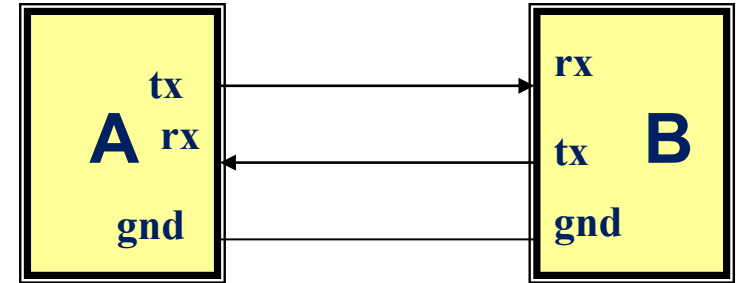
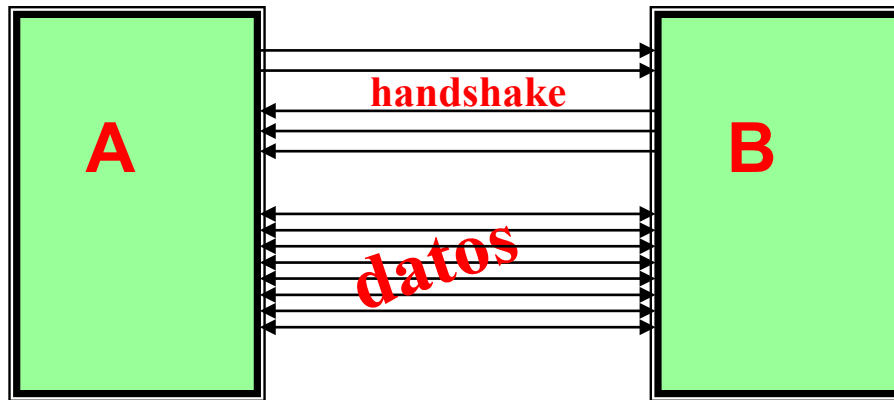


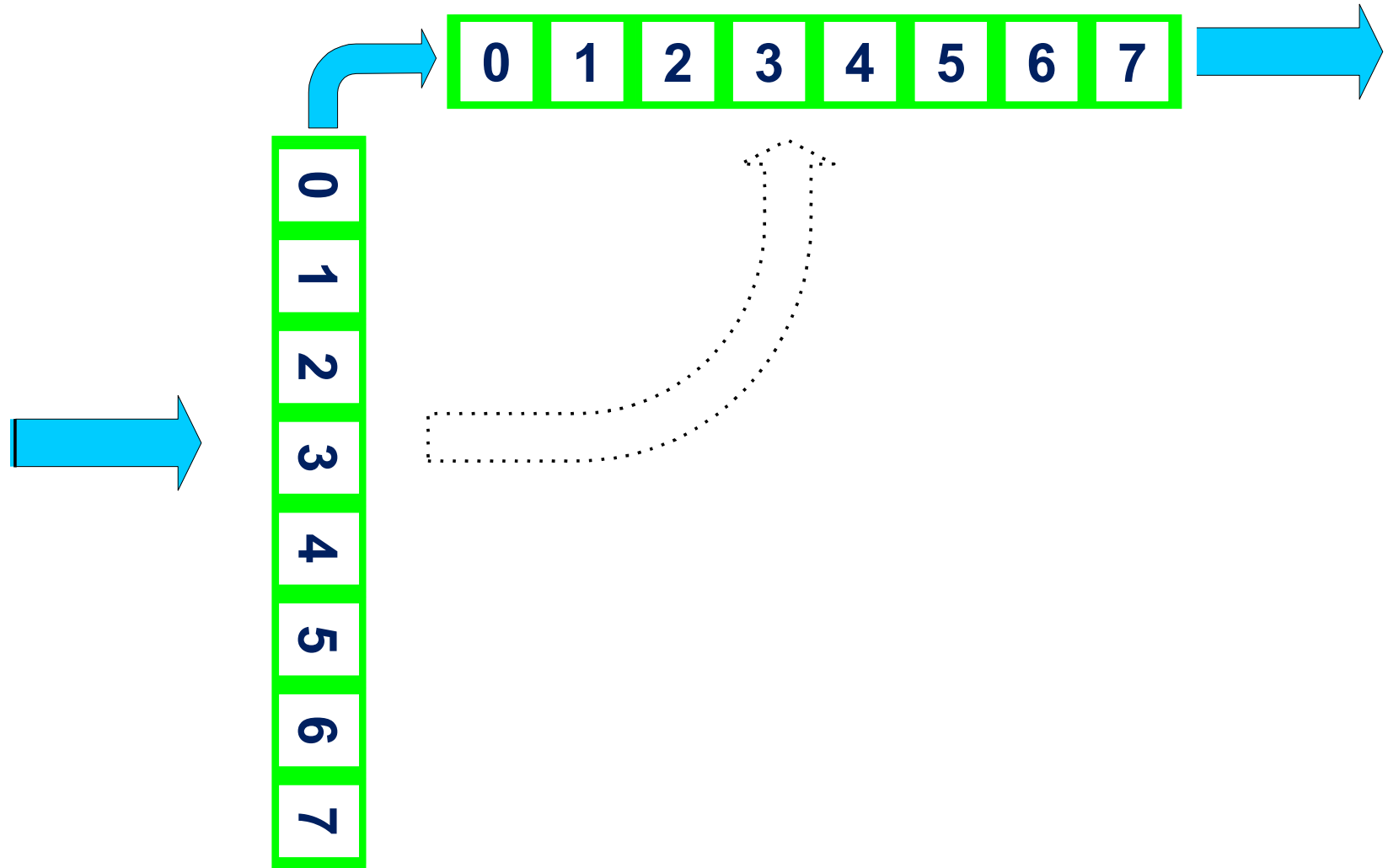
Tipos de comunicación:CARACTERÍSTICAS



- ❑ Gran velocidad.
- ❑ Distancias cortas. No se garantiza el tiempo de transito.
- ❑ Muchas líneas de interconexión.
- ❑ El dato esta presente en un bit por cada línea de datos.
- ❑ Volumen de software para la atención del sincronismo.

- ❖ Velocidad Inferior.
- ❖ Distancias grandes.
- ❖ Cantidad reducida de líneas de interconexión (mínimo tres).
- ❖ El dato se transmite bit por bit.
- ❖ ASINCRÓNICA o Sincrónica.

Comunicación Serie: Principios



Tipos de Comunicación Serie

Simplex

El envío de información es en un único sentido. El transmisor y el receptor están bien identificados. Se utiliza un solo canal de datos.



Half-Duplex

El envío de información es bidireccional pero multiplexada en el tiempo.

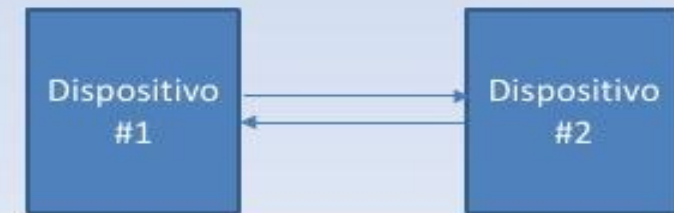
Como se muestra en el esquema, primero se transmite en un sentido (1) y luego en el otro sentido (2).

Se utiliza un solo canal de datos.



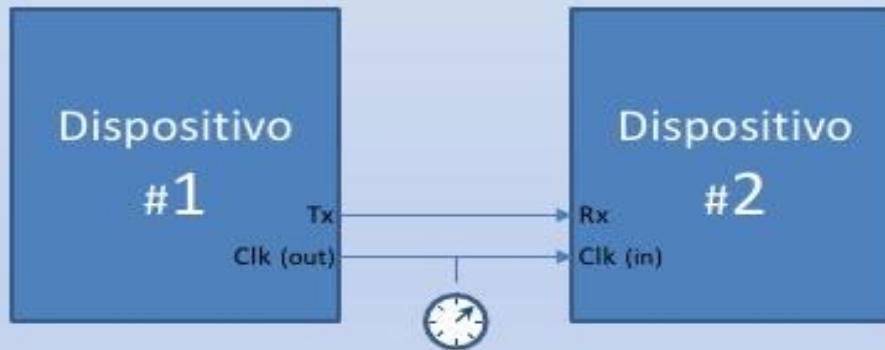
Full-Duplex

El envío de información es bidireccional simultáneo. Se utilizan dos canales de datos.



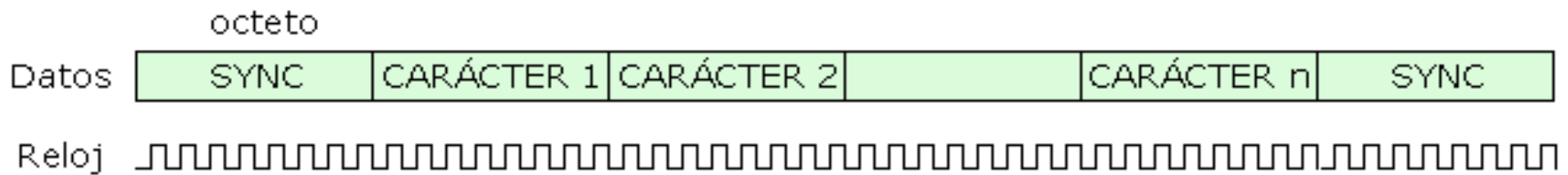
Comunicación serie síncrona

Comunicación Serie Síncronica



- Es necesario una señal de Clock común a ambos dispositivos.
- Es necesario un hilo extra para sincronizar la comunicación.

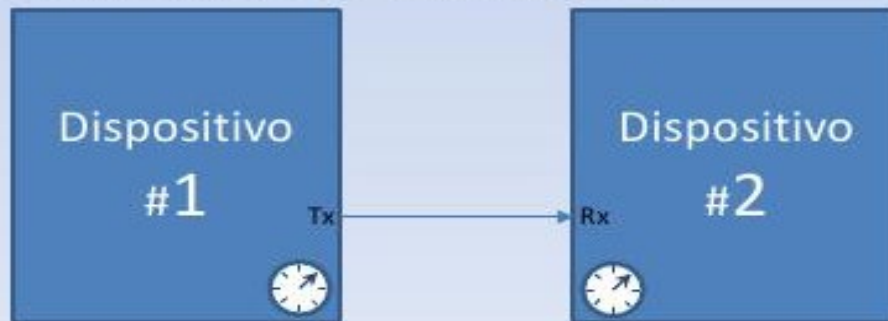
El clock es una línea física adicional o está de algún modo embebido en la transmisión de datos.



