

USANDO EL PROTOCOLO DE COMUNICACIÓN SERIAL SÍNCRONO SPI EN LA IMPLEMENTACIÓN DE INSTRUMENTACIÓN DIGITAL

Medina Rojas, Alexander, Sierra Solano, Milton y Gómez Tunjano, Daniel-
Fundación Universitaria de San Gil –UNISANGIL–
Facultad de Ciencias Naturales e Ingeniería
Ingeniería Electrónica.
San Gil, Santander, Colombia

amedina@unisangil.edu.co
miltonsierra@unisangil.edu.co
danielgomez1@unisangil.edu.co

Resumen— El propósito de este artículo es explicar y describir el bus de interfaz serial usando un microcontrolador Freescale de la familia MC9S08. En este informe hay dos aplicaciones realizadas; la primera es un monitoreo de temperatura con un sensor digital de temperatura TMP124; y la segunda aplicación es la configuración de un convertor digital a analógico (DAC) usando protocolo de interfaz serial (SPI).

Palabras clave— MC9S08JM60, SPI, sensor digital de temperatura, TMP124, convertidor digital-analógico, TLV5620

Abstract— The purpose of this report is explain and describe the serial peripheral interface bus using a Freescale microcontroller of the family MC9S08. In this report there are two applications realized, the first is temperature monitoring with digital temperature sensor TMP124, and the second is the configuration of a digital to analog converter (DAC) using serial interface (SPI).

Keywords— MC9S08JM60, SPI, Digital Temperature Sensor, TMP124, D/A Converter, TLV5620

I. INTRODUCCIÓN

A medida que la complejidad en las aplicaciones de usuario aumenta, los diseñadores de sistemas electrónicos necesitan comunicar dos o más microprocesadores con integrados de propósito específico para proporcionar funcionalidad en la aplicación. La comunicación entre diferentes integrados puede llegar a ser una tarea difícil, en especial porque existen fabricantes que usan diferentes arquitecturas en el diseño de estos dispositivos. Una solución a este problema es el uso de protocolos estándar de comunicación serial.

El bus serial de interfaz de periféricos (SPI) es un protocolo que ofrece facilidad para la comunicación entre diferentes integrados. Los microcontroladores de última generación incluyen dentro de sus periféricos al menos un módulo de comunicación serial SPI. Uno de estos microcontroladores es el MC9S08JM60 [1] del fabricante “Freescale”; por su alta gama de funciones es ideal para desarrollar aplicaciones en el campo del procesamiento digital de señales, en especial para desempeñar tareas de comunicación entre diferentes dispositivos necesarios para la adquisición y entrega de datos en este campo de la electrónica.

El objetivo de los fabricantes de circuitos integrados es facilitar el desarrollo de sistemas embebidos. Para tal fin, se pueden encontrar en el mercado dispositivos digitales como sensores de temperatura o convertidores de datos que usan comunicación serial SPI; de ahí la importancia de comprender el funcionamiento de este protocolo, además de la puesta en marcha sobre algún dispositivo, en este caso un sensor digital de temperatura TMP124 [2] y un convertidor de digital a analógico TLV5620 [3].

II. DEFINICIÓN SPI

El bus serial de interfaz de periféricos (SPI) es un protocolo síncrono estándar para datos, desarrollado por Motorola para facilitar la comunicación entre circuitos integrados (convertidores analógicos a digitales, memorias, etc.) [4]. El tipo de comunicación es full dúplex¹ y se realiza entre un dispositivo que trabaja como maestro y otro que trabaja como esclavo. El control de la transferencia la hace por completo el dispositivo maestro.

A. Descripción de las señales

- 1) *Salida Serial (MOSI)*: Esta señal está configurada para transmitir datos desde el dispositivo maestro o recibir datos si es esclavo.
- 2) *Entrada Serial (MISO)*: Esta señal está configurada para transmitir datos desde el dispositivo esclavo o recibir datos si es maestro.
- 3) *Reloj (CLCK)*: Es una salida de reloj serial que se emplea para sincronizar los datos que se transmiten y se reciben
- 4) *Selección del esclavo (SS)*: Esta señal se usa para configurar al dispositivo como maestro.

