

Tema 12: Comunicación Serie Síncrona

Sistemas Digitales Basados en Microprocesador

Universidad Carlos III de Madrid

Dpto. Tecnología Electrónica

Índice

- Introducción
- Comunicación Síncrona
 - 2-wire (I2C)
 - 3/4-wire (SPI)

Introducción

Comunicación Serie Síncrona (I)

- Es una comunicación serie en la que la secuencia de bits se controla mediante una **señal de reloj** explícita, que valida el bit por flanco o por nivel
 - No es necesario que la señal de reloj sea periódica ni continua, pero normalmente es denominada como **reloj (clk)**, aunque el nombre correcto debería ser **señal de validación**
 - La señal de reloj es muy crítica y no se puede ver afectada por interferencias, ya que impactaría en la sincronización de los bits. Esta es la razón por la que este tipo de comunicación sólo se considera para distancias muy cortas
- Por tanto, la comunicación serie síncrona está diseñada para la comunicación entre chips dentro de la placa de circuito impreso, y nunca se utiliza a través de cables
 - Las pistas son muy cortas, para que la resistencia y la capacidad sea muy baja y por tanto la transición entre '0's y '1's sea casi perfecta, incluyendo la señal de reloj

Comunicación Serie Síncrona (II)

- Hay muchos protocolos que implementan este tipo de comunicación, pero la industria ha ido seleccionando muy pocos de ellos, de forma que actualmente sólo se consideran:
 - I²C
 - SPI

I²C

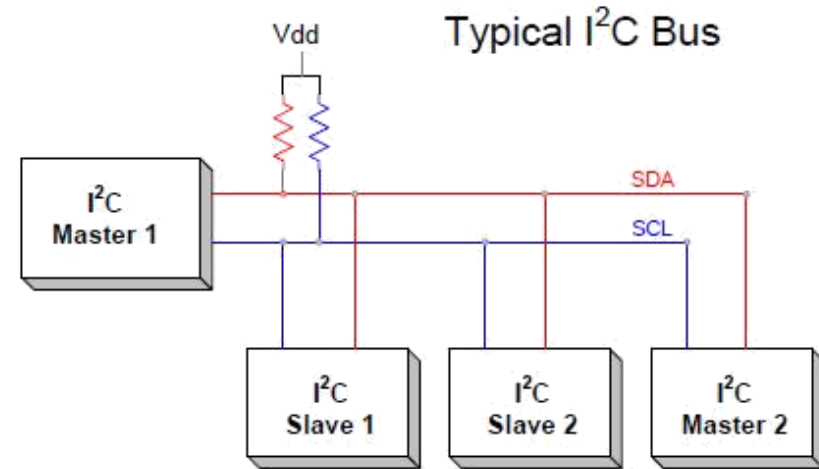


I²C: Características Generales



- Es un protocolo diseñado por Philips, bajo el nombre Inter Integrated Circuits
- Proporciona un interfaz a 2 hilos (2-wire)
 - SDA: Línea de Datos (bidireccional)
 - SCL: Línea de Reloj
- Utiliza niveles TTL (-> 2 niveles de tensión, por ejemplo, 0-5V o similar) por lo que no es posible modular la información utilizando distintas tensiones o frecuencias
- No permite comunicación full-dúplex, permitiendo sólo comunicación semi-dúplex
- La selección del dispositivo esclavo se hace a través de un sistema de direcciones, no mediante líneas hardware. La trama está limitada mediante una condición START y una condición STOP
- Velocidades de comunicación: 0.1 / 0.4 / 1.0 / 3.4 / 5.0 Mbps

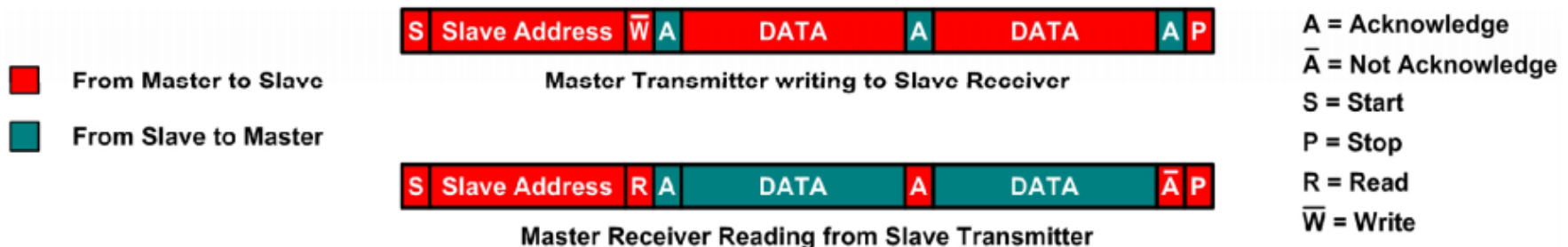
I²C: Conexión



Paso 1 El Maestro (micro) envía la dirección del dispositivo con el que se quiere comunicar
El Esclavo (sensor) responde ACK

