

# Lector de Señales en Cuadratura AquadB con Interfaz Serial Utilizando FPGA

Octavio Icasio Hernández

Centro Nacional de Metrología, División de Metrología Dimensional, Km. 4.5 Carretera a los Cués.  
El Marques, Querétaro. C.P. 76246, Apdo. postal 1-100 Centro. oicasio@cenam.mx

## Resumen

El presente trabajo muestra el diseño y la implementación en FPGA de un lector de señales en cuadratura AquadB generadas por un gran número de sistemas de medición lineales y angulares, que llevan integrado un encoder de tipo incremental. La lectura del sistema de medición es transmitida por interfaz serial a un software diseñado para mostrar las lecturas lineales o angulares. El código VHDL del lector generado con el presente trabajo estará a disposición de quien desee emplearlo y en el caso del CENAM será aplicado en la lectura de las escalas de Máquinas de Medición por Coordenadas y en la construcción de un interferómetro heterodino.

**Palabras clave:** AquadB, encoder, incremental.

## I. Introducción

La tecnología de la automatización está constantemente tratando de mejorar la eficiencia de los procesos de producción y de pruebas en todas las ramas de la industria. Más que antes, los movimientos mecánicos lineales, mesas rotatorias, brazos robóticos, actuadores, etc. son controlados numéricamente. Encoders lineales y angulares pueden medir cualquier tipo de movimiento mecánico y devolver el valor medido a una unidad de control. Las capacidades de estos encoders continúan mejorándose en términos de su resolución, exactitud y confiabilidad, intervalo de medición y mayor velocidad para la adquisición de las señales.

Existen varios tipos de *encoders*, uno de ellos es el denominado *encoder* incremental [1], estos encoders generan las señales AquadB que se pretenden leer a través de un FPGA para posteriormente enviar la lectura por interfaz serial a una computadora (PC).

El presente trabajo emula directamente lo realizado por el circuito integrado CMOS HCTL-20XX de Agilent [2], agregando como características adicionales un reloj de 50 MHz, Interfaz serial, modo de conteo fijo a 4X, soporte para 3 ejes, alta inmunidad al ruido, contador de 32 bits, etc., haciendo notar que al ser

programadas en un FPGA todas estas características son reconfigurables.

## II. Funcionamiento del encoder incremental

El *encoder* es un transductor rotativo o lineal que transforma un movimiento angular o lineal en una serie de pulsos digitales. El *encoder* se basa en la rotación o desplazamiento de un disco o una regla graduada con una retícula radial o lineal formada por espacios oscuros, alternados con espacios claros. Un sistema óptico de emisor receptor infrarrojo detecta el cambio en la superficie del disco o de la regla, generando dos señales en cuadratura (desfasadas  $90^\circ$ ), las señales se identifican como A y B, figura 1.

El *encoder*, como su nombre lo indica, es un dispositivo que codifica información del desplazamiento y su dirección, normalmente el mínimo desplazamiento es decodificado a partir de un ciclo completo de la señal A o B (modo de conteo 1X). Observando detalladamente la señal en cuadratura se puede apreciar que hay información del desplazamiento en cada flanco de las señales A y B, por lo que es posible decodificar la información del desplazamiento y dirección, al doble (2X) y cuádruple (4X) de la señal originalmente decodificada.

En las señales A y B en cuadratura se encuentra codificada la información correspondiente al avance y su dirección, la cual puede ser en el sentido de las manecillas del reloj (*Clockwise*, CW) o en sentido contrario (*Counterclockwise*, CCW).

