



I2C: Introducción (I)

- Interconexión de dispositivos mediante bus serie
 - Ventajas
 - Pocos cables de interconexión.
 - Componentes con encapsulado reducido.
 - Tarjetas reducidas.
 - Conexión de dispositivos a distancia.
 - Inconvenientes
 - Velocidad inferior a un bus paralelo
 - Disponibilidad de circuitos que soporten el bus
 - Parámetros generales
 - Número de hilos de conexión
 - Velocidad (bits/segundo)
 - Distancia máxima y número de dispositivos
 - Protocolo de acceso al medio compartido
 - Política de direccionamiento



I2C: Introducción (II)

- Origen del bus I2C (Inter Integrated Circuits Bus)
 - Desarrollado por Philips a principios de los 80 como medio de interconexión entre una CPU y dispositivos periféricos dentro de la electrónica de consumo.
 - Simplificar las conexiones entre los periféricos (pistas, decodificadores, ..)
 - Aumentar de la inmunidad al ruido.
 - Control de sistemas de audio y vídeo (baja velocidad).
 - Actualmente diseñan dispositivos basados en I2C muchos fabricantes:
 - Xicor, SGS-Thomson, Siemens, Intel, TI, Maxim, Atmel, Analog Devices
- Aplicaciones
 - Bus de interconexión entre dispositivos en una tarjeta o equipo.
 - Sistema de configuración y supervisión en ordenadores servidores.
 - Sistemas de gestión de alimentación.
 - Conexión en serie de dispositivos externos a un ordenador.
 - Tarjetas chip.

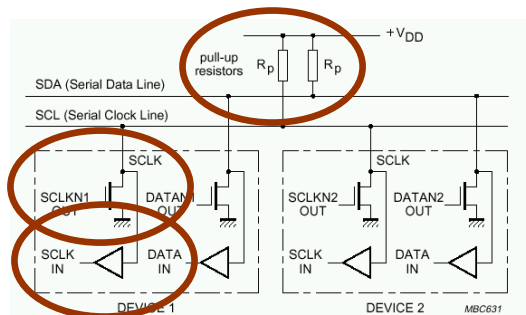


I2C: Características generales

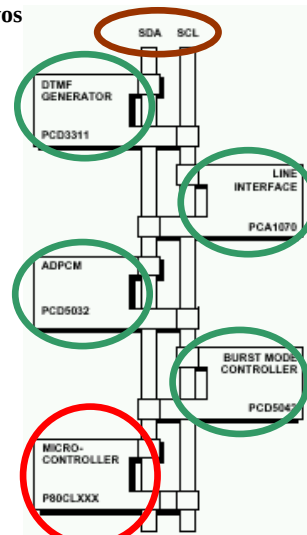
- Bus de comunicación síncrono.
 - La comunicación es controlada por una señal de reloj común.
- Bus formado por 2 hilos:
 - SDA (Serial Data Line): datos
 - SCL (Serial Clock line): reloj
 - También es necesaria una referencia común de masa.
- Velocidad de transmisión.
 - Standard: hasta 100 Kbits/s.
 - Fast: hasta 400 Kbits/s.
 - High-speed: hasta 3,4 Mbits/s.
- Cada dispositivo del bus tiene una **dirección única**.
 - 7 bits, I2C estándar
 - 11 bits, I2C mejorado.
- Distancia y número de dispositivos
 - Limitado por la capacidad del bus (inferior a 400pF). Normalmente 2 o 3 metros.
 - Drivers para largas distancias (centenares de metros).
- Protocolo de acceso al bus:
 - Maestro – esclavo.
 - I2C soporta protocolo multimaestro.

I2C: Conexión de dispositivos al bus

- Conexión en bus:
 - Todos los dispositivos conectados a las mismas líneas.
 - Las salidas deben ser en colector o drenador abierto.



Dispositivos



Microcontrolador



I2C: Intercambio de información

- El maestro controla la comunicación
 - Genera la señal de reloj del bus (SCL).
 - Inicia y termina la comunicación.
 - Direcciona a los esclavos.
 - Establece el sentido de la comunicación.
- El protocolo requiere que cada byte de información sea confirmado por el destinatario.
- Nomenclatura
 - Emisor: Dispositivo que envía datos al bus.
 - Receptor: Dispositivo que recibe datos del bus.
 - Maestro: Dispositivo que inicia una transferencia, genera las señales de reloj y termina la transferencia.
 - Esclavo: Dispositivo direccionado por un maestro.