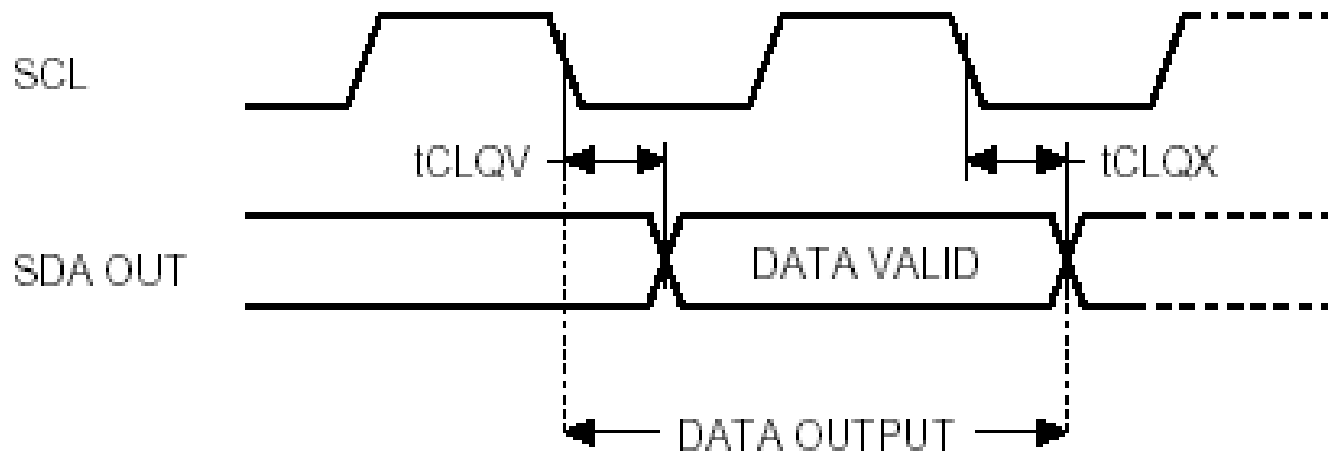
Paso 2 – (ej: envío de comandos o escritura de registros internos)
El Maestro manda la dirección interna del registro a escribir
El Esclavo responde ACK
El Maestro manda el dato
El dispositivo responde

Un esclavo no puede iniciar una transmisión.
Generalmente un sistema 1 maestro – varios esclavos



I²C: Operativa a nivel de bit

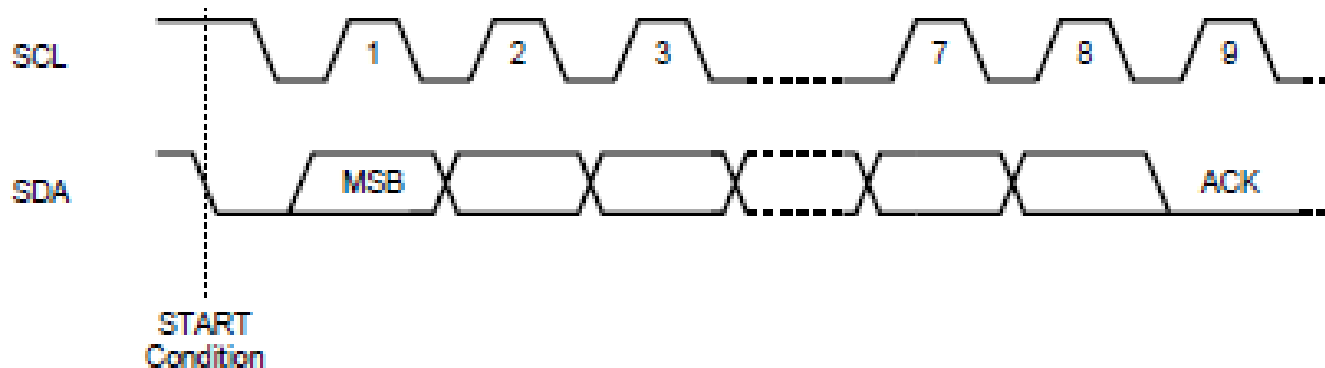
- **Primera regla:** El dato nunca puede cambiar mientras que SCL está a nivel alto (“1”). El dato solo puede cambiar a partir del flanco de bajada en SCL.
- **Segunda regla:** La validación del bit se realiza a través de un nivel alto en la línea de SCL



Bit level operation

I²C: Operativa a nivel de carácter

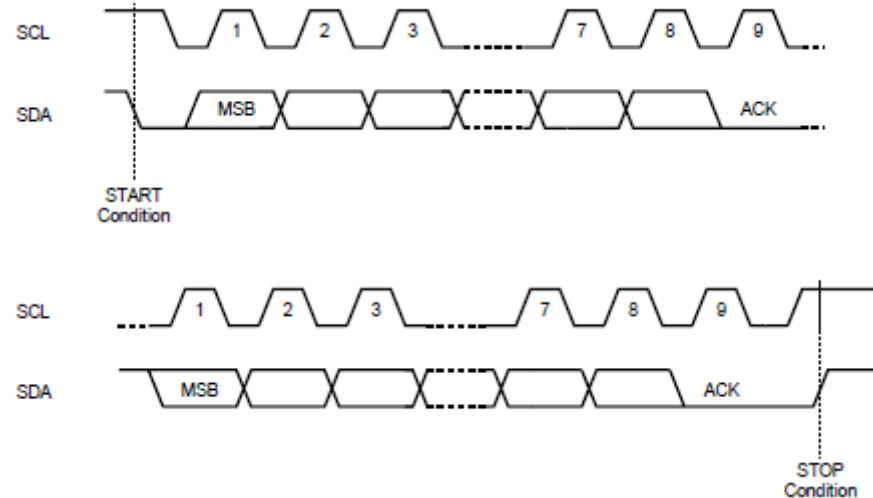
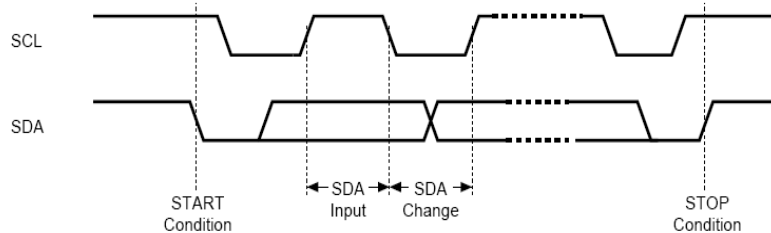
- **Tercera regla:** La primera regla sólo se puede romper para generar las condiciones de START y STOP, que encuadran la trama. Por ejemplo, en la imagen de abajo se puede ver el cronograma de una condición de START (SCL = "1" y SDA da un flanco de bajada)
- **Cuarta regla:** El protocolo proporciona un reconocimiento de recepción (ACK) para cada carácter (excepto para el último byte cuando se lee)
 - Tras el 8º bit transmitido, el transmisor deja la línea SDA en alta impedancia y la observa para detectar si el receptor la ha puesto a '0' (enviando un ACK), o la ve a '1' (indicando un NO-ACK – NAK)
- **Quinta regla:** Cada carácter se envía siempre transmitiendo el bit más significativo el primero



