Interfaces seriales

Transmisión de Datos

Ing. Luis Di Pinto (ldipinto@herrera.unt.edu.ar)

http://www.microprocesadores.unt.edu.ar/transmision/

Esquema General (2da parte)

- En esta segunda parte de la materia, vamos a enfocarnos en un conjunto de interfaces seriales.
  - Algunas de las más usadas y populares.



- A su vez, tendrá mucha más importancia la parte práctica, con laboratorios.
  - Empiezan a usar la EDU-CIAA.

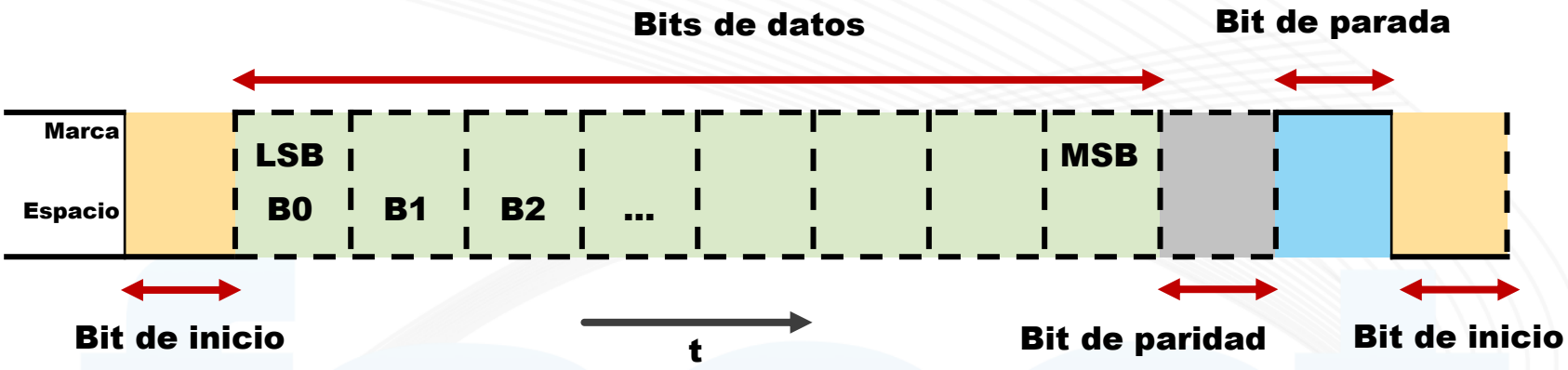
Temas que veremos

- Transmisión Serie Asincrónica.
  - Tramas, parámetros.
- Sincronización de relojes.
  - Oversampling, problemas de degradación.
- UART
  - Características principales
  - Configuración.

Transmisión Serie Asincrónica

- El formato general de la transmisión proviene del estándar EIA RS-232.
- Cada bit se transmite como un valor de tensión utilizando modulación NRZ-L (Non Return Zero Level):
  - Para la transmisión de un '1' se utiliza un valor de tensión denominado marca.
  - Para la transmisión de un '0' se utiliza un valor de tensión denominado espacio.
- Los periféricos integrados en los microcontroladores (SCI) usan Vcc como marca y Vss como espacio.

Transmisión Serie Asincrónica



- Se transmiten caracteres con 5 a 8 bits datos.
- El bit menos significativo se transmite primero y el más significativo se transmite al final.
- Tambien puede agregarse un bit de paridad para detectar errores en el carácter.
- Al principio de cada carácter se agrega un espacio (0) y al final una o dos marcas (1).

Transmisión Serie Asincrónica

- El receptor debe conocer los parámetros con los que el transmisor envía la información:
  - Velocidad de modulación.
  - Cantidad de bits de datos.
  - Existencia y forma del cálculo de paridad.
  - Cantidad de bits de parada.
- Suele emplearse como convención la terminología D/P/S:
  - 8/N/1 (u 8N1) indica 8 bits de datos, sin paridad y 1 bit de stop.
  - 7E1 indica 7 bits de datos, paridad par y 1 bit de stop.
- La velocidad de modulación puede ir desde 300 hasta 115200 baudios.
  - Una elección bastante común es de 9600 baudios.

Opciones de paridad

Caracter	ASCII	Paridad	Valor a transmitir
A	0x41	Ninguna	0100.0001
		Marca	0100.0001.1
		Espacio	0100.0001.0
		Par	0100.0001.0
		Impar	0100.0001.1
B	0x42	Par	0100.0010.0
		Impar	0100.0010.1
C	0x43	Par	0100.0011.1
		Impar	0100.0001.0

Parámetros de la comunicación

t_b duración de un bit.

n_d bits de datos en el caracter.

n_c total de bits en el caracter.

$t_c = n_c * t_b$ duración de un caracter.

