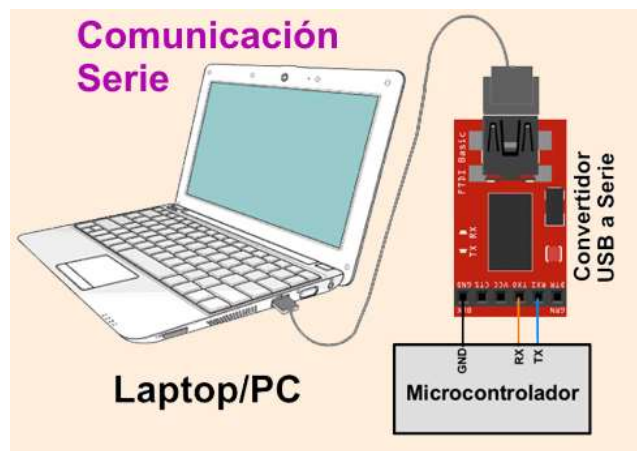


¿Qué es la comunicación de datos en serie?

La comunicación serial es un protocolo estandarizado que permite el intercambio de información en forma de bits entre dos o más dispositivos, enviando los datos un bit a la vez a través de un único canal de comunicación.

En contraste con la comunicación en paralelo, que envía varios bits a la vez utilizando múltiples canales, la comunicación en serie es más simple y económica, ya que requiere menos cables y componentes.



Característica de los protocolos serie

Aunque los detalles de implementación difieren entre protocolos, todos los protocolos serie tienen cuatro características básicas, las cuales también son importantes para el análisis y decodificación de los datos en serie:

- **Niveles:** cómo se utilizan los voltajes para representar los ceros y unos
- **Tiempo:** con qué frecuencia se envían los bits (tiempo de bit)
- **Trama:** cómo los bits se organizan en grupos y el rol de cada bit o grupo de bits
- **Protocolo:** qué mensajes se intercambian y bajo qué circunstancias

Niveles

Antes de realizar cualquier clase de decodificación digital, necesitamos poder hacer una distinción entre ceros y unos. En otras palabras, ¿cómo se determinan los valores de bit a partir de los niveles de voltaje? Un enfoque simple sería definir que un voltaje «bajo» es igual a cero y un voltaje «alto» es igual a uno. De hecho, así es como algunos protocolos en serie funcionan.

Los protocolos en serie que se usan en entornos desafiantes, como en automóviles, a menudo utilizan un voltaje diferencial, porque las señales diferenciales tienden a ser más inmunes al ruido.

Diferencial significa que un 0 o un 1 se define en base de la diferencia entre dos voltajes, y no con respecto a tierra.

Tiempo

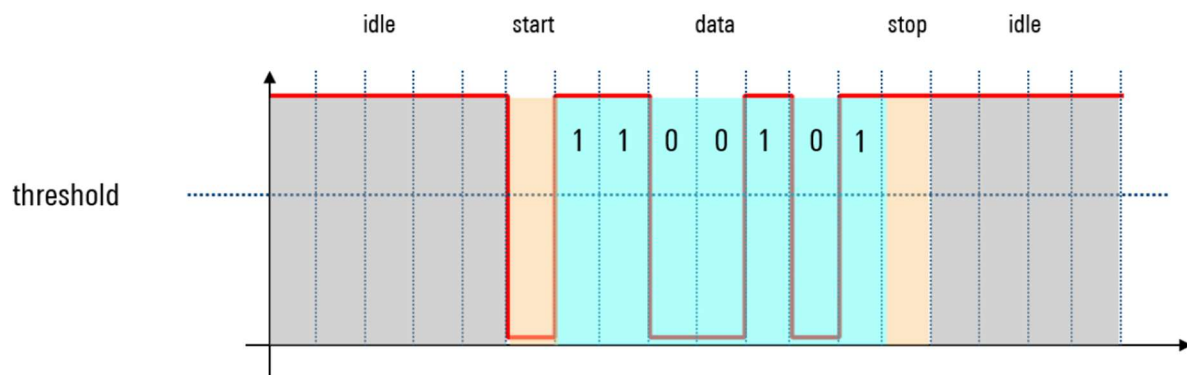
Definir los niveles de voltaje para «0» y «1» no es suficiente para determinar que bits se están recibiendo. Adicionalmente, es importante entender qué tan rápido se generan los bits o, dicho de otra manera, cuál es el «tiempo de bit» o «velocidad de bit».