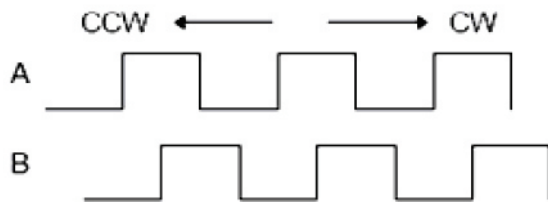
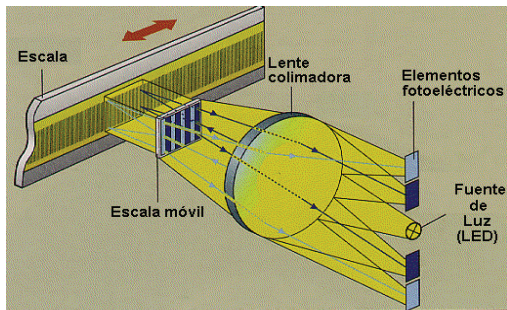


Fig. 1. Encoder incremental y señales en cuadratura. Cortesía [1].

La información correspondiente al desplazamiento se obtiene directamente de A o B, un ciclo de la señal corresponde al mínimo avance, se puede usar como referencia el flanco de subida o bajada; para un *encoder* de 600 pulsos por revolución el mínimo avance corresponde a  $360^\circ/600=0.6^\circ$ . Para determinar la dirección del desplazamiento se requiere de ambas señales; en la Fig. 2 se tiene un circuito a través del cual se determina el sentido del desplazamiento [8].

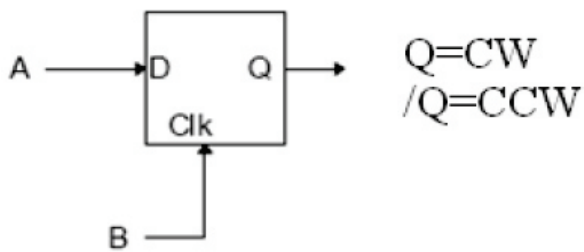


Fig. 2. Determinación del sentido de movimiento.

Se implementa con un *Flip-Flop* "D", la señal A se emplea como dato y B como señal de reloj, en el sentido CW (izquierda a derecha) se captura continuamente un nivel alto, esto, porque el flanco de subida de B coincide con el nivel alto de A. Para el sentido CCW (derecha a izquierda) el flanco de subida de B coincide ahora con el nivel bajo de A.

La mayor ventaja ofrecida por un encoder incremental es su alta resolución. Se pueden producir encoders incrementales del tipo rotatorios que producen 10000 pulsos por revolución a velocidades que excedan las 5000 RPM, o bien encoder incrementales con escalas graduadas cuyos periodos pueden ir hasta más de 0.0001 mm.

Su alta resolución, extrema durabilidad y fácil instalación hacen de los encoders incrementales en cuadratura ideales para una gran variedad de aplicaciones, entre otras:

Sistemas para controlar los drivers de un motor usan encoders en cuadratura para proveer retroalimentación en ciclo cerrado para controlar aplicaciones de posición y velocidad.

Impresoras, máquinas de fax y máquinas copiadoras usan encoders en cuadratura para sincronizar varias partes en movimiento y que no existan problemas en su operación.

Los elevadores usan encoders en cuadratura para mantener velocidad y aceleración y para corregir el alineamiento de la puerta.

Analizadores de sangre automatizados usan estos encoders para asegurar la exacta posición de frascos conteniendo muestras de sangre mientras desarrollan pruebas automatizadas.

### III. Módulos del sistema de lectura

El presente trabajo es una emulación en FPGA del circuito integrado CMOS HCTL-20XX de Agilent con características tales como un reloj de 50 MHz, Interfaz serial, modo de conteo fijo a 4X, soporte para 3 ejes, alta inmunidad al ruido, contador de 32 bits, etc. Para un detalle más específico de estas características basta observar

el diseño expuesto en la hoja de datos de este dispositivo [2], y ver los módulos que integran dicho dispositivo.

La figura 3 muestra las entradas y salidas del módulo principal diseñado para este trabajo.

### Descripción de entradas/salidas

**Exin(1:0), Eyin(1:0), Ezin(1:0):** Entradas AquadB para los ejes X, Y, Z. Son las entradas que aceptan las señales en cuadratura provenientes de un encoder incremental. Estas señales son del tipo mostrado en la figura 1 con un desfase de 90 grados.

