

# Informática II

## Comunicación serie y estándar RS-232

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional  
Facultad Regional Córdoba  
UTN-FRC

– 2024 –

# Introducción

En informática, la comunicación o transmisión de datos entre dispositivos se puede realizar de forma paralela o serie.

# Introducción

En informática, la comunicación o transmisión de datos entre dispositivos se puede realizar de forma paralela o serie.

1. **Comun. paralela:** permite transmitir varios bits de forma simultanea.
2. **Comun. serie:** se transmite un único bit a la vez de forma secuencial.

# Introducción

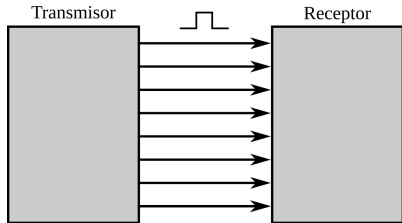
En informática, la comunicación o transmisión de datos entre dispositivos se puede realizar de forma paralela o serie.

1. **Comun. paralela:** permite transmitir varios bits de forma simultanea.
2. **Comun. serie:** se transmite un único bit a la vez de forma secuencial.

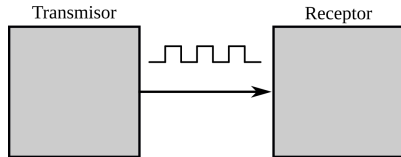
La transmisión de datos permite comunicar equipos electrónicos tales como:

- ▶ PC (Computadoras Personales), PLC (Controladores Lógicos Programables), instrumentos de laboratorios (multímetros, osciloscopios, etc.).
- ▶ Placas de desarrollo de sistemas embebidos (Arduino, Raspberry Pi, etc.).
- ▶ Componentes internos de una PC.

# Comunicación serie y paralela

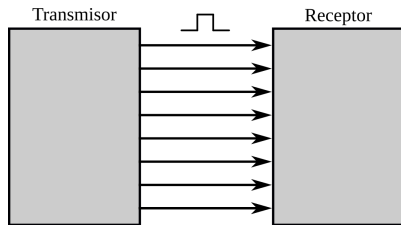


Interfaz paralela

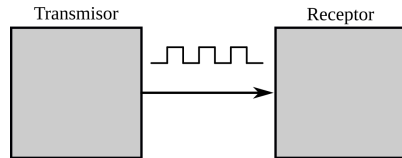


Interfaz serie

# Comunicación serie y paralela



Interfaz paralela



Interfaz serie

- ▶ En la comunicación paralela se necesitan tantas conexiones como bits se quieran transmitir.
- ▶ En la comunicación serie en un solo sentido se necesitan solo una conexión.

# Comunicación serie y paralela – Ejemplos en PC

► Interfaces o buses de comunicación entre comp. internos de una PC son:

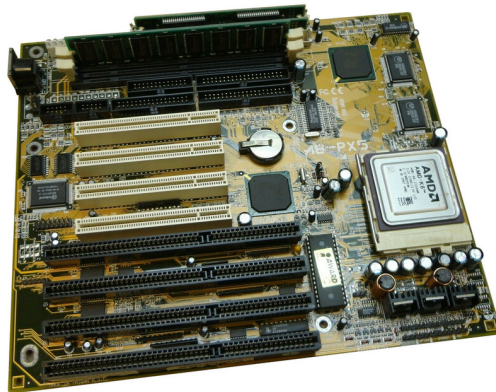
ISA: Industry Standard Architecture.

PCI: Peripheral Component Interconnect.

ATA/IDE: Advanced Technology Attachment/Integrated Drive Electronics.

# Comunicación serie y paralela – Ejemplos en PC

- Interfaces o buses de comunicación entre comp. internos de una PC son:
  - ISA: Industry Standard Architecture.
  - PCI: Peripheral Component Interconnect.
  - ATA/IDE: Advanced Technology Attachment/Integrated Drive Electronics.





# Comunicación serie y paralela – Ejemplos en PC

## ► Interfaces de comunicación externa en paralelo:

**IEEE-1284:** Incluido en la primeras PC de IBM y luego estandarizado por el Institute of Electrical and Electronics Engineers (IEEE).

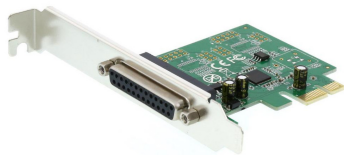
**IEEE-488:** Desarrollado originalmente por Hewlett-Packard para conectar dispositivos de testeo y medición con una PC. Luego estandarizado por el IEEE.

# Comunicación serie y paralela – Ejemplos en PC

## ► Interfaces de comunicación externa en paralelo:

**IEEE-1284:** Incluido en la primeras PC de IBM y luego estandarizado por el Institute of Electrical and Electronics Engineers (IEEE).

**IEEE-488:** Desarrollado originalmente por Hewlett-Packard para conectar dispositivos de testeo y medición con una PC. Luego estandarizado por el IEEE.



IEEE-1284



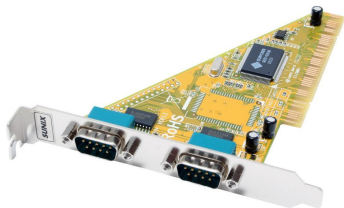
IEEE-488

# Comunicación serie y paralela – Ejemplos en PC

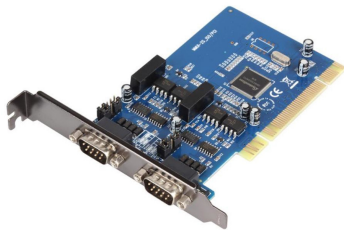
- Interfaces de comunicación externa en serie: [RS-232/EIA-232](#) (a estudiar en detalle) y [RS-485/EIA-485](#) estándar de comunicaciones en bus diferencial de la capa física (modelo OSI –ISO/IEC 7498-1).