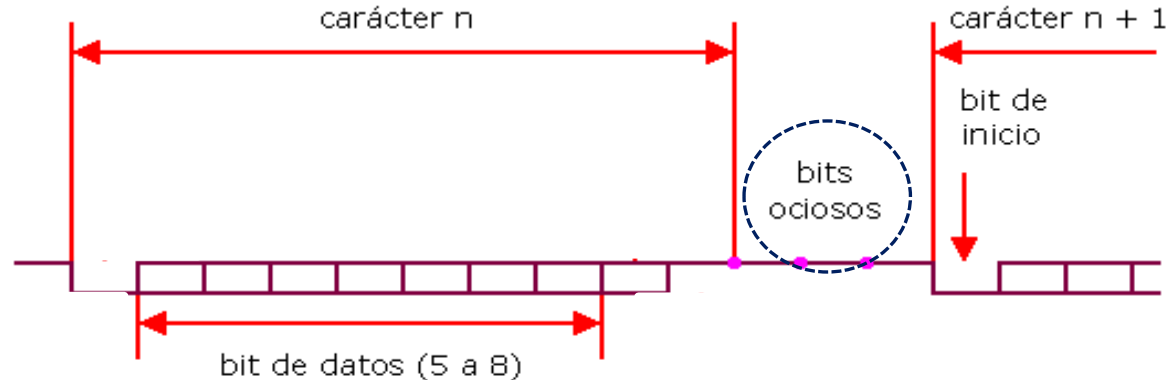
Transmisión síncrona

Comunicación serie asincrónica

Comunicación Serie Asincrónica

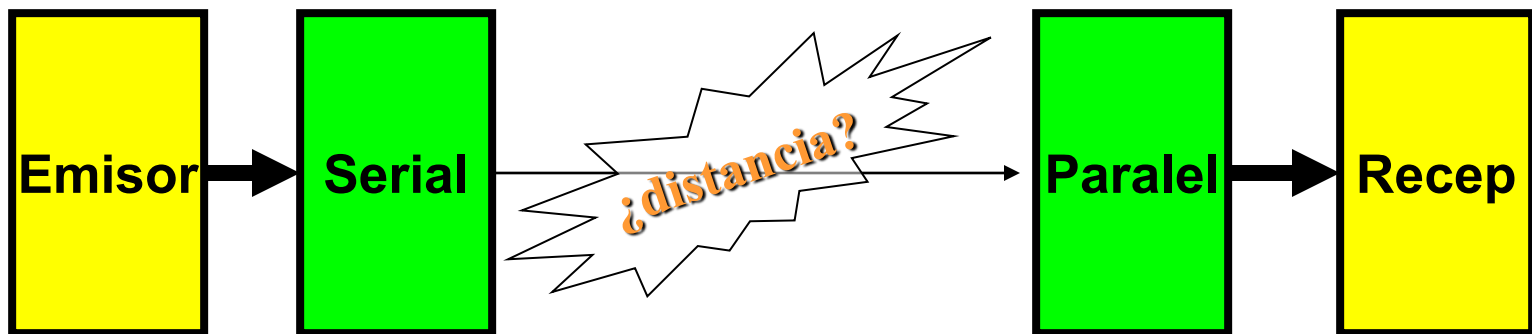


- Cada dispositivo posee su propia señal de clock.
- Los dispositivos deben conocer la velocidad de transmisión.
- La transmisión se sincroniza mediante los bits de Start y Stop.