**Clk:** reloj del sistema

**Pide\_cuenta:** Señal para activar la aceptación de las señales. Representa el botón de inicio de todo el sistema

**RST:** Reset asíncrono para los módulos “Salida Pic” y “Rs232RefComp”

**rstx, rsty, rstz:** resets síncronos del módulo “Contador” y del módulo “Bufér”. Estos comandos resetean los contadores de los ejes.

**RXD:** pin para la recepción serial

**DOUT:** datos que reciben el comando binario serial para poder mostrarlo en leds.

**TXD:** pin para transmisión serial.

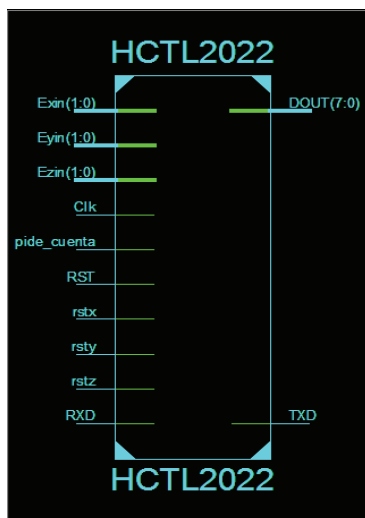


Fig. 3. Módulo principal en FPGA.

A partir del módulo principal se pueden extraer los módulos que lo conforman y estos módulos son los que se muestran en la figura 4.

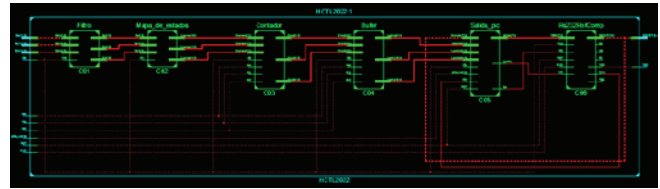


Fig. 4. Diagrama a bloques para lector de señales en cuadratura.

Se puede percibir que el diagrama no es muy claro sin embargo lo que sí es claro es que son en total 6 módulos los que conforman al módulo principal de la figura 3. Para ello a continuación se describirán cada uno de estos módulos de izquierda a derecha.

### Módulo “Filtro”



Fig. 5. Módulo “Filtro”.

El módulo “Filtro” es un filtro digital responsable de rechazar el ruido proveniente de las señales en cuadratura. Esto se logra gracias a la combinación de entradas con disparo tipo Schmitt trigger para rechazar niveles de voltaje inferiores a 1v y 3 ciclos de retardo. Esta combinación rechaza ruidos de bajo y largo nivel, así como picos de corta duración generados muy frecuentemente en aplicaciones que tienen involucrado un motor.

Las señales en cada canal son muestreadas en flancos de subida del reloj. Una historia temporal de las señales es almacenada en un registro de

desplazamiento de 4 bits. Cualquier cambio en la entrada es comprobado por un nivel estable en los 3 subsiguientes flancos de subida del reloj. Por lo tanto las formas de onda de las salidas filtradas pueden cambiar solo después de que un nivel de entrada tiene el mismo valor durante 3 consecutivos flancos de subida de reloj. Con este método los picos de duración corta entre flancos de subida son ignorados y los pulsos que son más cortos que dos periodos de reloj son rechazados.

Para este módulo Exin, Eyin y Ezin son las señales de entrada en cuadratura para cada eje y Ex, Ey, Ez son las señales de salida ya filtradas.

### Módulo “Mapa de Estados”

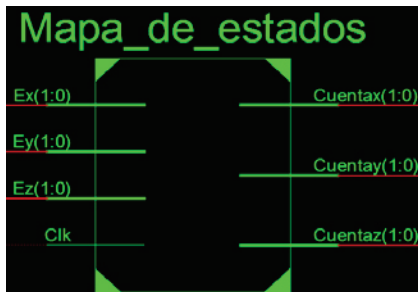


Fig. 6. Módulo “Mapa de estados”.

