

# Microprocesadores, Tema 8:

## Comunicaciones asíncronas con USART del PIC

Guillermo Carpintero del Barrio

Marta Ruiz Llata

Alejandro Quesada Pareja

Universidad **Carlos III** de Madrid

# Standard de Comunicación

## Protocolos Standard de Comunicación Serie

Interfase	Formato	Num. Dispositivos	Separación max.	Velocidad de Tx (bits/seg)
RS-232	Serie Asinc.	2	15 – 30 m	20k
IrDA	Serie Asinc.	2	2 m	115k
SPI	Serie Sinc.	8	3 m	2.1M
I2C	Serie Sinc.	40	5.5 m	400k
USB	Serie Asinc.	127	5 m	12M

# Objetivos

Parámetros de la **comunicación asíncrona**

Configuración de la USART del PIC para comunicaciones asíncronas

Detección de errores

Utilización de las interrupciones

# Bases de la Comunicación Asíncrona

## Características de la Comunicación Serie

La capa física solo es capaz de transportar un solo bit por unidad de tiempo.

Se emplea cuando la transmisión tiene lugar mas allá de unos metros.

Requiere controladores complejos y protocolos de comunicación.

Fuente debe descomponer los bytes en bits

Destino debe realizar operación inversa

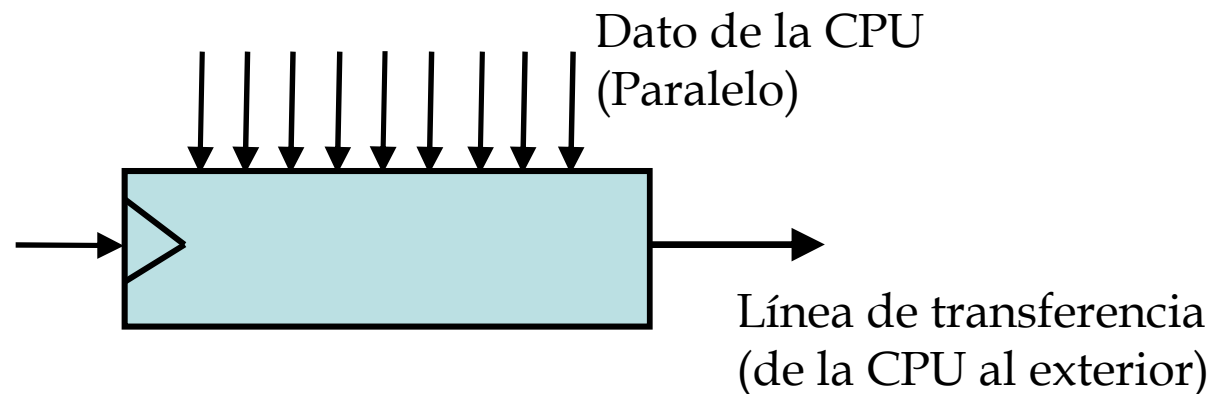
La unidad de velocidad de transferencia es BAUDIO (= bits/seg).

# Bases de la Comunicación Asíncrona

## Base de operación de un puerto Serie

Se trata de una conversión PARALELO a SERIE.

Registro de desplazamiento

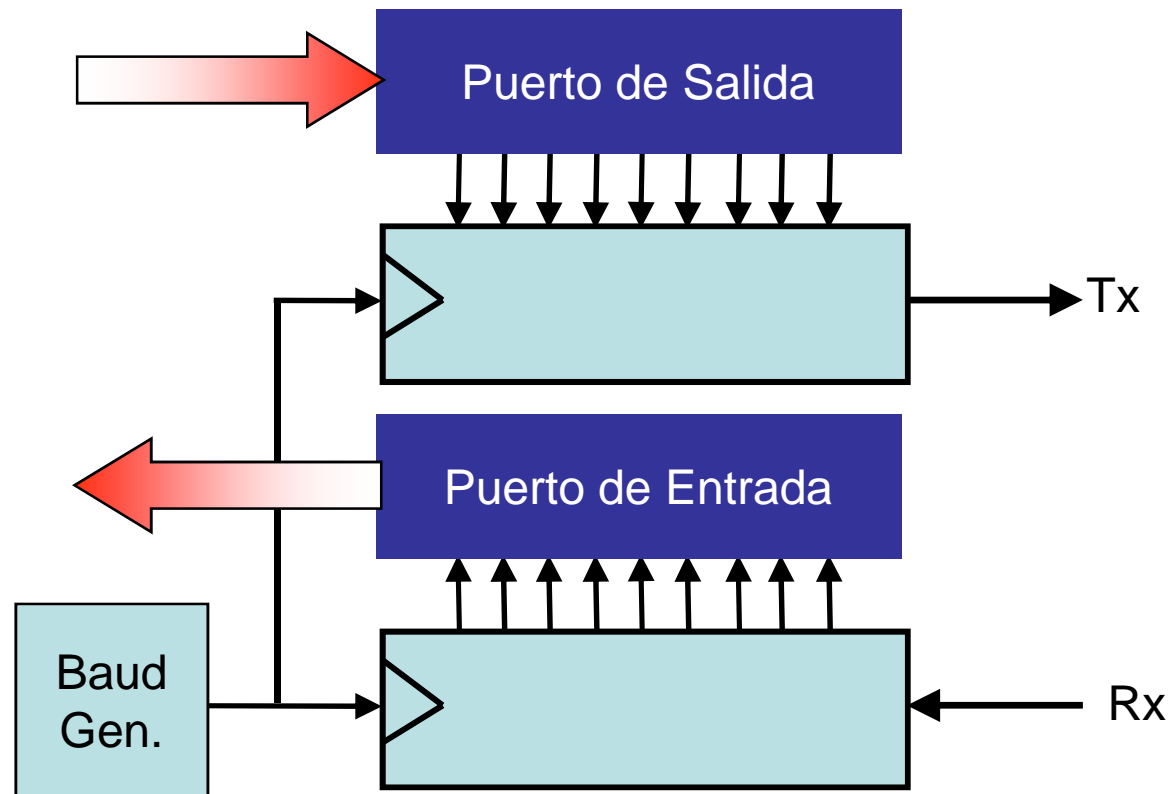


Puerto de Salida (Tx)  
**unidireccional !!!**

La **unidad de información** que se transfiere es el carácter, y no el byte.