$D = \frac{1}{t_b}$ velocidad de modulación

$cps = \frac{1}{t_c}$ caracteres por segundo

$R = \frac{n_d}{n_c} * D$ tasa de transferencia

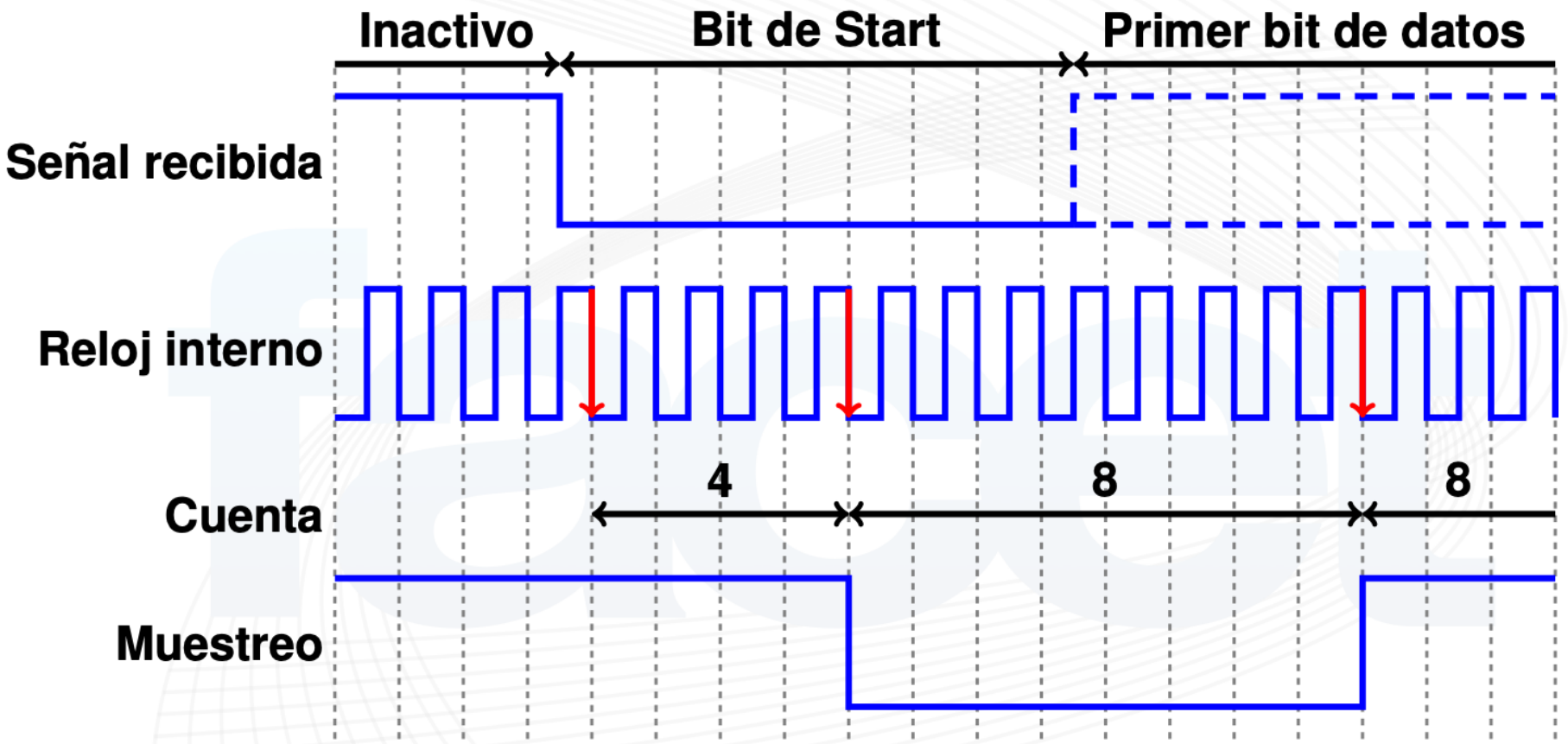
Sincronización de relojes

- Cuando el canal está inactivo permanece en nivel de marca.
- Recordemos que el transmisor inserta un bit de espacio al inicio y un bit de marca al final de cada caracter.
- Esto garantiza un flanco descendente al inicio de cada caracter, que se utiliza para sincronizar el reloj del receptor.
- El receptor utiliza un reloj con una frecuencia múltiplo *k* de la velocidad de modulación.
  - Usualmente, este factor es igual a 16.

Sincronización de relojes

- Cuando se detecta el flanco descendente al inicio de caracter, es posible que se trate de un bit de inicio, por lo que se comienzan a contar *k/2* flancos descendentes de reloj.
  - El objetivo es tratar de hacer la lectura en el centro del bit.
- Luego de esos *k/2* flancos, pueden ocurrir dos cosas: que en ese momento haya o no un espacio.
- Si no hay un espacio, entonces no era un bit de inicio.
  - El flanco descendente detectado puede haber sido ruido o un microcorte.
- Si se mantiene el espacio, entonces es un bit de inicio.
  - En este punto se comienza a generar la señal de muestreo, cada *k* flancos de la señal de reloj.

Sincronización de relojes



- En este gráfico $k = 8$.

Sincronización de relojes

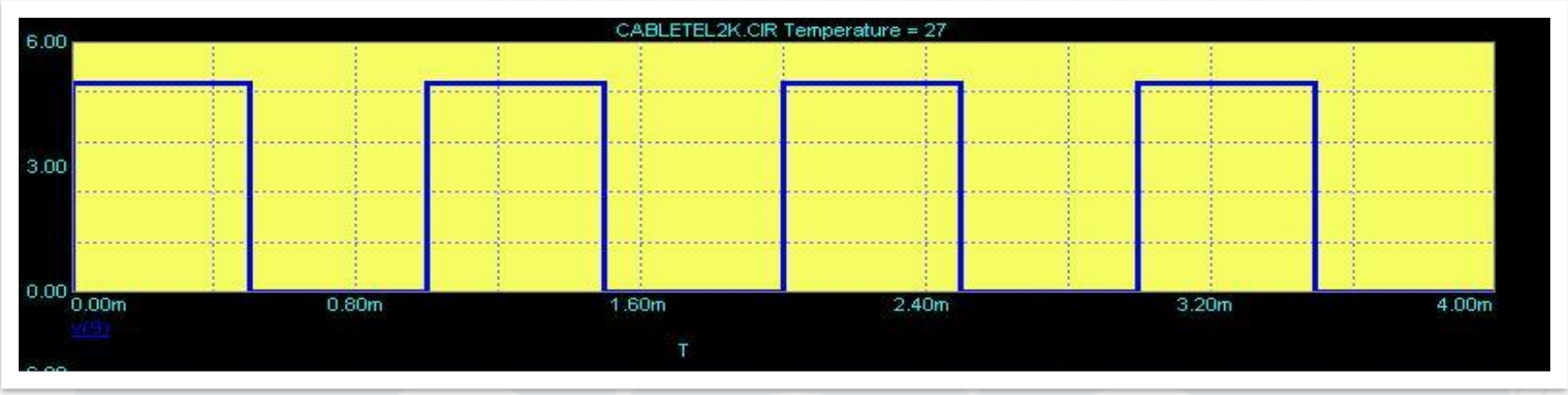


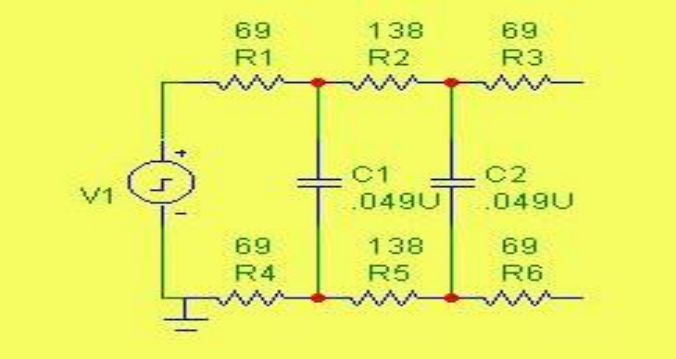
- La máxima diferencia de reloj admisible es media duración de bit (t_b) por caracter.

Degradación de la señal

- La resistencia y capacidad de un cable real degradan las señales cuadradas.
- Los flancos rectos se transforman en curvas exponenciales.
- Cuando aumenta la velocidad de modulación y la distancia los efectos se vuelven más significativos.