Este módulo permite la decodificación de las señales en cuadratura A y B ya filtradas (Ex, Ey, Ez) por el módulo “Filtro” en información para conteo. Este módulo podría multiplicar la resolución de las señales de entrada por un factor de uno, dos o cuatro (Decodificación 1x, 2x o 4x), ver [8]. Para este trabajo la multiplicación se realizó en el modo 4x, con lo que se alcanza la mayor resolución con los 4 estados representados por las 2 señales A y B, ver figura 7. Este módulo muestrea las salidas filtradas de las señales A y B. Basado en el estado binario pasado de las 2 señales, así como en el estado presente, este módulo entrega una señal de conteo llamada Cuentax, Cuentay y Cuentaz de 2 bits cada una, donde si el valor entregado es 01 el contador se incrementa, si es 00 el contador se decrementa y si es 11 el contador permanece fijo.

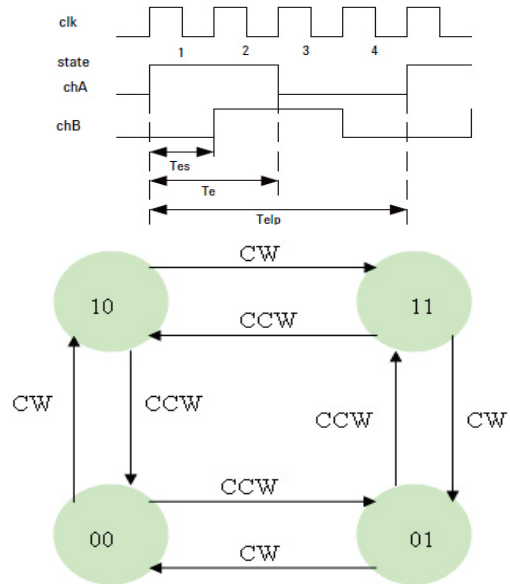


Fig. 7. Diagrama de estados.

Revisando los valores de izquierda a derecha que provienen de las señales chA y chB, se tienen los siguientes estados: A=1 B=0, A=1 B=1 A=0 B=1, A=0 B=0, en este caso son cuatro estados (estos estados se pueden ver si se traza una línea recta sobre los números 1, 2, 3 y 4 de la señal de reloj clk de la figura 7). El diagrama de estados de la figura 7 representa los estados y transiciones para el caso CW (sentido de las manecillas del reloj) y CCW (Sentido contrario a las manecillas del reloj).

Del diagrama de estados se obtiene la tabla 1, en esta se tienen los estados presentes y siguientes para A y B, de los diez y seis estados posibles únicamente se emplean ocho, los restantes se pueden descartar plenamente dado que por la naturaleza del sistema nunca serán presentes. Considerando CW=0 y CCW=1 y estados no ocupados=X (no importa).

Estado presente		Estado siguiente		DIR
B	A	b	a	
0	0	0	0	x
0	0	0	1	CCW
0	0	1	0	CW
0	0	1	1	x
0	1	0	0	CW
0	1	0	1	x
0	1	1	0	x
0	1	1	1	CCW
1	0	0	0	CCW
1	0	0	1	x
1	0	1	0	x
1	0	1	1	CW
1	1	0	0	x
1	1	0	1	CW
1	1	1	0	CCW
1	1	1	1	x

Tabla 1. Tabla de verdad para el modo 4x.

**Módulo “Contador”**

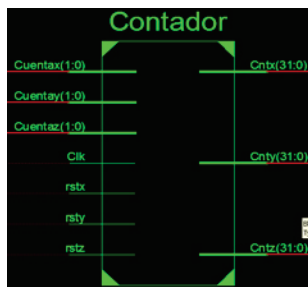


Fig. 8. Módulo “Contador”.

