

Control y adquisición serial de señales ultrasónicas con MATLAB

Control and serial acquisition of ultrasonic signals with MATLAB

Andrés D. Restrepo G.¹, Humberto Loaiza C.² y Eduardo F. Caicedo B.³

RESUMEN

Este artículo pretende brindar una introducción a la característica del programa MATLAB, a partir de la versión 6.1, de permitir la transferencia de datos a través del puerto serial para leerlos o escribirlos directamente desde su entorno, ya sea con el fin de soportar varias interfaces comerciales de instrumentación, o para manejar los puertos COM del PC. El trabajo toma como referencia un sistema microcontrolado diseñado con el fin de lograr un reconocimiento de formas de obstáculo mediante ecos ultrasónicos.

Palabras clave: adquisición de datos, pseudovisión ultrasónica, señal ultrasónica, puerto serial, MATLAB, sistemas microcontrolados, familia HC11.

ABSTRACT

This article intends to introduce features of MATLAB (version 6.1 onwards) which lets data be transferred through a serial port for reading it or writing it directly from Matlab. This can be done for supporting several commercial instrumentation interfaces or driving PC COM ports. This work is based on a micro-controlled system designed in such a way that the shape of an object can be recognised by ultrasonic echoes.

Keywords: data acquisition, sonar pseudo-vision, ultrasound, serial port, MATLAB, microcontroller system, HC11 family

Recibido: mayo 10 de 2005

Aceptado: octubre 2 de 2005

Introducción

Las características de MATLAB se ven potenciadas cuando se cuenta con una tarjeta de adquisición de datos; sin embargo, existen aplicaciones que no requieren demasiada precisión y en las cuales una costosa tarjeta de adquisición puede ser reemplazada por un hardware menos complejo y más barato. Ejemplo de esto es el hardware que se describe en la sección de descripción del sistema de exploración ultrasónica, consistente en un circuito con microcontrolador (μC) y una interfaz a MATLAB, diseñados para implementar un sistema de pseudovisión ultrasónica cuyo objetivo es lograr un reconocimiento de la forma del objeto irradiado (Restrepo,

2005a y Restrepo, 2005b). Tomando esta aplicación como referencia se mostrará la forma de implementar una interfaz serial manejada directamente desde MATLAB, explicando en las secciones de control de la secuencia de exploraciones y adquisición de muestras respectivamente el control de la exploración ultrasónica y la adquisición de las señales de eco. Luego, en la sección: Apartes del código en MATLAB, se revelan detalles del código de los procedimientos para establecer la comunicación entre MATLAB y el hardware de adquisición, terminando entonces con las conclusiones de esta exposición.

¹ Ingeniero electrónico y M.Sc. en automática de la Universidad del Valle. Profesor del programa de Ingeniería Electrónica y vinculado al Grupo de Instrumentación Electrónica (GIE) de la Universidad Santiago de Cali. adareg@telesat.com.co; adareg378@yahoo.com

² Ingeniero electricista y M.Sc. en Automática de la Universidad del Valle. Ph.D. en robótica y visión artificial de L'Université d'Evry, Francia. Profesor titular de la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle, y vinculado al Grupo de Investigación en Percepción y Sistemas Inteligentes (PSI), director del PPIEE de la misma universidad. hloaiza@eiee.univalle.edu.co

³ Ingeniero electricista de la Universidad del Valle, MSc en Informática Industrial y Ph.D. en Informática Industrial de la Universidad Politécnica de Madrid. Profesor titular de la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle y vinculado al Grupo de Investigación en Percepción y Sistemas Inteligentes (PSI). Director del Departamento de Educación Desescolarizada de la misma universidad. ecaicedo@eiee.univalle.edu.co

Descripción del sistema de exploración ultrasónica

La Figura 1 muestra un diagrama general del circuito mencionado en el que pueden reconocerse las siguientes etapas básicas:

- Módulos de emisión ultrasónica.
- Módulos de recepción ultrasónica.
- Etapa microcontrolada.
- Interfaz con MATLAB