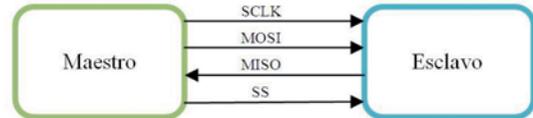


Figura 1. Señales comunicación SPI

Cada sistema SPI consiste en un maestro y uno o más esclavos (Fig. 1); el maestro inicializa la comunicación habilitando la señal SS. Cuando el dispositivo esclavo es seleccionado con las señales de reloj, se inicia la transmisión de los datos por la salida serial (MOSI); el maestro recibe por la entrada serial (MISO) y envía un bit por cada flanco de reloj. El maestro puede terminar la comunicación en cualquier momento.

III. DESCRIPCIÓN MÓDULO SPI MICROCONTROLADOR

Entre las principales partes de la arquitectura del SPI del microcontrolador MC9S08JM60 se tiene: los registros de datos, estado y control, un generador para la tasa de baudios², una unidad de control maestro/esclavo y los registros de desplazamiento para la entrada y salida de datos (Fig. 2).

Este módulo permite la comunicación del microcontrolador con cualquier dispositivo periférico; la comunicación que se realiza es dúplex, síncrona y serial [5].

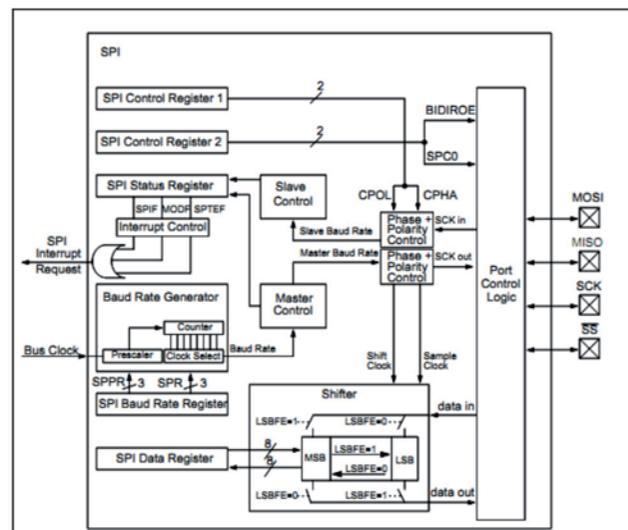


Figura 2. Arquitectura interna módulo SPI [5]

A. Registros de módulo SPI

Para poder manejar cualquier dispositivo que tenga comunicación SPI, es necesario realizar la configuración del módulo, la cual depende exclusivamente del integrado que se desea controlar o poner en funcionamiento. El fabricante de cada dispositivo es el que orienta esta tarea.

En la tabla 1 se tiene el banco de registros de módulo SPI del microcontrolador JM60.

BANCO DE REGISTROS MÓDULO SPI

	7	6	5	4	3	2	1	0
SPIC1	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
SPIC2	0	0	0	MOD	BIDR	0	SWAI	SPC0
SPIBR		SPPR0	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
SPIS	SPRF	0	SPTIEF	MODF	0	0	0	0
SPID	R7	R6	R5	R4	R3	R2	R1	R0

En general se cuenta con seis registros; cada uno de ellos compuestos de una serie de bits que tienen una función determinada [1].

- Registro de control 1: Este registro es el encargado de habilitar el módulo, además de indicar si se trabaja en el modo maestro o esclavo, también configura el reloj síncrono.
- Registro de control 2: Tienen la tarea de configurar si se trabaja de forma bidireccional, además de determinar el instante para la transmisión.
- Registro tasa de baudios: Los bits de este registro especifican la velocidad de transmisión de los datos.
- Registro de estado: Las banderas³ que contiene este registro indican si se han recibido o terminado de transmitir una serie de datos.
- Registro de datos: Este registro es el que almacena de forma temporal los datos que se están transmitiendo y recibiendo en una comunicación.