Este módulo representa a un contador binario ascendente/descendente de 32 bits, el cual cuenta flancos de subida como se describió en el módulo anterior. Este contador puede ser reseteado y comenzar en 0 para cualquiera de sus 3 ejes y cada reset es independiente para cada eje. Las cuentas generadas son luego pasadas al módulo “Bufere” para ser guardadas hasta que sean ocupadas. El contador se desborda hasta llegar a una cuenta máxima de  $2^{32}$  momento en el cual el contador se vuelve a hacer 0. Con este número de cuentas si tenemos una señal con periodo igual a por ejemplo 0.0001 mm tendríamos un alcance con este contador de 429496.7296 mm o 429.4967296 m, lo cual es un intervalo de medición bastante considerable y no se necesita conectar en cascada con otro contador, a menos que así se requiera se puede reconfigurar el

contador para poder enlazar en cascada uno o más contadores.

**Módulo “Bufere”**

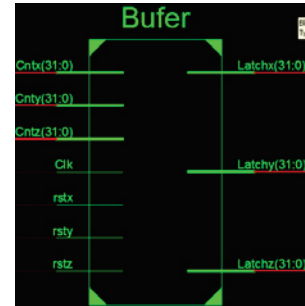


Fig. 9. Módulo “Bufere”.

El módulo “Bufere” es un candado de 32 bits el cual captura la posición otorgada por el módulo “Contador” en cada flanco de subida del reloj y cada eje puede ser reseteado en cualquier momento.

**Módulo “Salida Pic”**



Fig. 10. Módulo “Salida Pic”.

Este módulo es el enlace con el módulo que controla la interfaz serial (módulo “Rs232RefComp”), a través de este módulo el usuario recibe por la “Entrada” de 7 bits el comando para saber las cuentas de que eje mandar y en paquetes de 8 bits. En otras palabras este módulo funciona como un multiplexor de 32 a 8 líneas y el comando recibido en modo paralelo permite el acceso independiente a la parte alta, media o baja de la posición retenida en los candados del módulo anterior (Latchx, Latchy y Latchz) y además permite definir a cual eje acceder. También cuenta con la entrada

“pide\_cuenta” la cual funciona como botón de arranque para poder mandar las cuentas de manera paralela a través de “Salida”, al módulo que tiene la interfaz serial. Además tiene una entrada llamada “RD”, la cual determina si este módulo ha recibido o no datos por la “Entrada” a través del módulo que controla la interfaz serial, si RD es igual a 1 significa que el módulo que controla le interfaz serial a recibido datos por el puerto serial y el módulo “Salida Pic” ha recibido el comando para mandar los datos requeridos. Para poder lograr mandar los datos requeridos este módulo usa dos salidas llamadas “escribir” y “leer” a través de las cuales le comunica al módulo “Rs232RefComp” si se encuentra leyendo un comando o si requiere enviar los datos requeridos. Cuando “escribe” recibe un valor de 1 significa que el módulo “Salida Pic” le está pidiendo al módulo “Rs232RefComp” que requiere enviar los datos solicitados de acuerdo al comando recibido. La figura 12 muestra el código VHDL con los comandos aceptados por este módulo.

Este módulo fue implementado con una máquina de estados cuya gráfica queda expresada en la figura 11.

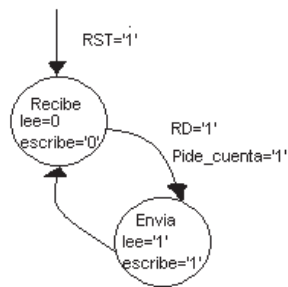


Fig. 11. Máquina de estados para enviar y recibir.