# Bases de la Comunicación Asíncrona

Controlador de interfase serie = Universal Asynch Rx Tx



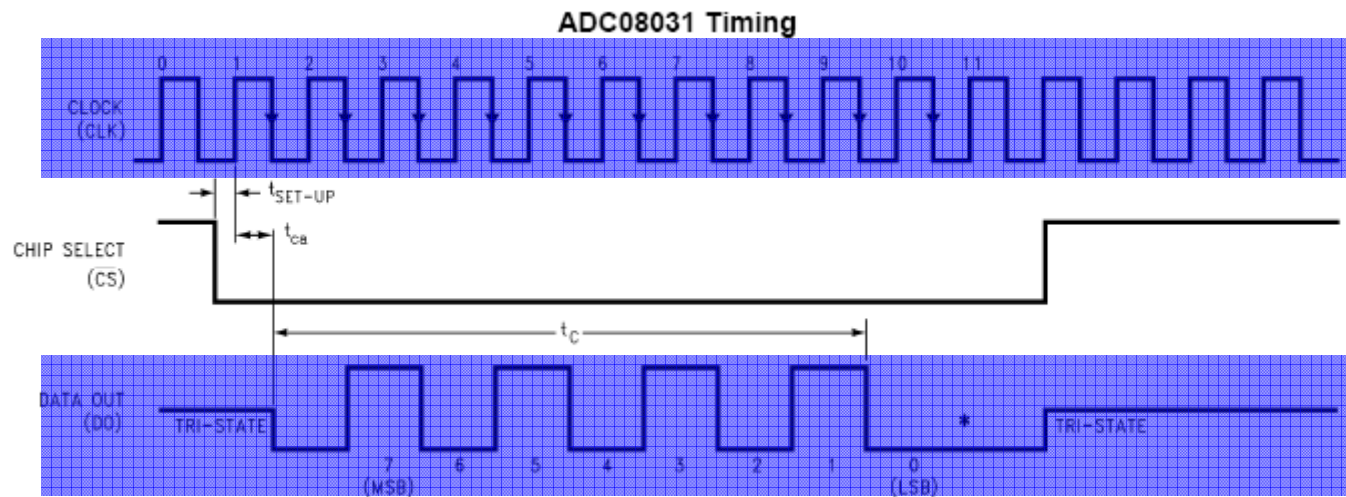
# Bases de la Comunicación Asíncrona

## Comunicación Serie en sus dos formas

Asíncrona



Síncrona



# Introducción a USART

**USART = Universal Synchronous Asynchronous Receiver Transmitter**

**Modos de operación:**

- **Síncrono: Utiliza una línea de reloj independiente de la de datos**
- **Asíncrono: No se utiliza reloj en la transmisión de los datos**

**Características de la USART del PIC:**

- **Transmite y recibe datos serie (de 8 ó 9 bits)**
- **Detecta errores**
- **Posibilidad de generar interrupción**
- **Almacena datos en buffers**



# Introducción a USART

Register Name	Description
TXREG	Transmit Data Register
RCREG	Receive Data Register

Register Name	Description
TXSTA	Transmit Status and Control
RCSTA	Receive Status and Control

Register Name	Description
SPBRG	Baud Rate Generator

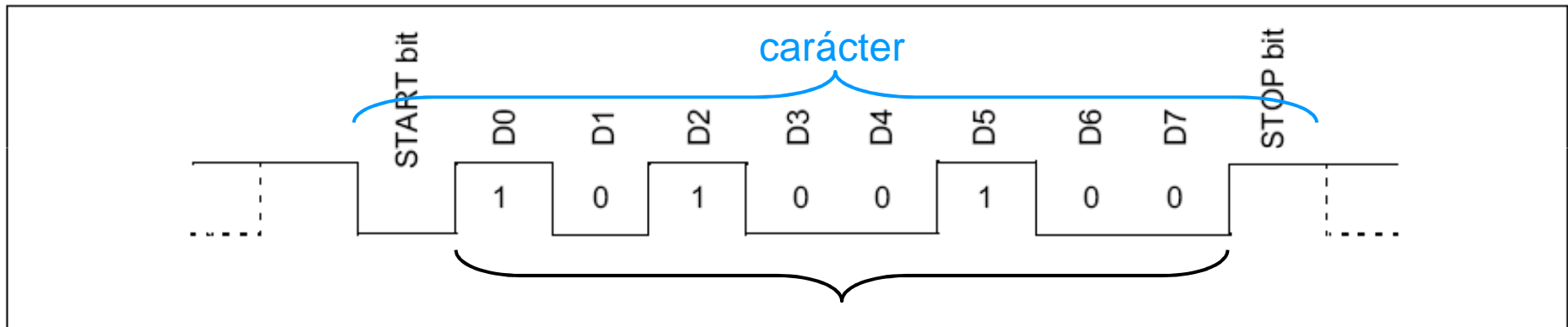
Register Name	Description
INTCON	Interrupt Control Register
RCON	RESET Control Register
PIE1, PIE3	Peripheral Interrupt Enable Registers
PIR1, PIR3	Peripheral Interrupt Flag Registers
IPR1, IPR3	Peripheral Interrupt Priority Registers

