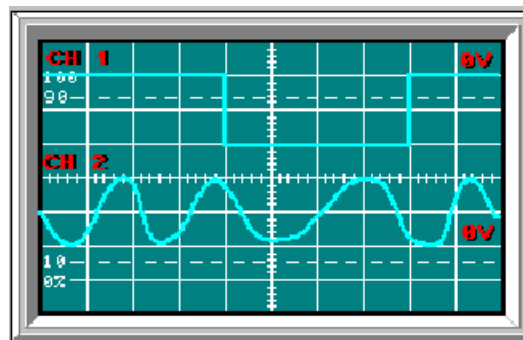
Quite ahora el conector de dos postes entre la señal NRZ y la entrada del modulador, y conecte a ésta la señal RZ que viene del codificador. En el osciloscopio, observe la señal de salida del modulador, y la señal en el punto P. ¿Qué fase tiene la portadora cuando hay cambios en la información? ¿Cuántas fases diferentes existen?

Conecte ahora la señal MAN como entrada al modulador. ¿Cómo es la señal modulada en este caso? Anote las diferencias y/o semejanzas con el caso anterior.

#### 4. Modulación FSK

Con un canal del osciloscopio observe la señal NRZ del bloque de codificación, y ajuste la escala de tiempo hasta observar al menos dos bits de la señal. ¿Cuál es la tasa de baudios de esta señal? Explique cómo la encuentra.

En el bloque de moduladores, conecte la señal NRZ a la entrada por medio de un conector de dos postes. Observe en el canal 2 del osciloscopio la señal de salida del modulador. Haga los ajustes necesarios hasta observar una señal de la siguiente forma:



Mida la tasa de baudios de la señal en el canal 2. Explique cómo lo hace.

Mida la frecuencia de salida cuando la señal banda base es 1, y cuando es cero.

¿Cuál es la relación entre la señal modulada y la banda base?

Observe la señal polar en el canal 2. ¿Cuál es la relación con la NRZ?

¿Cómo se forma la señal FSK? .

Observe la señal a la salida del modulador. ¿Qué tipo de señal observa?

#### 5. Recepción asíncrona de señales FSK

Conecte la salida del modulador FSK a la entrada del filtro pasabanda. Observe la señal de salida del filtro. Y compárela con la señal NRZ del codificador. ¿Qué

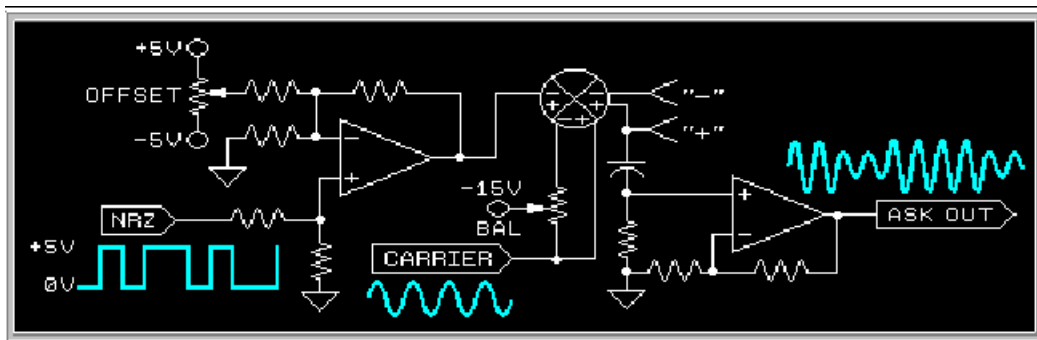
relación observa entre ambas señales? ¿Qué puede decir acerca de la recepción de la señal?

Coloque un conector de dos postes entre el filtro pasabanda y el bloque FWR (rectificador de onda completa). Observe la señal a la salida del comparador. Mueva la fuente positiva en sentido horario hasta que la señal sea similar a la NRZ. Observe las diferentes etapas del proceso de recuperación de la señal y explique cómo se logra obtener nuevamente la señal original.

## INFORME

Además de comentar y analizar de manera ordenada los resultados del laboratorio, el grupo debe investigar acerca de los siguientes tópicos:

1. Por medio de un diagrama de bloques explique el funcionamiento de un PLL, y con base en lo anterior explique cómo se utiliza para la detección síncrona de señales PSK y FSK.
2. ¿Qué ventajas tienen los dos tipos de detecciones: síncrona y asíncrona?
3. ¿Cómo funciona el siguiente circuito modulador de ASK?



4. ¿Qué tipo de modulación digital es más inmune al ruido? Explique mediante un análisis de cada una en presencia de ruido.
5. ¿Qué es un diagrama de ojos, y para qué se usa? En Matlab, haga una simulación de un sistema de comunicaciones digitales (mínimo 4 niveles) a través de un canal con ruido, y observe la salida del diagrama de ojos. ¿Qué observa? (Puede correr uno de los demos ya implementados)
6. ¿Qué es una constelación? Haga la constelación de las siguientes modulaciones:
  - 16 ASK
  - 8 PSK
  - QPSK
  - 16 QAM
7. Conclusiones
8. Bibliografía

# PRÁCTICA 5

## DETECCIÓN Y CORRECCIÓN DE ERRORES (FRAME RELAY)

### Objetivos:

- Ahondar en los fundamentos de la red Frame Relay
- Saber configurar una conexión Frame Relay básica.
- Realizar una configuración avanzada de una conexión Frame Relay: subinterfaces point-to-point y multipoint.
- Verificación de la configuración (comandos show)
- Troubleshooting (debugs)

### 1. Introducción:

#### 1.1 Modo trama (Frame)

Frame Relay fue desarrollada para hacer un mejor uso de la característica del ancho de banda compartido del modo trama, e incluso ahorrarse la desventaja de los largos retrasos en la red. Frame Relay aplica una técnica de conmutación que es considerablemente más rápido que la que se utilizaba en el modo paquete conforme a X.25.

Frame Relay es una técnica orientada a la conexión y tiene paquetes de longitud variable como en el modo paquete. Los paquetes se someten a un mínimo de monitorización durante a lo largo de su camino a través de la red. Si un paquete resulta defectuoso, no se pide la retransmisión: el equipo receptor tendrá que detectar el fallo. Esto supone que la velocidad a través de la red se puede incrementar considerablemente.

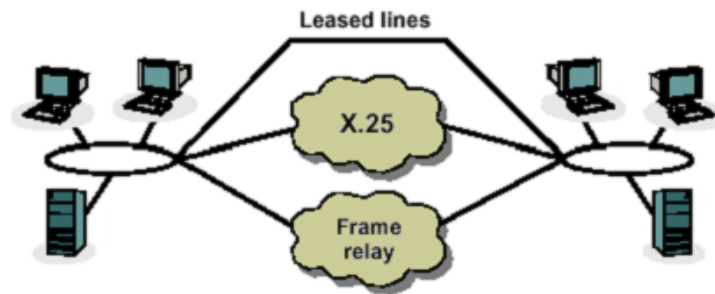
La red Frame Relay necesita enlaces de alta calidad: si no la retransmisión se traduciría en tiempo de respuesta y transferencia mayores.

Frame Relay se utiliza principalmente para interconectar redes de área local. EL tráfico dentro y entre LANs consta de ráfagas cortas pero intensas de datos que requieren una alta velocidad de transmisión. Ejemplos de velocidades de transmisión dentro de una LAN son de 10Mbits en Ethernet, 100Mbits en FastEthernet etc.

Antes de que se desarrollara Frame Relay, había principalmente (y todavía hay) dos posibilidades a la hora de interconectar LANs.

En muchos casos se utilizan líneas alquiladas, aunque esto suponga una infrutilización de la capacidad, ya que las líneas tienen que ser dimensionadas para soportar ráfagas de datos intensas. La otra alternativa es utilizar la red modo paquete

X.25 para la transmisión de datos entre diferentes LANs. Este método, por otra parte, es lento (64 kbit/s o en algunos casos 2Mbit/s)



Diferentes maneras de interconectar LANs

### 1.1.1 Diferencias y semejanzas entre X.25 y Frame Relay

Básicamente, Frame Relay funciona de la misma manera que X.25. Muchos usuarios comparten los recursos de red, y los datos se dividen en paquetes pequeños. Desde que se desarrolló X.25, cuando la calidad de los enlaces de transmisión era bastante mala, implementa funciones de detección de errores y control de flujo. Por ejemplo, un nodo que envía cuatro paquetes y espera el acuse de recibo por parte del nodo receptor antes de enviar otros cuatro paquetes. Este procedimiento explica por qué X.25 consume tanto tiempo y tiene un procesado tan pesado.