El módulo por default siempre está en el estado recibe, una vez que RD cambia a 1 (estado cambiado por el módulo “Rs232RefComp”) el comando es interpretado por el código abajo y se le pide al módulo “Rs232RefComp” que envíe los datos requeridos.

```

case stCur is
  when stReceive =>
    lee <= '0';
    escribe <= '0';
    if (pide_cuenta='1' and RD='1') then
      case Entrada is
        when "00110000" => Salida<=Latchx(7 downto 0);
        when "00110001" => Salida<=Latchx(15 downto 8);
        when "00110010" => Salida<=Latchx(23 downto 16);
        when "00110011" => Salida<=Latchx(31 downto 24);
        when "00110100" => Salida<=Latchy(7 downto 0);
        when "00110101" => Salida<=Latchy(15 downto 8);
        when "00110110" => Salida<=Latchy(23 downto 16);
        when "00110111" => Salida<=Latchy(31 downto 24);
        when "00111000" => Salida<=Latchz(7 downto 0);
        when "00111001" => Salida<=Latchz(15 downto 8);
        when "00111010" => Salida<=Latchz(23 downto 16);
        when "00111011" => Salida<=Latchz(31 downto 24);
        when others => Salida<="00110000";
      end case;
      stNext <= stSend;
    else
      stNext <= stReceive;
    end if;
  when stSend =>
    lee <= '1';
    escribe <= '1';
    stNext <= stReceive;
  end case;
end case;

```

Fig. 12. Código VHDL que recibe los comandos.

Los comandos recibidos son los caracteres 0 al 9 y los caracteres “.” y “,” (representación binaria el código ASCII de cada carácter). Si el carácter recibido es un “0” el módulo “Salida Pic” enviará el byte menos significativo del candado del contador de 32 bits para el eje X, si es un 3, enviara el byte más significativo del contador de 32 bits para el eje X y así respectivamente para cada eje y byte solicitado.

### Módulo “Rs232RefComp”

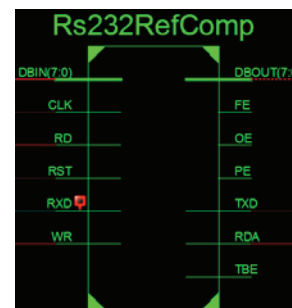
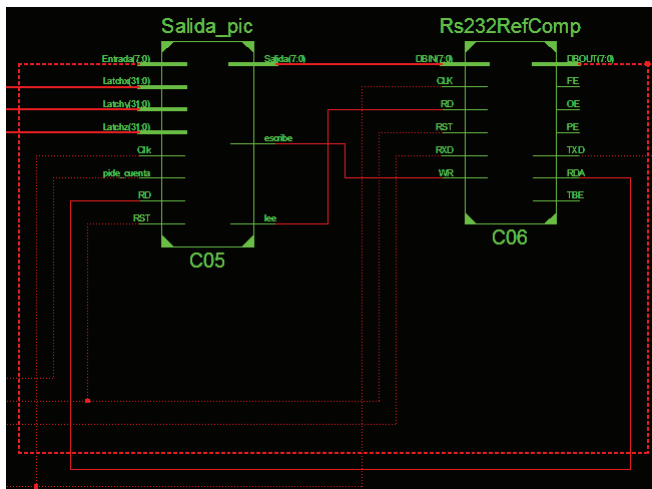


Fig. 13. Módulo “Rs232RefComp”.

Este módulo es el encargado de establecer la comunicación serial con la PC. Básicamente es un convertor serial a paralelo para los datos que recibe por el puerto serie, y es un convertor paralelo a serial para los datos que se deseen

enviar por el puerto serie. Este módulo activa la entrada/salida RDA cuando ha recibido algún dato por el puerto serie a través de su entrada RXD, esta señal es enviada al módulo “Salida Pic” por su entrada RD para decirle que hay datos en el puerto serie y que estos le serán enviados a su entrada “Entrada” a través de la salida DBOUT del módulo “Rs232RefComp”. Cuando el módulo “Salida Pic” recibe el comando RD=’1’ este inmediatamente le responde con los datos requeridos mandando un ‘1’ a sus salidas “lee” y “escribe”, las cuales están conectadas a las entradas RD y WR del módulo “Rs232RefComp”, estas entradas permiten que los datos recibidos por la entrada DBIN en el módulo “Rs232RefComp” que está conectada a la salida “Salida” del módulo “Salida Pic” sean enviadas al puerto serie mediante la salida TXD del módulo “Rs232RefComp” y así la comunicación serial con la PC queda establecida.