Para decodificar datos en serie, el receptor o instrumento debe utilizar la misma velocidad de bits que el transmisor.

Trama

Los protocolos en serie normalmente organizan los bits en las llamadas tramas. Los bits individuales o los grupos de bits en la trama tienen significados definidos. Para decodificar correctamente las tramas, es necesario conocer un poco sobre esta estructura.

Por ejemplo, al decodificar el protocolo en serie UART es importante saber que el estado de reposo de la línea es un nivel de alto voltaje. Una transición de alto a bajo es un «bit de inicio» que indica el inicio de la trama. Además, es necesario entender cuántos bits de datos leer, como también, el hecho de que hay un bit de parada de nivel de alto voltaje que termina la trama, seguido de un retorno al estado de reposo. Al conocer la estructura de la trama, los datos del usuario pueden extraerse de la secuencia de bits en serie, así como extraer otra información sobre la transmisión.



Protocolo

El último aspecto a considerar puede llamarse «protocolo». Una definición genérica de protocolo es un conjunto de reglas para cifrar e intercambiar información. Puede haber reglas sobre cómo y cuándo los datos se envían, además de los tipos y significados de los mensajes que los puntos de conexión intercambian.

Un protocolo sencillo sería enviar los datos apenas estén disponibles, independientemente de si el receptor esté listo o no para recibirlos. Un protocolo más sofisticado sería utilizar algún tipo de mecanismo que pida permiso antes de enviar los datos. Un protocolo aún más sofisticado supondría que el remitente espere un acuse de recibo explícito de que los datos fueron correctamente recibidos antes de enviar más datos o datos que faltan o con errores.

Protocolos de comunicación en serie

Los protocolos serie se utilizan en una amplia variedad de aplicaciones. Los tres principales protocolos serie que se utilizan para aplicaciones genéricas son UART, I²C y SPI..

Una categoría especial de protocolos serie son aquellos que se utilizan en la industria automotriz, como CAN, Lly FlexRay. Estos protocolos se diseñaron para lidiar con los exigentes requisitos de los vehículos como proporcionarles un funcionamiento eficiente a una combinación de velocidades bajas y altas y en un entorno con mucho ruido.

Diferencias entre I2C, UART y SPI

Cualquier microcontrolador que vaya de los 8 a los de 32 bits utilizará al menos uno de estos protocolos, junto con los pines GPIO, para su programación y envío de señales a periféricos sencillos. Estos tres protocolos serie tienen en común que son protocolos de bus. Mientras que I2C y UART utilizan esquemas de direccionamiento, el SPI funciona sin direccionamiento. El direccionamiento del UART es según la dirección base, por ejemplo, en el puerto COM1, la dirección 0 es usualmente 3F8h y la dirección 7 es 3FFh.

Aunque SPI funcione sin direccionamiento, no deja de ser un protocolo de bus y sigue pudiendo utilizarse para seleccionar los dispositivos de bajada que van a recibir los datos.

UART

El protocolo UART, viene de las siglas en inglés 'Universal Asynchronous Receiver-Transmitter' (Receptor-transmisor asíncrono universal), mientras que USART son las siglas de 'Universal Synchronous and Asynchronous Receiver-Transmitter'. La diferencia entre ellos es obvia, UART realiza solo comunicación en serie asíncrona, mientras que USART puede realizar procesos de comunicación en serie tanto síncronos como asíncronos.