La mejor manera de describir las diferencias entre X.25 y Frame Relay es tal vez mediante el modelo OSI. El protocolo X.25 comprende las capas 1-3 del modelo. La capa 3 describe cómo se pueden establecer muchos canales lógicos sobre la misma conexión física, y cómo se establecen y desconectan circuitos virtuales. La capa 2 describe cómo los datos son dispuestos en paquetes, cómo estos paquetes son dispuestos en tramas, y cómo los errores de bit son corregidos mediante la retransmisión. Tanto el nivel 2 como el 3 incluyen procedimientos para el control de flujo, los acuses de recibo, y para comprobar que tramas y paquetes son enviados en el orden correcto. En nivel 1 describe brevemente las conexiones físicas.

Hoy en día, en Frame Relay se utilizan únicamente la capa 1 y una *simplificada* capa 2. En vez de tener un complejo protocolo para la detección y corrección de errores, únicamente se utiliza un procedimiento simple de comprobación de errores para asegurar que los datos erróneos son descartados. Las capas superiores del equipo del destinatario serán las encargadas de cualquier retransmisión. La razón por la que ya no se utiliza la capa 3, es porque los circuitos de hoy son establecidos por el operador de red, lo que significa que ya no es necesaria ninguna función de establecimiento y desconexión de circuitos virtuales individuales (nivel 3), como en X.25. Únicamente se necesitan direcciones de nivel 2.

La desventaja de este método de funcionamiento es que las retransmisiones, si las hay, tienen que atravesar toda la red, y no únicamente el enlace donde tuvo lugar el error. Una mala calidad en el enlace supondrá un gran número de retransmisiones y, en consecuencia, retrasos indeseados. La tasa de error bit ( $<10 \text{ exp } -6$ ) se menciona a menudo como el valor límite para el que Frame Relay empieza a sobrepasar a X.25.



El control de flujo en Frame Relay es bastante más sencillo que en X.25. En vez de utilizar el procedimiento de acuse de recibo que verifica que los datos recibidos son correctos, los usuarios dan por supuesto que la transmisión de datos funciona como es debido. Si la red se satura, el nodo Frame Relay puede enviar una señal de alarma a la LAN, para conseguir que el flujo de tráfico saliente se reduzca.

### 1.1.2 Formato Frame Relay

La trama Frame Relay contiene datos de usuario y una dirección que es similar al LCN en X.25. La dirección, que se llama data link connection identifier (DLCI), indica a qué LAN debería ser enviada la trama actual. El direccionamiento final a determinado host en una LAN es llevado a cabo mediante datos de usuario.

Asignando diferentes valores de DLCI a diferentes tramas, se pueden crear varios circuitos virtuales simultáneos sobre la misma línea física. Como el valor de DLCI únicamente tiene significado local (entre dos nodos) puede que sea necesario cambiarlo durante su recorrido por la red. Los valores de DLCI se cambian en tablas de traducción (conversión) en los nodos Frame Relay.

La trama se delimita mediante dos banderas (una secuencia específica de bits). Entre estas banderas encontramos la dirección, los bits para el control de flujo, los datos de usuario y una función llamada secuencia de comprobación de trama (FCS). El propósito de esta FCS es detectar errores de bit.



Un dato de usuario es enviado en una trama a través de la red

Exactamente como pasaba en X.25, en una trama Frame Relay los campos para los datos de usuario son de longitud variable. El retraso al atravesar los nodos es por tanto impredecible: puede variar de una trama a otra. Esto significa que Frame Relay no se adapta bien a transferencias de servicios síncronos (como telefonía), que requieren que el

retraso sea lo menor posible y, sobre todo, constante. Sin embargo, con el desarrollo de Frame Relay que se sigue haciendo puede que se solucione este problema.

Actualmente, están estandarizados los circuitos virtuales permanentes (PVC). Este término designa a circuitos virtuales que están establecidos permanentemente entre dos LANs. El establecimiento es llevado a cabo por el operador, y los usuarios de las LANs conectadas no pueden controlar el establecimiento o desconexión de los circuitos.

Mediante señalización, un usuario de una LAN puede establecer lo que se conoce como circuito virtual conmutado a un usuario de otra LAN.

### 1.1.3 Resumen de Frame Relay

- Técnica orientada a la conexión.
- Paquetes de longitud variable.
- Velocidad alta: 2-50 Mbit/s.
- No hay retransmisión de link a link (únicamente de extremo a extremo).
- Requiere una buena calidad de transmisión (fibra óptica).

## 2. La red Frame Relay

Frame Relay utiliza un circuito virtual (VC), que es una conexión lógica creada entre dos equipos terminales de datos (DTE) a través de la red del proveedor de servicio.

El equipo del proveedor, al que se conecta el DTE, se llama DCE y se encarga de dar el reloj y los servicios de conmutación en la red.

Cada circuito virtual se identifica con un identificador llamado DLCI. El DLCI del circuito tiene ámbito local, ya que los switches de la red podrán conmutar el valor del DLCI a lo largo de todo el trayecto del circuito virtual.

Cuando se contrata un servicio Frame Relay, para cada circuito virtual se especifica un CIR, que es la velocidad de transmisión de datos promedio máxima que la red se compromete a transportar bajo circunstancias normales, es decir, si transmites a una velocidad superior a CIR los paquetes serán marcados como candidatos a ser descartados en caso de sobrecarga de la red.

El estándar de señalización entre el router (DTE) y el switch Frame Relay (DCE) se llama LMI y existen varios tipos, luego es necesario que el DCE y el DTE utilicen el mismo tipo para poder comunicarse.

Una conexión Frame Relay necesita que, en un circuito virtual, el DLCI local esté asociado (mapeado) a una dirección de nivel de red, por ejemplo dirección IP.

Cuando una trama entra en la red el switch realiza lo siguiente:

- Mira el valor de DLCI entrante.
- Consulta (en una tabla que mapea cada DLCI destino con un puerto) el valor correspondiente al DLCI del extremo remoto.
- Transmite la trama al puerto correspondiente incluyendo los dos valores de DLCI en la cabecera Frame Relay.

Cuando la trama sale por el otro extremo, ya sale etiquetada con el DLCI destino al que es asignada a la entrada de la red.

Este método permite tener múltiples DLCIs sobre un mismo puerto físico de un switch.

## 1. Configuración básica

Para el DTE:

```
Interface Serial 1
Ip address 10.16.0.2 255.255.255.0      **asigna una dir IP a la interface**
Encapsulation frame-relay ietf        **configura la interface como FR**
Bandwidth 64                          **este dato se utiliza para cálculos
en algoritmos de routing dinámico**
Frame-relay lmi-type ansi              **especifica tipo de señalización**
Frame-relay inverse-arp ip 16         **para el ARP inverso**
Frame-relay map ip 10.16.0.1 110 broadcast ietf **mapea el DLCI a una dir IP
destino**
```

Para el DCE, además de la configuración anterior se le añade:

```
Frame-relay switching                  **lo configura como DCE**
Frame-relay intf-type dce              **da el reloj y velocidad de transmisión**
Clock rate 2000000
```

## 2. Configuración avanzada (con subinterfaces)

Dentro de una interfaz física se pueden definir varias subinterfaces lógicas. Cada subinterfaz tendrá que estar conectada a una red distinta. Pueden ser point-to-point, que sólo pueden tener un DLCI, y las multipoint, que pueden tener varios DLCIs.

