

Diferencias en los protocolos de comunicación

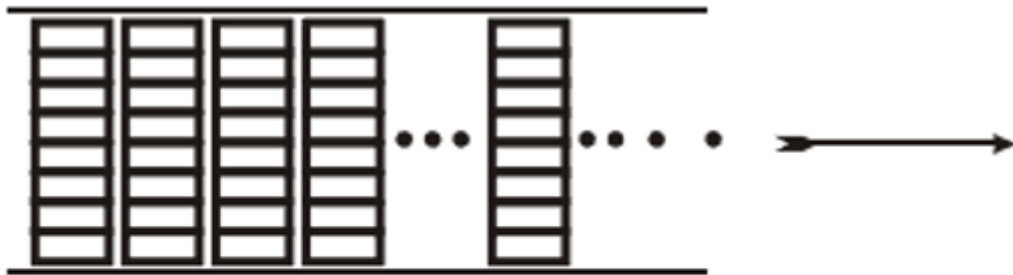
SPI VS I2C

Comunicación en serie

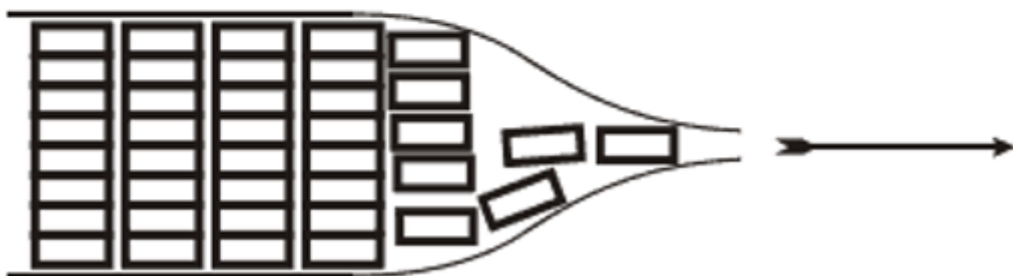
La comunicación en serie, serial o secuencial es la metodología más utilizada en lo que respecta a los sistemas integrados. Pero antes de hablar sobre los tipos de protocolos de comunicación en serie utilizados en la industria integrada y compararlos, vamos a ver que es una comunicación en serie primero. La principal ventaja de la comunicación en serie es que necesitamos un número más pequeño de líneas de transmisión que una comunicación paralela para transmitir la misma información. Además, elimina algunos problemas de transmisión o sincronización que presenta la comunicación en paralelo, por lo que suele ser la tecnología más usada actualmente.

Sin embargo, la comunicación en serie tiene un menor rendimiento a la misma frecuencia, es por ello por lo que para compensarse se trabaja a frecuencias más altas.

Transferencia en paralelo



Transferencia en serie



Un ejemplo muy claro de comunicación serial es el código Morse que todos conocemos, donde se envía la información secuencialmente dato a dato



'SOS' en código morse

I2C

El protocolo I2C, significa circuito inter-intergrado, 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.

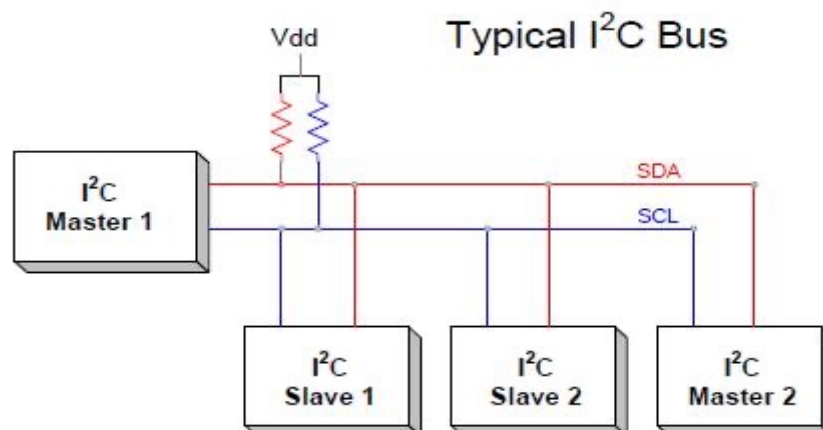
Este protocolo utiliza dos líneas para controlar otros dispositivos. La primera línea es el reloj llamado SCL (línea de reloj en serie), mientras que la otra es para datos llamado SDA (puerto de aceptación de línea de datos). Este protocolo es similar al UART, sin embargo, no se utiliza para la comunicación 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 (< 100Kbps)
- 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 datos, la calidad de cable y el ruido externo

