

Dispositivo inteligente de ultrasonido para construir un mapa de entorno

Víctor Campos, Leonardo Doré, Alfonso Alfonsi

Grupo de Arquitecturas de Sistemas de Control, Departamento de Computación y Sistemas, Escuela de Ingeniería y Ciencias Aplicadas, Universidad de Oriente, Núcleo de Anzoátegui, Barcelona, 6001, Venezuela.

Ave. Alterna, Barcelona, Anzoátegui.

Email: asas@cantv.net, Tlf. +58-281-2623224.

Resumen

Se desarrolla un sensor inteligente que obtiene datos por medio de transductores de ultrasonido de forma replicada; estos son procesados por un microcontrolador e integrados utilizando técnicas de fusión de datos basadas en la teoría de Bayes, con el propósito de construir un mapa de entorno. Se utiliza la metodología estructurada y técnicas de diseño modular para desarrollar el hardware; en el caso del Software, se usa el Lenguaje de Modelado Unificado dentro de los pasos de la ingeniería de software. El dispositivo permite dentro de un espacio muestral limitado determinar la presencia de objetos, proporcionando la distancia a su ubicación, con respecto a un punto de referencia, caracterizándolos dentro del escenario. Este aporte es importante, ya que en aplicaciones en robots, el uso de sensores son herramientas vitales y de bajo costo en la gestión de trayectorias dentro de un entorno.

Palabras claves: Sensor de ultrasonido, fusión de datos, mapa de entorno, microcontrolador, metodología estructurada, UML.

Intelligent ultrasound device to build an environment map

Abstract

An intelligent sensor is developed to obtains data, by means of transducers of ultrasonic in a replied way, and they are processed by a microcontroller and integrated using technical of data fusion, based on the theory of Bayes, with the purpose to building an environment map. Structured methodology and technical of design is used to modulate to develop the hardware; and in the case of the Software, it is used the of Unified Modelling Language inside the steps of the software engineering. The device allows inside a space limited muestral to determine the presence of objects, providing the distance to its location, with regard to a reference point, characterising them inside the scenario. This contribution is important, in Robots applications since, the use of sensors is a vital tools and low cost in the administration of trajectories inside an environment.

Keywords: Ultrasound sensor, data fusion, environment map, microcontroller, structured methodology, UML.

1. INTRODUCCIÓN

Los avances tecnológicos en la ciencia de los computadores y la comunicación han hecho que el procesamiento de sensores se forme como una disciplina con teorías y propios enfoques que permiten el uso de algoritmos complejos para obtener información en tiempo real.

Existen trabajos que fundamentan el presente aporte, Bennet y colaboradores [1], describen un instrumento inteligente que evalúa la distancia de un

objeto. Por su parte, Dahlkemper y Lagen [2], estudian el desarrollo de sensores que permitan reducir el costo de un sistema de automatización. Blanes y colaboradores [3], describen un sensor de ultrasonido, que registra la construcción de un mapa, enfocándose en obtener un mapa con un mínimo de lecturas aplicando interpolación. Alfonsi y colaboradores [4], presenta un sistema de instrumentación virtual para el procesamiento y generación de señales, utilizando técnicas de procesamiento digital de señales, diseño electrónico y desarrollo de software.

El objetivo es desarrollar un sensor inteligente basado en el uso de transductores de ultrasonido replicados para el muestreo de un escenario y posterior construcción del mapa de entorno. Para lograr esto, se obtienen los datos por medio de los transductores de ultrasonido, (dos pares Emisor / Receptor), que luego son procesados por un microcontrolador para obtener distancias, permitiendo, caracterizar los objetos dentro del escenario. Ésta información arrojada, será integrada utilizando técnicas de fusión de datos basadas en las teorías Bayesianas, preprocesamiento que suministra la información necesaria para la representación en forma gráfica, a través del software, del mapa de entorno.

2. METODOLOGÍA

El dispositivo desarrollado, esta integrado por dos subsistemas: Hardware, formado por un arreglo de transductores de ultrasonido replicados, un motor paso a paso y un microcontrolador, empleando la metodología estructurada [5] para el desarrollo del software de control; y Software, denominado Software Generador de Mapa de Entorno (SGME), que se encarga de procesar los datos y generar el mapa de entorno a través de una interfaz gráfica; aquí, se utilizan los pasos sugeridos por la ingeniería de software [5]: Análisis y Especificación de los requerimientos, Diseño, Codificación, Prueba e Integración. E Lenguaje de Modelado Unificado (UML), proporciona diagramas que permiten visualizar, especificar y documentar los componentes de la aplicación [6].