# Comunicación serie y paralela – Ejemplos en PC

- Interfaces de comunicación externa en serie: [RS-232/EIA-232](#) (a estudiar en detalle) y [RS-485/EIA-485](#) estándar de comunicaciones en bus diferencial de la capa física (modelo OSI –ISO/IEC 7498-1).



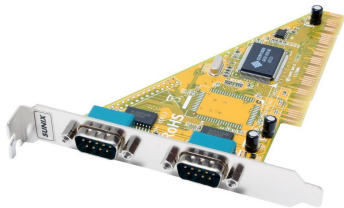
RS-232



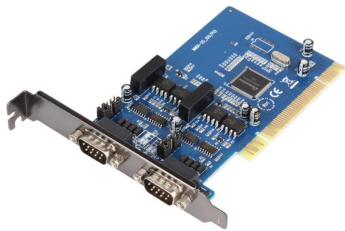
RS-485

# Comunicación serie y paralela – Ejemplos en PC

- ▶ Interfaces de comunicación externa en serie: **RS-232/EIA-232** (a estudiar en detalle) y **RS-485/EIA-485** estándar de comunicaciones en bus diferencial de la capa física (modelo OSI –ISO/IEC 7498-1).



RS-232

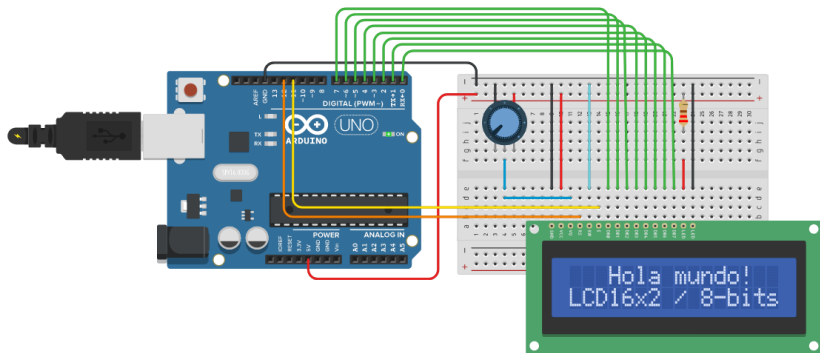


RS-485

- ▶ Interfaces de com. en Sistemas Embebidos (SE): **SPI**, **I<sup>2</sup>C/TWI**, **1-Wire**, etc.

# Comunicación serie y paralela – Ejemplos en SE

## Ejemplo de comunicación paralela en SE

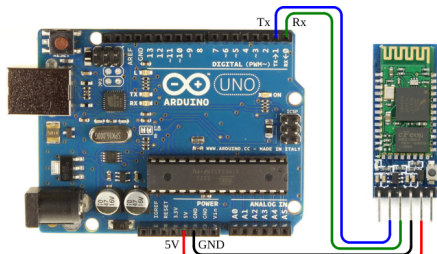


Conexión entre placa Arduino UNO y LCD de 16x2

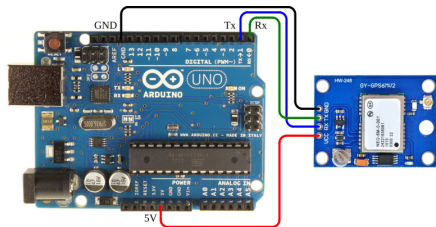
- ▶ Utiliza 8 bits de datos en paralelo.
- ▶ Bits de control: RS (Register Select), E (Enable) y R/W (Read/Write).

# Comunicación serie y paralela – Ejemplos en SE

## Ejemplo de comunicación serie en SE



Módulo Bluetooth HC-05



Módulo GPS NEO-6MV2

Conexión entre placa Arduino UNO con módulos Bluetooth y GPS a través de interfaz serie UART-TTL.

# Transmisión serie síncrona vs. asíncrona

- ▶ Cuando dos dispositivos se comunican intercambian algún tipo de señal detectable que representa los datos (sin comunicación no se enviará ninguna señal a través del canal de comunicación).



# Transmisión serie síncrona vs. asíncrona

- ▶ Cuando dos dispositivos se comunican intercambian algún tipo de señal detectable que representa los datos (sin comunicación no se enviará ninguna señal a través del canal de comunicación).
- ▶ Cuando haya datos para enviar, el dispositivo de envío comenzará a enviar señales.

# Transmisión serie síncrona vs. asíncrona

- ▶ Cuando dos dispositivos se comunican intercambian algún tipo de señal detectable que representa los datos (sin comunicación no se enviará ninguna señal a través del canal de comunicación).
- ▶ Cuando haya datos para enviar, el dispositivo de envío comenzará a enviar señales.
- ▶ Debe entonces haber una forma para que el dispositivo de destino sepa cuándo comenzar a leer datos.

# Transmisión serie síncrona vs. asíncrona

- ▶ Cuando dos dispositivos se comunican intercambian algún tipo de señal detectable que representa los datos (sin comunicación no se enviará ninguna señal a través del canal de comunicación).
- ▶ Cuando haya datos para enviar, el dispositivo de envío comenzará a enviar señales.
- ▶ Debe entonces haber una forma para que el dispositivo de destino sepa cuándo comenzar a leer datos.
- ▶ Se debe establecer y mantener algún tipo de sincronización entre los dispositivos para que las señales se produzcan y detecten con precisión.