Degradación de la señal





Modelo de 2 Km
de cable telefónico

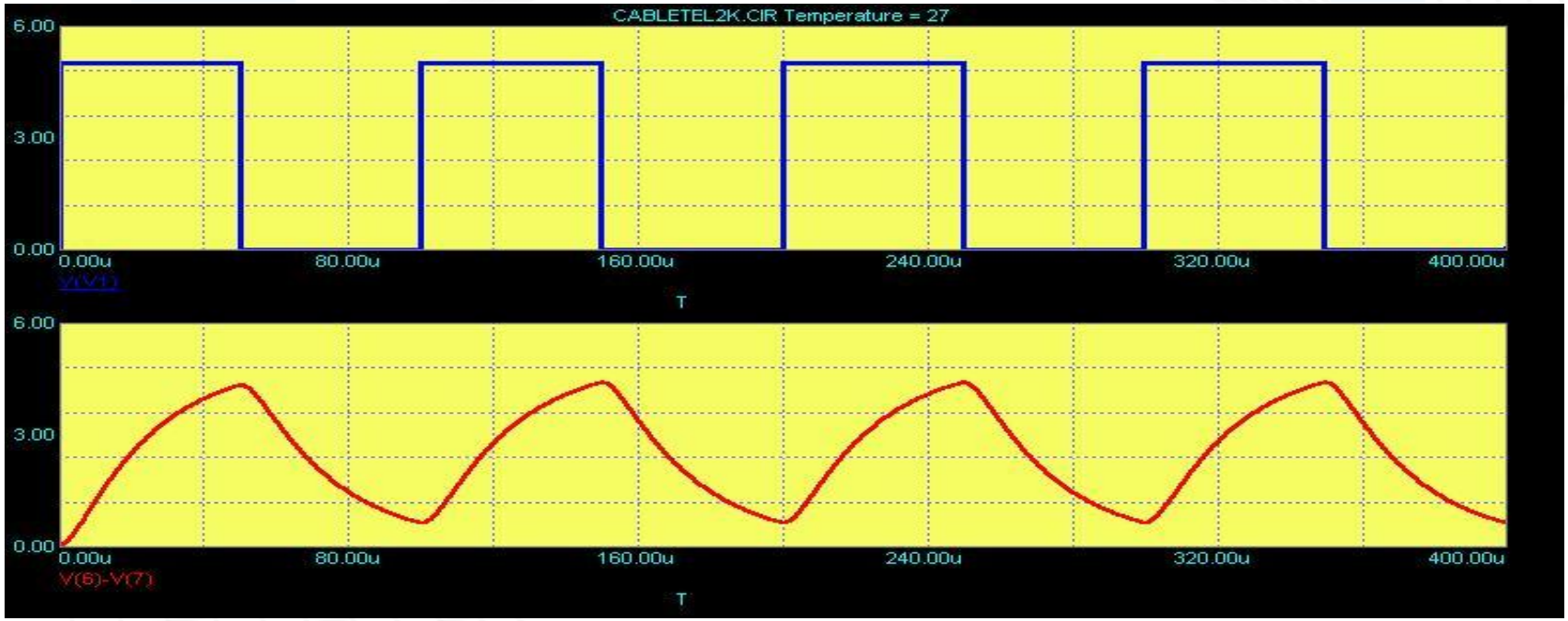
Degradación de la señal

- Degradación de una señal de 2 Kbps transmitida por un par telefónico.



Degradación de la señal

- Degradación de una señal de 20 Kbps transmitida por un par telefónico.

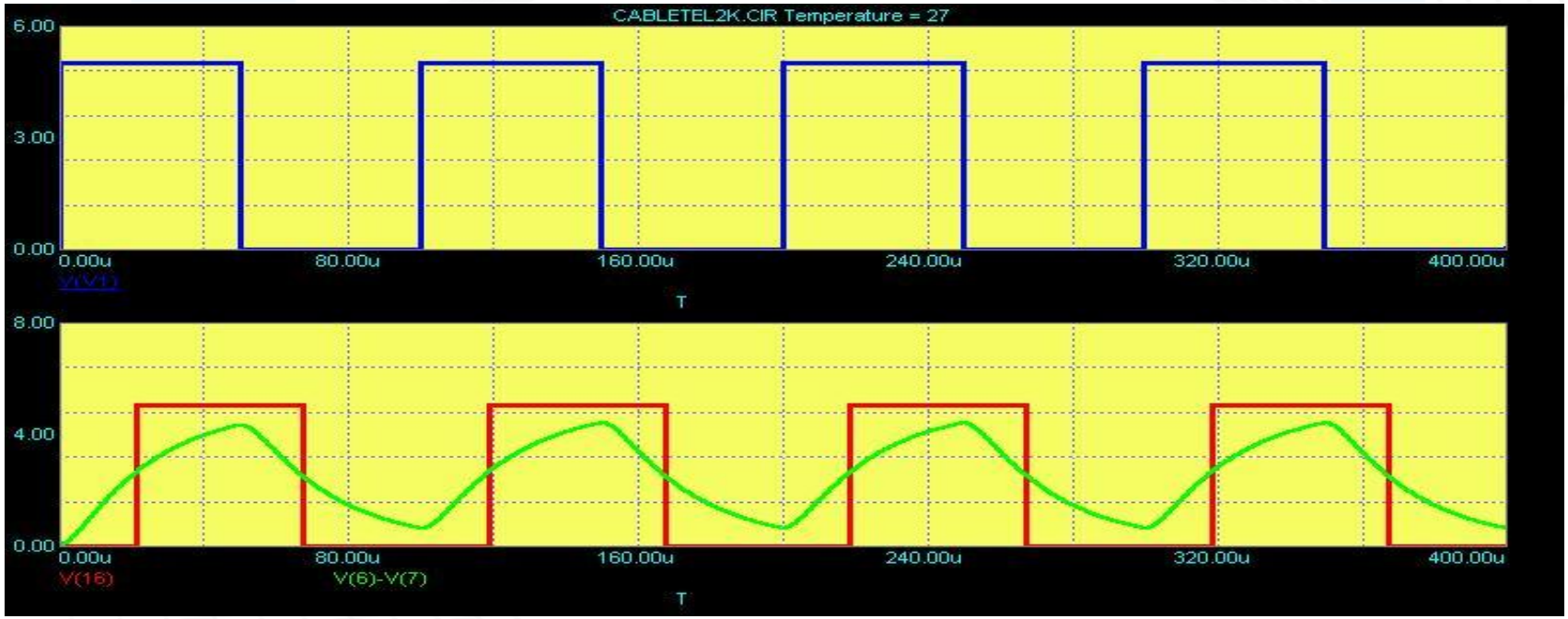


Degradación de la señal

- Para reconstruir la señal se utilizan circuitos comparadores.
- Un comparador es un amplificador operacional a lazo abierto:
  - Cuando la señal de entrada es menor al valor de referencia la salida satura en Vss.
  - Cuando es mayor la salida satura en Vcc.