Otro método utilizado es el de Mapas de acumulación [7], que representa eficientemente el espacio ocupado y vacío, basándose en la construcción de un mapa cartesiano bidimensional dividido en un conjunto de $M \times N$ celdas cuadradas de tamaño $D \times D$. Cada celda tiene asociado un valor que indica el grado de Certeza de la Presencia de un Obstáculo (CPO) en esa localización.

En la elaboración del mapa de acumulación hay varias fuentes de incertidumbre, como son los falsos ecos, las medidas falsas, la no-reflexión, etc., y es por ello que se apoya en el modelo de la teoría Bayesiana, teniendo en cuenta la suposición siguiente de independencia: todos los puntos de rejilla son variables aleatorias independientes, lo que se hace es

actualizar la probabilidad de las celdas de la rejilla con las lecturas hechas por el sensor, como se indica en la Ecuación 1, Elfes [7] aborda este tópico.

$$P(OCC_{new}) = \frac{p(r/OCC)P(OCC_{old})}{p(r/OCC)P(OCC_{old}) + p(r/FREE)(1-P(OCC_{old}))} \quad (1)$$

2.1 Análisis y diseño del hardware

El subsistema de hardware, Figura 1, se encarga de controlar y procesar los ecos recibidos de un par de transductores de ultrasonido (telémetro) Devantech SRF04 [8], que obtienen los datos necesarios para el cálculo de la distancia del objeto detectado, y son rotados 360 grados sobre su eje mediante un motor paso a paso de 7,5 grados/paso. Para controlar ambos dispositivos se utiliza un microcontrolador PIC 16C74B-04 de Microchip Technology [9], sobre el cual recae la tarea del cálculo de la distancia y establece comunicación asíncrona con el computador (PC) a través del puerto paralelo. Lo anterior lo realiza mediante el software del control del hardware, que refleja el flujo y el contenido de la información (datos de control), entre los elementos del sistema [4, 5].

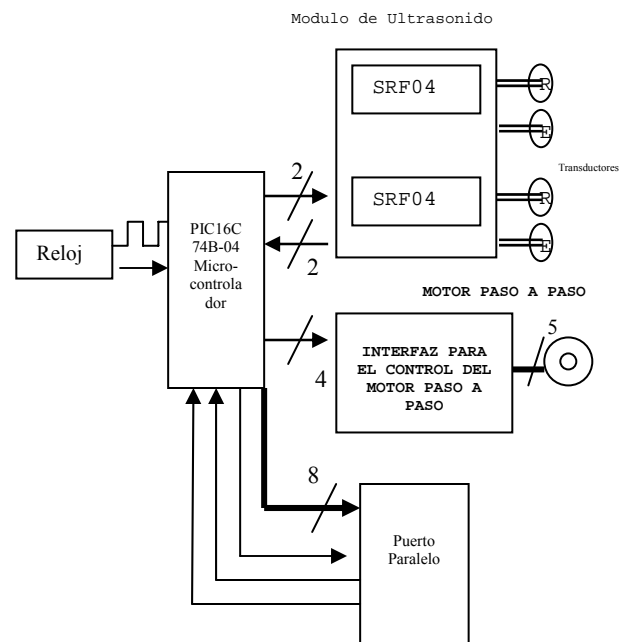


Figura 1. Diagrama de bloques del subsistema de hardware

2.2 Análisis y especificación de requerimiento del software

En la Figura 2 se muestra el diagrama de casos de uso de los requerimientos del SGME. Comportándose de la siguiente manera: el caso de uso capturar datos recibe una petición de un usuario para la adquisición de datos provenientes del sensor, habilitándolo para realizar los procedimientos que sirven de puente entre los dos subsistemas. Estos datos son almacenados en un “buffer”, representado por el archivo temporal para su posterior procesamiento.

El procesamiento de datos es llevado a cabo por el caso de uso generar mapa, hay que tomar en cuenta que los datos, se pueden manipular a través de las rutinas básicas de entrada y salida de archivo por medio del caso de uso manejo de archivo, así como también puede realizarse un listado impreso de los mismos por medio del caso de uso generar listado. Otro punto es la ayuda, donde el sistema debe comunicarse con el archivo de ayuda que contiene información sobre diferentes temas del software, y contiene texto que solo entiende el compilador del sistema de ayuda de Windows [10].

