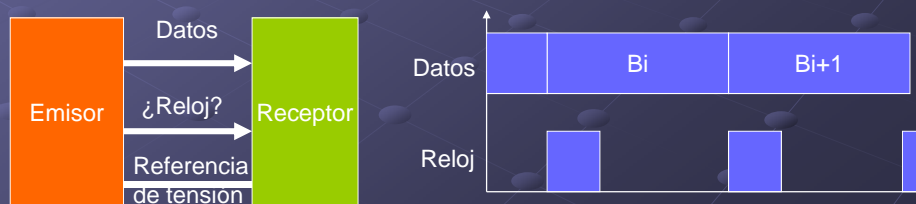


# LOS PIC16F87X

## Comunicación serie asíncrona USART

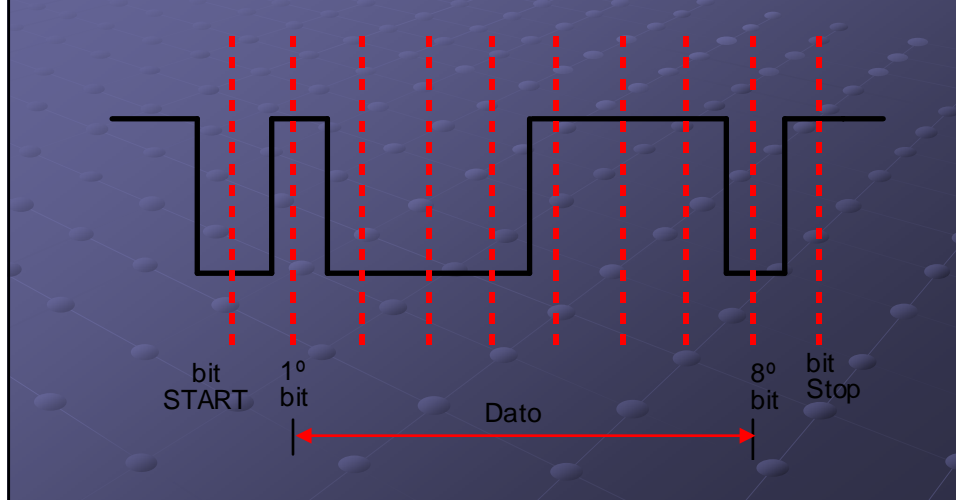
### Comunicación Serie

- Los datos se envían bit a bit por una misma línea y durante un tiempo fijo.
- Velocidad de transmisión: es el número de bits por segundo (baudios)
- Transferencia síncrona: se envía señal de reloj para sincronizar cada bits.
- Transferencia Asíncrona: no se envía señal de reloj. Se necesitan relojes en el emisor y en el receptor de la misma frecuencia y fase.



## Comunicación Serie asíncrona

Ejemplo de transmisión asíncrona

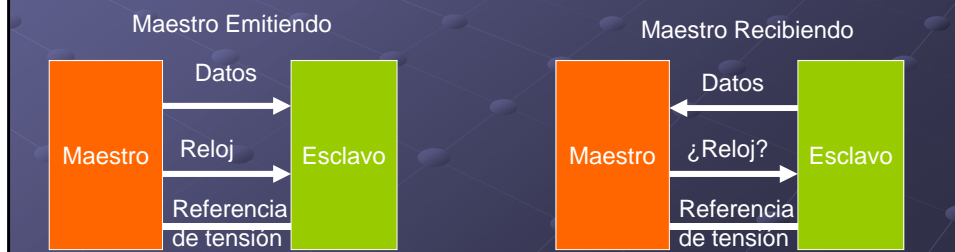


## Comunicación Serie

- Se emplean dos registros de desplazamiento (uno es el emisor y otro en el receptor) encadenados para la conversión paralelo/serie en la emisión y la serie/paralelo en la recepción.
- Sincronizaciones:
  - De los sucesivos bits
  - De cada paquete de bits (8 o 9 bits)
- Se envía la señal de reloj si la distancia entre Emisor y Receptor es corta: menores retardos en las transiciones y menores flancos en la señal de reloj recibida.
- Codificaciones posibles en cada bits
  - NRZ (nivel alto =1 / nivel bajo = 0)
  - NRZI (cambio de nivel:1 / sin cambio de nivel:0)
  - RZ (impulso:1 / sin impulso:0)
  - ...

## Transferencia Síncrona

- Dispositivo Maestro: el que genera la señal de reloj, es el que tiene capacidad de iniciar o finalizar una transferencia.
- Dispositivo esclavo: recibe la señal de reloj, no tiene capacidad para iniciar la transferencia de información.
- Es posible una transmisión continua de bits, no hay límite en tamaño de datos.