Funcionamiento

Como hemos dicho, mediante los dos cables SDA y SCL se establece la comunicación del maestro con los esclavos, El primero inicia la transferencia de datos por el bus y genera un reloj para abrir el dispositivo transferido y, cualquier dispositivo direccionado se considera un dispositivo esclavo

La relación entre los dispositivos maestro y esclavo, transmitiendo y recibiendo en el bus no es constante.

Depende de la dirección de transferencia de datos en ese momento

Por lo tanto, cuando es el maestro el que desea enviar datos al esclavo, primero deberá dirigirse al esclavo o esclavos antes de enviar cualquier dato

Entonces, el maestro completará la transferencia de datos. Si el maestro desea recibir datos del esclavo, es el maestro el que debe volver a dirigirse al esclavo primero. Luego, el maestro (Host) recibe los datos enviados por el esclavo y finalmente, el receptor finaliza el proceso de recepción.

Por otra parte, es necesario conectar la fuente de alimentación a través de la resistencia pull-up. Cuando el bus está inactivo, ambas líneas funcionan a un nivel de potencia alto.

La capacitancia en la línea afectará la velocidad de transmisión del bus. Como la potencia actual en el bus es pequeña, cuando la capacitancia es demasiado grande, puede causar errores de transmisión. Por lo tanto, su capacidad de carga debe ser de 400pF, por lo que se puede estimar la lo

Protocolo

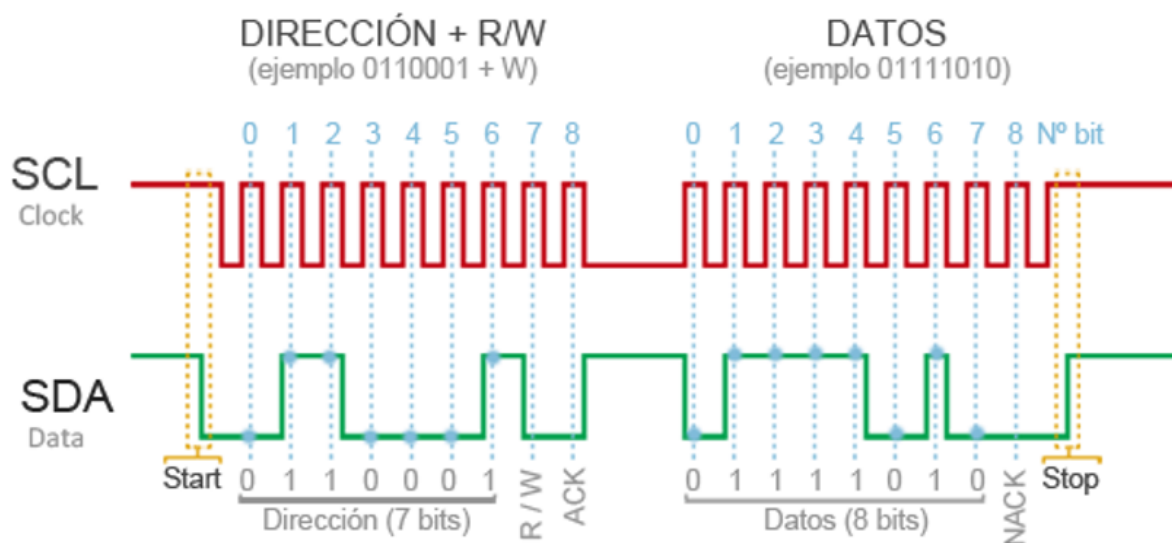
El funcionamiento es un poco más complejo que UART, pero al final es muy parecido. El maestro envía a cada esclavo la dirección de 7 o 10 bits del esclavo y un bit de lectura/escritura al esclavo con el que desea comunicarse.

