**ENLACE**

## Descripción:

Se trata de una aplicación que implementa una entidad de enlace.

* Permite comprender el concepto de abstracción de niveles en el modelo de referencia OSI o en cualquier otro modelo arquitectónico de redes.
* Se ha codificado un protocolo simple de nivel de enlace de acuerdo con las especificaciones de las normas OSI.
* Se ha implementado una biblioteca de funciones de acceso al nivel de enlace con las primitivas (o funciones) especificadas en las normas ISO.

## Entorno de desarrollo y lenguaje:

* Windows XP/VISTA/7.
* Lenguaje de Programación C.
* Biblioteca de simulación del puerto serie que permite enviar y recibir carácter a carácter.

## Características:

La aplicación permite la comunicación de dos procesos utilizando los servicios de nivel de enlace. Se implementa un protocolo de nivel de enlace orientado a carácter utilizando el código ASCII (CCITT nº 5).

* + Un único programa ofrece las funcionalidades de ambos extremos en la comunicación.
  + Se muestra en pantalla la primitiva del servicio invocada así como la composición de las tramas.
  + Provee de un método de control de errores (checksum o CRC)
  + Los nombres de las primitivas y los servicios proporcionados se ajustan al modelo OSI.
  + Para mostrar el funcionamiento del servicio se ha implementado una aplicación usuaria del nivel de enlace que permite obtener el código Morse de frases cortas.

## Nivel Físico

### Biblioteca de nivel físico

Para simular el comportamiento del puerto serie se utiliza una biblioteca (fisico.dll) que proporciona las siguientes primitivas:

BOOL f\_inicio\_puerto (char lugar, char \*ordenador, char \*clave,   
 DWORD bps, void (\*funcion\_lectura)(void), DWORD tasa\_error)

Inicia la comunicación creando la tubería que simulará el puerto serie. Los parámetros de entrada son:

* + lugar, indica el lado de la comunicación en la tubería. I lado izquierdo, D lado derecho.  
    Ha de abrirse primero el lado I.
  + ordenador, específica en que equipo se va a abrir la tubería. Si el valor es NULL indica que se trabajará en el ordenador local.
  + clave, clave que identificará la tubería
  + bps, velocidad de transmisión en bits por segundo. Simulará un retardo en escribir un carácter.
  + funcion\_lectura, función que será llamada cuando existan datos listos para ser leídos. El prototipo de la función será void f (void).
  + tasa\_error, tasa de error que se introducirá en el proceso de escritura para simular errores en nivel físico. Se mide en número medio de caracteres enviados por cada error. Por ejemplo, si se indica el valor 10 significa que estadísticamente se generará un error por cada 10 caracteres transmitidos. El valor 0 indica transmisión sin errores.

DWORD f\_puedo\_escribir(void)

Devuelve 1 (verdadero) si se puede escribir, o (falso) si no se puede escribir.

DWORD f\_esperar\_escribir(timeout)

Espera, sin consumir CPU, hasta que se pueda escribir. Devuelve 1 (verdadero) si se puede escribir y 2 si ha vencido el timeout . El timeout se expresa en milisegundo. Si el valor del timeout es INFINITE la espera es hasta que se pueda escribir y si el valor es 0 se examina si se puede escribir y retorna inmediatamente.

DWORD f\_escribir(char c)

Escribe un carácter. Devuelve 0 en caso de éxito y -1 si hay sobreescritura.  
Si se ha iniciado con una tasa de errores el carácter será alterado aplicando dicha tasa.

DWORD f\_puedo\_leer(void)

Devuelve 0 (falso) si no hay carácter y la conexión sigue activa. Devuelve 1 si hay carácter para leer, 2 si han cortado la conexión, 3 ambos casos.

DWORD f\_esperar\_leer (timeout)

Espera, sin consumir CPU, hasta que se pueda leer . Devuelve los mismos valores que f\_puedo\_leer más el valor 4 si ha vencido el timeout. El timeout se expresa en milisegundo. Si el valor del timeout es INFINITE la espera es hasta que se pueda leer y si el valor es 0 se examina si se puede leer y retorna inmediatamente.

