10º Congreso Nacional de Mecatrónica
Noviembre 3 y 4, 2011. Puerto Vallarta, Jalisco.

Desarrollo de Interface en FPGA para Protocolo SPI

Ávila Mauricio Saúl[1*], Héctor García Ruiz[1], Hernández Hernández Ricardo Francisco José[1], Gutiérrez Granados Cuitláhuac[1], Pedraza Ortega Jesús Carlos[2] y Ramos Arreguín Juan Manuel[2**]

[1] Universidad Tecnológica de San Juan del Río, Carrera de Mecatrónica
[2] Universidad Autónoma de Querétaro, Facultad de Informática

[*]zaxenedar_696@hotmail.com, [**]jramos@mecamex.net

Resumen

Los sistemas digitales requieren de comunicarse con diversos sistemas, como son: memorias, comunicación serial RS232 ó USB, convertidores analógico-digital y digital-analógico, o bien, sistemas como acelerómetros que permiten conocer cómo se está comportando un sistema (vibraciones). Las formas de comunicación son muy variadas, por ejemplo: paralelo, RS232 o Interfaz Periférica Serial (SPI).

Este trabajo muestra el diagrama descriptivo de bloques de la implementación de una comunicación SPI. Esto es aplicado a un acelerómetro, del cual se explica a continuación. En resultados, se muestran los diagramas de tiempo resultantes.

Palabras clave: FPGA, comunicación, SPI, bloques descriptivos de hardware.

1. Introducción

El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

Incluye una línea de reloj, dato entrante, dato saliente y un pin de chip select, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. De esta forma, este estándar permite multiplexar las líneas de reloj.

Los dispositivos conectados al bus son definidos como maestros y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control. Un esclavo es un dispositivo controlado por el maestro. Cada esclavo es controlado sobre el bus a través de una línea selectora llamada Chip Select o Select Slave, por lo tanto es esclavo es activado solo cuando esta línea es seleccionada. Generalmente una línea de selección es dedicada para cada esclavo. En un tiempo determinado T1, solo podrá existir un maestro sobre el bus. Cualquier dispositivo esclavo que no esté seleccionado, debe deshabilitarse (ponerlo en alta impedancia) a través de la línea selectora (chip select).

El bus SPI emplea un simple registro de desplazamiento para transmitir la información. Todas las líneas del bus transmiten la información en una sola dirección. La señal de reloj (SCLK) es generada por el maestro y se utiliza para sincronizar la transferencia de datos hacia los dispositivos esclavo. La línea MOSI (Master Out Slave In) transporta los datos del dispositivo maestro hacia los dispositivos esclavo. La línea MISO (Master In Slave Out) transporta los datos de alguno de los esclavos hacia el maestro. Esto se puede apreciar en la figura 1.

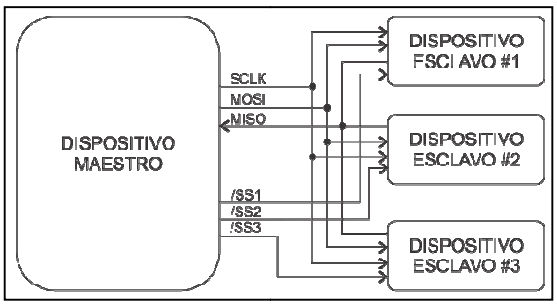

Fig. 1. Concepto de la comunicación SPI.

La transferencia de los datos, es sincronizada por la línea de reloj de este bus. Un bit es transferido por cada ciclo de reloj.

La mayoría de las interfaces SPI tienen 2 bits de configuración, llamados CPOL (Clock Polarity = Polaridad de Reloj) y CPHA (Clock Phase = Reloj de Fase). CPOL determina si el estado Idle de la línea de reloj esta en bajo (CPOL=0) o si se encuentra en un estado alto (CPOL=1). CPHA determina en que filo de reloj los datos son desplazados hacia dentro o hacia fuera. (Si CPHA=0 los datos sobre la línea MOSI son detectados cada filo de bajada y los datos sobre la línea MISO son detectados cada filo de subida).

Cada BIT tiene 2 estados, lo cual permite 4 diferentes combinaciones, las cuales son incompatibles una de la otra. Por lo que si dos dispositivos SPI desean comunicarse entre sí, estos deben tener el mismo la misma Polaridad de Reloj (CPOL) y la misma Fase de Reloj (CPHA).