TEMA 2-I2C-5



Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación

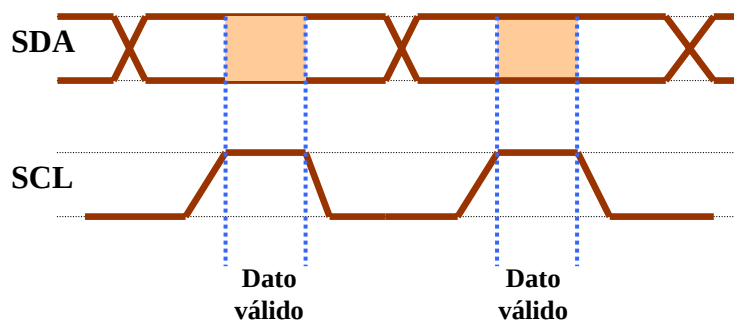


JMVC/FJRS



I2C: Nivel de enlace

- Transmisión de bits:
 - Los bits de datos van por SDA.
 - Por cada bit de información es necesario un pulso de SCL.
 - Los datos sólo pueden cambiar cuando SCL está a nivel bajo.



TEMA 2-I2C-6



Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación

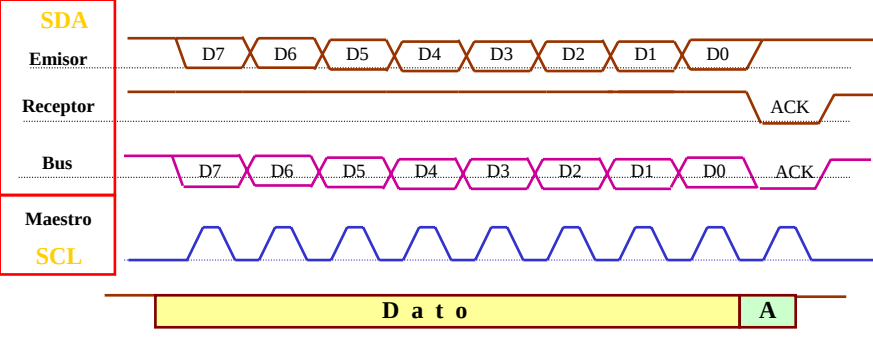


JMVC/FJRS



I2C: Nivel de enlace (II)

- Transmisión de datos
 - La unidad básica de transmisión es el byte.
 - Las transferencias de datos son de 8 bits.
 - Cada byte enviado requiere una respuesta de confirmación.
 - ACK: el destinatario (maestro o esclavo) mantiene SDA a nivel bajo durante un tiempo de bit (si no lo hace → NACK).
 - El maestro genera un pulso de SCL.



TEMA 2-I2C-7

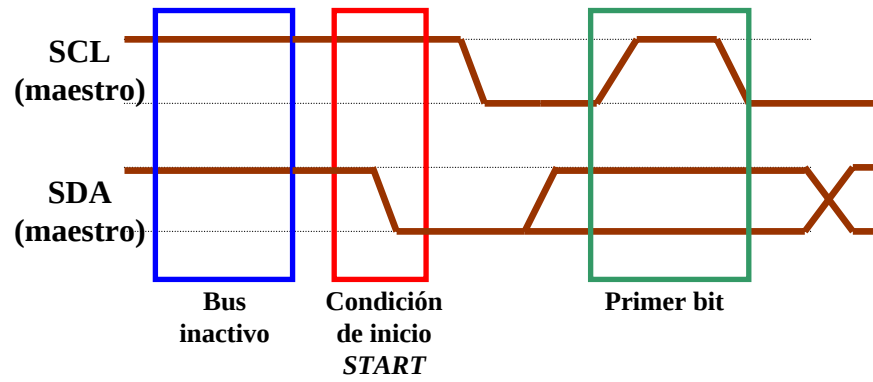

 Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación


 JMVC/FJRS



I2C: Nivel de enlace (III)

- Inicio de transmisión (START)
 - La transmisión la inicia el maestro.
 - Flanco de bajada en SDA con SCL a nivel alto.
 - Cuando nadie accede al bus hay un nivel alto en SCL y SDA.



