



**Máster Oficial en Sistemas Electrónicos para Entornos Inteligentes**

Del mundo físico al controlador: sensores, interfaces y comunicaciones

**Guía de desarrollo GEC11b**

# **Configuración del módulo SPI del MSP430F5529 para su conexión a los dispositivos MCP4261 y MCP3008**

V1.0

Dpto. Tecnología Electrónica

Universidad de Málaga

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b>EVALUACIÓN CONTINUA EC11B .....</b>	<b>4</b>
<b>MÓDULO USCI EN MODO SPI .....</b>	<b>5</b>
<b>INICIALIZACIÓN DEL MSP430F5529 COMO MAESTRO SPI CONEXIÓN A MCP4261 Y MCP3008.....</b>	<b>6</b>
Configuración del módulo SPI del MSP430F5529 como maestro .....	7
<b>REFERENCIAS .....</b>	<b>11</b>

## INTRODUCCIÓN

Este documento constituye la guía de desarrollo para el ejercicio de evaluación continua EC11b.

Esta guía está orientada al bus SPI y centrada en la programación de la función que inicializa al microcontrolador MSP430F5529 como dispositivo maestro para su conexión a dispositivos que presentan interfaz SPI como el potenciómetro digital MCP4261 o el conversor analógico digital MCP3008.

Tal y como se recogió en el documento "[Interfaces serie síncronos: I2C y SPI](#)" el bus SPI es bastante sencillo. Para poner en funcionamiento este bus es necesario, por una parte, la conexión de los terminales de datos (MISO y MOSI), el terminal de selección del dispositivo esclavo y la señal de reloj, y por otra parte, programar las opciones de funcionamiento que se limitan a la frecuencia de la señal de reloj del bus, de su polaridad y de su fase.

Tal y como se indicó en "[Interfaces serie síncronos: I2C y SPI](#)", la polaridad y fase de la señal de reloj establecen 4 modos de operación diferentes o formatos de la señal de reloj. El objetivo no es otro que el de poder acomodarse con facilidad a la amplia variedad de dispositivos de distintos fabricantes que presentan esta interfaz serie.

En este ejercicio se va a hacer hincapié en la programación de estos parámetros de la comunicación SPI y se va a ver cómo habría que configurar al MSP430F5529 para conectarse a los dispositivos MCP4261 [\[1\]](#) y MCP3008 [\[2\]](#).

Como en principio no se dispone de los dispositivos MCP4261 y MCP3008<sup>1</sup> el ejercicio solo se va a basar en el análisis de sus hojas de características.

El **material necesario** para llevar a cabo este ejercicio de evaluación continua es el siguiente:

- Documento "[Interfaces serie síncronos: I2C y SPI](#)". En este documento se recogen los fundamentos de ambos buses. Su lectura previa es necesaria ya que muestra los aspectos básicos de operación del bus serie SPI.
- [Datasheet del microcontrolador MSP430F5529](#). Este documento es necesario ya que nos permitirá localizar el módulo hardware que implementa la interfaz SPI, así como los terminales del GPIO que se corresponden con las señales MISO, MOSI y SCLK del bus SPI. Por lo tanto se trata de un documento de consulta puntual.
- [Guía de la familia MSP430F5xx\\_6xx](#). En esta guía, en el capítulo 37, se describe la implementación que hace el fabricante Texas Instruments de la interfaz SPI para la familia MSP430F5xx\_6xx. Se trata del módulo USCI en modo SPI, los detalles de funcionamiento, y registros de programación se encuentran recogidos en este capítulo. Su lectura no es necesaria, si acaso se podrá utilizar como documento de consulta puntual.

---

<sup>1</sup> Dado que al final no ha sido posible que todos los estudiantes tengan acceso al conversor analógico digital MCP3008, el ejercicio de conexión se podrá plantear fuera de la evaluación.

