

Transmisión serie

RS232, I2C y SPI

Estándar RS-232

- Interface between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange
 - *EIA Recommended Standard number 232*
- Especificaciones de la conexión RS-232
 - Especificaciones eléctricas
 - Niveles de voltaje (-15v hasta +15v); un bit por baudio, forma de la señal que representa un 1 y un 0; el propósito o función de cada uno de los 25 pines que conforman la interface.
 - Tipo de conector: DB25, DB9
 - Protocolo
 - Método para iniciar y terminar el flujo de datos; método para coordinar al emisor y al receptor
 - Consideraciones NO especificadas en el estándar
 - La forma en que los caracteres se representan con bits

- Hasta no hace mucho tiempo era el método más popular para interconectar un ordenador (DTE) y un periférico o sensor (DCE)
- La versión más popular de RS-232 es la RS-232C. La versión más reciente es la RS-232E.
 - La recomendación ITU V.24 junto con la ITU V.28 son equivalentes a RS-232



DTE: Data Terminal Equipment
DCE: Data Communication Equipment

Niveles de tensión y su significado lógico: simbolos “1” y “0”

- Un uno binario se denomina “marca” (*mark*) y se representa por un voltaje de -3 a -15 voltios.
- Un cero binario se denomina espacio (*space*) y se representa por un voltaje de +3 a +15 voltios.
- Cualquier voltaje entre -3 y +3 voltios se considera inválido.
- Una corriente de corto circuito no puede exceder los 500mA



Señales balanceadas/no balanceadas

- Dos formas de poder realizar la conexión entre dos equipos
 - Interfaces balanceados. La información se transmite por dos cables (señal y /señal)
 - Interfaces no balanceados. Un único cable + masa
- El sistema balanceado se utiliza para transmisiones de mayor velocidad y distancia, en entornos donde se pueden producir interferencias

Conexión RS232

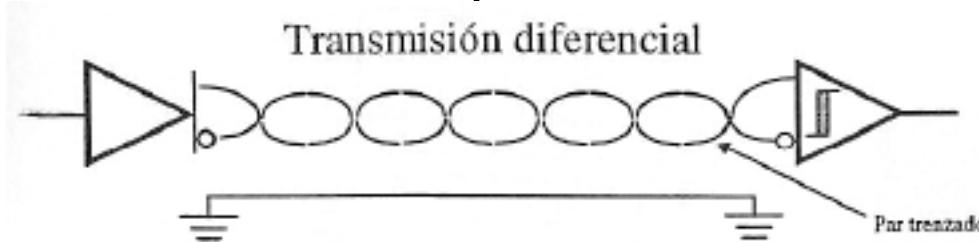
- RS-232
 - Red punto a punto
 - Transmisión no balanceada
 - Velocidad máxima de 120 Kbps (pocos metros)



- Ventajas: bajo coste y fácil de implantar.
- Desventajas: Le afecta el ruido industrial, corta distancia de linea, baja velocidad, desplazamiento de la masa.

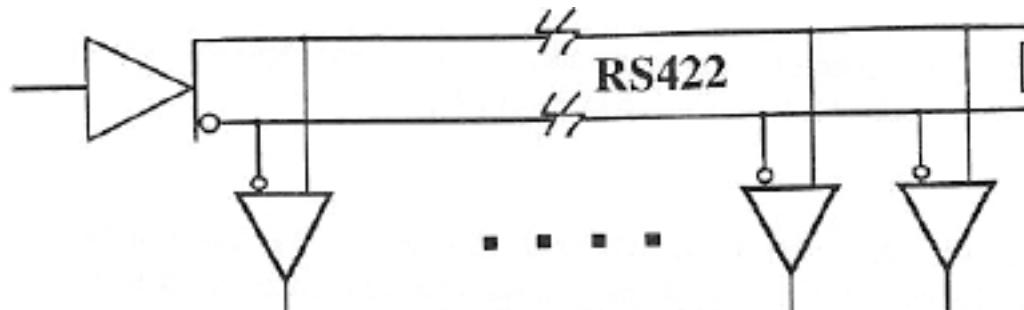
Conexión RS-422 y RS-485

- RS-422 y RS-485
 - Velocidad de transmisión de hasta 10 Mbps
 - Longitud máxima 1200 mts
 - Transmisión diferencial (alta inmunidad al ruido)
 - Ventajas: Gran rechazo al ruido, mayor velocidad, rechazo al desplazamiento de la masa, longitudes de linea grandes.
 - Desventajas: Mas complejo, mas hilos de cobre, mas coste.

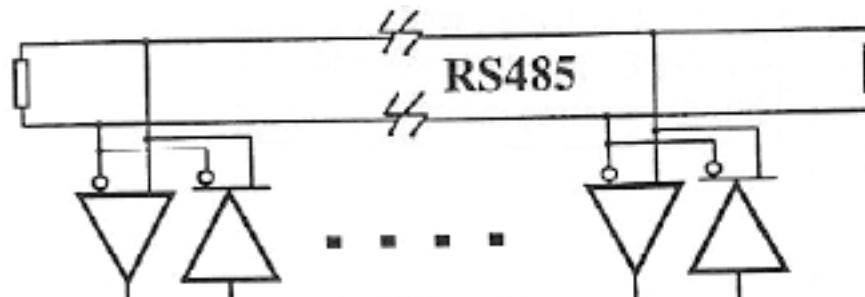


Modos de transmisión soportados

- RS-422: modo simplex, único transmisor y varios receptores (hasta 10)



- RS-485: modo half duplex, todos pueden emitir/ recibir información. Hasta 32 transceptores