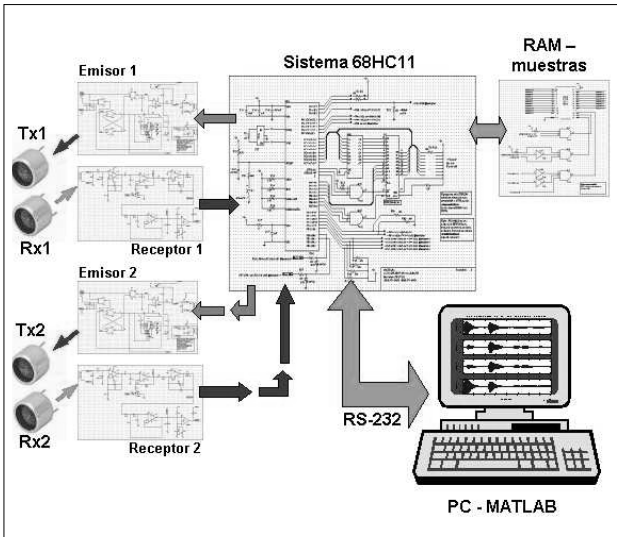


Figura 1. Diagrama general del montaje.

Módulos de emisión ultrasónica.

El funcionamiento del módulo (Figura 2) se sintetiza de esta manera: una señal de disparo /TRIG manejada por el μC inicia la emisión de un tren de pulsos de igual frecuencia que la de resonancia del elemento piezoeléctrico del transductor (Restrepo, 1999 y Gwirs et al.).

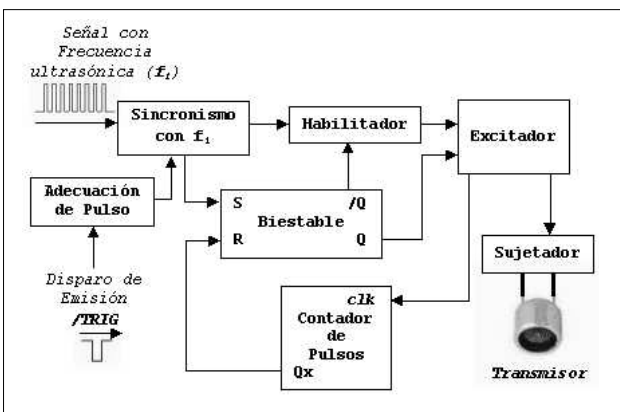


Figura 2. Diagrama de bloques del Módulo de Emisión

Cuando le llega este pulso de disparo, el módulo se sincroniza con la onda cuadrada de entrada de tal forma que todos los pulsos de emisión resulten del mismo an-

cho. La cantidad de estos pulsos (2, 4 u 8) puede ajustarse con jumpers que seleccionan las líneas adecuadas de un contador binario. De otra parte, se cuenta con una línea de salida que puede usarse como señal de desactivación de la recepción durante la zona muerta del sistema pulso - eco.

Módulos de recepción ultrasónica

Este receptor (Figura 3) consiste en un recortador seguido por un primer amplificador de ganancia fija que además filtra las frecuencias superiores a 40KHz; a continuación, una red RC filtra cualquier posible componente inferior a 40KHz para que posteriormente una segunda etapa de amplificación, que puede configurarse como amplificador de ganancia programable (PGA) lleve a la señal de eco a una amplitud de voltaje adecuado (salida eco). Después de un nuevo desacople de DC, aparece un rectificador activo de media onda (salida *eco_rect*), para continuar con un filtro pasa bajos de 2do orden con el fin de obtener a través de la salida *envolv*, la envolvente del eco.

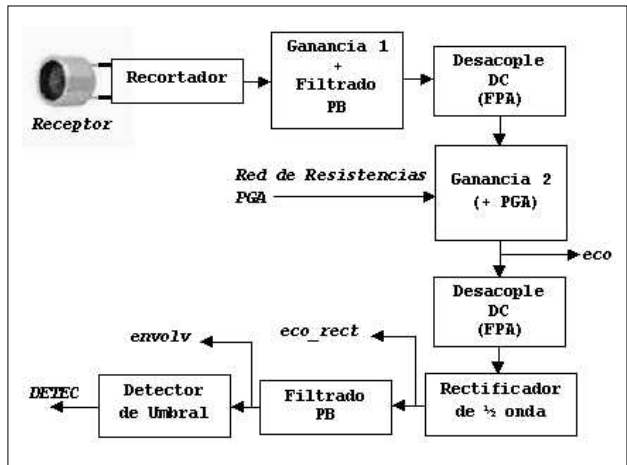


Figura 3. Diagrama de bloques del Módulo de Recepción

Por último, se tiene un detector ajustable de umbral que genera la señal de detección de eco (DETEC). En el desarrollo del estudio sobre el reconocimiento de formas con ultrasonido se deseaba tener toda la información del eco tal cual es captado, razón por la cual se eligió la salida eco para ser digitalizada.