Existen cuatro modos de reloj definidos por el protocolo SPI, llamados A, B, C y D, como se muestra en la figura 2.

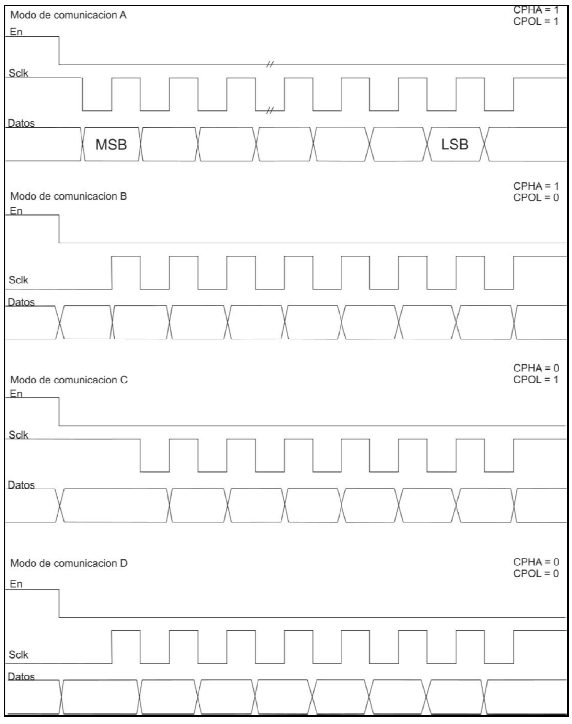

Fig. 2. Modos de la comunicación SPI.

2. Acelerómetros.

La medición de aceleración puede realizarse midiendo la fuerza necesaria para acelerar un objeto de masa conocida, para lo cual a su vez bastará medir la deflexión de un dinamómetro que sostiene a dicha masa. Es éste el principio físico en el que se basa el funcionamiento de la serie ADXL. El modelo mecánico correspondiente a esta idea se muestra en la figura 3 [1] [2].

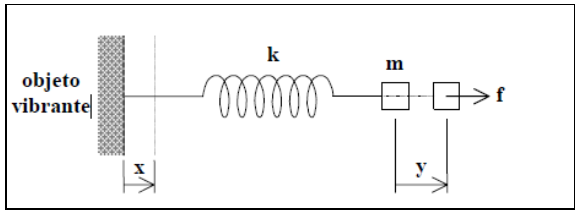

Fig. 3. Modelo de funcionamiento de un acelerómetro.

La disposición muestra un objeto vibrante cuya aceleración se desea medir. Sobre el mismo está montado el dispositivo de medición formado por una masa m y un resorte con constante elástica k. Lo que se desea medir es la derivada segunda del desplazamiento del objeto vibrante, es decir,

$$a(t) = x(t) \qquad (1)$$

Para ello se mide, en realidad, la deformación x(t) − y(t) del resorte, ya que no es sencillo medir directamente la aceleración respecto a un sistema inercial.

Se busca, por lo tanto, una relación entre esta deformación y la aceleración a(t). Llamando f a la fuerza aplicada por el resorte a la masa m, tenemos:

$$f = k(x - y) = my \qquad (2)$$

De donde se obtiene una ecuación diferencial que relaciona x e y:

$$my + k\,y = k\,x \qquad (3)$$

Aplicando la transformación de Laplace (suponiendo condiciones iniciales nulas), se obtiene:

$$Y(s) = \frac{X(s)}{1+\frac{m}{k}s^2} = \frac{X(s)}{1+\left(\frac{s}{\omega_0}\right)^2} \qquad (4)$$

Siendo ωo = k / m la frecuencia natural o frecuencia de resonancia del sistema. De aquí podemos obtener X - Y:

$$X(s) - Y(s) = \frac{X(s)\left(\frac{s}{\omega_0}\right)^2}{1+\left(\frac{s}{\omega_0}\right)^2} = \frac{\frac{1}{\omega_0{}^2}}{1+\left(\frac{s}{\omega_0}\right)^2} A(s) \qquad (5)$$

Donde A(s) es la transformada de la aceleración buscada. Esta ecuación muestra que muy por debajo de la resonancia, es decir para ω<< ωo, la deformación del resorte es aproximadamente proporcional a la aceleración, es decir:

$$x(t) - y(t) \cong \frac{1}{\omega_0{}^2} a(t) \qquad (6)$$

3. Acelerómetro ADXL345.

Para comprender lo que registra el acelerómetro, debemos primero ver cómo es que este nos entrega el dispositivo como resultado de su medición, esto nos obliga a revisar la tabla 1 de registros, donde se marca la forma de operación y como modificar estos resultados hasta que la serie de datos nos satisfaga por completo [3] [4].