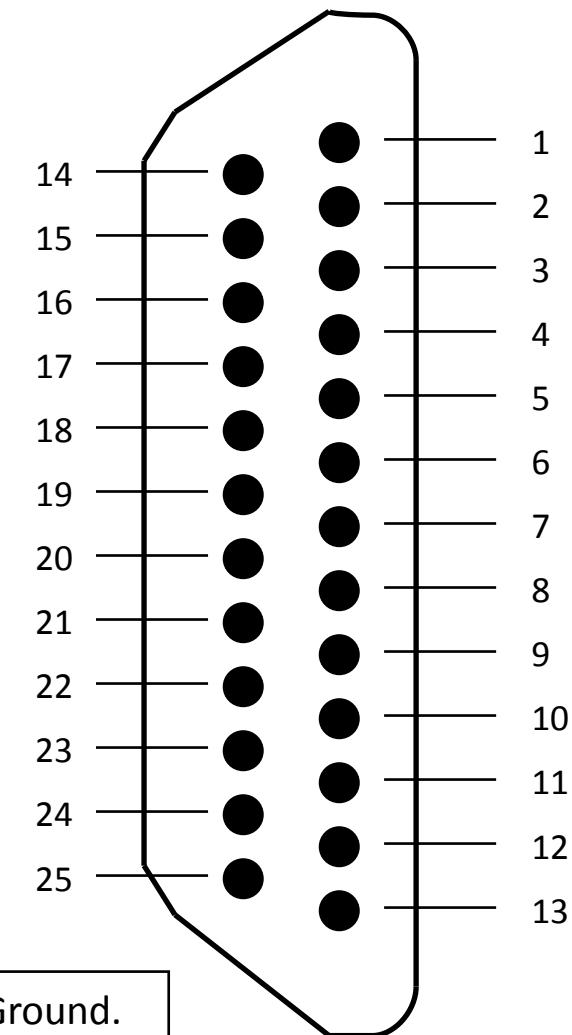
Conectores

- Tipicamente se emplean conectores normalizados de 9 pines o 25 pines.
- En la práctica, casi nunca se usan todos.



Interface RS-232 en un conector tipo D de 25 pines

No.pin	Descripción	Label
1	Protective ground (shield)	GND
2	Transmitted data	TD
3	Received data	RD
4	Request to send	RTS
5	Clear to send	CTS
6	Data set ready (DCE Ready)	DSR
7	Signal ground/Common return	SG
8	Primary carrier detect	CD
9	Positive DC Test Voltage	
10	Negative DC Test Voltage	
11	Unassigned	
12	Secondary carrier detect	
13	Secondary clear to send	
14	Secondary transmitted data	
15	DCE transmission signal timing	
16	Secondary received data	
17	Receiver signal timing	
18	(Local Loop Back)	
19	Secondary request to send	
20	Data terminal ready (DTE Ready)	DTR
21	Signal quality detector (Remote lookback)	CG
22	Ring indicator	RI
23	Data signal rate selector	CH/C1
24	DTE transmit signal timing	
25	Busy	



El circuito más simple en RS-232 sólo requiere dos pines: Signal y Ground. Normalmente el *two-wire RS-232* se implementa en palmtops.

Correspondencia entre un conector de 9 pines (DB-9) y uno de 25 pines (DB-25)

DB-9	Descripción	DB-25
1	Carrier Detect	8
2	Receive Data	3
3	Transmitted Data	2
4	Data Terminal Ready	20
5	Signal Ground	7
6	Data Set Ready	6
7	Request To Send	4
8	Clear To Send	5
9	Ring Indicator	22

Esta tabla sirve para construir un conversor de 25 a 9 pines.
Por ejemplo, el hilo para *carrier detect* debe ser soldado en el pin 1 del conector DB-9 y en el pin 8 del conector DB-25

Representación de caracteres

- RS-232 NO dice como representar caracteres (7 u 8 bits es la forma más común, pero podrían ser 5 ó 6). Cuando no se envían datos la señal se debe mantener en estado de marca (un uno lógico, conocido también como *RS-232 idle state*). El comienzo de flujo de datos se reconoce porque la señal pasa de “marca” a “espacio”.
- Dependiendo de la implementación, pueden existir unos bits de sincronización conocidos como bits de arranque o inicio (start bits). El emisor y el receptor deben ponerse de acuerdo si hay cero, uno o dos bits de arranque.
- Después de los bits que representan los datos (5,6,7, u 8 bits) puede seguir un bit de paridad (que es opcional, depende de la implementación) para ayudar a determinar si ha ocurrido un error durante la transmisión. Este error se llama *Parity Error* y puede ser causado por una configuración desigual en el emisor y el receptor.

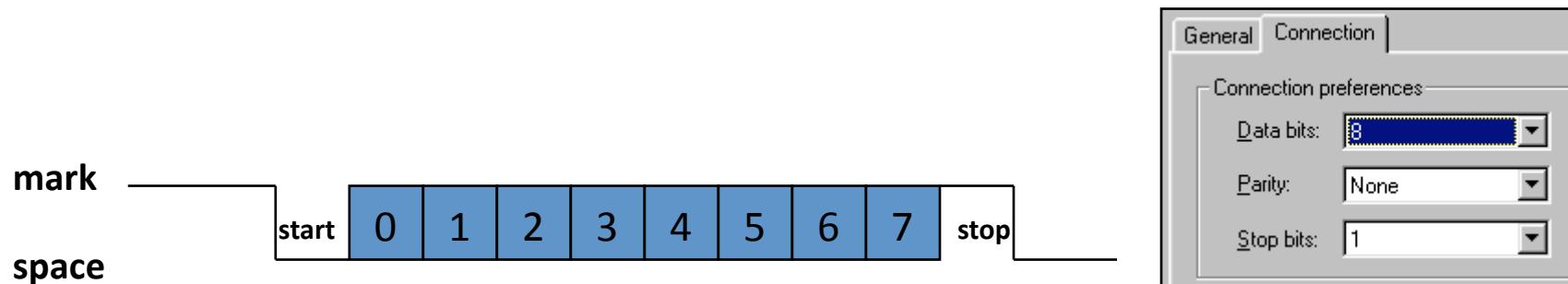
Representación de caracteres

- La paridad puede configurarse de diversas formas:
 - No Parity (sin paridad): No se transmite bit de paridad
 - Even Parity (paridad “par”): el bit de paridad es uno (1) si el carácter lleva un cantidad par de unos.
 - Odd Parity (paridad “impar”): el bit de paridad es uno (1) si el carácter lleva una cantidad impar de unos.
 - Mark Parity (paridad de “marca”): el bit de paridad siempre es uno
 - Space Parity (paridad de “espacio”): el bit de paridad siempre es cero