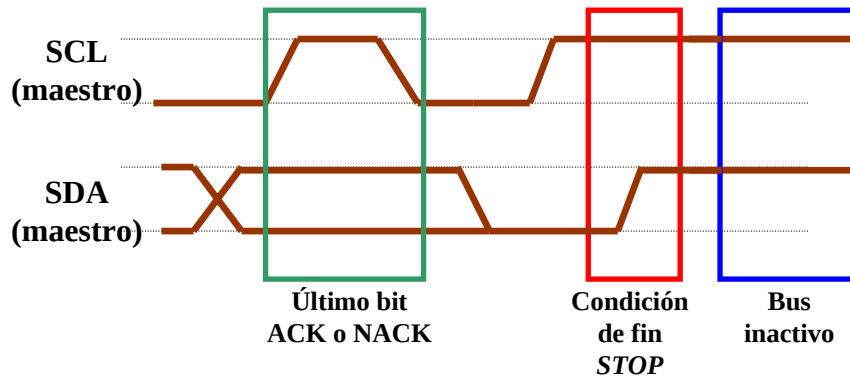
TEMA 2-I2C-8


 Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación


 JMVC/FJRS

I2C: Nivel de enlace (III)

- Finalización de transmisión (STOP)
 - La transmisión la finaliza el maestro
 - Flanco de subida en SDA con SCL a nivel alto



TEMA 2-I2C-9



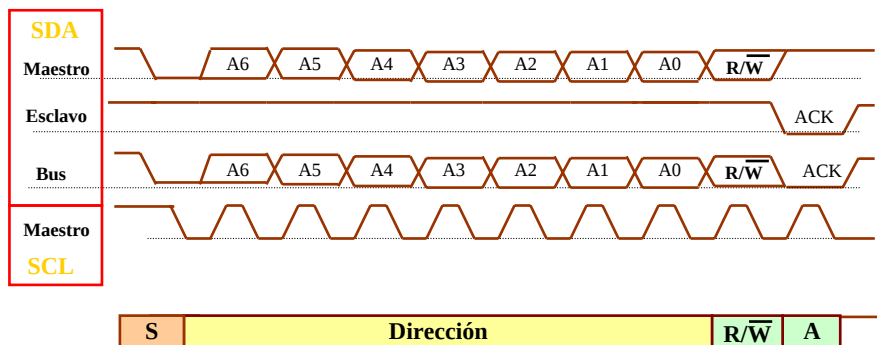
Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación



JMVC/FJRS

I2C: Nivel de enlace (IV)

- Intercambio de datos
 - Direccionamiento
 - Tras la condición de inicio el maestro envía:
 - Dirección del esclavo (7 bits)
 - Comando de lectura o escritura ($R=1 - W=0$)