- [Módulo SPI de la librería MSP430 DriverLib](#) para los dispositivos de la familia MSP430F5xx\_6xx versión 2.91.03.00. Este módulo contiene las funciones que permiten programar la interfaz SPI.
- [Datasheet del potenciómetro digital MCP4261](#). En este documento se recogen las especificaciones del potenciómetro digital y de su conexión SPI. La consulta del datasheet de este dispositivo es necesaria de forma puntual para consultar los modos de funcionamiento de los que dispone este dispositivo.
- [Datasheet del conversor analógico digital MCP3008](#). En este documento se recogen las especificaciones del conversor analógico digital y de su conexión SPI. La consulta del datasheet de este dispositivo es necesaria de forma puntual para consultar los modos de funcionamiento de los que dispone este dispositivo.

<b>EVALUACIÓN CONTINUA EC11B</b>
----------------------------------

Esta guía constituye el ejercicio de evaluación continua EC11b sobre el módulo USCI en modo SPI del microcontrolador MSP430F5529.

Este ejercicio pretende evaluar si se ha asimilado el funcionamiento del interfaz SPI y si se ha aprendido a interpretar la información dada por los fabricantes respecto a los conceptos de fase y polarización.

Las respuestas a las cuestiones planteadas en la evaluación continua, aunque están reproducidas en esta guía se deberán contestar a través del cuestionario "[Cuestionario del ejercicio de Evaluación Continua EC11b](#)" que se indica en la página de campus virtual de la asignatura.

## MÓDULO USCI EN MODO SPI

Antes de comenzar con la configuración del bus SPI se presenta el módulo que implementa de forma hardware esta interfaz en el microcontrolador MSP430F5529.

El módulo que implementa en la familia de microcontroladores MSP430F5xx\_6xx la interfaz SPI es el módulo USCI. El módulo USCI soporta múltiples modos de comunicaciones serie (UART, SPI, IrDA e I2C). Los microcontroladores de la familia MSP430F5xx\_6xx integran distintos módulos USCI que implementan diferentes modos de comunicación serie. Cada tipo de módulo USCI es nombrado con una letra distinta. El número de módulos USCI y tipo es específico de cada microcontrolador.

El microcontrolador MSP430F5529, tal y como se ve en la Figura 1, posee dos tipos de módulos USCI: USCI\_A y USCI\_B.

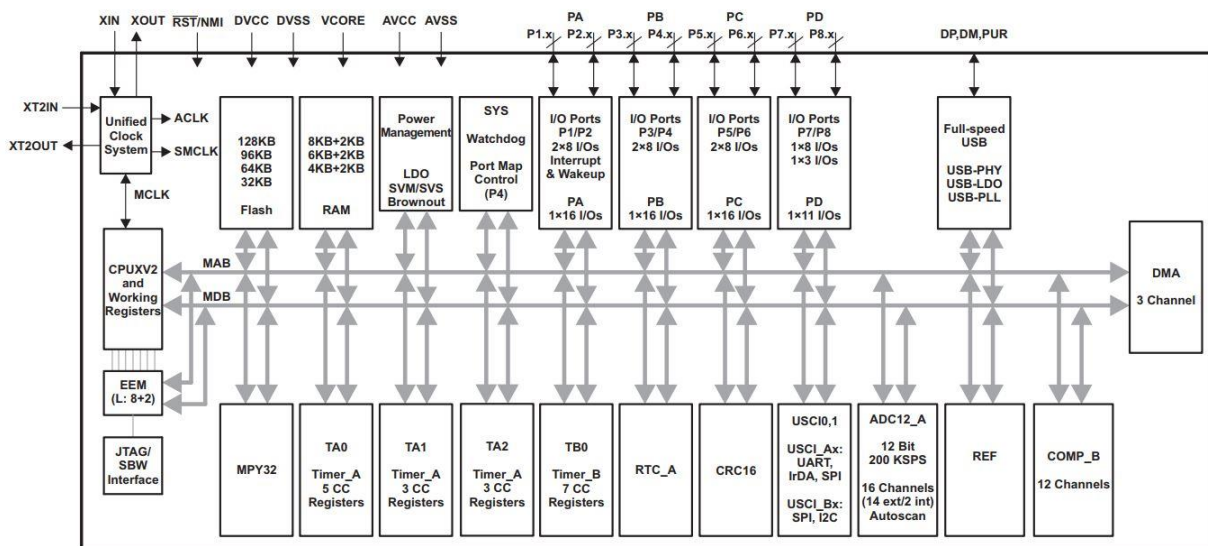


Figura 1. Diagrama de bloques del microcontrolador MSP430F5529. Fuente [3]

Tanto la USCI\_A como la USCI\_B soportan el modo SPI. Así el microcontrolador posee cuatro módulos SPI: USCI\_A0, USCI\_A1, USCI\_B0 y USCI\_B1.

