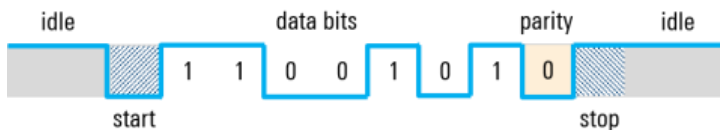


3) Que es una transmisión serie o UART. Que significan las siguientes propiedades: Baud Rate, Data Bits, Parity, Stop Bits, Send XON/XOFF, terminal type

¿Qué es UART?

UART (**universal asynchronous receiver / transmitter**, por sus siglas en inglés), define un protocolo o un conjunto de normas para el intercambio de datos en serie entre dos dispositivos. UART es sumamente simple y utiliza solo dos hilos entre el transmisor y el receptor para transmitir y recibir en ambas direcciones. Ambos extremos tienen una conexión a masa. La comunicación en UART puede ser **simplex** (los datos se envían en una sola dirección), **semidúplex** (cada extremo se comunica, pero solo uno al mismo tiempo), o **dúplex completo** (ambos extremos pueden transmitir simultáneamente). En UART, los datos se transmiten en forma de tramas.



¿Dónde se utiliza UART?

UART fue uno de los primeros protocolos en serie. Los puertos en serie, que en su día proliferaron a gran escala, se basan casi siempre en el protocolo UART, y los dispositivos que utilizan interfaces RS-232, módems externos, etc. son ejemplos típicos de la aplicación de UART. En los últimos años la popularidad de UART ha disminuido, y se ha ido sustituyendo por protocolos como SPI e I2C para la comunicación entre chips y componentes. En lugar de comunicarse por un puerto en serie, la mayoría de los ordenadores y periféricos modernos utilizan hoy tecnologías como Ethernet y USB. Sin embargo, UART se sigue utilizando para

aplicaciones de baja velocidad y bajo rendimiento, puesto que es muy simple, económico y fácil de integrar.

Temporización y sincronización de protocolos UART

Una de las mayores ventajas de UART es que es asíncrono: el transmisor y el receptor no comparten la misma señal de reloj. Si bien esto simplifica en gran medida el protocolo, plantea determinados requisitos en el transmisor y el receptor. Puesto que no comparten un reloj, ambos extremos deben transmitir a la misma velocidad, previamente concertada, con el fin de mantener la misma temporización de los bits. Las velocidades en baudios más habituales en UART que se utilizan actualmente son 4800, 9600, 19,2 K, 57,6 K, y 115,2 K. Además de tener la misma velocidad en baudios, ambos extremos de una conexión UART deben utilizar también la misma estructura y parámetros de trama. La forma más sencilla de entender esto es observando una trama UART.

Formato de trama en UART

Como en la mayoría de los sistemas digitales, un nivel de tensión «alto» se utiliza para indicar un «1» lógico, y un nivel de tensión «bajo» se emplea para indicar un «0» lógico. Dado que el protocolo UART no define tensiones específicas o rangos de tensión para estos niveles, a veces se denomina al nivel alto «marca» y al bajo «espacio». Obsérvese que en el estado de reposo (cuando no se transmiten datos) la línea se mantiene en el estado alto. Esto permite detectar con facilidad una línea o un transmisor averiado.

Bits de inicio y de parada

Puesto que UART es asíncrono, el transmisor necesita señalar que los bits de datos están llegando. Este se realiza con el bit de inicio. El bit de inicio es una transición del estado de reposo alto a un estado bajo, seguido inmediatamente de bits de carga útil (de datos). Una vez que finalizan los bits de datos, el bit de parada indica el fin de la carga útil. El bit de parada es o bien una transición de retorno al estado alto o de reposo, o bien la permanencia en el estado alto por un tiempo de bit adicional. Se puede configurar un segundo bit de parada (opcional), normalmente para dar al receptor tiempo para prepararse para la siguiente trama, si bien no es una práctica muy común.

Bits de datos (Data Bits)

Los bits de datos son la carga útil o los bits «útiles», y llegan inmediatamente después del bit de inicio. Puede haber de 5 a 9 bits de carga útil, si bien lo más habitual son 7 u 8 bits. Estos bits de datos se transmiten generalmente con el bit menos significativo primero.

Por ejemplo:
Si deseamos enviar la letra mayúscula «S» en ASCII de 7 bits, la secuencia de bits es 1 0 1 0 0 1 1. Primero invertimos el orden de los bits para colocarlos en el orden del bit menos significativo, es decir, 1 1 0 0 1 0 1, antes de enviarlos. Una vez que se han enviado los últimos bits de datos, el bit de parada se utiliza para finalizar la trama y la línea retrocede al estado de reposo.