DATOS	Cantidad de unos	BIT DE PARIDAD			
		EVEN	ODD	MARK	SPACE
1010101	4	1	0	1	0
1111111	7	0	1	1	0
1010000	2	1	0	1	0
0101010	3	0	1	1	0
1111110	6	1	0	1	0
0011111	5	0	1	1	0
1000000	1	0	1	1	0

Representación de caracteres

- Después del bit de paridad (si lo hay) vienen los bits de parada (stop bits). Estos sirven para decir dónde termina el carácter. Pueden ser uno o dos bits de parada (en esto también deben ponerse de acuerdo el transmisor y el receptor). Algunas implementaciones cortan la transmisión del segundo bit de parada a la mitad, se dice entonces que utiliza uno y medio bits de parada. Los bits de parada se transmiten como unos lógicos (mark).



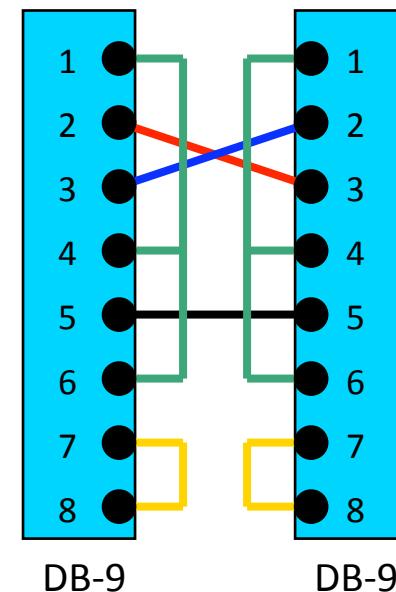
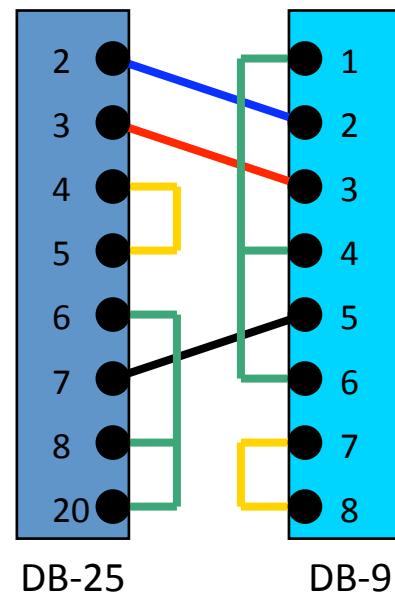
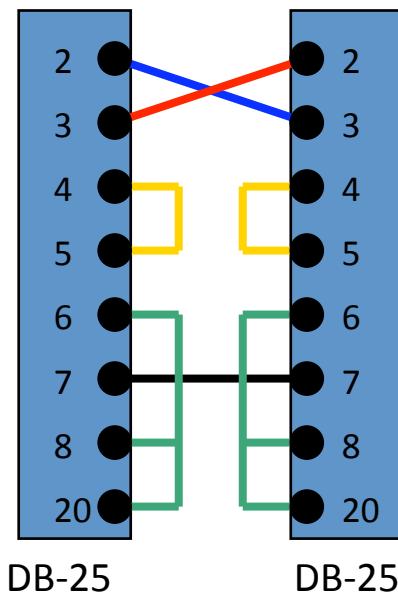
Cuando el bit de parada no se encuentra se produce un *Framing Error*. En estos casos es bueno revisar que el emisor y el receptor esperan la misma cantidad de bits de parada.

Control de flujo

- En RS-232 el control de flujo se puede hacer de dos maneras: por hardware (RTS/CTS) o por software (Xon/Xoff).
- Por Software: El carácter Xoff (ASCII 19, CTRL-S) es utilizado por el receptor para decir que su buffer está lleno y el emisor debe esperar. Cuando vuelva a tener espacio en el buffer, enviará el carácter Xon (ASCII 17, CTRL-Q), diciéndole que puede volver a transmitir. Esto economiza cables en la interconexión, pero ocupa espacio en el canal (perfecto para un PDA).
- Por hardware: a diferencia del anterior, este tipo de control de flujo requiere que entre su PC y su MODEM se conecten dos hilos: RTS y CTS. Cuando el buffer del receptor se llena (supongamos que es un modem “lento”) le dice al PC que espere, desactivando la señal CTS. Cuando vuelva a tener espacio en el buffer, activa nuevamente el CTS para decir que está nuevamente listo (esto puede pasar cuando la UART es más rápida que el MODEM).

RS-232 Null MODEM

- Un cable null modem se utiliza para conectar dos DTEs directamente a través de interfaces RS-232. Los siguientes diagramas muestran null modems elaborados con solo tres hilos. La idea es hacer pensar al DTE que está conectado a un DCE. El indicador de ring (pin 22 en DB-25 y pin 9 en DB-9) no se necesitan pues no hay línea telefónica.