Configuración de subinterfaces point-to-point:

```
!
Interface serial 0
No ip address                          **cada subinterface tendrá su ip particular**
Encapsulation frame-relay ietf
!
Interface serial 0.110 point-to-point   **especifica que es point-to-point**
Ip address 10.17.0.1 255.255.255.0     **asigna una dirección IP y máscara**
Bandwidth 64
Frame relay interface-dlci 110         **asigna un DLCI a todo el subinterface**
!
interface serial 0.120 point-to-point
ip address 10.18.0.1 255.255.255.0     **tiene que pertenecer a una red distinta**
bandwidth 64
Frame-Relay interface-dlci 120
!
```

Configuración de subinterfaces multipoint:

```
!  
interface serial 2  
no ip address                               **la dir IP se asigna a la subinterface**  
encapsulation frame-relay ietf  
!  
interface serial 2.2 multipoint             **especifica que es multipoint**  
ip address 10.17.0.1 255.255.255.0  
bandwidth 64  
  
frame-relay map ip 10.17.0.2 120 broadcast  **realize un mapeo de DLCI  
para cada dirección distinta**  
frame-relay map ip 10.17.0.3 130 broadcast  
!
```

### 3- Verificación de la configuración

**show interface serial n<sup>o</sup> pto** Muestra la información relativa a los DLCIs usados en la interface.

**show frame-relay traffic** Muestra las estadísticas globales de Frame Relay desde la última reinicialización del router.

**clear frame-relay inarp** Borra los mapeos Frame Relay creados dinámicamente.

**show frame-relay lmi** Muestra las estadísticas sobre la LMI (Local Management Interface)

**show frame-relay map** Muestra las entradas de mapeo actuales y la info sobre las conexiones.

**show frame-relay pvc** Muestra las estadísticas sobre los circuitos virtuales permanentes (PVCs) para las interfaces Frame Relay.

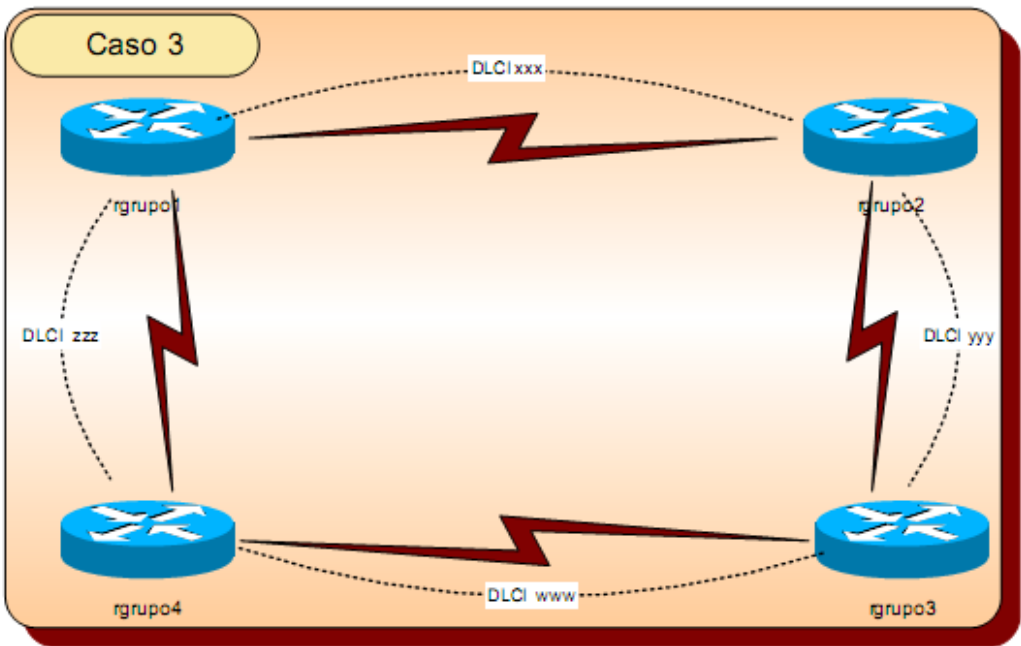
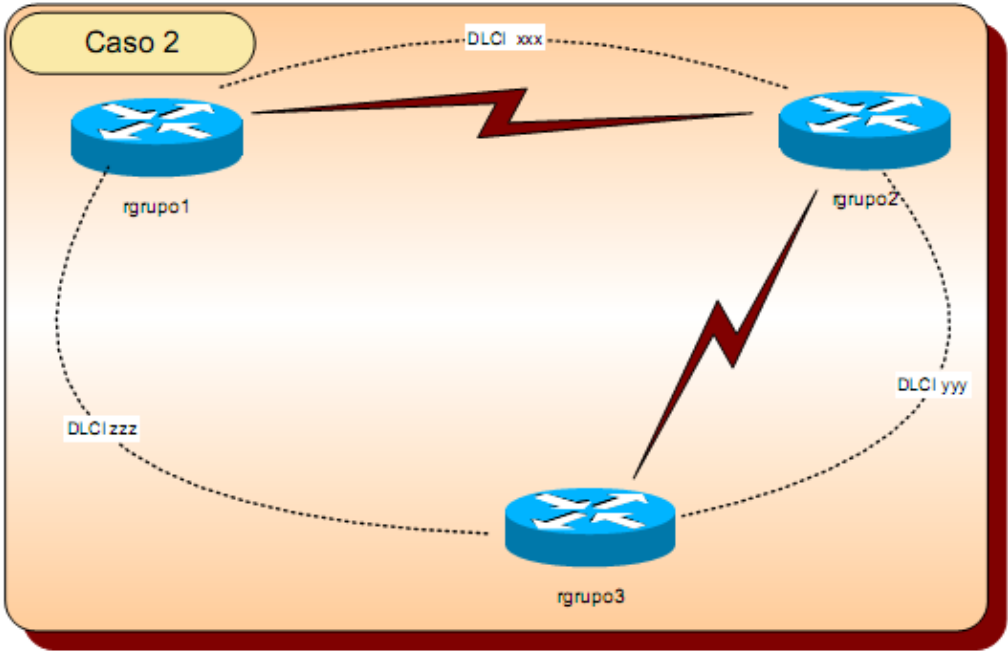
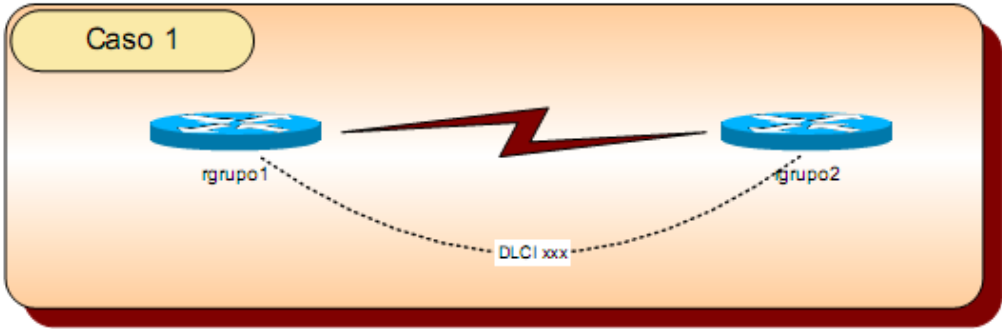
### 4- Troubleshooting

**debug frame-relay** Muestra información sobre los paquetes recibidos en la interface Frame Relay.

**debug frame-relay lmi** Muestra información de los paquetes LMI intercambiados entre el router y el proveedor del servicio.

## 5- Desarrollo

- 1- Caso 1: Entre dos grupos configurar un enlace Frame relay básico
- 2- Caso 2: Realizar la configuración del dibujo-caso2 donde tres routers están conectados por Frame Relay y hay definidos tres circuitos según la figura. Probar desde la LAN de cada router (no viene dibujada) que tienen conectividad a todos los segmentos de la red mediante ping.
- 3- Caso 3: Realizar la configuración de la figura del caso 3. Probar desde la LAN de cada router (no viene dibujada) que tienen conectividad a todos los segmentos de la red mediante ping.
- 4- Enumerar las principales diferencias entre los tres casos





# PRÁCTICA 6

## CONTROL DE FLUJO EN LA TRANSMISIÓN

### 1. Objetivos

Analizar los distintos aspectos de la interfaz serie RS-232.

