

TRABAJO PRACTICO N°1

ARQUITECTURA-Y-CONECTIVIDAD

TAREA 1- GRUPO- 4

MATERIA: Arquitectura y conectividad

PROFESOR: Ing. Jorge Morales

ALUMNO: Carrizo Esteban Dario

INTRODUCCION:

En telecomunicaciones e informática, la **comunicación paralela** es un método para transmitir múltiples dígitos binarios (bits) de manera simultánea. Es lo opuesto a la comunicación en serie, en la que se transmiten los bits uno a uno. Esta distinción es una de las características básicas de los canales de comunicaciones.

La diferencia básica entre un canal de comunicación en serie y uno en paralelo radica en el número de conductores eléctricos usados en la capa física para transmitir bits. La comunicación en paralelo implica el uso de más de uno de estos conductores. Por ejemplo, un canal de 8 bits transmitirá 8 bits (un byte) de manera simultánea, mientras que un canal en serie solo podrá transmitir esta cadena de manera secuencial. Si ambos canales operan a la misma frecuencia de reloj, el canal paralelo sería ocho veces más rápido. Un canal paralelo puede tener conductores adicionales para otras señales, como una señal de reloj para moderar el flujo de datos, una señal para controlar el direccionamiento de datos y señales de handshaking.

1) ¿Cuáles son los protocolos paralelo más usados?, nombre alguno y descríbalos.

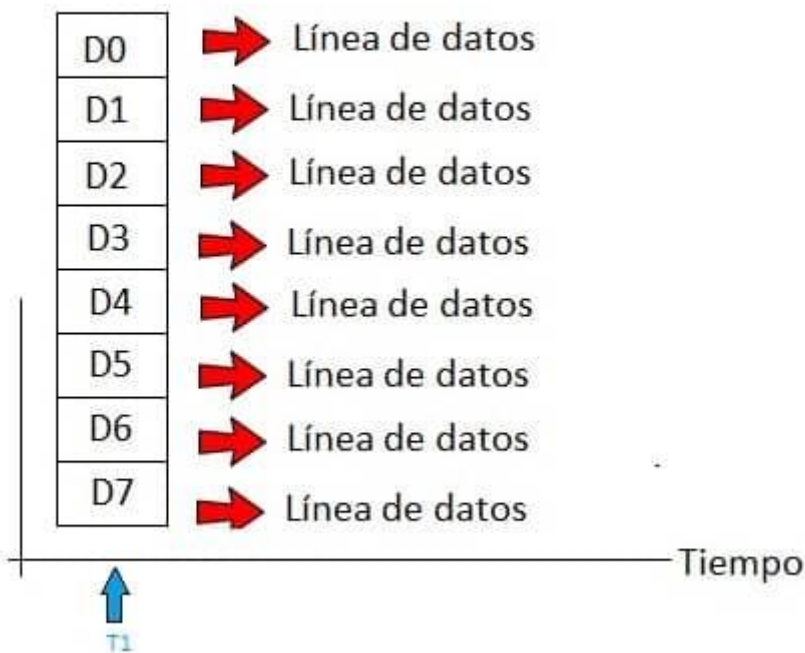
Ejemplo de sistemas de comunicación paralela:

-IBM System/360 *Direct Control Feature* (1964), Estándar System/360 poseía un puerto con un ancho de 8 bits. El controlador de procesos variante Modelo 44 tenía un ancho de 32 bits.

-Buses para periféricos de PC: ISA, ATA, SCSI, PCI y Front side bus, y puertos de impresión IEEE-1284 / Centronics.

-Bus para instrumentación de laboratorio IEEE-488.

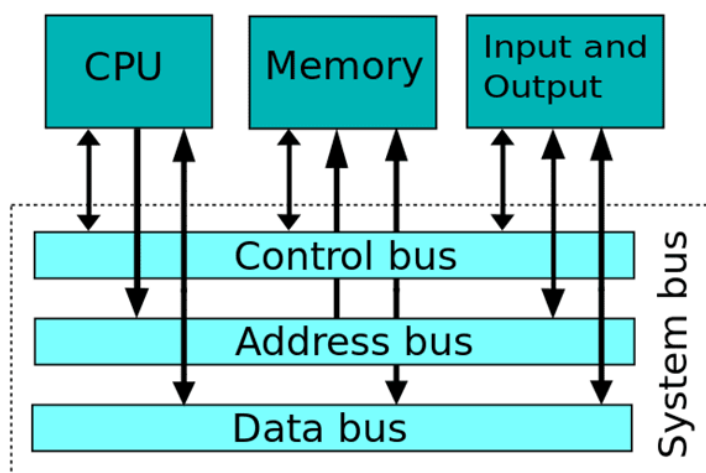
Bus de datos paralelo



Bus de datos paralelo: Este bus es el más rápido y por ende el más utilizado en las arquitecturas internas de los microcontroladores, MPUs, microprocesadores, etc es decir todos estos integrados de control actualmente tienen buses de datos internos que comunican los diferentes sub sistemas, en procesadores es común tener hoy 64bits en bus de datos y en los microcontroladores con arquitectura ARM tenemos buses de datos internos de 32bits, sin embargo por fuera de los circuitos integrados este bus no es tan común, se usa para control de display, pantallas, adquirir datos RAW de sensores.