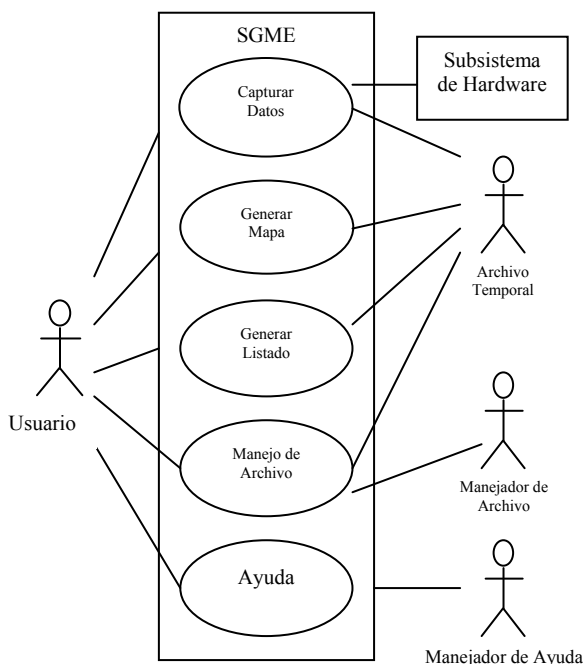


Figura 2. Diagrama de casos de uso para el SGME

2.3 Diseño del software

Aborda el diseño de: la interfaz de usuario y la estructura de software. La primera comprende la elaboración de la zona de comunicación usuario-programa de la aplicación, aquí se selecciona el estilo de pantalla y menú, (Figura 3), y la estructura de software debe soportar los documentos de especificación de requerimientos y el diseño de la interfaz de usuario. En la Figura 4 se presenta el diagrama de clase de la aplicación principal TForm1, existen cuatro (4) clases bien definidas y con comportamientos independientes, los cuales se relacionan a través de la clase principal, y una clase de particular interés que es la que maneja los errores. El diagrama de secuencia, Figura 5, muestra la forma en que se desencadena la ejecución de los casos de uso del sistema, y el diagrama de despliegue muestra la disposición física de los recursos de ejecución computacional, tales como un computador y sus interconexiones, que forman parte del sistema, Figura 6.