Character level operation

I²C: Operativa a nivel de trama

- En la imagen se puede ver una trama completa, empezando con la condición de START (SCL = "1" y flanco de bajada en SDA) antes del primer carácter y una condición de STOP (SCL = "1" y flanco de subida en SDA) tras el último carácter



Frame level operation

I²C con una memoria M24Cxx (I)



Write Control = 1 -> Memory disabled

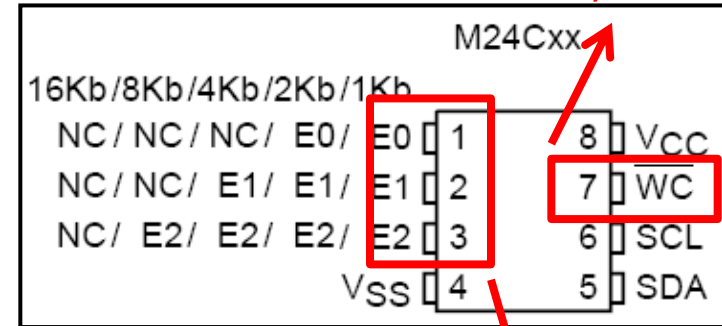


Table 1

DEVSEL	Device Type Identifier ¹				Chip Enable ^{2,3}			R \bar{W}
	b7	b6	b5	b4	b3	b2	b1	b0
M24C01 Select Code	1	0	1	0	E2	E1	E0	R \bar{W}
M24C02 Select Code	1	0	1	0	E2	E1	E0	R \bar{W}
M24C04 Select Code	1	0	1	0	E2	E1	A8	R \bar{W}
M24C08 Select Code	1	0	1	0	E2	A9	A8	R \bar{W}
M24C16 Select Code	1	0	1	0	A10	A9	A8	R \bar{W}

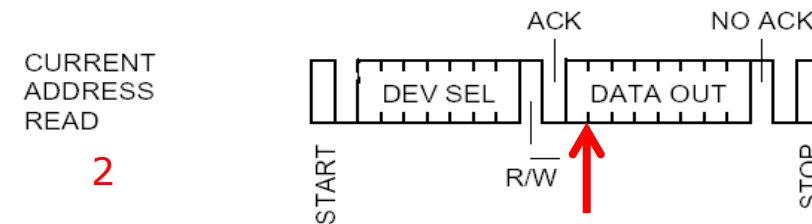
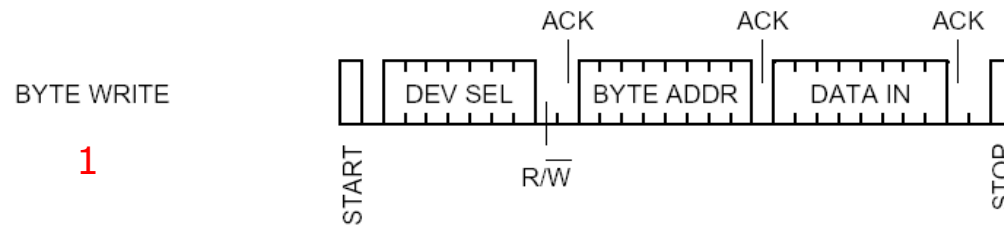
Códigos de selección para las memorias M24Cxx

0: Micro -> Memory
1: Memory -> Micro

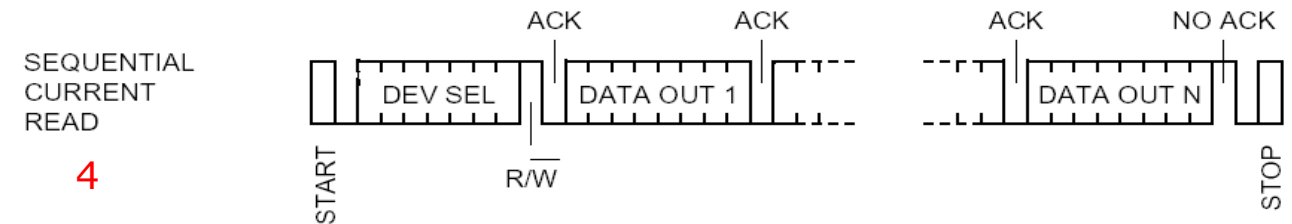
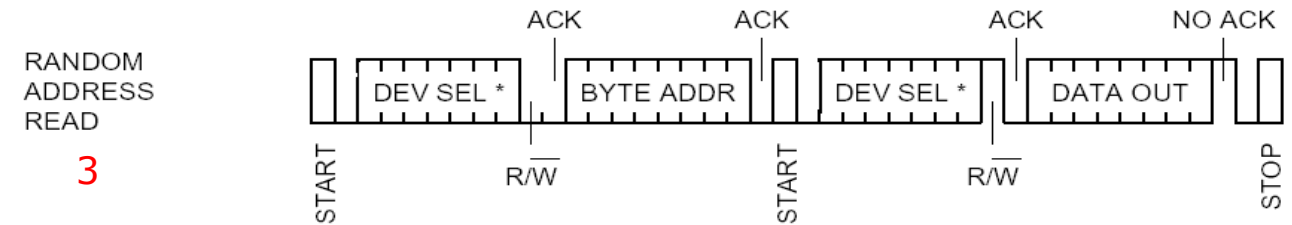
I²C con una memoria M24Cxx (II)