## Transferencia Asíncrona

- Se emplean relojes de igual frecuencia (se acuerda y configura la velocidad de transmisión) pero es necesario que estén en fase (sincronizados)
- Cada paquete de bits de tamaño fijo se "enmarca" con bits de arranque y de parada que sirven para sincronizar los relojes del emisor y del receptor.
- La línea de datos inactiva a "1", si se desea enviar un dato se manda un bits de arranque que sitúa a "0" la línea durante el tiempo correspondiente a un bits.
- Al finalizar el envío de un dato, la línea se sitúa a "1" al menos durante el tiempo de un bit: bit de parada.



## Módulos de comunicación serie en los microcontroladores PIC

### ● Módulo SCI (Serial Communication Interface) ó USART (Universal Synchronous Asynchronous Receiver Transmitter)

- Dos pines asignados para la comunicación
- Modo Asíncrono (full duplex)
- Modo Síncrono (semi duplex)

RC6/TX/CK  
RC7/RX/DT

### ● Módulo SSP (Synchronous Serial Port)

- Cuatro pines asignados para la comunicación

- Interface Síncrono

- SPI (Serial Peripheral Interface) full dúplex
- I2C (Inter-Integrated Circuit) semi duplex

RC5/SDO  
RC4/SDI  
RC3/SCK  
RA5/SS

## Módulo SCI

Configurable en tres modos de trabajo

### 1. Asíncrono (full dúplex)

- Recepción y transmisión independientes compartiendo generador de relación de baudios (BGR)
  - TX: pin de transmisión (Salida)
  - RX: pin de recepción (Entrada)

### 2. Síncrono modo Maestro (semi dúplex)

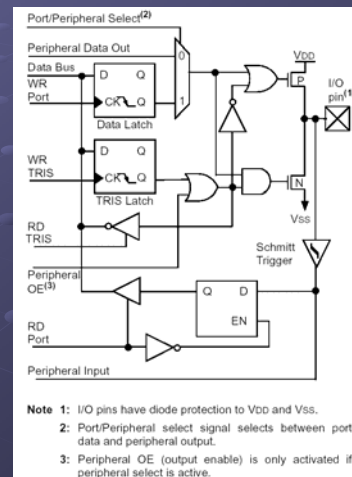
- CK: reloj generado por el PIC (salida)
- DT: datos entrantes (recepción) o salientes (Transmisión)

### 3. Síncrono modo Esclavo (semi duplex)

- CK: reloj entrante
- DT: datos entrantes (recepción) o salientes (Transmisión)

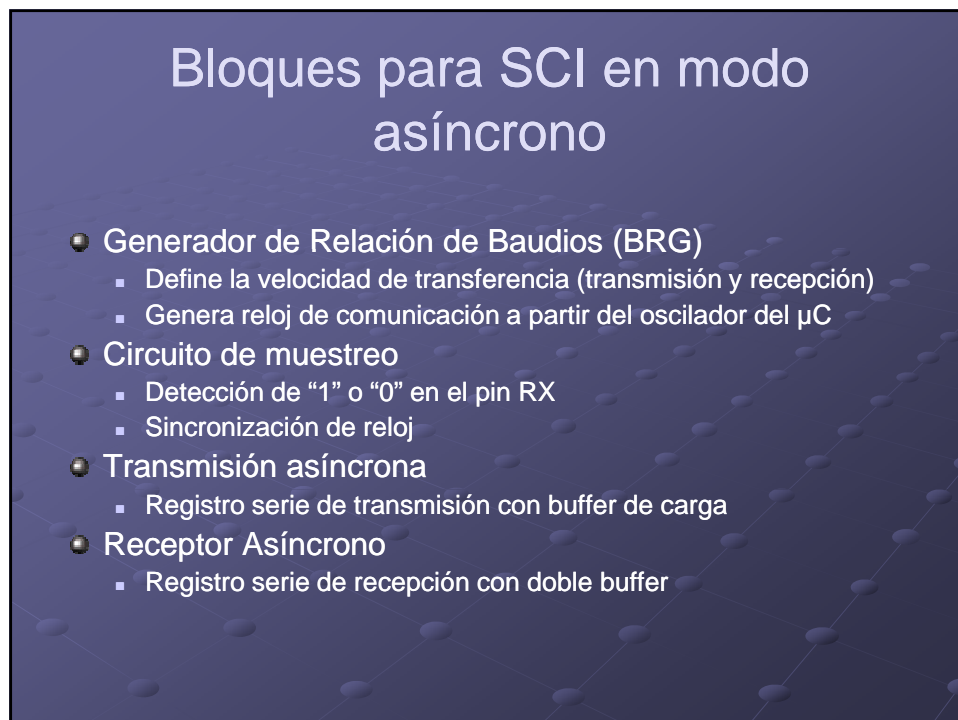
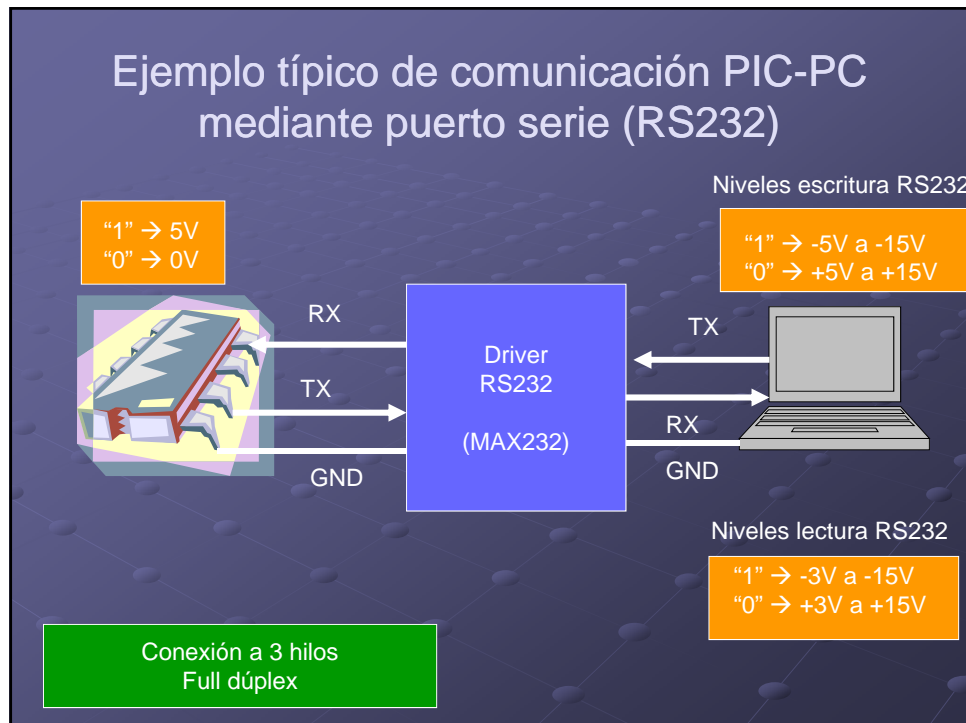
## Módulo SCI