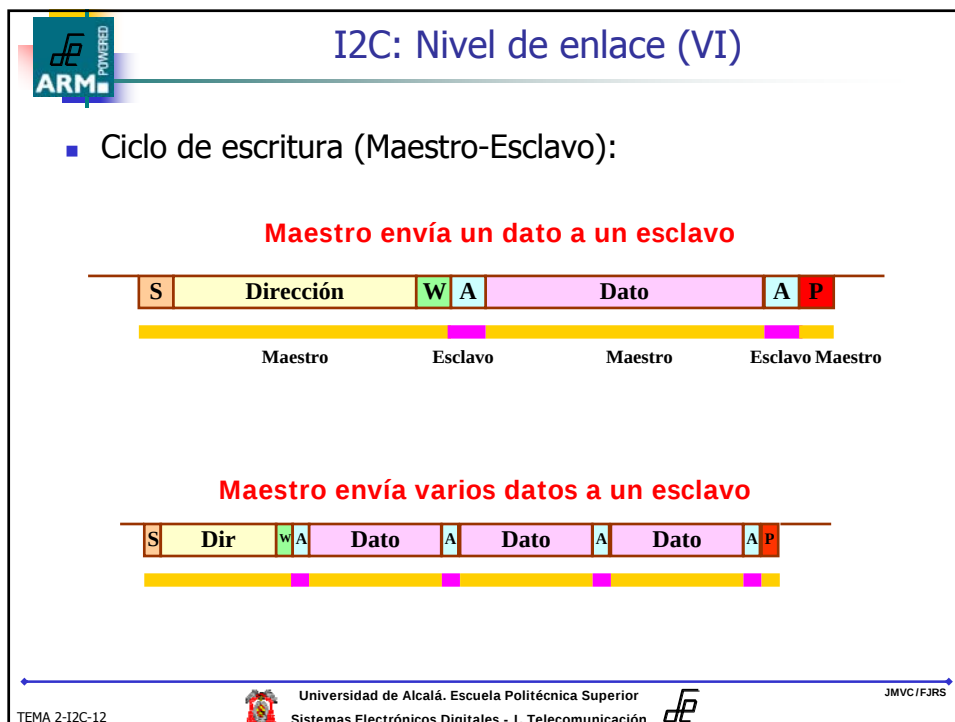
TEMA 2-I2C-10



Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación

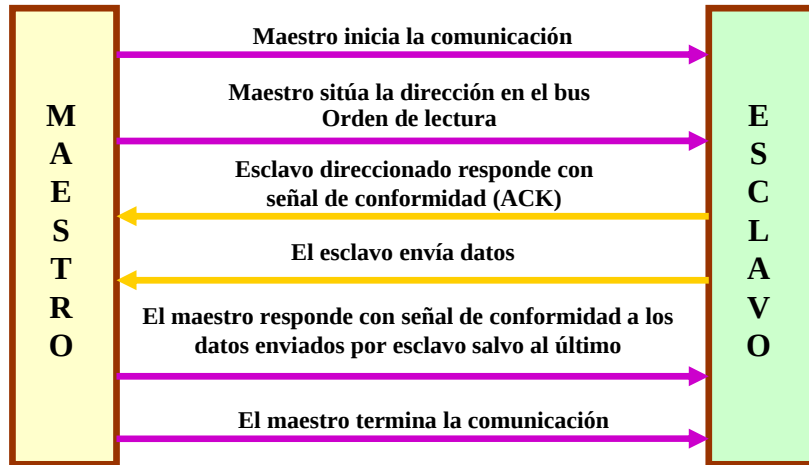


JMVC/FJRS



I2C: Nivel de enlace (VII)

Maestro lee datos de un esclavo (Lectura)



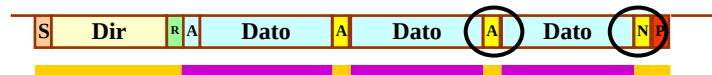
I2C: Nivel de enlace (VIII)

Ciclo de lectura (Maestro-Eslavo):

Maestro lee un dato de un esclavo



Maestro lee varios datos de un esclavo



I2C: Nivel de enlace (IX)

Ciclo de escritura-lectura (Sin Stop)

Maestro escribe y lee (con Re-START y sin STOP)



I2C: Nivel de enlace (X)

- Casos particulares de respuesta:
 - Esclavo ocupado y no responde al ACK de dirección.
 - Maestro genera condición de STOP.
 - El esclavo quiere interrumpir una recepción periódica de datos.
 - El esclavo no responde a un dato con ACK.
 - El maestro genera la condición de STOP.
 - El maestro quiere interrumpir una recepción de datos de un esclavo.
 - El maestro no responde con ACK.
 - El esclavo deja de transmitir.
 - El maestro genera condición de STOP.



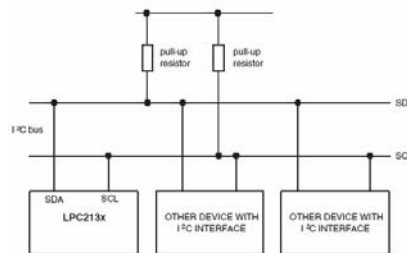
I2C: Nivel de enlace (XI)

