

¿Qué es el SPI?

La SPI fue desarrollada por Motorola (ahora parte de NXP Semiconductors) aproximadamente en 1985. Se trata de una interfaz serial síncrona prevista para la comunicación entre dispositivos a corta distancia. Desde entonces, se ha convertido en un estándar de-facto empleado por muchos fabricantes de semiconductores, especialmente en microprocesadores y microcontroladores.

El motivo de la popularidad de SPI radica en sus muchas ventajas. La primera es que es una interfaz direccionada de hardware simple que ofrece completa flexibilidad para la cantidad de bits transferidos. Usa un modelo de maestro-secundario con un maestro simple y puede manejar varios dispositivos secundarios usando comunicaciones dúplex que operan a velocidades de reloj de hasta 50 MHz. No usa un protocolo estándar y transfiere solo paquetes de datos, lo que la hace ideal para transferir flujos de datos largos.

SPI usa un máximo de cuatro líneas de señal (Figura 1). El dispositivo maestro, por lo general un procesador o controlador, suministra y controla el reloj (SCK) y líneas de selección de chip (CS). La operación multiplexor completa se maneja a través de las líneas de datos Master Out Slave In (MOSI) y Master In Slave Out (MISO). En un maestro individual simple, con configuración del dispositivo secundario individual, la línea de selección de chip puede eliminarse y se puede forzar la entrada de CS al dispositivo secundario al estado lógico habilitado. Si el dispositivo secundario solo puede enviar datos (comunicación semidúplex), luego la línea MOSI también puede eliminarse, y así reducir el conteo de señales adicionalmente. Los datos salen a través de la señal del reloj de tal forma que la transferencia de datos se asemeja a un registro de turnos con un bit cambiado para cada reloj.

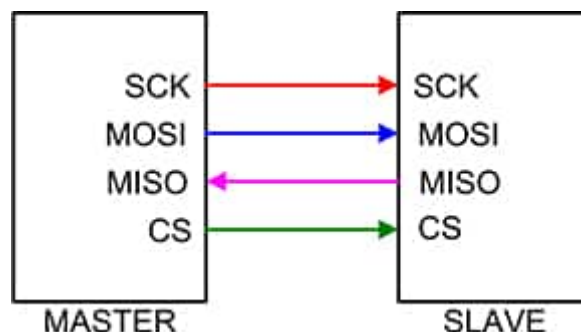


Figura 1: La conexión dúplex SPI básica usa dos líneas de datos (MOSI, MISO), una línea de reloj (SCK) y una línea de selección de chip (CS). MOSI en un dispositivo secundario es, a veces, rotulada como entrada de datos de dispositivo secundario (SDI). MISO puede

denominarse como salida de datos secundarios (SDO). (Fuente de la imagen: Digi-Key Electronics)

Hay dos enfoques para manejar varios dispositivos secundarios (Figura 2).