- Comprender las comunicaciones asíncronas. Entramado.
- Mostrar, manejar y comprender eléctricamente el puerto serie.
- Conocer el equipo de terminal de datos VT-100.
- Manejar un software de comunicaciones del entorno PC.
- Analizar los distintos mecanismos de control de flujo inherentes al puerto serie.

### 2. Material

El material que se empleará en la práctica y que el alumno deberá aprender a manejar es el que se describe a continuación y se esquematiza en la Figura 1.1.



Figura 1.1 Configuración del puesto de trabajo para cuatro alumnos.

- Equipos Hardware.
  - 2 PCs que actúan como DTE's con entorno Windows interconectados mediante los respectivos puertos serie.
  - Un analizador de redes de área extendida Agilent Advisor, intercalado en la línea RS232 que une los dos DTE's.
  - 1 osciloscopio con memoria.
- Software
  - Programa de gestión del puerto serie: Terminal de Windows.
  - Programa Visual HDLC de simulación del protocolo HDLC.
- Varios ficheros de pruebas.

## Control de flujo

Durante la transferencia de información entre dispositivos, puede ocurrir que la velocidad de procesado de uno de ellos sea inferior a la del otro, con lo que se podría perder parte de dicha información si no se utilizase algún mecanismo de arbitraje. Este mecanismo es precisamente el control de flujo.

Un ejemplo muy claro de la necesidad de estas técnicas es la comunicación entre un ordenador personal y una impresora. Evidentemente, la impresora no es capaz de procesar (imprimir) la información a la misma velocidad que le es enviada por el ordenador, debido a sus limitaciones mecánicas. Así, deberá tener algún modo de indicarle al ordenador cuando no puede procesar más información, y cuando vuelve a poder hacerlo. En la interfaz RS-232, existen dos modalidades: el control de flujo hardware y el control de flujo software.

### Control de flujo hardware.

En este caso el control de flujo se realiza mediante las líneas RTS y CTS del RS-232. Cuando el DTE llena su buffer de recepción por encima de un determinado umbral, desactiva la línea RTS para avisar al DCE, el cual deja de transmitir hasta que el DTE no vuelva a activar la línea RTS. Análogamente cuando el DCE no puede procesar la información recibida y quiere avisar al DTE desactiva la línea CTS.

### Control de flujo software.

En este caso tan solo es necesario utilizar las líneas de datos para llevar a cabo el control de flujo. Cuando el receptor no puede procesar más información, lo indica al emisor enviando un carácter predeterminado (**Xoff**) por la línea de datos. En el momento en que puede volver a procesar datos, envía otro carácter (**Xon**) con lo que el emisor vuelve a transmitir.

Debemos hacer hincapié en el hecho que cada una de las distintas modalidades de control de flujo está asociada a un nivel distinto de la arquitectura OSI vista anteriormente. Así, mientras el control de flujo hardware está asociado al nivel físico ya que utiliza circuitos específicos para ello, el control de flujo software no está asociado al nivel físico, sino que se está interpretando la información que viaja por la línea, esta interpretación corresponde, por tanto, a niveles superiores.

### Control de flujo

A parte de los procedimientos vistos anteriormente que también representan un control de flujo intrínseco, cualquiera de los terminales tiene la posibilidad de informar al otro que no se encuentra en disposición de procesar la información. Esto se realiza mediante la trama de supervisión RNR. En el caso de que una estación envíe un RNR, la otra deberá interrogarla periódicamente, con el comando RR, con el bit de poll a 1. Mientras no esté dispuesta, la trama RR tendrá como respuesta una trama RNR, y cuando lo esté, responderá con una RR. Estas dos posibles respuestas tendrán el bit de final igual al de poll de la trama a la que responden.

### 3. Desarrollo de la práctica.

- Busque la codificación ASCII de los caracteres Xon y Xoff.
  - a) Observe la forma de onda correspondiente a la transmisión del BACKSPACE, con la configuración 1200 8N1, y obtenga su codificación ASCII.
  - b) Finalmente, volviendo a la configuración original, lleve a cabo la transferencia del fichero labtel1.txt en modo texto (transferencia ASCII), y mida el tiempo total de la transmisión en el analizador de protocolos<sup>5</sup>.

## Control de flujo

Como ya hemos comentado, el control de flujo se hace necesario para adaptar las velocidades de procesado de datos de distintos elementos en una comunicación. En el caso que nos ocupa, tenemos dos ETDs con la misma velocidad de procesado por lo que, en general, dicho control de flujo será innecesario. Sin embargo, podemos simularlo de forma que nos quedará muy clara la forma en que se realizaría si fuese necesario. Por otra parte, para poder observar lo que ocurre en la línea, se conectará y configurará adecuadamente el analizador de protocolos.

- c) Control de flujo Software: Seleccione en ambos ETDs la modalidad de control de flujo software. Desde uno de los terminales transmita los caracteres de control Xon y Xoff y observe el efecto que produce en el otro. Los caracteres Xon y Xoff pueden simularse tecleando las combinaciones CNTRL+q y CNTRL+s.

Con la ayuda del analizador de protocolos, compruebe la codificación de los caracteres Xon y Xoff en línea, y deduzca que ocurre con la información tecleada en un ETD cuando el otro ETD le ha inhabilitado la transmisión mediante un Xoff.

Realice una transferencia en modo texto y observe el efecto de los caracteres Xon y Xoff al teclearlos desde el terminal receptor. Observe que si el control de flujo seleccionado es software, el estado de las líneas RTS y CTS es indiferente, con lo que incluso podemos abrirlas en el Null-Modem.

- d) Control de flujo Hardware: Configure ambos terminales con control de flujo hardware. Desactive las líneas RTS y CTS del Null-Modem y observe como afectan cada una de ellas a cada terminal. ¿Qué ocurre al activar de nuevo dichas líneas?

¿Como influye la posición del Null-modem respecto al analizador de protocolos (a la derecha o a la izquierda según el esquema de la Figura 1.1) para que las líneas RTS/CTS afecten a uno u otro terminal?

Observe qué ocurre ahora al transmitir los caracteres Xon/Xoff.

- e) Piense algún método para que el terminal receptor no pueda procesar la información recibida y se vea obligado a activar el mecanismo de control de flujo. Póngalo en práctica tanto para el control de flujo software como para el control de flujo hardware.



# PRÁCTICA 7

## MULTIPLEXAJE POR DIVISION DE FRECUENCIA

# PRÁCTICA 8

## MULTIPLEXAJE POR DIVISIÓN DE TIEMPO

### 1. Objetivo

Entender el concepto de multiplexación por división de frecuencia y por división de tiempo

### Introducción

Para conseguir los objetivos propuestos tomaremos como ejemplo un caso real, concretamente, la Red Telefónica Conmutada (RTC). Esto, además, nos permitirá adquirir los conocimientos básicos de la estructura de esta red, su funcionamiento genérico y el de los principales elementos que la componen.

En un primer momento se expondrá la problemática de la interconexión entre N usuarios, obteniendo en una primera aproximación la estructura de la red. Con ello se podrá profundizar en cada uno de los elementos que la componen dando su descripción modular según las funcionalidades de cada uno de ellos. A continuación se expone la estructura real de la RTC donde se hace necesaria la interconexión entre centrales. Esta interconexión se puede realizar por medio de técnicas de multiplexación en el tiempo (TDM) o en frecuencia (FDM).

### Multiplexación por división en el tiempo, sistema MIC.

Evidentemente, la señal entregada por un abonado a la RTC es una señal analógica que deberá ser procesada convenientemente antes de intentar trabajar con ella en una central digital. Este proceso de conversión analógico-digital de la señal se lleva a cabo utilizando la técnica de modulación por impulsos codificados (Pulse-Code Modulation, PCM). En este apartado daremos unos breves conceptos acerca de esta técnica con el fin de que el alumno pueda aprovechar convenientemente la sesión de prácticas. Un esquema de bloques de un sistema de este tipo es el de la Figura 2.16.