Degradación de la señal

- Resultado obtenido al reconstruir la señal comparando con 2,5V

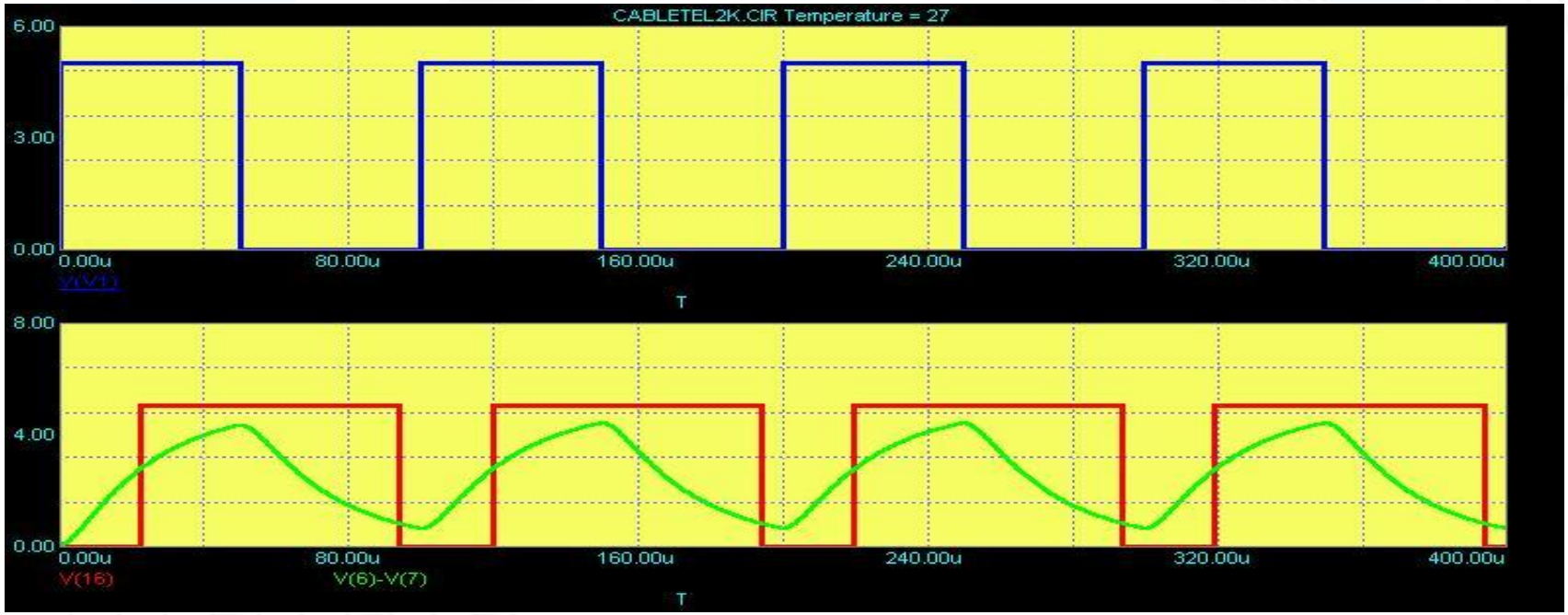


Degradación de la señal

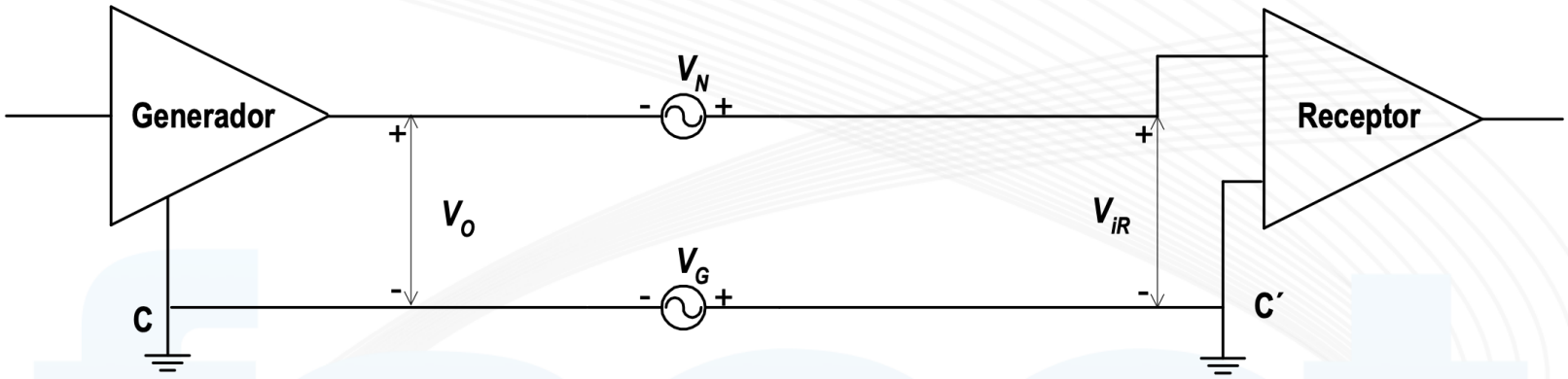
- Para lograr inmunidad al ruido se utilizan comparadores con histéresis:
  - El cambio de '0' a '1' utiliza una referencia mayor que el cambio de '1' a '0'.
  - El nivel ruido que el sistema puede tolerar está dado por la diferencia entre las tensiones de referencia del comparador.
- Esta técnica cambia el ancho de los bits y agrega complicaciones en la decodificación.

Degradación de la señal

- Resultado obtenido al reconstruir la señal comparando con 0,8V y 2,7V



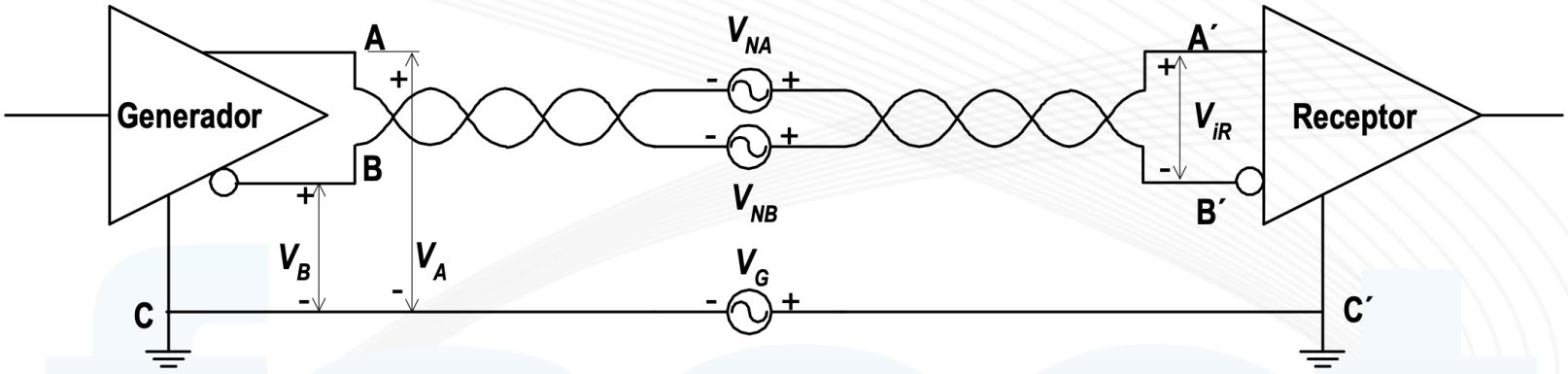
Interfaces en modo común



$$V_{iR} = V_0 + V_N - V_G$$

- Se utiliza una línea con referencia a tierra.
- Problemas con los ruidos inducidos en las líneas.
- Problemas por diferencias de potencial entre las tierras.

