

Protocolos de comunicación en serie

Los protocolos de comunicación serie son conjuntos de reglas y estándares que determinan cómo se transmite, recibe y procesa la información en sistemas de comunicación que utilizan la transmisión de datos en serie. Estos protocolos definen aspectos como el formato de los datos, la velocidad de transmisión, la sincronización, la detección de errores y otros aspectos relevantes para asegurar una comunicación confiable entre dispositivos.

Tres protocolos serie que se utilizan para aplicaciones genéricas son:

- UAR
- I²C
- SPI

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.

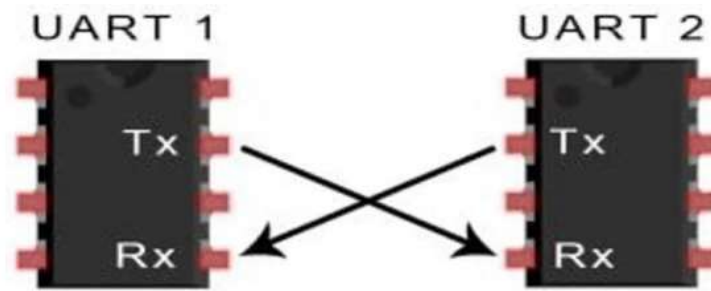
Las principales características son:

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.



Funcionamiento

En cuanto a su funcionamiento es simple, los datos que fluyen desde TX del UART transmisor hasta el RX del UART receptor, una vez se haya conectado e iniciado la comunicación. Sin embargo, la comunicación entre dispositivos puede realizarse de 3 formas:

- Símplex = transmisión de datos en una dirección.
- Half-duplex = transmisión de datos en cualquier dirección, pero no simultáneamente.
- Full-duplex = transmisión de datos en ambas direcciones simultáneamente.

La transmisión de UART convierte los datos en paralelo del dispositivo maestro (por ejemplo, la CPU) en forma de serie y los transmite en serie a la UART receptora. Después volverá a convertir los datos en serie en datos paralelos para el dispositivo receptor. Además, al ser asíncrono, no hace uso de relojes, el direccionamiento distribuye la información adecuadamente. UART agrega bits de inicio y parada que se transfieren para representar el inicio y el final de un mensaje, ayudando así al receptor a saber cuándo comenzar y dejar de leer bits. Cuando el UART receptor detecta un bit de inicio, leerá los bits equivalentes a la tasa de BAUD definida. Esta tasa define la velocidad de transmisión de bits, la cual es similar a la de símbolos, y normalmente la velocidad viene predeterminada en 115.200 baudios.

Es importante que ambos UART tengan definida la misma velocidad en baudios, ya que si la diferencia es de más del 10% la temporización de los bits puede estar desactivada y dejar los datos inutilizables.

Ventajas UART

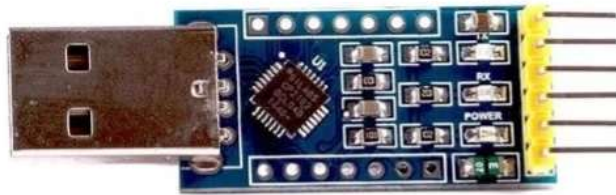
- Proporciona comunicación en serie tanto síncrona como asíncrona
- Disponibilidad de varias velocidades en baudios, lo que la hace adecuada para aplicaciones y dispositivos amplios
- Es una de las formas más sencillas de comunicación en serie. Además, es fácil de operar y está bien documentado, ya que es un método ampliamente utilizado con muchos recursos en línea.
- No se necesita reloj
- Bit de paridad para permitir la verificación de errores

Desventajas UART

- Puede conectar solo dos dispositivos a la vez. No se pueden utilizar varios sistemas maestros y esclavos
- El tamaño de la trama de datos está limitado a solo 9 bits
- Baja velocidad de transmisión de datos.
- Las velocidades en baudios de cada UART deben estar dentro del 10% entre sí para evitar la pérdida de datos.

Ejemplos de UART en microcontroladores

El ejemplo más típico sería un controlador de puente de USB a UART integrado. Este proporciona una solución simple para actualizar los diseños de RS-232 a USB con un mínimo de componentes y espacio de PCB. Proporciona conectividad USB a dispositivos con interfaz UART.



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:

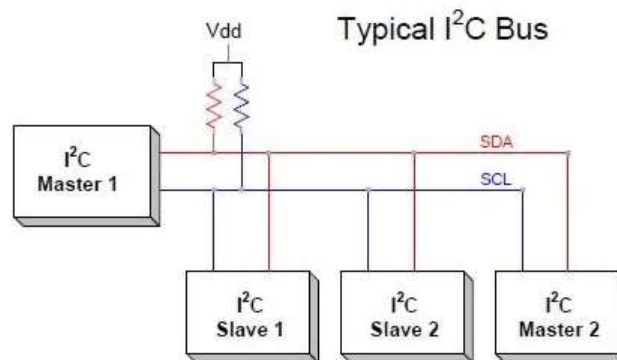
- Maestro: dispositivo que proporciona un reloj para la comunicación
- Esclavo: dispositivo que no es el maestro que utiliza el reloj del maestro para comunicar

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.



Funcionamiento

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 una 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 longitud permitida del bus y el número de dispositivos conectados.