IV. RESULTADO DE LA IMPLEMENTACIÓN DEL PROTOCOLO SPI EN INSTRUMENTACIÓN DIGITAL

A continuación se presenta el resultado del uso del protocolo SPI para dos aplicaciones específicas, las cuales consisten en comunicación con un sensor digital de temperatura y la configuración de un conversor digital a analógico.

A. Sensor digital de temperatura TMP124

Este es un sensor de tipo digital, programable mediante comunicación serial SPI (fig. 3). Es de alta precisión con un rango de lectura de temperatura entre -40°C a $+125^{\circ}\text{C}$ con la posibilidad de programar umbral alto y bajo para aplicaciones de termostato.

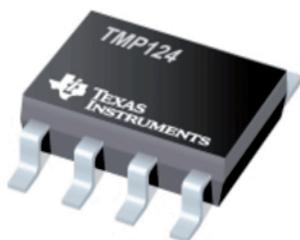


Figura 3. Imagen sensor de temperatura [2]

- 1) *Aplicaciones útiles del sensor TMP124:* Este tipo de sensor es ideal para aplicaciones de monitoreo térmico de fuentes de alimentación, en equipos de electrónica de consumo como celulares, cargadores de baterías, etc. [2]. También puede ser usado para monitoreo de

temperatura ambiente y controles de termostato⁴ en equipos electrónicos de oficina.

- 2) *Esquema interno del sensor TMP124:* El TMP124 internamente contiene un sensor de temperatura a base de diodo y un conversor de analógico a digital, el cual se comunica con la interfaz serial. El pin 1 SO/I es usado para comunicarse de modo serial con el micro controlador; el pin 2 SCK es la entrada de los ciclos de reloj de protocolo SPI; el pin 7 CS sirve para habilitar o deshabilitar la comunicación con el sensor; y el pin 5 ALERT es de alarma programable para aplicaciones de termostato (Figura 4).

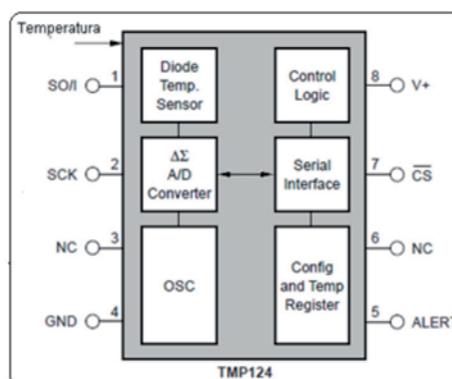


Figura 4. Esquema interno sensor TMP124 [2]

- 3) *Comunicación con el sensor TMP124:* Este sensor se comunica con el microcontrolador usando el SPI en modo bidireccional. Por lo tanto, primero se envía un comando de lectura y luego se recibe una respuesta del sensor. Debido a lo anterior, los pines MISO y SS no son necesarios en esta aplicación. En la Fig. 5 se puede observar la conexión típica entre la unidad microcontrolada y el sensor.

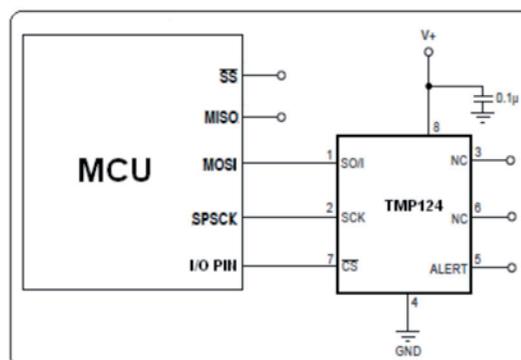


Figura 5. Conexión típica del TPMP124

Para establecer comunicación con el TMP124 es necesario que el pin 7 del sensor (CS) reciba un nivel lógico bajo. La primera trama⁵ de datos de 16 bits retorna el dato de temperatura. Si el usuario mantiene el pin (CS) en nivel bajo se pueden enviar y recibir varias tramas que podrán ser comandos de lectura o escritura (Fig. 6).