# Comunicaciones asíncronas

Al no existir reloj → Necesidad de sincronizar RX y TX

- Velocidad de transmisión fija y predeterminada
- Bits de comienzo y final

FIGURE 1: ASYNCHRONOUS MODE SIGNAL



Características:

- Línea en reposo → Siempre a nivel alto
- 1Bit de comienzo: Siempre a nivel bajo
- El primer bit del dato a transmitir es el LSB
- 1Bit de final: Detrás del MSB, siempre a nivel alto

# Comunicaciones asíncronas

¿Qué es configurable en una comunicación asíncrona?

- Velocidad de transmisión
- Datos de 9 bits → Utilización de un bit de paridad
- Número de bits de parada: 1, 1.5, 2

{ Par  
Impar  
Mark  
Space

PIC 18F2525

## Configuración de datos:

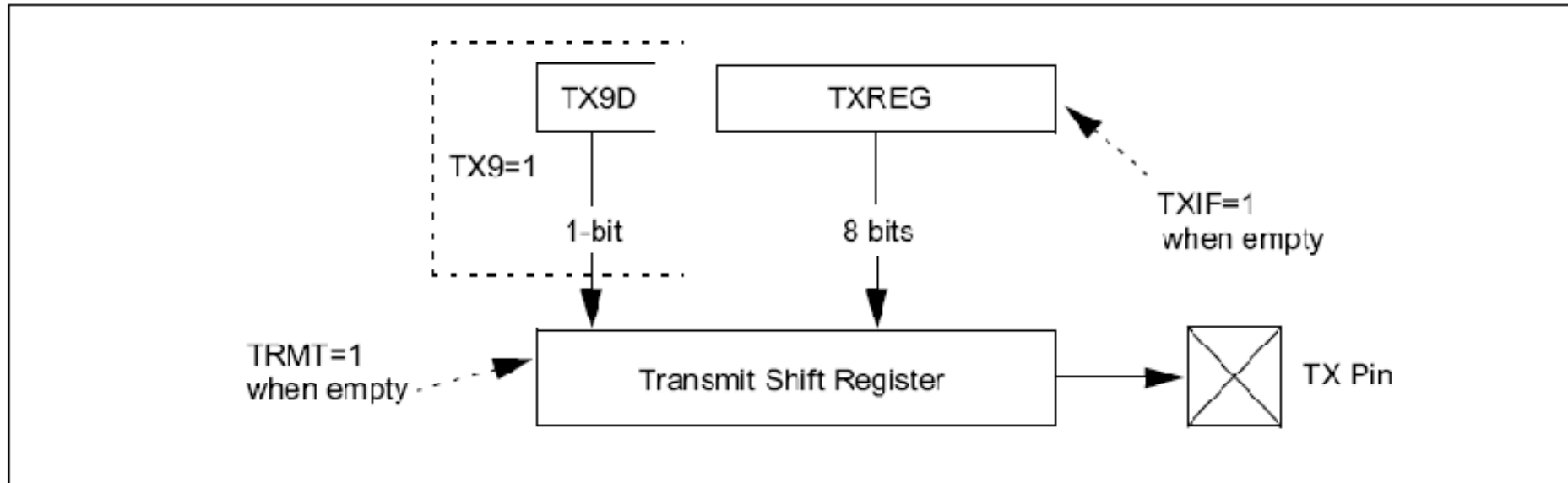
Datos: 8 bits	Bits parada: 1
Datos: 8 bits	Bits parada: 2
Datos: 9 bits	Bits parada: 1

*La paridad a utilizar depende del programador*

## Velocidad configurable:

- Registro SPBRG
- Bit BRGH: Bit de alta velocidad
- Bit BRG16: Bit de habilitación del SPBRG como registro de 16 bits