- Los pines relacionados con las comunicación no es necesario configurarlos como salida en el TRISC, si está operativo el módulo prevalecen sobre el registro de dirección de datos.
- Registros asociados al SCI:
  - Registro de control de estado **TXSTA & RCSTA**
  - Registro de Relación de Baudios **SPBRG**
  - Registro de datos de transmisión **TXREG**
  - Registro de recepción de datos **RCSTA**



## Módulo SCI en la transmisión Asíncrona

- Es la conexión más adecuada para la comunicación con un equipo alejado
- Los niveles lógicos de las señales se corresponden con los niveles eléctricos de alimentación del microcontrolador
- Hay varias normas de transmisión asíncrona: RS232, RS485, RS422... Que emplean niveles de tensión más inmunes al ruido (RS232) o que emplean tensiones diferenciales (RS485, RS422) y que son más apropiadas para distancias largas entre dispositivos.
- Para implementar estas transmisiones, sería necesario la adaptación de niveles eléctricos mediante los correspondientes circuitos integrados de adaptación (*drivers* o *transceivers*)
- Si los microcontroladores a comunicar están cercanos podría obviarse esa adaptación.



## Generador de Relación de Baudios (BGR)

- Se emplea para determinar la frecuencia de reloj para los registros de desplazamiento de los bloques de transmisión y de recepción.
- Depende del valor X cargado en el registro de generación de la relación de baudios **SPBRG** (dirección 0x99) y del estado del registro **TXSTA**.

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit7							bit 0

1= Modo Síncrono ó 0= modo asíncrono

BRGH = 1 (velocidad alta):baudios =  $fosc/16*(x+1)$   
 BRGH = 0 (Velocidad baja):baudios =  $fosc/(64*(x+1))$

Nota.- Esto es valido para SCI Asíncrono (en modo síncrono BRGH se ignora y es otra formula)

## Registros asociados al Generador de Baud Rate

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR.BOR	Valor en el resto de Reset
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000-010	0000-010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
99h	SPGBRG	Registro generador de Baud Rate									

## Generador de Relación de Baudios

- Al ser SPBRG un registro de 8 bits, puede que no sea posible alcanzar de manera exacta la velocidad de transmisión deseada
- Calculo del Error = (valor calculado – valor deseado) / Valor deseado
- Ejemplo:
  - Valor deseado = 9600 baudios
  - Fosc = 16 MHz
  - BRG = 0  $\rightarrow$   $9600 = 16000000 / (64 * (x+1))$   
 $x = 25,042 \rightarrow$  SPBRG = 25
  - Valor calculado =  $16000000 / (64 + (25+1)) = 9615$  baudios
  - Error =  $((9615-9600)/9600 = 0,16\%$
- Puede alcanzarse menor error configurando velocidad alta (BRGH=1).