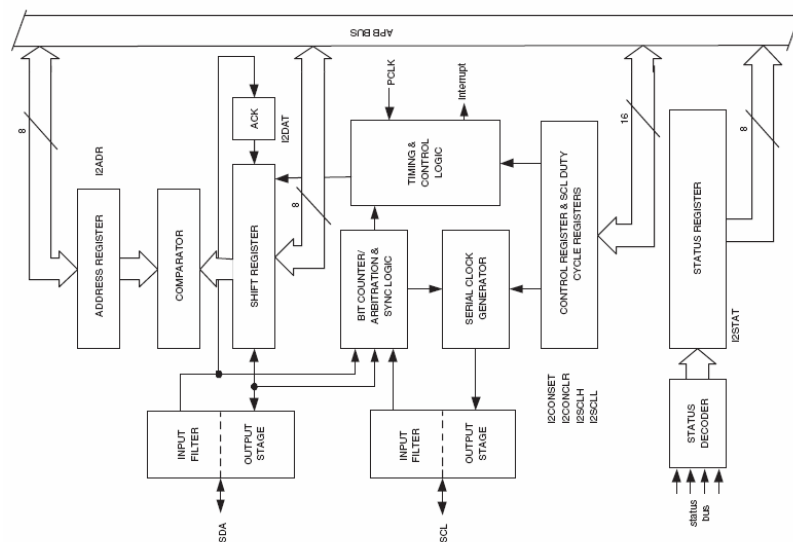
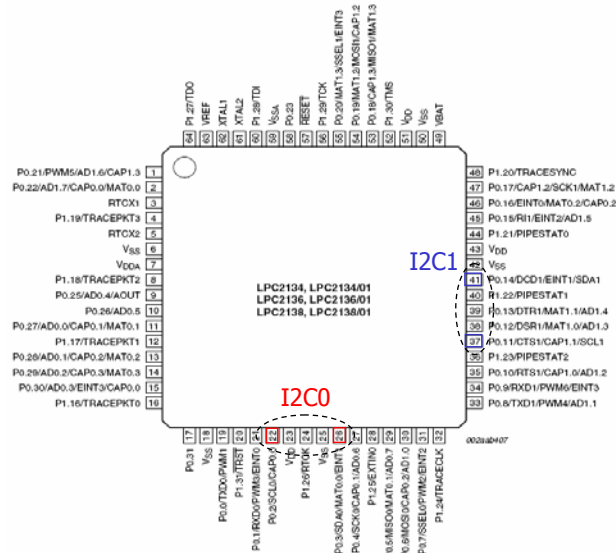
- Protocolo multimaestro
 - Con dos maestros en el bus existe posibilidad de conflicto.
- Arbitración:
 - Procedimiento para asegurar que sólo un maestro tiene el control del bus en un instante.
- Si un maestro está utilizando el bus no puede ser interrumpido por otro.
 - Desde START hasta STOP.
- Si dos maestros intentan comenzar a utilizar el bus a la vez:
 - Conexión del bus en colector abierto → prevalecen los ceros.
 - A la vez que ponen datos en el bus, escuchan la línea.
- Si un maestro está intentando enviar un nivel alto y lee un nivel bajo.
 - Existe otro maestro utilizando el bus.
 - Deja de transmitir esperando que la línea quede libre (condición de STOP).



I2C: LPC213x (Características)

- Compatible con el estándar I2C y posibilidad de trabajar como master o slave.
- 2 Interfaces I2C (SDA0,1 y SCL0,1).
- 4 modos de funcionamiento:
 - Master transmisor.
 - Master receptor.
 - Slave transmisor.
 - Slave receptor.
- Mecanismo de arbitración en modo multimaestro.
- Velocidad de transferencia (clock del bus) programable.
- Mecanismo de sincronización del reloj para permitir comunicar con varios dispositivos a diferentes velocidades.





I2C: Registros de control y estado

- **I2CnCONSET, I2CnCONCLR.** Registros de control que permiten controlar:
 - La condición START y Re-START, inician la transferencia serie.
 - La condición STOP.
 - La velocidad.
 - El reconocimiento de dirección.
 - El bit reconocimiento (A) de un dato.
- **I2CnSTAT.** Registro de estado (5 bits)
 - Hasta 26 estados diferentes en función del modo de operación del bus I2C.
- **I2CnADR.** Registro de dirección Slave (7 bits).
 - El bit de menor peso sirve de reconocimiento de la dirección.
- **I2CnDAT.** Registro de datos.
 - Ha de cargarse con el byte a transmitir, o es el registro dónde se recibe el último dato.
- **I2CnSCLH, I2CnSCLL.** Selección del reloj.

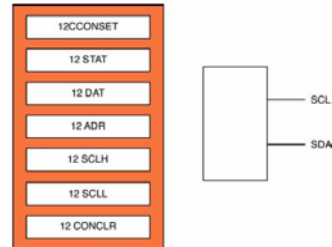


Table 144. Abbreviations used to describe an I²C operation

Abbreviation	Explanation
S	Start Condition
SLA	7-bit slave address
R	Read bit (HIGH level at SDA)
W	Write bit (LOW level at SDA)
A	Acknowledge bit (LOW level at SDA)
NA	Not acknowledge bit (HIGH level at SDA)
Data	8-bit data byte
P	Stop condition