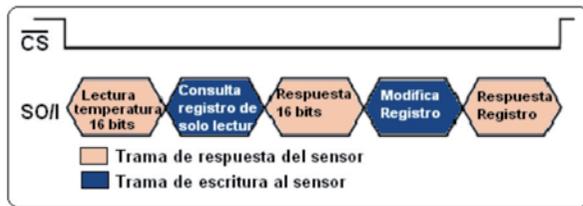


Figura 6. Secuencia múltiple de comandos

- 4) *Registros configurables del sensor:* Este sensor tiene cuatro registros configurables de 16 bits. Un registro para fijar umbral alto, un registro de umbral bajo, el registro de lectura de temperatura y un registro de configuración. En especial el registro de configuración tiene dos bits para ajustar la resolución entre 0.5°C y 0.0625°C. También hay asignados dos bits para el tiempo retardo entre cada conversión. Esta última característica permite eliminar conversiones innecesarias para disminuir el consumo de corriente aproximadamente a 20[uA].
- 5) *Estrategia de programación:* El diagrama de flujo describe únicamente lo relacionado con la programación del sensor de temperatura (fig. 7). La visualización de la temperatura se hizo mediante una pantalla LCD alfanumérica⁶. Todos los registros del sensor de temperatura y el resto de código de la aplicación se realizó en lenguaje ANSI C [6] debido a su versatilidad y uso generalizado en sistemas embebidos. El resultado final de esta aplicación se puede observar en la figura 8.

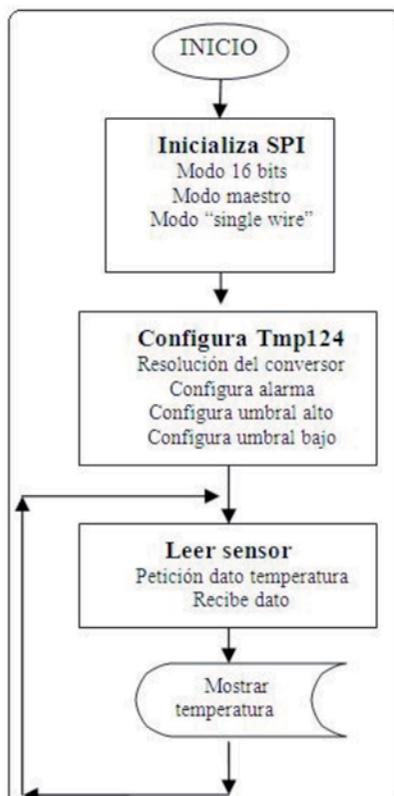


Figura 7. Diagrama de flujo configuración TMP124

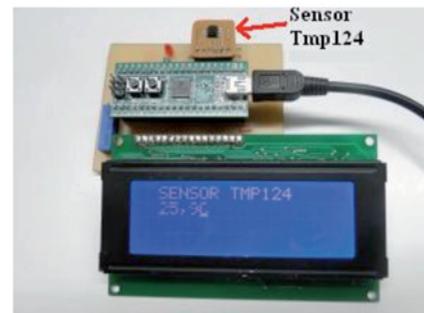


Figura 8. Registro fotográfico aplicación TMP124

B. Conversor digital/análogo (DAC) TLV5620

El convertidor digital-análogo (DAC) TLV5620 [3] está compuesto internamente por cuatro DACs. El núcleo de cada DAC es un resistor de 256 "taps"; esto quiere decir que la salida del DAC proporciona 256 niveles de voltaje según el código binario. La comunicación con este circuito integrado es mediante protocolo serial basado en tecnología CMOS, haciéndolo compatible con la mayoría de microprocesadores y microcontroladores existentes en el mercado.