Tabla 1. Mapa de registros del acelerómetro.

Registro	Descripción
0x00	ID del dispositivo.
0x01 a 0x01C	Reservado. No accesar.
0x1D	Umbral de movimiento.
0x1E	Ajuste del eje X.
0x1F	Ajuste del eje Y.
0x20	Ajuste del eje Z.
0x21	Duración del movimiento.
0x22	Retardo de la movimiento
0x23	Ventana de movimiento.
0x24	Umbral de actividad.
0x25	Umbral de inactividad.
0x26	Tiempo de inactividad.
0x27	Habilitación del control de actividad inactividad de los ejes.
0x28	Umbral de caída libre.
0x29	Tiempo de la caída libre.
0x2A	Control de los ejes por movimiento/doble movimiento.
0x2B	Fuente del movimiento.

Tabla 1 (continuación)

Registro	Descripción
0x2C	Rango de datos y control de modo de operación.
0x2D	Control de modo de consumo de operación.
0x2E	Control de habilitación de interrupción
0x2F	Control del mapeo de interrupciones
0x30	Fuente de interrupciones.
0x31	Control del formato de datos.
0x32	Dato 0 del eje X.
0x33	Dato 1 del eje X.
0x34	Dato 0 del eje Y.
0x35	Dato 1 del eje Y.
0x36	Dato 0 del eje Z.
0x37	Dato 1 del eje Z.
0x38	Control del FIFO.
0x39	Estado del FIFO.

Los registros que se configuraran directamente son el 0x27, el 0x31, 0x2D, para la forma de operación y en este modo o dejando al último el 0x2D, debido a que este último es el que hace que el dispositivo empiece a hacer la adquisición de datos.

El registro 0x27 se toma para la configuración puesto que en él se maneja el modo de operación en base a tipo de corriente y forma de detección (en activo o inactivo) de cada eje y la activación de cada eje bajo el tipo de detección. El registro 0x31 se debe configurar por la necesidad de establecer el tipo de comunicación del SPI (D6) si es a 4 hilos o solo a tres, además de declarar el tipo de resolución que se debe de tomar si es modo full o si se tomara la resolución de 10 bits (D3), también el formato de envió del MSB si es izquierda o derecha médiate el manejo del bit D2 y el rango de operación para la adquisición de datos (D1 y D0). Y el registro 0x2D se configura para el inicio de operación de medición (D3) y el trabajo en formas de modo link (D5) que sirve para el modo de detección doble (activo o inactivo); es decir, el bit al estar establecido trabajara ambas detecciones generando un retardo a la función de actividad mientras la función de inactividad trabaja. Después de que la detección de actividad la detección por inactividad inicia atando pendiente de la detección por actividad. Si el bit esta en 0 las actividades se toman de forma normal.

Los registros que serán configurados conforme a las pruebas son del 0x1D a 0x26 y de 0x28 a 0x2C que serán poco a poco configurados a consideración de la forma en que el dispositivo entrega los datos conforme a su operación o en otras palabras se configuran conforme los datos se trabajen.

4. DESARROLLO EN FPGA

Ahora después de dejar en claro que es lo que se debe de trabajar ahora queda el problema como hacer para que la serie de datos que arroja el acelerómetro traducirlos y también hacer que nos de los datos en la forma en que queremos, esto nos deja en generar un diseño para que el FPGA acceda al dispositivo y haga las operaciones necesarias por lo cual se propone la siguiente forma de diseño.

Los Bloques Descriptivos de Hardware [5] permiten planear la implementación de un sistema digital. La figura 4 muestra la entidad principal, que es el módulo de mayor jerarquía.

En la figura 4 se puede apreciar el diseño que nos permitirá la comunicación con el dispositivo, donde se nota los diferentes módulos que se requieren para establecer la comunicación SPI, tomando en cuenta el protocolo de datos del esclavo. A continuación se hace una breve descripción de cada uno de los módulos involucrados.