Las características más importantes de este bus son

1. Cantidad de datos: Baja, se requiere una línea de conexión por cada dato esto limita la cantidad de datos máxima
2. Velocidad de transmisión: Alta, puede transmitir múltiples datos al tiempo
3. Distancia de transmisión: Baja, el dato al propagarse por un conductor eléctrico va perdiendo potencia
4. Numero de receptores: Bajo, normalmente es uno a uno, sin embargo en ocasiones se maneja una señal de habilitador para conectar varios dispositivos, pero esto agrega una línea de control por cada dispositivo
5. Sincronismo: Requiere una señal de sincronismo para saber en que momento leer el dato

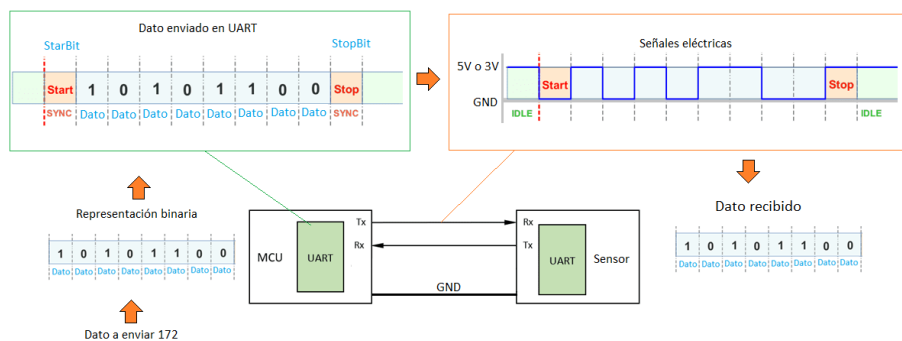


Bus de datos serial: Este bus es el más utilizado en comunicación entre sistemas de control y periféricos.

- **UART (Universal Asynchronous Receiver-Transmitter)** también llamado serial, es el bus más fácil de implementar tanto a nivel de hardware como de software, en su versión más básica usa 2 pines de conexión uno para envío de datos y uno para recepción, en este protocolo se requiere que la tierra sea igual tanto en el receptor como en el emisor

Características

- 1.
1. Cantidad de datos: Baja, este protocolo solo envía de 8 bits por vez
2. Velocidad de transmisión: Media, puede transmitir hasta 1Mbps el tipo de comunicación puede ser halfdúplex o full duplex. debido a que tiene un canal de recepción de datos **Rx** y otro de transmisor de datos **Tx** separados
3. Distancia de transmisión: Media, puede viajar unos metros antes de tener ruido por perdidas
4. Numero de receptores: Bajo, normalmente es uno a uno, sin embargo existen arquitecturas donde se usa este protocolo modificando su capa física para conectar hasta 254 dispositivos
5. Sincronismo: es un protocolo asincrónico no tiene señal de reloj esto implica que el emisor y el receptor deben tener la misma velocidad de lectura entre las cuales la más común es 9600bps



Aplicaciones:

Configuración:

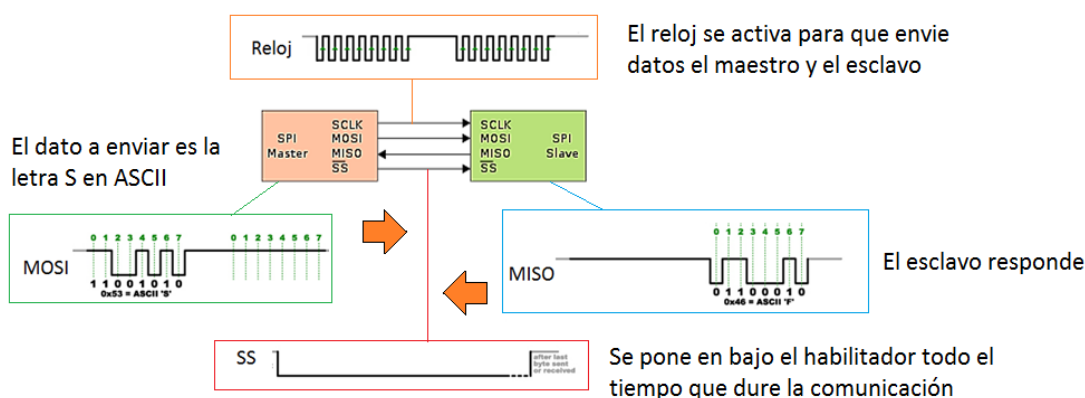
- Se usa como interfase para comunicarse con otro dispositivo y seleccionar sus características o modo de operación
- Monitoreo: Como en el caso del monitor serial de arduino se usa para visualizar datos
- Comunicación con sensores: Sensores como GPS normalmente tiene los datos de salida por puerto serial esto debido a que envían varios datos como

altitud, latitud, tiempo, etc y estos datos no se podrían enviar como un dato analógico o usando pines digitales en modo paralelo