- 1) *Aplicaciones típicas para el DAC TLV5620:* Fuentes programables de voltaje, amplificadores/atenuadores digitalmente controlados, reproductores de música digital, comunicaciones móviles, equipos de testeo automático, controladores de motores de corriente directa, síntesis de señales y procesos de monitoreo y control, entre otros.
- 2) *Condiciones de operación del DAC TLV5620:* En la tabla 2 se presentan las condiciones de alimentación y los rangos permitidos por el fabricante [5], los cuales deben ser tenidos en cuenta para el diseño de una aplicación con este dispositivo.

TABLA II
CONDICIONES DE OPERACIÓN DEL DAC TLV5620

Descripción	Valor
VDD, voltaje de alimentación	2.7 a 5.3 V
VIH, voltaje nivel alto de entrada	0.8*VDD V
VIL, voltaje nivel bajo de entrada	0.8 V
Vref, voltaje de referencia por canal	VDD-1.5 V
RL, resistencia de carga	10 kΩ
TW, tiempo mínimo de duración LOAD	250 ns
CLK, frecuencia de reloj	1 MHz
TA, temperatura de operación aire libre.	0 a 70 °C
IDD corriente de alimentación a Vdd=3.3 v	2 mA
Iref, corriente de referencia, Vdd=3.3v, Vref=1.5v	10 A

- 3) *Conexión típica del DAC TLV5620:* El control digital del DAC es realizado a través de tres conexiones mediante interfaz serial SPI. Una conexión de datos, una conexión de frecuencia reloj la cual debe ser 1 MHz y la conexión de LOAD que habilita o deshabilita la comunicación.

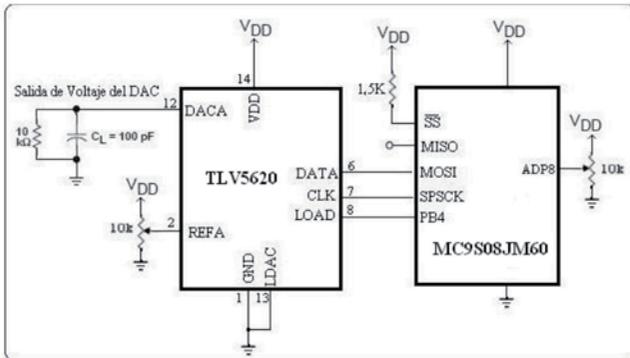


Figura 9. Diagrama de conexión entre el MC9S08JM60 y el DAC TLV5620

- 4) *Comunicación con el DAC TLV5620:* La puesta en marcha de este convertor consiste en enviarle desde un microcontrolador un código binario para que el DAC proporcione un nivel de voltaje.

Para establecer la comunicación es necesaria una palabra de 11 bits, donde 8 bits corresponden al dato binario y los siguientes 3 bits corresponden a los bits de control. Los bits de datos son los que determinan el nivel de voltaje que debe proporcionar el DAC y los bits de control son para seleccionar uno de los cuatro DACs internos para realizar la conversión y seleccionar el rango respectivo de la salida.

Cuando el microcontrolador envía un comando de 11 bits al DAC, el pin LOAD debe estar en nivel alto para que los bits puedan ser llevados al terminal DATA. Los bits ingresan con cada flanco descendente del reloj. Una vez todos los bits de datos han sido ingresados, es necesario colocar el pin LOAD en nivel bajo para transferir los datos desde el registro de entrada hasta el canal del DAC seleccionado (Fig. 10).

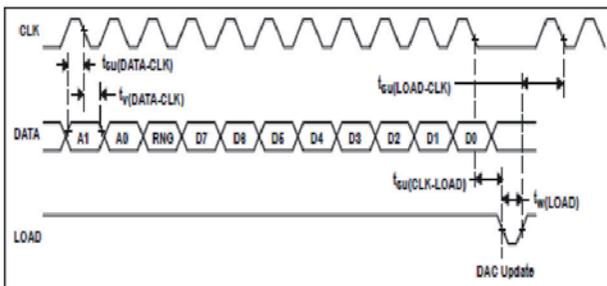


Figura 10. Actualización controlada por LOAD [5]

- 5) *Desarrollo del algoritmo en el microcontrolador:* Esta aplicación consiste en hacer uso de un potenciómetro para enviar un voltaje variable al convertor A/D del microcontrolador MC9S08JM60. El valor digital obtenido de la conversión se envía al DAC, el cual entrega a su salida su equivalente analógico que ingresa nuevamente a otro canal del convertor A/D. Lo anterior con el fin de comparar el voltaje variable contra la salida del DAC y visualizarlos en una pantalla LCD alfanumérica.