## Transmisión asíncrona. Funcionamiento



### Cuestiones de interés

Si la transmisión es de 9 bits, TX9D se debe escribir antes que TXREG

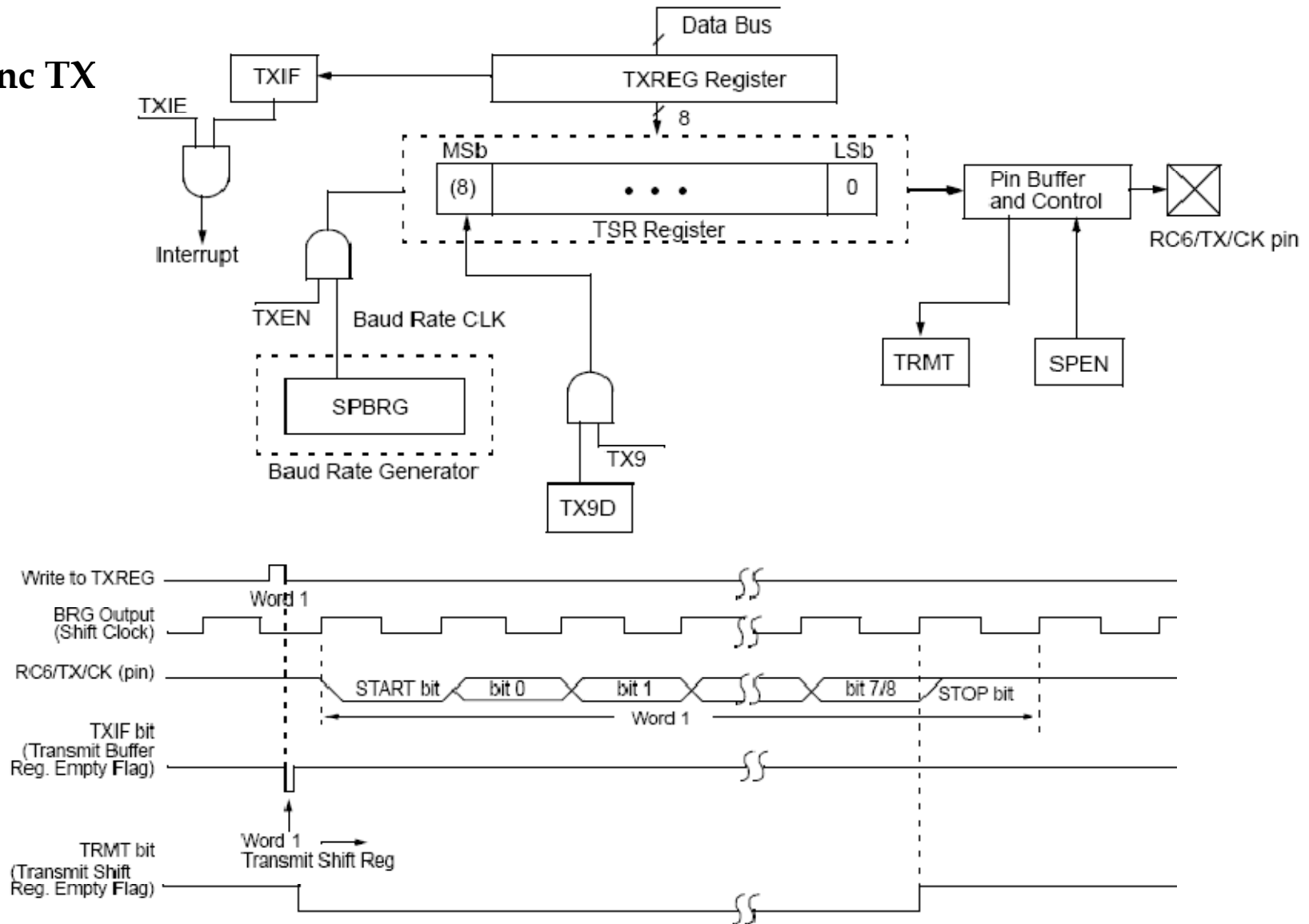
Se permite la escritura en TXREG cuando se está transmitiendo (FIFO de 2 bytes)

Hay un retraso de una instrucción entre escribir en TXREG y TXIF=0;

# Transmisión asíncrona. Funcionamiento

## Transmitter

### Asinc TX



# Transmisión asíncrona. Configuración

REGISTER 18-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

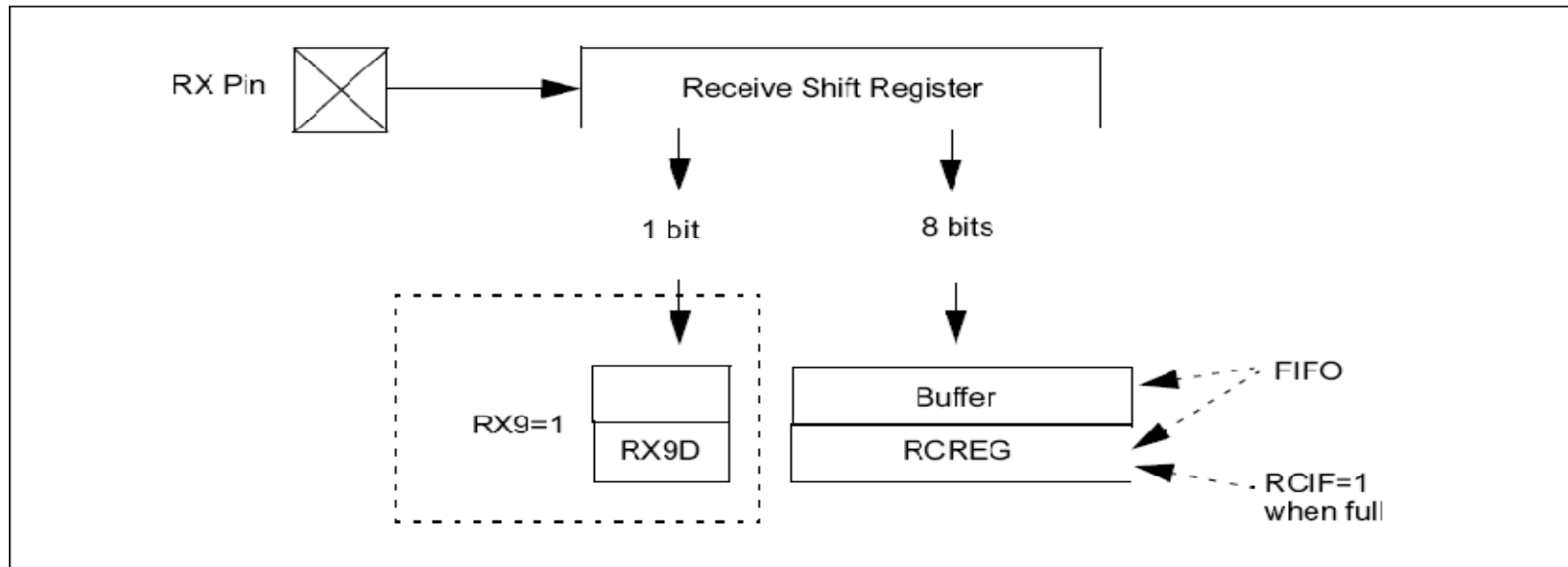
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
						bit 7	bit 0