El esclavo comparará la dirección con la suya. Si la dirección coincide, el esclavo devolverá un bit ACK que cambia la línea SDA a baja por un bit. Si la dirección no coincide con su dirección, el esclavo deja la línea SDA alta.

A continuación, el maestro enviará o recibirá la trama de datos. Al acabar esta acción, la transferencia de datos, el dispositivo receptor devuelve otro bit ACK al remitente para reconocer la transmisión exitosa.

Por otro lado, para detener la transmisión de datos, el maestro envía una señal de parada al esclavo cambiando SCL alto antes de cambiar SDA alto.

Todo esto se puede apreciar un poco mejor en el siguiente diagrama, donde vemos el inicio de la transmisión, el envío de la dirección junto con el bit de escritura o lectura (R/W) y por último el envío de los datos y la señal de parada (stop).



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)

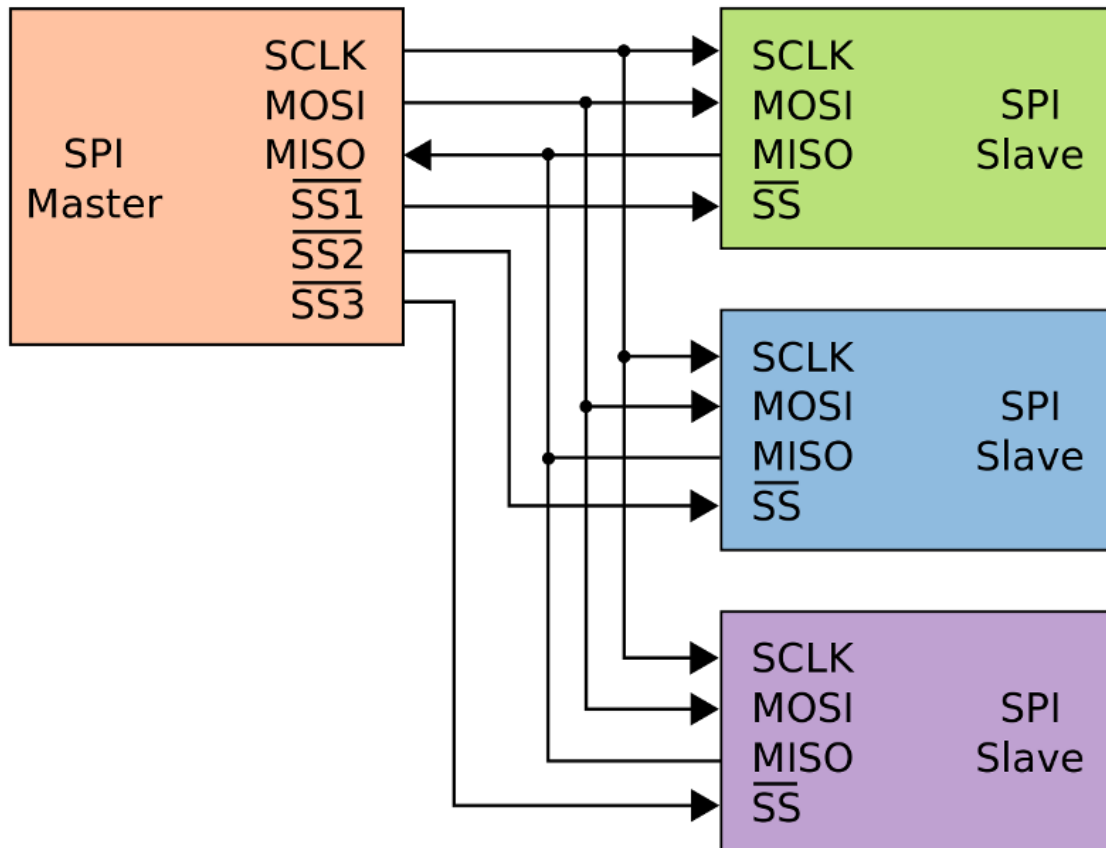
Funcionamiento

Por lo general es un protocolo más rápido que el I2C debido a que es más simple.

