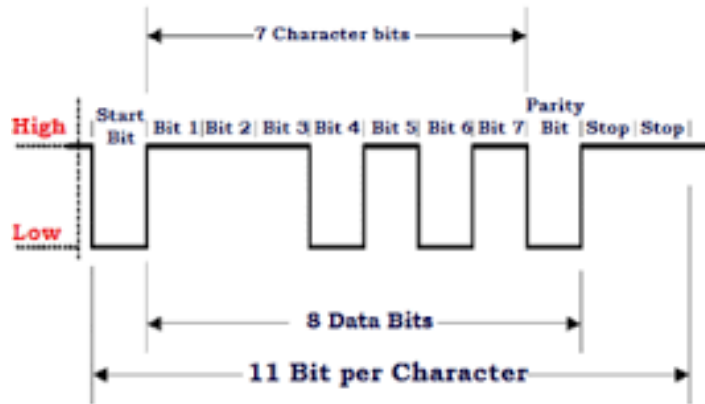


Laboratorio de Microcomputadoras

Comunicación serial USART

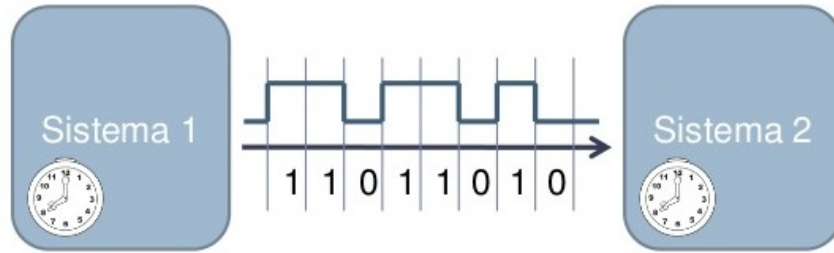


Diana A. Cruz Hernández
Amaranto de J. Dávila Jáuregui

Antecedentes

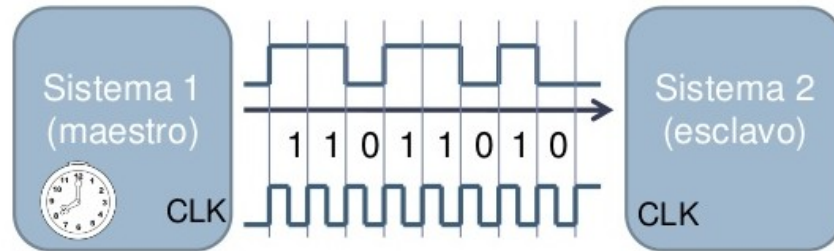
- Las microcomputadoras de la familia PIC16 cuentan con un módulo de comunicaciones seriales USART
- Este módulo permite establecer interfaces de alta velocidad necesarias para que la microcomputadora se comuniquen con otros dispositivos

Comunicación síncrona y asíncrona



➤ Asíncrona:

- Cada sistema tiene su propio reloj local.
- Sincronización mediante bit de Start y Stop.
- Sólo 1 hilo.



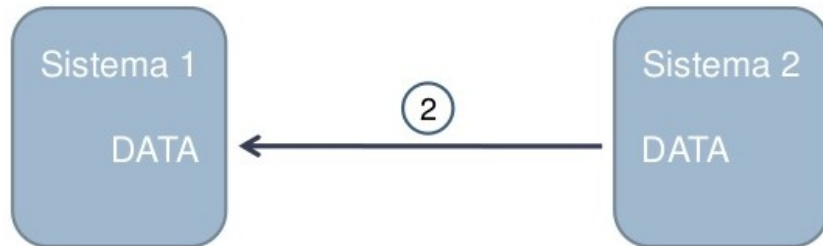
➤ Síncrona:

- Una señal de reloj común.
- El *maestro* genera el reloj.
- Dos hilos.
- Velocidades mayores.