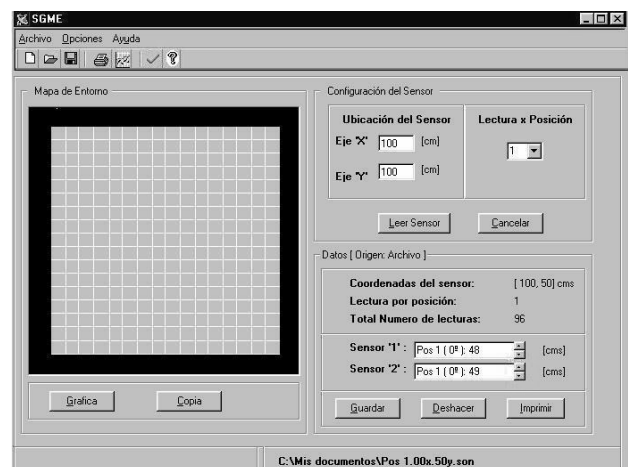


Figura 3. Pantalla principal del SGME

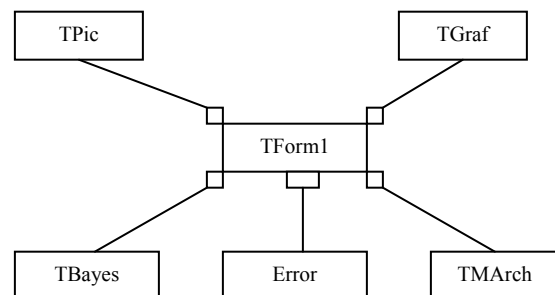


Figura 4. Diagrama de clases para la Aplicación Principal

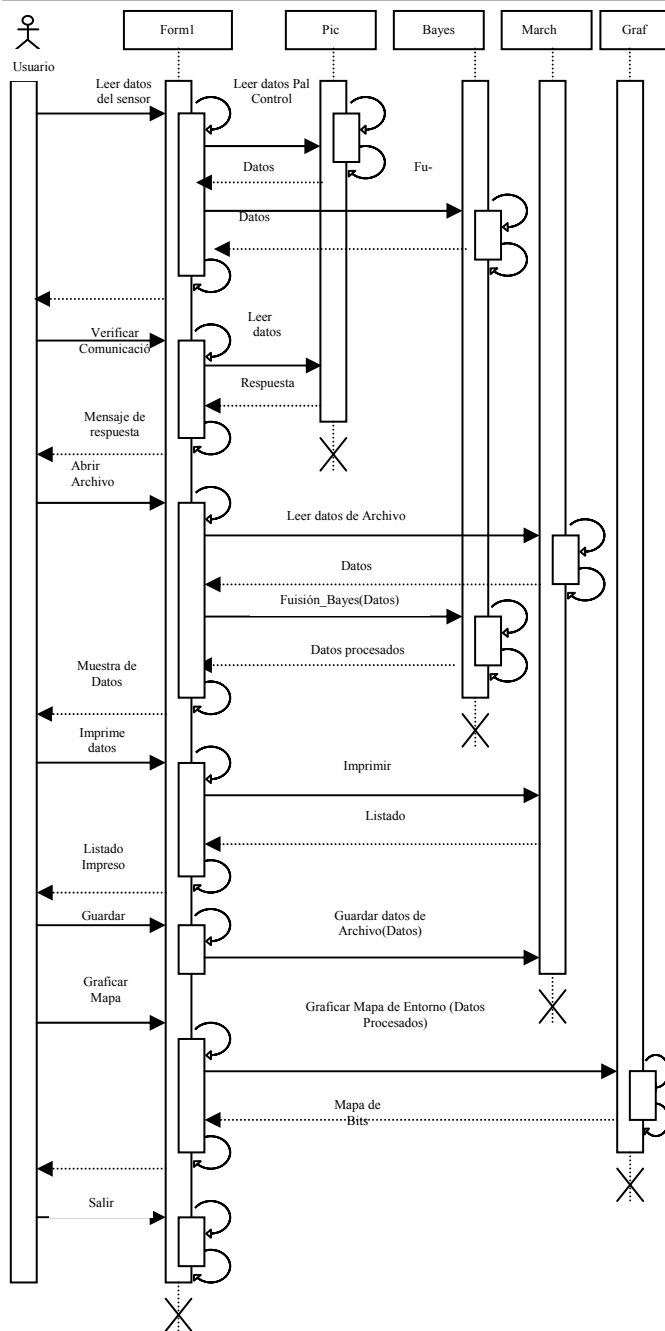


Figura 5. Diagrama de Secuencia para el Software

2.4 Codificación

El lenguaje seleccionado es el Delphi [11], trabaja sobre plataforma Windows, ofrece herramientas gráficas incorporadas en la biblioteca de componentes visuales (VCL) y la facultad para permitir incorporar código en el lenguaje de ensamble en secciones del programa.

Para construir el código encargado de manejar y controlar la adquisición de datos del PIC, se utiliza el lenguaje de ensamble. Trabajando con el sistema de interrupciones del microprocesador para acceder al puerto paralelo del PC [4], ya que es de vital importancia para la comunicación del sensor con el subsistema de software. De forma similar la especificación del código para el procesamiento de datos se basa principalmente en la implementación de las ecuaciones formuladas por Elfes [7].

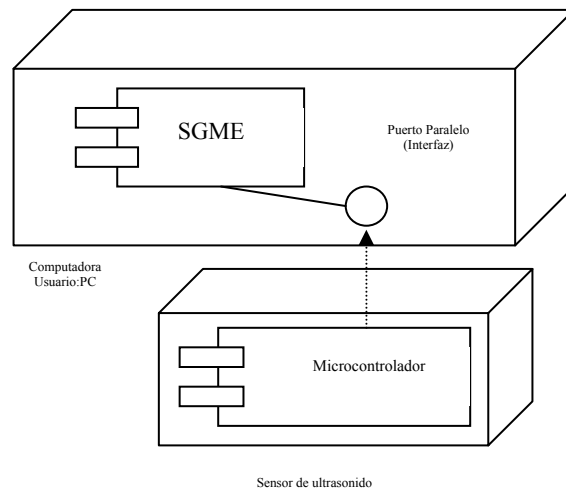


Figura 6. Diagrama de despliegue del sistema

2.5 Pruebas e integración

Para la elaboración del SGME, las pruebas fueron llevadas a cabo en paralelo con la codificación. Para validar las entradas de datos al programa, se recurrió a utilizar la prueba de unidad, en específico la caja negra, donde se validan los datos requeridos para la configuración de las lecturas que debe realizar el sensor y que representan las únicas entradas que requiere el SGME para realizar el muestreo y graficar el mapa del entorno.