Operaciones con M24Cxx:

Hay que enviar un comando "Byte Write", y pararlo, para enviar un comando "Current Address Read"

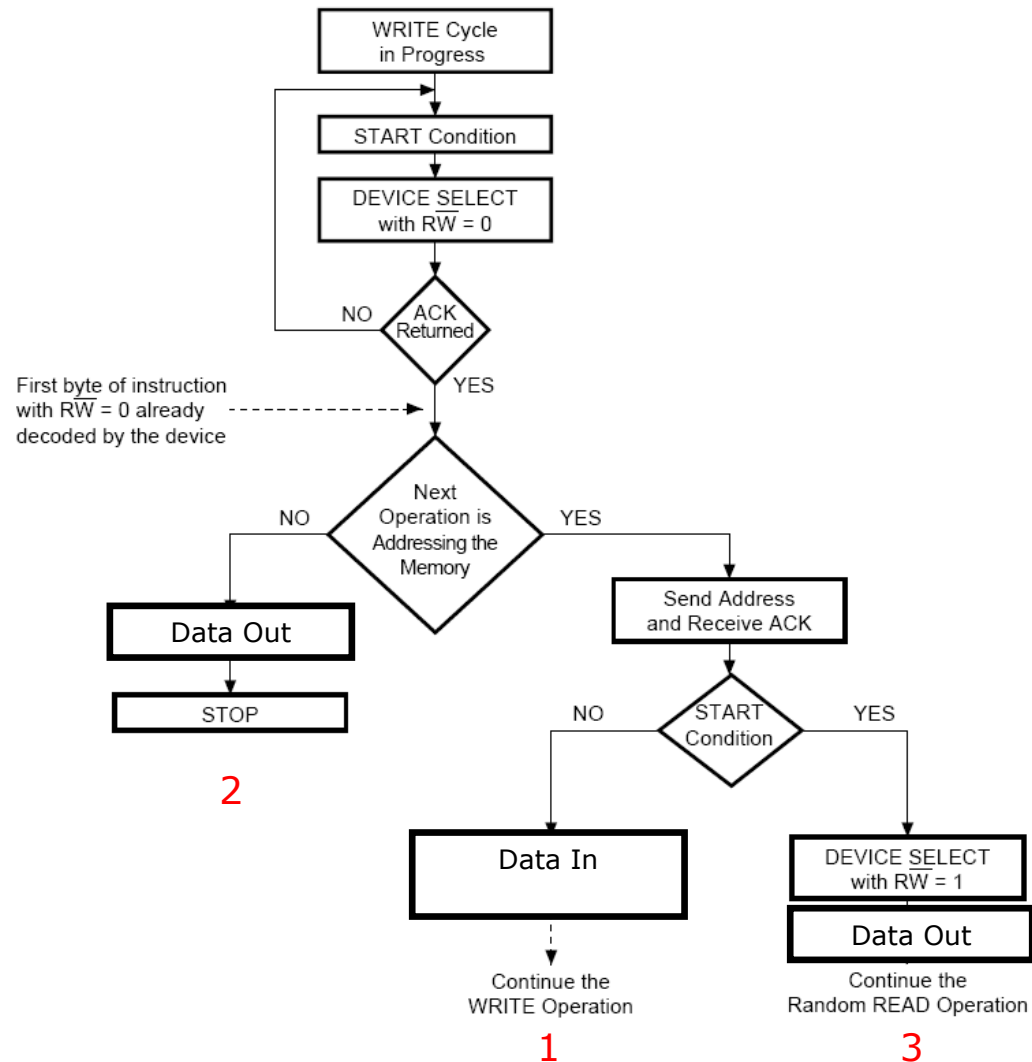


Hay siempre un puntero apuntando a la dirección actual



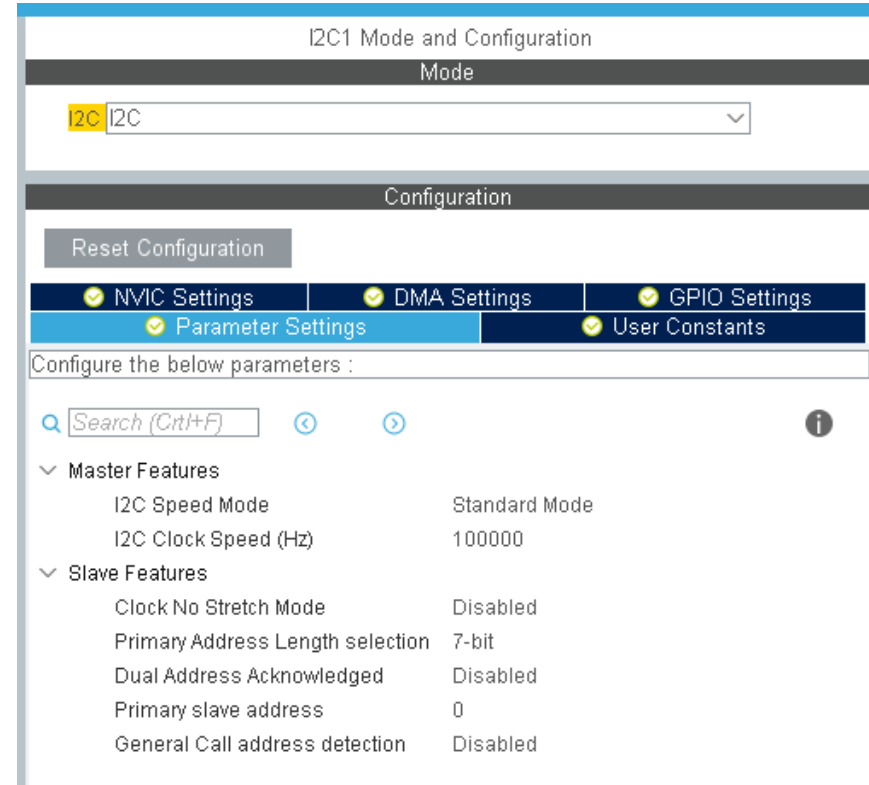
I²C con una memoria M24Cxx (III)

Diagrama de flujo para las secuencias read/write, incluyendo polling para saber si el dispositivo está ocupado o disponible



Usando I²C con las HAL

- Se va a considerar sólo el modo Master
- CubeMX:
 - Selecciona los pins para usar el I2C
 - Activa el I2C en modo I2C
 - Selecciona el Speed Mode
 - Selecciona el Clock Speed
- Funciones:
 - HAL_I2C_Init()
 - HAL_I2C_IsDeviceReady()
 - HAL_I2C_Master_Transmit(h, a, d, s, t)
 - HAL_I2C_Master_Receive(h, a, d, s, t)
 - También están disponibles con IRQs
 - Parámetros:
 - h – handler; a – dirección; d – datos; s – tamaño de los datos; t – timeout)

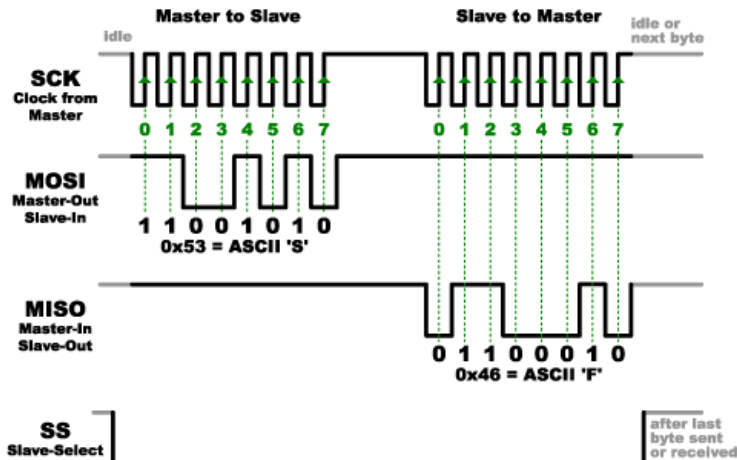
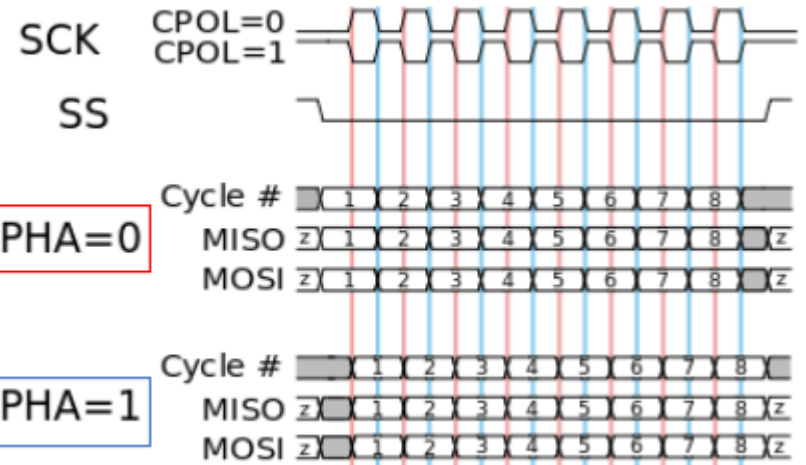
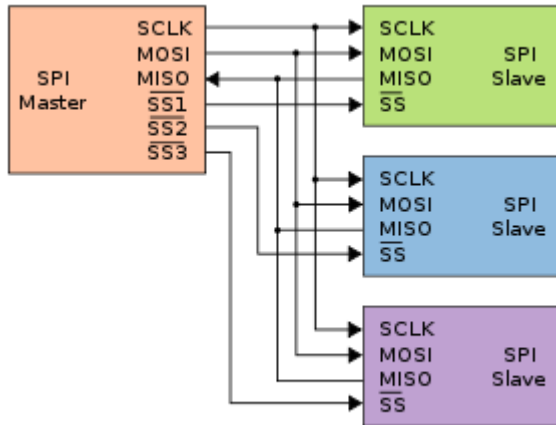