Etapa microcontrolada

Esta etapa (Figura 4), controla la emisión de cada transmisor ultrasónico, adquiere la señal captada durante el intervalo de recepción, almacena las muestras digitales y se comunica con el PC a través del puerto serial.

El corazón de la misma lo constituye el μC de 8 bits 68HC11A8 de MOTOROLA (M86HC11Reference Manual, 1991 y Restrepo, 1999), del cual, se utilizan dos canales de conversión A/D con sus respectivas protecciones, uno por cada receptor; además, dos salidas digitales, una por cada emisor, para generar los pulsos de disparo

de emisión con un ancho de 24 s. Como memoria externa de programa se tiene una EEPROM 28C64, mientras que para el almacenamiento de las 2.000 muestras digitales se emplea una RAM 6116. Así mismo, el sistema de comunicación serial asíncrona (SCI) del μC es ideal para establecer una comunicación bidireccional con el PC a través del puerto serial con velocidades de hasta 250 Kbaudios con un cristal de 16 MHz. Todo el sistema está basado en una frecuencia de reloj de 16 MHz, lo que a su vez genera un período de muestreo de $8\mu\text{s}$.

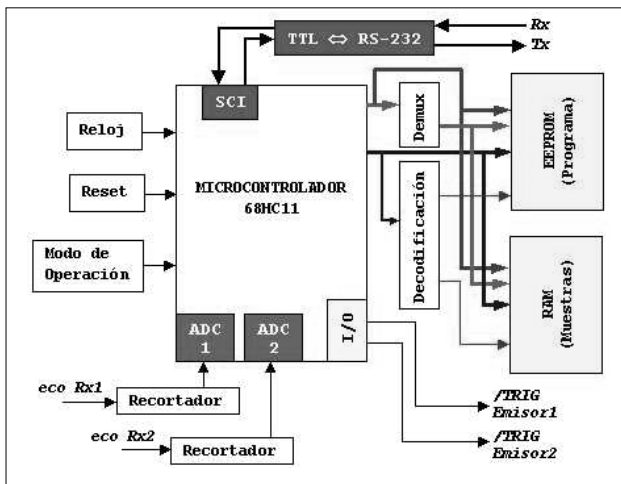


Figura 4. Diagrama de bloques de la etapa microcontrolada.

Interfaz con MATLAB

La comunicación serial RS-232 entre el microcontrolador y el PC se estableció a una velocidad de 9,6 Kbaudios, con ayuda del integrado MAX232. El mando general del sistema y la lectura de datos corren por cuenta de una función en MATLAB desarrollada con las instrucciones del toolbox de instrumentación (Instrument Control Toolbox, Versión 1. MATHWORKS). Adicionalmente, y para controlar la interacción entre el programa en MATLAB y el μC , se creó un conjunto de seis comandos, mostrados a continuación con sus nombres y valores decimales:

- REQUERIMIENTO DE RECONOCIMIENTO = 170
- RECONOCIMIENTO INICIAL = 85
- EMISIÓN = 240
- SEÑAL DETECTADA = 15
- LECTURA DE MUESTRAS = 51
- NUEVA EXPLORACIÓN = 204

Control de la secuencia de exploraciones

Los transductores ultrasónicos están ubicados en las esquinas de un rectángulo imaginario, similar al implementado en Kleeman, 2001. A través de los dos receptores en la parte inferior del montaje se obtienen dos puntos de vista del eco producido por la reflexión de la onda generada en uno solo de los emisores en la parte superior; por consiguiente, teniendo dos emisores se re-

cogerán en total cuatro perspectivas del objeto irradiado, o sea, cuatro señales de eco.

En la Figura 5 se presenta el esquema de exploración llevada a cabo en cuatro pasos claramente distinguibles. Fue necesario emitir dos veces con el mismo transductor debido a que muestrear dos canales a la vez haría incrementar al doble el período de muestreo del μC por efecto de la multiplexación de los canales análogos. El esquema de exploración se implementa tanto en la función desarrollada para MATLAB como en el Programa Principal del μC .