# Transmisión serie síncrona vs. asíncrona

- ▶ Cuando dos dispositivos se comunican intercambian algún tipo de señal detectable que representa los datos (sin comunicación no se enviará ninguna señal a través del canal de comunicación).
- ▶ Cuando haya datos para enviar, el dispositivo de envío comenzará a enviar señales.
- ▶ Debe entonces haber una forma para que el dispositivo de destino sepa cuándo comenzar a leer datos.
- ▶ Se debe establecer y mantener algún tipo de sincronización entre los dispositivos para que las señales se produzcan y detecten con precisión.

**Comun. síncrona:** la sincronización se mantiene mediante un bit especial en cada bloque de datos proporcionado por una señal adicional de reloj.

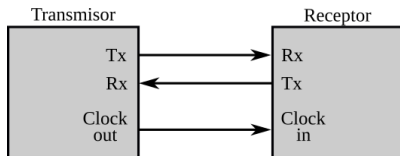
# Transmisión serie síncrona vs. asíncrona

- ▶ Cuando dos dispositivos se comunican intercambian algún tipo de señal detectable que representa los datos (sin comunicación no se enviará ninguna señal a través del canal de comunicación).
- ▶ Cuando haya datos para enviar, el dispositivo de envío comenzará a enviar señales.
- ▶ Debe entonces haber una forma para que el dispositivo de destino sepa cuándo comenzar a leer datos.
- ▶ Se debe establecer y mantener algún tipo de sincronización entre los dispositivos para que las señales se produzcan y detecten con precisión.

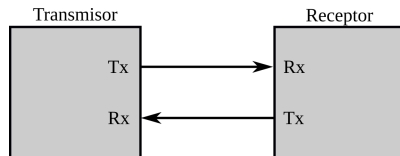
**Comun. síncrona:** la sincronización se mantiene mediante un bit especial en cada bloque de datos proporcionado por una señal adicional de reloj.

**Comun. asíncrona:** la sincronización se restablece con la transmisión de cada carácter mediante el uso de bits de inicio y parada.

# Transmisión serie síncrona vs. asíncrona

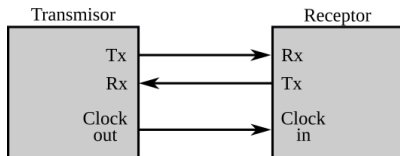


Comunicación síncrona

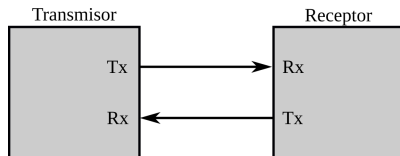


Comunicación asíncrona

# Transmisión serie síncrona vs. asíncrona



Comunicación síncrona



Comunicación asíncrona

- ▶ En ambos casos, los dos dispositivos pueden transmitir y recibir información.
- ▶ En la comunicación síncrona además de las señales de transmisión y recepción de datos se incluye también una señal de reloj o clock.
- ▶ En la comunicación asíncrona la señal de reloj no es necesaria.

# Transmisión serie Half-Duplex y Full-Duplex

- ▶ Las sesiones de comunicación de datos son en general de naturaleza bidireccional.
- ▶ Incluso si el objetivo de la comunicación es enviar un archivo desde el remitente al destino.
- ▶ Generalmente alguna comunicación debe ir desde el destino de regreso al remitente.



# Transmisión serie Half-Duplex y Full-Duplex

- ▶ Las sesiones de comunicación de datos son en general de naturaleza bidireccional.
- ▶ Incluso si el objetivo de la comunicación es enviar un archivo desde el remitente al destino.
- ▶ Generalmente alguna comunicación debe ir desde el destino de regreso al remitente.

## Comunicación Full-Duplex

Ambos dispositivos pueden transmitir al mismo tiempo (análogo a una conversación en persona).

# Transmisión serie Half-Duplex y Full-Duplex

- ▶ Las sesiones de comunicación de datos son en general de naturaleza bidireccional.
- ▶ Incluso si el objetivo de la comunicación es enviar un archivo desde el remitente al destino.
- ▶ Generalmente alguna comunicación debe ir desde el destino de regreso al remitente.

## Comunicación Full-Duplex

Ambos dispositivos pueden transmitir al mismo tiempo (análogo a una conversación en persona).

## Comunicación Half-Duplex

Uno de los dispositivos puede oír o hablar en un momento determinado (similar a una conversación con walkie-talkies donde solo puede hablar aquel que presione el botón).

# Transmisión serie Half-Duplex y Full-Duplex

- ▶ Las sesiones de comunicación de datos son en general de naturaleza bidireccional.
- ▶ Incluso si el objetivo de la comunicación es enviar un archivo desde el remitente al destino.
- ▶ Generalmente alguna comunicación debe ir desde el destino de regreso al remitente.

## Comunicación Full-Duplex

Ambos dispositivos pueden transmitir al mismo tiempo (análogo a una conversación en persona).

## Comunicación Half-Duplex

Uno de los dispositivos puede oír o hablar en un momento determinado (similar a una conversación con walkie-talkies donde solo puede hablar aquel que presione el botón).

Cuando la comunicación es unidireccional, se conoce con el nombre de **Simplex**.



# El estándar RS-232

- ▶ Se utiliza ampliamente en la industria debido principalmente a su relativa simplicidad en el hardware.