SPI

SPI: Características Generales

- Es un interfaz a 3/4 hilos (como I²C pero con 2 líneas de datos. Su nombre viene de Serial Peripheral Interface
 - SCK: Reloj
 - MOSI: Master out – Slave in (or SI)
 - MISO: Master in – Slave out (o SO)
 - CS#: Chip select
- Utiliza niveles TTL (-> 2 tensiones distintas, por ejemplo, 0-5V o similar) por lo que es imposible modular información utilizando distintas tensiones o frecuencias
- La selección del dispositivo se hace utilizando líneas hardware (Chip Select - CS)
- Tiene 4 modos de funcionamiento, aunque son prácticamente compatibles 2 a 2, y dependen de la polaridad de SCK (CPOL) y de su fase (CPHA) :
 - Si CPOL=0, el reloj se queda a '0' en reposo y cada ciclo se marca por un pulso a '1'.
 - El flanco inicial es de subida y el flanco final es de bajada
 - Si CPOL=1 el reposo es a '1' y cada ciclo se marca por pulso a '0'
 - El flanco inicial es de bajada y el flanco final es de subida
 - Si CPHA=0 la transmisión (p. ej. MO) cambia en el flanco final del ciclo anterior y el receptor (p. ej. SI) lo coge en el flanco inicial
 - Si CPHA=1 la transmisión (p. ej. MO) cambia en el flanco inicial del ciclo actual y el receptor (p. ej. SI) lo coge en el flanco final del ciclo actual
- No hay ninguna señal de inicio y final de trama, sino que la delimitación de la trama se hace por la activación de la línea CS
- Tasas de velocidad: 1 – 70 Mbps (actualmente hay adaptadores que llegan a frecuencias de reloj de 100MHz)

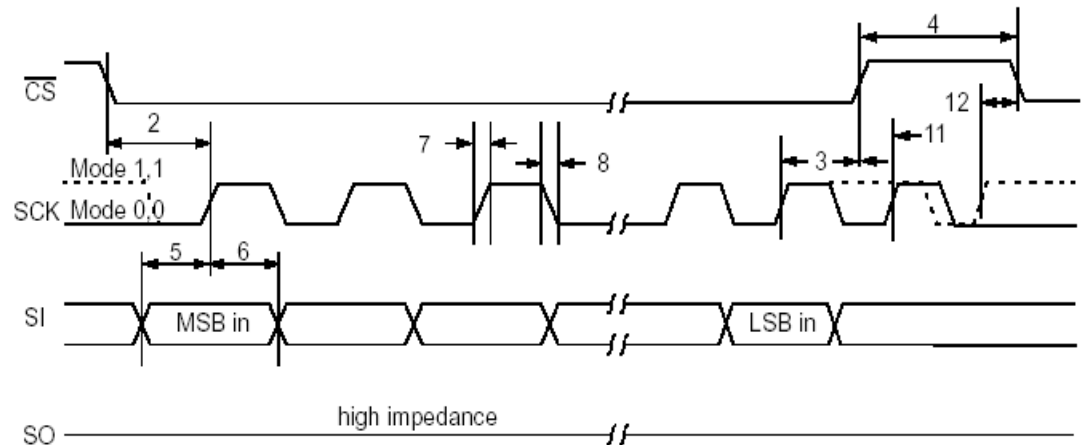
SPI: Conexión



SPI: Operativa a nivel de carácter Master -> Slave

- Para iniciar la comunicación, el maestro pone la señal CS a “0”, transmite los datos y cuando ha terminado toda la trama, coloca CS a “1” para terminar la transmisión
- No hay especificación para la longitud de los datos, es decir, pueden ser 8 o 16 bits, o incluso cualquier otro valor

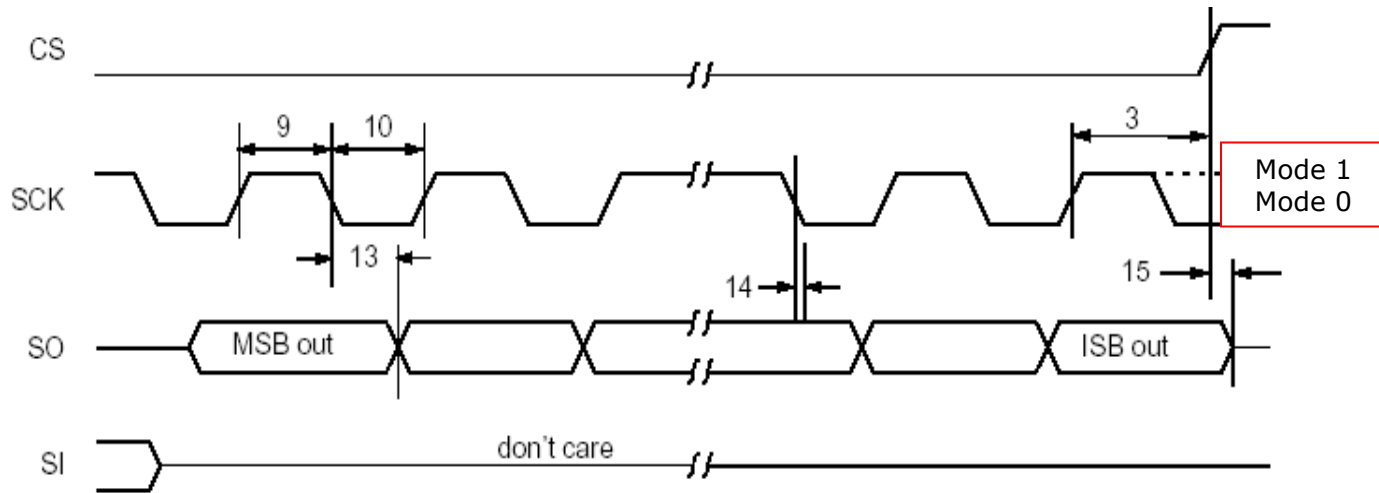
Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1