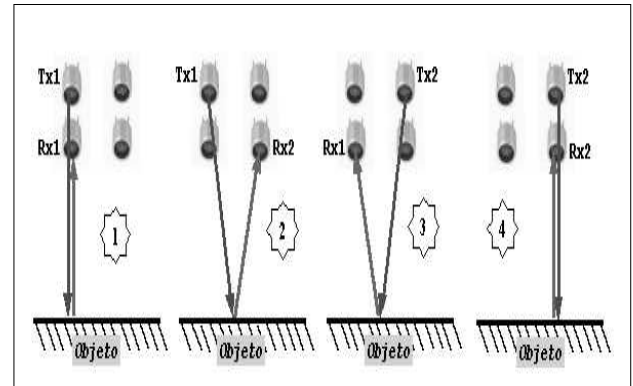


Figura 5. Esquema de exploración implementado

Durante la exploración y en cada emisión, se toman y se almacenan en RAM 2000 muestras de la señal de eco captada por la línea de recepción activa, más una muestra del nivel de DC en dicha línea justo antes de iniciar una nueva emisión. Esta muestra será usada para compensar el offset introducido por los circuitos con el fin de trabajar con conversores unipolares. En conclusión, durante cada vaciado de RAM se transmiten serialmente 2001 bytes.

Adquisición de muestras

El diagrama de flujo de la Figura 6a corresponde al programa ejecutado en el PC para la exploración, en el cual cada bloque de Emisión T_{xi} - Recepción R_{xj} comparte el diagrama de flujo de la Figura 6b.

El programa se implementó como una función de nombre `explor1.m`, ejecutándose de esta forma:

```
>>ecos=explor1(ultrason,margent);
```

La variable `ultrason` corresponde al objeto definido para manejar el puerto serial, y la variable `margent` es un valor real que representa el tiempo en milisegundos que debe esperar la rutina antes de iniciar una nueva emisión. Por su parte, la variable `ecos` contendrá las 8000 muestras provenientes de las cuatro emisiones que componen una exploración completa. Cada conjunto de 2000 muestras es almacenado en una variable de nombre `eco_Tx#Rx#` donde: $Tx\#$ y $Rx\#$ representan respectivamente el transmisor y el receptor activos en dicha exploración parcial. Al final, los cuatro vectores fila anteriores se reúnen formando las filas de una matriz de dimensión 42000 bautizada con el nombre de `ecos`.