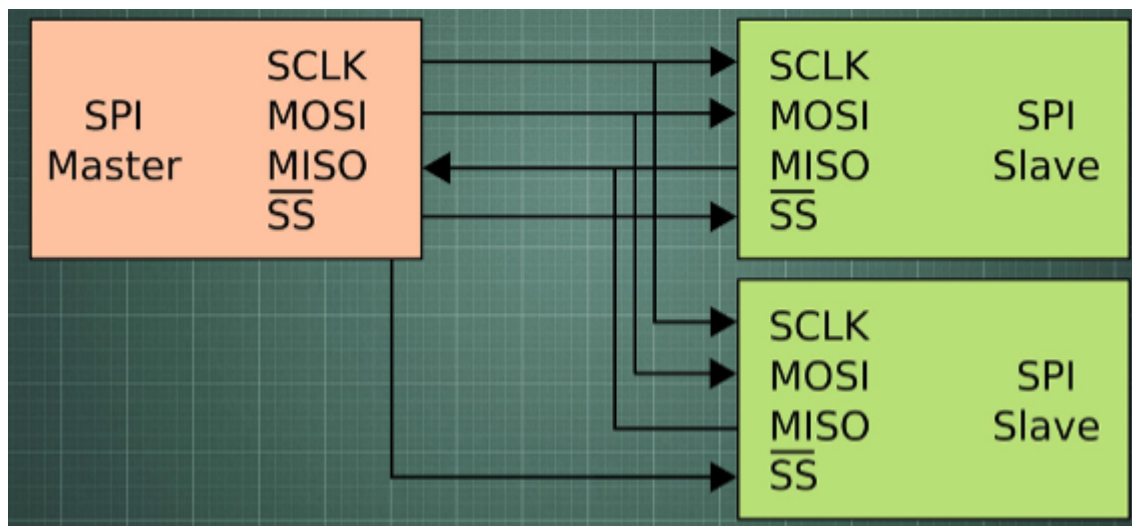
Existen dos modos de comunicarse con este protocolo:

- En un primer modo, si se utiliza un único dispositivo controlador para activar un único dispositivo de bajada, se trata simplemente de una topología punto a punto. Seleccionar cada dispositivo con una línea Chip Select, por lo que se necesita una línea de selección de chip separada para cada dispositivo. Esta es la forma más común en que los Raspberry Pi utilizan actualmente SPI, por ejemplo. La activación de varios dispositivos depende del número de salidas de selección de chip que proporcione el controlador (modo estándar).
- El segundo modo utiliza la conexión en cadena margarita, en la que una única salida de selección de dispositivo activa sucesivamente cada uno de los dispositivos de la propia cadena margarita.

Por lo tanto, no hay límite para la cantidad de dispositivos SPI que se pueden conectar. Sin embargo, existen límites prácticos debido al número de líneas de selección de hardware disponibles en el dispositivo principal (el máster) con el método de selección de chip o la complejidad de pasar datos a través de dispositivos en el método de conexión en cadena.



Conexión en modo estándar



Conexión modo margarita

Protocolo

Este protocolo precisa un poco más de complejidad hardware que el sistema I2C cuando se requiere de múltiples esclavos, ya que cada esclavo requiere una señal de habilitación separada.

La interfaz SPI son en realidad dos registros de desplazamiento simples en el hardware interno. Los datos transmitidos son de 8 bits y se transmiten bit a bit bajo la señal de habilitación del esclavo y el pulso de cambio generado por el dispositivo maestro. El bit alto está en el frente y el bit bajo está en la parte posterior.

La interfaz SPI es una transmisión de datos en serie sincrónica entre la CPU y el dispositivo periférico de baja velocidad. Bajo el pulso de cambio del dispositivo maestro, los datos se transmiten bit a bit. El bit alto está en el frente y el bit bajo está en la parte posterior.

Es una comunicación full-duplex y la velocidad de transmisión de datos es en general más rápida que la del bus I2C ya que puede alcanzar velocidades de unos pocos Mbps.