Interfaces balanceadas



$$V_{iR} = (V_A + V_{NA} - V_G) - (V_B + V_{NB} - V_G) = (V_A - V_B) + (V_{NA} - V_{NB})$$

- Se utilizan dos líneas y se transmiten los datos por la diferencia de tensión entre ambas.
- Resuelve los problemas de potencial entre las tierras y minimiza los problemas de ruido.

UART y USART

- UART: *Universal Asynchronous Receiver Transmitter*
  - Sólo puede operar en modo asincrónico.
  - Más simple de configurar y más barata.
- USART: *Universal Synchronous Asynchronous Receiver Transmitter*
  - Puede operar en modo sincrónico y asincrónico.
  - Más compleja de configurar y más cara.
  - Se puede transmitir a mayores velocidades.

UART y USART

- TXD: Terminal para la transmisión de datos.
- RXD: Terminal para la recepción de datos.
- DIR: Terminal para controlar la dirección del canal en comunicaciones half duplex.
- UCLK: Terminal de salida o entrada de reloj cuando se utiliza para modo sincrónico.
- BAUD: Terminal de salida de la señal de reloj multiplicada por 16.

Esquema de una UART

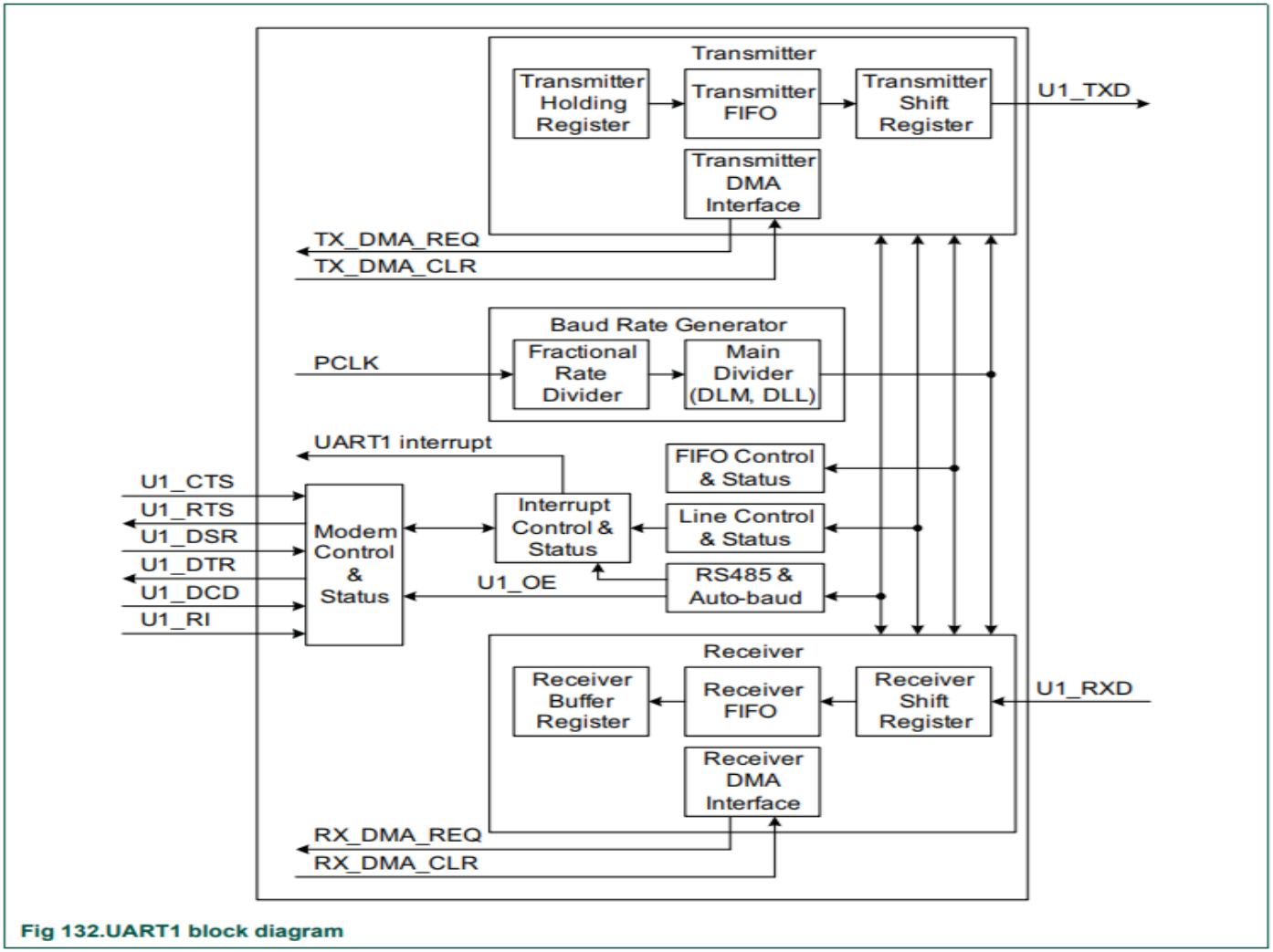


Fig 132.UART1 block diagram

Esquema de una USART

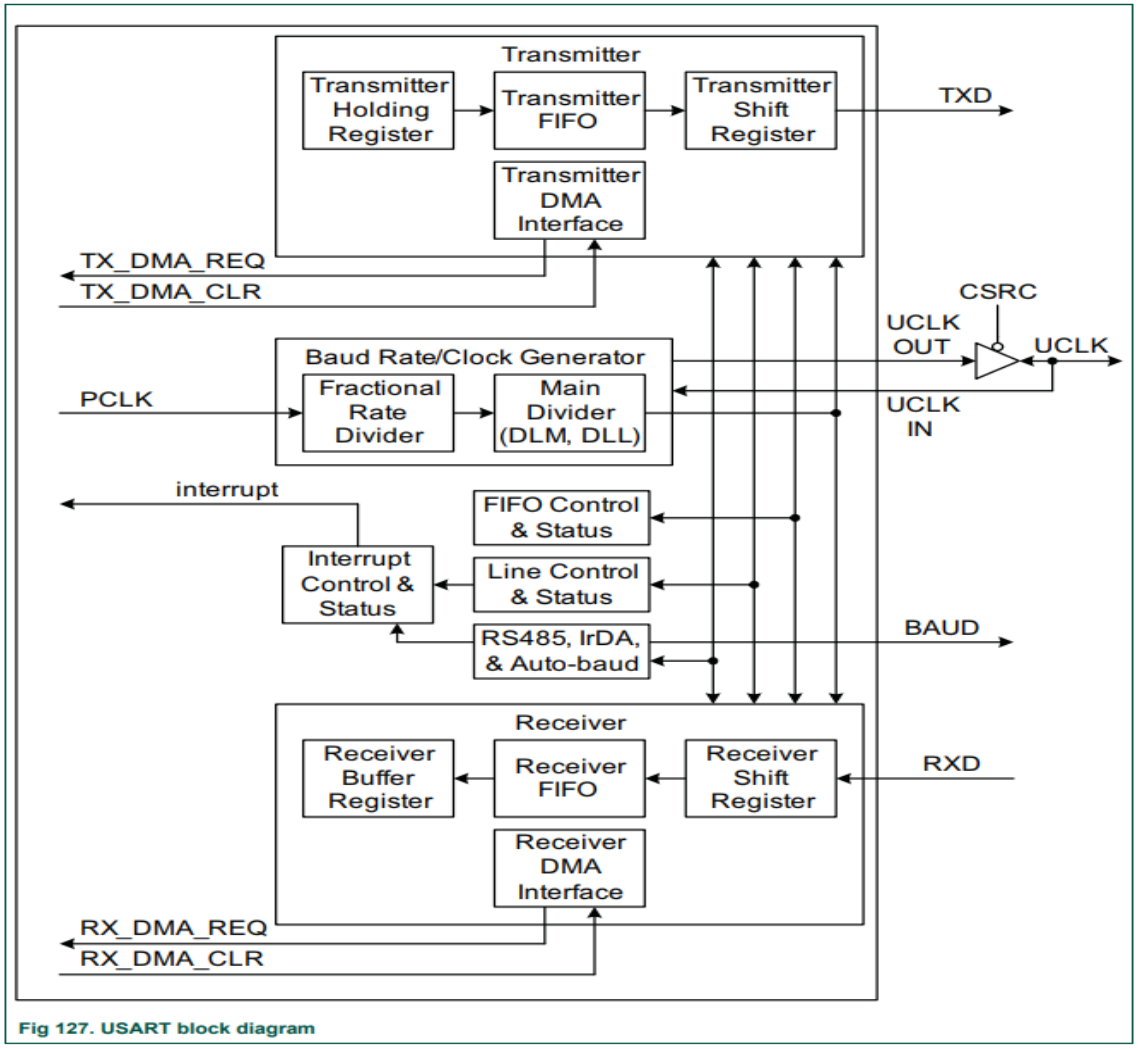


Fig 127. USART block diagram

Descripción de los terminales

Pin function	Direction	Description
USART0		
U0_RXD	I	**Serial Input.** Serial receive data.
U0_TXD	O	**Serial Output.** Serial transmit data.
U0_DIR	I/O	RS-485/EIA-485 output enable/direction control.
U0_UCLK	I/O	Serial clock input/output for USART0 in synchronous mode.
USART2		
U2_RXD	I	**Serial Input.** Serial receive data.
U2_TXD	O	**Serial Output.** Serial transmit data.
U2_DIR	I/O	RS-485/EIA-485 output enable/direction control.
U2_UCLK	I/O	Serial clock input/output for USART2 in synchronous mode.
USART3		
U3_RXD	I	**Serial Input.** Serial receive data.
U3_TXD	O	**Serial Output.** Serial transmit data.
U3_DIR	I/O	RS-485/EIA-485 output enable/direction control.
U3_UCLK	I/O	Serial clock input/output for USART3 in synchronous mode.
U3_BAUD	O	USART3 baud output. U3_BAUD is an active LOW signal of the single clock cycle and is generated at each rising edge of a 16x clock signal for the transmitter section of the UART. The clock rate is established by the USART3 clock frequency divided by the fractional divider and the divisor specified in the baud generator divisor latches. U3_BAUD can be used as an input to an external IrDA module.

Descripción de los registros

Table 923. Register overview: USART0/2/3 (base address: 0x4008 1000 (UART0), 0x400C 1000 (UART2), 0x400C 2000 (UART3))

Name	Access	Address offset	Description	Reset value	Reference
RBR	RO	0x000	Receiver Buffer Register. Contains the next received character to be read (DLAB = 0).	NA	Table 924
THR	WO	0x000	Transmit Holding Register. The next character to be transmitted is written here (DLAB = 0).	NA	Table 925
DLL	R/W	0x000	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 (DLAB = 1).	0x01	Table 926
DLM	R/W	0x004	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 (DLAB = 1).	0x00	Table 927
IER	R/W	0x004	Interrupt Enable Register. Contains individual interrupt enable bits for the 7 potential USART interrupts (DLAB = 0).	0x00	Table 928
IIR	RO	0x008	Interrupt ID Register. Identifies which interrupt(s) are pending.	0x01	Table 929