## Ejemplos con BRGH=0

**TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)**

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0



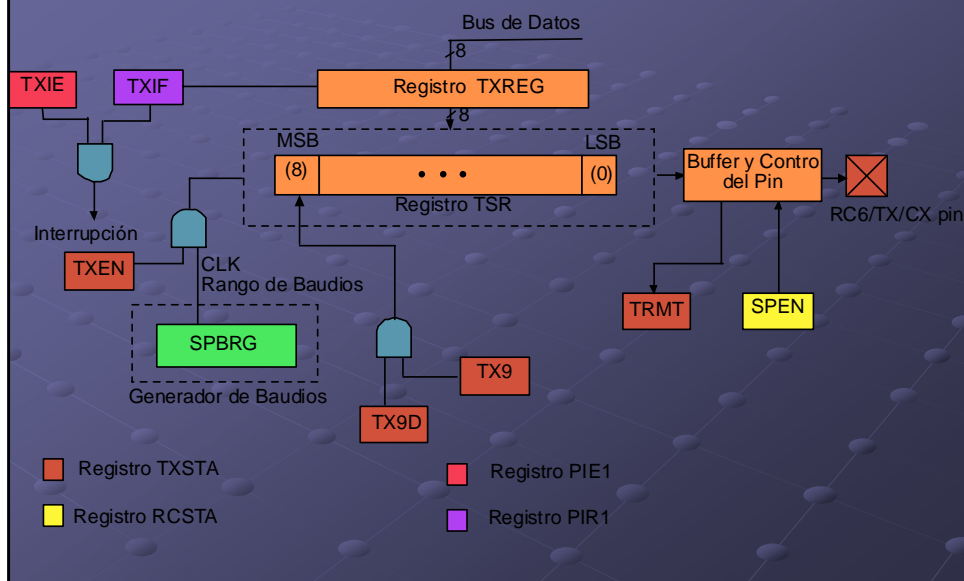
## Ejemplos con BRGH=1

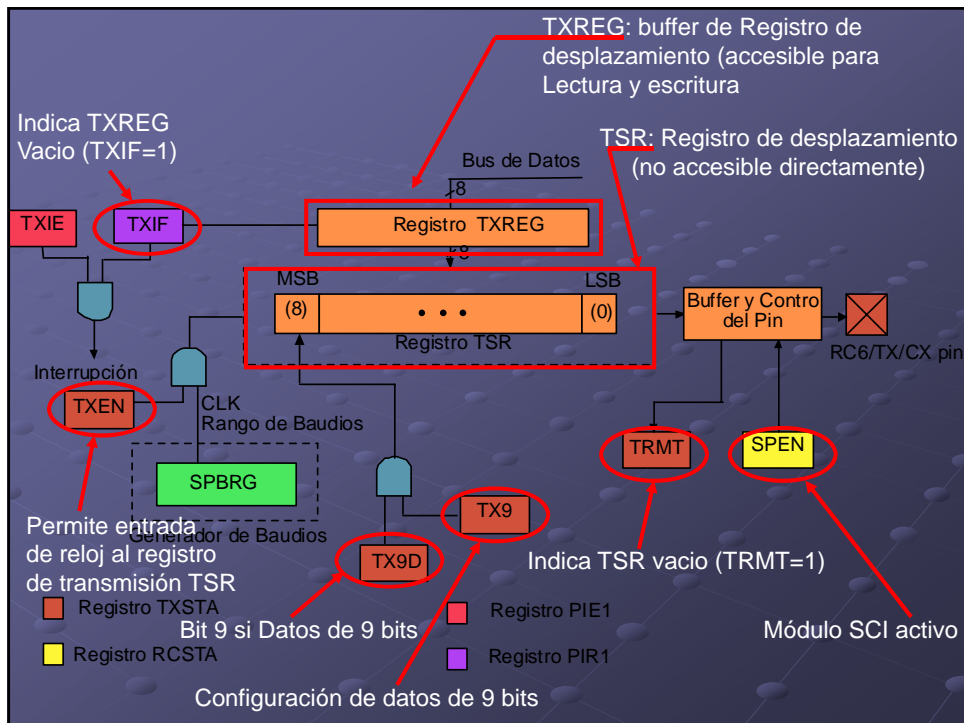
BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