- bit 7 **CSRC:** Clock Source Select bit  
Asynchronous mode:  
 Don't care.  
Synchronous mode:  
 1 = Master mode (clock generated internally from BRG)  
 0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit  
 1 = Selects 9-bit transmission  
 0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit  
 1 = Transmit enabled  
 0 = Transmit disabled  
**Note:** SREN/CREN overrides TXEN in Sync mode.
- bit 4 **SYNC:** EUSART Mode Select bit  
 1 = Synchronous mode  
 0 = Asynchronous mode
- bit 3 **SENDB:** Send Break Character bit  
Asynchronous mode:  
 1 = Send Sync Break on next transmission (cleared by hardware upon completion)  
 0 = Sync Break transmission completed  
Synchronous mode:  
 Don't care.
- bit 2 **BRGH:** High Baud Rate Select bit  
Asynchronous mode:  
 1 = High speed  
 0 = Low speed  
Synchronous mode:  
 Unused in this mode.
- bit 1 **TRMT:** Transmit Shift Register Status bit  
 1 = TSR empty  
 0 = TSR full
- bit 0 **TX9D:** 9th bit of Transmit Data  
 Can be address/data bit or a parity bit.

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown

## Recepción asíncrona. Funcionamiento



### Cuestiones de interés

El proceso de recepción empieza al recibir el bit de comienzo

Se comprueba la integridad del bit de parada

Se almacena el dato en la FIFO (1º byte recibido se almacena en RCREG)

Se permite un máximo de 2 bytes almacenados mientras se recibe un tercero

Si al terminar la recepción del tercero, la FIFO sigue llena → Error de Overrun

El bit RCIF permanece a 1 hasta que se leen todos los datos de la FIFO





# Recepción asíncrona. Configuración

REGISTER 18-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x	
SPEN	RX9	SREN	CREN	ADDEN	FERR	OEPR	RX9D	
bit 7								bit 0

- bit 7 SPEN: Serial Port Enable bit**  
 1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)  
 0 = Serial port disabled (held in Reset)
- bit 6 RX9: 9-bit Receive Enable bit**  
 1 = Selects 9-bit reception  
 0 = Selects 8-bit reception
- bit 5 SREN: Single Receive Enable bit**  
Asynchronous mode:  
 Don't care.  
Synchronous mode – Master:  
 1 = Enables single receive  
 0 = Disables single receive  
 This bit is cleared after reception is complete  
Synchronous mode – Slave:  
 Don't care.
- bit 4 CREN: Continuous Receive Enable bit**  
Asynchronous mode:  
 1 = Enables receiver  
 0 = Disables receiver  
Synchronous mode:  
 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)  
 0 = Disables continuous receive
- bit 3 ADDEN: Address Detect Enable bit**  
Asynchronous mode 9-bit (RX9 = 1):  
 1 = Enables address detection, enables interrupt and loads the receive buffer when RD<0> is set  
 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit  
Asynchronous mode 8-bit (RX9 = 0):  
 Don't care.
- bit 2 FERR: Framing Error bit**  
 1 = Framing error (can be updated by reading RCREG register and receiving next valid byte)  
 0 = No framing error
- bit 1 OEPR: Overrun Error bit**  
 1 = Overrun error (can be cleared by clearing bit OREN)  
 0 = No overrun error
- bit 0 RX9D: 9th bit of Received Data**  
 This can be address/data bit or a parity bit and must be calculated by user firmware.

<b>Legend:</b>		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	1' = Bit is set	'0' = Bit is cleared    z = Bit is unknown

# Velocidad de transmisión

TABLE 18-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64(n+1)]$
0	0	1	8-bit/Asynchronous	$F_{osc}/[16(n+1)]$
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	$F_{osc}/[4(n+1)]$
1	0	x	0-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGH:SPBRG register pair

Depende del registro SPBRG y de los bits BRG16 y BRGH

**Ejemplo: 16 MHz oscillator, 9600 baudios**

**BRG16=0**

BRGH = 1

$SPBRG = 16000000 / (16 \times 9600) - 1 = 51.08 \rightarrow 51 \rightarrow$  Baud Rate= 9615.33 baud

BRGH = 0

$SPBRG = 16000000 / (64 \times 9600) - 1 = 12.02 \rightarrow 12 \rightarrow$  Baud Rate= 9615.38 baud

**BRG16=1**

BRGH = 0

$SPBRG = 16000000 / (16 \times 9600) - 1 = 51.08 \rightarrow 51 \rightarrow$  Baud Rate= 9615.33 baud

BRGH = 1

$SPBRG = 16000000 / (4 \times 9600) - 1 = 415.67 \rightarrow 416 \rightarrow$  Baud Rate= 9592.33 baud



# Registro BAUDCON

REGISTER 18.3: BAUDCON: BAUD RATE CONTROL REGISTER

R/W-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	
bit 7								bit 0

- bit 7 **ABDOVF**: Auto-Baud Acquisition Rollover Status bit  
 1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software)  
 0 = No BRG rollover has occurred
- bit 6 **RCIDL**: Receive Operation Idle Status bit  
 1 = Receive operation is Idle  
 0 = Receive operation is active
- bit 5 **Unimplemented**: Read as '0'
- bit 4 **SCKP**: Synchronous Clock Polarity Select bit  
Asynchronous mode:  
 Unused in this mode.  
Synchronous mode:  
 1 = Idle state for clock (CK) is a high level  
 0 = Idle state for clock (CK) is a low level
- bit 3 **BRG16**: 16-bit Baud Rate Register Enable bit  
 1 = 16-bit Baud Rate Generator – SPBRGH and SPBRG  
 0 = 8-bit Baud Rate Generator – SPBRG only (Compatible mode), SPBRGH value ignored
- bit 2 **Unimplemented**: Read as '0'
- bit 1 **WUE**: Wake-up Enable bit  
Asynchronous mode:  
 1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge  
 0 = RX pin not monitored or rising edge detected  
Synchronous mode:  
 Unused in this mode.
- bit 0 **ABDEN**: Auto-Baud Detect Enable bit  
Asynchronous mode:  
 1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion  
 0 = Baud rate measurement disabled or completed  
Synchronous mode:  
 Unused in this mode.