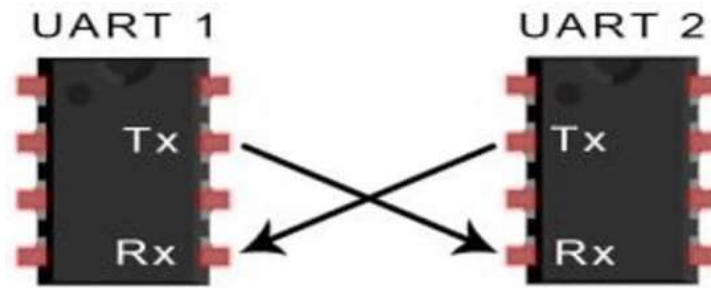
Las principales características son:

Tiene dos líneas de datos, una para transmitir (TX) y otra para recibir (RX), que se utilizan para comunicarse a través del pin digital 0, pin digital 1.

TX y RX siempre están conectados entre dos dispositivos. (por ejemplo, USB y un ordenador)

Admite la transmisión de datos bidireccional, asíncrona y en serie.

También puede manejar problemas de administración de sincronización entre ordenadores y dispositivos seriales externos.



I2C

El protocolo I2C, significa circuito inter-integrado, de las siglas inglesas 'Inter-Integrated Circuit'. Se suele pronunciar I-cuadrado C o IIC por sus siglas. Fue desarrollado por Philips Semiconductor para televisiones en los años 1980's.

Este protocolo utiliza dos líneas para controlar otros dispositivos. La primera línea es el reloj llamado SCL (línea de reloj en serie), mientras que la otra es para datos llamado SDA (puerto de aceptación de línea de datos). Este protocolo es similar al UART, sin embargo, no se utiliza para la comunicación PC-dispositivo, sino con módulos y sensores, siendo el PC el "maestro" y el resto los llamados "esclavos". Estos curiosos términos podemos definirlos como:

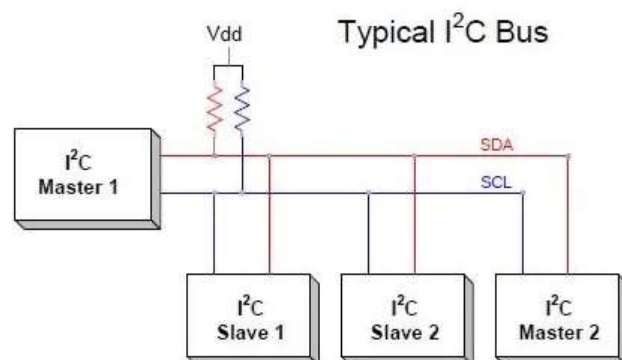
- Maestro: dispositivo que proporciona un reloj para la comunicación
- Esclavo: dispositivo que no es el maestro que utiliza el reloj del maestro para comunicar

A su vez, tiene también tres modos según su velocidad:

- Baja (< 100 Kbps)
- Media (400 Kbps)
- Alta (3.4 Mbps) – I2C v.2.0

I2C es un bus serie síncrono de dos cables (SDA y SCL) bidireccional simple y solo requiere dos cables para transmitir información entre los dispositivos conectados al bus. Es por tanto de tipo half-duplex.

En cuanto a la velocidad, esta depende de la velocidad de los datos, la calidad del cable y el ruido externo.