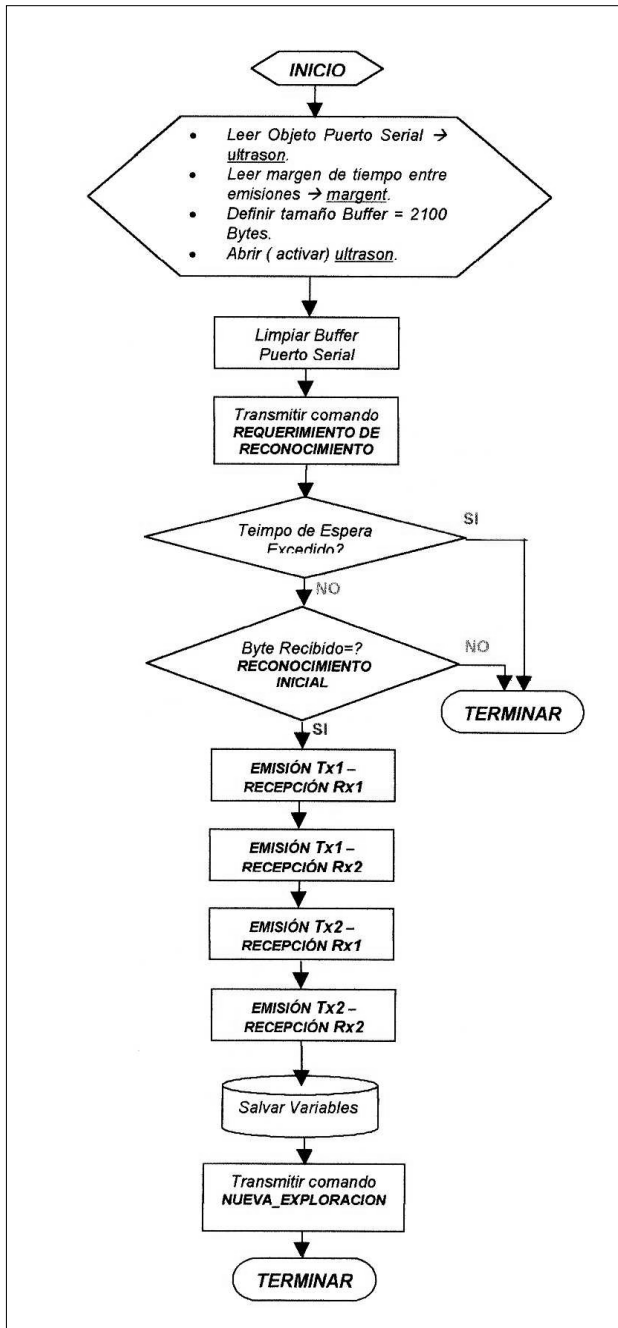


Figura 6a. Diagrama de flujo del programa en MATLAB (PC): esquema general

Existen dos casos que conllevan a dar por terminada la ejecución de esta función: el primero consiste en un valor erróneo del comando recibido, mientras que el segundo se da cuando se ha alcanzado un tiempo determinado de espera sin que ningún dato llegue al PC, o sin que lleguen completas las 2001 muestras de eco.

Apartes del código en MATLAB

A continuación se presentan algunas instrucciones importantes para establecer la comunicación a través de MATLAB. La primera instrucción que debe verse es:

```
ultrason = serial('COM1');
```

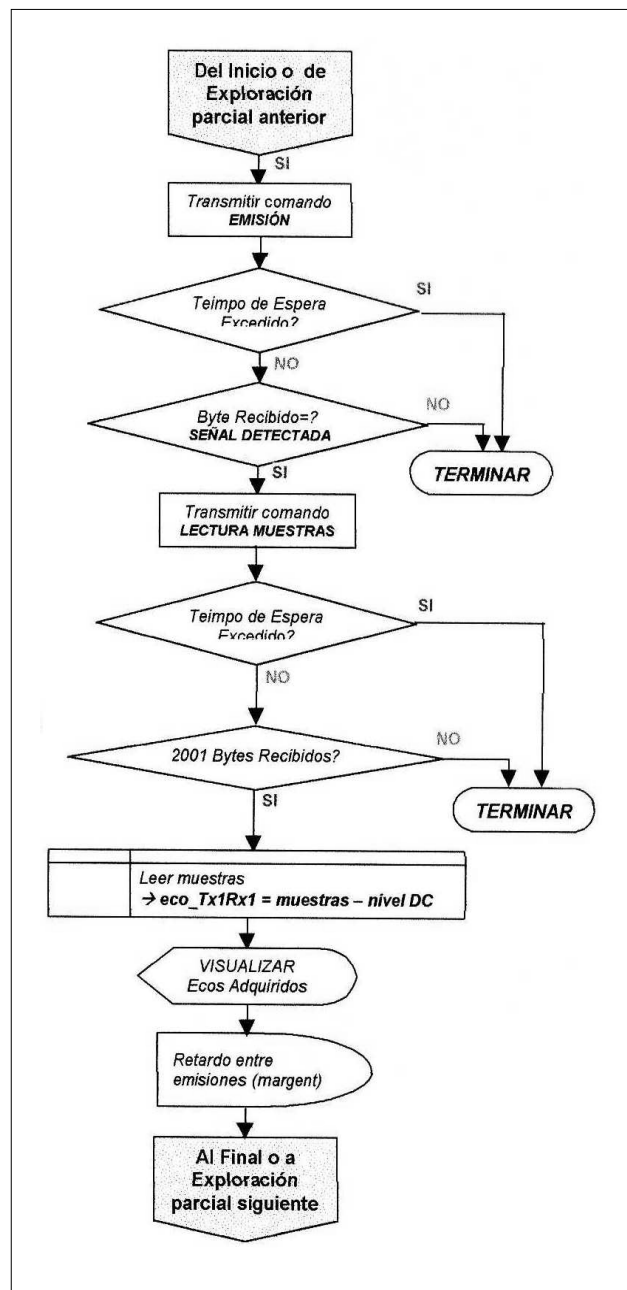


Figura 6b. Diagrama de flujo del programa en MATLAB - Rutinas de emisión y recepción