DWORD f\_leer(char \*c)

Escribe el carácter. Devuelve 0 en caso de éxito y   
-3 si se cortó la comunicación,  
-2 si no hay nada que leer,  
-1 si sobreescritura

### Ejemplos de uso

1) [**Sondeando con consumo de CPU**](https://diaweb.usal.es/diaweb/archivos/10015927ejemplo1.c)

2) [**Esperando, sin consumo de CPU**](https://diaweb.usal.es/diaweb/archivos/10015928ejemplo2.c)

3) [**Función de rellamada (Invocada cuando exista dato listo para leer)**](https://diaweb.usal.es/diaweb/archivos/10015929ejemplo3.c)

NOTA: En estos ejemplos no se han tenido en cuenta los errores que pueden devolver las funciones f\_\*. En la implementación de la aplicación si se han considerado.

## Nivel de Enlace

### Servicio proporcionado

Se facilita un servicio con conexión. Las primitivas del servicio son:

L\_CONNECT () Establecimiento de la conexión. Servicio confirmado   
L\_DATA (dato). Envío de datos no confirmado   
L\_DISCONNECT () Desconexión. Servicio no confirmado  
L\_RESET () Notificación de errores. Servicio no confirmado

|  |
| --- |
| L\_CONNECT.request (source\_address, destination\_address, priority) L\_CONNECT.indication (source\_address, destination\_address, priority) L\_CONNECT.response (source\_address, destination\_address, priority) L\_CONNECT.confirm (source\_address, destination\_address, priority) |
| L\_DISCONNECT.request (source\_address, destination\_address) L\_DISCONNECT.indication (source\_address, destination\_address, reason) |
| L\_DATA.request (source\_address, destination\_address, data) L\_DATA.indication (source\_address, destination\_address, data) |
| L\_RESET.indication (source\_address, destination\_address, reason) |

a) Servicio con conexión

### Especificación del protocolo

#### Tipo de Protocolo y Formato de las tramas

Protocolo orientado a carácter utilizando el código ASCII (CCITT nº 5). Los caracteres que se utilizan en transmisión de datos son:

|  |  |
| --- | --- |
| Para delimitar de bloques | |
| SYN (Syncronous Idle) | Dos o más de estos proporcionan la sincronización de trama. |
| SOH (Start of Header) | Indica el principio de una secuencia de caracteres que constituyen la cabecera de un mensaje. |
| STX (Start of Text) | Principio de los datos del bloque, datos de usuario. |
| ETX (End of Text) | Indica el final de un bloque (cuando se han transmitido varios trozos) y el final de texto de un mensaje (cuando todo el texto va en la misma trama). |
| Para control del dialogo | |
| EOT (End of Transmision) | Final de la transmisión y normalmente implica la liberación del enlace. |
| ENQ (Enquiry) | Petición de respuesta de la otra estación o para establecer el enlace. |
| ACK (Afirmative Acknowledgement) | Respuesta afirmativa, un bloque se ha recibido correctamente. |
| Para garantizar la transparencia | |
| DLE (Data Link Escape) | Se utiliza para cambiar el significado de los caracteres de control que le siguen. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Char | Dec | Oct | Hex | Char | Dec | Oct | Hex | Char | Dec | Oct | Hex | char | Dec | Oct | Hex |
| (nul) | 0 | 0000 | 0x00 | (sp) | 32 | 0040 | 0x20 | @ | 64 | 0100 | 0x40 | ` | 96 | 0140 | 0x60 |
| (soh) | 1 | 0001 | 0x01 | ! | 33 | 0041 | 0x21 | A | 65 | 0101 | 0x41 | a | 97 | 0141 | 0x61 |
| (stx) | 2 | 0002 | 0x02 | " | 34 | 0042 | 0x22 | B | 66 | 0102 | 0x42 | b | 98 | 0142 | 0x62 |
| (etx) | 3 | 0003 | 0x03 | # | 35 | 0043 | 0x23 | C | 67 | 0103 | 0x43 | c | 99 | 0143 | 0x63 |
| (eot) | 4 | 0004 | 0x04 | $ | 36 | 0044 | 0x24 | D | 68 | 0104 | 0x44 | d | 100 | 0144 | 0x64 |
| (enq) | 5 | 0005 | 0x05 | % | 37 | 0045 | 0x25 | E | 69 | 0105 | 0x45 | e | 101 | 0145 | 0x65 |
| (ack) | 6 | 0006 | 0x06 | & | 38 | 0046 | 0x26 | F | 70 | 0106 | 0x46 | f | 102 | 0146 | 0x66 |
| (bel) | 7 | 0007 | 0x07 | ` | 39 | 0047 | 0x27 | G | 71 | 0107 | 0x47 | g | 103 | 0147 | 0x67 |
| (bs) | 8 | 0010 | 0x08 | ( | 40 | 0050 | 0x28 | H | 72 | 0110 | 0x48 | h | 104 | 0150 | 0x68 |
| (ht) | 9 | 0011 | 0x09 | ) | 41 | 0051 | 0x29 | I | 73 | 0111 | 0x49 | i | 105 | 0151 | 0x69 |
| (nl) | 10 | 0012 | 0x0a | \* | 42 | 0052 | 0x2a | J | 74 | 0112 | 0x4a | j | 106 | 0152 | 0x6a |
| (vt) | 11 | 0013 | 0x0b | + | 43 | 0053 | 0x2b | K | 75 | 0113 | 0x4b | k | 107 | 0153 | 0x6b |
| (np) | 12 | 0014 | 0x0c | , | 44 | 0054 | 0x2c | L | 76 | 0114 | 0x4c | l | 108 | 0154 | 0x6c |
| (cr) | 13 | 0015 | 0x0d | - | 45 | 0055 | 0x2d | M | 77 | 0115 | 0x4d | m | 109 | 0155 | 0x6d |
| (so) | 14 | 0016 | 0x0e | . | 46 | 0056 | 0x2e | N | 78 | 0116 | 0x4e | n | 110 | 0156 | 0x6e |
| (si) | 15 | 0017 | 0x0f | / | 47 | 0057 | 0x2f | O | 79 | 0117 | 0x4f | o | 111 | 0157 | 0x6f |
| (dle) | 16 | 0020 | 0x10 | 0 | 48 | 0060 | 0x30 | P | 80 | 0120 | 0x50 | p | 112 | 0160 | 0x70 |
| (dc1) | 17 | 0021 | 0x11 | 1 | 49 | 0061 | 0x31 | Q | 81 | 0121 | 0x51 | q | 113 | 0161 | 0x71 |
| (dc2) | 18 | 0022 | 0x12 | 2 | 50 | 0062 | 0x32 | R | 82 | 0122 | 0x52 | r | 114 | 0162 | 0x72 |
| (dc3) | 19 | 0023 | 0x13 | 3 | 51 | 0063 | 0x33 | S | 83 | 0123 | 0x53 | s | 115 | 0163 | 0x73 |
| (dc4) | 20 | 0024 | 0x14 | 4 | 52 | 0064 | 0x34 | T | 84 | 0124 | 0x54 | t | 116 | 0164 | 0x74 |
| (nak) | 21 | 0025 | 0x15 | 5 | 53 | 0065 | 0x35 | U | 85 | 0125 | 0x55 | u | 117 | 0165 | 0x75 |
| (syn) | 22 | 0026 | 0x16 | 6 | 54 | 0066 | 0x36 | V | 86 | 0126 | 0x56 | v | 118 | 0166 | 0x76 |
| (etb) | 23 | 0027 | 0x17 | 7 | 55 | 0067 | 0x37 | W | 87 | 0127 | 0x57 | w | 119 | 0167 | 0x77 |
| (can) | 24 | 0030 | 0x18 | 8 | 56 | 0070 | 0x38 | X | 88 | 0130 | 0x58 | x | 120 | 0170 | 0x78 |
| (em) | 25 | 0031 | 0x19 | 9 | 57 | 0071 | 0x39 | Y | 89 | 0131 | 0x59 | y | 121 | 0171 | 0x79 |
| (sub) | 26 | 0032 | 0x1a | : | 58 | 0072 | 0x3a | Z | 90 | 0132 | 0x5a | z | 122 | 0172 | 0x7a |
| (esc) | 27 | 0033 | 0x1b | ; | 59 | 0073 | 0x3b | [ | 91 | 0133 | 0x5b | { | 123 | 0173 | 0x7b |
| (fs) | 28 | 0034 | 0x1c | < | 60 | 0074 | 0x3c |  | 92 | 0134 | 0x5c | | | 124 | 0174 | 0x7c |
| (gs) | 29 | 0035 | 0x1d | = | 61 | 0075 | 0x3d | ] | 93 | 0135 | 0x5d | } | 125 | 0175 | 0x7d |
| (rs) | 30 | 0036 | 0x1e | > | 62 | 0076 | 0x3e | ^ | 94 | 0136 | 0x5e | ~ | 126 | 0176 | 0x7e |
| (us) | 31 | 0037 | 0x1f | ? | 63 | 0077 | 0x3f | \_ | 95 | 0137 | 0x5f | (del) | 127 | 0177 | 0x7f |