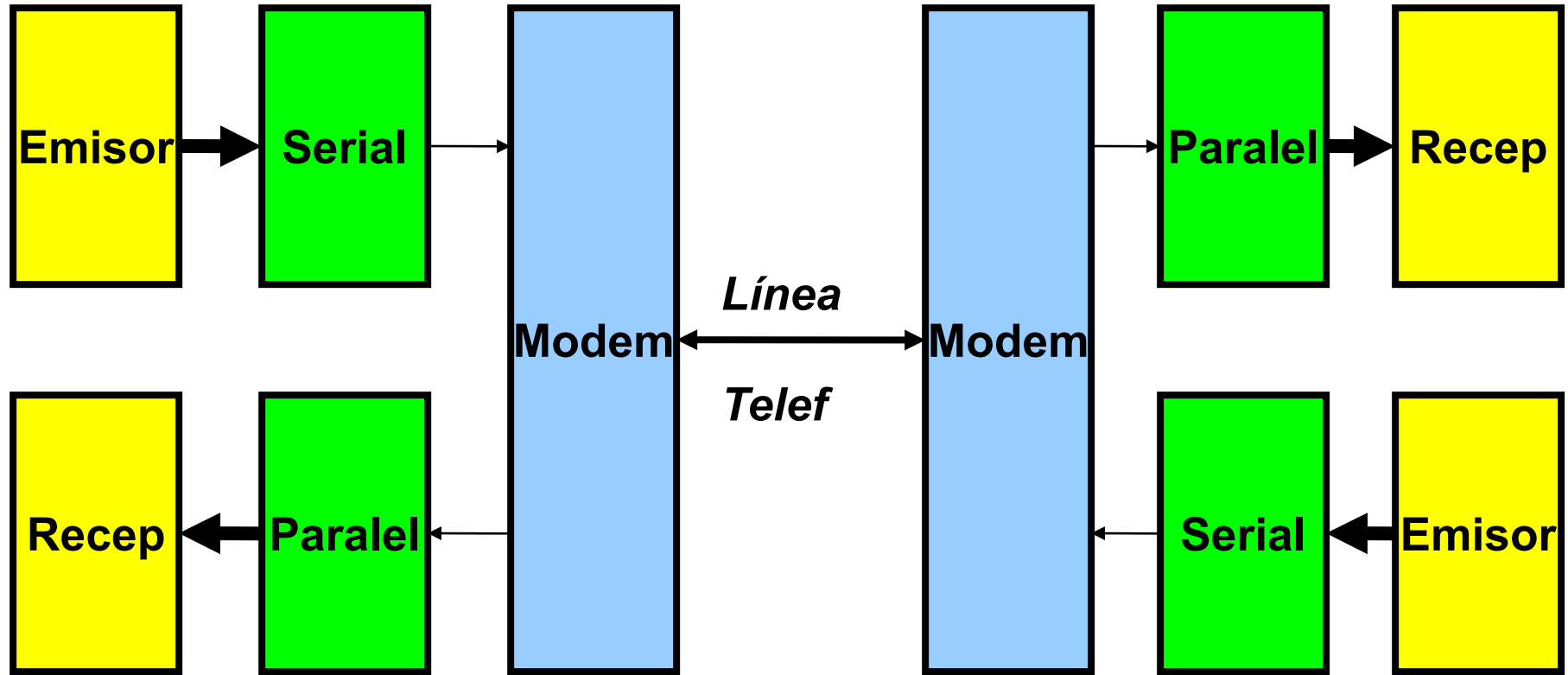


Transmisión asincrónica con velocidad menor que la máxima posible

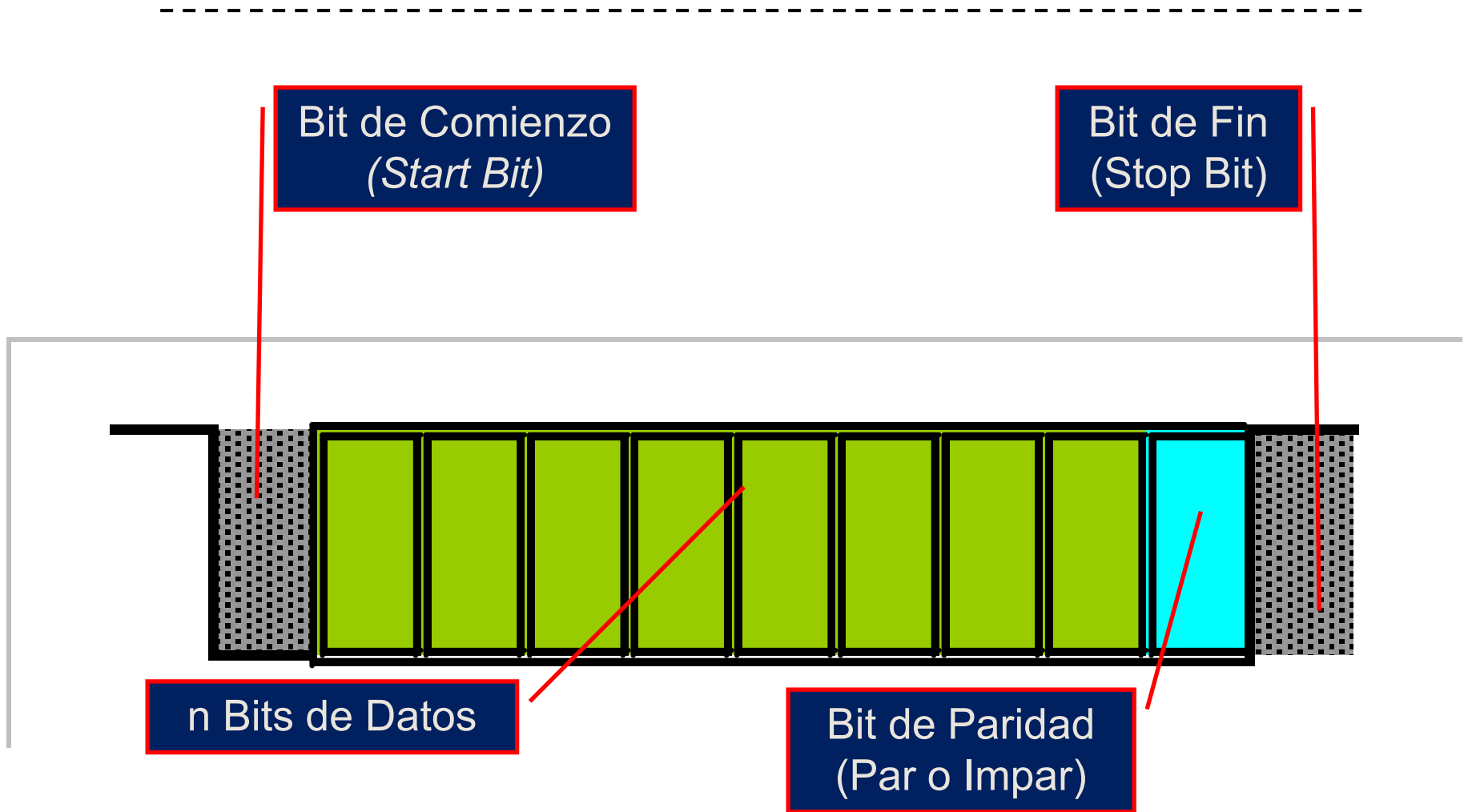
Comunicación Serie: Principios



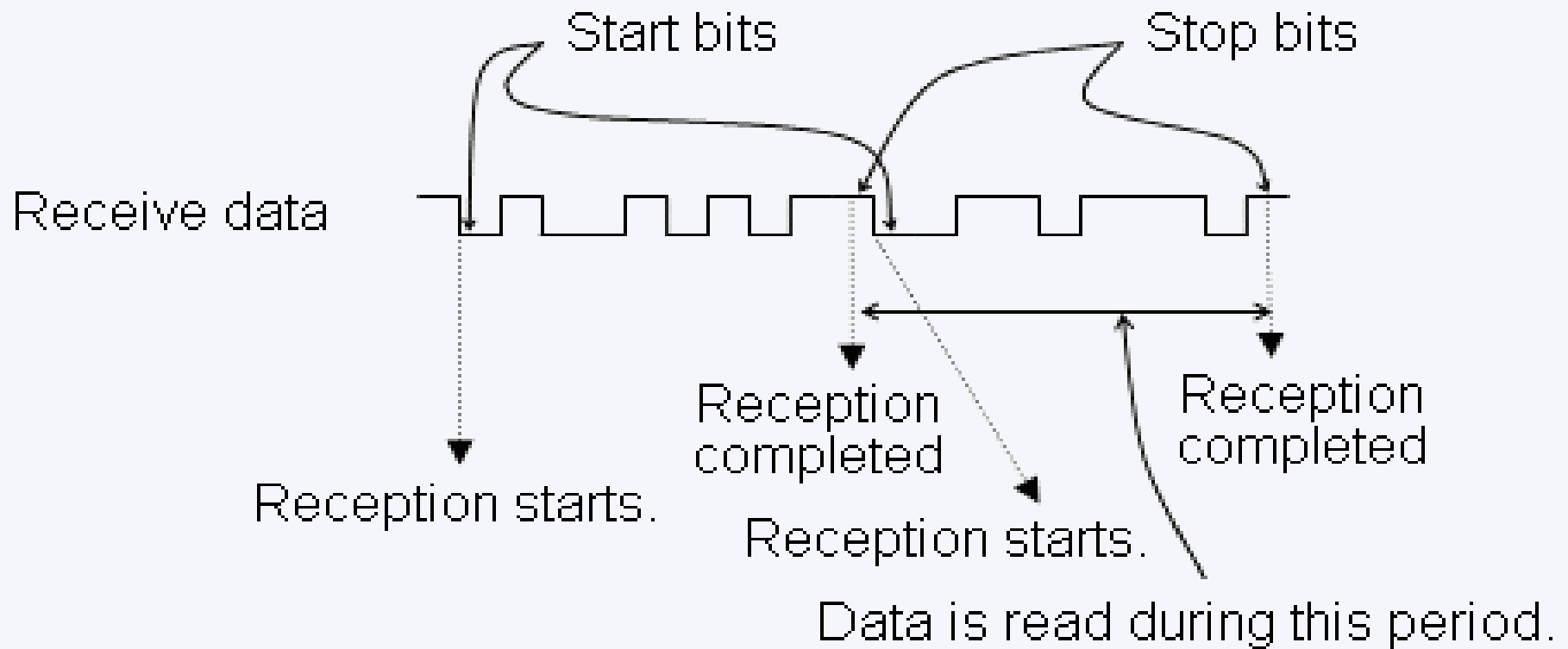
Comunicación Serie: Principios



Trama asincrónica



Entendiendo la trama asincrónica



Errores en Com. Asincrónica

- **Error de paridad:** cuando se detecta un error en el bit de paridad.
- **Error de trama:** Cuando se espera el bit de parada y en su lugar se detecta un nivel bajo, implica que se perdió la sincronización.
- **Error de sobre escritura:** Cuando un símbolo previo no se leyó y es reemplazado por un nuevo símbolo recibido.
- **Error de break:** Cuando la línea de Rx se queda en estado bajo durante un tiempo $>$ al de un carácter luego de una trama válida.

Normas de estandarización

Son emitidas por diferentes organismos internacionales:

EIA (Electronic Industry Association (normas RS)

ANSI(American national Standard Industry) (normas X)

ISO (International Standard organization) (normas ISO)

FCC (Federal Communication Committe) (normas FCC)

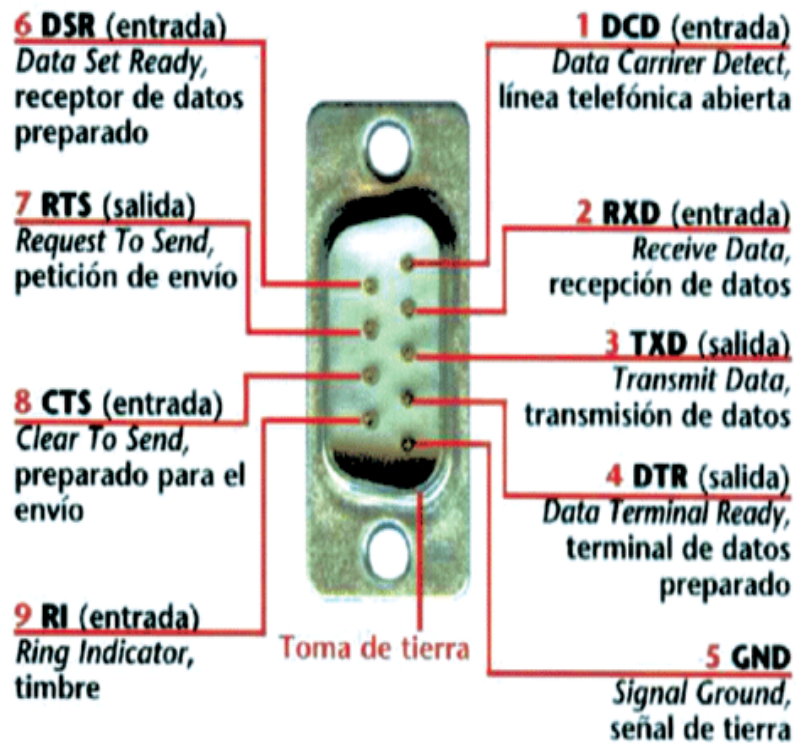
ITU (International Telecommunication Union))(normas V)

Estas normas especifican:

- **Eléctricas:** Niveles de tensión, impedancias de salida en los transmisores, impedancia de entrada en los receptores, impedancia de línea, etc.
- **Mecánicas:** Características de los conectores y de los cables.
- **Lógicas:** Nombre de las señales, funciones de cada señal, asignación de los pines de los conectores, etc.

Conectores estándar

Asignación de pins en conectores DB9



Asignación de pins en conectores DB25



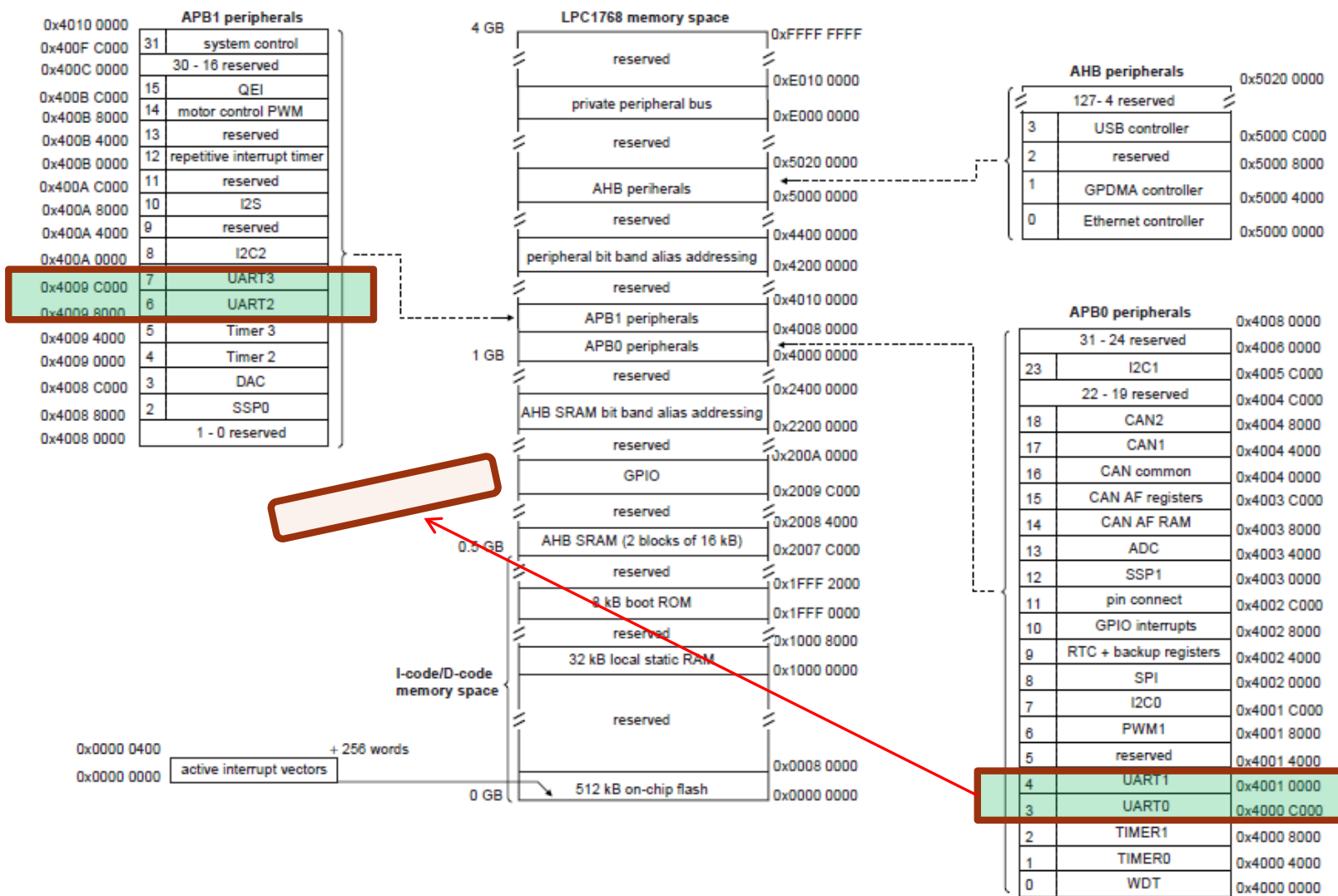
Presentamos la UART

Universal Asynchronous Receiver Transmitter

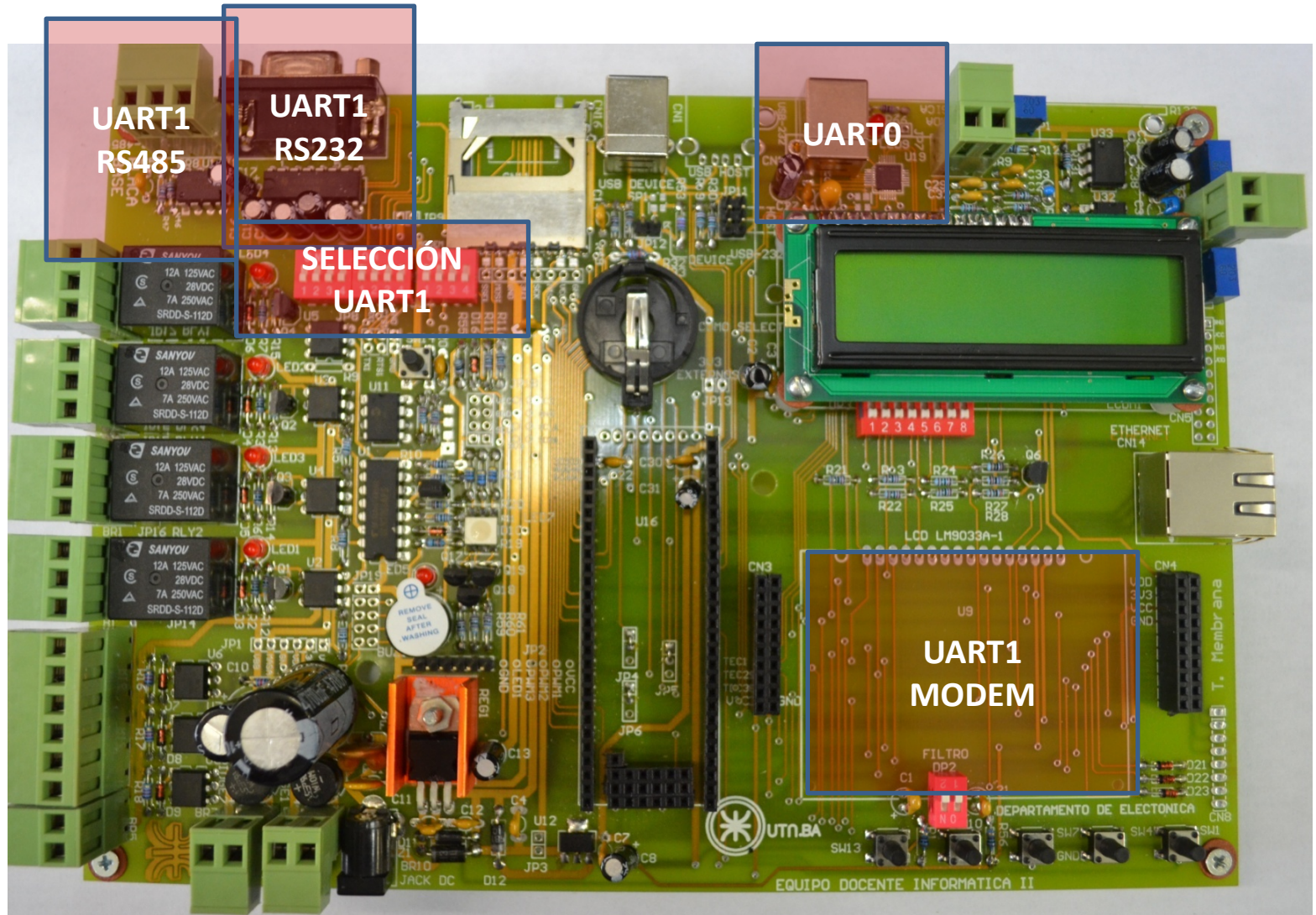
Los diseñadores de la PC incluyeron un dispositivo de HW que le resolviera al μ p la elaboración de la “trama” y la comunicación serie en general: El chip **8250** (luego 16550/16650 y 16750)

Mas de 30 años después de su introducción en el mercado, el LPC1769 incluye un dispositivo similar.