# El estándar RS-232

- ▶ Se utiliza ampliamente en la industria debido principalmente a su relativa simplicidad en el hardware.
- ▶ Estándares de comunicación en serie más utilizado: EIA/TIA-232-E.

# El estándar RS-232

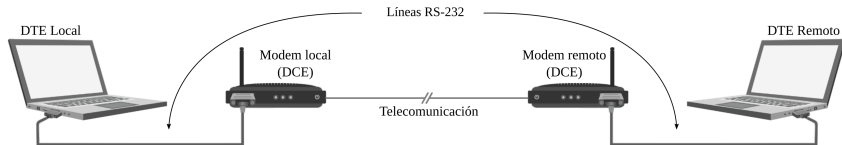
- ▶ Se utiliza ampliamente en la industria debido principalmente a su relativa simplicidad en el hardware.
- ▶ Estándares de comunicación en serie más utilizado: EIA/TIA-232-E.

El estándar se refiere a la comunicación de datos entre un sistema host (DTE, Data Terminal Equipment) y un sistema periférico (DCE, Data Circuit-Terminating Equipment o Data Communication Equipment).

# El estándar RS-232

- ▶ Se utiliza ampliamente en la industria debido principalmente a su relativa simplicidad en el hardware.
- ▶ Estándares de comunicación en serie más utilizado: EIA/TIA-232-E.

El estándar se refiere a la comunicación de datos entre un sistema host (DTE, Data Terminal Equipment) y un sistema periférico (DCE, Data Circuit-Terminating Equipment o Data Communication Equipment).

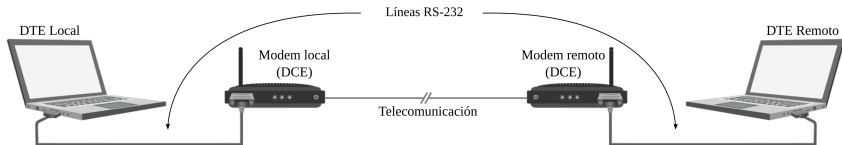




# El estándar RS-232

- ▶ Se utiliza ampliamente en la industria debido principalmente a su relativa simplicidad en el hardware.
- ▶ Estándares de comunicación en serie más utilizado: EIA/TIA-232-E.

El estándar se refiere a la comunicación de datos entre un sistema host (DTE, Data Terminal Equipment) y un sistema periférico (DCE, Data Circuit-Terminating Equipment o Data Communication Equipment).



El EIA/TIA-232-E fue desarrollado por:

*Electronic Industry Association* y la *Telecommunications Industry Association*, se conoce más popularmente simplemente como **RS-232**, donde “RS” significa *Recommended Standard*.

# El estándar RS-232

- Gran parte de la terminología RS-232 refleja su origen como estándar para las com. entre un terminal de computadora (PC) y un módem externo.

# El estándar RS-232

- ▶ Gran parte de la terminología RS-232 refleja su origen como estándar para las com. entre un terminal de computadora (PC) y un módem externo.
- ▶ En la actualidad resulta más frecuente el uso del puerto RS-232 para conectar una PC a un sistema embebido o bien dos sistemas embebidos entre sí (UART-TTL).

# El estándar RS-232

- ▶ Gran parte de la terminología RS-232 refleja su origen como estándar para las com. entre un terminal de computadora (PC) y un módem externo.
- ▶ En la actualidad resulta más frecuente el uso del puerto RS-232 para conectar una PC a un sistema embebido o bien dos sistemas embebidos entre sí (UART-TTL).

El EIA-TIA-232 es un estándar “completo” que garantiza la compatibilidad entre el host y los sistemas periféricos mediante la especificación de:

# El estándar RS-232

- ▶ Gran parte de la terminología RS-232 refleja su origen como estándar para las com. entre un terminal de computadora (PC) y un módem externo.
- ▶ En la actualidad resulta más frecuente el uso del puerto RS-232 para conectar una PC a un sistema embebido o bien dos sistemas embebidos entre sí (UART-TTL).

El EIA-TIA-232 es un estándar “completo” que garantiza la compatibilidad entre el host y los sistemas periféricos mediante la especificación de:

1. señales y niveles comunes de voltaje (caract. eléctricas),

# El estándar RS-232

- ▶ Gran parte de la terminología RS-232 refleja su origen como estándar para las com. entre un terminal de computadora (PC) y un módem externo.
- ▶ En la actualidad resulta más frecuente el uso del puerto RS-232 para conectar una PC a un sistema embebido o bien dos sistemas embebidos entre sí (UART-TTL).

El EIA-TIA-232 es un estándar “completo” que garantiza la compatibilidad entre el host y los sistemas periféricos mediante la especificación de:

1. señales y niveles comunes de voltaje (caract. eléctricas),
2. configuraciones de pines y cableado (caract. mecánicas) e

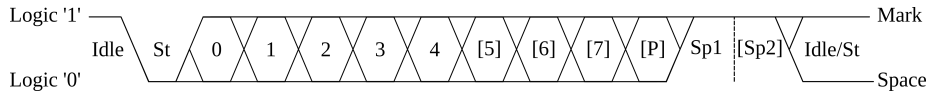
# El estándar RS-232

- ▶ Gran parte de la terminología RS-232 refleja su origen como estándar para las com. entre un terminal de computadora (PC) y un módem externo.
- ▶ En la actualidad resulta más frecuente el uso del puerto RS-232 para conectar una PC a un sistema embebido o bien dos sistemas embebidos entre sí (UART-TTL).

