

Tema 7: Comunicación serie asíncrona

SOLUCIÓN DE EJERCICIOS PROPUESTOS

Ejercicio 1

En los últimos años han aumentado los sistemas basados en una codificación diferencial. Explique el fundamento de dicha codificación y diga 2 ejemplos de aplicación, explicando para cada uno de ellos las mejoras en comparación con los antiguos sistemas utilizados anteriormente.

La codificación diferencial es una forma de transmitir bits en una comunicación serie, utilizando una diferencia de tensión entre 2 hilos. De este modo, si la diferencia es positiva, los bits tendrán un valor correspondiente (0 ó 1), mientras que si la diferencia es negativa, los bits tendrán el valor opuesto (1 ó 0).

En general, se puede decir que utilizando una codificación diferencial a través de un cable de par trenzado de 2 hilos, la comunicación es más robusta que cualquier otra comunicación diferente en paralelo o en serie. Por lo tanto, podemos alcanzar más distancias o comunicaciones más rápidas o ambas cosas a la vez.

Hay muchos ejemplos de codificación diferencial en la actualidad, pero tres de ellos podrían ser los siguientes (otros son SATA, Firewire, Ethernet, etc.).

- RS-485: Fue uno de los primeros casos de comunicación diferencial y consiguió aumentar las distancias para la comunicación serie (desde 20 m con RS-232 hasta 1,2 km con RS-485).
- USB: Este modo de comunicación ha sustituido completamente hoy en día a RS-232. La comunicación diferencial garantiza una distancia de máx. 25 metros utilizando velocidades muy altas (y hubs) (desde 20 Kbps con RS-232 hasta 12 Mbps con USB 1.1 o incluso 10 Gbps con USB3.1).

Ejercicio 2

Explique las ventajas de la comunicación USB frente a la comunicación RS-232 y las diferencias entre USB y Firewire.

1) Ventajas de USB frente a RS-232.

- Permite más velocidad, longitud de cable e inmunidad a las perturbaciones electromagnéticas debido a la transmisión diferencial.
- Permite alimentar los dispositivos.
- Es Plug & Play.
- Se puede conectar en caliente.
- USB es realmente un bus, mientras que RS-232 es un puerto punto a punto.

2) Diferencias entre USB y Firewire: Son 2 interfaces complementarias orientadas a diferentes aplicaciones

- Firewire está orientado a la transmisión multimedia con una comunicación isócrona y el objetivo de la comunicación es completamente diferente a USB: lo más importante NO es que la información llegue CORRECTA. Lo más importante es que la información llegue A TIEMPO.
- USB está orientado a la transmisión de datos sin errores, por lo que la comunicación es más lenta que Firewire (1,2 Gbytes con el estándar 1394b), aunque a partir de USB3.0, USB es más rápido que Firewire.

Ejercicio 3

Indique las características principales de una conexión Firewire.

La característica principal de Firewire es que está orientado a la transmisión multimedia con una comunicación isócrona y el objetivo de la comunicación es completamente diferente a USB. Lo más importante NO es que la información llegue CORRECTA. Lo más importante es que la información llegue A TIEMPO.

Inicialmente se permitían las siguientes velocidades de transferencia (S100: 98 Mbps, S200: 196 Mbps, S400: 393 Mbps) pero el estándar 1394b desarrollado en 2002 permite velocidades de transferencia entre 800 y 1200 Mbps.

Los dispositivos pueden conectarse utilizando cualquier configuración (estrella, Daisy Chain, etc.) y el número máximo de dispositivos para conectarse a un único dispositivo es de 63 pudiendo llegar hasta 60.000 dispositivos utilizando bridges.

La conexión física se puede realizar con.

- Cable de 6 hilos (alimentación y 2 pares trenzados con doble apantallamiento) para aplicaciones normales, con hasta 4,5 metros con alimentación de 8-30V/1,5A -> Caro, grande y rígido.
- Cable de fibra óptica de plástico (POF) para aplicaciones profesionales, con hasta 70 metros -> Fibra más barata.