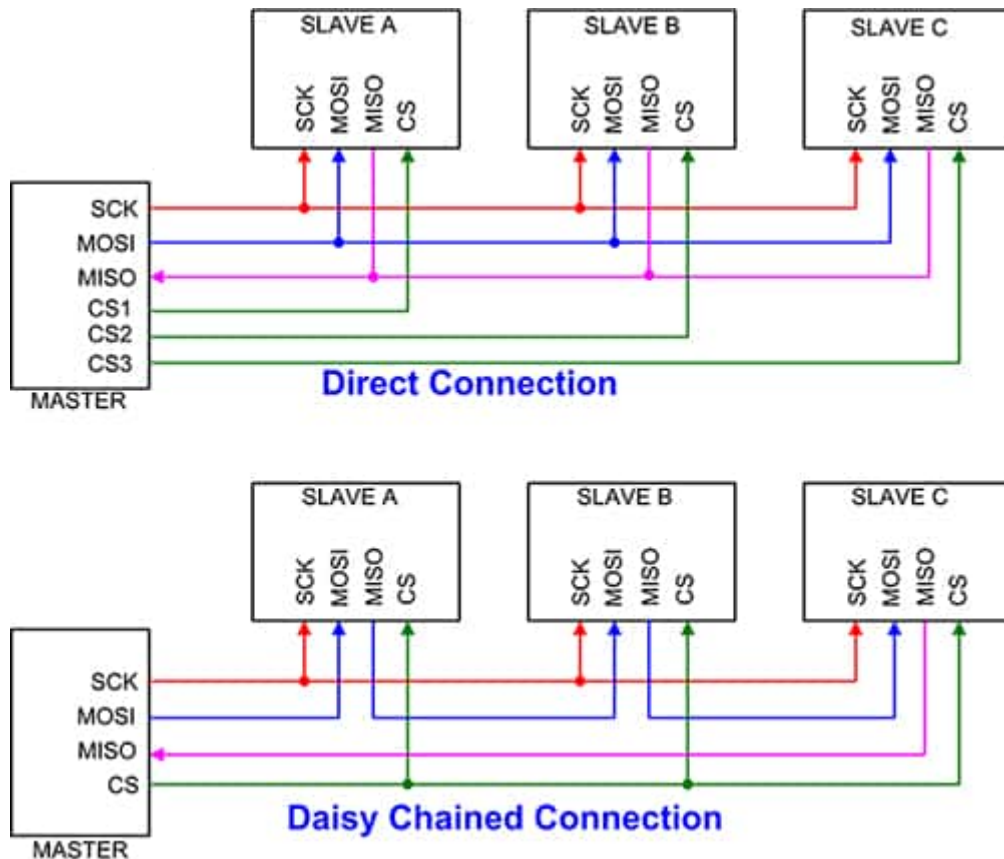


Figura 2: Dos configuraciones para tratar con interfaces de varios dispositivos secundarios. La conexión directa requiere una selección de chips para cada dispositivo secundario. La conexión encadenada usa una selección de chip individual y combina todos los datos en una sola línea. (Fuente de la imagen: Digi-Key Electronics)

La conexión directa usa una línea de selección de chips para cada dispositivo secundario. La mayoría de los microprocesadores posee tres o cuatro líneas de selección de chip. Esto limita la cantidad máxima de dispositivos secundarios al número de líneas de selección de chip. En la mayoría de los casos, esto no resulta ser un problema, pero si un diseño requiere más dispositivos en el bus, se pueden configurar algunos usando el enfoque encadenado. Con un encadenamiento, se usa una selección de chips comunes para varios dispositivos secundarios y los datos se transfieren hacia afuera en una línea de datos común. Una vez más, si se usa el modelo de los dispositivos secundarios SPI como un registro de turnos, los datos de los dispositivos secundarios se propagan en un flujo multiplexado serial.

ventajas y desventajas del SPI

Algunas de las ventajas más importantes de el bus SPI

- Tiene total flexibilidad para los bits transferidos, es decir, no se limita a una palabra de 8 bits
- Tiene una interfaz de hardware muy simple
- No se limita a ninguna velocidad de reloj máxima, lo que permite una velocidad potencialmente alta
- Es más rápido que el serial asíncrono(UART)
- Soporta múltiples esclavos.
- Soporta comunicaciones full duplex.
- Tiene una señal de bus única por dispositivo llamada SS y todas las demás señales se comparten
- No hay modos de fallo de arbitraje.
- No necesita de transceptores(adaptadores)

Por otro lado, entre las desventajas más importantes del bus SPI:

- Requiere más pines de conexión con respecto al bus I2C.
- No admite nodos en forma dinámica (en caliente)
- Solo soporta un dispositivo maestro.
- No hay protocolo de comprobación de errores.
- El bus SPI requiere generalmente de líneas SS separadas para cada esclavo, lo que puede ser problemático si se necesitan numerosos esclavos.
-

test de funcionamiento

Libreria Arduino SPI

Existen diversas formas de verificar el correcto funcionamiento de la comunicación SPI, una de ellas es usando las librerías que nos provee arduino (Para microcontroladores avanzados)

<https://docs.arduino.cc/tutorials/duemilanove/duemilanove-extended-spi>

Proyectos OpenSource.

También tenemos proyectos opensource como pueden ser Logic Sniffer, que está escrito en java, usa un puerto USB para los datos y FPGA, lo que hace es analizar una entrada digital de estados, la vuelca en la memoria RAM y luego la vuelca en la PC.

hay otras opciones, por ejemplo usando arduino

Este es el código de el programa:

Código: https://github.com/gillham/logic_analyzer

Programa : <https://www.lxtreme.nl/ols/>

Repositorio Electgpl: <https://sites.google.com/site/electgpl/storage>

Blog: <https://electgpl.blogspot.com/>

Video de el funcionamiento de el programa:

<https://www.youtube.com/watch?v=Q2t70aMPV9c>

Usando un Osciloscopio

También podemos usar un osciloscopio en este caso el MSO-X 2014A para decodificar y disparar el tráfico de BUS serial SPI de dos hilos:

Video de ejemplo: <https://www.youtube.com/watch?v=6AiAgZxPfp8>