En la figura 5 se muestra el diseño del componente Divisor_M que genera un pulso de reloj cada M ciclos de reloj. En nuestro caso, cada ciclo de reloj tiene un periodo de 20 ns. A este módulo también se le llama divisor de frecuencia, pues a partir de una frecuencia maestra, permite obtener un pulso cada cierto periodo. El pulso generado tiene una duración de un ciclo de reloj.

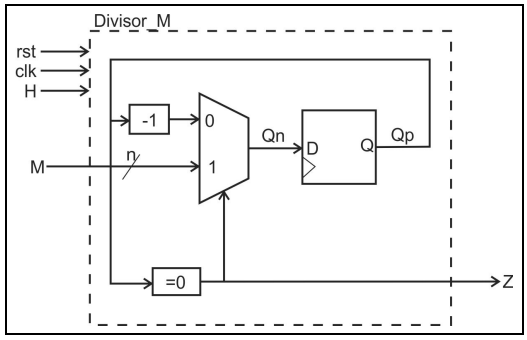

Fig. 5. Módulo divisor de frecuencia.

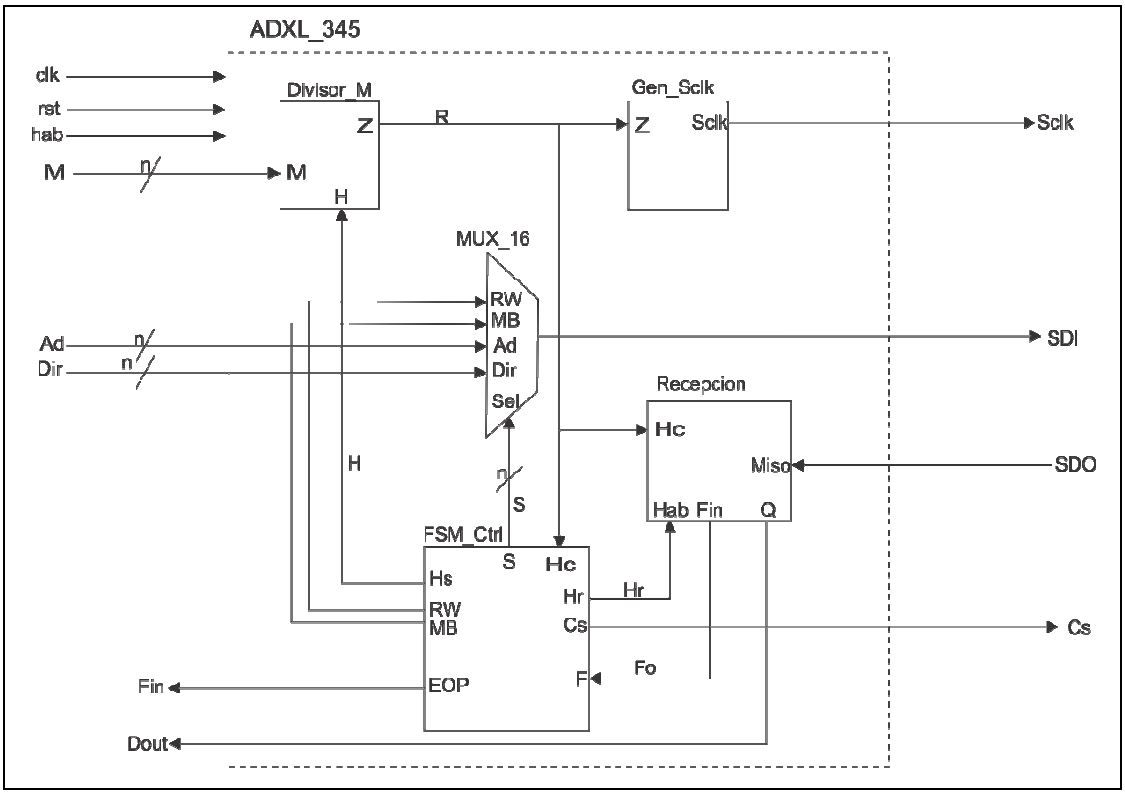

Fig. 4. Entidad principal de la comunicación SPI.

El módulo GEN_SCLK genera la señal de reloj serial o SCLK. Al recibir la señal Z del divisor de frecuencia, la máquina cambia su estado de 0 a 1 lógico, y viceversa. La figura 6 muestra esta máquina secuencial.