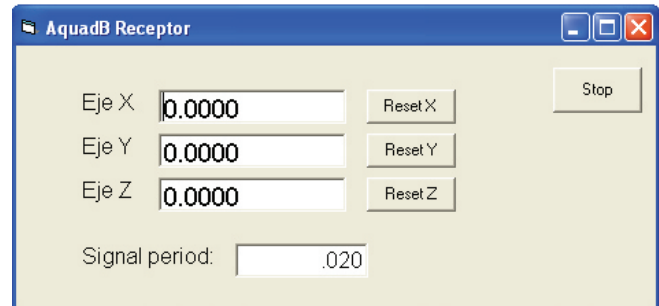
**Fig. 14. Interconexión entre módulos “Rs232RefComp” y “Salida Pic”.**

Para una mejor referencia y uso adecuado del módulo “Rs232RefComp” ver [4] y [5].

#### IV. Interfaz con PC

Finalmente para lograr la comunicación con la PC fue necesario crear una interfaz la cual se muestra en la figura 15. Esta interfaz permite leer hasta 3 ejes y resetear a cada eje de manera

independiente, además permite establecer en unidades del sistema internacional o sistema ingles el periodo de la señal de entrada, para traducir la cuentas otorgas por el contador en una posición real bajo un sistema de unidades determinado. La configuración serial para comunicación fue de 9600 baudios, 8 bits, sin paridad y 1 bit de parada.



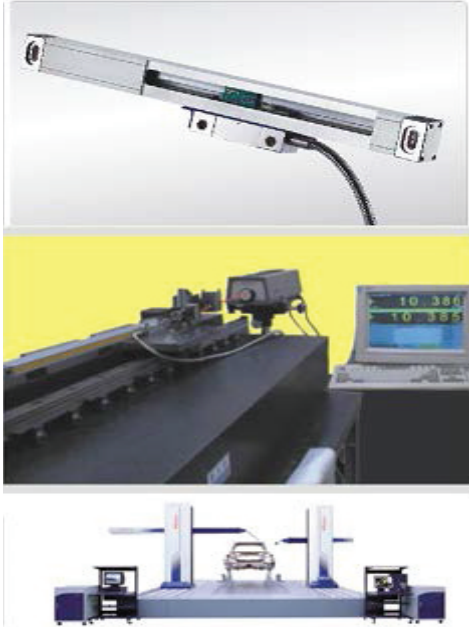
**Fig. 15. Interfaz con PC.**

El código de esta interfaz consiste en estar leyendo cada cierto tiempo (fijado por código) los valores de los 4 bytes para cada eje y presentar la lectura que resulta de multiplicar las cuentas recibidas por el periodo de la señal establecido. Para ello se debió hacer una función que permitiera juntar los 4 bytes leídos en un sólo tipo de dato de 32 bits, que representara las cuentas mandadas por el FPGA.

#### V. Resultados

La interfaz con PC y el código VHDL para FPGA (utilizando la tarjeta NEXYS2 de Digilent con el FPGA Xilinx Spartan 3 XC3S500E) fueron probados a través de una escala generadora de señales AquadB de esas que se montan en centros de maquinado o tornos. Además también fue probado el correcto funcionamiento mediante un interferómetro láser heterodino construido en la División de Metrología Dimensional del CENAM [7], sin dejar de lado las Máquinas de Medición por Coordenadas (CMM) en las que también fue exitosa la lectura de las escalas que al igual que un interferómetro generan las señales

AquadB[6]. La figura 16 muestra los dispositivos utilizados para probar el correcto funcionamiento de la interfaz con PC y el código VHDL.