FCR	WO	0x008	FIFO Control Register. Controls USART FIFO usage and modes.	0x00	Table 931
LCR	R/W	0x00C	Line Control Register. Contains controls for frame formatting and break generation.	0x00	Table 932
-	-	0x010	Reserved	-	-
LSR	RO	0x014	Line Status Register. Contains flags for transmit and receive status, including line errors.	0x60	Table 933

Descripción de los registros

Table 923. Register overview: USART0/2/3 (base address: 0x4008 1000 (UART0), 0x400C 1000 (UART2), 0x400C 2000 (UART3))

Name	Access	Address offset	Description	Reset value	Reference
			and break generation.		
-	-	0x010	Reserved	-	-
LSR	RO	0x014	Line Status Register. Contains flags for transmit and receive status, including line errors.	0x60	Table 933
-	-	0x018	Reserved	-	-
SCR	R/W	0x01C	Scratch Pad Register. Eight-bit temporary storage for software.	0x00	Table 934
ACR	R/W	0x020	Auto-baud Control Register. Contains controls for the auto-baud feature.	0x00	Table 935
ICR	R/W	0x024	IrDA control register (USART3 only)	0x00	Table 936
FDR	R/W	0x028	Fractional Divider Register. Generates a clock input for the baud rate divider.	0x10	Table 938
OSR	R/W	0x02C	Oversampling Register. Controls the degree of oversampling during each bit time.	0xF0	Table 939
-	-	0x030 - 0x03C	Reserved	-	-
HDEN	R/W	0x040	Half-duplex enable Register		Table 940
-	-	0x044	Reserved	-	-
SCICTRL	R/W	0x048	Smart card interface control register	0x00	Table 941
RS485CTRL	R/W	0x04C	RS-485/EIA-485 Control. Contains controls to configure various aspects of RS-485/EIA-485 modes.	0x00	Table 942
RS485ADRMATCH	R/W	0x050	RS-485/EIA-485 address match. Contains the address match value for RS-485/EIA-485 mode.	0x00	Table 943
RS485DLY	R/W	0x054	RS-485/EIA-485 direction control delay.	0x00	Table 944
SYNCCTRL	R/W	0x058	Synchronous mode control register.	0x00	Table 945
TER	R/W	0x05C	Transmit Enable Register. Turns off USART transmitter for use with software flow control.	0x01	Table 946

Descripción de los registros

Table 924. USART Receiver Buffer Registers when DLAB = 0, Read Only (RBR, addresses 0x4008 1000 (USART0), 0x400C 1000 (USART2), 0x400C 2000 (USART3)) bit description

Bit	Symbol	Description	Reset value
7:0	RBR	Receiver buffer. The USART Receiver Buffer Register contains the oldest received byte in the USART RX FIFO.	undefined
31:8	-	Reserved	-

Table 925. USART Transmitter Holding Register when DLAB = 0, Write Only (THR, addresses 0x4008 1000 (USART0), 0x400C 1000 (USART2), 0x400C 2000 (USART3)) bit description

Bit	Symbol	Description	Reset value
7:0	THR	Transmit Holding Register. Writing to the USART Transmit Holding Register causes the data to be stored in the USART transmit FIFO. The byte will be sent when it reaches the bottom of the FIFO and the transmitter is available.	NA
31:8	-	Reserved	-

Configuración del divisor

- Se usa para dividir el reloj del USART_PCLK y producir la velocidad en baudios del reloj, que debe ser 16X la velocidad de modulación deseada.
- DLL contiene los 8 bits inferiores del divisor y DLM contiene los 8 bits más altos del divisor.
- El bit de acceso a la configuración de divisores (DLAB) en LCR debe ser 1 para poder acceder a los registros del divisor USART.