Figura 5. Caracteres ASCII y su numeración en decimal, octal y hexadecimal. Los caracteres que aparecen como nombres entre paréntesis (Por ejemplo, (nl)) son caracteres no imprimibles.

#### Estructura de la trama

En la figura 6 se muestra la estructura de las tramas de control y de datos. Normalmente al principio de cada una se envían un par o más de caracteres SYN para mantener la sincronización de carácter.

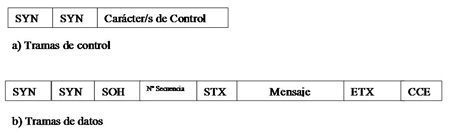


Figura 6. Estructura de la trama  
NOTA: CCE = caracteres de control de error.

#### Funcionamiento del protocolo

##### Establecimiento

En la figura 7 se muestra la secuencia para establecer el enlace. La repetición de la secuencia también tiene un número de intentos límite.

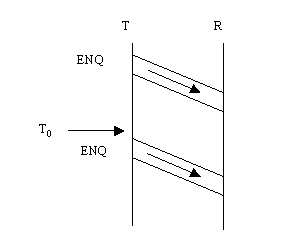
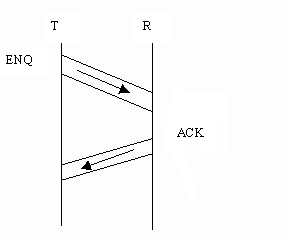


Figura 7. Establecimiento de conexión y repetición de establecimiento de conexión.

##### Terminación del enlace

Se logra mediante el envío de EOT. Este carácter no necesita aceptación, por lo que si se pierde no se alcanza la situación de reposo. Por este motivo, a veces antes de iniciar el enlace mediante ENQ se envía EOT.

##### Transferencia de información

Al transmitir cada trama se establece un plazo de espera que si vence provoca la retransmisión de la trama. En recepción se establece otro plazo de espera al recibir el principio de trama, con el fin de proteger en el caso de que por un error no llegue el final de trama. El receptor enviará ACK nº trama esperada tanto en caso de asentimiento de la trama si coinciden el nº de trama recibido con el esperado como en el caso contrario donde el receptor descartará la trama duplicada pero igualmente notifica con un asentimiento la trama que espera recibir. Igualmente si la trama no es correcta (falla el código de control de errores) o no se ha recibido el final de trama se enviará ACK nº\_trama\_esperada.

##### Transparencia

Para garantizar la transparencia se utiliza la técnica de inserción de carácter. Este carácter es DLE, que precederá a los caracteres de control. En caso de que aparezca DLE en el texto, se duplica el carácter DLE. Por ejemplo:   
Información a transmitir:

a 9 STX ACK 26 DLE ETX 3 ty

Trama transmitida:

DLE STX a 9 STX ACK 26 DLE DLE ETX 3 ty DLE ETX CCE

##### Implementación

Se establecen las restricciones siguientes:   
TRANSMISION UNIDIRECCIONAL DE DATOS   
TRANSMISOR Y RECEPTOR SIEMPRE DISPUESTOS   
CANAL DE COMUNICACION REAL QUE COMETE ERRORES   
TIEMPO DE PROCESO NO DESPRECIABLE

Al enviar una trama se arranca un plazo de espera (timeout) y en caso de vencimiento se retransmite la trama y cierto número de veces. Puede suceder que se envíe una trama duplicada (cuando se pierde la señal de asentimiento). Por tanto, hay que diferenciar la trama que se envía por primera vez de la retransmisión. Para ello las tramas se numeran con un número de secuencia que irá en la cabecera de la trama. En este caso basta con un bit (ya que no se emite la siguiente trama hasta recibir el asentimiento del anterior) por lo que las tramas irán numeradas 0, 1 alternativamente.