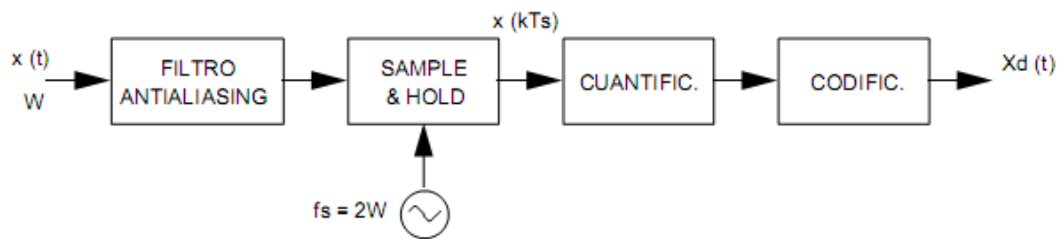


Figura 2.16 PCM. Pulse Code Modulation

La señal de voz  $x(t)$ , proveniente del aparato telefónico, se pasa en primer lugar por un filtro cuya misión es limitarla en banda para evitar el fenómeno de aliasing cuando se proceda a su muestreo. Este muestreo se lleva a cabo en el bloque siguiente (Sample & Hold), con una frecuencia de muestreo  $f_s$  que deberá cumplir la condición de Nyquist, obteniendo una señal discreta en tiempo  $x(kT_s)$  función del bloque siguiente, el cuantificador, es la de discretizar también la señal en amplitud, redondeando los valores de las muestras hasta el valor discreto más cercano en un conjunto de  $L$  niveles. Finalmente, cada uno de estos niveles será codificado convenientemente con lo que la señal analógica original queda definitivamente transformada en una señal digital.

Una vez comprobado que el muestreo de una señal analógica nos permite conservar toda su información siempre y cuando se respete el criterio de Nyquist, observamos que la calidad del sistema PCM va a depender en gran parte de la bondad del bloque de cuantificación. Un cuantificador uniforme subdivide el margen dinámico de la señal en  $L$  niveles, con un paso de cuantificación  $\Delta$  tal que  $\Delta = 2 \cdot V_{\max} / L$ .

Suponiendo la señal de entrada al cuantificador normalizada con  $|x(t)| \leq 1$ , la característica del cuantificador sería como la mostrada en la Figura 2.17.

Con este tipo de cuantificador, se observa que el error máximo para una muestra concreta  $n$  queda acotado a la mitad del paso de cuantificación  $|e(n)| \leq \Delta/2$

De una forma intuitiva, puede decirse que la potencia de ruido dependerá, en general, de la frecuencia relativa de cada muestra de voz, es decir, de su estadística; y que será mínimo cuando sea menor el error en las muestras más frecuentes. Esto sugiere el uso de una cuantificación no uniforme, es decir, que adapte el paso de cuantificación (y por tanto el error instantáneo) a la frecuencia relativa de cada muestra (es decir, el paso de cuantificación será tanto más pequeño cuanto más probable sea la muestra asociada). En telefonía (salvo

E.E.U.U., Canadá y Japón) se usa la ley de cuantificación conocida como ley A. Dicha ley será visualizada en la pantalla del osciloscopio durante la realización de la práctica.

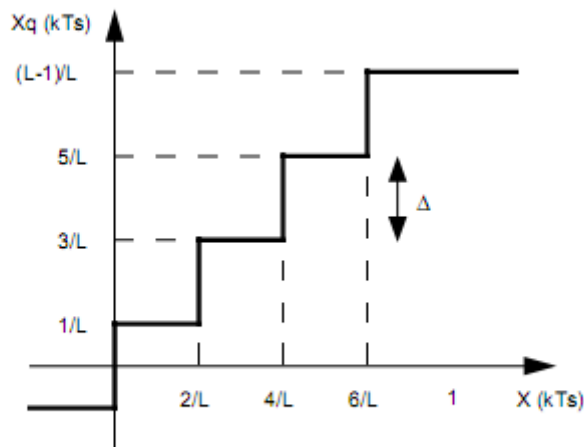


Figura 2.17 Cuantificador uniforme.

## El múltiplex TDM. Sistema MIC 30+2

Ya hemos comentado que en telefonía se toma un ancho de banda para la señal de voz de 4 KHz. Esto nos lleva a una frecuencia de muestreo de 8 KHz para cumplir el criterio de Nyquist. Por otra parte, el cuantificador que se utiliza dispone de un conjunto de 256 niveles distribuidos de forma no uniforme, por lo que se necesitan 8 bits para codificar cada uno de ellos. Con esto, la velocidad de transmisión necesaria para un canal de voz será la de la expresión (2.1).

$$v_t = 8000 \frac{\text{muestras}}{\text{segundo}} \times 8 \frac{\text{bits}}{\text{muestra}} = 64 \text{Kbps} \quad (2.1)$$

Es decir, cada  $T_s$  segundos nuestro sistema ha de ser capaz de transmitir los 8 bits correspondientes a la codificación de la muestra. Sin embargo, supongamos que somos capaces de enviar estos 8 bits de una forma más rápida, por ejemplo en  $(T_s/2)$  segundos. Esto nos llevaría a disponer de otros  $(T_s/2)$  segundos durante los cuales no enviaríamos información al canal. Evidentemente, parece lógico aprovechar este intervalo de tiempo para enviar la codificación de las muestras correspondientes a otro canal de voz. Esta es la idea en la que se basa la multiplexación por división en tiempo (TDM): podemos aumentar el número de canales, siempre y cuando las muestras de cada uno de ellos lleguen al receptor con la suficiente celeridad, es decir, una cada  $T_s$  segundos.

En particular, el sistema MIC 30+2 multiplexa 30 canales vocales, junto a uno de señalización y otro de sincronismos, formando una trama como la de la Figura 2.18.

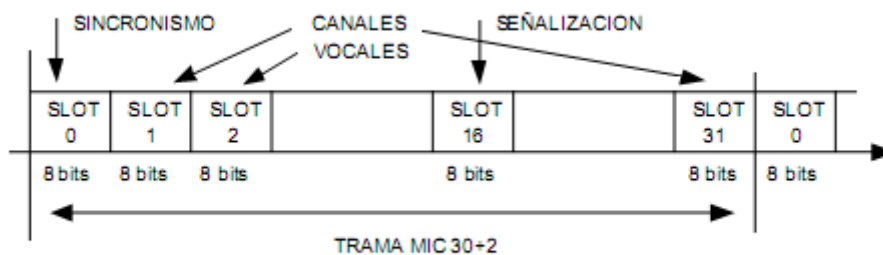


Figura 2.18 Sistema MIC 30+2.

Durante la sesión de laboratorio, se visualizará en el osciloscopio la estructura de estas tramas. En general, si el canal del que se dispone puede transmitir a una velocidad de C bits/seg, entonces el número de señales simultáneas (número de canales) que puede transmitir será el resultante de la expresión. Es decir, se pueden transmitir tantas señales simultáneas como muestras codificadas "caben" entre dos instantes de muestreo consecutivos, tal como se ve en la expresión (2.2).

$$N_{\text{canales}} = C \frac{\text{bits}}{\text{seg}} \times \frac{1 \text{ muestra}}{n \text{ bits}} \times \frac{1}{2 f_{\text{max}} \text{ muestras/seg}} \quad (2.2)$$

## 2. EQUIPO Y MATERIAL

- Osciloscopio digital.
- Generador de funciones.
- Multímetro digital.
- Resistencias ajustables.
- Sondas y cables de conexión.

## 3. DESARROLLO

### Central digital. Conmutación por división en tiempo.

Digitalización y codificación de la señal

- Realice el circuito de la Figura 2.25.
- Sitúe todos los interruptores del módulo ELS-2 en la posición OFF y el selector de modo en la posición PABX.
- Ajuste el generador a 1KHz y 3Vpp
- Coloque el codec 0, casilla 0 en modo lazo (transmita y reciba por la misma ranura temporal) y compruebe que la señal recibida es la misma que la transmitida
- Conecte el canal B del osciloscopio al bus Tx y Dispare el osciloscopio con el canal A y flanco de bajada con la señal de sincronismo de trama (Figura 2.26)

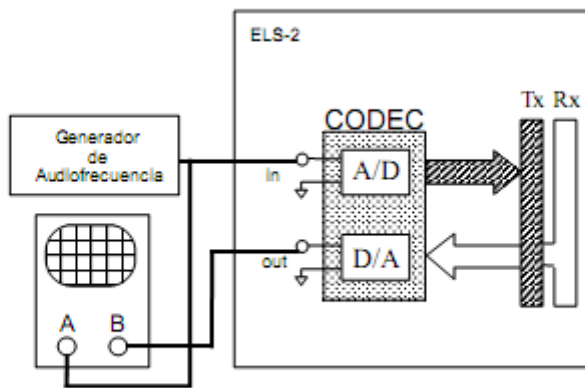


Figura 2.25 Digitalización y codificación de la señal.