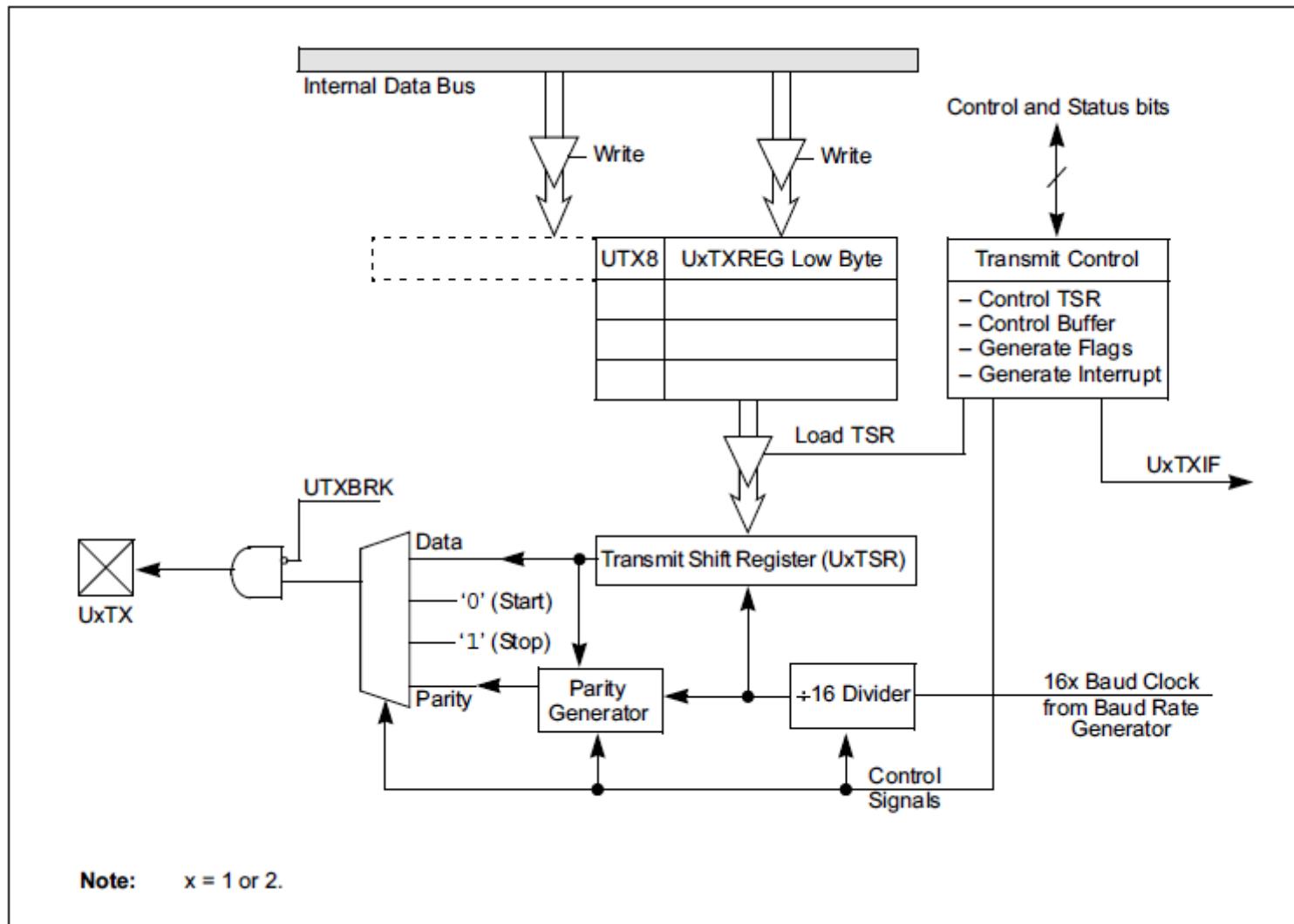
UART

(Universal Asynchronous Receiver/Transmitter)

- La UART es el dispositivo que controla la interface entre una MCU y los dispositivos serie. Específicamente permite al microcontrolador o PC utilizar la interface RS-232C para conectarse con los dispositivos pertinentes.
 - Convierte los bytes recibidos por la UART, en paralelo, en un flujo de bits serie y viceversa: los que llegan como flujo de bits los convierte en bytes en paralelo.
 - Agrega (a los bits que salen) y chequea (a los bits que entran) el bit de paridad.
 - Agrega (a los bits que salen) y retira (a los bits que entran) los bits de arranque (start bits) y parada (stop bits)

Modulo UART del dsPIC30F6014

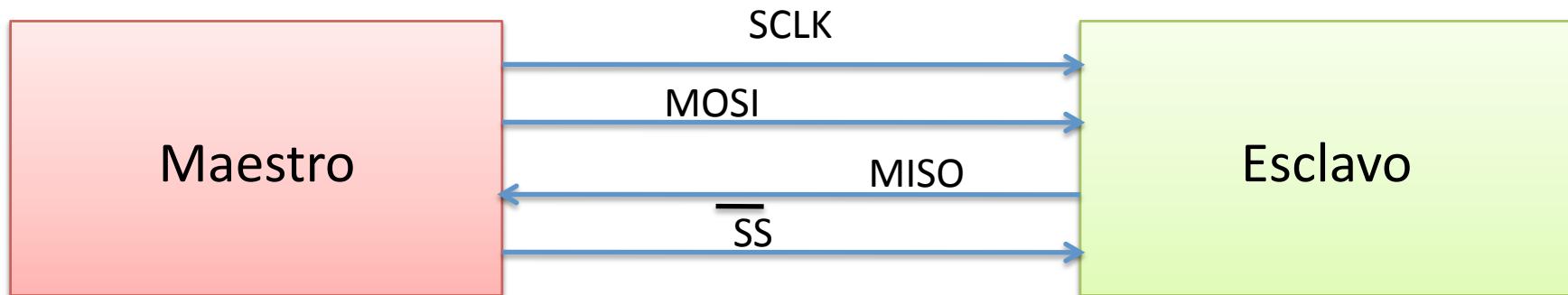
Diagrama de bloques



Interfaz SPI (Serial Peripheral Interface)