El EIA-TIA-232 es un estándar “completo” que garantiza la compatibilidad entre el host y los sistemas periféricos mediante la especificación de:

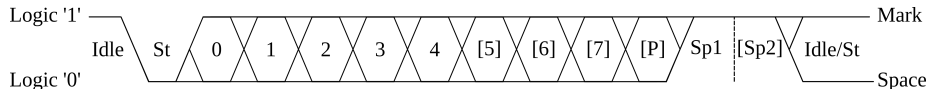
1. señales y niveles comunes de voltaje (caract. eléctricas),
2. configuraciones de pines y cableado (caract. mecánicas) e
3. información de control mínima entre el host y los sistemas periféricos (caract. funcionales).

# El estándar RS-232 – Trama de comunicación



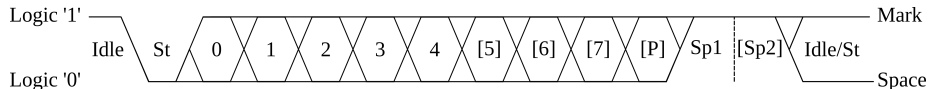


# El estándar RS-232 – Trama de comunicación



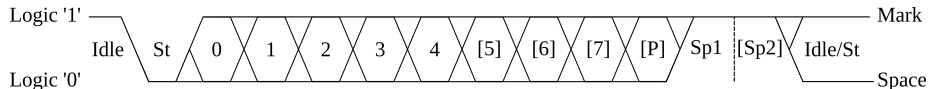
- Idle: Ocioso, no hay transferencia de datos (TxD y RxD) [Logic 1]

# El estándar RS-232 – Trama de comunicación



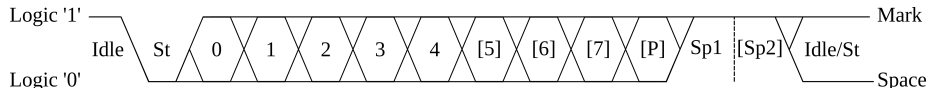
- Idle: Ocioso, no hay transferencia de datos (TxD y RxD) [Logic 1]
- St: Start bit (bit de arranque) [Logic 0]

# El estándar RS-232 – Trama de comunicación



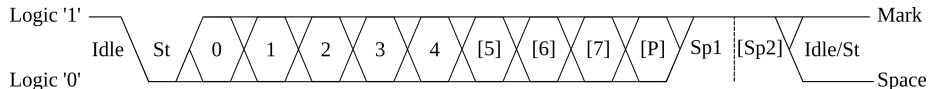
- Idle: Ocioso, no hay transferencia de datos (TxD y RxD) [Logic 1]
- St: Start bit (bit de arranque) [Logic 0]
- (n): Bits de datos

# El estándar RS-232 – Trama de comunicación



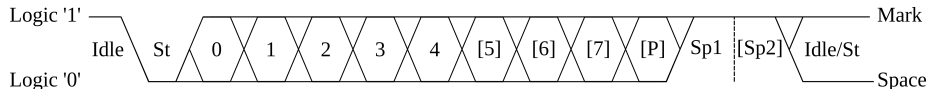
- ▶ Idle: Ocioso, no hay transferencia de datos (TxD y Rx) [Logic 1]
- ▶ St: Start bit (bit de arranque) [Logic 0]
- ▶ (n): Bits de datos
- ▶ P: Parity bit (bit de paridad)

# El estándar RS-232 – Trama de comunicación



- ▶ Idle: Ocioso, no hay transferencia de datos (TxD y RxD) [Logic 1]
- ▶ St: Start bit (bit de arranque) [Logic 0]
- ▶ (n): Bits de datos
- ▶ P: Parity bit (bit de paridad)
- ▶ Sp: Stop bit (bit de parada) [Logic 1]

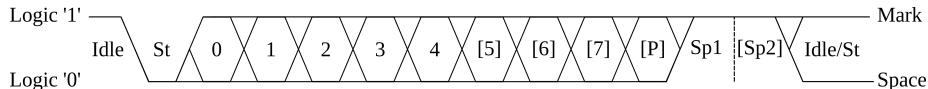
# El estándar RS-232 – Trama de comunicación



- Idle: Ocioso, no hay transferencia de datos (TxD y RxD) [Logic 1]
- St: Start bit (bit de arranque) [Logic 0]
- (n): Bits de datos
- P: Parity bit (bit de paridad)
- Sp: Stop bit (bit de parada) [Logic 1]

El bit menos significativo (LSb) se envía primero.

# El estándar RS-232 – Trama de comunicación



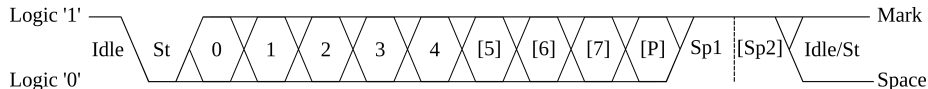
- ▶ Idle: Ocioso, no hay transferencia de datos (TxD y Rx) [Logic 1]
- ▶ St: Start bit (bit de arranque) [Logic 0]
- ▶ (n): Bits de datos
- ▶ P: Parity bit (bit de paridad)
- ▶ Sp: Stop bit (bit de parada) [Logic 1]

El bit menos significativo (LSb) se envía primero.

Algunas alternativas de tramas

- ▶ 8N1: 8 bits de datos, sin (N) bit de paridad y 1 bit de stop.
- ▶ 5E2: 5 bits de datos, paridad par (E) y 2 bits de stop.