- Sitúe todos los interruptores del módulo ELS-2 en la posición OFF y el selector de modo en la posición PABX.
- Ajuste el generador a 1KHz y 3Vpp
- Coloque el codec 0, casilla 0 en modo lazo (transmita y reciba por la misma ranura temporal) y compruebe que la señal recibida es la misma que la transmitida
- Conecte el canal B del osciloscopio al bus Tx y Dispere el osciloscopio con el canal A y flanco de bajada con la señal de sincronismo de trama (Figura 2.26)

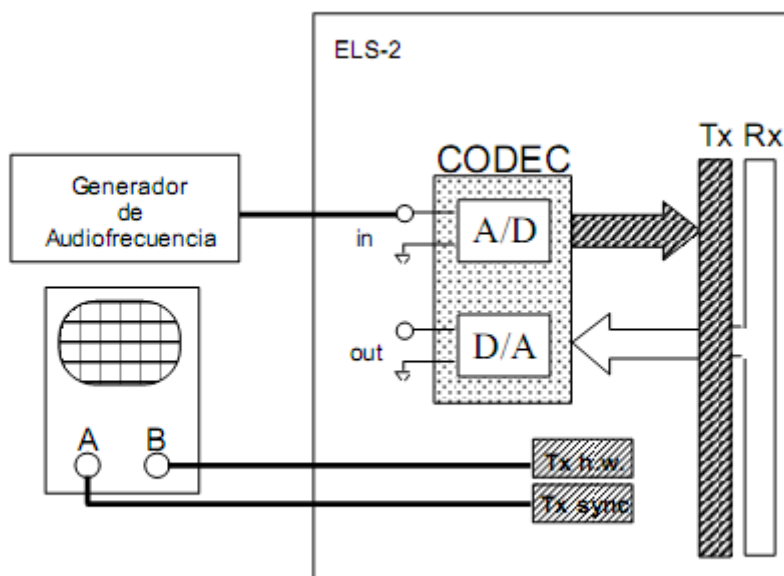


Figura 2.26 Visualización trama MIC 30+2

- Mida la duración de una trama MIC 30+2. ¿Se cumple el criterio de Nyquist?, ¿Cuál es la velocidad de transmisión de una trama MIC 30+2?

- Aplique al CODEC las tensiones continuas de 3V, 0V y -3V. Anote la palabra que se observa en la pantalla (Téngase en cuenta que el CODEC invierte los bits pares de la palabra para transmitirla por el tren MIC). ¿Por qué es difícil visualizar el valor 0?

#### **Simulación de una conmutación**

Para ello asigne dos teléfonos de la central a los codec 4 y 5. Establezca su transmisión y recepción en las ranuras temporales del MIC de forma que se consiga una conmutación entre ambos codecs, o también con otras palabras, que ambos usuarios puedan iniciar una conversación. En esta situación explicar si sería posible incluir un nuevo abonado a la conversación

# PRÁCTICA 9

## CONMUTACIÓN DIGITAL Y ENRUTAMIENTO

### 1. Objetivos

- Analizar los servicios y capacidades ofrecidas por los centros de conmutación digital que componen la estructura de la RDI y de la RDSI-BE.
- Configurar y administrar una red privada constituida por varios centros de conmutación digital, estableciendo el valor de los parámetros que determinen su comportamiento para satisfacer la demanda de servicio de los usuarios de acuerdo a unas determinadas especificaciones.

### 2. Equipamiento

Para la realización de las diferentes partes de la práctica se utilizará el siguiente equipamiento:

- Sistema de conmutación digital (PABX) Siemens HiPath 3550.
- Teléfonos analógicos.
- Teléfonos digitales de Siemens para bus Upn (OptiSet E Advanced Plus y/o OptiSet Base).
- PC con tarjeta RDSI "Conceptronic":
  - Mediante el software RVS-COM Lite (teléfono, videoconferencia, fax,...) y la tarjeta RDSI se podrán establecer comunicaciones RDSI.

### 3. Desarrollo

Para conseguir los objetivos, la práctica se divide en dos partes:

- Una primera parte en la que se realizarán tareas de configuración y administración de la PABX de manera individual, para ofrecer un conjunto de servicios y facilidades acordes a unas determinadas especificaciones.
- Una segunda parte en la que se configurará una red privada formada por tres PABX interconectadas entre sí, permitiendo la comunicación entre todos los terminales de la red de acuerdo a los requisitos indicados.



## Primera parte. Gestión y administración de una PABX

La administración del sistema de conmutación se realizará desde la aplicación de gestión (*HiPath 3000 Manager E*) instalada en un PC conectado a la PABX a través de uno de los buses S0 (interfaz de acceso básico RDSI) configurado como Eurobus.

El modo de funcionamiento de la PABX viene determinado por los parámetros de configuración almacenados en una base de datos denominada KDS. Se partirá de una KDS vacía a la que se le irán añadiendo y modificando los valores adecuados para ofrecer los servicios requeridos.

El primer paso será arrancar la aplicación *HiPath 3000 Manager E* y descargar la KDS almacenada en la PABX, para conocer los valores de configuración actuales. Por defecto, la aplicación *HiPath 3000 Manager E* guarda esta configuración en un archivo denominado *Lastload.kds*. Este nombre puede lógicamente ser cambiado por el alumno.

Una vez transferida la KDS al PC, la aplicación ofrece diferentes menús para realizar todas las tareas de configuración necesarias. Se recomienda al alumno guardar en un fichero propio la configuración de la PABX al final de cada sesión de laboratorio. De esta forma, se puede al inicio de la siguiente sesión transferir desde el PC dicha configuración.

En concreto, se recomienda seguir las siguientes indicaciones:

1. El primer día de laboratorio descargar la KDS vacía.
2. Realizar las modificaciones de configuración necesarias.
3. Transferir la KDS desde el PC a la PABX en modo "delta"; este modo permite "subir la KDS" de forma más rápida, ya que sólo se transfieren las modificaciones realizadas desde la última transferencia; se debe tener en cuenta que las modificaciones realizadas no son efectivas hasta que no se realiza este paso.
4. Repetir los pasos 2 y 3 las veces que sea necesario para cumplir los requisitos de las diferentes partes de la práctica.
5. Guardar la configuración actual propia en un fichero al final de cada sesión de laboratorio.
6. Al inicio de las siguientes sesiones transferir la KDS propia a la PABX y, una vez hecho esto, volver a descargarla al PC, para seguir realizando la configuración del sistema sobre el fichero *Lastload.kds* y subir las últimas modificaciones utilizando siempre el modo delta.

### Configuración de los usuarios

En este apartado, el alumno deberá configurar las extensiones de los terminales conectados a la PABX. Para ello, debe en primer lugar definir un plan de numeración adecuado y asignar a cada terminal el número de extensión planificado.

### Definición del plan de numeración

A cada PABX se encuentran conectados tres terminales digitales Upn, tres terminales analógicos, un terminal telefónico IP y cuatro buses S0, de los cuales en esta primera parte de la práctica se usarán solamente dos. A estos dos buses están conectados dos PC con tarjeta RDSI.

El alumno debe asociar un número de llamada (con tres cifras es suficiente) y un nombre a cada terminal, siguiendo los pasos indicados en el manual<sup>1</sup>. Para esto se recomienda utilizar la numeración definida en las tablas siguientes, donde la letra "X" se sustituirá por un número. Éste corresponderá con la última cifra del Número de llamada de la extensión. Por ejemplo, si el primer terminal digital Upn que se añade en el puesto de la Izquierda es el 100, su nombre debe ser "Upn\_0\_I".