Para demostrar la prueba, se procede a realizar una partición de equivalencia. En la Tabla 1 se observan las clases realizadas para verificar el comportamiento de la entrada de datos para configurar la posición del sensor, que debe coincidir con la ubicación real del sensor dentro del área establecida para el muestreo, y en la Tabla 2 se puede apreciar los casos de prueba realizados basándose en dichas equivalencia.

Tabla 1. Clases de equivalencia para la realización de pruebas

#	Clase de equivalencia	Valor Válido	Valor Inválido
1	La magnitud contiene números positivos.	√	-
2	La magnitud debe estar entre 1 y 300.	√	-
3	La magnitud del número es mayor que 300 o menor que 1.	-	√
4	La longitud del número debe ser mayor de cuatro cifras.	-	√
5	La magnitud es cero.	-	√
6	La magnitud contiene caracteres {A-Z, a-z}.	-	√
7	La magnitud contiene caracteres especiales como *,?,.,=,(), etc.	-	√
8	La magnitud del número contiene números negativos.	-	√

Tabla 2. Casos de prueba

Número	Clase involucradas	Resultado
/58	(7,1)	Inválido
Reyt	6	Inválido
Er78	(6,1)	Inválido
-9	8	Inválido
5	(1,3)	Inválido
356	(1,2)	Inválido
0	(2,5)	Inválido

Una vez que las pruebas de unidad fueron realizadas, se procedió a efectuar la integración, en éste proceso, los grupos de funciones y procedimientos pertenecientes a las clases agregadas se fueron integrando con las funciones de la clase (TForm1). El método de integración utilizado en el software es el hilvanado [5].

Para facilitar la representación de la integración se utilizó como herramienta los diagramas de paquetes proporcionados por UML [7], Figura 7. Las fases se describen a continuación:

Fase I: En esta se agregaron las clases TPic, TBayes, TGraf y TMArch, a la clase TForm1, encargada de inicializar el programa, además de mantener la comunicación con Windows, e incluye la barra de título, la barra de herramientas, los menús y la barra de estado.

Fase II: Parte fundamental de lo que es la interface entre el PC y el sensor de ultrasonido. En este proceso se integraron progresivamente los procedimientos necesarios para permitir la lectura de datos y verificar la comunicación con el PIC, los cuales se encuentran encapsulados en la clase TPic, y trabajan con operaciones o instrucciones en lenguaje de bajo nivel, lo cual permite tener acceso al puerto paralelo del computador. Para integrar los procedimientos a la ventana principal, fueron asociados a través de eventos, por medio de los botones “lectura de datos” y “verificar comunicación” respectivamente.

Fase III: Parte más importante del programa debido a que se incorpora las rutinas de procesamiento al SGME, la cual permite obtener un mapa de acumulación, representado por una rejilla bidimensional. El paquete procesamiento de datos: consta del procedimiento fusión de datos (de la clase TBayes), que agrupa las rutinas necesarias para la actualización de las celdas que forman parte del mapa de acumulación, el procedimiento establecer coherencia espacial (de la clase TGraf), que se encarga de relacionar a través de métodos trigonométricos la posición del sensor con las lecturas asentadas en el mapa de acumulación, y el procedimiento graficar mapa de entorno (de la clase TGraf), que plasma una representación gráfica del mapa de acumulación en la pantalla de bits, además de utilizar funciones de conversión para llevar coordenadas reales a coordenadas de pantalla, utilizando la ecuaciones presentadas por Pérez [12], integradas en la Figura 8, donde se puede apreciar parte de los procedimientos implementados en la clase TGraf.

Fase IV: Permite integrar al programa los procedimientos de entrada y salida de datos a través de archivos, entre éstos se pueden denotar abrir archivo, guardar archivo e imprimir los cuales se encuentran encapsulados en la clase TMArch, procedimientos que utilizan algunos componentes de VCL de Delphi [11], para poder realizar sus tareas.