# El estándar RS-232 – Trama de comunicación



- ▶ Idle: Ocioso, no hay transferencia de datos (TxD y Rx) [Logic 1]
- ▶ St: Start bit (bit de arranque) [Logic 0]
- ▶ (n): Bits de datos
- ▶ P: Parity bit (bit de paridad)
- ▶ Sp: Stop bit (bit de parada) [Logic 1]

El bit menos significativo (LSb) se envía primero.

Algunas alternativas de tramas

- ▶ 8N1: 8 bits de datos, sin (N) bit de paridad y 1 bit de stop.
- ▶ 5E2: 5 bits de datos, paridad par (E) y 2 bits de stop.

¿Qué duración tiene cada bit?



# El estándar RS-232 – Características eléctricas

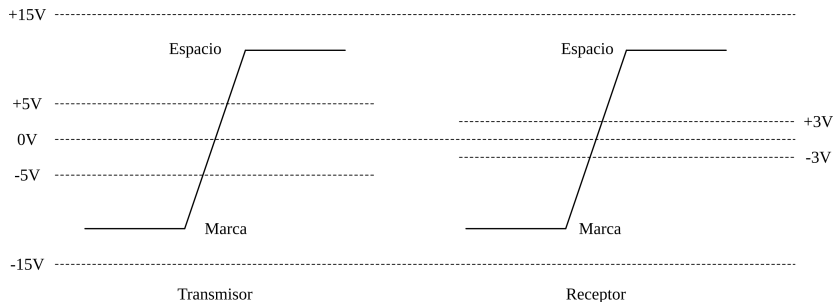
Especificaciones eléctricas del **puerto serie** o **RS-232**:

- ▶ Un cero lógico o *espacio* entre +5 y +15V (transmisor).
- ▶ Un uno lógico o *marca* entre -5 y -15V (transmisor).
- ▶ Los niveles de tensión de entrada tienen un margen de ruido de 2V.
- ▶ Los valores entre +3V y -3V representan estados indefinidos.

# El estándar RS-232 – Características eléctricas

Especificaciones eléctricas del **puerto serie** o **RS-232**:

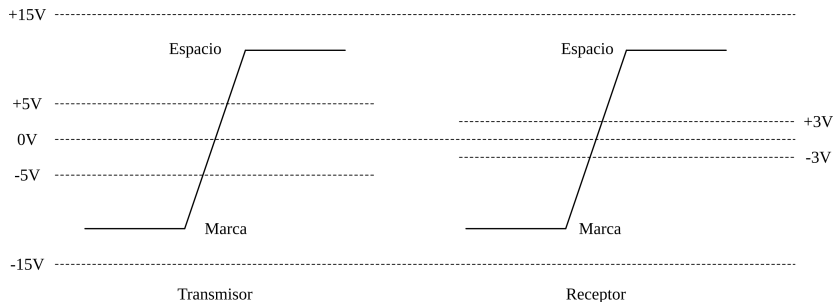
- ▶ Un cero lógico o *espacio* entre +5 y +15V (transmisor).
- ▶ Un uno lógico o *marca* entre -5 y -15V (transmisor).
- ▶ Los niveles de tensión de entrada tienen un margen de ruido de 2V.
- ▶ Los valores entre +3V y -3V representan estados indefinidos.



# El estándar RS-232 – Características eléctricas

Especificaciones eléctricas del **puerto serie** o **RS-232**:

- ▶ Un cero lógico o *espacio* entre +5 y +15V (transmisor).
- ▶ Un uno lógico o *marca* entre -5 y -15V (transmisor).
- ▶ Los niveles de tensión de entrada tienen un margen de ruido de 2V.
- ▶ Los valores entre +3V y -3V representan estados indefinidos.



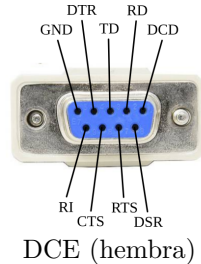
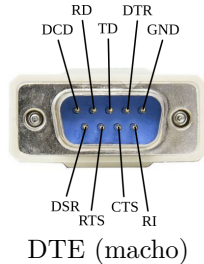
(Otros parámetros son: corriente de cortocircuito, capacitancia máxima de línea, tasa de cambio del niveles de las señales, impedancia de la línea, etc.)

# El estándar RS-232 – Características mecánicas

- ▶ El RS-232 especifica un conector de 25 pines (tamaño mínimo para albergar todas las señales definidas en la parte funcional del estándar).
- ▶ El conector para el DTE tiene una carcasa hembra con pines de conexión macho, y para el DCE es macho para la carcasa y hembra para los pines.
- ▶ El conector más popular es el DB9 (9 pines).

# El estándar RS-232 – Características mecánicas

- ▶ El RS-232 especifica un conector de 25 pines (tamaño mínimo para albergar todas las señales definidas en la parte funcional del estándar).
- ▶ El conector para el **DTE** tiene una carcasa hembra con pines de conexión macho, y para el **DCE** es macho para la carcasa y hembra para los pines.
- ▶ El conector más popular es el DB9 (9 pines).

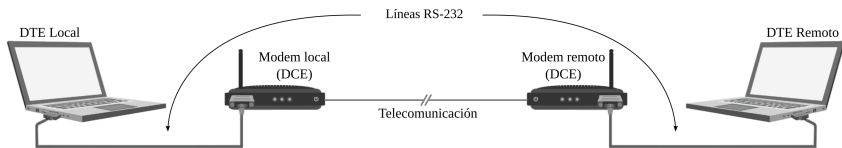


# El estándar RS-232 – Características funcionales

- ▶ El RS-232 define la función de las diferentes señales que se utilizan en la interfaz.
- ▶ Pocas aplicaciones requieren todas estas señales definidas en el estándar.
- ▶ De hecho, aplicaciones como la comunicación por módem requiere solo nueve señales (dos señales de datos, seis señales de control y referencia).