Ejercicio 4

Se tiene un SED (sistema electrónico digital) que quiere comunicar de manera serie con un PC utilizando los puertos tradicionales (LPT y COM). La salida serie asíncrona del SED utiliza niveles TTL (0-5V) con 19200 baudios, 8 bits de datos, 1 bit de paridad y 1 bit de parada.

a) Indique los componentes y las conexiones básicas para esta comunicación, suponiendo que el PC está a 2 metros de distancia del DES.

b) ¿Cómo cambiaría el sistema para que la comunicación funcione con un PC situado a 100 metros del SED?

a) El puerto serie tradicional de un PC convencional es un puerto RS-232, que es asíncrono y completamente compatible con los requisitos, excepto con los niveles

TTL. La solución sería utilizar un transductor TTL-RS232, por ejemplo un MAX232. Para garantizar la comunicación de 2 metros el MAX232 se situaría lo más cerca posible del SED para que la comunicación larga sea RS-232.

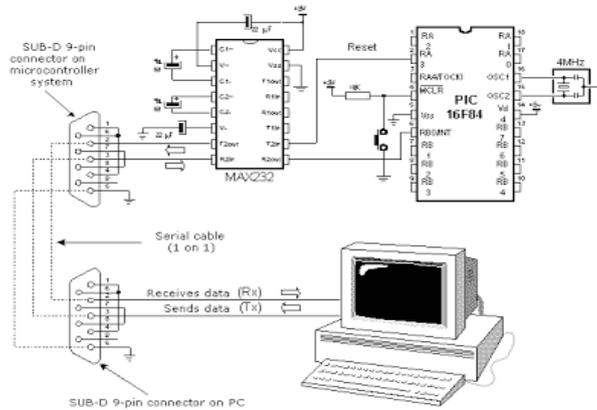


Figura: <http://embeddedmy.blogspot.com/2015/07/lesson-6-part-1-usart-transmitting-data.html>

b) Los niveles de tensión de RS-232 no garantizan más de 20 metros, por lo que es necesario utilizar otro interfaz para conseguir 100 metros. Por ejemplo una comunicación RS-485, que puede llegar a 1,2 km utilizando una codificación diferencial.

La solución sería utilizar un transductor TTL-RS485, por ejemplo un MAX485 conectado al SED. Pero también es necesaria una conversión RS-485-RS-232 para la conexión al PC, por ejemplo utilizando un MAX-485 para la conversión de RS-485 a TTL y un MAX-232 para convertir de TTL a RS-232 para el puerto RS-232. La solución completa es así.

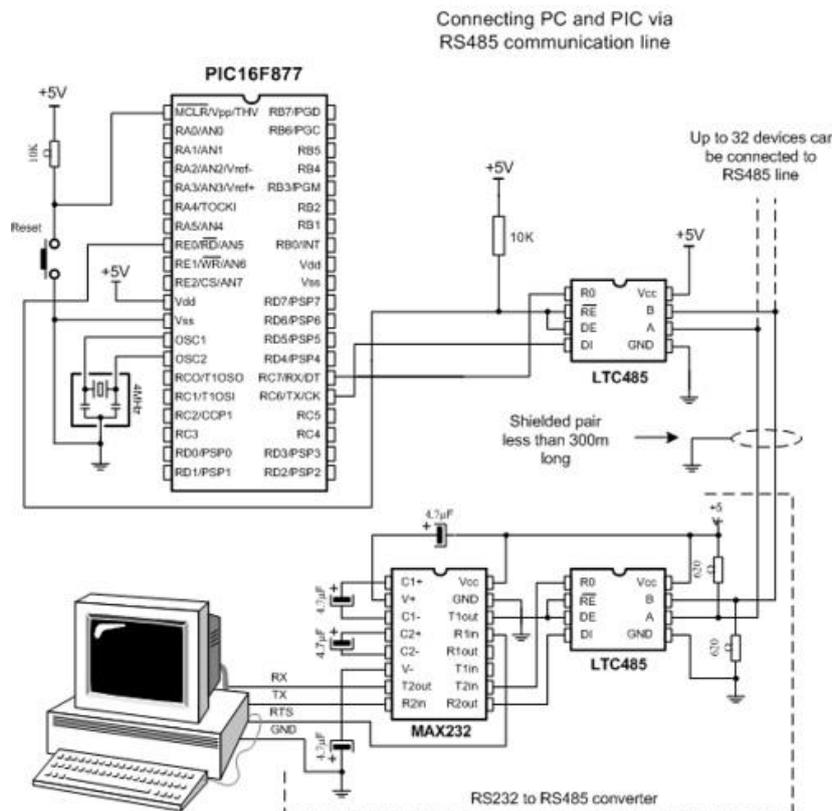
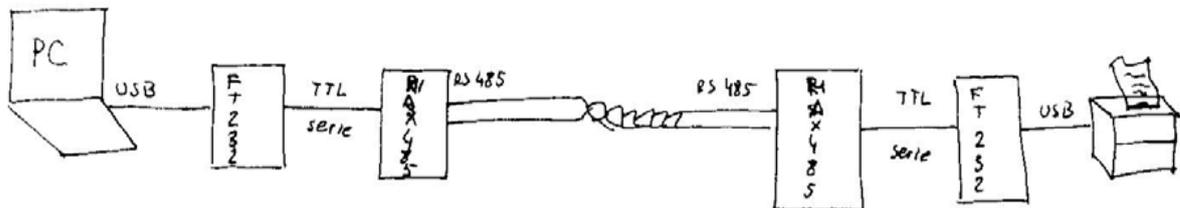