¿Dónde están mapeadas las UART del LPC1769?



INFOTRONIC – PUERTOS SERIE



Prestaciones en el LPC1769

➤ Características comunes

- *Datos de 5, 6, 7, y 8 bits.*
- *Generación y verificación de paridad: impar, par o sin paridad.*
- *Uno o dos bits de parada.*
- *FIFO de 16 bytes para transmisión y recepción.*
- *Generador de velocidad de transmisión (divisor de frecuencia fraccional)*
- *Soporte de DMA en la transmisión y/o en la recepción .*
- *Capacidad de Auto-baud*
- *Generación y detección de Break.*
- *Soporte para software con control de flujo*
- *Modo de direccionamiento Multiprocesador.*

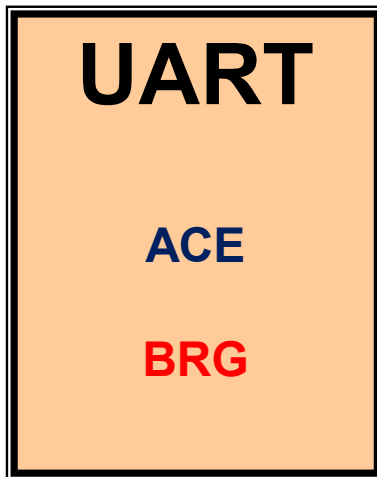
➤ Solo para las UART0,2,3

- *Modo IrDA para dar soporte a comunicación infrarroja*

➤ Solo para la UART1

- *Soporte RS-485.*
- *Líneas para control de módem*

La UART tipo 16550: Generalidades

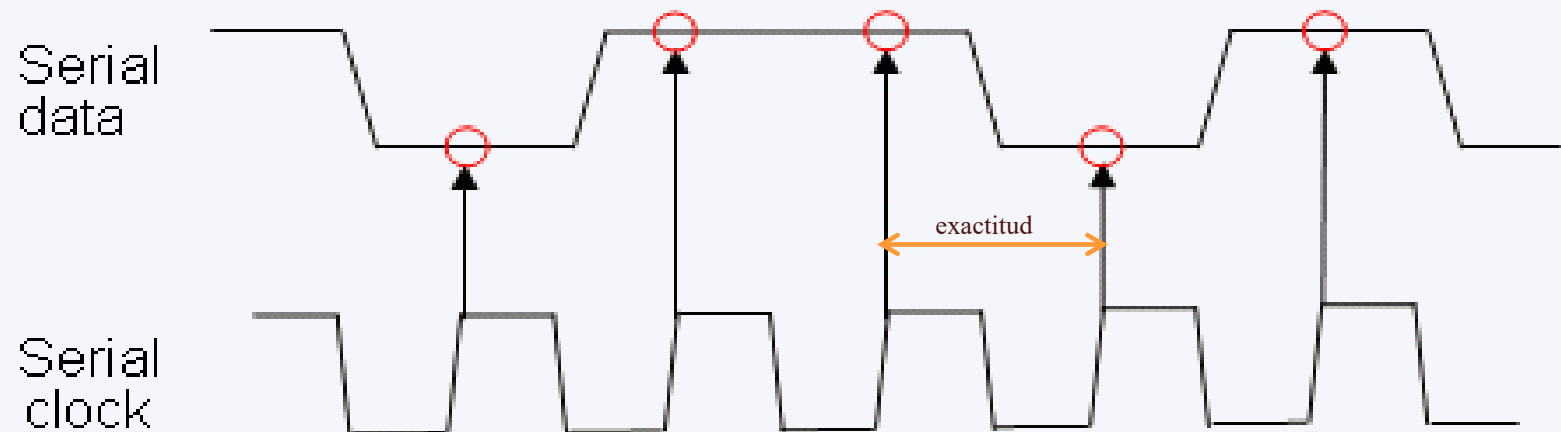
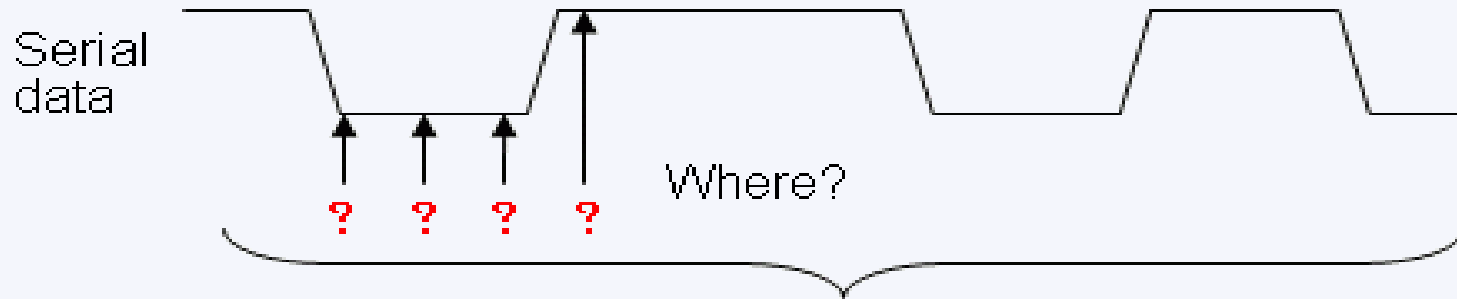


Se ocupa del armado de la trama asincrónica.

- **ACE: Asynchronous Communication Element**
- **BRG: Baud Rate Generator**

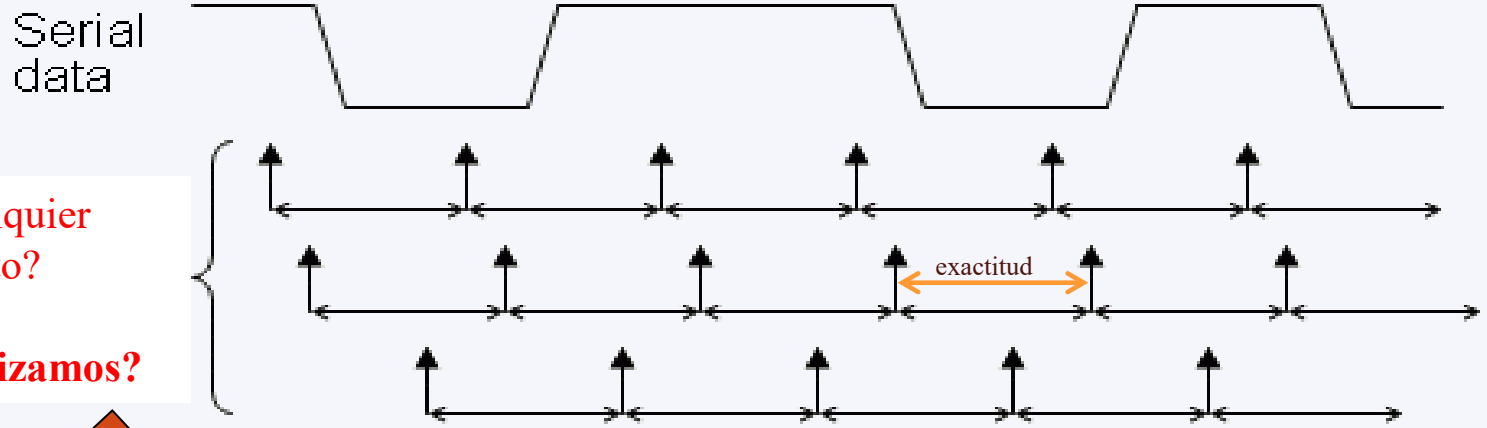
Se ocupa de generar la velocidad de Tx.
(baudrate)

La UART: ¿para qué el clock?

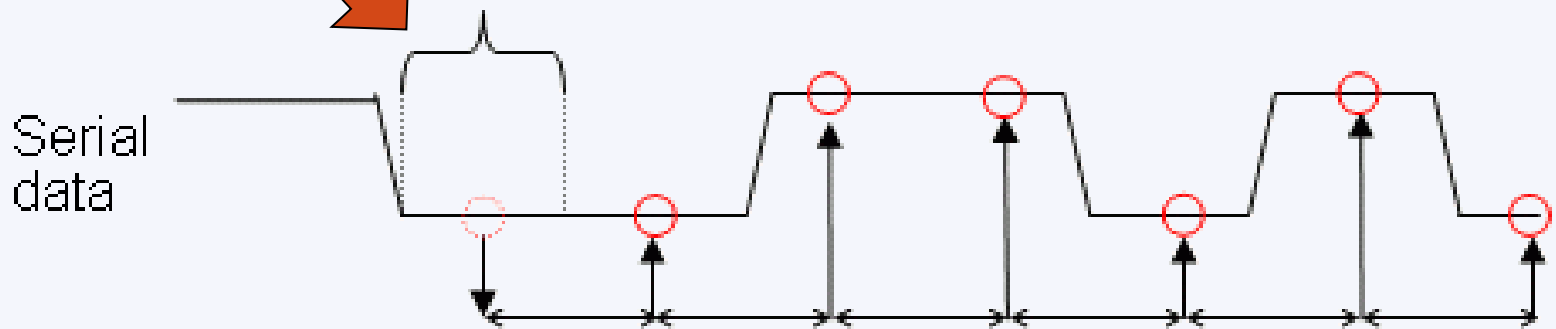


La UART: ¿Cómo nos sincronizamos?