- Comunicación con otros periféricos: Se usa para conectar 2 sistemas de control entre si, donde estos deban compartir datos, por ejemplo un conversor de serial a ethernet
- **SPI (Serial Peripheral Interface)**: entre los protocolos seriales es el de más alta velocidad, su implementación en hardware no requiere acondicionamientos adicional, a nivel de software tiene mayor complejidad que el puerto serial, en su versión estándar usa 4 pines uno para enviar datos, otro para recibir, otro para reloj y finalmente uno para habilitar, con este ultimo se logra que SPI pueda manejar más de un dispositivo, esto lo logra habilitando el dispositivo a usar y des habilitando los otros habitualmente no se tienen más de 3 dispositivos conectados por el mismo SPI

Características

-
- Cantidad de datos: Alta, este protocolo esta diseñado para envío de trama de datos
- Velocidad de transmisión: Alta, puede transmitir hasta 60Mbps ademas el tipo de comunicación es full duplex es decir puede enviar y recibir datos al mismo tiempo ya que tiene un pin para enviar datos llamado **MOSI** (*Master Output Slave Input*) y un pin para recibir datos llamado **MISO** (*Master Input Slave Output*)
- Distancia de transmisión: Media, puede viajar unos 10 metros máximo
- Numero de receptores: Bajo, normalmente es uno a uno, pero puede usar un pin de habilitación llamado **SS** (slave select) donde normalmente se llega a manejar unos 3 dispositivos
- Sincronismo: es un protocolo sincrónico usa una pin de reloj llamado **SCLK**

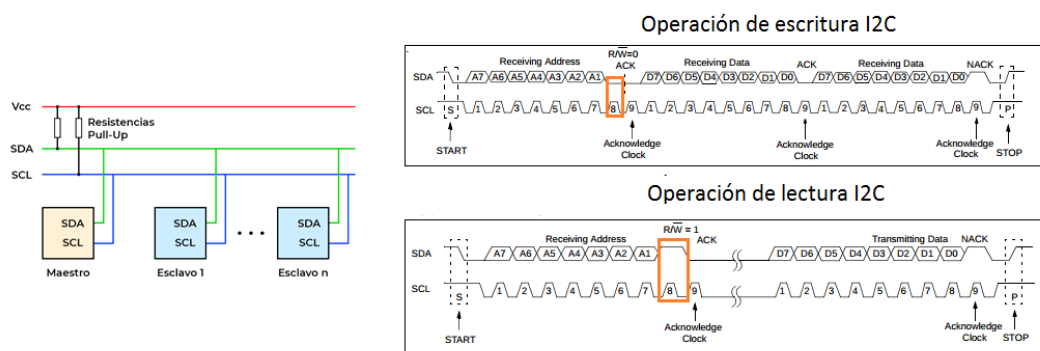


Aplicaciones:

-
- Pantallas: Para controlar pantallas cuando no van por puertos dedicados, debido a que se tienen que enviar muchos datos por segundo el protocolo

más usado para esto es SPI, como el caso de varias pantallas para Arduino, Raspberry, entre otros

- Cámaras: Sucede lo mismo que las pantallas las cámaras envían muchos datos por segundo y si no se usa un puerto dedicado SPI es la mejor alternativa
- Comunicación con sensores: Sensores que tienen envío de datos de alta frecuencia como algunos IMU, radares lidar, entre otros
- Comunicación con memorias: Para almacenar datos en memorias SD u otro tipo de memorias externas el SPI es el protocolo más usado
- Comunicación Ethernet, WiFi: Para adicionar periféricos de comunicación que requieren gran cantidad de datos SPI es la mejor alternativa
- **I2C (Inter-Integrated Circuit):** De los 3 protocolos seriales más usados I2C es el único que está diseñado para conectar múltiples dispositivos al mismo bus de datos y esto hace que tenga mayor complejidad en hardware y en software, además solo requiere dos pines un pin de datos y un pin de reloj
-
- Cantidad de datos: Alta, este protocolo está diseñado para envío de trama de datos
- Velocidad de transmisión: Media, puede transmitir hasta 3Mbps en hs-mode, sin embargo lo más común son velocidades bajas en modo estándar 100kHz y en modo alta velocidad 400kHz, esta comunicación es half duplex eso quiere decir que solo puede transmitir un dato a la vez ya sea de lectura o de escritura y esto por que solo tiene una línea para datos **SDA** (Serial Data)
- Distancia de transmisión: Media, puede viajar unos 8 metros máximo
- Numero de receptores: Alto, este protocolo está diseñado para conectar varios dispositivos al mismo bus de datos a un máximo de 127 cuando se maneja direccionamiento de 7 bits
- Sincronismo: es un protocolo síncrono usa una pin de reloj llamado **SCL** (Serial Clock)



Aplicaciones:

- Comunicación con sensores: Es la aplicación más común sobre todo cuando se quiere hacer una red de sensores o cuando se tienen múltiples sensores, esto por que reduce el conexionado
- Comunicación con periféricos: Los periféricos que no necesitan altas frecuencia de refresco por ejemplo las LCD 2X16 se pueden conectar con conversores I2C