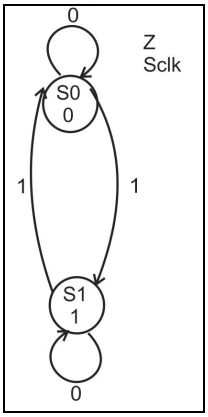

Fig. 6. Máquina secuencial generadora de SCLK.

La información enviada por un dispositivo SPI es recibida por el módulo maestro, en este caso el FPGA, a través de la línea MISO. Sin embargo, debido a que esta información es recibida en forma serial, es necesario el desarrollo de un módulo que reciba datos en forma serie y los entregue en formato paralelo y puedan ser procesados en módulos consecuentes. El módulo de la figura 7 es un registro serie-paralelo, en el que destacan las siguientes señales:

- HAB : Señal de habilitación. Con '1' lógico el módulo es habilitado para recibir datos seriales; y con un '0' se deshabilita.
- MISO : Entrada de datos seriales provenientes del acelerómetro (esclavo).
- HC : Actualización del registro. Se actualiza cuando la señal vale '1'.
- FIN : Indica que un dato ya está listo para ser leído. Cuando vale '1' lógico, se puede leer el dato. Cuando vale '0', el módulo está recibiendo datos y no debe ser leído aún.
- Q : Es el registro de salida de n bits. En nuestro caso, los registros son de 16 bit.

El módulo de multiplexor de la figura 4 es utilizado para el envío serial de la información, y poder configurar el acelerómetro (esclavo).

En la figura 4, el módulo FSM_Ctrl es el responsable de sincronizar al resto de los módulos mostrados en la figura. Las señales utilizadas en este módulo se muestran en la figura 8, así como una breve descripción de la utilidad de cada una son mostradas a continuación.

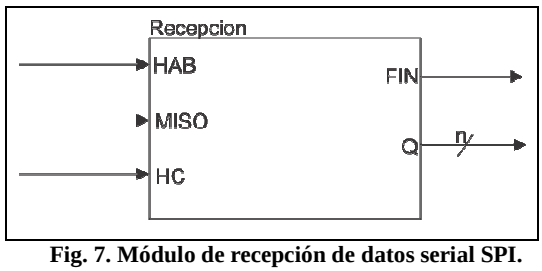

Fig. 7. Módulo de recepción de datos serial SPI.

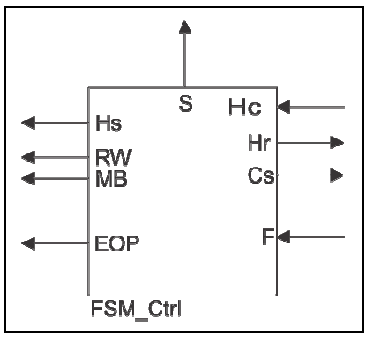

Fig. 8. Señales utilizadas por el módulo de control principal.

- Hc : Es la base de tiempo que viene del divisor de frecuencia. Cada que vale '1', el módulo es habilitado para poder cambiar de estado si ello es necesario.
- Hs : Habilita al módulo divisor de frecuencia para que inicie la generación de la base de tiempo del SCLK.
- Hr : Habilita al módulo de recepción para cambiar de estado. Esta habilitación dura solamente un ciclo de reloj.
- RW : Indica si la operación a realizar es de lectura o escritura. Un '1' indica lectura y el '0' escritura.
- MB : Indica el modo de funcionamiento de la comunicación con el esclavo.
- EOP : Fin de la comunicación. Si vale '1', módulo está libre y puede ser utilizado. Si es '0', se encuentra ocupado comunicándose con un esclavo.
- Cs : Habilita al dispositivo esclavo para que obedezca la información del bus SPI.
- F : Recibe el estado del módulo de recepción para saber en qué momento se tiene una recepción de datos completa.

La arquitectura interna del módulo de la figura 7 es una máquina secuencial de estados finitos, y se muestra en la figura 9.

Una vez definido el diagrama a bloques descriptivo de hardware, se lleva a cabo la implementación de la descripción.

5. Implementación del SPI.

La implementación es realizada en una tarjeta Nexys 2, ya que es un entorno de desarrollo mínimo para los ambientes FPGA y se ajusta a las necesidades de trabajo para la comunicación. La figura 10 muestra el diagrama de la tarjeta mencionada.

Fig. 10. Sistema de desarrollo Nexys 2.

Las características principales de la tarjeta se muestran en la figura 11. Para la implementación de las interfaces diseñadas, se tomaron en cuenta los conectores de expansión (Pmod x4), de los cuales la tarjeta cuenta con 4.