Fase V: Agrupa los procedimientos para la ayuda del software. El sistema de ayuda presenta los tópicos referentes a cada una de las partes necesarias para manipular el SGME y tiene capacidad de permitir ayuda sensible al contexto, esta característica se hace disponible a través del procedimiento contenido de ayuda.

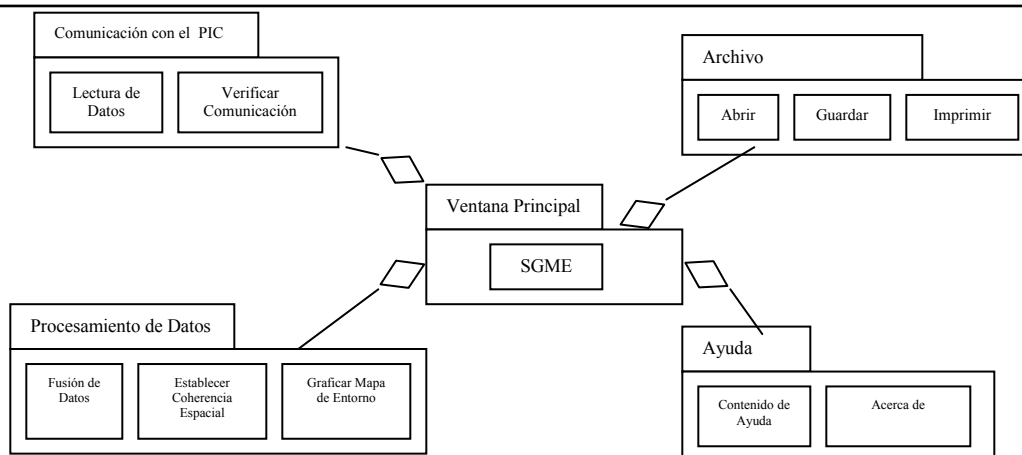


Figura 7. Diagrama de paquetes para representar la integración del Software

2.6 Integración de Software y Hardware

El control del hardware viene dado directamente por la acción de las líneas del puerto paralelo sobre cada una de las etapas que conforman el circuito del sistema, Tabla 3. Este se centra en la manipulación de los estados de las líneas del puerto mediante las instrucciones de lectura y escritura de datos representadas por las instrucciones de la función y los procedimientos en Delphi [11].

Cuando se programa el PIC16C74A-04, se hace uso de algunos recursos como: El puerto paralelo esclavo, que permite la interconexión directa del puerto D del PIC con el bus de datos del computador,

Tabla 3. Cuadro detallado de líneas de datos, control y estado del módulo sensor inteligente

Registro	Bit	Nombre Señal	Valor Lógico
Control	0	RD	0
Control	1	WR	0
Control	2	CS	0
Control	3	INT	1
Estado	3	ACT	1
Estado	4	SRD	1
Datos	0	D0	X
Datos	1	D1	X
Datos	2	D2	X
Datos	3	D3	X
Datos	4	D4	X
Datos	5	D5	X
Datos	6	D6	X
Datos	7	D7	X

teniendo este el control a través de las líneas RD, WR y CS del puerto E (que están configuradas como entradas). Perro guardián “Watch Dog Timer”, evita que el sistema se quede “colgado”, que permanezca en un ciclo infinito o que no obtenga una respuesta del computador. Está configurado para un tiempo de 2,3 segundos. Modo de reposo (SLEEP), característica útil para los sistema que utilizan baterías. Y las interrupciones consisten en una detención del programa en curso para realizar una determinada rutina.

```
//Transformación de puntos a pixeles
Funtion TGraf. Conversion(x1,x2,y2:integer;X:real): integer;
Var r:integer;
    m,b:real;
begin
    m:=((y2-y1)/(x2-x1));
    b:=y2-(x2*m);
    r:=trunc(m*X)+b;
    Conversion:=r;
end;
//Parte del código para Gráfica del Mapa de Entorno
for i:=1 to MaxDis do
    forj:=1 to MaxDis do
        begin
            temp:=GRID3[i][j];
            //puntos a graficar
            a:=Conversion(1,0,0,255,temp);
            BMP.Canvas.Pen.Color:=RGB(a,a,a);
            //asignación color
            BMP.Canvas.Brush.Color:=RGB(a,a,a);
            x:=conversion(0,Margen,300,300+Margen,i); //llamada a
función de conv.
            y:=conversion(0,300+Margen,300,Margen,j);
            BMP.Canvas.MoveTo(x,y); //dibuja punto en un mapa de
bits
            BMP.Canvas.LineTo(x,y+1);
        end;
    end;
```