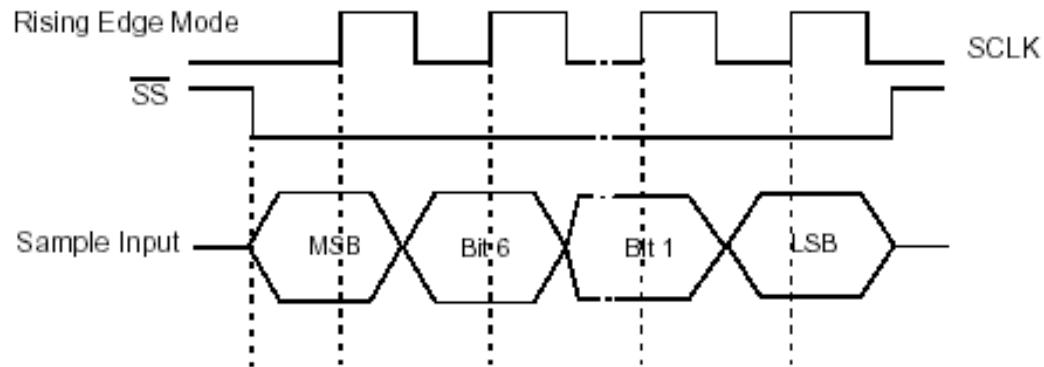
- Bus de comunicaciones Serie síncronas.
- Interface SPI
 - Basado en un registro de desplazamiento de 8-bit
 - SCK (serial shift clock) proporcionada por el dispositivo master.
 - SCK señal de reloj.
 - SCK permanece inhibida mientras se produce la transfeerncia
- 3-hilos
 - Serial clock (SCK)
 - SDI – Data In
 - SDO – Data Out

Conexionado SPI



- Transmisión full duplex
- Sistema Master/slave
- 2 señales de datos:
 - MOSI – master data output, slave data input
 - MISO – master data input, slave data output
- 2 señales de control:
 - SCLK – clock
 - /SS – slave select (no existe direccionamiento)

Protocolo SPI



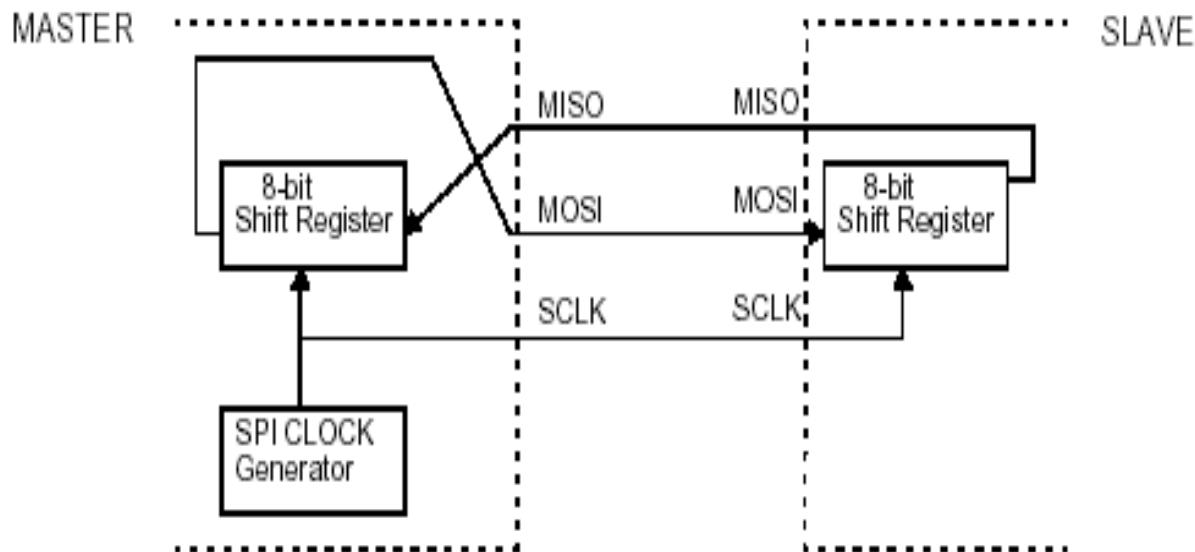
- 2 Parametros determinan el flanco activo del reloj: Clock Polarity (CPOL) y Clock Phase (CPHA)

CPOL	CPHA	Active edge
0	0	Rising
0	1	Falling
1	0	Falling
1	1	Rising

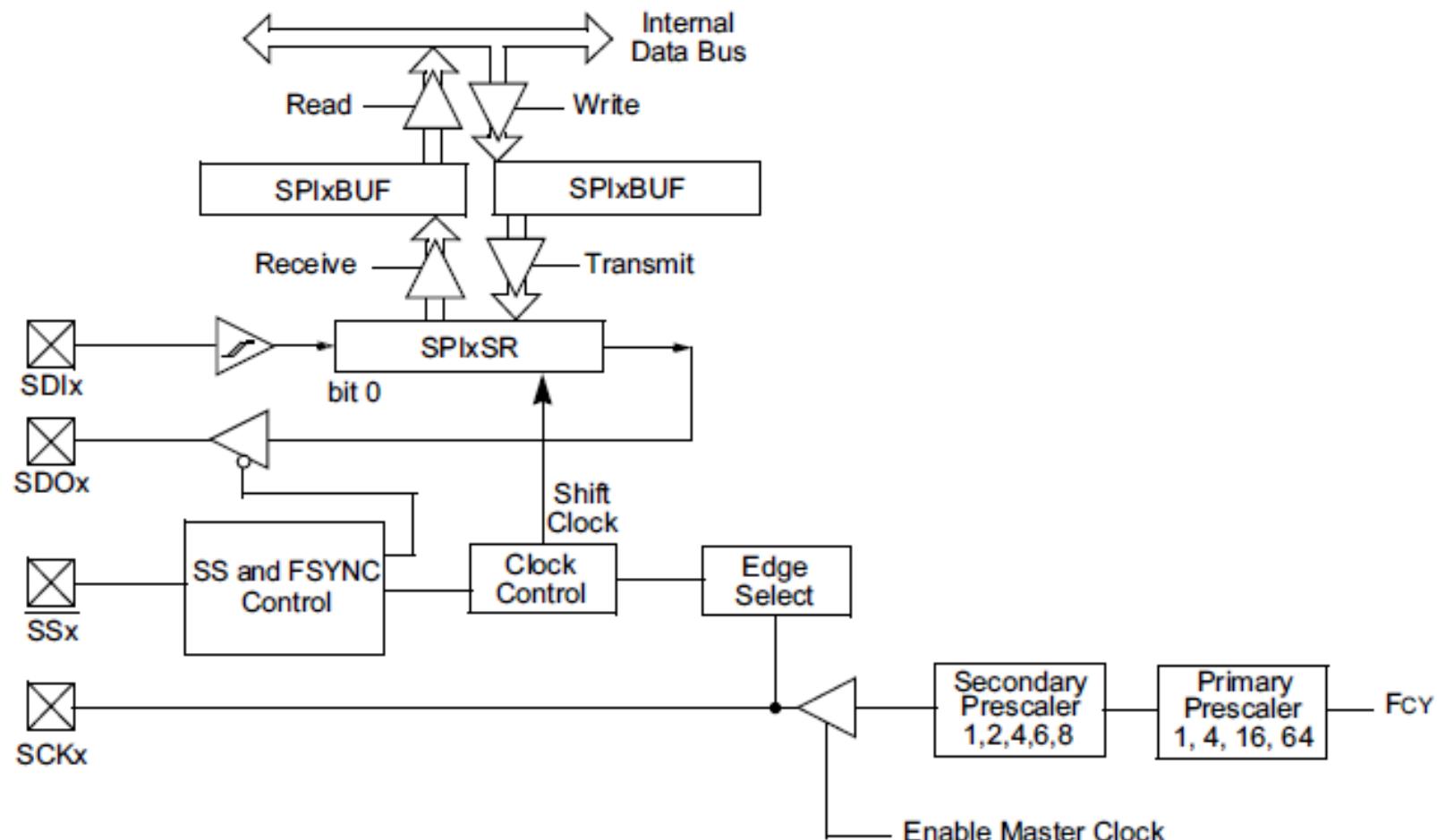
- El Maestro y el esclavo deben estar de acuerdo en este par de parámetros para poder realizar la comunicación.