La función serial permite crear el objeto (al cual se le ha dado el nombre **ultrason**) que contendrá todas las características del puerto físico que se usará y el tipo de interfaz. Al requerirse el manejo de uno de los puertos seriales del PC, se pasa como argumento de la función el nombre que identifica a dicho puerto (COM1) según las definiciones del toolbox. Si se va a usar cualquier otra interfaz de instrumentación reconocida por MATLAB, se pasa como argumento el nombre que le identifica. Como toda variable tipo objeto, **ultrason** va a tener una lista de propiedades configurables por el usuario. Para ver la lista de estas propiedades puede usarse la función set en esta forma:

```
>>set (ultrason)
```

La misma función permite ajustar el valor o cualidad para una propiedad específica. De esta manera, desde la aplicación que corra en MATLAB es posible alterar la configuración del puerto sin necesidad de hacerlo desde el sistema operativo. Por ejemplo, para ajustar la tasa de baudios a 9600 bps basta con hacer:

```
set(ultrason, 'BaudRate', 9600).
```

es válido también: **ultrason.BaudRate=9600.**

Adicionalmente, fue necesario ajustar el tamaño del *buffer* de entrada, ya que por defecto es de 512 bytes de capacidad, y el número total de bytes que se leerán de la RAM es de 2001. Por tanto, se ejecuta la siguiente instrucción, dejando una cierta holgura en la capacidad del buffer:

```
ultrason.inputbuffersize=2100
```

Una vez que se han ajustado las propiedades requeridas para el puerto, es necesario habilitarlo (conectarlo o abrirlo) para la lectura y escritura de datos mediante la orden:

```
fopen(ultrason)
```

Mientras tanto, la lectura y la escritura asíncronas se realizan empleando las funciones **fread** y **fwrite**, las cuales trabajan con datos binarios de distintos tipos; por defecto los datos son de tipo carácter o enteros de 8 bits.

Antes de entablar con el hardware el enlace de comunicación con el comando REQUERIMIENTO_RECONOCIMIENTO, es aconsejable detectar al inicio de cada exploración completa, si en el puerto existen bytes no leídos con el fin de asegurar que el primer carácter que llegue después del envío de dicho comando sea la respuesta de reconocimiento del C. Para ello se necesita ahora obtener el valor actual de la propiedad de bytes disponibles del puerto:

```
if ultrason.bytesavailable~=0
    datoin = fread(ultrason,ultrason.
        bytesavailable,'uchar')
end
```

El comando get también puede utilizarse para el propósito de indagar por el valor de una propiedad del puerto:

```
noleidos=get(ultrason, 'bytesavailable')
o simplemente:
noleidos=ultrason.bytesavailable
```

Mientras tanto, la función **fread** necesita como argumentos el nombre del objeto que representa al puerto, el número de bytes que se leerán y el tipo de dato que se recuperará.

En este momento, ya es seguro enviar el comando de petición de reconocimiento:

```
fwrite(ultrason, REQUERIMIENTO_RECONOCIMIENTO, 'uchar')
```

Como se observa, la función **fwrite** necesita como argumentos: el nombre del objeto, el dato a enviar y el tipo de dato.

De otra parte, puesto que el μ C está programado para responder siempre a cualquiera de los comandos que le envíe el PC, como una medida de seguridad se implementa una rutina de espera que aborta toda la operación si se excede un tiempo máximo determinado.

```
t=clock;
while ultrason.bytesavailable==0,
    duracion=etime(clock,t)
    if (duracion>5)
        fclose(ultrason)
        errordlg('No se pudo establecer
            Comunicacion ...')
    return
end
end
```

La función **clock** entrega el tiempo actual basado en el reloj del PC, mientras que **etime** compara el tiempo actual (mediante **clock**) con otro instante previo y entrega la diferencia en segundos. En este caso, si después de 5 segundos no ha llegado nada al buffer de entrada al puerto (**ultrason.bytesavailable==0**) se cancela todo el procedimiento de exploración. Al mismo tiempo, es vital que se realicen las operaciones contrarias a las que se hicieron para la activación del puerto serial, detalle que se retoma más adelante. Si se recibe un dato en el puerto dentro del tiempo permitido, se ejecuta entonces la lectura correspondiente:

```
datoin = fread(ultrason,1,'uchar')
```