## Comunicación Serie Asíncrona

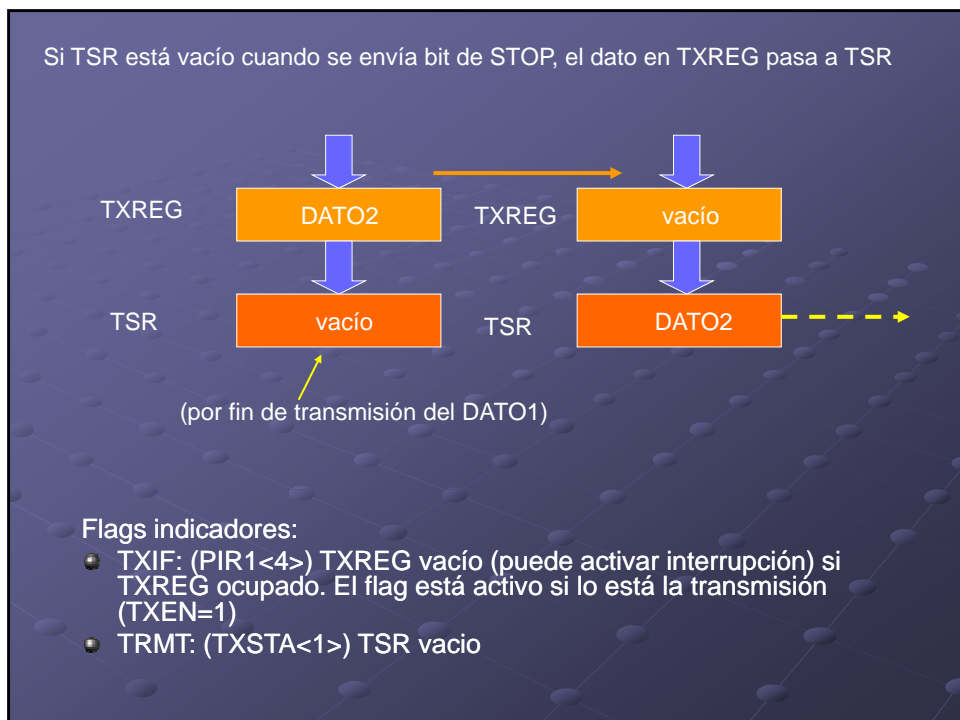
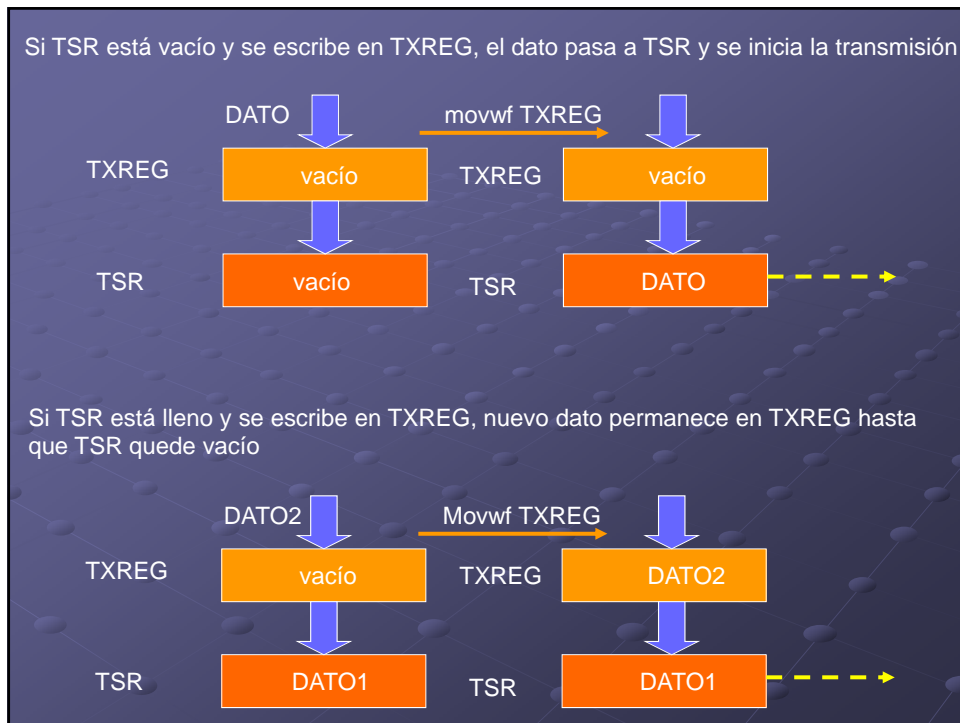




## Comunicación Serie Asíncrona

Pasos para realizar una transmisión:

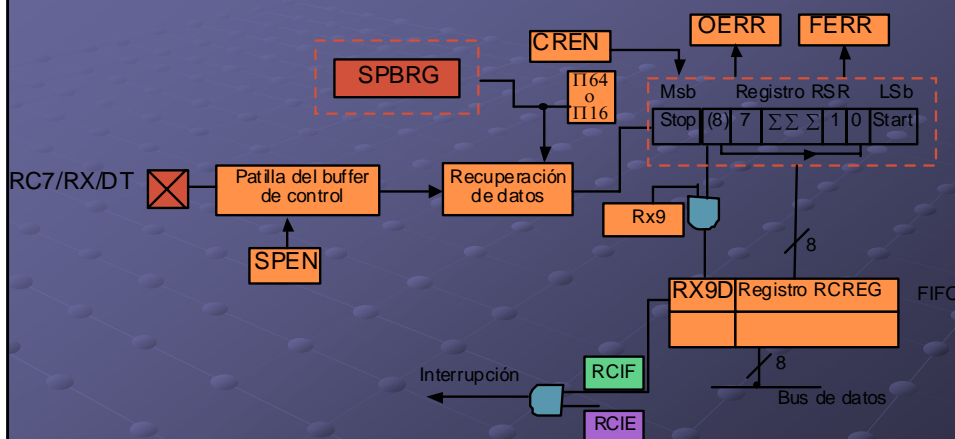
- Configurar las líneas **RC6/TX/CK** como salida y **RC7/RX/DT** como entrada
- Inicializar el registro **SPBRG** para generar la frecuencia adecuada, con el valor correspondiente en **BRGH**
- Poner **SYNC="0"** y **SPEN="1"** para activar el USART en modo asíncrono.
- Para activar interrupción **TXIE="1"** (si se desea)
- Si datos de 9 bits, **TX9="1"** y el noveno bits valdrá "1" pues TXREG está vacío.
- Habilita la transmisión poniendo a uno **TXEN**, el cual pondrá **TXIF** a "1"
- Cargar dato a transmitir **TXREG** comienza la transmisión
- Si se usa interrupción, asegurar que los bits **GIE** y **PEIE** del registro **INTCON** están a "1".

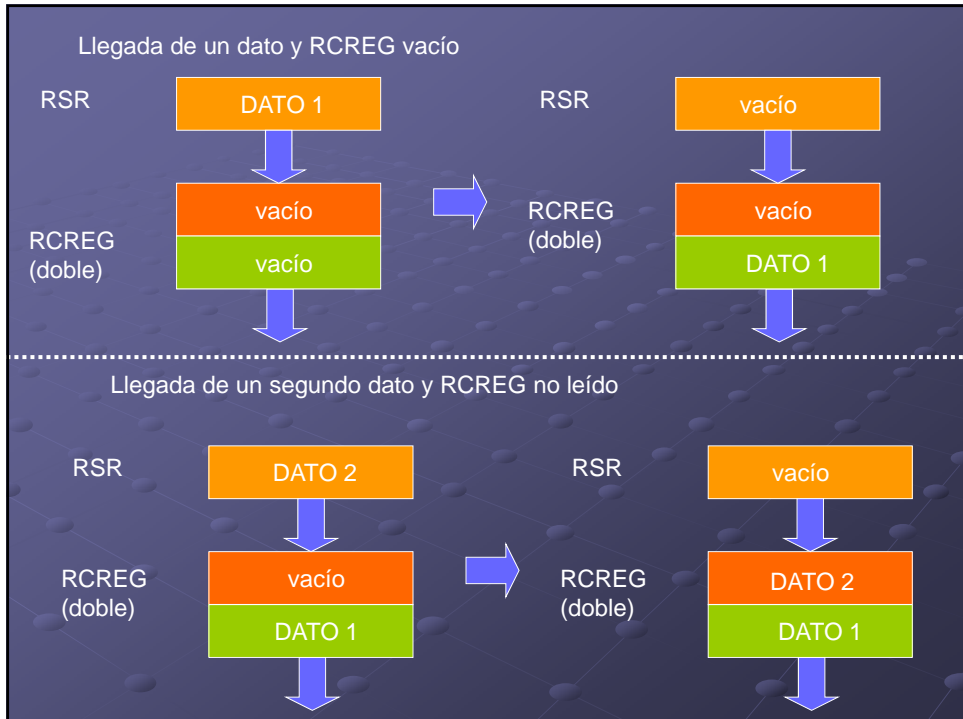
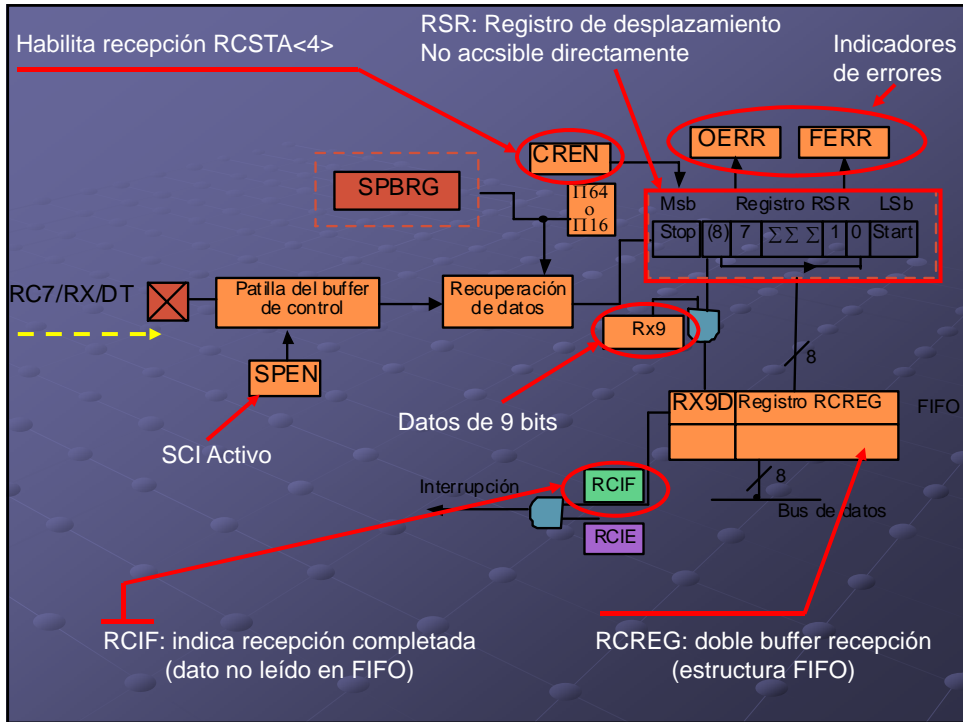