# El estándar RS-232 – Características funcionales

- ▶ El RS-232 define la función de las diferentes señales que se utilizan en la interfaz.
- ▶ Pocas aplicaciones requieren todas estas señales definidas en el estándar.
- ▶ De hecho, aplicaciones como la comunicación por módem requiere solo nueve señales (dos señales de datos, seis señales de control y referencia).



Interfaz entre DTE y DCE

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           |  |
|----------------------|---------|---------------------|--|
| 3 (2)                | TD      | Transmit Data       |  |
| 2 (3)                | RD      | Receive Data        |  |
| 7 (4)                | RTS     | Request To Send     |  |
| 8 (5)                | CTS     | Clear To Send       |  |
| 6 (6)                | DSR     | Data Set Ready      |  |
| 5 (7)                | SG      | Signal Ground       |  |
| 1 (8)                | DCD     | Data Carrier Detect |  |
| 4 (20)               | DTR     | Data Terminal Ready |  |
| 9 (22)               | RI      | Ring Indicator      |  |



# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev.   | Full name            | PC — Perif. |
|----------------------|-----------|----------------------|-------------|
| 3 (2)                | <b>TD</b> | <b>Transmit Data</b> | DTE → DCE   |
| 2 (3)                | RD        | Receive Data         |             |
| 7 (4)                | RTS       | Request To Send      |             |
| 8 (5)                | CTS       | Clear To Send        |             |
| 6 (6)                | DSR       | Data Set Ready       |             |
| 5 (7)                | SG        | Signal Ground        |             |
| 1 (8)                | DCD       | Data Carrier Detect  |             |
| 4 (20)               | DTR       | Data Terminal Ready  |             |
| 9 (22)               | RI        | Ring Indicator       |             |

**TD:** Salida de datos seriales, TxD

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           | PC — Perif. |
|----------------------|---------|---------------------|-------------|
| 3 (2)                | TD      | Transmit Data       | DTE → DCE   |
| 2 (3)                | RD      | Receive Data        | DTE ← DCE   |
| 7 (4)                | RTS     | Request To Send     |             |
| 8 (5)                | CTS     | Clear To Send       |             |
| 6 (6)                | DSR     | Data Set Ready      |             |
| 5 (7)                | SG      | Signal Ground       |             |
| 1 (8)                | DCD     | Data Carrier Detect |             |
| 4 (20)               | DTR     | Data Terminal Ready |             |
| 9 (22)               | RI      | Ring Indicator      |             |

RD: Entrada de datos seriales, RxD

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev.    | Full name              | PC — Perif. |
|----------------------|------------|------------------------|-------------|
| 3 (2)                | TD         | Transmit Data          | DTE → DCE   |
| 2 (3)                | RD         | Receive Data           | DTE ← DCE   |
| 7 (4)                | <b>RTS</b> | <b>Request To Send</b> | DTE → DCE   |
| 8 (5)                | CTS        | Clear To Send          |             |
| 6 (6)                | DSR        | Data Set Ready         |             |
| 5 (7)                | SG         | Signal Ground          |             |
| 1 (8)                | DCD        | Data Carrier Detect    |             |
| 4 (20)               | DTR        | Data Terminal Ready    |             |
| 9 (22)               | RI         | Ring Indicator         |             |

**RTS:** Le indica al módem que la UART está lista para comunicarse

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           | PC — Perif. |
|----------------------|---------|---------------------|-------------|
| 3 (2)                | TD      | Transmit Data       | DTE → DCE   |
| 2 (3)                | RD      | Receive Data        | DTE ← DCE   |
| 7 (4)                | RTS     | Request To Send     | DTE → DCE   |
| 8 (5)                | CTS     | Clear To Send       | DTE ← DCE   |
| 6 (6)                | DSR     | Data Set Ready      |             |
| 5 (7)                | SG      | Signal Ground       |             |
| 1 (8)                | DCD     | Data Carrier Detect |             |
| 4 (20)               | DTR     | Data Terminal Ready |             |
| 9 (22)               | RI      | Ring Indicator      |             |

**CTS:** Indica que el módem está listo para comunicarse

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           | PC — Perif. |
|----------------------|---------|---------------------|-------------|
| 3 (2)                | TD      | Transmit Data       | DTE → DCE   |
| 2 (3)                | RD      | Receive Data        | DTE ← DCE   |
| 7 (4)                | RTS     | Request To Send     | DTE → DCE   |
| 8 (5)                | CTS     | Clear To Send       | DTE ← DCE   |
| 6 (6)                | DSR     | Data Set Ready      | DTE ← DCE   |
| 5 (7)                | SG      | Signal Ground       |             |
| 1 (8)                | DCD     | Data Carrier Detect |             |
| 4 (20)               | DTR     | Data Terminal Ready |             |
| 9 (22)               | RI      | Ring Indicator      |             |