Figura: <https://dinotronics.wordpress.com/2008/11/19/>

Ejercicio 5

Se tiene que conectar una impresora USB a 200 metros de un PC. Dibuje y justifique brevemente cómo implementaría esta comunicación teniendo en cuenta que el único puerto serie del PC es un puerto USB.

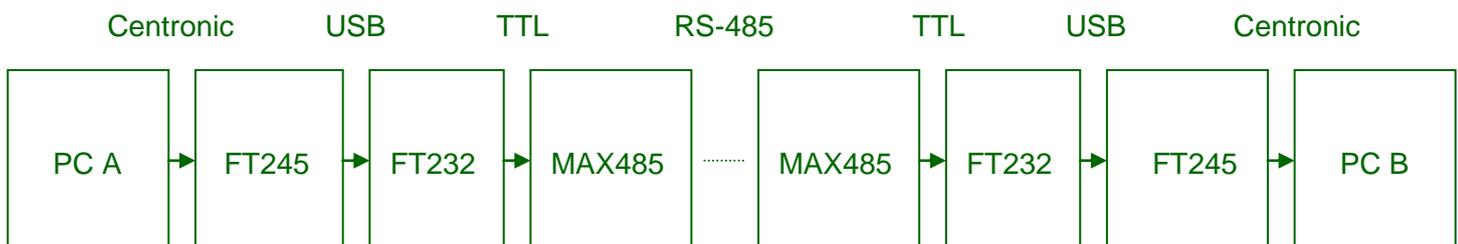
Para conseguir una distancia tan larga, es necesario sustituir la comunicación USB por otro tipo que permita más de 5 metros, por ejemplo, utilizando una comunicación RS-485. Esto se puede conseguir convirtiendo de USB a TTL (usando el transductor FT232) y luego convirtiendo de TTL a RS-485 (usando el transductor MAX485). La misma solución debe estar en el otro extremo de la comunicación.



Ejercicio 6

Se tienen que conectar 2 PCs entre sí (A y B), que sólo tienen puertos Centronics, para que la comunicación sea unidireccional (A envía a B). Los PCs están a 500 metros de distancia el uno del otro. Dibuje y justifique brevemente cómo implementaría esta comunicación utilizando únicamente los componentes explicados en el curso.

Para conseguir una distancia tan larga, es necesario sustituir la comunicación Centronics por otro tipo que permita más de 10 metros, por ejemplo, utilizando una comunicación RS-485. Esto se puede conseguir convirtiendo de Centronics a USB (usando el transductor FT245), luego convirtiendo de USB a TTL (usando el transductor FT232), y finalmente convirtiendo de TTL a RS-485 (usando un transductor MAX485). La misma solución debe estar en el otro extremo de la comunicación.

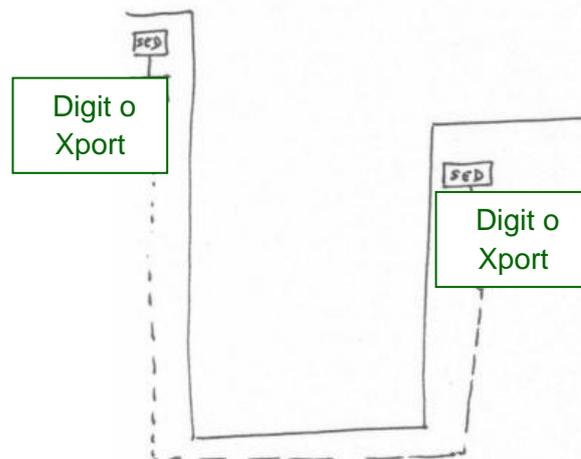


Ejercicio 7

Un SED basado en un microcontrolador LPC2129 debe conectarse a otro SED similar situado en el edificio de enfrente. La distancia directa entre los edificios es de 50 metros, pero el primer SED está situado en la planta 20 (300 metros de altura) y el segundo SED está situado en la planta 10 del otro edificio (150 metros de altura). La velocidad debe ser de 1Mbps. Resuelva este problema con solución cableada dibujando la solución y explicando brevemente su decisión.