Full duplex y Half duplex



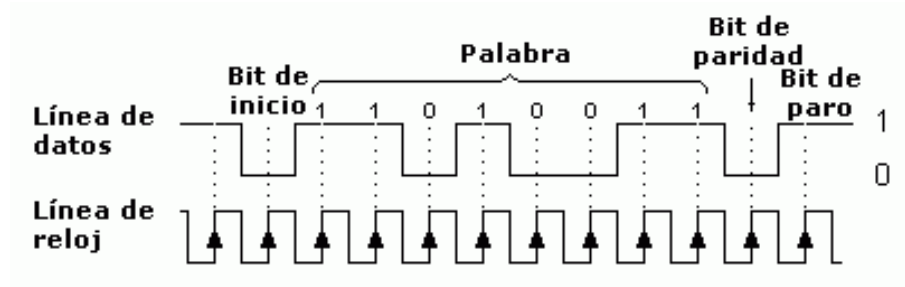
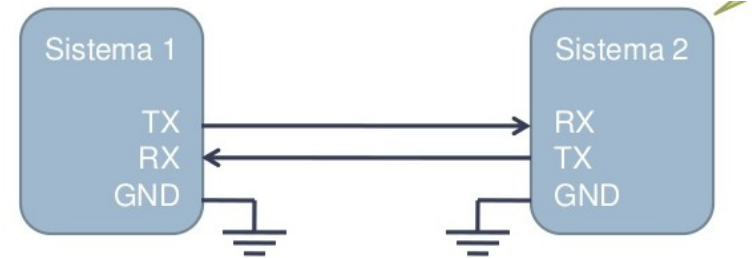
- Full-duplex
 - Comunicación bidireccional simultanea.
 - Dos canales de datos (TX, RX).
 - Dos hilos.



- Half-duplex
 - Comunicación bidireccional multiplexada en el tiempo.
 - Un único canal (DATA).
 - Primero en un sentido, luego en el otro (protocolo).
 - Un hilo

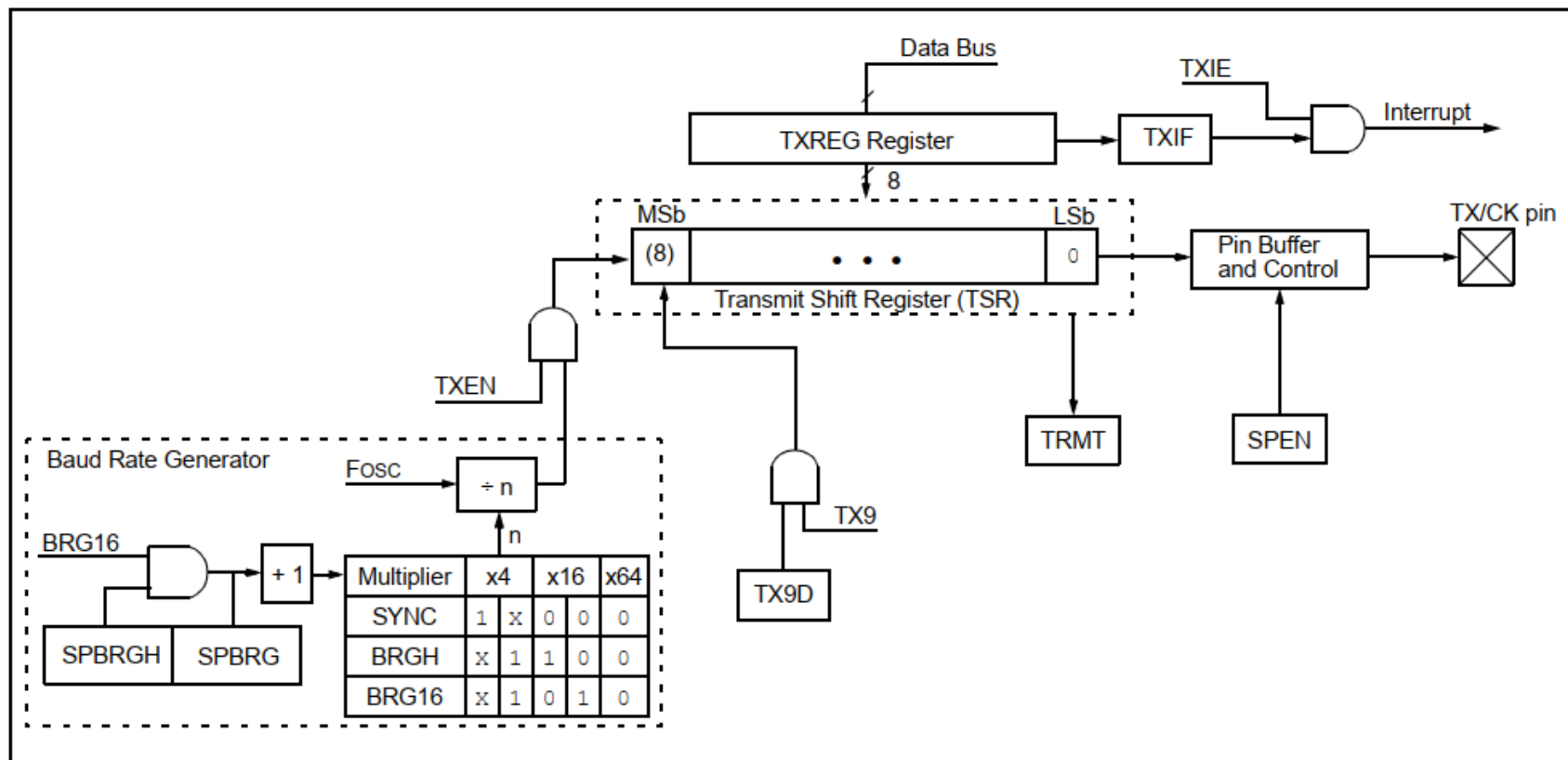
USART

- Comunicación por medio de dos buses
- Una línea de referencia, GND
- 8 bits de datos
- 1 bit de parada 1 bit de inicio
- Sin paridad