Protocolo SPI

- SPI define solo las líneas de comunicación y el flanco del reloj.
- No se ha especificado un control de flujo. No existe un mecanismo de confirmación de la recepción del dato (ACK)
- La implementación hardware, se suele hacer con un registro de desplazamiento.



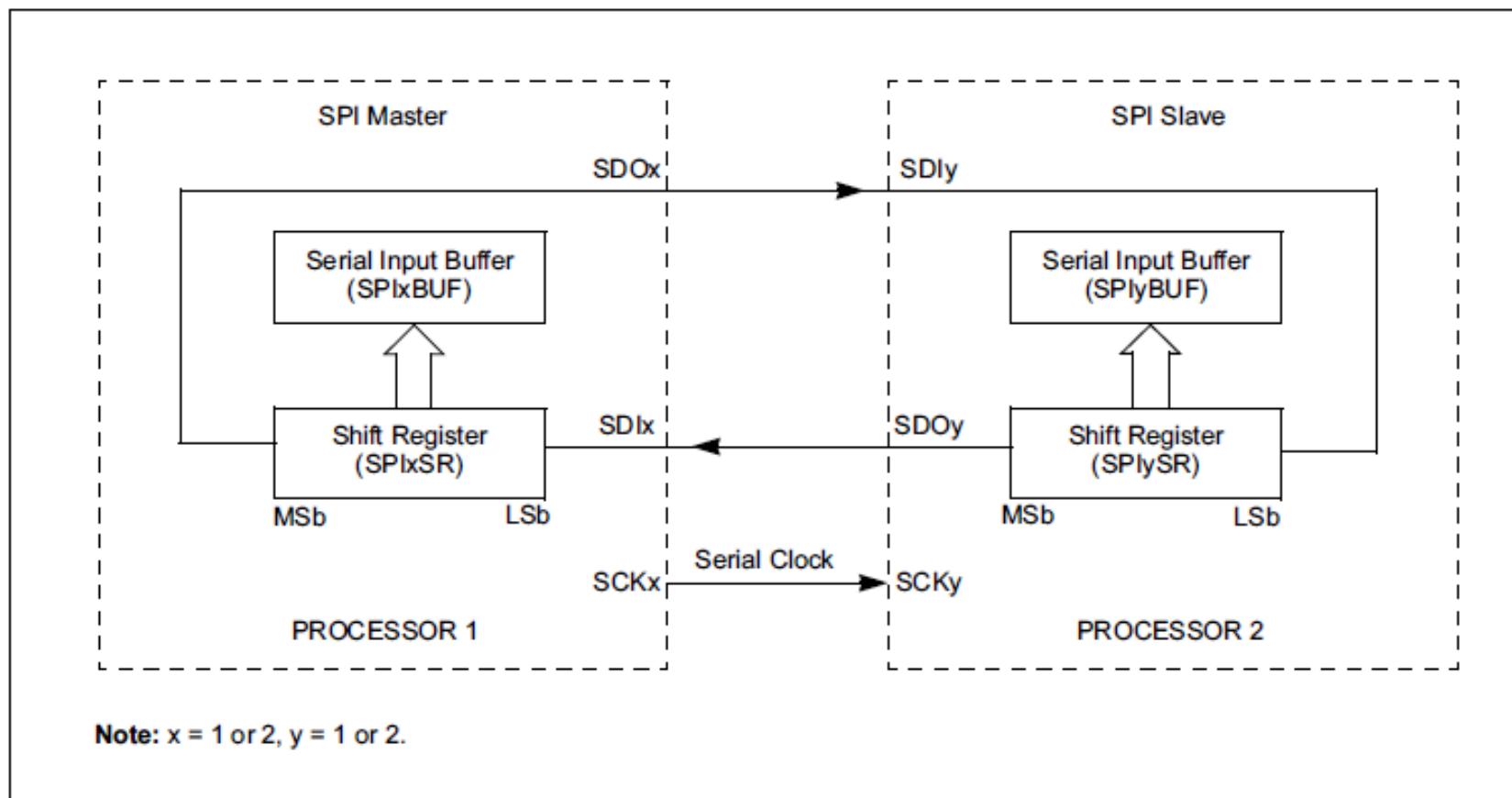
Modulo SPI en familia dsPIC



Note: $x = 1$ or 2 .