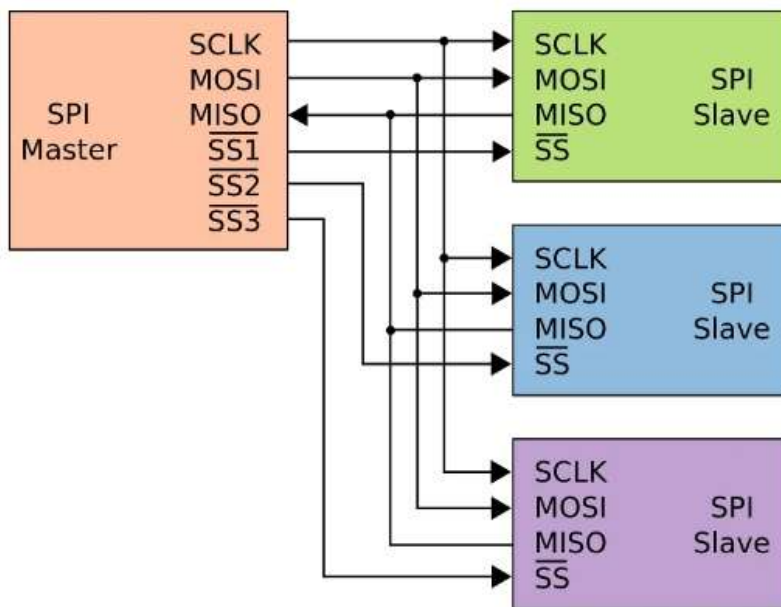


SPI

Las siglas SPI vienen del inglés “Serial Peripheral Interface” o interfaz de periféricos en serie, siendo un protocolo similar al I2C pero especialmente pensado para microcontroladores. También es usado en situaciones donde la velocidad es importante como, por ejemplo, tarjetas SD, módulos de visualización o cuando la información se actualiza y cambia rápidamente como termómetros.

Este protocolo de comunicación serial es de tipo serie de tipo síncrono y consta de dos líneas de datos (MOSI y MISO), una línea de reloj (SCK) y una línea de selección esclava (SS). Estas líneas tienen las siguientes funciones:

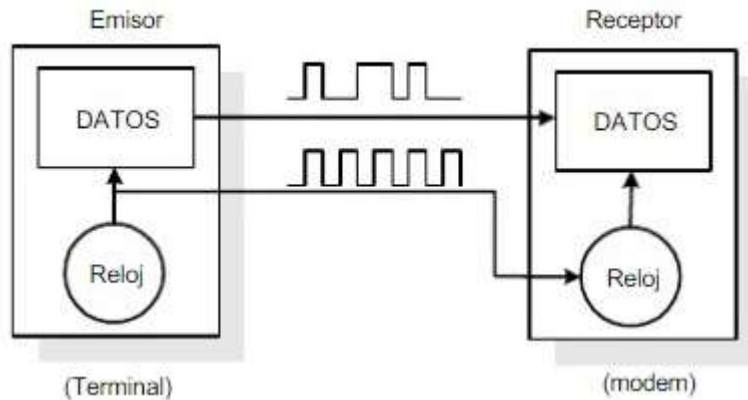
- MOSI: Master Output Slave In - Salida del maestro entrada del esclavo (línea a través de la cual el maestro envía datos a sus esclavos)
- MISO - Master In Slave Out – Entrada del maestro salida del esclavo (línea a través de la cual los esclavos responde al maestro)
- SCK - Serial Clock (reloj proporcionado por el dispositivo maestro)
- SS - Slave Select (línea utilizada para seleccionar el esclavo con el que el maestro desea comunicarse)



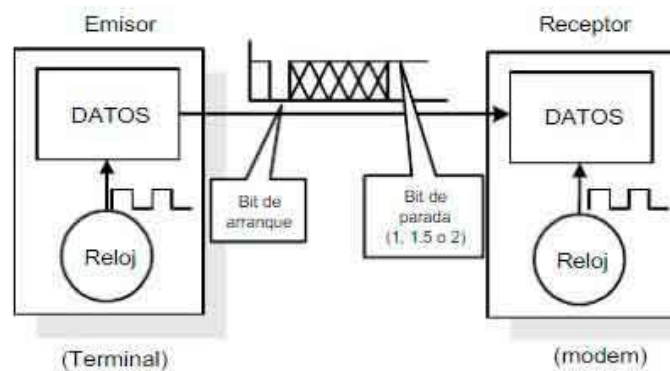
Tipos de comunicación en serie:

Existen dos tipos principales de comunicación en serie:

- **Síncrona:** La transmisión de datos se sincroniza con una señal de reloj externa. Esto garantiza una alta precisión y confiabilidad, pero requiere un cable adicional para la señal de reloj.



- **Asíncrona:** No utiliza una señal de reloj externa, por lo que es más simple y económica. Sin embargo, es menos precisa y puede ser más susceptible al ruido.



Aplicaciones:

La comunicación en serie se utiliza en una amplia variedad de aplicaciones, incluyendo:

- Conexión de periféricos a computadoras: ratones, teclados, impresoras, módems, etc.
- Comunicación industrial: controladores lógicos programables (PLC), sensores, actuadores, etc.
- Comunicación en redes: routers, módems, etc.
- Dispositivos médicos: marcapasos, monitores de signos vitales, etc.