Ahora se compara el dato leído con el comando que debería haber llegado (rutina de comprobación), que en este punto corresponde al de RECONOCIMIENTO_INICIAL:

```
if datoin ~= RECONOCIMIENTO_INICIAL
    fclose(ultrason)
    errordlg('Respuesta incorrecta
        del...','Error de ...')
    return
end
```

En general, después de cada envío de un comando hacia el μC se ejecuta una rutina de espera, y luego de la lectura de un comando enviado por el μC se corre una pequeña rutina de comprobación, ambas similares a las anteriores. Enseguida se continúa con el proceso de exploración enviando el comando de EMISION:

```
fwrite(ultrason, EMISION, 'uchar')
```

Ante el comando anterior el μC , en este prototipo inicial del sistema ultrasónico, sabe qué transductor debe excitar para armar la secuencia correcta de emisiones y recepciones, de acuerdo al esquema de exploración explicado en el punto 3. Luego se lee el comando de respuesta (SENAL_DETECTADA):

```
datoin = fread (ultrason ,  
1, 'uchar')
```

Después de la comprobación, se ordena el vaciado de la memoria RAM donde se almacenaron las muestras obtenidas de la señal de eco correspondiente:

```
f w r i t e ( u l t r a s o n ,  
LECTURA_MUESTRAS, 'uchar')
```

Puesto que son 2001 muestras en total, la rutina de espera cambia respecto de las anteriores en la siguiente línea:

```
while ultrason .bytesa-  
vailable < 2001
```

Si no existe ninguna contingencia, se procede con la lectura de las muestras:

```
muestras =  
fread(ultrason,ultrason.bytesavailable,'uchar')
```

Se finaliza el proceso de emisión - recepción con la organización de las muestras de eco en un vector y la compensación del offset, y antes de iniciar una nueva emisión se espera el tiempo que pasó como argumento en la variable **margent**. Todo el procedimiento se repite hasta completar las cuatro emisiones y se envía el comando de NUEVA_EXPLORACION para dejar listo al μC ante una nueva exploración. Una vez que esto sucede, debe hacerse lo pertinente para inhabilitar las operaciones de lectura y escritura sobre el puerto, liberar el mismo o borrarlo definitivamente. De hecho, para culminar la función **explorus** se inhabilita (se desconecta o se cierra) el puerto serial, se quita de la memoria el espacio reservado para el objeto creado, y finalmente se quita el mismo del WORKSPACE, mediante las órdenes respectivas:

```
fclose(ultrason); delete(ultrason);  
clear ultrason
```

En realidad, la función **explorus** es invocada por un programa principal, por lo que la creación de la variable objeto para manejar el puerto, la liberación de memoria y la eliminación del mismo objeto, pueden ejecutarse en el programa principal; en mientras tanto, la conexión y desconexión del puerto corre por cuenta de la función **explorus**. Esta estrategia permite mantener el puerto serial disponible en tanto el programa principal se ejecute para que en cualquier momento se pueda hacer una exploración, evitando siempre la situación en la que se intente abrir el puerto cuando no ha sido previamente cerrado, ya que se producirá un error. En la Figura 7 aparece un ejemplo de visualización de las señales de eco captadas.