<b>Legend:</b>			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

# Inicialización

Configurar  
PIC para USART

- Bit SPEN (RCSTA<7>) activo
- Bit TRISC<7> activo
- Bit TRISC<6> activo

Configurar  
velocidad

- Registro SPBRG
- Bit BRGH
- Bit BRG16

Configurar TX/RX

- Registro TXSTA
- Registro RCSTA

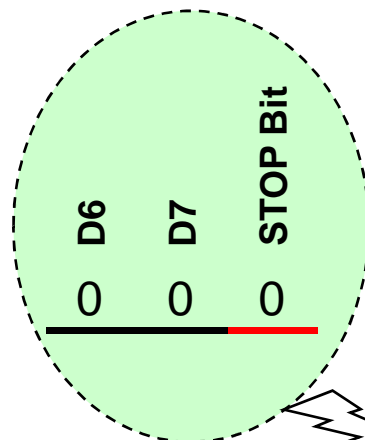
Configurar  
Interrupciones

- Bit IPEN=1
- Registro INTCON
- Registro PIE1
- Registro PIR1
- Registro IPR1

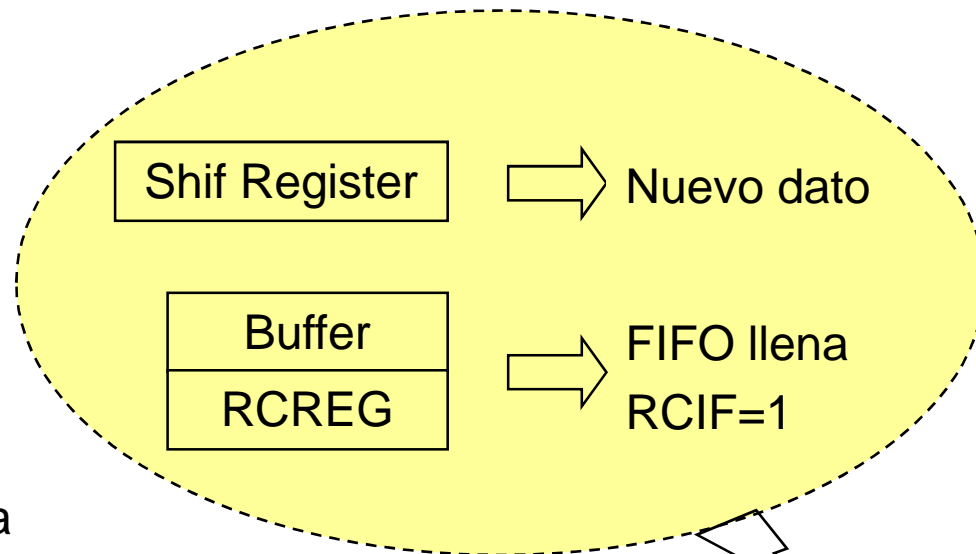
## Detección de errores

La USART del PIC puede detectar dos tipos de errores:

- Error de trama
- Error de desbordamiento



Error de trama  
FERR=1



Error de desbordamiento  
OERR=1

Más errores se pueden detectar por software:

- Bit de paridad
- CRC
- Checksum

RS232

# RS232 Standard

## Origen del Standard RS232

Diseñado para comunicar un sistema DTE con un sistema DCE.

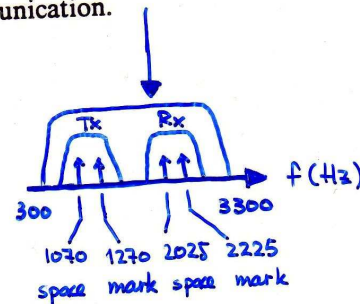
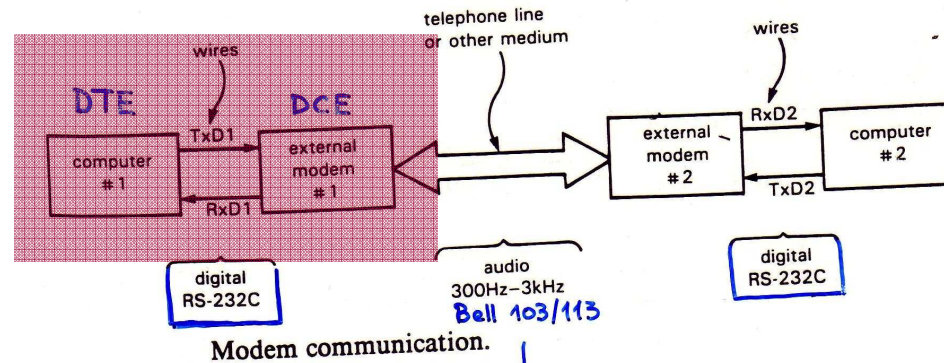
DTE (Data Terminal Equipment)

usualmente, un Ordenador

DCE (Data Communications Equipment)

usualmente, un módem

Empleando dos líneas de datos, TX (Transmisión) y RX (Recepción)



[Visto desde modem que origina la llamada.]