## Registros asociados a la transmisión asíncrona

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR,BOR	Valor en el resto de reset
0Bh, 8Bh, 10Bh, 8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	ROIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	--	FERR	OERR	RX9D	0000 -00X	0000 -00X
19h	TXREG	REGISTRO DE TRANSMISIÓN DEL USART								0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	--	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	REGISTRO GENERADOR DE RATIO DE BAUDIOS								0000 0000	0000 0000

## Receptor asíncrono





Llegada de un tercer dato y RCREG doble no leído

RSR

DATO 3

RCREG (doble)

DATO 2

DATO 1

Error de Overrun :OERR=1 (grave)

Se pierde el DATO 3

Bloqueo total de la Recepción

Se debe resetear el sistema de Recepción CREN=0 y luego CREN=1

**Flags indicadores**

- **RCIF:** (PIR1<5>) (puede activar la interrupción). Indica dato/s disponible/s para lectura en RCREG. Se pone a cero automáticamente (no por software) cuando RCREG está vacío (el doble buffer debe estarlo).
- **OERR(=RCSTA<1>) Error de Overrun:** hay que resetear el receptor para volver a "0".
- **FERR:( RCSTA<2>) Error de trama:** el bit de STOP debería ser "0" pero lee "1" (También presenta un doble buffer FIFO).

## Cronograma de recepción de datos

Llegada de un primer dato:  
bit de STOP

Llegada de un segundo dato:  
bit de STOP

Tercer dato  
Sin haber realizado lectura  
(error de Overrun)

RX (pin)

START bit bit0 bit1 bit7/8 STOP bit

START bit bit0 bit1 bit7/8 STOP bit

START bit bit0 bit1 bit7/8 STOP bit

Rcv Shift Reg

Rcv Buffer Reg

Read Rcv Buffer Reg RCREG

RCIF (Interrupt Flag)

OERR bit

CREN

Word 1 RCREG

Word 2 RCREG

Lecturas posteriores a la llegada del 3er dato

Reset de Recepción Para volver OERR a "0"

RCREG vacío al principio

## Comunicación Serie Asíncrona

Pasos para realizar una Recepción:

- Cargar el registro SPBRG para trabajar con la frecuencia deseada.
- Poner SYNC = "0" y SPEN = "1" para activar el USART en modo asíncrono
- Para activar interrupción con la llegada del bit de parada, RCIE = "1"
- Si transmisión de 9 bits, poner bit RX9 = "1".
- Habilitar la recepción poniendo CREN = "1"
- Al completarse la recepción, RCIF se pone a "1" y se produce una interrupción si está permitida
- Leer el registro RCSTA para coger el bit 9 y ver si hay errores en la transmisión.
- Leer los 8 bits del RCREG
- Si hubo algún error, pone CREN=0
- Si se usa la interrupción, asegurar que GIE y PEIE de INTCON están a "1".

## Registros asociados a la recepción Asíncrona

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Bh, 8Bh 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMREIF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00X	0000 000X
1Ah	RCREG	Registro Receptor USAR								0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Registro generador de Baud Rate									

## Registro TXSTA (98h)

Registro de estado y control (Transmisor)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
<b>CSRC</b>	<b>TX9</b>	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit7							bit 0

**CSRC:** Bit de selección de reloj

**Modo Asíncrono**

Sin importancia

**Modo Síncrono**

1 = Modo master (reloj generado internamente para BRG )

0 = Modo esclavo (reloj externo)

**TX9 :** Habilita transmisión de 9-8 bits

1 = Transmisión de 9 bits

0 = Transmisión de 8 bits

## Registro TXSTA (98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSRC	TX9	<b>TXEN</b>	<b>SYNC</b>	—	BRGH	TRMT	TX9D
bit7							bit 0