SPI en dsPIC

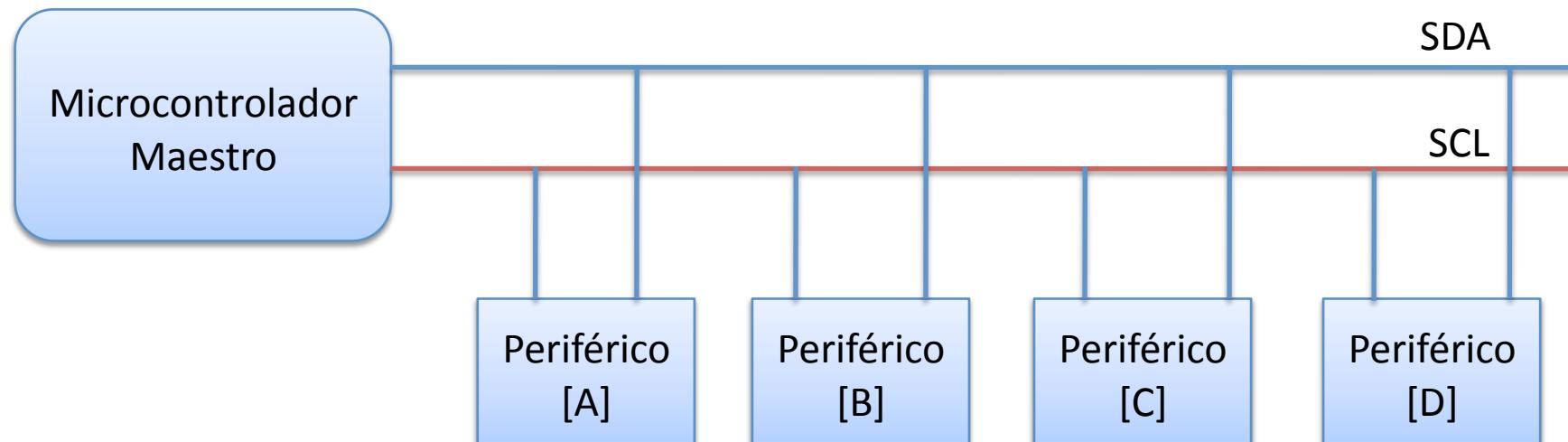
Conexión Maestro-esclavo



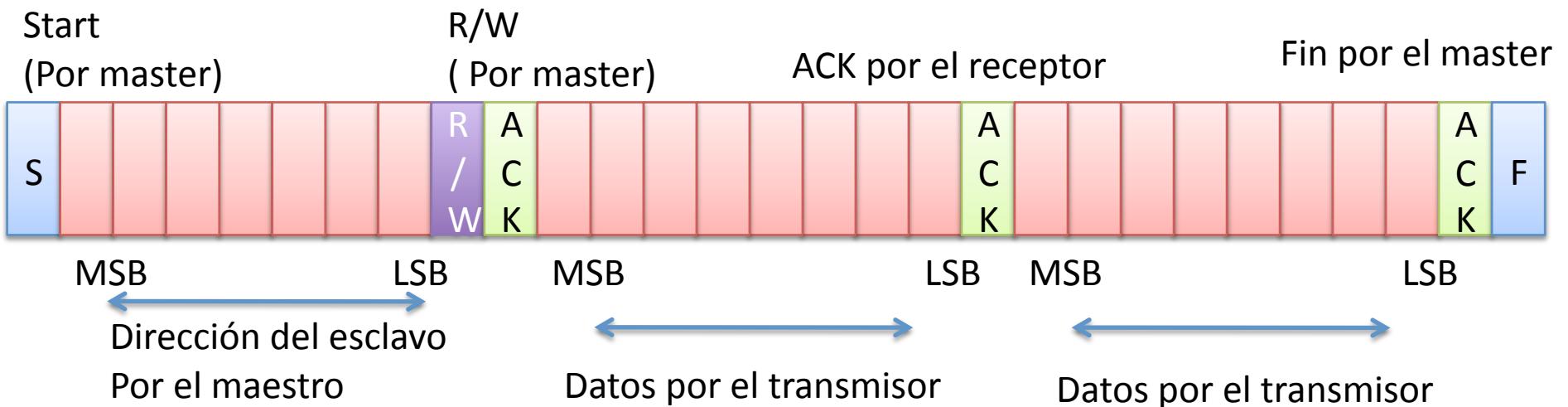
Bus I2C

- Inter-Integrated Circuit (I2C), patentado por Philips para comunicaciones de baja velocidad entre periféricos o circuitos integrados.
- Bus serie bidireccional de 2-hilos
 - SCL - Clock Line
 - SDA – Data Line
- Tres velocidades de transferencia de datos
 - Standard mode - 100 kbps
 - Fast mode - 400 kbps
 - High speed mode – hasta 3.4 Mbps
- Sistema Maestro- Esclavo