Figura 8. Código para rutinas gráficas

3. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Al desarrollar el sistema inteligente, primero se analizó y diseñó el circuito de la estructura física del hardware que incluye: Sensores de ultrasonido, motor paso a paso y el PIC16C74B, Figura 9; y luego se describe la programación del PIC, que corresponde con el software de control del hardware, utilizando la metodología estructurada, que luego se convierte en programa modular. La utilización del PIC, representa un incremento potencial en la eficiencia del sensor, al hacerlo independiente del software del computador en el control de los dispositivos. Además de minimizar la utilización de componentes y circuitería externa encapsulándola a un nivel de software.

Como segundo paso, está la confección del software generador del mapa de entorno (SGME), cuya área de trabajo se mostró en la Figura 3, donde se siguió los pasos del desarrollo de software. Destacando que para la representación se utiliza el lenguaje de modelado unificado (UML), y al Delphi [11] como lenguaje de programación; ya en Alfonsi y colaboradores [4] se utilizan éstas herramientas, desde el punto de vista de adquisición de datos, sin intervención de un microcontrolador como dispositivo con capacidad de procesamiento y control, ya que el proceso lo realiza un PC.

Para su implementación se requiere del sistema operativo Windows 98 o superior, y que el computador donde se ejecutará la aplicación, posea entre otras, las siguientes características mínimas de hardware: Procesador Pentium o Celeron de 366 MHz o



(a) Módulo de Ultrasonido (b) Vista de la Tarjeta PIC



(c) Vista Frontal del módulo de sensor ultrasonido
Figura 9. Componentes del hardware

mayor, 32 Mb de memoria RAM, disponibilidad del puerto paralelo bidireccional, unidad de disco duro, monitor de 14 pulgadas SVGA, fuente de voltaje de 5 voltios, 1 amperio, un cable para puerto paralelo de unos 5 metros.

Los resultados experimentales que se describen a continuación fueron realizados en un escenario como se muestra en la Figura 10. El dispositivo es colocado y accionado en cuatro posiciones diferentes a lo largo y ancho del escenario que abarca un área de 3 m², donde se ubicaron objetos de diferentes características en forma vertical, entre ellos dos botellas de vidrio (diámetro de 8 cm y altura de 40 cm), una botella de plástico (diámetro de 6 cm y altura de 30 cm), palo de escoba (diámetro de 2,5 cm y 1m de alto), una lámina de hierro (60 cm x 75 cm), dos cajas de cartón (60 cm x 50 cm).

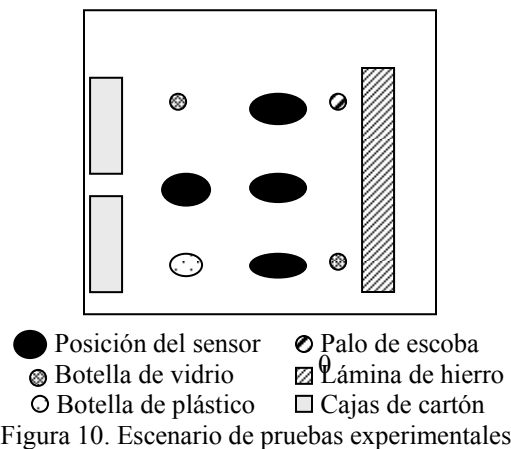


Figura 10. Escenario de pruebas experimentales

En la Figura 11 muestra como se va generando el mapa de acumulación en las diferentes posiciones donde es colocado el sensor. Observándose, como se van delimitando los diferentes objetos presentes en el entorno. A su vez, las celdas que corresponden con objetos ven como va aumentando su CPO (se intensifica el color negro), a medida que se van realizando más medidas sobre la zona del espacio que ocupan. Las falsas medidas desaparecen del mapa con el paso del tiempo, y las correctas inducen un aumento en la CPO de la celda correspondiente.