La solución puede ser con cables y hay que alcanzar una distancia de $300+150+50$ metros, más las esquinas del sistema de cableado, así que aproximadamente 600 metros. Puede elegir entre RS-485 o Ethernet (10BaseT con repetidores).

Pero la velocidad es de 1 Mbps, y esto sólo es posible con Ethernet, por lo que la única solución podría ser utilizar un transductor Xport Digit o Xport Lantronix conectado al SED.



Ejercicio 8

Debe ampliar un SED diseñado originalmente con un microprocesador LPC2129. En el diseño se encuentra un límite porque es imposible comunicar con otros dispositivos utilizando el puerto serie. Pero este sistema debe comunicar con un PC situado en un pueblo a 250 km de distancia. En esta ciudad sólo se encuentran los servicios de suministros básicos (agua, energía eléctrica, gas y red telefónica hasta 56 kbps).

Otra restricción es que la ampliación del sistema debe realizarse con extrema rapidez, por lo que hay que intentar utilizar el mayor número de dispositivos comerciales posible, evitando el desarrollo de nuevos circuitos si es posible. Para evitar costes adicionales, el cliente final suministra algo de material.

- 5 tarjetas inalámbricas
- 3 tarjetas Ethernet, 10Base-5
- 1 pendrive USB para Bluetooth

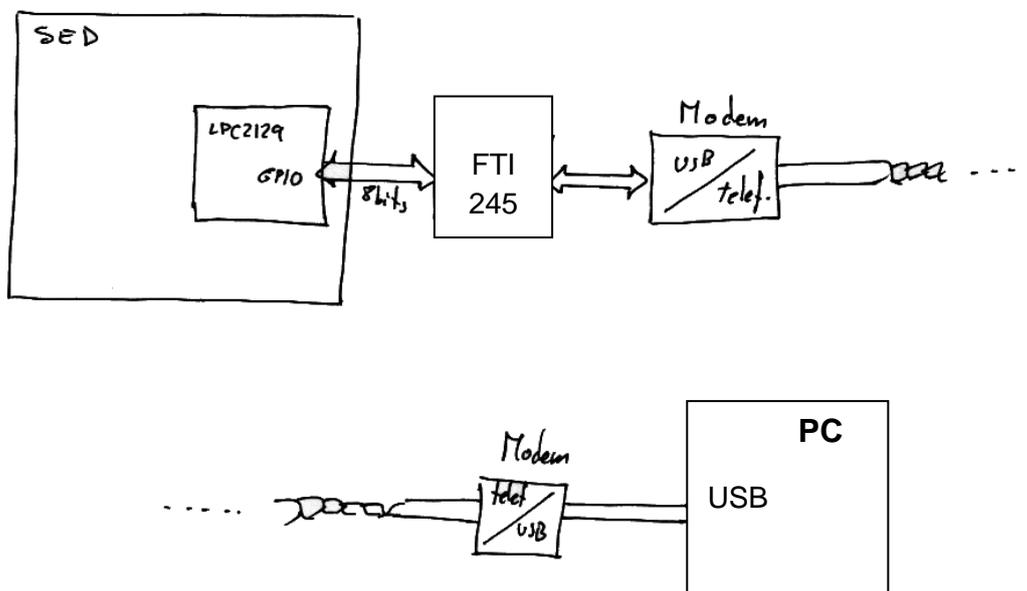
- 3 módems de 33,6 kbps con conexión USB
- 2 tarjetas PCI para GPIB
- 8 tarjetas Firewire

Los requisitos para la transferencia de datos son que cada mensaje debe tener 25 bytes y cada mensaje debe ser transferido cada 5 minutos.

Describe la solución tecnológica para este problema, dibujando un diagrama de bloques con los diferentes dispositivos y sus conexiones. Recuerde que el SED está basado en un microcontrolador LPC2129. Justifica su respuesta.