I2C: Mapa de registros

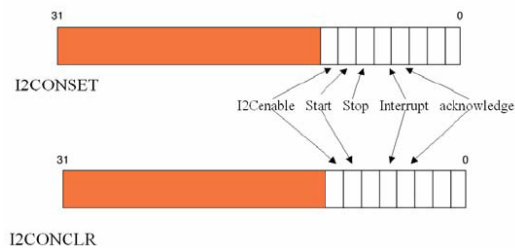
Table 135. I²C register map

Name	Description	Access	Reset value ¹	I ² C0 Address and Name	I ² C1 Address and Name
I2CONSET	I ² C Control Set Register. When a one is written to a bit of this register, the corresponding bit in the I ² C control register is set. Writing a zero has no effect on the corresponding bit in the I ² C control register.	R/W	0x00	0xE001 C000 I2C0CONSET	0xE005 C000 I2C1CONSET
I2STAT	I ² C Status Register. During I ² C operation, this register provides detailed status codes that allow software to determine the next action needed.	RO	0xF8	0xE001 C004 I2C0STAT	0xE005 C004 I2C1STAT
I2DAT	I ² C Data Register. During master or slave transmit mode, data to be transmitted is written to this register. During master or slave receive mode, data that has been received may be read from this register.	R/W	0x00	0xE001 C008 I2C0DAT	0xE005 C008 I2C1DAT
I2ADR	I ² C Slave Address Register. Contains the 7-bit slave address for operation of the I ² C interface in slave mode, and is not used in master mode. The least significant bit determines whether a slave responds to the general call address.	R/W	0x00	0xE001 C00C I2C0ADR	0xE005 C00C I2C1ADR
I2SCLH	SCH Duty Cycle Register High Half Word. Determines the high time of the I ² C clock.	R/W	0x04	0xE001 C010 I2C0SCLH	0xE005 C010 I2C1SCLH
I2SCLL	SCL Duty Cycle Register Low Half Word. Determines the low time of the I ² C clock. I2nSCLL and I2nSCLH together determine the clock frequency generated by an I ² C master and certain times used in slave mode.	R/W	0x04	0xE001 C014 I2C0SCLL	0xE005 C014 I2C1SCLL
I2CONCLR	I ² C Control Clear Register. When a one is written to a bit of this register, the corresponding bit in the I ² C control register is cleared. Writing a zero has no effect on the corresponding bit in the I ² C control register.	WO	NA	0xE001 C018 I2C0CONCLR	0xE005 C018 I2C1CONCLR

I2C: Registro de control (I2CONSET)

Table 136. I²C Control Set register (I2CONSET: I2C0, I2C0CONSET - address 0xE001 C000 and I2C1, I2C1CONSET - address 0xE005 C000) bit description

Bit	Symbol	Description	Reset value
1:0	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	AA	Assert acknowledge flag. See the text below.	
3	SI	I ² C interrupt flag.	0
4	STO	STOP flag. See the text below.	0
5	STA	START flag. See the text below.	0
6	I2EN	I ² C interface enable. See the text below.	0
7	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA



I2C: Registro de control (I2CONSET)

Table 133. I2C0CONSET and I2C1CONSET used to configure Master mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	0	-	-

Table 147. I2C0CONSET and I2C1CONSET used to initialize Slave Receiver mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	1	-	-

Table 146. I2C0ADR and I2C1ADR usage in Slave Receiver mode

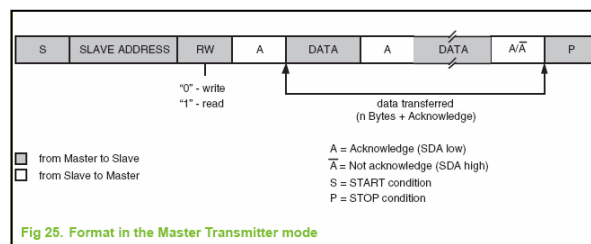
Bit	7	6	5	4	3	2	1	0
Symbol	own slave 7-bit address							GC

GC: Bit llamada general (Dir=0x00)

I2C: Modos de funcionamiento

■ Modo Master Transmisor.

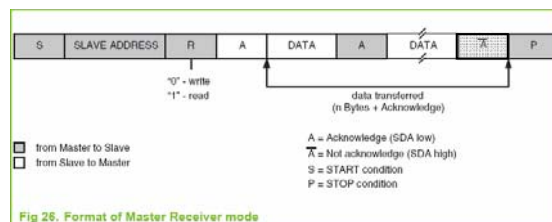
- Los datos son enviados del Master al Slave.
- El registro I2CnCONSET ha de inicializarse con los valores:
 - I2EN=1, para habilitar I2C.
 - STA=STO=SI=0. El uC entra en modo master al poner STA=1, enviando la condición START (si el bus está libre).
- Después de enviar la condición START, SI=1 → I2CSTAT = 0x08 → Puede causar interrupción para cargar I2CDAT con la dirección (7 bits) + el bit R/W (SLA+W). Para borrar SI, SIC=1 en I2CnCONCLR.
- Después de enviar la dirección + el bit R/W, y recibido el bit ACK, SI=1, indicando que I2CSTAT contiene un valor que indica el estado de la transferencia (ver tabla 148).
 - I2CSTAT= 0x18, 0x20 ó 0x38 (Si modo Master)
 - I2CSTAT= 0x68, 0x78 ó 0xB0 (Si modo Slave → AA=1)