## *RS232 standard*

### *Señales del Standard RS232*

<b>Nombre</b>	<b>Dirección DTE ↔ DCE</b>	<b>Función</b>	<b>Comentario</b>
TD	⇒	Transmitted data	Par de Datos
RD	⇐	Received Data	
RTS	⇒	Request to Send	Par de Handshake
CTS	⇐	Clear to Send	
DTR	⇒	Data Terminal Ready	Par de Handshake
DSR	⇐	Data Set Ready	
DCD	⇐	Data Carrier Detect	Habilitan DTE
RI	⇐	Ring Indicator	

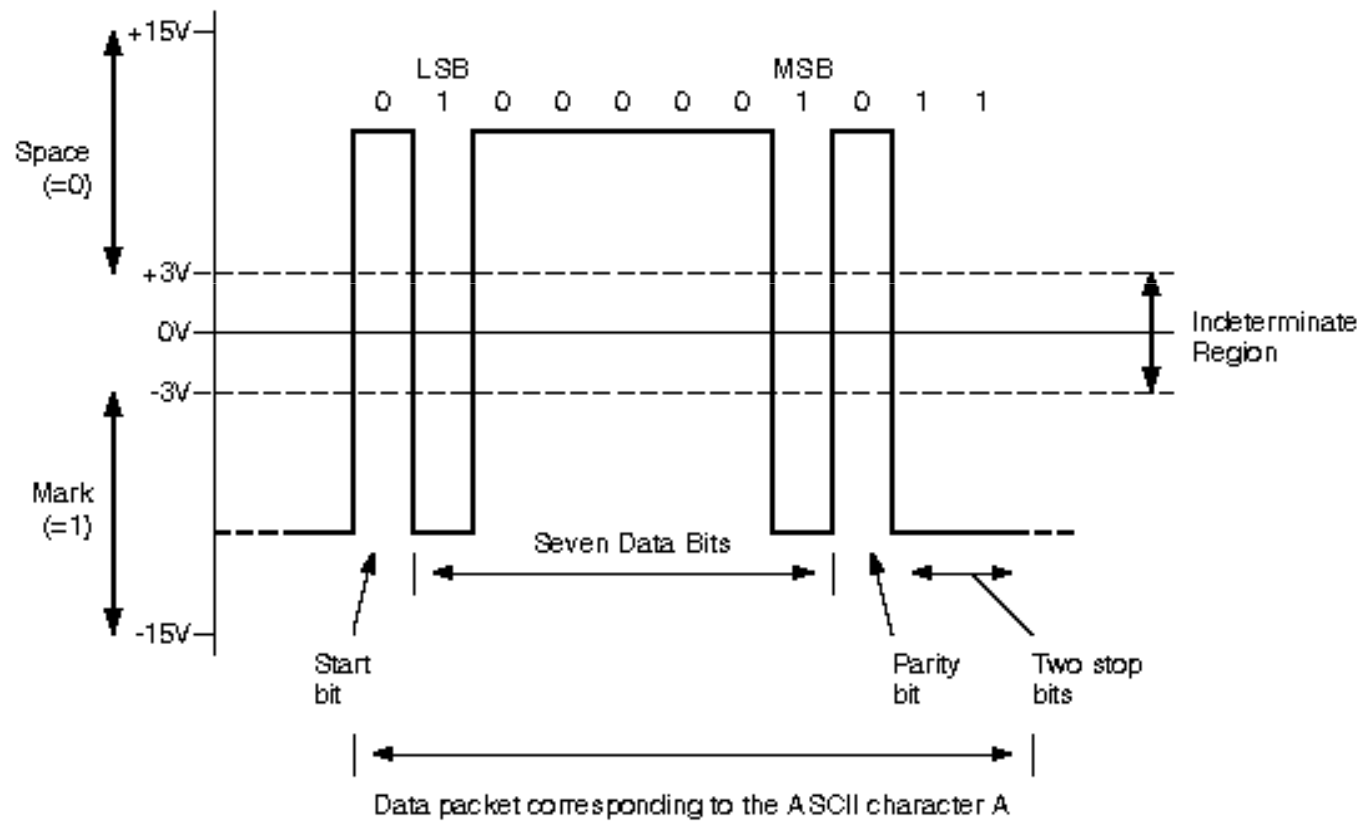


## RS232 standard

SPECIFICATIONS		RS232	RS423
Mode of Operation		SINGLE -ENDED	SINGLE -ENDED
Total Number of Drivers and Receivers on One Line		1 DRIVER 1 RECVR	1 DRIVER 10 RECVR
Maximum Cable Length		50 FT.	4000 FT.
Maximum Data Rate		20kb/s	100kb/s
Maximum Driver Output Voltage		+/-25V	+/-6V
Driver Output Signal Level (Loaded Min.)	Loaded	+/-5V to +/-15V	+/-3.6V
Driver Output Signal Level (Unloaded Max)	Unloaded	+/-25V	+/-6V
Driver Load Impedance (Ohms)		3k to 7k	>=450
Max. Driver Current in High Z State	Power On	N/A	N/A
Max. Driver Current in High Z State	Power Off	+/-6mA @ +/-2v	+/-100uA
Slew Rate (Max.)		30V/uS	Adjustable
Receiver Input Voltage Range		+/-15V	+/-12V
Receiver Input Sensitivity		+/-3V	+/-200mV
Receiver Input Resistance (Ohms)		3k to 7k	4k min.