Ventajas:

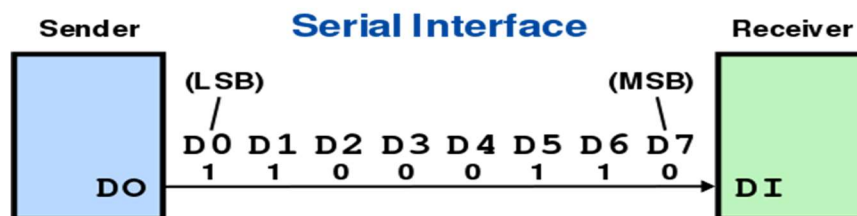
- **Simplicidad:** Requiere menos cables y componentes que la comunicación en paralelo.
- **Bajo costo:** Es una solución más económica para la transmisión de datos.
- **Robustez:** Es menos susceptible al ruido y a las interferencias.
- **Versatilidad:** Se puede utilizar para conectar una amplia variedad de dispositivos.

Desventajas:

- **Velocidad:** La velocidad de transmisión es generalmente más lenta que la comunicación en paralelo.
- **Distancia:** La distancia máxima de transmisión es generalmente más corta que la comunicación en paralelo.
- **Complejidad de los protocolos:** Los protocolos de comunicación en serie pueden ser más complejos que los protocolos en paralelo.

Funcionamiento:

Transmisión bit a bit: La información se divide en bits individuales, que se envían uno tras otro a través del canal de comunicación. Cada bit se representa por un cambio en el voltaje o la corriente eléctrica.

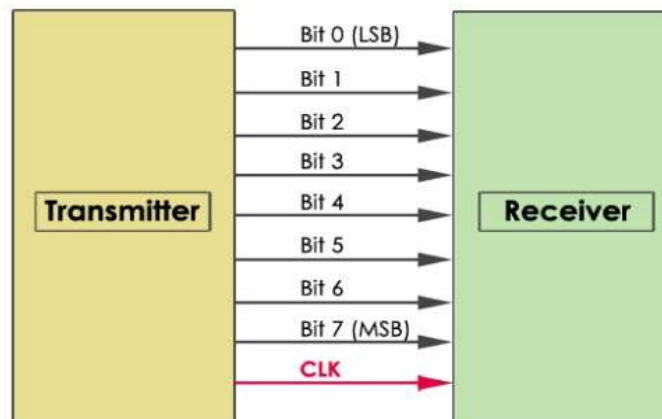


Dispositivos involucrados

La comunicación en serie involucra dos dispositivos principales:

Dispositivo transmisor: Encarga de convertir los datos a una señal eléctrica adecuada para la transmisión en serie.

Dispositivo receptor: Recibe la señal eléctrica, la interpreta y la convierte de nuevo en datos utilizables.



Transmisión de datos

En la comunicación en serie, los datos se transmiten secuencialmente, comenzando por el bit menos significativo (LSB) o el más significativo (MSB), dependiendo del protocolo utilizado.

Sincronización

Para garantizar que el receptor pueda interpretar correctamente los datos, tanto el transmisor como el receptor deben estar sincronizados en cuanto a la velocidad de transmisión. Esto se logra utilizando algún tipo de reloj o mediante el envío de señales de sincronización junto con los datos.

Inicio y fin de la transmisión

En muchos protocolos de comunicación en serie, se utilizan bits especiales para indicar el inicio y el fin de la transmisión de datos. Por ejemplo, el protocolo UART utiliza bits de inicio y de parada para delimitar cada byte de datos.

Detección de errores

Para garantizar la integridad de los datos transmitidos, se pueden utilizar técnicas de detección y corrección de errores. Esto puede implicar el uso de bits de paridad, checksums o algoritmos más avanzados como CRC (Cyclic Redundancy Check).

Velocidad de transmisión (baud rate)

La velocidad de transmisión se mide en baudios y determina la cantidad de bits que se pueden transmitir por segundo. Es importante que tanto el transmisor como el receptor estén configurados con la misma velocidad de transmisión para garantizar una comunicación adecuada.