**Fig. 16. Algunos dispositivos generadores de señales AquadB.**

## VI. Conclusiones

Varias son las conclusiones que podemos ofrecer con este tipo de trabajos, algunas de ellas son el poder ofrecer el código VHDL para un uso adecuado del mismo. Además este trabajo representa la fase posterior a un trabajo de tesis, ver [6], cuyo objetivo está encaminado a corregir errores de geometría sistemáticos en CMM, y la etapa de lectura de las escalas de las CMM tiene que lograrse de alguna manera, por lo que el FPGA es una manera adecuada. Cabe aclarar que esta corrección sistemática actualmente ya se realiza a través del uso de microcontroladores y el integrado HCTL2022 pero dicha corrección está limitada al uso de CMMs del tipo manual y no a las del tipo CNC por problemas de velocidad, cuyo obstáculo debería ser vencido utilizando FPGA, de ahí la motivación y el esfuerzo para utilizar esta tecnología. Otro punto importante como se menciona en los resultados,

es que, la División de Metrología Dimensional del CENAM recientemente ha construido un interferómetro heterodino, ver [7], y se pretende que toda la electrónica de lectura de posicionamiento e interpolación sea realizada a través de FPGAs.

El código VHDL para el lector AquadB se puede encontrar en alguna otra fuente, ver [3], aunque no con las mismas características que el que se presenta en este trabajo, sin embargo, la interfaz serial para un lector de señales AquadB con PC no se ha localizado en ninguna otra fuente.

## VII. Referencias

- [1] Digital Linear and Angular Metrology. Position Feedback for Machine Manufacture and Electronics Industry. Heidenhain. Alfons Ernst.1998 third, completely revised and enlarged edition.
- [2] [www.agilent.com/semiconductors](http://www.agilent.com/semiconductors), December 2005.
- [3] (<http://www.atmel.com>), under the 3037.zip archive.
- [4] [http://www.digilentinc.com/data/products/nexys2/nexys2\\_rm.pdf](http://www.digilentinc.com/data/products/nexys2/nexys2_rm.pdf), 11 de Julio de 2011.
- [5] <http://www.digilentinc.com/data/documents/Reference%20Designs/PwmRefComp1.zip>, 21 de Julio de 2008.
- [6] Tesis profesional de maestría de Octavio Icasio Hernández. “Método y Módulo Electrónico para mejorar la exactitud de Máquinas de Medición por Coordenadas”. Universidad de Guanajuato, México 2005.
- [7] Tesis profesional de Miguel Angel Benitez Morales. Sistema de Medición e interpolación para interferometro heterodino. Instituto Tecnológico de Tuxtla Gutiérrez. Mayo de 2009.
- [8] Multiplicador Electrónico para Encoder Incremental. Agustín Cruz Contreras, Edgar A. Portilla Flores y Ramón Silva Ortigoza A. Cruz Contreras, Centro de



Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional, México, D. F. (teléfono:57296000 Ext. 52536; e-mail:acruz@ipn.mx). 25 de noviembre del 2008.

### VIII. Autores

**M. en I. Octavio Icasio Hernández**, obtuvo su título como Ing. En Sistemas Computacionales en el Instituto Tecnológico de Celaya y su título de Maestría en Ingeniería con especialidad en Procesamiento de señales en la Facultad de Ingeniería Mecánica Eléctrica y Electrónica perteneciente a la Universidad de Guanajuato y ubicada en la ciudad de Salamanca.

Actualmente ha sido aceptado en el Doctorado en Tecnología Avanzada en el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (CICATA) campus Querétaro perteneciente al Instituto Politécnico Nacional (IPN), en el área de análisis de imágenes con el proyecto de reconstrucción de interiores 3D utilizando un fibroscopio. Desde 1998 trabaja como Metrólogo en el Centro Nacional de Metrología (CENAM) en el cual desempeña labores en los laboratorios de Máquinas de Medición por Coordenadas y Máquinas de Medición de Redondez y ha tenido estancias por extensión de 1 año primero con la empresa Unimetrik ubicada en Vitoria España en el área de metrología dimensional y también con la empresa API ubicada en Maryland USA, en el uso y aplicación de Laser Tracker.