Los módulos USCI\_A0 y USCI\_A1 comparten la librería `usci_a_spi` y los módulos USCI\_B0 y USCI\_B1 comparten la librería `usci_b_spi`. Esto es así porque existen registros de programación independientes para módulos tipo A y para los módulos tipo B.

## INICIALIZACIÓN DEL MSP430F5529 COMO MAESTRO SPI CONEXIÓN A MCP4261 Y MCP3008

En este apartado vamos a completar el código que se encarga de programar al microcontrolador MSP430F5529 como el dispositivo maestro del bus SPI que es capaz de conectarse con los dispositivos MCP4261 y MCP3008. Será por tanto necesario establecer cuáles van a ser los terminales de conexión y qué modos de fase y polarización soportan los dispositivos a los que se quiere conectar. En este sentido, **es importante tener en cuenta que cada fabricante de los dispositivos maestro o esclavo puede expresar con distintos valores los conceptos de polaridad** (nivel de la señal de reloj en reposo) **y fase** (flancos en los que se realizan las escrituras y lecturas de datos).

A continuación en Código 1 se muestra el programa que permite inicializar al maestro SPI MSP430F5529 para su comunicación SPI con el dispositivo MCP4261. En dicho código aparece también el diagrama de conexión de los terminales del microcontrolador y el dispositivo.

```
//          MSP430F5529
//          -----
//          |          |
//          |    (40)P3.3|-----/CS----->  #CS MCP4261
//          |    (37)P3.0|-----SIMO----->  SDI MCP4261
//          |    (38)P3.1|<-----SOMI-----  SDO MCP4261
//          |    (39)P3.2|-----UCLK----->  SDK MCP4261
//          |          |
//          |          |

1. //Set /CS Inactive State ("1") to start a command /CS signal
2. //must transition from inactive state to active state ("0")
3. uint8_t returnValue = 0x00;
4. void main(void){
5.     WDT_A_hold(WDT_A_BASE);

6.     GPIO_setOutputHighOnPin(GPIO_PORT_P3,GPIO_PIN3);

7.     GPIO_setAsOutputPin(GPIO_PORT_P3,GPIO_PIN3);

8.     //P3.0,1,2 option select
9.     GPIO_setAsPeripheralModuleFunctionInputPin(
        GPIO_PORT_P3,GPIO_PIN0 + GPIO_PIN1 + GPIO_PIN2);

10. //Initialize Master
11. USCI_B_SPI_initMasterParam param = {0};
12. param.selectClockSource = USCI_B_SPI_CLOCKSOURCE_SMCLK;
13. param.clockSourceFrequency = UCS_getSMCLK();
14. param.desiredSpiClock = 400000;
15. param.msbFirst = USCI_B_SPI_MSB_FIRST;
16. param.clockPhase = X;
17. param.clockPolarity = Y;
18. returnValue = USCI_B_SPI_initMaster(Z, &param);

19. if (STATUS_FAIL == returnValue){
20.     return;
21. ....
22. }
```

Código 1. Extracto de código para la programación del microcontrolador MSP430F5529 como maestro SPI conectado al dispositivo MCP4261c

## **Configuración del módulo SPI del MSP430F5529 como maestro**

En Código 1 aparecen configurados los terminales del microcontrolador que se van a conectar a los terminales del dispositivo SPI, en este caso al dispositivo MCP4261 (líneas de la 6 a la 9). La función `GPIO_setAsPeripheralModuleFunctionInputPin (GPIO_PORT_P3,GPIO_PIN0+GPIO_PIN1 + GPIO_PIN2)`; configura como función específica SPI los terminales (P3.0,P3.1 y P3.3).