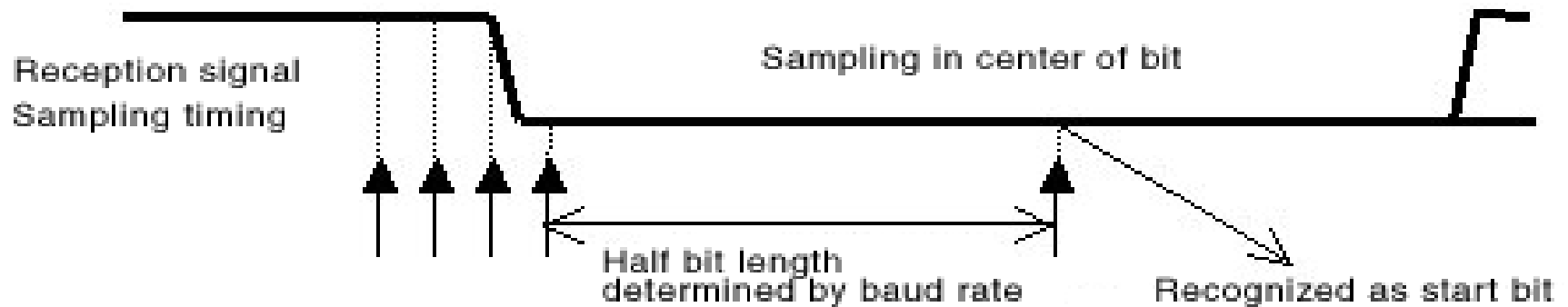
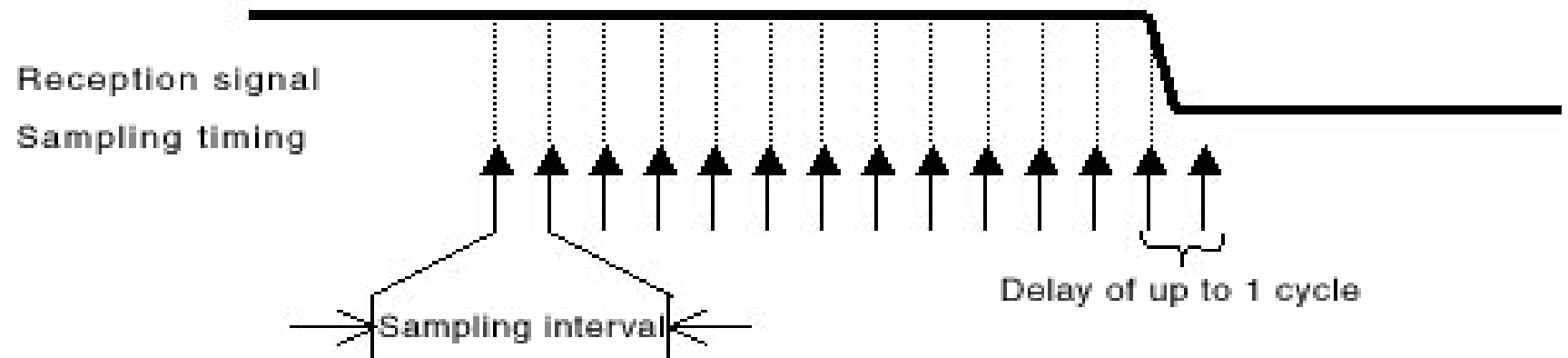
¿en cualquier momento?
¿Cómo sincronizamos?



Hasta aquí, el intervalo es conocido (viene dado por la frecuencia del clock), pero el “timing” NO.

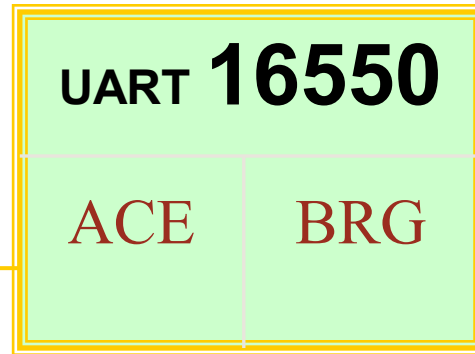


¿porqué evaluar en el centro del bit?



La UART: el BRG (Baud Rate Generator)

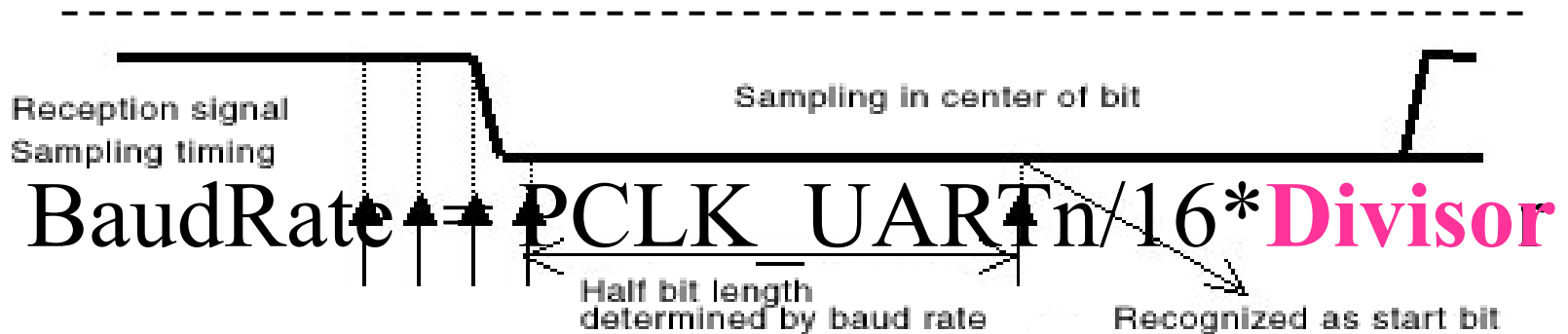
N bits/seg



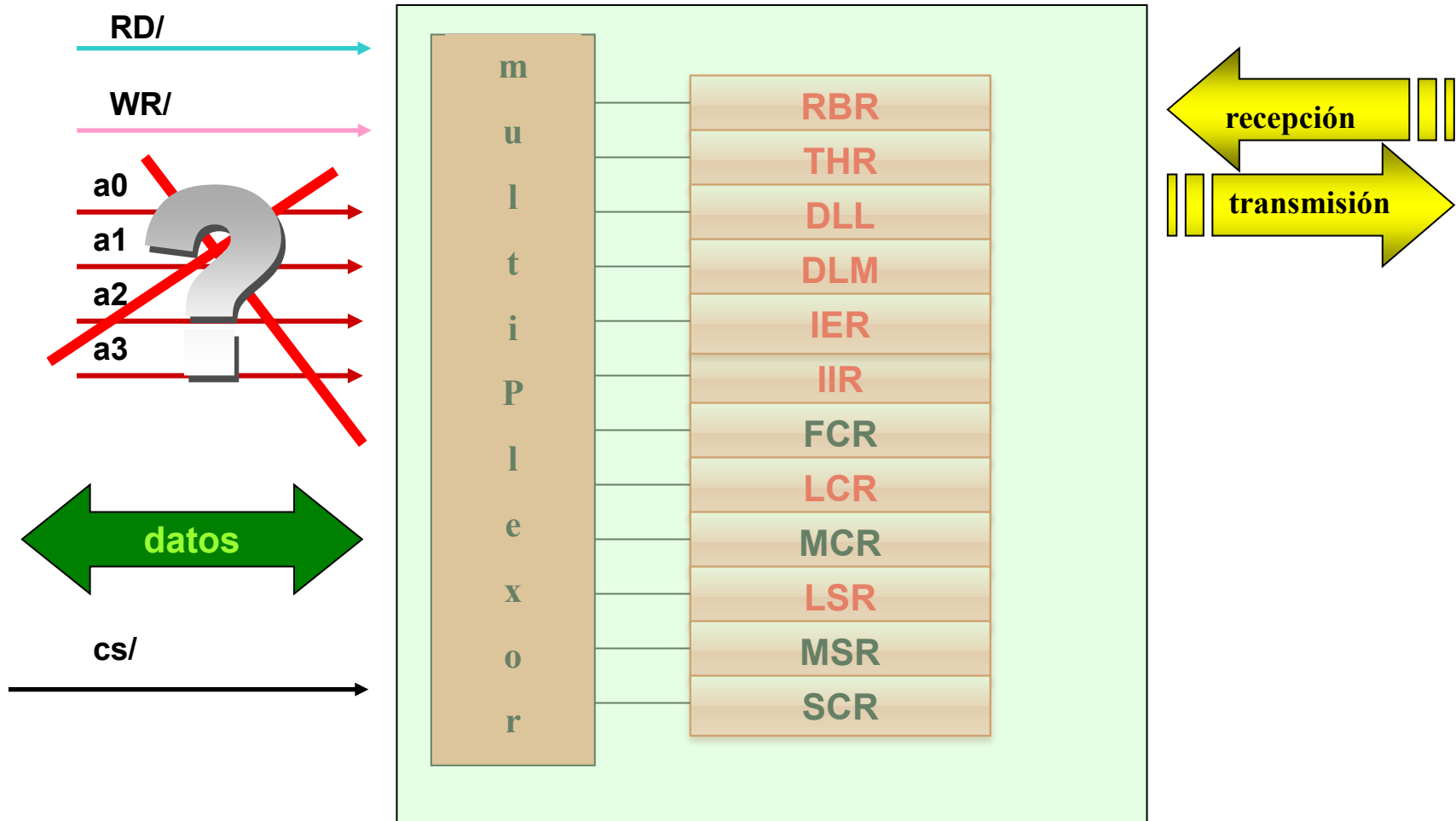
Si CCLK=100Mhz y
PLKSEL0 y 1 = 00

PCLK_UARTn
25 MHz

El BRG divide por defecto PCLK_UARTn por un divisor programable y por 16, con el objeto de sobremuestrear la señal recibida.

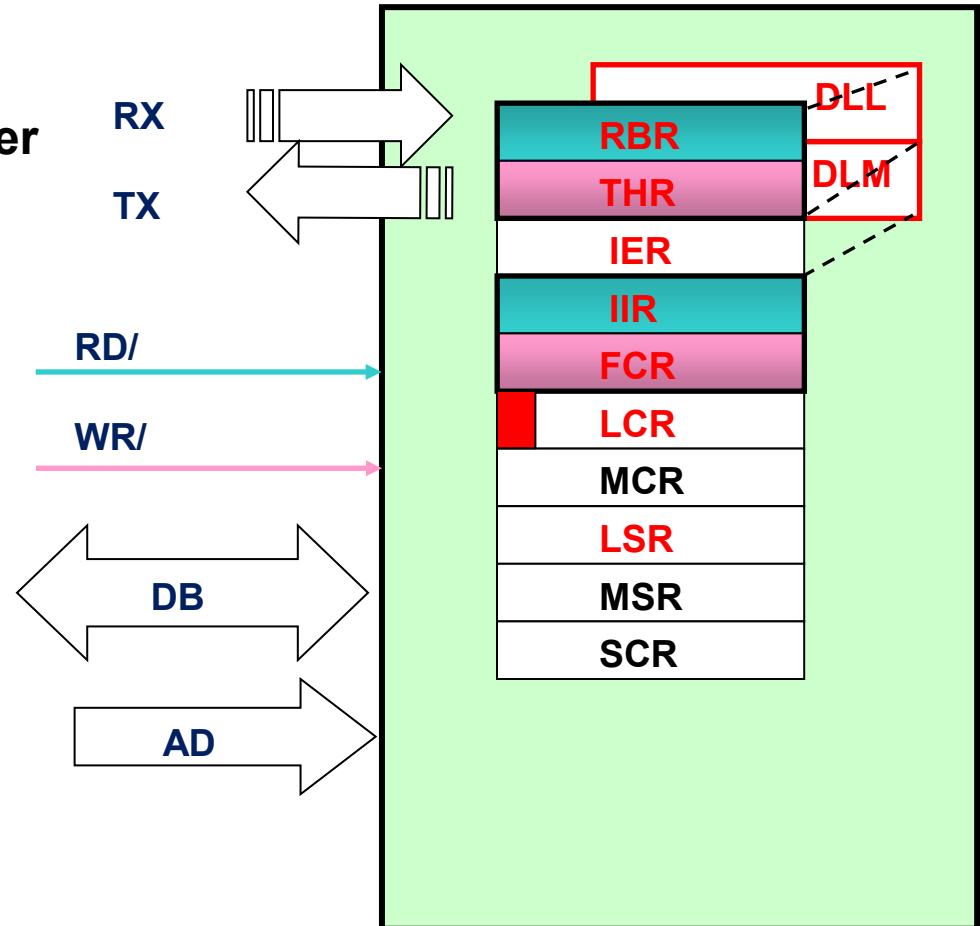


La UART 16550: sus registros y direccionamiento

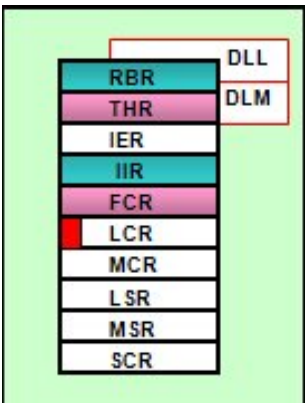


La UART 16550: Registros

RBR: Receiver Buffer Register
THR: Transmitter Holding Register
DLL: Divisor Latch LSB
DLM: Divisor Latch MSB
IER: Interrupt Enable Register
IIR: Interrupt Identification Register
FCR: FIFO Control Register
LCR: Line Control Register
MCR: MODEM Control Register
LSR: Line Status Register
MSR: MODEM Status Register
SCR: Scratch Register