I2C: Modos de funcionamiento

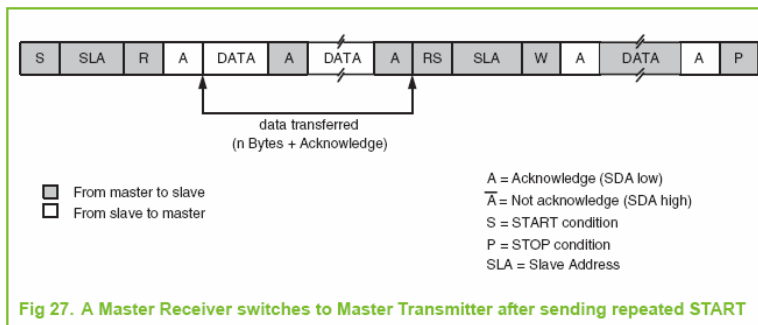
■ Modo Master Receptor.

- Se reciben un número de bytes del Slave transmisor.
- La Transferencia se inicia igual que en el modo Master transmisor.
- Una vez enviada la dirección del slave + el bit R/W, y recibido el bit ACK, S=1 indicando que el registro I2CSTAT contiene un valor que indica el estado de la transferencia (Ver tabla 149).
 - I2CSTAT= 0x40, 0x48 o 0x38 (si en modo Master)
 - I2CSTAT=0x68, 0x78, ó 0xB0 (si en modo Slave → AA=1)

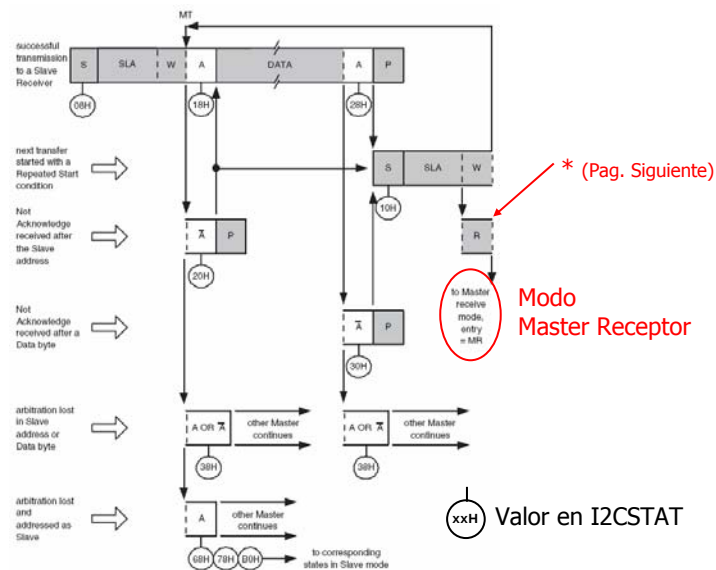


I2C: Modos de funcionamiento

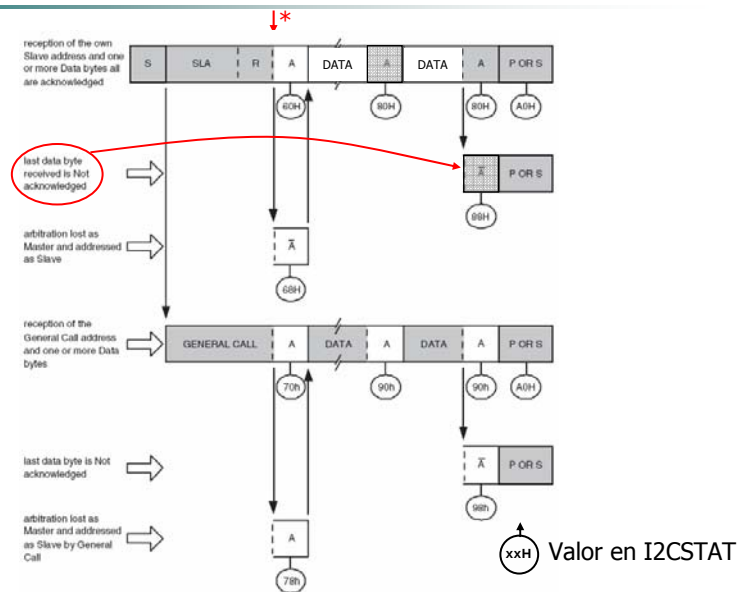
- **Modo Master Receptor con cambio a Master Transmisor.**
 - Ciclo de lectura-escritura