**TXEN:** Bit de habilitación del transmisor

1 = Habilita transmisión

0 = Deshabilita transmisión

**Nota:** **SREN/CREN** anula **TXEN** en modo **SYNC**.

**SYNC:** Bit de selección del modo USART

1 = Modo Síncrono

0 = Modo Asíncrono



## Registro TXSTA (98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSRC	TX9	TXEN	SYNC	—	<b>BRGH</b>	TRMT	TX9D
bit7							bit 0

### **BRGH:** Bit de selección del rango de baudios alto

#### Modo Asíncrono

1 = Alta velocidad

0 = Baja velocidad

#### Modo Síncrono

No se usa este modo

## Registro TXSTA (98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	<b>TRMT</b>	<b>TX9D</b>
bit7							bit 0

### **TRMT:** Bit de estado del registro de desplazamiento del transmisor (TSR)

1 = TSR vacío

0 = TSR lleno

### **TX9D:** Noveno bit de datos de transmisión. Se puede emplear como bit de paridad.

## Registro RCSTA (18h)

### Registro de estado y control (Receptor)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>SPEN</b>	<b>RX9</b>	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit 0

#### **SPEN: Bit de habilitación del puerto serie**

1 = Habilita puerto serie (configura patillas **RC7/RX/DT** y **RC6/TX/CK** para el puerto serie)

0 = Deshabilita puerto serie

#### **RX9: Habilita recepción de 9-8 bits**

1 = Selecciona 9 bits de recepción

0 = Selecciona 8 bits de recepción

## Registro RCSTA (18h)

### Registro de estado y control (Receptor)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPEN	RX9	<b>SREN</b>	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit 0

#### **SREN: Bit de habilitación de recepción única**

##### Modo asíncrono

Este modo no se usa

##### Modo Síncrono master

1 = Habilita una recepción única

0 = Deshabilita una recepción única

Este bit se borra después de completar la recepción

##### Modo Síncrono esclavo

En este modo no se usa

## Registro RCSTA (18h)

### Registro de estado y control (Receptor)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPEN	RX9	SREN	<b>CREN</b>	ADDEN	FERR	OERR	RX9D
bit7							bit 0

#### **CREN: Bit de habilitación de recepción continua**

##### Modo Asíncrono

- 1 = Habilita recepción continua
- 0 = Deshabilita recepción continua

##### Modo Síncrono

- 1 = Habilita recepción continua hasta que el bit **CREN** sea borrado (**CREN** anula **SREN**)
- 0 = Deshabilita la recepción continua

## Registro RCSTA (18h)

### Registro de estado y control (Receptor)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPEN	RX9	SREN	CREN	<b>ADDEN</b>	<b>FERR</b>	OERR	RX9D
bit7							bit 0

#### **ADDEN: Bit de dirección**

##### Modo Asíncrono con 9 bits (RX9=1)

- 1= Activa la detección de dirección, activa la interrupción y descarga el buffer de recepción al activarse RSR<8>.
- 0= Desactiva la detección de dirección, todos los bits son recibidos y el noveno bit puede utilizarse como bit de paridad.

#### **FERR: Bit de error de empaquetamiento**

- 1 = Error de empaquetamiento (Puede actualizarse al leer el registro **RCREG** y recibir el siguiente byte)
- 0 = No hay error

## Registro RCSTA (18h)

Registro de estado y control (Receptor)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPEN	RX9	SREN	CREN	ADDEN	FERR	<b>OERR</b>	<b>RX9D</b>
bit7							bit 0

### **OERR: Bit de error de desbordamiento**

1 = Error de desbordamiento (puede ser borrado al borrar **CREN**).

0 = No hay desbordamiento

### **RX9D: Noveno bit de datos de recepción.**

Se puede emplear como bit de paridad

## Comunicación serie asíncrona

Los PIC16F87X poseen un módulo de comunicaciones Asíncrona (USART), elSCI.

- SCI (Serial Communication Interface)
  - Comunicación Asíncrona full-duplex (bidireccional)
  - Comunicación a 2 hilos:
    - TX (Transmisión)
    - RX (Recepción)

## Comunicación serie asíncrona

- Protocolo asíncrono utilizado RS-232
- Cada dato se envía independientemente de los demás
- Datos de 8 o 9 bits
- Datos precedidos de un bit de arranque
- Datos terminados con un bit de parada
- Formato de bit NRZ (Non Return to Zero)
- Datos transmitidos a frecuencia fija y normalizada
- El primer bit que transmite y recibe es el LSB
- No soporta paridad por HW, debe hacerse por SW

## Comunicación serie asíncrona

- El módulo USART contiene:
  - Generador de frecuencia en baudios
  - Circuito de muestreo
  - Transmisor asíncrono
  - Receptor síncrono

## Comunicación Serie asíncrona

- **Generador de frecuencia en baudios (BRG)**
  - Valor de la frecuencia controlada por el contenido del registro SPBRG.
  - Se puede elegir entre alta y baja velocidad según el bit BRGH del registro TXSTA2
    - Si BRGH = "0", velocidad baja
    - Si BRGH = "1", velocidad alta