

## Tema 10: Comunicación Asíncrona Serie (USART)

### SOLUCIÓN DE AUTOEVALUACIÓN

#### Ejercicio 1

En la comunicación serie asíncrona se trata de transmitir la información:

- Dividiendo la información en palabras
- Cada palabra se transmite bit a bit empezando por el menos significativo
- Es necesario el uso de marcas de inicio y fin para establecer puntos de sincronismo
- a) y c) son correctas
- b) y c) son correctas
- a), b) y c) son correctas**
- ninguna es correcta

#### Ejercicio 2

En la comunicación serie asíncrona es imprescindible:

- Para la transmisión de símbolos, se necesita protocolizar el envío de forma tal que en ambos extremos se sepa de antemano cuánto dura cada símbolo, así como la estructura de la trama.**
- No se debe señalar el inicio y el fin de la trama de bits ya que existe un reloj que nos permite recuperar la información con cada flanco.
- Tener un dispositivo Máster que coordina el envío de la información entre los dos extremos.
- Ninguna de las anteriores
- Todas las anteriores

#### Ejercicio 3

En la comunicación serie asíncrona usando la norma RS232:

- Los símbolos siempre se representan con tensiones unipolares
- Los símbolos siempre se representan con tensiones bipolares**
- Supera a la norma RS485 en cuanto a distancia de trabajo máxima
- Todas las anteriores

#### Ejercicio 4

Si la USART se configura como (9600, 8, N, 2), la comunicación serie es:

- Síncrona binaria a 9600Bd, con reloj a 8 MHz, sin bits de paridad.
- No es una configuración realizable.
- Es una configuración asíncrona a 9600Bd con datos de 8 bits, control de paridad par y 1 bit de parada.
- Es una configuración asíncrona a 9600Bd con datos de 8 bits, control de paridad par y 2 bits de parada.
- Es una configuración asíncrona a 9600Bd con datos de 8 bits, sin control de paridad y 2 bits de paridad.**

## Ejercicio 5

En una USART con velocidad de señalización asíncrona a 9600Bd:

- La velocidad de señalización y de transmisión coinciden.
- El tiempo que ocupa un bit es de 0.1ms aproximadamente.
- El tiempo que ocupa un bit lo determina el reloj del dispositivo Máster.
- La a) y la b) son correctas.**
- Ninguna es correcta

## Ejercicio 6

El chequeo de paridad par es:

- Bit de relleno que se añade al final del carácter transmitido para que se cumpla que, en todo carácter transmitido, la cantidad de 1s sea un número par.**
- Bit de relleno que se añade al final del carácter transmitido para que se cumpla que, en todo carácter transmitido, la cantidad de 0s sea un número par.
- Bit que se añade al final de un carácter transmitido para indicar el fin de la trama.
- Bit que se añade al inicio de un carácter transmitido para indicar el inicio de la trama.

## Ejercicio 7

En la tarjeta Discovery estudiada, para realizar envíos de datos mediante una comunicación serie asíncrona con el PC a través de un puerto USB:

- La información va y viene a través del cable USB conectado para alimentación, depuración, etc.
- La información debe pasar por un circuito que convierta los datos de salida a la norma RS232.
- La información debe pasar por un circuito que convierta los datos de salida NRZ a tramas USB.**
- La tarjeta ya tiene un circuito MAX232 que convierte a la norma RS232. Por esa razón se debe conectar a un puerto serie.

## Ejercicio 8

En el STM32L152:

- No existen interrupciones asociadas a la Transmisión y Recepción de datos mediante la USART. Sólo existen flags que indican que los buffers de entrada o salida están llenos.
- La Transmisión y Recepción de datos mediante la USART puede realizarse mediante polling o mediante interrupciones. Las interrupciones saltan cuando los buffers están vacíos.
- La Transmisión y Recepción de datos mediante la USART puede realizarse mediante polling o mediante interrupciones. Las interrupciones saltan dependiendo del estado de los buffers de salida y entrada.**
- La Transmisión y Recepción de datos mediante la USART puede controlar solamente mediante polling para la transmisión y mediante interrupción para la recepción.

## Ejercicio 9

La instrucción HAL\_UART\_Receive(&huart1, texto, 1, 10000);:

- Provoca que la USART2 se quede esperando un carácter durante 10s. Si no llega se sale del ciclo. Si en ese intervalo llega un dato, lo almacena en texto.
- Provoca que la USART1 se quede esperando un carácter un tiempo determinado. Si en ese intervalo llega un dato, lo almacena en la variable texto.**
- Provoca que la USART1 se quede esperando una trama de 10000 caracteres durante 10s y lo almacena en la variable texto. Si no llegan en 10s se sale del ciclo.

## Ejercicio 10

Si utiliza las funciones HAL\_UART\_Receive\_IT(...) y HAL\_UART\_Transmit(...) en su programa principal, probablemente:

- Va a programar la Transmisión y Recepción de datos con un modelo de atención por interrupciones.
- Va a programar la Transmisión de datos con un modelo de atención por interrupciones.
- Va a programar la Transmisión y Recepción de datos con un modelo de atención por polling.
- Ninguna de las anteriores**

## Ejercicio 11

El *flag* de buffer de transmisión:

- El *flag* TXE de Transmisión de la USART siempre estará activo al tener el buffer vacío
- El *flag* TXE sólo se desactiva cuando está transmitiendo.
- Si se activa la interrupción por transmisión, no se ejecutará la RAI hasta que el buffer esté lleno
- Ninguna de las anteriores
- a) y b) son correctas.**

## Ejercicio 12

Cuando se recibe un carácter:

- La USART pone el *flag* RXNE = 0 indicando que el buffer está ocupado y ha llegado algo. Si la interrupción de recepción está habilitada, se ejecutará la RAI
- La USART pone el *flag* RXNE = 1 indicando que el buffer está ocupado y ha llegado algo. Si la interrupción de recepción está habilitada, se ejecutará la RAI
- El dato recibido estará en DR. Esto provoca que RXNE=0 de forma automática. Si no se lee el dato antes de que llegue el siguiente carácter, se detectará un error de *overrun*.
- El dato recibido estará en CR. El *flag* RXNE se debe forzar a 0 por software.
- a) y d) son correctos
- b) y c) son correctas**