Ventajas de I2C

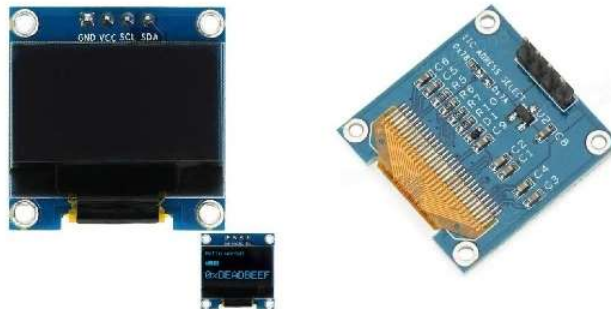
- Flexible, ya que admite comunicación multimaestro y multiesclavo.
- Soporta múltiples maestros.
- Proporciona comunicación en serie síncrona que es mucho más confiable que la asíncrona.
- Es la forma más rápida de comunicación en serie
- Simple, ya que solo usa 2 cables bidireccionales para establecer comunicación entre múltiples dispositivos.
- Adaptable ya que puede adaptarse a las necesidades de varios dispositivos esclavos, y hasta 127 distintos.
- Tiene un recuento bajo de pines/señales incluso con numerosos dispositivos en el bus

Desventajas de I2C

- Requiere más espacio ya que las resistencias consumen un valioso espacio de PCB.
- Velocidad más lenta ya que requiere resistencias pull-up en lugar de resistencias push-pull utilizadas por SPI. También tiene un diseño de drenaje abierto = velocidad limitada.
- Solo el maestro tiene control sobre todo el proceso de comunicación; no hay dos esclavos que puedan comunicarse entre sí directamente
- Puede volverse complejo a medida que aumenta el número de dispositivos.

Ejemplos de I2C en microcontroladores

Un ejemplo de uso simple del protocolo I2C es en los sistemas Grove de la marca Seeed. Además, la mayoría de los sensores Grove tienen una función de cambio de nivel para trabajar en el voltaje adecuado. También hay sensores y componentes más comunes que usan este protocolo, como por ejemplo las pantallas OLED.



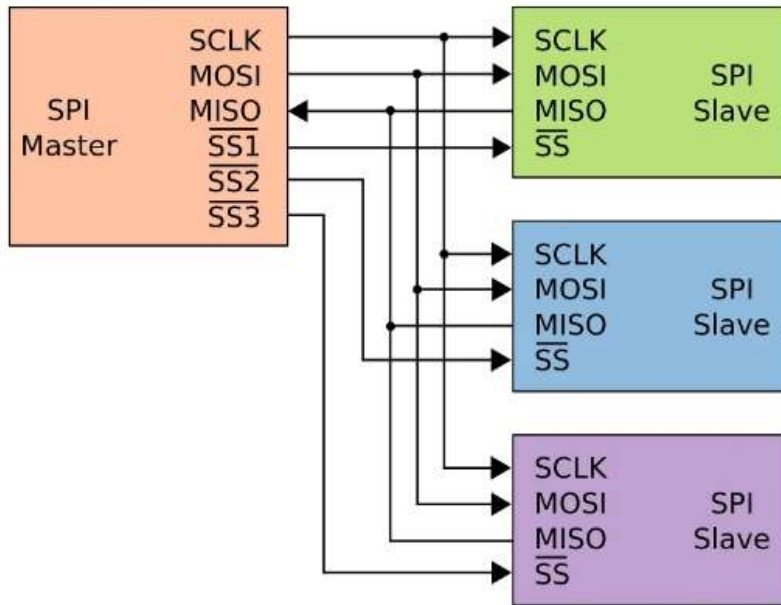
Pantalla OLED 0,96" 128x64 Azul I2C 4 pines - Referencia P0096

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.

Ventajas de SPI

- Es el protocolo más rápido en comparación con UART e I2C.
- El protocolo es simple ya que no existe un sistema de direccionamiento esclavo complicado como I2C.
- Proporciona comunicación en serie síncrona que es mucho más confiable que la asíncrona.
- Líneas MISO y MOSI separadas, lo que significa que los datos se pueden transmitir y recibir al mismo tiempo.

- Sin bits de inicio y parada a diferencia de UART, lo que significa que los datos se pueden transmitir de forma continua sin interrupciones

Desventajas de SPI

- Utiliza cuatro líneas: MOSI, MISO, CLK, NSS.
- Se ocupan más puertos Pin, por lo que es un límite práctico para varios dispositivos.
- No se especifica ningún control de flujo y ningún mecanismo de reconocimiento confirma si los datos se reciben a diferencia de I2C.
- Solo 1 maestro y además solo el maestro tiene control sobre todo el proceso de comunicación; no hay dos esclavos que puedan comunicarse entre sí directamente.
- No hay forma de verificación de errores a diferencia de UART (usando bit de paridad).

Ejemplos de SPI en microcontroladores

Es la forma más común de comunicación series d las Raspberry Pi, por lo que los dispositivos que están basados en esta plataforma usan SPI. Pero también dispositivos más simples que busquen la gran velocidad que ofrece, como por ejemplo módulos para conectividad Ethernet.



Módulo W5500 Lite Ethernet Network SPI - Referencia W0107