La función que se encarga de configurar el módulo SPI del microcontrolador MSP430F5529 como maestro es la función `USCI_B_SPI_initMaster(USCI_Z_BASE, &param)`, cuya estructura de parámetros se inicializa en las líneas de la 11 a la 17 del Código 1. Esta función también deshabilita el módulo SPI para la programación de sus parámetros. Esta función se encuentra descrita con detalle en la [librería usci\\_b\\_SPI](#).

Dentro de los parámetros que es necesario configurar para programar al maestro SPI se encuentran los siguientes:

- El campo `msbFirst` que indica el primer bit en ser transmitido o recibido (MSB o LSB). En este caso se ha programado que el primer bit recibido o transmitido sea el más significativo, `USCI_B_SPI_MSB_FIRST`, tal y como se necesita en la comunicación SPI con el dispositivo MCP4261 (Línea 15).
- Los parámetros que tienen que ver con la señal de reloj:
  - Frecuencia de la señal de reloj: Entre los distintos campos que se deben asignar se encuentra la definición de la fuente de reloj utilizada por el generador de reloj del módulo SPI, que en este caso es la señal de reloj SMCLK, así como el valor a la que se ha programado, que en este caso es su valor por defecto, esto es, aproximadamente un 1Mhz. En el código el valor al campo `clockSourceFrequency` se asigna con la función `UCS_getSMCLK()`, que lo que hace es leer el valor de los bits del registro que programa el reloj SCMLK para ver a qué frecuencia se encuentra. El campo de la estructura `desiredSpiClock` define la frecuencia deseada para la señal de reloj UCLK, que en este caso se ha fijado a 400KHz. Dividiendo la frecuencia de la señal de reloj SMCLK por el valor del campo `desiredSpiClock` se obtiene el valor del divisor que actúa en el generador de reloj del SPI para obtener la señal UCLK de aproximadamente 400KHz. Esta frecuencia está dentro de las admitidas por el dispositivo MCP4261<sup>2</sup>.
  - Polaridad y fase de la señal de reloj: Los campos `clockPhase` y `clockPolarity` determinan la fase y la polaridad de la señal de reloj. La combinación de estos dos campos da lugar a 4 posibles modos de funcionamiento SPI. Para establecer el valor

---

<sup>2</sup> El dispositivo MCP4261 soporta transferencias SPI hasta 10MHz. El dispositivo MCP3008 la máxima velocidad de transferencia soportada depende de la tensión de alimentación para una tensión de 2.7v está en torno a los 1.35MHz. En el caso del microcontrolador MSP430F5529 la máxima velocidad de transferencia viene dada por el reloj del sistema.



de estos dos campos hay que tener en cuenta qué modos tiene implementados el dispositivo con el que se quiere conectar. Determinar cuáles son estos parámetros y realizar una interpretación correcta del datasheet son los objetivos del presente ejercicio.

### CUESTIÓN 1:

Completar las líneas de código que permiten configurar al microcontrolador MSP430F5529 como dispositivo maestro del bus SPI al que se conecta el dispositivo MCP4261 como dispositivo esclavo. Determinar las etiquetas que les corresponden a las incógnitas **X**, **Y** y **Z**, del código.

Completar **X** e **Y** para todos los modos soportados por el dispositivo MCP4261.

Para poder responder a esta pregunta hay que tener en cuenta a qué terminales del microcontrolador se han conectado los del dispositivo MCP4261 según el Código 1, así como algunos datos de las hojas de características de los dispositivos implicados.

### Modos de funcionamiento SPI implementados por el MSP430F5529

El microcontrolador MSP430F5529 implementa los 4 modos de funcionamiento de fase y polaridad que establece el estándar SPI. La programación de estos modos se hace a través de 2 bits del registro de control 0 del módulo SPI, tal y como se ve en la Imagen 1.

#### 37.5.1 UCBxCTL0 Register

USCI\_Bx Control Register 0

Figure 37-16. UCBxCTL0 Register

7	6	5	4	3	2	1	0
UCCKPH	UCCKPL	UCMSB	UC7BIT	UCMST	UCMODEx		UCSYNC
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

Can be modified only when UCSWRST = 1.