- La velocidad de transmisión necesaria es muy baja, por lo que no es un límite para el sistema.
- La distancia es tan grande que no es posible utilizar soluciones tecnológicas "locales" (comunicaciones paralelas, I2C, SPI, RS-232, RS-485, USB, Firewire) Así que es obligatorio utilizar la red telefónica. Es seguro que no hay Internet o ADSL en el pueblo.
- La solución debe ser utilizar una comunicación por módem a 33,6 Kbps, utilizando el módem suministrado.
- Éste se comunicará con el PC directamente a través de la interfaz USB y con el SED utilizando un transductor FTDI paralelo-USB (FT245). Es imposible utilizar USART.

Por tanto, el diagrama de bloques podría ser el siguiente.



Ejercicio 9

Comente brevemente las diferencias entre ADSL y Ethernet, así como su funcionalidad conjunta.

Ambos métodos de comunicación tienen medios de acceso físicos diferentes, pero son compatibles entre sí. ADSL está diseñado para utilizar la red telefónica convencional, mientras que Ethernet utiliza cables Ethernet (10baseT).

Ethernet está diseñado para comunicar PCs entre sí en sistemas "locales" (oficinas, edificios, empresas, etc.), mientras que ADSL está diseñado para conectar con sistemas remotos.

ADSL permite una comunicación más lenta que Ethernet (ADSL: Enlace de descarga 50 a 100Mbit/s - Enlace de subida; 10 a 20Mbit/s, Ethernet: 100Mbit/s o más).

Sin embargo, gracias al uso de módems (de todos modos necesarios para ADSL) ambos tipos de comunicación pueden conectarse entre sí. La señal llega a casa a través de ADSL utilizando la red telefónica y el router la convierte en Ethernet, que puede ser difundida en los dispositivos del hogar utilizando una red local.

Ejercicio 10

Explique brevemente la funcionalidad de los códigos de detección de errores explicados en el curso. Si tiene la siguiente trama: 10001010-10010100-01010010-01010010-101010-10110111, ¿cuál sería la trama a enviar con el LRC resultante?

1) Bit de paridad (par o impar).

Se añade un 0 (impar) o un 1 (par) al final de cada byte del mensaje, el receptor hace lo mismo con el mensaje recibido y ambos bits de paridad deben ser iguales.

Pero sólo detecta el error si el número de bits de error es sólo impar. Si el error cambia 2, 4, 6 bits, etc., este método no funcionará, por lo que es un sistema muy débil.

Fue útil hace años, cuando era muy difícil desarrollar algoritmos potentes de detección de errores debido a los PCs, pero hoy en día sólo se utiliza para sistemas muy simples que no necesitan un algoritmo de detección muy exacto.

Original Data	Even Parity	Odd Parity
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

Figura: <https://www.codespeedy.com/find-the-parity-of-a-number-using-python/>

2) Códigos de paridad con más de 1 bit.

Trata de aumentar la seguridad utilizando circuitos combinatoriales para el control de los bytes transmitidos, por lo que ahora es posible detectar errores si el número de ellos es par o impar. Así, la seguridad del mensaje transmitido aumenta.

Se añaden varios bits (C1 ...) al final del byte, que son resultado de varios circuitos combinatoriales, el receptor hace lo mismo con el mensaje recibido y estos bits (C1 ...) deben ser iguales.

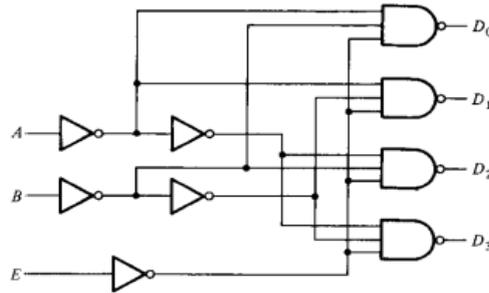


Figura: <https://www.transtutors.com/questions/a-combinational-circuit-is-specified-by-the-following-three-boolean-functions-implement-1737573.htm>

3) Cyclic Redundancy Check (LRC).

La idea ahora es utilizar la trama completa (con todos los bytes). Se hace una operación XOR entre todos los bytes de la cadena transmitida, por lo que el código LRC es el resultado de esta operación. Se envía ahora al final de la comunicación.

El receptor realiza la misma operación y el resultado debe ser el mismo o cero (si también ha realizado la operación XOR con el código enviado por el emisor).