La UART1 en el LPC1769



R
e
g
i
s
t
r
o

Name	Description	Access	Reset Value ^[1]	Address
U1RBR (when DLAB =0)	Receiver Buffer Register. Contains the next received character to be read.	RO	NA	0x4001 0000 (when DLAB=0)
U1THR (when DLAB =0)	Transmit Holding Register. The next character to be transmitted is written here.	WO	NA	0x4001 0000 (when DLAB=0)
U1DLL (when DLAB =1)	Divisor Latch LSB. Least significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	R/W	0x01	0x4001 0000 (when DLAB=1)
U1DLM (when DLAB =1)	Divisor Latch MSB. Most significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	R/W	0x00	0x4001 0004 (when DLAB=1)
U1IER (when DLAB =0)	Interrupt Enable Register. Contains individual interrupt enable bits to the 7 potential UART1 interrupts.	R/W	0x00	0x4001 0004 (when DLAB=0)
U1IIR	Interrupt ID Register. Identifies which interrupt(s) are pending.	RO	0x01	0x4001 0008
U1FCR	FIFO Control Register. Controls UART1 FIFO usage and modes.	WO	0x00	0x4001 0008
U1LCR	Line Control Register. Contains controls for frame formatting and break generation.	R/W	0x00	0x4001 000C
U1MCR	Modem Control Register. Contains controls for flow control handshaking and loopback mode.	R/W	0x00	0x4001 0010
U1LSR	Line Status Register. Contains flags for transmit and receive status, including line errors.	RO	0x60	0x4001 0014
U1MSR	Modem Status Register. Contains handshake signal status flags.	RO	0x00	0x4001 0018
U1SCR	Scratch Pad Register. 8-bit temporary storage for software.	R/W	0x00	0x4001 001C
U1ACR	Auto-baud Control Register. Contains controls for the auto-baud feature.	R/W	0x00	0x4001 0020
U1FDR	Fractional Divider Register. Generates a clock input for the baud rate divider.	R/W	0x10	0x4001 0028
U1TER	Transmit Enable Register. Turns off UART transmitter for use with software flow control.	R/W	0x80	0x4001 0030
U1RS485CTRL	RS-485/EIA-485 Control. Contains controls to configure various aspects of RS-485/EIA-485 modes.	R/W	0x00	0x4001 004C
U1ADRMATCH	RS-485/EIA-485 address match. Contains the address match value for RS-485/EIA-485 mode.	R/W	0x00	0x4001 0050
U1RS485DLY	RS-485/EIA-485 direction control delay.	R/W	0x00	0x4001 0054

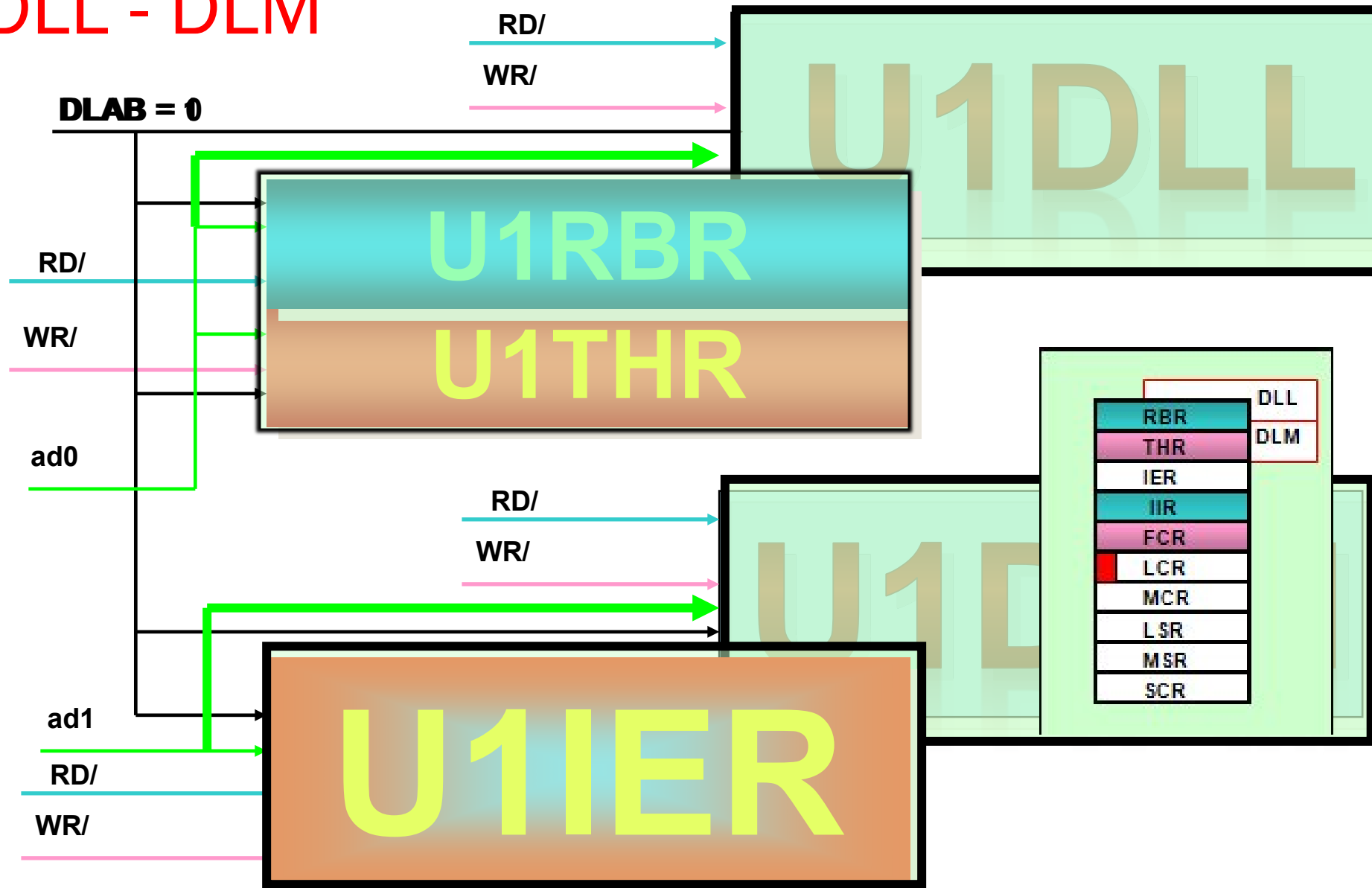
16550 Universal Asynchronous

Receiver/Transmitter with
FIFOs

LPC1769

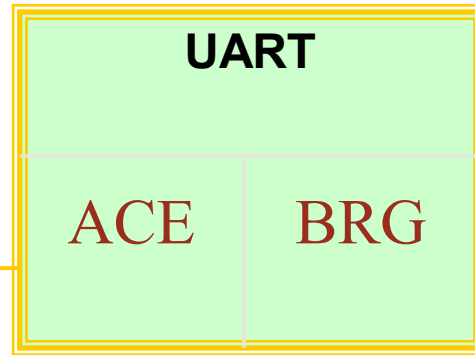


DLL - DLM



Recordando

N bits/seg

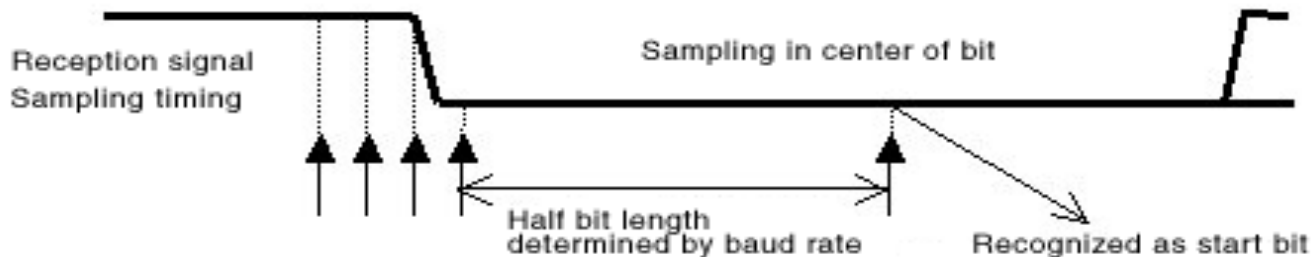


Si CCLK=100Mhz y
PLKSEL0 y 1 = 00

PCLK_UARTn
25 MHz

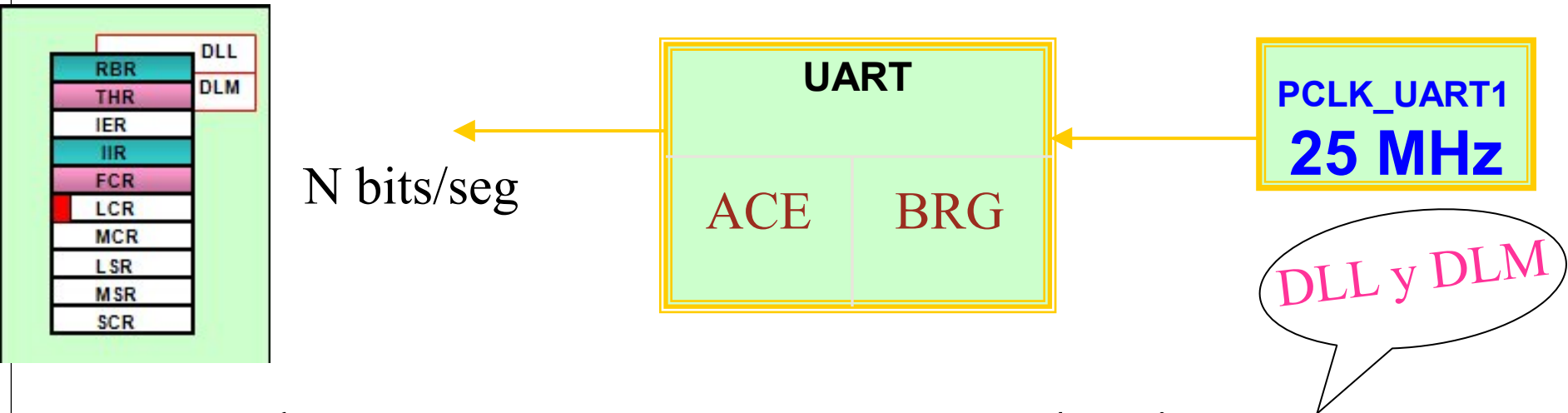
El BRG divide por defecto PCLK_UARTn por un divisor programable y por **16**, con el objeto de sobremuestrear la señal recibida.