Table 37-15. UCBxCTL0 Register Description

Bit	Field	Type	Reset	Description
7	UCCKPH	RW	0h	Clock phase select 0b = Data is changed on the first UCLK edge and captured on the following edge. 1b = Data is captured on the first UCLK edge and changed on the following edge.
6	UCCKPL	RW	0h	Clock polarity select 0b = The inactive state is low. 1b = The inactive state is high.

Imagen 1. Bits del registro de control 0 del módulo SPI que programan la fase y la polaridad de la señal de reloj. Fuente [\[4\]](#)

El cronograma en el que se expresa cuándo el microcontrolador MSP430F5529, actuando como maestro, realiza la lectura de datos del bus (captura de datos) y la escritura de datos en el bus (cambio de datos) es el que se ve en la Imagen 2.

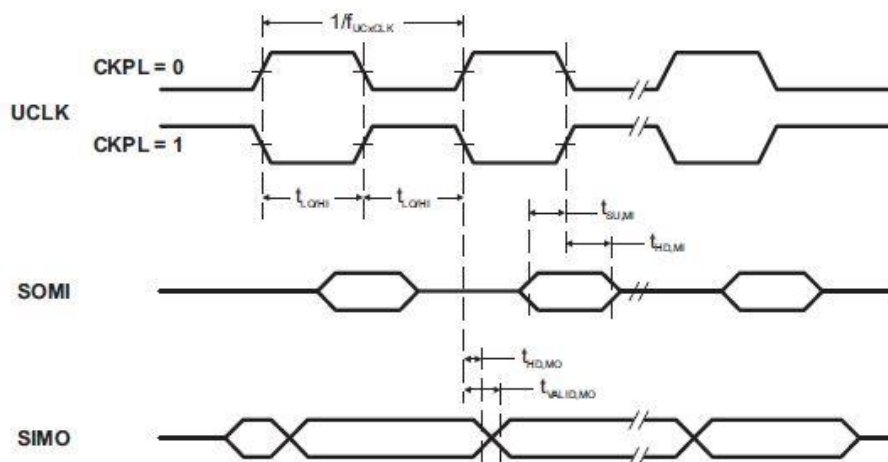


Figure 5-11. SPI Master Mode, CKPH = 0

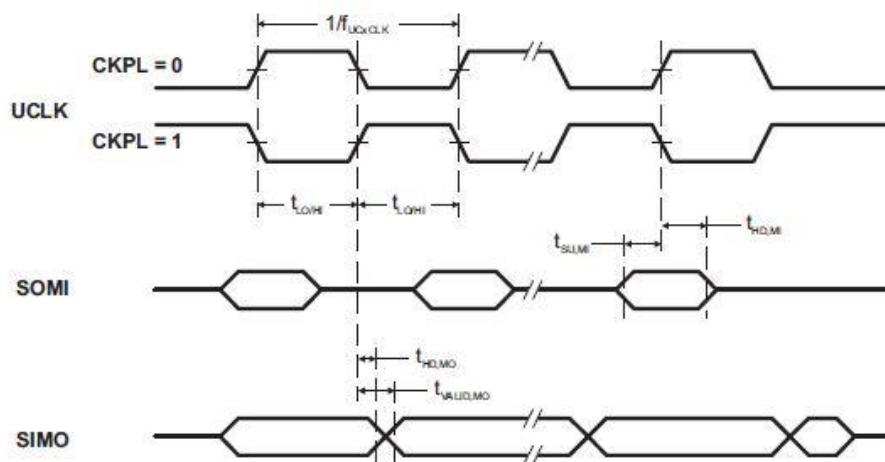


Figure 5-12. SPI Master Mode, CKPH = 1

Imagen 2. Cronograma de los modos de funcionamiento del módulo SPI del microcontrolador MSP430F5529. Fuente [\[3\]](#)

Las etiquetas que se definen en la librería del módulo SPI (Módulo SPI de la librería MSP430 DriverLib) son las que aparecen en la Imagen 3.

uint8\_t USCI\_B\_SPI\_initMasterParam::clockPhase

Is clock phase select.