La implementación de la comunicación SPI se realizó de manera exitosa, y la figura 12 muestra una gráfica del resultado obtenido.

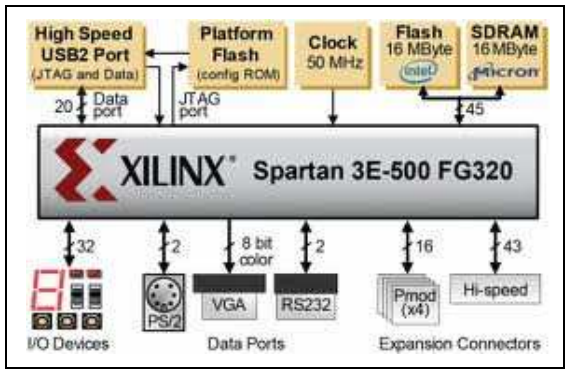

Fig. 11. Características de la tarjeta Nexys 2.

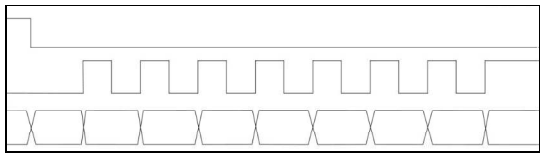

Fig. 12. Resultado de la simulación SPI.

6. Resultados.

Lo anterior nos hace ver que el desarrollo en FPGA es una opción en la generación de dispositivos a medida, puesto que al ser un circuito a describir nos da la libertad de hacer en él un dispositivo que cubra satisfactoriamente todas las necesidades que el usuario requiera, y que por su gran capacidad de almacenamiento, puede aceptar en cierta medida varias funciones o sistemas, evitando así el hacer en diferentes descripciones y en diferentes PLD´s las mismas funciones que requerimos.

Ahora bien, el desarrollo de protocolos de comunicación SPI es en gran parte una ventaja con base que, sí el usuario así lo requiriera, puede trabajar con un intervalo de datos a la medida que su aplicación lo requiera, sin estar forzado a limitarse a las normalizaciones de otros protocolos similares, y que aparte por ser una comunicación sincronizada, el error que llegase a encontrar seria porque el dispositivo este dañado.

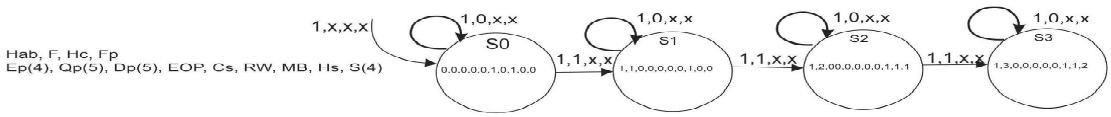

Fig. 9. Arquitectura del módulo de control.

Otra de las cosas que podemos apreciar es que el uso de transductores modificables, es un gran impulso a la instrumentación, debido a que se puede adecuar el dispositivo de medición conforme el operador del sistema lo necesite, modificando su forma de operación, de adquisición, entrega de datos, etc.

7. Conclusiones

Este trabajo permite establecer la manera como se puede implementar en hardware el algoritmo para comunicación entre dispositivos que utilizan el protocolo SPI. Sin embargo, cada dispositivo tiene diferentes maneras de funcionar, por lo que es necesario adaptar los diagramas a bloques aquí mostrados, de tal manera que se tenga una correcta comunicación.

La comunicación lograda cumplió con las expectativas establecidas al principio de este trabajo, logrando una comunicación exitosa con el dispositivo.

La continuación de este trabajo, va enfocada a interpretar correctamente la información obtenida, aplicada a un sistema mecatrónico real, donde se pueda medir posición y vibraciones utilizando la información proveniente del acelerómetro de 3 ejes, tanto de rotación como traslación.

Referencias

[1] Analog Devices. “Using Accelerometers in Low-g Applications” Hoja de aplicaciones AN-374.

[2] Analog Devices. “Increasing the Frequency Response of the ADXL Series Accelerometers”. Hoja de aplicaciones AN-377.

[3] Analog Devices. “Hoja de datos ADXL345”

[4] Ingeniería en microcontroladores,” Protocolo SPI”, Distrito Federal, México

[5] Romero R. *“Electrónica Digital y lógica programable”*, Universidad De Guanajuato, México, primera edición año 2007.