$$\text{BaudRate} = \text{PCLK_UARTn} / 16 * \text{Divisor}$$



DLL y DLM

Volvemos: Cálculo del BaudRate



$$\text{BaudRate} = \text{PCLK_UARTn} / 16 * \text{Divisor}$$

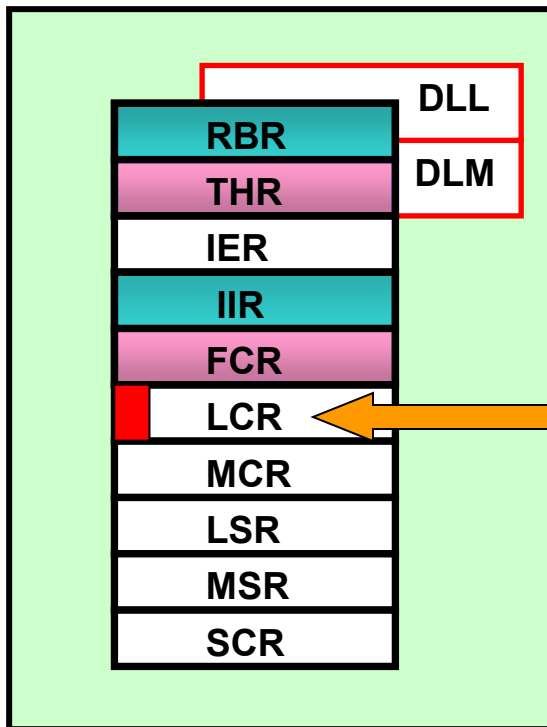
$$\text{BR}_{(\text{UART1})} = \text{PCLK_UART1} / 16 * (\text{U1DLM} * 256 + \text{U1DLL})$$

*Podría ocurrir que para el valor de PCLK_UARTn de trabajo no hubiera un valor de DLL-DLM que ajuste a un BR entero \Rightarrow **ERROR**.*

LCR: Line Control Register (1:3)

Base + 3 (activo alto)

DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
-------------	-----------	-----------	------------	------------	------------	-------------	-------------



- DLAB:** Divisor Latch access bit
- SB:** Set Break
- SP:** Stick parity
- EPS:** Even parity select
- PEN:** Parity enable
- STB:** number of stop bits
- WLS1/0:** word length

LCR: Line Control Register (2:3)

DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
-------------	-----------	-----------	------------	------------	------------	-------------	-------------

SB	Set break condition (Sout)
0	DISABLE
1	ENABLE

SP	EPS	PEN	bit de paridad forzado
1	1	1	Fuerza un 0
1	0	1	Fuerza un 1
0	x	x	DISABLE SP

DLAB	Permite acceso a...
0	RBR, THR e IER
1	Divisor Latches of the Baud Generator.

WLS1	WLS0	Descripción
1	1	8 bits de datos
1	0	7 bits de datos
0	1	6 bits de datos
0	0	5 bits de datos

PEN	¿Paridad?	EPS	Cual?
0	OFF	0	Odd
1	ON	1	Even

STB	largo	Stop bits
0	5,6,7 u 8	1
1	5	1,5
1	6,7 u 8	2

IER: Interrupt Enable Register

IER: Interrupt Enable Register– Base + 1 (enable=1)

0	0	0	0	MSI (Modem)	Rx line (er rx)	THR (Tx)	RBR (Rx)
---	---	---	---	----------------	-----------------------	-------------	-------------

RBR : Enable RBR Interrupt (¿Llegó un dato al RBR? => *interrumpe*)

THR : Enable THR Interrupt (¿Se vació el THR?=> *interrumpe*)

RX Line Status : Enable Received Line Status Interrupt (¿se produjeron errores en la línea de recepción? => *interrumpe*)

Modem Status Interrupt Enable : (¿errores en el Modem? => *interrumpe*) >>>> **SOLO UART1**

	DLL
RBR	
THR	DLM
IER	
IIR	
FCR	
LCR	
MCR	
LSR	
MSR	
SCR	

Pag. 302
y 322
del
manual

Configurando la comunicación serie

Realizaremos nuestra primera inicialización de una UART.

Partiremos de los “cosas comunes” que siempre debemos tener en cuenta a la hora de inicializar un periférico.

No basaremos en lo hecho para los Timers

Recordando

Timers – Configuración básica

1. **Alimentación:** registro PCONP, set bits PCTIM0/1/2/3.

Nota: en reset, Timer0/1 habilitados ($PCTIM0/1 = 1$), y Timer2/3 deshabilitados ($PCTIM2/3 = 0$).

2. **Clock:** en el registro PCLKSEL0, elegir PCLK_TIMER0/1; en el registro PCLKSEL1, elegir PCLK_TIMER2/3.

3. **Pines:** Elegir los pines mediante los registros PINSEL. Elegir el modo de los pines mediante los registros PINMODE.

4. **Interrupciones:** Ver registros T0/1/2/3MCR y T0/1/2/3CCR para eventos de captura y match. Las interrupciones se activan en el NVIC usando los registros apropiados.

5. **DMA:** Se pueden usar hasta dos condiciones de match para generar pedidos de DMA temporizados.

Recordando

Power Control for Peripherals register

(PCONP - 0x400F C0C4)

- El registro **PCONP** permite apagar los **periféricos que no se usan** para ahorrar energía. Esto se cumple cortando el clock a los periféricos deseados. Solo unos pocos periféricos no pueden ser apagados (Watchdog timer, Pin Connect y System Control).
- **Cada bit en PCONP controla un periférico**. Si el bit correspondientes es 1 en periféricos está habilitado.

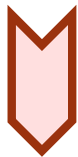
Recordando

Configuración básica - PCONP

Table 46. Power Control for Peripherals register (PCONP - address 0x400F C0C4) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved.	NA
1	PCTIM0	Timer/Counter 0 power/clock control bit.	1
2	PCTIM1	Timer/Counter 1 power/clock control bit.	1
3	PCUART0	UART0 power/clock control bit.	1
4	PCUART1	UART1 power/clock control bit.	1
5	-	Reserved.	NA
6	PCPWM1	PWM1 power/clock control bit.	1
7	PCI2C0	The I ² C0 interface power/clock control bit.	1
8	PCSPI	The SPI interface power/clock control bit.	1
9	PCRTC	The RTC power/clock control bit.	1
10	PCSSP1	The SSP 1 interface power/clock control bit.	1
11	-	Reserved.	NA
12	PCADC	A/D converter (ADC) power/clock control bit. Note: Clear the PDN bit in the AD0CR before clearing this bit, and set this bit before setting PDN.	0
13	PCCAN1	CAN Controller 1 power/clock control bit.	0
14	PCCAN2	CAN Controller 2 power/clock control bit.	0
15	PCGPIO	Power/clock control bit for IOCON, GPIO, and GPIO interrupts.	1
16	PCRIT	Repetitive Interrupt Timer power/clock control bit.	0

Pag. 63 del manual



Sigue...

PCLKSEL 0 y 1

Peripheral Clock Selection registers 0 and 1

(PCLKSELO - 0x400F C1A8 and PCLKSEL1 - 0x400F C1AC)

→ Cada par de bits de este registro controla el clock que recibe cada periférico como submúltiplo de la señal de clock del Sistema (CCLK).

Table 40. Peripheral Clock Selection register 0 (PCLKSELO - address 0x400F C1A8) bit description

Bit	Symbol	Description	Reset value
1:0	PCLK_WDT	Peripheral clock selection for WDT.	00
3:2	PCLK_TIMER0	Peripheral clock selection for TIMER0.	00
5:4	PCLK_TIMER1	Peripheral clock selection for TIMER1.	00
7:6	PCLK_UART0	Peripheral clock selection for UART0.	00
9:8	PCLK_UART1	Peripheral clock selection for UART1.	00
11:10	-	Reserved.	NA
13:12	PCLK_PWM1	Peripheral clock selection for PWM1.	00
15:14	PCLK_I2C0	Peripheral clock selection for I ² C0.	00
17:16	PCLK_SPI	Peripheral clock selection for SPI.	00

Table 42. Peripheral Clock Selection register bit values

PCLKSELO and PCLKSEL1 individual peripheral's clock select options	Function
00	PCLK_peripheral = CCLK/4
01	PCLK_peripheral = CCLK
10	PCLK_peripheral = CCLK/2
11	PCLK_peripheral = CCLK/8


Sigue...

Pag. 56 del
manual

ejemplo
initUART()

Pasos para la inicialización general

Pasos necesarios para configurar la UART0 (resto ídem) debido a su pertenencia a la familia LPC176x

1. Power:

PCONP |= 0x01<<3; //ver pag. 63 user manual
//1.- Registro PCONP (0x400FC0C4) – bit 4 en 1 prende la UART1

2. Peripheral clock: Selección del Clock

Bits **PCLK_UART0** en **PCLKSEL0**

PCLKSEL0 and PCLKSEL1 individual peripheral's clock select options	Function	Reset value
00	PCLK_peripheral = CCLK/4	00
01	PCLK_peripheral = CCLK	
10	PCLK_peripheral = CCLK/2	
11	PCLK_peripheral = CCLK/8, except for CAN1, CAN2, and CAN filtering when "11" selects = CCLK/6.	

3. UART FIFO (opcional): 16 bytes de buffer para transmisión y recepción

U0FCR |= 0x01 //ver pag. 305 user manual

4. Pins: Selección de pines con **PinSEL()** y los modos con **PinMODE()**

//El pin de recepción no debe tener habilitada la resistencia de pull-down.

Ejemplo Inicialización UART0

Ejemplo: 9600,8,N,1

DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
1	0	0	0	0	0	1	1

Table 52. Interrupt

Bit	Name
0	ISE_WDT

IER: Interrupt Enable Register

IER: Interrupt Enable Register- Base + 1 (enable=1)

0	0	0	0	EDSSI (Modem)	ELSI (errx)	ETBEI (Tx)	ERBFI (Rx)
---	---	---	---	------------------	----------------	---------------	---------------

ERBFI: Enable Received Buffer Full Interrupt
(¿ Llegó un dato al RBR? => *interrumpe*)

errupt.

tes that the interrupt is enabled.

LSR: Line Status Register (1:4)

Base + 5 (activo alto)

X	TEMT	THRE	BI	FE	PE	OE	DR
----------	-------------	-------------	-----------	-----------	-----------	-----------	-----------

Estado de la línea

TEMT: Transmitter Empty

THRE: Tr. Holding Register Empty

DR : Data Ready

¿Qué error (de Rx)?