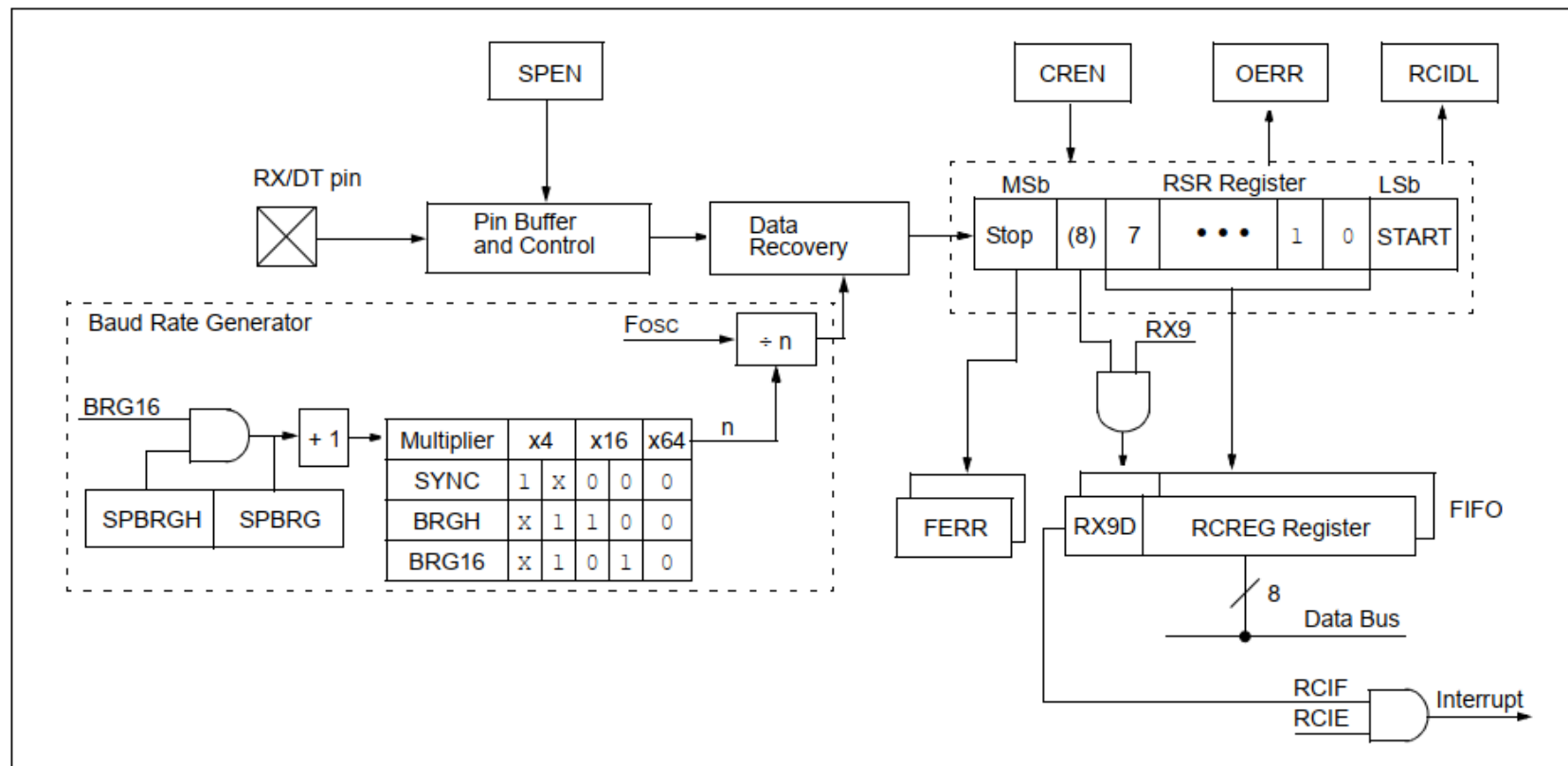
Módulo Usart PIC16 Transmisor.