- «S» en ASCII con 7 bits (0x52) = 1 0 1 0 0 1 1
- Orden de bit menos significativo = 1 1 0 0 1 0 1



Bits de inicio y de parada



Bits de datos

Bit de paridad (Parity)

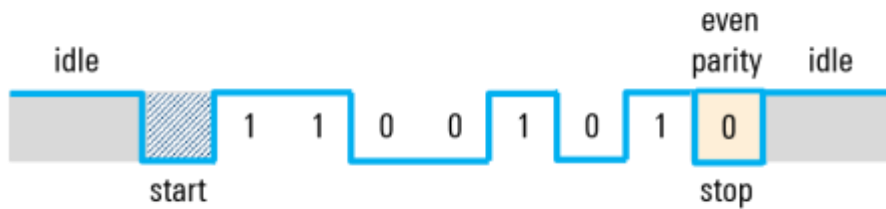
Una trama UART también puede contener un bit de paridad opcional que se utiliza para la detección de errores. Este bit se inserta entre los bits de fin de datos y el bit de parada. El valor del bit de paridad depende del tipo de paridad utilizado (par o impar):

- En la **paridad par**, este bit se ajusta de tal modo que el número total de unos en la trama será par.
- En la **paridad impar**, este bit se ajusta de tal modo que el número total de unos en la trama será impar.
-

Ejemplo:

La letra «S» mayúscula (1 0 1 0 0 1 1) contiene en total tres ceros y cuatro unos. Si se utiliza la paridad par, el bit de paridad es cero, puesto que ya existe un número par de unos. Si se utiliza la paridad impar, entonces el bit de paridad deberá ser uno para que la trama tenga un número impar de unos.

El bit de paridad solo puede detectar un único bit invertido. Si hay más de un bit invertido, no es posible detectarlo con fiabilidad utilizando un único bit de paridad.



Ejemplo de bit de paridad

TASA DE BAUDIOS (BAUD RATE)

El baudio es una unidad de medida utilizada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital. Cada símbolo puede codificar 1 o más bits dependiendo del esquema de modulación, un bit siempre representa dos estados, por lo tanto baudios por segundo no siempre es equivalente a bits por segundo, los símbolos son las unidades de información estas se representan en bits, de manera que la tasa de bits será igual a la tasa de baudios solo cuando halla 1 bit por símbolo.

STOP BITS

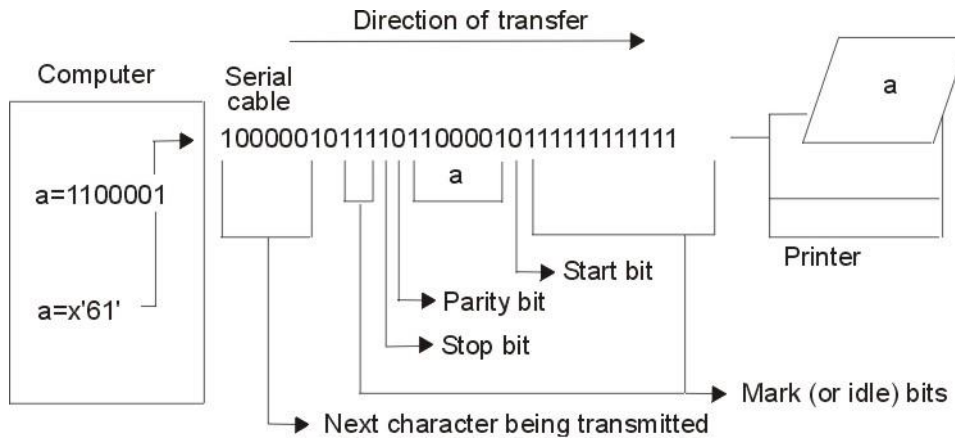
Los bits de inicio y parada se utilizan en la comunicación asíncrona como un medio de temporización o sincronización de los caracteres de datos que se transmiten.

Sin el uso de estos bits, los sistemas de envío y recepción no sabrán dónde termina un carácter y comienza otro.

Otro bit utilizado para separar los caracteres de datos durante la transmisión es el bit RS de marca (o inactivo). Este bit, un 1 binario, se transmite cuando la línea de comunicación está inactiva y no se envían ni reciben caracteres.

Cuando el sistema recibe un bit de inicio (0 binario), se entiende que un bit de carácter de número fijo (determinado por el parámetro de **bits por carácter**), e incluso un bit de paridad (determinado por el parámetro de **paridad**), sigue al bit de inicio. . Luego, el sistema recibe un bit de parada (1 binario). En el siguiente ejemplo, el bit de **paridad** está presente y los **bits por carácter** son 7.