Es bastante robusto y muy sencillo de desarrollar de forma hardware o software.

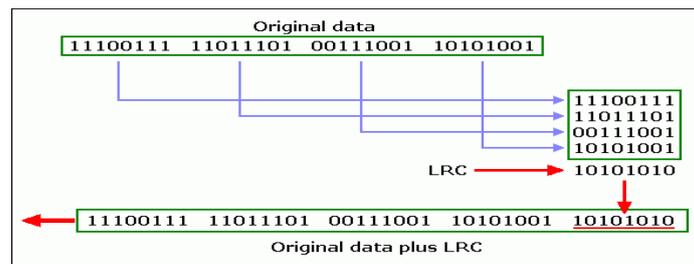


Figura: <https://t4planet.tripod.com/longitudinalredundancycheck.html>

4) Comprobación de redundancia cíclica (CRC).

Este último método es el más robusto pero más difícil de implementar.

La filosofía es utilizar un código creado por el resto de la división de toda la cadena transmitida por un divisor conocido.

Pero hay que adaptar esta idea para simplificar las operaciones y el circuito, de forma que la aritmética utilizada se base siempre en base 2 -> Eso significa sólo operaciones XOR y desplazamiento.

Características importantes a tener en cuenta.

- Los divisores se suelen representar con polinomios basados en base 2.
- El divisor debe tener un bit más que el resto para poder obtenerlo. Por ejemplo, un CRC de 16 bits debe utilizar un divisor de 17 bits.
- Los polinomios razonables deben tener el bit más significativo y el menos significativo con "1".

Existen muchos códigos CRC, por lo que es importante que el emisor y el receptor conozcan el código CRC a utilizar. Hay varias versiones.

- CRC-12 -> $x^{12} + x^{11} + x^3 + x + 1$ (110000001011) -> Para disquetes

- CRC-16 -> $x^{16} + x^{12} + x^2 + 1$ -> Para discos duros
- CRC-CCITT -> $x^{16} + x^{15} + x^2 + 1$ -> El CRC más utilizado actualmente
- CRC-32 -> $x^{32} + x^{26} + x^{23} + \dots + 1$ -> Se utiliza para tramas muy largas

Es importante utilizar el CRC necesario para el mensaje enviado. Por ejemplo, no es lógico utilizar un CRC-32 para enviar sólo 5 bits. En este caso sería lógico el método de paridad.

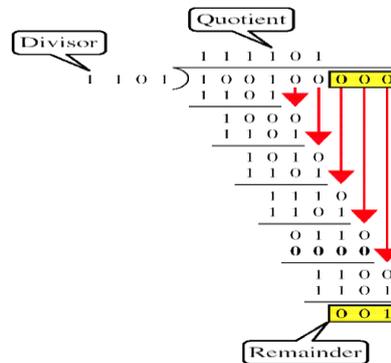


Figura: <https://prayasnotty.wordpress.com/2010/11/23/>

Con la trama 10001010-10010100-01010010-010010-101010-10110111, hay que hacer la operación XOR de los bytes para obtener el LRC final.

```

10001010
10010100
----- XOR
00011110 Resultado
01010010
----- XOR
01001100 Resultado
01010010
----- XOR
00011110 Resultado
10101010
----- XOR
10110100 Resultado
10110111
----- XOR
00000011 Resultado

```

Por tanto, la trama final a enviar sería la siguiente.

10001010-10010100-01010010-01010010-10101010-10110111-**00000011**

Ejercicio 11

Si se tiene la siguiente trama: 1111101010011, ¿cuál sería la trama a enviar con un CRC de 3 bits del tipo x^3+x^2+1 ?

Con la trama 1111101010011 y un CRC de 3 bits del tipo x^3+x^2+1 (1101), necesito hacer operaciones XOR y desplazar los resultados hasta obtener el último resto.

```

1111101010011 000
1101
----- XOR
0010101010011 000
  1101
----- XOR
0001111010011 000
  1101
----- XOR
0000010010011 000
  1101
----- XOR
0000001000011 000
  1101
----- XOR
0000000101011 000
  1101
----- XOR
0000000011111 000
  1101
----- XOR
000000000101 000
  110 1
----- XOR
0000000000011 100
  11 01
----- XOR
0000000000000 110
  
```

El resto de la división es: **110**.
 Así que la trama final a enviar sería la siguiente.
1111101110111-110