Character input by SI

SPI: Operativa a nivel de carácter Slave -> Master

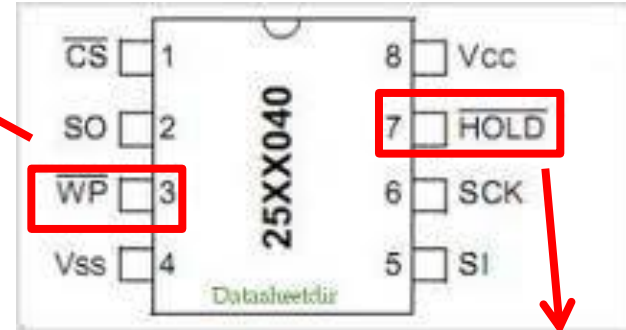
- El cronograma muestra la comunicación en la otra dirección



Character output by SO

SPI con la memoria 25AA040 (I)

Write Protection = 1 -> Write forbidden



Hold = 1 -> Memory disabled

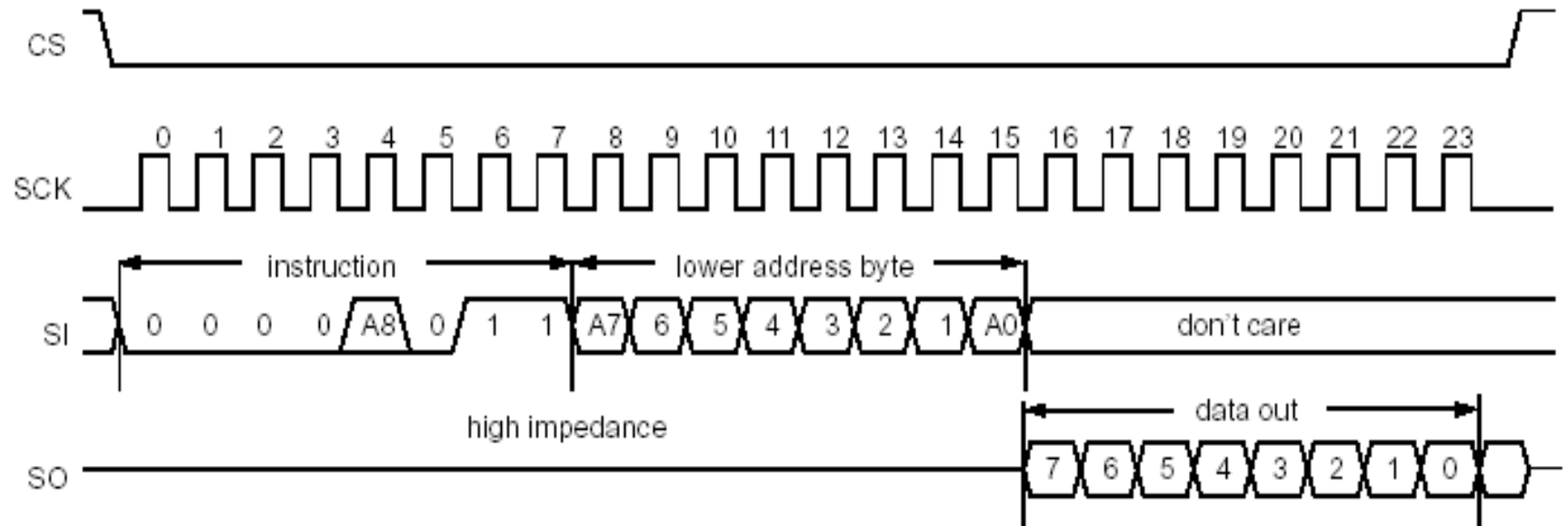
Instruction Name	Instruction Format	Description
READ	0000 A8011	Read data from memory array beginning at selected address
WRITE	0000 A8010	Write data to memory array beginning at selected address
WRDI	0000 0100	Reset the write enable latch (disable write operations)
WREN	0000 0110	Set the write enable latch (enable write operations)
RDSR	0000 0101	Read status register
WRSR	0000 0001	Write status register

Note: A8 is the 9th address bit necessary to fully address 512 bytes.

25AA040 Instructions

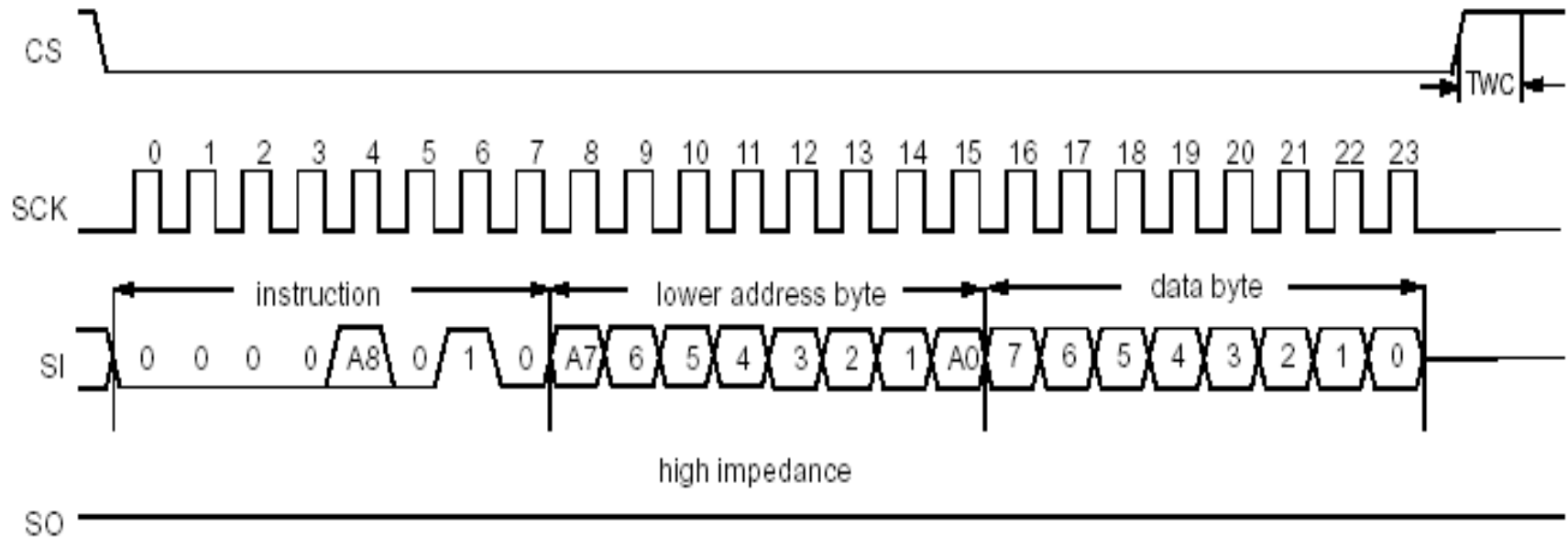
SPI con la memoria 25AA040 (II)