Table 926. USART Divisor Latch LSB Register when DLAB = 1 (DLL, addresses 0x4008 1000 (USART0), 0x400C 1000 (USART2), 0x400C 2000 (USART3)) bit description

Bit	Symbol	Description	Reset value
7:0	DLLSB	Divisor latch LSB. The USART Divisor Latch LSB Register, along with the DLM register, determines the baud rate of the USART.	0x01
31:8	-	Reserved	-

Table 927. USART Divisor Latch MSB Register when DLAB = 1 (DLM, addresses 0x4008 1004 (USART0), 0x400C 1004 (USART2), 0x400C 2004 (USART3)) bit description

Bit	Symbol	Description	Reset value
7:0	DLMSB	Divisor latch MSB. The USART Divisor Latch MSB Register, along with the DLL register, determines the baud rate of the USART.	0x00
31:8	-	Reserved	-

Registro fraccional del divisor

- Toma el reloj de entrada PCLK y genera un reloj de salida de acuerdo a los requisitos fraccionales especificados.
- Este registro controla el pre-scaler del reloj para la generación de la velocidad en baudios.

Table 938. USART Fractional Divider Register (FDR, addresses 0x4008 1028 (USART0), 0x400C 1028 (USART2), 0x400C 2028 (USART3)) bit description

Bit	Function	Description	Reset value
3:0	DIVADDVAL	Baud rate generation pre-scaler divisor value. If this field is 0, fractional baud rate generator will not impact the USART baud rate.	0
7:4	MULVAL	Baud rate pre-scaler multiplier value. This field must be greater or equal 1 for USART to operate properly, regardless of whether the fractional baud rate generator is used or not.	1
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0

Velocidad de modulación

$$baudrate_{UART} = \frac{PCLK}{16 \times (256 \times DLM + DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

- PCLK es el reloj de periféricos.
- DLM y DLL son los registros divisores de la UART.
- DivAddVal y MulVal son los fraccionales de la UART
- El valor MulVal debe estar en el rango de 1 a 15
- El valor DivAddVal debe estar en el rango de 0 y 14
- El valor DivAddVal debe ser menor que MulVal

Cálculo del fraccional del divisor

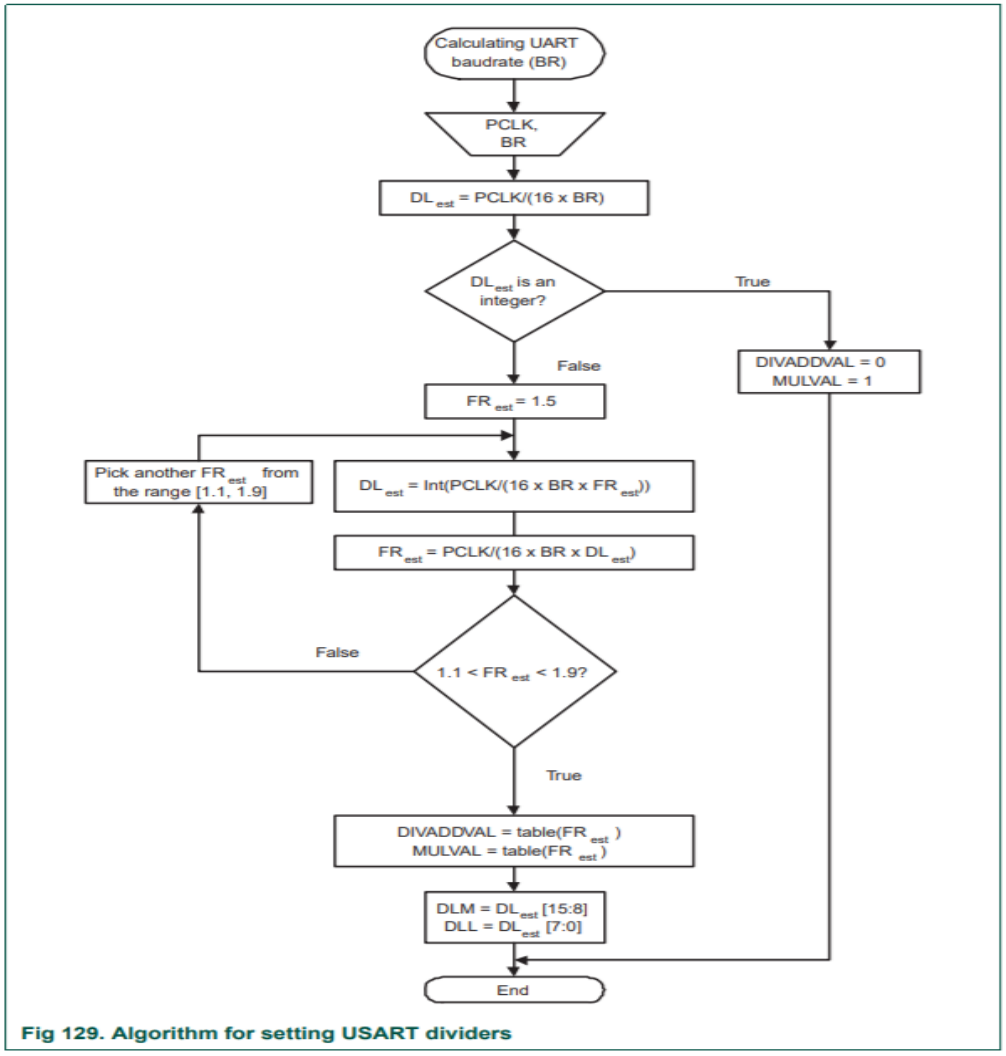


Fig 129. Algorithm for setting USART dividers

Tabla de valores del fraccional

Table 947. Fractional Divider setting look-up table

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

Ejemplo de cálculo de divisores (1)

- Supongamos una frecuencia de reloj interna de 14.7456 MHz (PCLK), y que queremos transmitir a una velocidad de 9600 baudios (BR).
- Calculamos DL_{est} = PCLK / (16 * BR)
  - DL_{est} = 14.7456 MHz / (16 * 9600) = **96**
- Como DL_{est} es un entero:
  - DivAddVal = 0 y MulVal = 1
  - DLM = 0 y DLL = 96