Valid values are:

- USCI\_B\_SPI\_PHASE\_DATA\_CHANGED\_ONFIRST\_CAPTURED\_ON\_NEXT [Default]
- USCI\_B\_SPI\_PHASE\_DATA\_CAPTURED\_ONFIRST\_CHANGED\_ON\_NEXT

Referenced by USCI\_B\_SPI\_initMaster().

uint8\_t USCI\_B\_SPI\_initMasterParam::clockPolarity

Valid values are:

- USCI\_B\_SPI\_CLOCKPOLARITY\_INACTIVITY\_HIGH
- USCI\_B\_SPI\_CLOCKPOLARITY\_INACTIVITY\_LOW [Default]

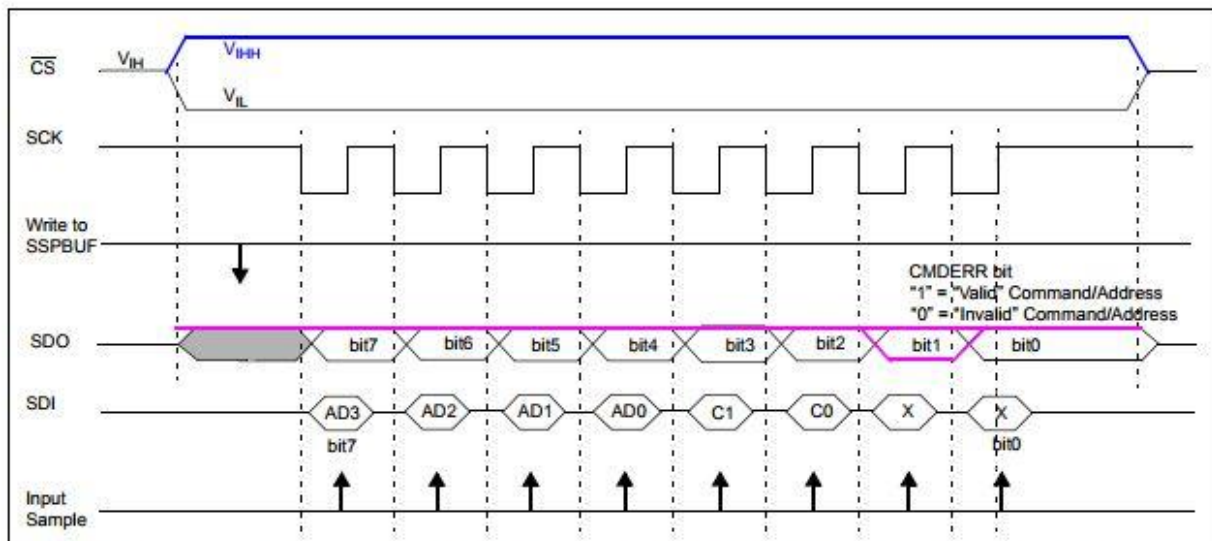
Referenced by USCI\_B\_SPI\_initMaster().