FIGURE 12-1: EUSART TRANSMIT BLOCK DIAGRAM



Módulo Usart PIC16 Receptor

FIGURE 12-2: EUSART RECEIVE BLOCK DIAGRAM



Baud Rate

Número de símbolos transmitidos por segundo

TABLE 12-3: 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	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

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

Registros

TABLE 12-1: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	159
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	32
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	33
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	35
RCREG	EUSART Receive Data Register								155
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	158
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	160
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	160
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	54
TXREG	EUSART Transmit Data Register								150
TXSTA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	157

Legend: x = unknown, — = unimplemented read as '0'. Shaded cells are not used for Asynchronous Transmission.

Pasos para configurar la USART

1) Configurar el generador de bauds

1) Inicializar el valor de los registros SPBRGH y SPBRG

Son dos registros de 8 bits cada uno, en ellos se escribe el valor obtenido de la tabla 12-3 o de las tablas 12-5 (página siguiente).

2) Establecer o borrar los bits BRGH y BRG16

Podemos encontrar en la hoja de datos tablas (página siguiente) con valores ya calculados para diferentes valores de oscilador

Valores a utilizar

TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.2k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

Ejemplo. Configurar el módulo USART de PIC16F886 a una velocidad de **9600 bauds**. Considerar que la frecuencia de oscilación del sistema es de **4 Mhz**.

De la tabla vemos que a 4 Mhz el valor del **SPBRGH:SPBRG = 25 = 0x0019**, con el bit **SYNC=0, BRGH=1, BRG16=0**, entonces **SPBRGH=0x00** y **SPBRG=0x19**

Pasos para configurar la USART

2) Habilitar el puerto serial asíncrono

1) Limpiando el bit SYNC y

2) Encendiendo el bit SPEN

El bit SYNC se encuentran en el registro TXSTA (bit 4) y el bit SPEN en el registro RCSTA (bit 7), el bit SYNC es cero ya que se establece una comunicación asíncrona, y el bit SPEN habilita los buffer de entrada / salida de los pines de la microcomputadora en donde se conectan los buses TX/RX que son RC6 y RC7 respectivamente.

Pasos para configurar la USART

3) Si se desea el noveno bit de transmisión, este bit se utiliza para direccionamiento de esclavos, es este caso no se va a utilizar.

- Se enciende el bit TX9 y/o RX9

El noveno bit (TX9D/RX9D) se leerá o escribirá respectivamente en el bit menos significativo de los registros TXSTA y RCSTA y corresponde con el bit más significativo de la palabra que se recibe o envía

Pasos para configurar la USART

4) Habilitar la transmisión y/o recepción

1) Transmisión

1) Encender el bit TXEN, lo cual encenderá la bandera TXIF

2) Recepción

1) Encender el bit CREN, la bandera RCIF se encenderá cuando se reciba un caracter

El bit TXEN se encuentran en el registro TXSTA (bit 5) y el bit CREN en el registro RCSTA (bit 4)

Pasos para configurar la USART

5) Lectura y/o escritura del dato a enviar o recibir

1) Transmisión

- 1) Escribir el valor a transmitir en el registro TXREG
- 2) Esperar a que el bit TRMT se encienda lo que indica que se terminó de transmitirse el dato, cuando TRMT=1, se puede enviar un nuevo dato.

2) Recepción

- 1) Esperar a que el bit RCIF se encienda indicando que se recibió un dato
- 2) Leer el valor recibido en el RCREG, lo que hará que el bit RCIF se apague automáticamente.

Ejemplos.

Ejemplo1

Programa que envía de manera asíncrona través de la USART una cadena de caracteres. Esta cadena puede ser recibida mediante el puerto serie RS232 de una PC usando un software de comunicación tal como la hiperterminal de windows, en este caso se va a utilizar la terminal virtual de Proteus.

Para probar el programa cargar el archivo `transmite.HEX` en la microcomputadora de Proteus (archivo `ejemplos_serial`)

Ejemplo 2

Programa que hace el eco del carácter recibido, conforme recibe un carácter del puerto serie, lo regresa sin cambio por el mismo puerto. El proceso se detiene cuando el carácter recibido es un <Esc> o código ASCII 1Bh

Para probar el programa cargar el archivo `eco.HEX` en la microcomputadora de Proteus (archivo `ejemplos_serial`)

Práctica

Diseñar un programa que reciba una cadena de caracteres y la compare con la palabra clave “pumas” (o cualquier otra elegida arbitrariamente), cuando la cadena recibida coincide con la palabra clave encenderá un Led conectado a la línea RB0 del puerto B y enviará a través del puerto serial la palabra “OK”. De lo contrario, (cuando reciba cualquier otra cadena) apagará el LED y enviará la palabra “Error”.