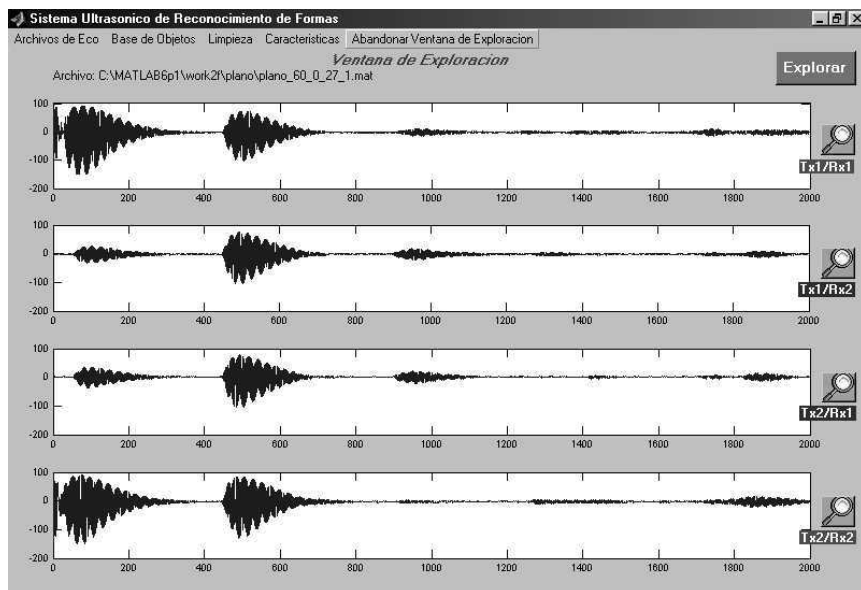


Figura 7. Visualización de señales captadas

Conclusiones

En el presente artículo se ha presentado un sistema barato de adquisición de señales ultrasónicas desarrollado alrededor de un μC de 8 bits de la familia HC11 de Motorola, y controlado directamente desde MATLAB a través del puerto serial. La ventaja principal de aprovechar esta cualidad de transferencia serial de datos, consiste en que se puede tener una interfaz completa que interactúe con un hardware específico usando únicamente los comandos y funciones de MATLAB, y desapareciendo la necesidad de desarrollar programas en otros lenguajes para que sean invocados por la herramienta al realizar las tareas de comunicación con el exterior del PC. Si bien esta característica es aprovechable en la medida en que la velocidad de ejecución de las rutinas de adquisición en MATLAB no constituya un elemento crítico para el buen funcionamiento de la aplicación que haga uso de ella, cabe la posibilidad como trabajo futuro de experimentar con la interfaz USB, más veloz, a través de los llamados puertos COM virtuales. Específicamente, el sistema ultrasónico de reconocimiento de formas pre-

sentado en el artículo tuvo un desempeño exitoso haciendo uso de esta utilidad, permitiendo que con un solo click en una interfaz gráfica de MATLAB se hiciera todo el proceso, desde explorar el obstáculo hasta obtener un resultado de su ubicación y su forma.

Bibliografía

Kleeman, L. Advanced Sonar Sensing., Proceedings 10th International Symposium Robotics Research, Lorne Victoria, Australia, pp. 286-295, noviembre 2001.

Gwirc, S.; Ferdeghini, F; Comastri, A. y Lupi, D. Sensores ultrasónicos: respuesta a distintas formas de onda., Centro de Investigación y Desarrollo en Electrónica e Informática (Citei), Instituto Nacional de Tecnología Industrial, Argentina., [on line] <http://sumar.utic.com.ar/ondas.pdf>

Instrument Control Toolbox, Versión 1., MATHWORKS.

M86HC11Reference Manual., Rev. 3, Motorola Inc, 1991.

Restrepo, A. y Millán, J. Medición de nivel en líquidos inmiscibles con un sensor ultrasónico inteligente., tesis de Ingeniería Electrónica, Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, 1999.

Restrepo, A. D. Concepción de un sistema de pseudovisión ultrasónica aplicando técnicas frecuenciales y de inteligencia computacional., tesis de maestría, PPIEE, Universidad del Valle, Cali, Colombia, 2005a.

Restrepo, A., Loaiza, H. y Caicedo, E. Sistema ultrasónico de reconocimiento de formas basado en técnicas frecuenciales e inteligencia computacional., Memorias del X Simposio de Señales, Imágenes y Visión Artificial, Universidad del Valle, octubre de 2005b.

Laboratorio de geotecnia



SERVICIOS

- Ensayos de caracterización física y mecánica de suelos, rocas, enrocados y agregados
- Ensayos de caracterización de materiales asfálticos
- Ensayos para control de calidad de agregados y mezclas asfálticas producidas en caliente.
- Diseño de mezclas asfálticas.
- Preparación de muestras para ensayo
- Investigación Básica y Aplicada

ROCAS



SUELOS



PAVIMENTOS



Mayor información:

Universidad Nacional de Colombia, Bogotá. Edificio 406 - IEI Primer piso. Tel: (57 1) 3165000 Ext. 13342.

Fax 13303. E-mail: labsuropa_fibog@unal.edu.co

Contacto: Ing. Gloria I. Beltrán C. Ext. 13305. E-mail: gibeltranc@unal.edu.co