Imagen 3. Etiquetas definidas en la librería usci\_b\_spi

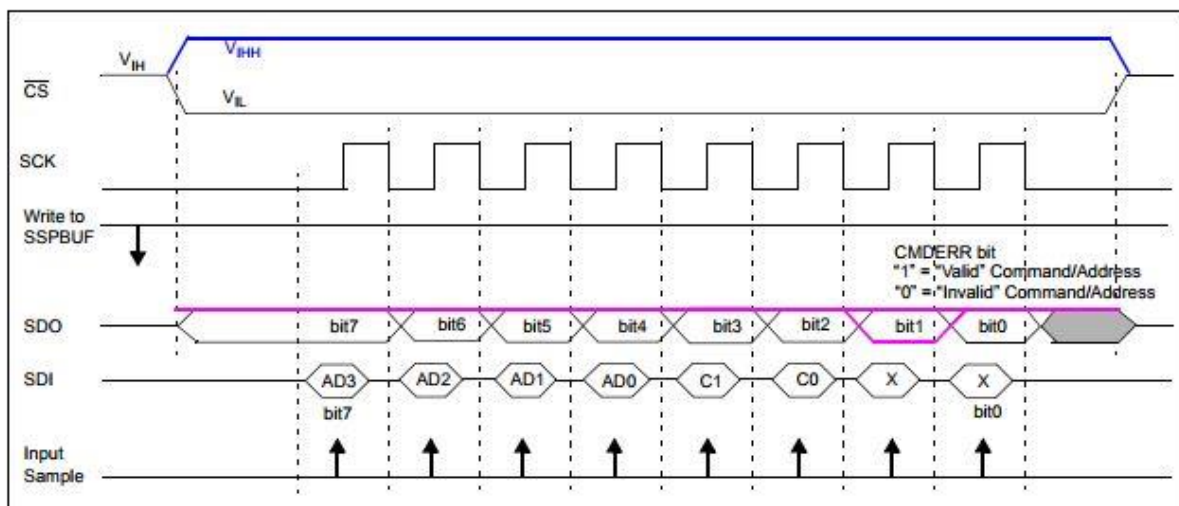
### Modos de funcionamiento SPI implementados por el dispositivo MCP4261

El dispositivo MCP4261 solo implementa 2 modos de los 4 modos que presenta el estándar SPI. Los dos modos implementados son los que denomina el fabricante de este dispositivo: MODE 0,0 y MODE 1,1.

Ambos modos se pueden ver en la Imagen 4, en la que se representan los cronogramas de transferencias de datos de tamaño 8 bits. En el cronograma se indican los flancos de la señal de reloj en los que se produce el cambio de datos a la salida del dispositivo (terminal SDO) y la captura de datos en la entrada del dispositivo (terminal SDI).



**FIGURE 6-7:** 8-Bit Commands (Increment, Decrement, Modify Write Protect or WiperLock Technology) - SPI Waveform with PIC MCU (Mode 1,1).



**FIGURE 6-8:** 8-Bit Commands (Increment, Decrement, Modify Write Protect or WiperLock Technology) - SPI Waveform with PIC MCU (Mode 0,0).

Imagen 4. Cronogramas de los modos SPI MODE 1,1 y MODE 0,0 del MCP4261

En resumen, para responder a esta primera cuestión se debe tener claro cómo funciona el dispositivo esclavo.

**IMPORTANTE:** para solucionar este ejercicio hay que tener en cuenta que la programación sólo se puede hacer a nivel del dispositivo maestro (microcontrolador), porque no hay nada en el dispositivo esclavo que permita programar uno u otro modo, sino que la recepción en el esclavo de la señal de reloj, con su nivel de reposo, establece directamente el modo en el esclavo. Por lo tanto, hay que ver con qué fase y polaridad hay que programar al dispositivo maestro para que se pueda producir la comunicación correcta en base a los modos que tiene implementados el dispositivo esclavo.

#### CUESTIÓN 2:

El dispositivo MCP4261 solo implementa 2 modos de los 4 modos SPI estándar. Estos modos son los que denomina: MODE 0,0 y MODE 1,1. En el caso del dispositivo MCP3008, que es del mismo fabricante, Microchip, también solo soporta estos modos.

**Explica y reflexiona por qué razón crees que en estos dispositivos esclavos solo se implementan dos modos** de los cuatro modos establecidos en el estándar de SPI. También reflexiona **por qué se han implementado estos dos modos (modo 0,0 y modo 1,1) y no los otros dos modos (modo 0,1 y modo 1,0).**

<b>REFERENCIAS</b>
--------------------

[1] Microchip. Datasheet MCP4261 7/8-Bit Single/Dual SPI Digital POT with Non-Volatile Memory (Rev. 2008)

[2] Microchip. Datasheet MCP3008. 8-Channel 10-bit A/D Converters with SPI Serial Interface (Rev. 2008)

[3] Texas Instruments Datasheet MSP430F5529. SLAS590M (Rev. Noviembre2015)

[4] Texas Instruments. User's Guide. MSP430x5xx and MSP430x6xx Family SLAU208P (Rev. Octubre 2016)