A continuación se muestra en pseudocódigo de cómo pueden ser los procedimientos emisor y receptor:

Procedimiento emisor:

Pide mensaje  
Inicializa t-siguiente=0 /\* numero de secuencia de trama a enviar \*/  
Repetir

Construye trama; /\* Inserta el nº de secuencia t-siguiente \*/

Envía trama

Arranca plazo de espera

Espera evento /\* trama correcta, error o timeout \*/

Si trama correcta

Pide mensaje

Actualiza t-siguiente; /\* 0,1 alternativamente \*/

Siempre

Procedimiento receptor:   
  
Inicializa t-esperada=0 /\* trama esperada \*/  
Repetir

Espera evento /\* trama correcta o error \*/

Si trama correcta

Recoge la trama

                    Si nº de secuencia correcto

                             Entrega el mensaje

         Actualiza t-esperada /\* 0,1 alternativamente \*/

                    Envía señal

Siempre

El tiempo del plazo de espera (timeout) es mayor que 2td + 2tp donde:   
td es el tiempo del desplazamiento o transmisión de la señal de origen a destino  
tp es el tiempo empleado en procesar la señal (se supone igual en ambos ordenadores)

## Nivel de aplicación

Para mostrar el funcionamiento del protocolo de enlace implementado se ha realizado una aplicación que obtiene el código Morse de una frase corta. Uno de los extremos establece la conexión con el otro y le envia una frase que se introduce por teclado. El otro extremo devuelve el código Morse correspondiente a la frase que una vez recibido se muestra por pantalla. Los caracteres del código Morse se separan con el caracter #. Para finalizar la ejecución de la aplicación, hay que escribir la palabra FIN (mayúsculas).  
  
Para lanzar ambos extremos automáticamente, se provee de los scripts correspondientes ([**morse\_1.bat**](https://diaweb.usal.es/diaweb/archivos/10015862morse_1.bat) y [**morse\_2.bat**](https://diaweb.usal.es/diaweb/archivos/10015863morse_2.bat))  
c:enlace>morse\_1   
c:enlace>morse\_2  
  
extremo 1 (se conecta y envía una frase)>Enlace en Redes  
extremo 2 (Calcula el código Morse de la frase y se lo envía)

extremo 1 (Muestra por pantalla el código Morse)>

.  # -. # .-.. # .- # -.-. # .# # . # -. #  # .-. # . # -.. # . # ... #

extremo 1> Continuar enviando frases o FIN para finalizar

Se supone que los datos que intercambian ambos extremos se limitan siempre al tamaño máximo de carga útil de la trama.  
  
El usuario podrá configurar (desde código):

* Nombre del equipo con el que se realiza la comunicación
* Velocidad de la transmisión
* Tasa de errores
* Tamaño máximo de la carga útil de la trama
* Si se habilita o no la visualización de las primitivas y composición de las trama

Para ello, hay que configurar un fichero denominado enlace.ini (p. ej.: [**enlace.ini**](https://diaweb.usal.es/diaweb/archivos/10015864enlace.ini)) que el programa deberá leer para configurarse adecuadamente. A continuación se muestra el contenido de el fichero enlace.in:

; Este es el fichero enlace.ini  
[DLL]  
; Poner Equipo=nombreEquipo para especificar la comunicación con otro PC  
Equipo=  
; Tasa de errores  
Errores=0  
  
[Serie]  
Velocidad=9600  
  
[Enlace]  
tamTrama=80  
  
[Preferencias]  
; 1 para si, 0 para no  
verTramasYprimitivas=1

Al finalizar ambos extremos, muestran en pantalla la siguiente información:

* Número total de tramas transmitidas (las correctas + las erróneas)
* Número de tramas erróneas
* Máximo número de retransmisiones para una misma trama

## 