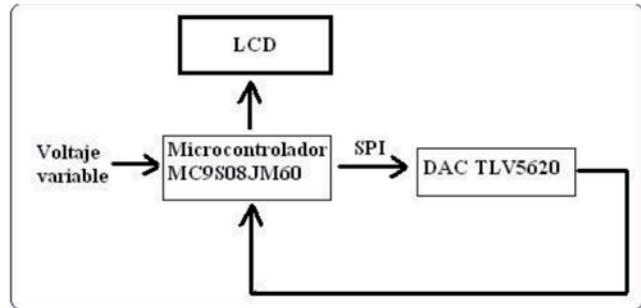


Figura 11. Aplicación realizada con el DAC TLV5620

El pseudocódigo del algoritmo desarrollado para la implementación del DAC se presenta a continuación:

Seudocódigo

Inicio

- Declaración de variables
- Inicialización canales ADC
- Inicialización módulo SPI como maestro
- Establecer la frecuencia a 1MHz
- Inicialización display LCD

Inicio Ciclo infinito

- Leer convertor ADC resistencia variable
- Mostrar valor de voltaje de resistencia variable
- Fijar en uno el pin de salida a LOAD del DAC
- Esperar que el registro de transmisión Tx este vacío
- Enviar serialmente bits de control del DAC
- Esperar que el registro de transmisión Tx este vacío
- Enviar serialmente bits de datos al DAC
- Fijar en cero el pin de salida a LOAD del DAC
- Leer convertor ADC valor salida del DAC
- Mostrar valor salida del DAC

Fin ciclo infinito

Fin

V. CONCLUSIONES

Implementar algoritmos para la comunicación serial síncrona entre dos o más dispositivos, abre la posibilidad de crear aplicaciones y desarrollar dispositivos electrónicos en diferentes campos, como el procesamiento de señales o sistemas de control.

Para el sensor digital de temperatura TMP124 —aunque no es recomendable para ambientes húmedos—, aplicaciones como protección térmica de equipos o monitoreo de temperatura ambiente se adaptan perfectamente, teniendo así una solución confiable y a bajo costo.

Son innumerables las aplicaciones en las que se puede utilizar un DAC a un bajo costo y un gran desempeño. Por su alto nivel de precisión se puede implementar en comunicaciones móviles, servomecanismos como controles de velocidad y posición, equipos de medición automático entre otros.

REFERENCIAS

- [1] MC9S08JM60 Series data sheet [en línea]. Disponible en: http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08JM60.pdf?fsrch=1&sr=1
- [2] 1.5°C Accurate programmable digital temp Sensor SPI Interface (Rev. B) [en línea]. Disponible en: <http://focus.ti.com/lit/ds/symlink/tmp124.pdf>.
- [3] TLV5620C, TLV5620I, quadruple 8-bit digital-to-analog converters, [en línea]. Disponible en: <http://focus.ti.com/lit/ds/symlink/tlv5620.pdf>
- [4] F. Pereira. HCS08 Unleashed designer's guide to the HCS08 microcontroller. Lexington: Booksurge, 2009. 413p. ISBN 978-1-4196-8592-7
- [5] SPI block guide V.3.06 [en línea]. Disponible en: <http://www.ee.nmt.edu/~teare/ee308I/datasheets/S12SPIV3.pdf>
- [6] G. Galeano. Programación de sistemas embebidos en C. Alfaomega, 2009. 544p. ISBN 978-958-682-770-6.