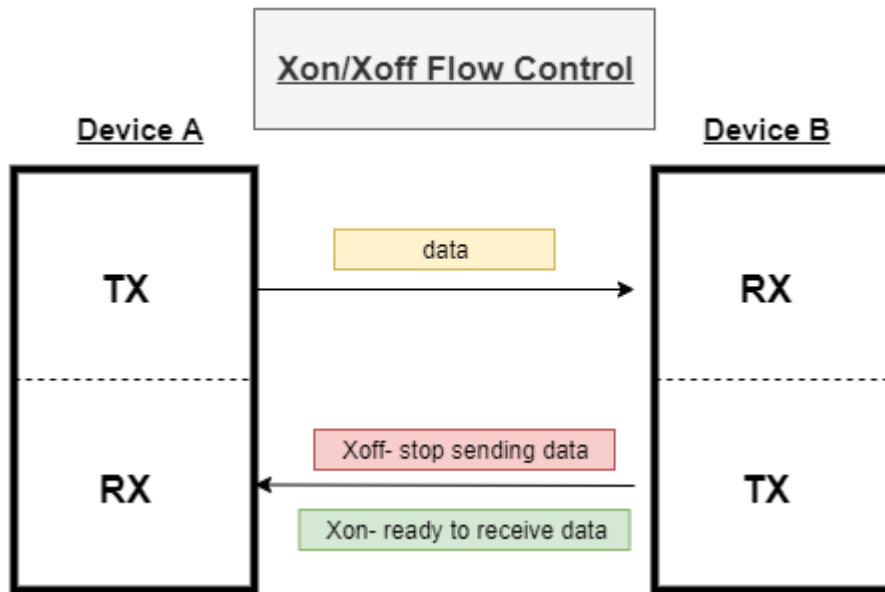
Figura 1. Bits de inicio, parada y marca



SEND XON/XOFF

Xon-Xoff es un método de control de flujo utilizado en la comunicación entre dispositivos. Se distingue del control de flujo de hardware, que se realiza mediante señales específicas fuera de banda, como [DTR/DSR](#) o [RTS/CTS](#) en el protocolo estándar RS232.

Por ejemplo, tiene dos dispositivos, *A* y *B* , y *A* es más rápido para enviar datos que el dispositivo *B* para recibirlos y procesarlos. El dispositivo *B* llegará muy rápidamente a un punto en el que ya no podrá procesar más datos y se verá abrumado por la cantidad de datos que envía el dispositivo *A*. En este punto, el dispositivo *B* necesitaría enviar un carácter *Xoff* al dispositivo *A* para que deje de enviar datos. No enviará más datos al dispositivo *B* hasta que el dispositivo *B* haya enviado un carácter *Xon* al dispositivo *A*, lo que permite que el dispositivo *A* sepa que el dispositivo *B* está listo para recibir más datos.



Xon-Xoff utiliza códigos especiales acordados tanto por el transmisor como por el receptor.

La siguiente tabla es un ejemplo de algunas de las formas de expresar los códigos Xon-Xoff enviados en el flujo de datos, aunque se pueden establecer otros valores.

Código	Sentido	ASCII	DIC	MALEFICIO	Teclado
XAPAGADO	Pausar transmisión	DC3	19	13	Control + S
XON	Reanudar transmisión	DC1	17	11	Control + Q

Ventajas de Xon/Xoff

- La principal ventaja del control de flujo Xon-Xoff es que solo requiere la cantidad mínima de cableado de señal, **tres**. Esto se debe a que el control de flujo Xon-Xoff solo requiere dos señales (enviar, recibir)

y, por supuesto, un solo cable de tierra común. El control de flujo de hardware requiere al menos dos cables adicionales entre los dos dispositivos, como en [RTS/CTS](#) y [DTR/DSR](#) . El costo de esto fue significativo en los primeros días de la informática.

Desventajas de Xon/Xoff

- El envío de caracteres Xoff/Xon ocupa ancho de banda porque son señales dentro de la banda. Debido a esto, no pueden confundirse con datos, sino que deben reconocerse como comandos de control de flujo. Esto significa que se necesita codificación para la transmisión de los caracteres *Xon* y *Xoff* para que el receptor no confunda los comandos con comandos de control del dispositivo. Esto generalmente se hace mediante algún tipo de caracteres de escape, pero esto significa que estos comandos ocupan más ancho de banda. Por otro lado, las señales de hardware como RTS/CTS pueden afirmarse muy rápidamente como señales fuera de banda, ocupando menos ancho de banda.

TERMINAL TYPE

USB se ha convertido en el protocolo de comunicación en serie preferido entre productos digitales como computadoras y dispositivos periféricos, ya que puede transmitir datos a través de cables más largos y ofrece velocidades de transmisión más altas. Sin embargo, UART todavía se usa hoy en día para ciertas aplicaciones y, a menudo, se ve en dispositivos más antiguos.

Para cerrar la brecha entre los dispositivos más antiguos y los más nuevos, los desarrolladores han mantenido viables las rutas de transmisión heredadas entre los sistemas USB y UART. Un convertidor de USB a UART es un circuito integrado que se utiliza para enviar o recibir datos en serie desde un puerto USB a datos en serie que pueden recibirse o enviarse mediante una interfaz UART. Este es un dispositivo pequeño que se conecta a su puerto USB y tiene al menos salidas de tierra, Rx y Tx. Actúa como un puerto serie para su computadora y la computadora envía datos a este puerto donde el módulo los convierte en señales UART.