A dos de las extensiones configuradas como puerto sin terminal, se deben asignar los números para los PC. Dichos números también serán acordes con el plan de numeración definido en las tablas y debe ser configurado tanto en el adaptador RDSI del PC como en la tabla de extensiones de la PABX.

*Tabla 1. Plan de numeración PABX Puesto Izquierda*

Número llamada	Nombre	Tipo terminal
100-109	Upn_X_I	Teléfono digital
110-119	Analogico_X_I	Teléfono analógico
120	Bus1_I	Bus S0_1
121-128	PC_Bus1_X_I	PC con tarjeta RDSI
130	Bus2_I	Bus S0_2
131-138	PC_Bus2_X_I	PC con tarjeta RDSI

*Tabla 2. Plan de numeración PABX Puesto Derecha*

Número llamada	Nombre	Tipo terminal
300-309	Upn_X_D	Teléfono digital
310-319	Analogico_X_D	Teléfono analógico
320	Bus1_D	Bus S0_1
321-328	PC_Bus1_X_D	PC con tarjeta RDSI
330	Bus2_D	Bus S0_2
331-338	PC_Bus2_X_D	PC con tarjeta RDSI

## Asignación del número de extensión a los terminales

Una vez definido el Plan de numeración, hay que pasar a configurarlo usando la herramienta *HiPath 3000 Manager E*. Para ello, lo primero es asignar a los buses S0 su condición de Eurobuses para poder después darle un número de llamada. Esto se consigue poniendo el Flag RDSI del módulo STLS4 en los puertos físicos 5-1 y 5-2 con la opción "SO: Euro-bus"<sup>2</sup>.

A continuación hay que asignar los Números de llamada y Nombres correspondientes<sup>3</sup> a cada tipo de terminal según las tablas 1 y 2. Es importante remarcar que se deben introducir los valores en la fila en la que el tipo de terminal (columna Tipo) y el módulo de acceso (columna Access) son los que corresponden con el terminal que se está configurando. Para los PC con tarjeta RDSI hay que elegir una fila en la que el Tipo sea "Puerto sin terminal" o "Extensión S0". Estos, una vez que realicen una llamada, quedarán registrados automáticamente por la centralita en la tabla con su correspondiente "Tipo" y "Acceso".

Completado este punto, transferir la KDS a la PABX y comprobar que los terminales tienen asignados los números de extensión adecuados, realizando las oportunas llamadas entre todos ellos.

## Asignación de facilidades a las extensiones

En este punto se pretende que el alumno realice tareas de configuración de los parámetros de las extensiones. Para ello, se especifican diferentes niveles de privilegio, que conllevan cada uno de ellos una serie de facilidades. El nivel de las distintas extensiones se puede apreciar en la siguiente tabla:

Tabla 3. Planificación de facilidades para los terminales

Nivel de privilegio	Facilidades asociadas	Terminales asignados
Bajo	Llamada en espera	Dos terminales analógicos
Medio	Llamada en espera Ruptura del estado No molestar	Un terminal analógico y un Upn
Alto	Ruptura del estado No molestar Intercalación Supervisión silenciosa	Dos terminales Upn

A continuación, hay que comprobar si la configuración realizada cumple con las especificaciones dadas, para lo que se deben realizar los siguientes apartados y responder a las cuestiones planteadas en ellos:

1. Activar el servicio "No molestar" en una extensión cualquiera. Realizar una llamada desde una extensión con el nivel de privilegio alto o medio a la extensión bloqueada, y después otra llamada desde una extensión con el nivel bajo. Se deben explicar razonadamente las diferencias entre ambos casos.
2. Establecer una comunicación entre dos usuarios cuyo nivel de privilegio sea medio o bajo, e intentar unirse a ella, primero desde un terminal con el nivel bajo, y posteriormente desde otro con el nivel alto. ¿Pueden unirse a la conversación previamente establecida? Justificar lo que ocurre en cada caso y comentar las diferencias.
3. Establecer una comunicación entre un usuario cuyo nivel de privilegio sea alto y otro cuyo nivel sea medio; a continuación, realizar una llamada desde otra extensión a cada uno de ellos. Comentar las diferencias entre ambos casos, explicando por qué el sistema actúa de esta manera y su utilidad.
4. Realizar una llamada desde un terminal con nivel bajo a uno de nivel medio. A continuación, desde un terminal de nivel alto, hacer una supervisión silenciosa de la llamada establecida. Intentar lo mismo desde un terminal de nivel bajo o de nivel medio. Justificar los resultados y comportamiento observado en cada caso.

## **Configuración de puesto de operador (Interceptación)**

En el sistema cabe la posibilidad de configurar alguna extensión para que funcione como "Puesto de operador". Esta extensión, además de recibir llamadas realizadas a su número de directorio, también recibe aquellas dirigidas al puesto de operador, cuyo número es "9" por defecto (este parámetro es configurable). En la PABX se pueden definir distintos operadores según el modo en el que se encuentre ("día" o "noche"), así como las extensiones que van a poder conmutar el sistema entre ambos modos.

Se debe configurar el sistema para que cumpla los siguientes requisitos:

- Un terminal Upn debe actuar como operador en modo noche. Este teléfono será usado más adelante para el apartado Team/Top.
- Otro terminal Upn debe ser el operador en modo día.
- Un terminal analógico será la única extensión capaz de cambiar el modo del sistema<sup>7</sup>.



Realizada la configuración, debe analizarse cómo funciona el servicio. Para ello, seguir el siguiente procedimiento y responder a las cuestiones planteadas, indicando los pasos realizados en cada punto:

1. Llamar directamente al operador (marcando "9") en modo "día". ¿Quién responde?
2. Conmutar el sistema a modo "noche" y llamar de nuevo al operador ¿Quién responde ahora?
3. Volver al modo "día".

## Configuración de Línea directa (Hot Line)

Mediante este servicio una extensión, al descolgar el teléfono, llama a un número predeterminado si no realiza ninguna otra acción.

El alumno deberá configurar dos "Hot Line": uno entre un Upn y un terminal analógico, para que, al descolgar el teléfono, se realice la llamada inmediatamente. El otro "Hot Line" debe configurarse para que otro de los terminales Upn llame a otro terminal analógico después de 5 segundos de descolgar (es decir, una "Llamada de socorro"). Para ello, deben seguirse los pasos indicados a continuación<sup>8</sup>:

1. Definir los dos "Hot Line" especificados.
2. Asignar las extensiones correspondientes a cada uno de ellos

Para verificar el funcionamiento del servicio se tienen que realizar los siguientes pasos y responder a las cuestiones planteadas, indicando las acciones llevadas a cabo para la configuración de los "Hot Line":

1. Descolgar el teléfono Upn del primer "Hot Line" y comentar qué ocurre.
2. Repetir lo mismo con el terminal Upn del segundo "Hot Line" y esperar 5 segundos para ver qué ocurre.
3. Cambiar el segundo "Hot Line" para que el Upn llame al Upn del primer "Hot Line" transcurridos 5 segundos después de haber descolgado. Describir los pasos seguidos y comentar qué ocurre en este caso
4. Eliminar los "Hot Line" configurados, indicando los pasos seguidos.

## Grupos de telecaptura

Un "grupo de telecaptura" es aquel en el que la llamada recibida por un miembro, se señala mediante un mensaje al resto de miembros que tienen un terminal con pantalla. Si el terminal llamado suena un número determinado de veces, la llamada puede ser capturada por otro sin necesidad de conocer a qué extensión están llamando.

El alumno tiene que crear un "grupo de telecaptura"<sup>9</sup> siguiendo los siguientes apartados y respondiendo a las cuestiones planteadas:

1. Crear un "grupo de telecaptura" con las siguientes extensiones: dos terminales Upn y un terminal analógico. Indicar los pasos seguidos y comentar las pruebas llevadas a cabo para comprobar su funcionamiento.
2. Indicar y comprobar otra posible manera de realizar una telecaptura desde un terminal que no pertenece al grupo.
3. Deshacer el grupo antes de pasar al siguiente apartado.