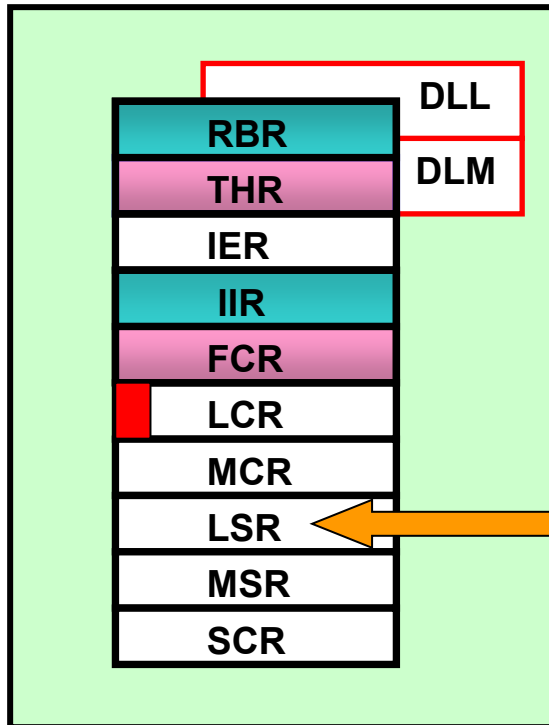
BI: Break Interrupt

FE: Framing Error

PE: Parity Error

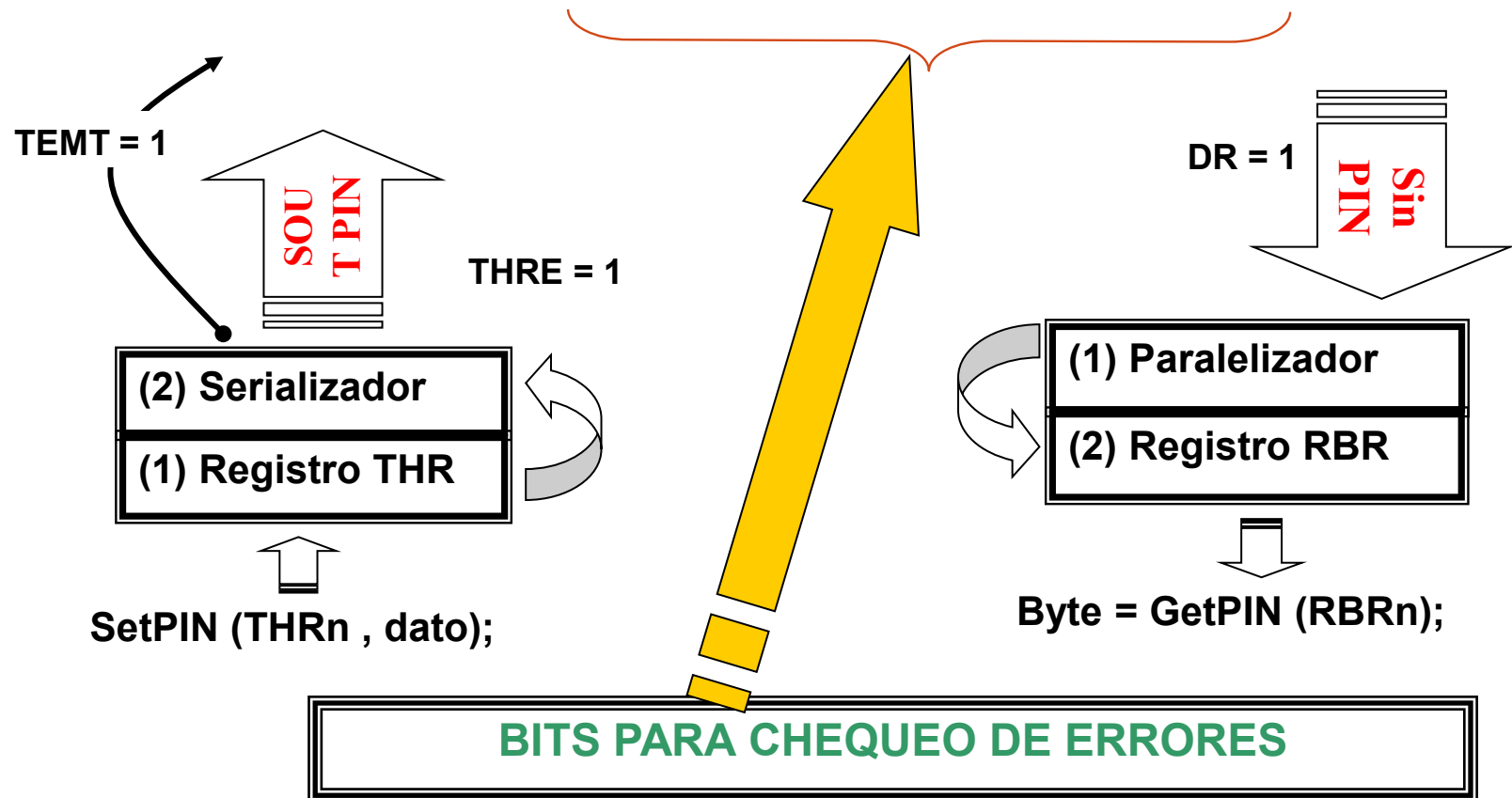
OE: Overrun Error

LSR



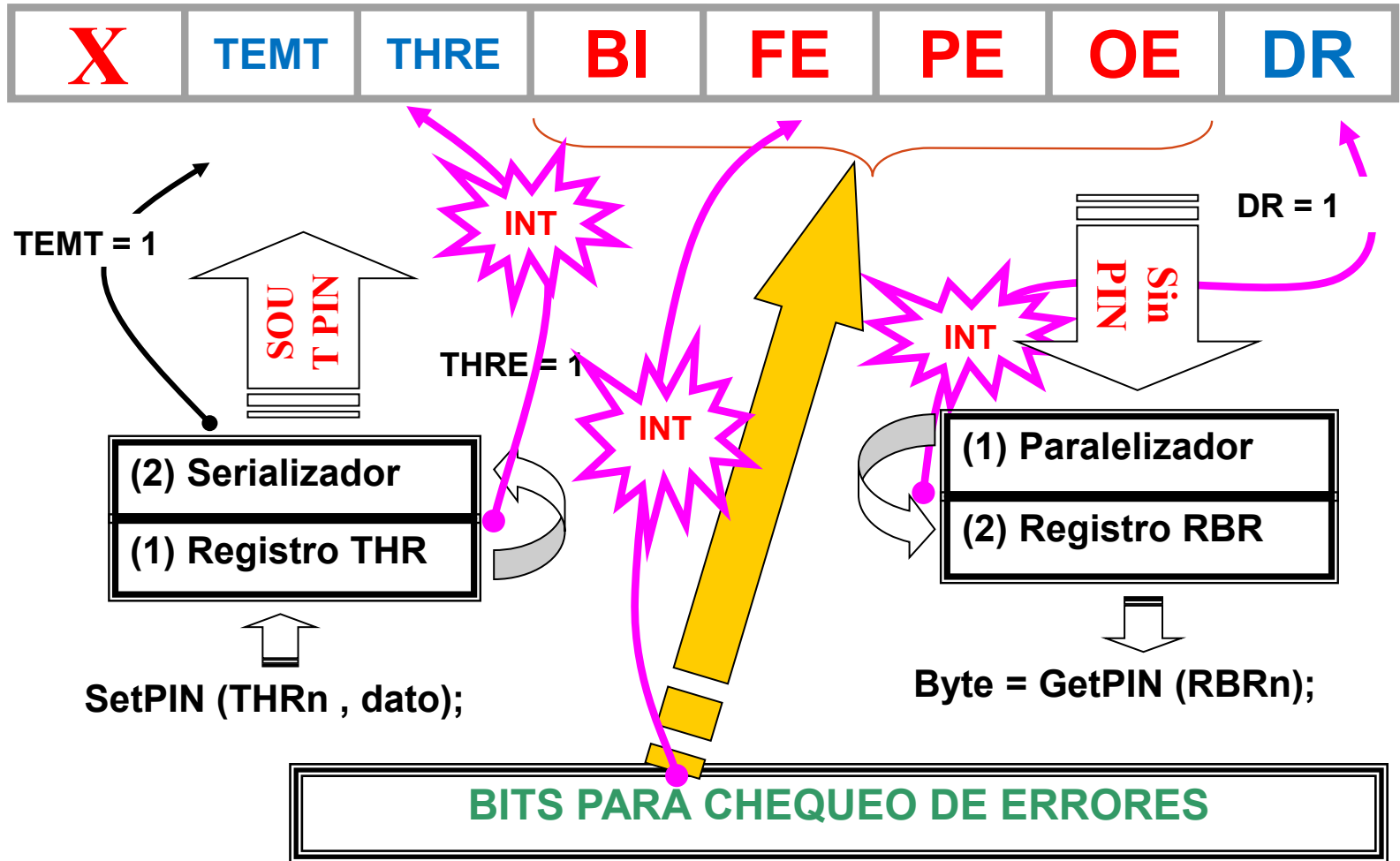
LSR: Line Status Register (2:4)

Base + 5 (activo alto)



LSR: Line Status Register (3:4)

Base + 5 (activo alto)



LSR: Line Status Register (4:4)

Base + 5 (activo alto)



BITS PARA CHEQUEO DE ERRORES

Break Interrupt

Se activa cuando la línea de rx se queda en estado bajo durante un tiempo \gt al de un carácter luego de una trama válida.

Framing Error

Indica que el dato recibido no tiene los bits de stop. correctos

Parity Error

Se activa cuando el dato recibido no tiene el bit de paridad correcto

Overrun Error

El dato no fue leído por el procesador a tiempo. Al llegar uno nuevo se sobre escribe y en consecuencia se activa este bit.

IIR

IIR: Interrupt Identif. Register

IIR: Interrupt Identification Register (Base + 2)

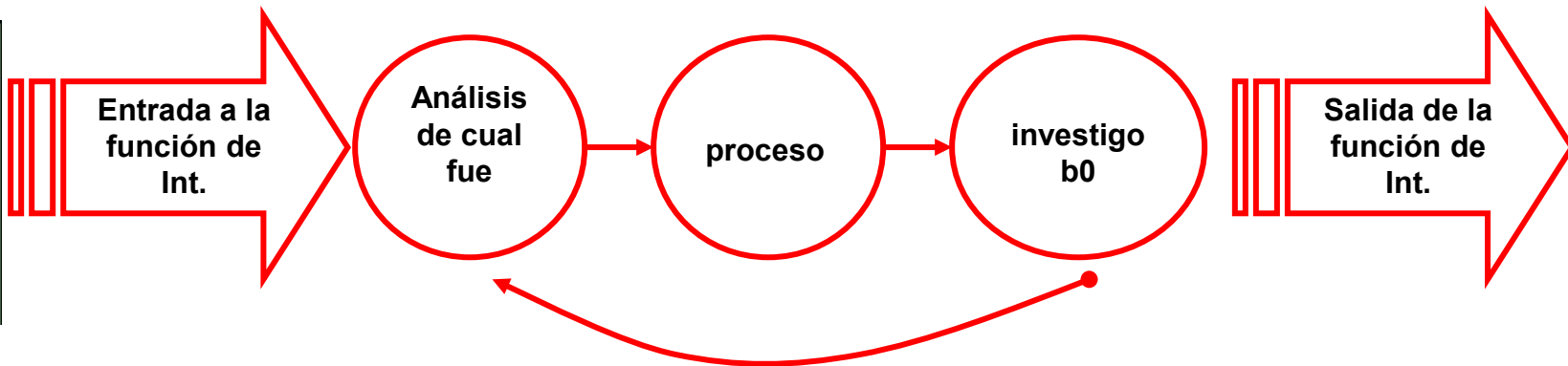
FIFO	0	0	X	b2	b1	b0
-------------	----------	----------	----------	-----------	-----------	-----------

b0 = 0 ⇒ Hay una interrupción pendiente de atención

b0 = 1 ⇒ No hay una interrupción pendiente



b2	b1	Descripción
1	1	Errores y break (>)
1	0	Dato disponible
0	1	THR disponible
0	0	Estado del Modem (<)



Resumen de los registros

LCR: Line Control Register – Base + 3 (activo alto)

DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
------	----	----	-----	-----	-----	------	------

LSR: Line Status Register – Base + 5 (activo alto)

X	TEMT	THRE	BI	FE	PE	OE	DR
----------	------	------	----	----	----	----	----

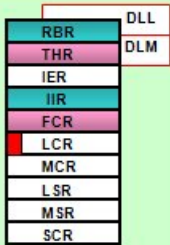
IER: Interrupt Enable Register– Base + 1 (activo alto)

0	0	0	0	EDSSI	ELSI	ETBEI	ERBFI
----------	----------	----------	----------	-------	------	-------	-------

Modem err Rx Tx Rx

IIR: Interrupt Identification Register– Base + 2

FIFO	0	0	X	b2	b1	b0
-------------	----------	----------	----------	----	----	----



Agradecimientos

Ing. Marcelo Trujillo

Ing. Marcelo Giura

Ing. Gabriel Socodato

Ing. Marcelo Romeo