# RS232 standard

## Señales en las líneas RS232



# RS232 standard

## Circuitos Conversores de Nivel

TTL a RS232

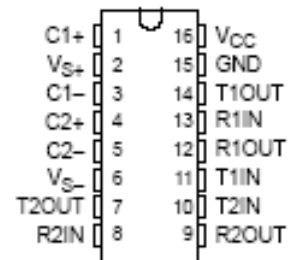
RS232 a TTL

### MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

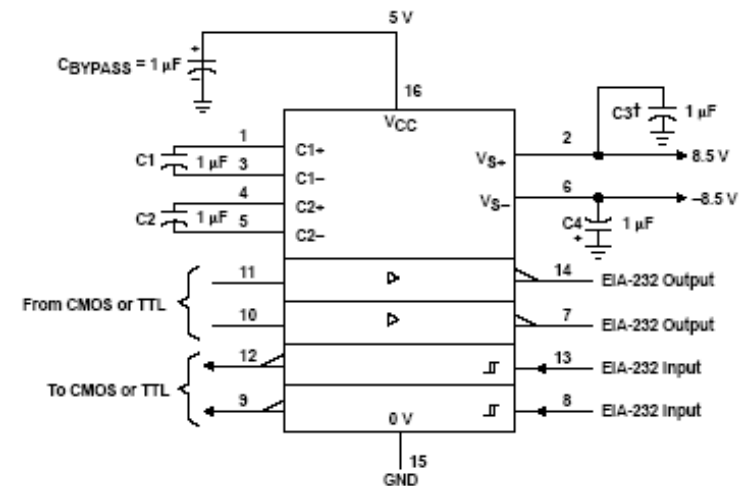
SLLS0471 - FEBRUARY 1989 - REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30$ -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 - 2000-V Human-Body Model (A114-A)
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE  
MAX232I . . . D, DW, OR N PACKAGE  
(TOP VIEW)

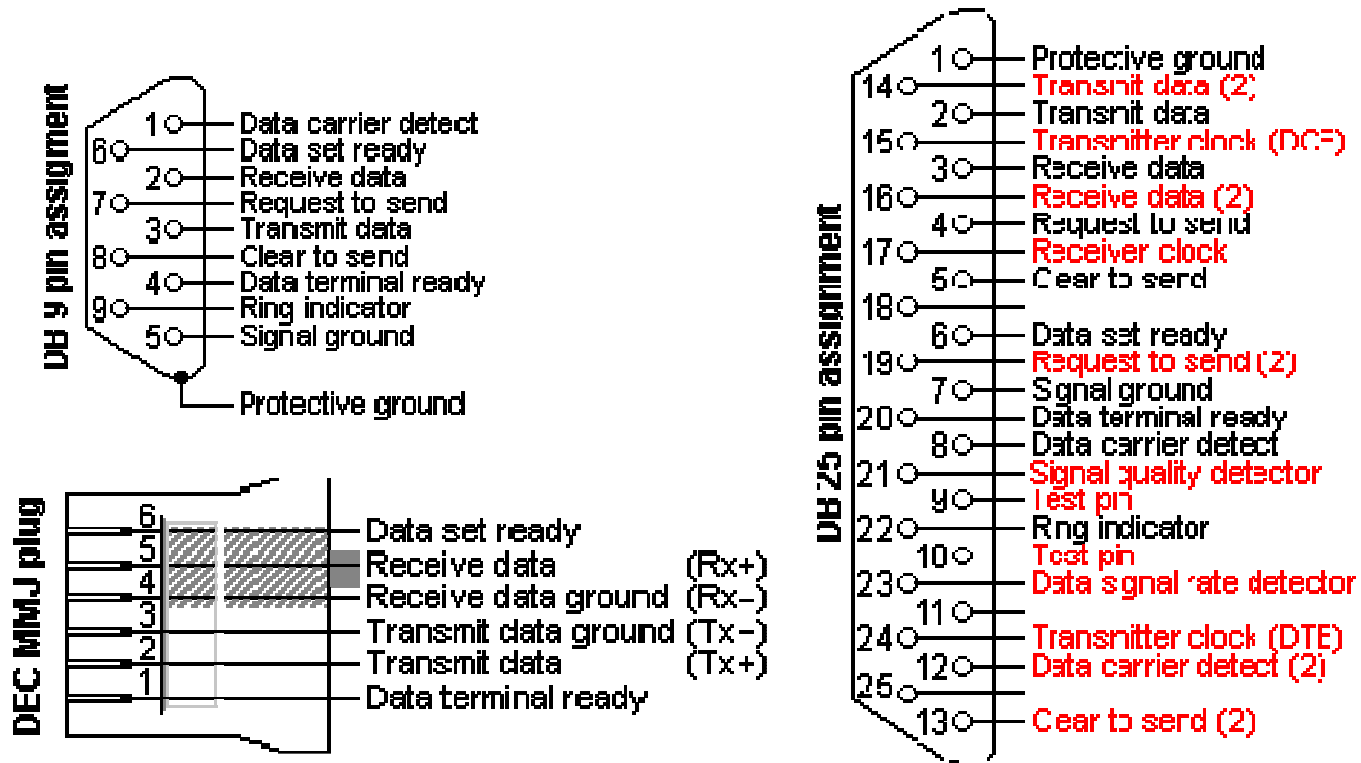


#### APPLICATION INFORMATION



# RS232 standard

## Conectores habituales en RS232



## RS232 standard

### Conexión Null-Modem

Para evitar las señales de sincronismo, y conectar dos dispositivos a través del par Tx y Rx