Configuración del bus I2C



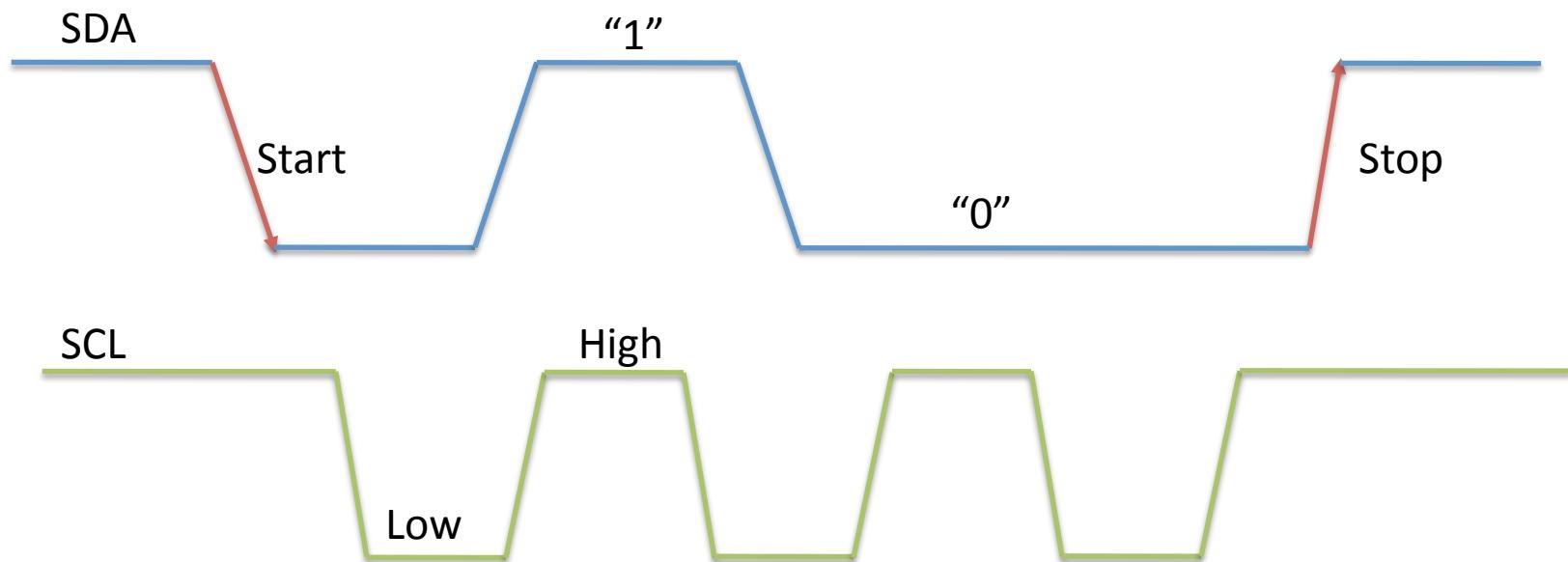
- Bus serie de 2 hilos: Serial data (SDA) y Serial clock (SCL)
- Comunicación Half-duplex, síncrona, bus multi-maestro
- No se requiere pin de “chip select” o lógica de arbitraje.
- Líneas conectadas a nivel alto mediante resistencia de “pull up”. La conexión a nivel bajo se realiza mediante transistores en drenador abierto “open-drain” (wired-AND)



1. El Maestro envía la condición de start (S) y controla la señal de reloj.
2. El Master envía la dirección del esclavo, de 7-bit
3. El Master envía el bit de lectura escritura (R/W) – 0 - esclavp recibe, 1 - esclavo transmite.
4. El Receptor envía señal con el bit de reconocimiento (ACK)
5. El Transmisor (esclavo o maestro) transmite 1 byte de datos

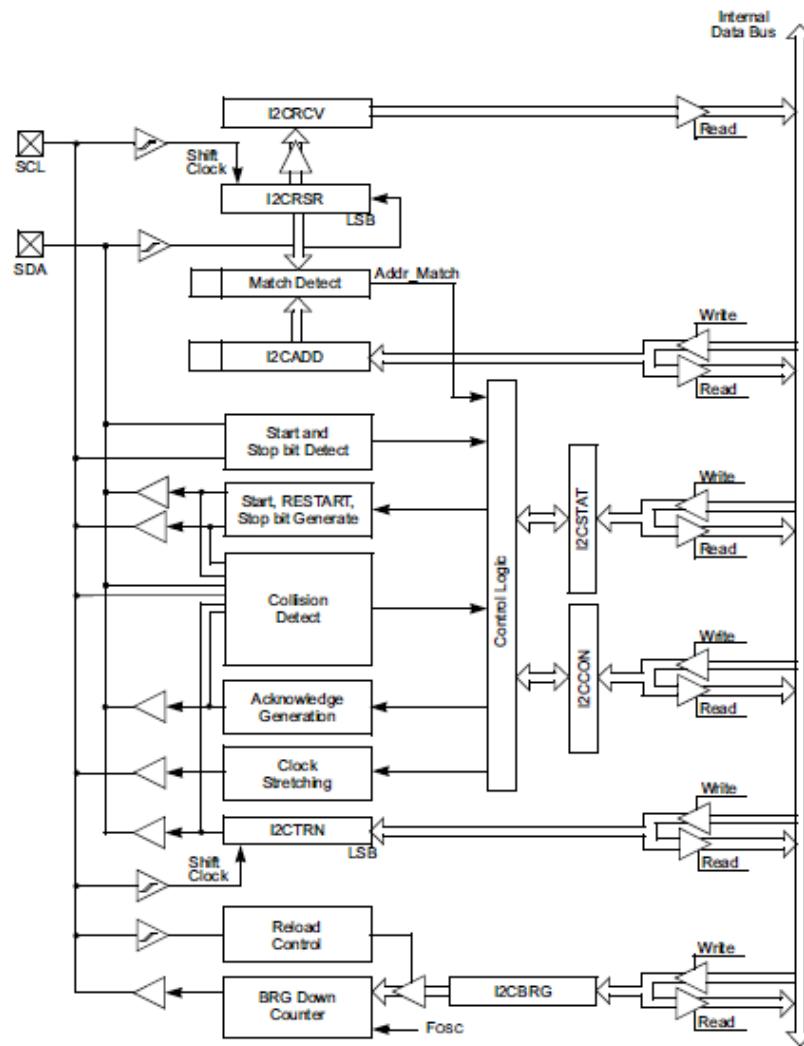
6. El receptor expide un bit de ACK por cada byte recibido.
7. Se repiten los pasos 5 y 6 si se transmiten mas bytes.
- 8.a) Para la transacción de escritura (maestro transmitiendo), el maestro expide la condición de stop (P) despues del ultimo byte de datos.
- 8.b) Para la transacción de lectura (maestro recibiendo), el maestro no envia el bit de ACK al final del byte de datos sino que expide la condición de stop (P) para decirle al sclavo que la transmisión se ha completado.

Señales I2C



- Start – high-to-low transition of the SDA line while SCL line is high
- Stop – low-to-high transition of the SDA line while SCL line is high
- Ack – receiver pulls SDA low while transmitter allows it to float high
- Data – transition takes place while SCL is slow, valid while SCL is high

I2C de la familia dsPIC



Referencias

- [http://perso.wanadoo.es/pictob/
comserie.htm](http://perso.wanadoo.es/pictob/comserie.htm)