Ejemplo de cálculo de divisores (2)

- Supongamos ahora que PCLK es 12 MHz, y que queremos transmitir a 115200 baudios (BR).
- Calculamos DL_{est} = PCLK / (16 * BR)
  - DL_{est} = 12 MHz / (16 * 115200) = **6,51**
- Como DL_{est} **no** es un entero, hay que estimar el valor de FR.
- Usamos un FR_{est} = 1,5 y recalculamos DL_{est} y FR_{est}
  - DL_{est} = Int[PCLK / (16 * BR * FR_{est})] = **4**
  - FR_{est} = PCLK / (16 * BR * DL_{est}) = **1,628**
- Como FR_{est} está dentro del rango [1.1, 1.9], buscamos los valores de DivAddVal y MulVal de la tabla anterior:
  - DivAddVal = 5 y MulVal = 8
- Finalmente DLM = 0 y DLL = 4.
- Si calculamos la velocidad de modulación, resulta de 115384 baudios, lo que representa un error de 0,16%.

Registro de control de línea (LCR)

- Configura los parámetros de la conexión.

Table 932. USART Line Control Register (LCR, addresses 0x4008 100C (USART0), 0x400C 100C (USART2), 0x400C 200C (USART3)) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	WLS		Word Length Select.	0
		0x0	5-bit character length.	
		0x1	6-bit character length.	
		0x2	7-bit character length.	
		0x3	8-bit character length.	
2	SBS		Stop Bit Select.	0
		0	1 stop bit.	
		1	2 stop bits (1.5 if LCR[1:0]=00).	
3	PE		Parity Enable	0
		0	Disable parity generation and checking.	
		1	Enable parity generation and checking.	
5:4	PS		Parity Select.	0
		0x0	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	
		0x1	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	
		0x2	Force HIGH. Forced 1 stick parity.	
		0x3	Force LOW. Forced 0 stick parity.	
6	BC		Break Control.	0
		0	Disabled. Disable break transmission.	
		1	Enabled. Enable break transmission. Output pin USART TXD is forced to logic 0 when LCR[6] is active high.	
7	DLAB		Divisor Latch Access Bit.	0
		0	Disabled. Disable access to Divisor Latches.	
		1	Enabled. Enable access to Divisor Latches.	
31:8	-	-	Reserved	-

Registro de estado de línea (LSR)

- Provee información sobre la conexión.

Table 933. USART Line Status Register Read Only (LSR, addresses 0x4008 1014 (USART0), 0x400C 1014 (USART2), 0x400C 2014 (USART3)) bit description

Bit	Symbol	Value	Description	Reset Value
0	RDR		Receiver Data Ready. LSR[0] is set when the RBR holds an unread character and is cleared when the USART RBR FIFO is empty.	0
		0	Empty. RBR is empty.	
		1	Data. RBR contains valid data.	
1	OE		Overrun Error. The overrun error condition is set as soon as it occurs. A LSR read clears LSR[1]. LSR[1] is set when USART RSR has a new character assembled and the USART RBR FIFO is full. In this case, the USART RBR FIFO will not be overwritten and the character in the USART RSR will be lost.	0
		0	Inactive. Overrun error status is inactive.	
		1	Active. Overrun error status is active.	
2	PE		Parity Error. When the parity bit of a received character is in the wrong state, a parity error occurs. A LSR read clears LSR[2]. Time of parity error detection is dependent on FCR[0]. **Note:** A parity error is associated with the character at the top of the USART RBR FIFO.	0
		0	Inactive. Parity error status is inactive.	
		1	Active. Parity error status is active.	
3	FE		Framing Error. When the stop bit of a received character is a logic 0, a framing error occurs. A LSR read clears LSR[3]. The time of the framing error detection is dependent on FCR0. Upon detection of a framing error, the RX will attempt to re-synchronize to the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no Framing Error. **Note:** A framing error is associated with the character at the top of the USART RBR FIFO.	0
		0	Inactive. Framing error status is inactive.	
		1	Active. Framing error status is active.	

Registro de estado de línea (LSR)

- Provee información sobre la conexión.

Table 933. USART Line Status Register Read Only (LSR, addresses 0x4008 1014 (USART0), 0x400C 1014 (USART2), 0x400C 2014 (USART3)) bit description *...continued*

Bit	Symbol	Value	Description	Reset Value
4	BI		Break Interrupt. When RXD1 is held in the spacing state (all zeros) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until RXD1 goes to marking state (all ones). A LSR read clears this status bit. The time of break detection is dependent on FCR[0]. **Note:** The break interrupt is associated with the character at the top of the USART RBR FIFO.	0
		0	Inactive. Break interrupt status is inactive.	
		1	Active. Break interrupt status is active.	
5	THRE		Transmitter Holding Register Empty. THRE is set immediately upon detection of an empty USART THR and is cleared on a THR write.	1
		0	Not empty. THR contains valid data.	
		1	Empty. THR is empty.	
6	TEMT		Transmitter Empty. TEMT is set when both THR and TSR are empty; TEMT is cleared when either the TSR or the THR contain valid data.	1
		0	Not empty. THR and/or the TSR contains valid data.	
		1	Empty. THR and the TSR are empty.	
7	RXFE		Error in RX FIFO. LSR[7] is set when a character with a RX error such as framing error, parity error or break interrupt, is loaded into the RBR. This bit is cleared when the LSR register is read and there are no subsequent errors in the USART FIFO.	0
		0	No error. RBR contains no USART RX errors or FCR[0]=0.	
		1	Error. USART RBR contains at least one USART RX error.	
8	TXERR		Error in transmitted character. A NACK response is given by the receiver in Smart card T=0 mode. This bit is cleared when the LSR register is read.	0
		0	No error. No error (normal default condition).	
		1	NACK. A NACK response is received during Smart card T=0 operation.	
31:9	-	-	Reserved	-

Resumen

- Transmisión Serie Asincrónica:
  - Formato de la trama
  - Parámetros de la comunicación: velocidad de modulación y D/P/S.
  - Sincronización de relojes:
    - Oversampling, con factor *k* (=16).
    - Degradación de la señal, principalmente por aumento de velocidad de modulación y de la distancia.
  - Interfaces de modo común y balanceado.
- UART: implementación básica de interfaz serial en dispositivos.
  - Terminales principales: TX y RX.
  - Como todo dispositivo, posee registros de configuración y de control, mapeados en memoria a partir de una dirección base y manejados por interrupciones.
  - Registros destacados:
    - RBR y THR: usados para la recepción y para la transmisión, respectivamente.
    - DLL, DLM y FDR: usados para establecer la velocidad de modulación.
    - LCR y LSR: usados para configurar y controlar el estado de la comunicación

Agradecimientos

- Las diapositivas de este tema fueron basadas en las realizadas por el Ing. Esteban Volentini y por el Ing. Nicolás Majores Padilla.