## Grupos de llamada

La centralita ofrece la posibilidad de crear grupos de extensiones con un determinado número de directorio, de manera que todas ellas reciban las llamadas dirigidas al grupo, o bien puedan responderlas en un orden establecido.

El alumno debe crear dos grupos de llamada con las siguientes características<sup>10</sup>:

- Uno de nombre "grupo\_A", cuyos miembros sean 2 terminales analógicos, de forma que las llamadas a dicho grupo se dirijan a todos los miembros a la vez. El número de llamada queda a elección del alumno.
- Otro con el nombre "grupo\_B" y cuyos miembros sean 2 terminales Upn, en el que las llamadas se encaminen siempre a un Upn y, si éste está ocupado, hacia el otro del grupo. El número de llamada también queda a decisión del alumno.

Para verificar el correcto funcionamiento, el alumno debe realizar las siguientes acciones y responder a las cuestiones que en ellas se plantean, indicando los pasos realizados para la configuración de cada grupo:

1. Hacer una llamada al número del grupo A y describir lo que ocurre.
2. Realizar una llamada al grupo B y observar qué sucede. Llamar de nuevo al número del grupo B. ¿Hay alguna diferencia respecto a la llamada anterior? ¿Y si el terminal Upn al que se tienen que encaminar las llamadas primero está comunicando?
3. Llamar al grupo B y no atender la llamada desde ningún terminal. Describir lo que sucede en este caso.
4. Configurar el grupo B para que las llamadas sean recibidas por el siguiente miembro al que atendió la última. Describir los pasos seguidos. Llamar al grupo B y observar qué ocurre. Tras ello volver a llamar al mismo grupo. ¿Existe alguna diferencia? ¿Y si el Upn al que se tienen que encaminar las llamadas primero está comunicando?

Deshacer los grupos de llamada antes de pasar al siguiente apartado.

## Restricción interna de tráfico

Mediante este servicio se pueden restringir las llamadas internas relativas a la centralita.

Se deben crear cuatro grupos con distintas restricciones<sup>11</sup>:

- **"Grupo 1"**: integrado únicamente por un Upn (1), que puede llamar al resto de grupos y sólo puede ser llamado por otro Upn (2). Éste Upn (1) tiene que ser el mismo que se asignó como operador en modo noche, apartado 4.1.4.
- **"Grupo 2"**: formado por el otro Upn (2), puede llamar y recibir llamadas del resto.
- **"Grupo 3"**: lo forman un terminal Upn (3) y un terminal analógico (1); ambos pueden llamar a todas las extensiones excepto al Upn (1), pudiendo recibir llamadas de todos.
- **"Grupo 4"**: Sus miembros son dos terminales analógicos (2 y 3). Estos pueden llamar a todas las extensiones excepto al Upn (1), pero no se pueden llamar entre ellos. Pueden recibir llamadas de todos (excepto de ellos mismos).

Para comprobar las relaciones de los grupos y verificar su funcionamiento, el alumno deberá realizar los siguientes apartados y responder a sus cuestiones planteadas, indicando los pasos seguidos:

1. Comentar brevemente los pasos seguidos para configurar los grupos. Incluir la captura de la matriz VBZ.
2. Llamar desde el Grupo 4 al Grupo 1. ¿Es posible? Probar todas las posibilidades. ¿Es posible llamar desde el Grupo 4 al Grupo 2?
3. Pasar al modo "noche" y llamar desde todos los grupos al operador. Comentar qué sucede y explicar los motivos.
4. Volver al modo "día".

## RESULTADOS

## CONCLUSIONES

## BIBLIOGRAFÍA



# PRÁCTICA 10

## CREACIÓN DE UNA RED LAN

### Objetivo de la práctica:

Implementar una red LAN con diferentes ordenadores, haciendo uso correcto y adecuado de las direcciones IP, de las máscaras y de las rutas por defecto, incluyendo aspectos de gestión y documentación.

### Material:

Apuntes de teoría, varios ordenadores, cable cruzado de la práctica anterior, cables directos, un hub y un switch.

### Primera parte: Documentación

En el laboratorio, disponemos de unos ordenadores conectados a Internet. La conexión se realiza a través de un esquema similar al que se muestra en la figura, es decir los equipos de red para implementar una LAN y esta conectada a un router, que tiene salida a Internet.



En el supuesto de tener que administrar dicha red, necesitaríamos previamente un documento que nos permitiera estudiar el cableado estructurado introducido en el aula.

**Alumno:** Indica la relación de rosetas etiquetadas que disponemos en el laboratorio.

Rosetas: .....

**Alumno:** Indica el nombre de tu ordenador y la roseta en la que está conectado

Ordenador (nombre): ..... Roseta:.....

Para conocer el nombre de tu ordenador, puedes ejecutar el comando "hostname" desde el símbolo del sistema.

**Alumno:** Observando el armario de cableado que disponemos en la planta donde se encuentra el laboratorio, identifica las rosetas del aula que están conectadas y dónde.

Rosetas conectadas: .....

Rosetas no conectadas: .....

¿Existe fibra óptica en el armario de cableado? ¿de qué tipo?.....

Describe los equipos de red que aparecen en el armario de cableado, especificando marca, modelo y número de puertos, explicando brevemente la funcionalidad de cada uno de ellos:

-  
-

**Alumno:** Sobre la figura anterior, indicar el tipo de dispositivo asociado a cada símbolo y tipo de cable utilizado: directo o cruzado.

### Segunda parte: Configuración de IP en Windows

Como parte de la documentación de una red, es importante conocer de la figura anterior el rango de direcciones IP disponibles para los hosts en la red.

**Alumno: Ejercicio:** Suponed que la dirección del router o puerta por defecto es la IP 10.10.10.1 y máscara 255.255.255.128. ¿cuántos bits dispone la máscara, es decir, cuántos 1s?..... ¿qué rango de IP tendremos disponibles para los hosts?.....

Arranquemos el ordenador y seleccionemos en el menú de arranque el sistema operativo Microsoft Windows XP. Vamos inicialmente a analizar la configuración de cada ordenador desde dicho sistema operativo. Para ello, simplemente vamos a realizar consultas a través de la línea de comandos. Debemos realizar los siguientes pasos, tal como se muestran en la figura:

**Paso 2.1.-** Abrir la ventana del símbolo de llamadas al sistema:



Y ejecutar el comando *ping* para ver si existe conectividad a la red (*ping "destino"* es un comando que permite averiguar si el destino existe, está activo o si lo podemos alcanzar nosotros, por medio de una pregunta que el destino nos responde). Concretamente vamos a realizar *ping* a 147.156.1.1 que es la máquina de resolución de nombres dentro de la *Universitat* (DNS). Analizar las opciones del comando, simplemente tecleando el comando sin argumentos.



Y también vamos a realizar la siguiente prueba, para ver el camino realizado para llegar a 147.156.1.1 con el comando *tracert*



**Paso 2.2.-** Documentación de dirección IP

**Alumno:** ¿Qué MAC e IP/máscara tiene tu ordenador?

MAC: \_\_\_\_\_ IP/ Máscara \_\_\_\_\_ / \_\_\_\_\_

Para ello ejecutamos el siguiente comando *ipconfig / all*

**Alumno:** ¿Qué ruta por defecto tiene tu ordenador? \_\_\_\_\_ ¿puedes hacerle ping a la ruta por defecto? \_\_\_\_\_

Para ello ejecutamos el siguiente comando *route print*

**Alumno:** ¿Qué rango de IPs pueden ser utilizadas con dicha configuración? \_\_\_\_\_

**Paso 2.3.-** Modificación de dirección IP (**no lo vamos a realizar**).

Para modificar la IP de nuestro ordenador, deberíamos seguir los siguientes pasos desde el menú de inicio: configuración> conexiones de red y acceso telefónico > conexión de área local, y sobre él pulsar el botón derecho del ratón, para seleccionar propiedades de la pila TCP/IP y allí modificar la IP, máscara, puerta de enlace y DNS. Estos pasos se describen en las siguientes figuras.



## CONCLUSIONES

## BIBLIOGRAFÍA