- En el cronograma se puede ver una instrucción de lectura en la memoria. A diferencia de lo que ocurre en I2C, aquí la instrucción contiene toda la información, incluida la dirección de lectura



READ instruction

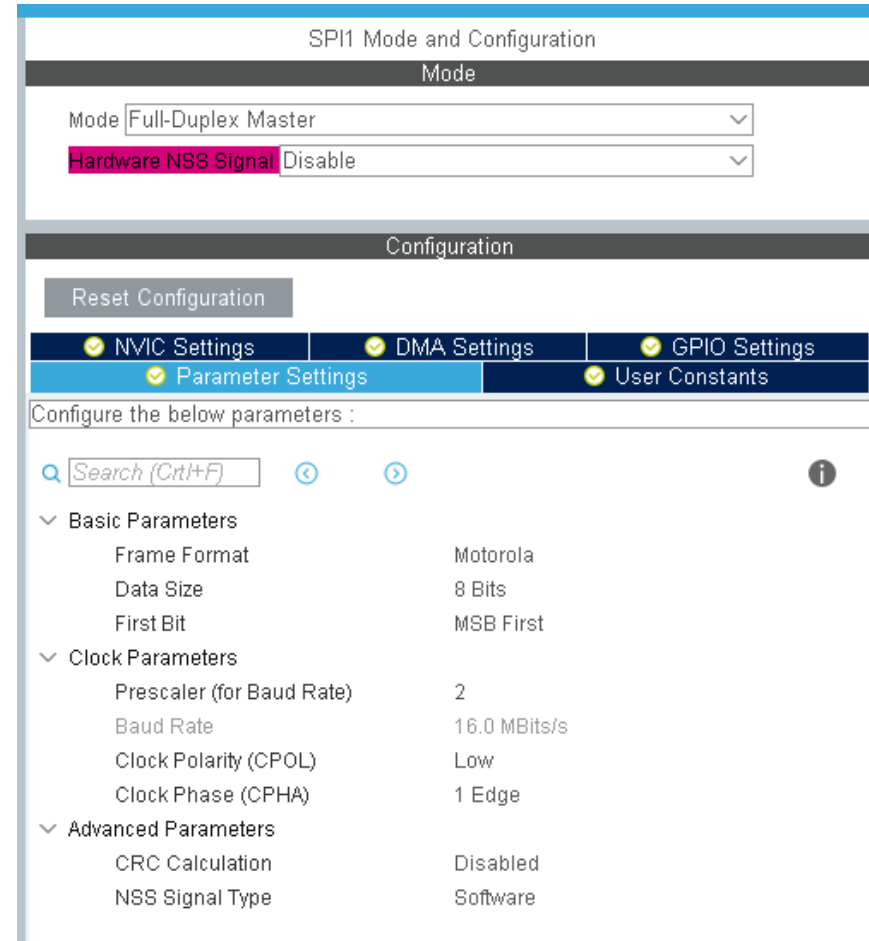
SPI con la memoria 25AA040 (III)



WRITE instruction

Uso del SPI con HAL

- Se va a considerar sólo el modo Master Full Duplex
- CubeMX:
 - Selecciona los pines a usar como SPI
 - Selecciona el modo de operación de SPI
 - Selección el tamaño de los datos
 - Selecciona cuál es el bit inicial
 - Selecciona el modo de SPI con CPOL y CPHA
- Funciones:
 - HAL_SPI_Init()
 - HAL_SPI_Transmit(h, x, s, t)
 - HAL_SPI_Receive(h, r, s, t)
 - HAL_SPI_TransmitReceive(h, x, r, s, t)
 - También disponible con IRQs
 - Parámetros:
 - h – handler; x – datos de transmisión; r – datos de recepción; s – tamaño de los datos; t – timeout)



SPI1 Mode and Configuration

Mode

Mode: Full-Duplex Master

Hardware NSS Signal: Disable

Configuration

Reset Configuration

✓ NVIC Settings | ✓ DMA Settings | ✓ GPIO Settings

✓ Parameter Settings | ✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First

Clock Parameters

Prescaler (for Baud Rate)	2
Baud Rate	16.0 Mbits/s
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge

Advanced Parameters

CRC Calculation	Disabled
NSS Signal Type	Software