I2C: Modos de funcionamiento (Master Transmisor)



I2C: Modos de funcionamiento (Master Receptor)



I2C: Velocidad de transferencia

$$I^2C_{bitfrequency} = \frac{PCLK}{I2CSCLH + I2CSCLL}$$

Table 143. Example I²C clock rates

I2SCLL + I2SCLH	I ² C Bit Frequency (kHz) at PCLK (MHz)						
	1	5	10	16	20	40	60
8	125						
10	100						
25	40	200	400				
50	20	100	200	320	400		
100	10	50	100	160	200	400	
160	6.25	31.25	62.5	100	125	250	375
200	5	25	50	80	100	200	300
400	2.5	12.5	25	40	50	100	150
800	1.25	6.25	12.5	20	25	50	75

Nota: I2CSCLL e I2CSLH ≥ 4



I2C: Software (I)

■ Inicialización

■ Ejemplo inicialización interfaz I2C (Master/Slave)

- 1. Cargar I2ADR con la dirección como Slave.
- 2. Habilitar interrupciones.
- 3. I2CONSET=0x44 (I2EN=1, AA=1), para habilitar funciones como Slave.
- 4. I2CONSET=0x40 (I2EN), como Master.

■ Inicio modo Master transmisor.

■ Configurar buffer de transmisión, puntero y contador de datos para iniciar la transferencia (START):

- 1. Inicializar contador de datos del Master.
- 2. Fijar la dirección del Slave y el bit R/W=0 (Escritura).
- 3. I2CONSET=0x20 (STA=1) → generación de la condición START.
- 4. Inicializar puntero del buffer transmisor.
- 5. Cargar el número de datos a transmitir en el contador.
- 6. Salir.



I2C: Software (II)

■ Inicio modo Master Receptor.

- 1. Inicializar contador de datos del Master.
- 2. Fijar la dirección del Slave y el bit R/W=1 (Lectura).
- 3. I2CONSET=0x20 (STA=1) → generación de la condición START.
- 4. Inicializar puntero del buffer receptor.
- 5. Cargar el número de datos a recibir en el contador.
- 6. Salir.

■ Rutina de interrupción.

- 1. Leer el estado de I2CSTAT → *switch (I2CSTAT){ ...*
- 2. Actuar en consecuencia en función del estado del bus.
 - (Ver pag. 161 a 164 del manual)





I2C: ISR (Modo Master transmisor)

```
void I2CISR (void) // I2C interrupt routine
{
    switch (I2STAT) // Read result code and switch to next action
    {
        case (0x08): // Start bit
            I2CONCLR = 0x20; // Clear start bit
            I2DAT = I2CAddress; // Send address and
            // write bit
            break;

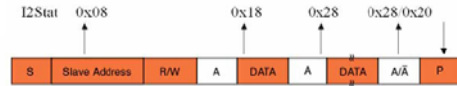
        case (0x18): // Slave address+W, ACK
            I2DAT = I2Cdata; // Write data to tx register
            break;

        case (0x20): // Slave address +W, Not ACK
            I2DAT = I2CAddress; // Resend address and write bit
            break;

        case (0x28): // Data sent, Ack
            I2CONSET = 0x10; // Stop condition
            break;

        default :
            break;
    }

    I2CONCLR = 0x08; // Clear I2C interrupt flag
    VICVectAddr = 0x00000000; // Clear interrupt in
}
```



TEMA 2-I2C-33



Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación



JMVC/FJRS



I2C: ISR (Modo Master receptor)

```
case (0x40) : // Slave Address +R, ACK
    I2CONSET = 0x04; // Enable ACK for data byte
    break;

case (0x48) : // Slave Address +R, Not Ack
    I2CONSET = 0x20; // Resend Start condition
    break;

case (0x50) : // Data Received, ACK
    message = I2DAT;
    I2CONSET = 0x10; // Stop condition
    lock = 0; // Signal end of I2C activity
    break;

case (0x58) : // Data Received, Not Ack
    I2CONSET = 0x20; // Resend Start condition
    break;
```



TEMA 2-I2C-34



Universidad de Alcalá. Escuela Politécnica Superior
Sistemas Electrónicos Digitales - I. Telecomunicación



JMVC/FJRS