**DSR:** Le indica a la UART que el módem está listo para establecer una conexión

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           | PC — Perif. |
|----------------------|---------|---------------------|-------------|
| 3 (2)                | TD      | Transmit Data       | DTE → DCE   |
| 2 (3)                | RD      | Receive Data        | DTE ← DCE   |
| 7 (4)                | RTS     | Request To Send     | DTE → DCE   |
| 8 (5)                | CTS     | Clear To Send       | DTE ← DCE   |
| 6 (6)                | DSR     | Data Set Ready      | DTE ← DCE   |
| 5 (7)                | SG      | Signal Ground       | DTE — DCE   |
| 1 (8)                | DCD     | Data Carrier Detect | DTE ← DCE   |
| 4 (20)               | DTR     | Data Terminal Ready |             |
| 9 (22)               | RI      | Ring Indicator      |             |

DCD: Indica que el módem detecta “portadora”

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           | PC — Perif. |
|----------------------|---------|---------------------|-------------|
| 3 (2)                | TD      | Transmit Data       | DTE → DCE   |
| 2 (3)                | RD      | Receive Data        | DTE ← DCE   |
| 7 (4)                | RTS     | Request To Send     | DTE → DCE   |
| 8 (5)                | CTS     | Clear To Send       | DTE ← DCE   |
| 6 (6)                | DSR     | Data Set Ready      | DTE ← DCE   |
| 5 (7)                | SG      | Signal Ground       | DTE — DCE   |
| 1 (8)                | DCD     | Data Carrier Detect | DTE ← DCE   |
| 4 (20)               | DTR     | Data Terminal Ready | DTE → DCE   |
| 9 (22)               | RI      | Ring Indicator      |             |

**DTR:** Opuesto a DSR. Le indica al módem que la UART está lista para establecer conexión

# El estándar RS-232 – Características funcionales

| DB-9 (DB-25) Pin No. | Abbrev. | Full name           | PC — Perif. |
|----------------------|---------|---------------------|-------------|
| 3 (2)                | TD      | Transmit Data       | DTE → DCE   |
| 2 (3)                | RD      | Receive Data        | DTE ← DCE   |
| 7 (4)                | RTS     | Request To Send     | DTE → DCE   |
| 8 (5)                | CTS     | Clear To Send       | DTE ← DCE   |
| 6 (6)                | DSR     | Data Set Ready      | DTE ← DCE   |
| 5 (7)                | SG      | Signal Ground       | DTE — DCE   |
| 1 (8)                | DCD     | Data Carrier Detect | DTE ← DCE   |
| 4 (20)               | DTR     | Data Terminal Ready | DTE → DCE   |
| 9 (22)               | RI      | Ring Indicator      | DTE ← DCE   |

RI: Se activa ante la presencia de llamada



# La UART

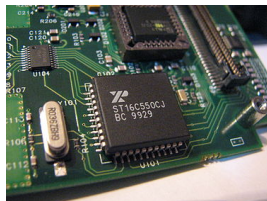
- ▶ Las señales necesarias para la comunicación en serie son generadas y recibidas por un circuito integrado (CI) conocido como UART (Universal Asynchronous Receiver/Transmitter).
- ▶ Este circuito integrado hace de conversor paralelo/serie para “serializar/des-serializar” los datos.

# La UART

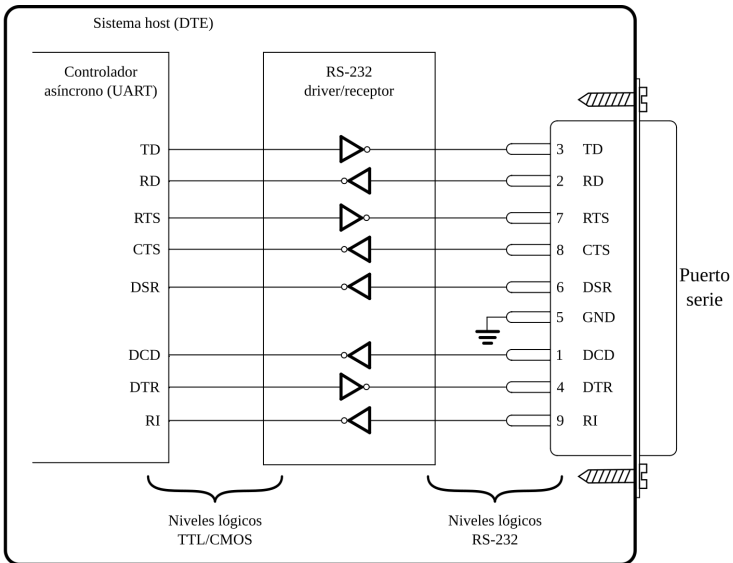
- ▶ Las señales necesarias para la comunicación en serie son generadas y recibidas por un circuito integrado (CI) conocido como UART (Universal Asynchronous Receiver/Transmitter).
- ▶ Este circuito integrado hace de conversor paralelo/serie para “serializar/des-serializar” los datos.
- ▶ La UART del sistema host genera los bits de inicio y parada para indicar al sistema periférico cuándo se inicia y termina la comunicación.
- ▶ Hay otro CI que hace de driver de entrada/salida del RS-232 y adapta los nivel de tensión necesarios entre la UART y el RS-232.

# La UART

- ▶ Las señales necesarias para la comunicación en serie son generadas y recibidas por un circuito integrado (CI) conocido como UART (Universal Asynchronous Receiver/Transmitter).
- ▶ Este circuito integrado hace de conversor paralelo/serie para “serializar/des-serializar” los datos.
- ▶ La UART del sistema host genera los bits de inicio y parada para indicar al sistema periférico cuándo se inicia y termina la comunicación.
- ▶ Hay otro CI que hace de driver de entrada/salida del RS-232 y adapta los nivel de tensión necesarios entre la UART y el RS-232.



# La UART



# La UART del $\mu$ C ATmega328