La utilización de sensores replicados, mejora la confianza del sistema al hacerlo tolerable a un número determinado de fallas, ya que las lecturas ofrecidas por los dispositivos son a menudo dispares, y ésta característica es precisamente la que lo hace atractivo

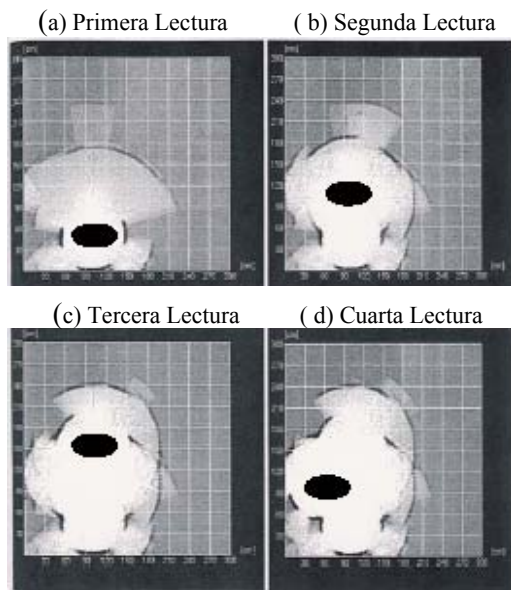


Figura 11. Generación del mapa y variación en las diferentes posiciones

al combinarla con técnicas de fusión de datos. No obstante, siguen teniendo un área geográfica de acción limitada, por tal motivo Blanes y colaboradores [3], atacan la construcción de un sistema, incorporándole métodos de interpolación para lograr una alta resolución angular, utilizando para ello cuatro transductores.

En el procesamiento de los datos crudos se debe resaltar que el uso del mapa de entorno constituido por celdas que indican la CPO, se consigue una mejora notable en la disminución de los errores y comprensión de las diversas fuentes de incertidumbre que influyen en el proceso de adquisición y generación del mapa. Además, las superficies lisas dejan de ser un elemento peligroso y prácticamente invisible para el sistema de ultrasonido, teniendo por lo general, algún tipo de irregularidad que hace posible su localización. Por otro lado, si el eje del transductor es perpendicular a la superficie, éste será detectada sin problemas, es decir, a pesar que el número de lecturas de los sensores no detecte este tipo de elementos, algunas medidas si reflejarán sobre el mapa su posición.

Como futuro desarrollo, se incorporarán técnicamente sensores dispares, con la finalidad de que un sensor compense las deficiencias del otro, aumentando la confiabilidad y la tolerancia a fallos, mejorando la detección o eliminación del ruido. Y la incorporación de un sensor de temperatura que permita

obtener la medida de la temperatura del aire y de ésta manera corregir los valores de la velocidad del sonido que se ven poco afectados por la misma.

4. CONCLUSIONES

Se desarrolló un sistema inteligente basado en el uso de sensores de ultrasonido, para el muestreo de un escenario, que permite su representación en mapa cartesiano. Se logró el propósito al utilizar dos sensores de ultrasonido replicados, esto para complementar las deficiencias y errores, en busca de redundancia. La información recibida por los sensores en forma individual, se integra con algoritmos de fusión de datos utilizando el teorema de Bayes, permitiendo un preprocesamiento de la señal que suministra información suficiente para la construcción del mapa de entorno. La reconstrucción del escenario a través de un mapa ayuda a determinar la presencia de objetos dentro de un espacio muestral limitado, proporcionando la distancia en donde está ubicado el mismo, con respecto al punto de referencia (ubicación del sensor). Se elaboró el software utilizando los procedimientos que brinda la metodología de la Ingeniería de software, sumado con las herramientas proporcionadas por el UML.

Como siguiente paso, se hará la integración del dispositivo como nodo en una red de área local de tiempo real, como parte del sistema de navegación de un robot.

AGRADECIMIENTO

Los autores quieren agradecer a los profesores Zulirais García y José Guevara, del departamento de computación y sistemas de la Universidad de Oriente, por las observaciones en la metodología empleada a nivel de Software.

REFERENCIAS

- [1] Bennet G., Albadejo J., Rodas A. and Gil P. (1992): "An intelligent sensor for ranging in an industrial distributed control system", In: Proc. Symposium on intelligent components and instruments for control applications, Málaga, pp. 47-52.

- [2] Dahlkemper J. and Langen A. (1994): "Sensor based solutions for automation", In: CAN in Automation e.V. Proc. 1er. International CAN Conference, Erlangen, pp. 11-2-11-9.
- [3] Blanes F., Bennet G., Simó J. and Pérez P. (1999): "Enhancing the real-time response of on ultrasonic sensor for map building task". In: Int Symposium Intelligent E, Bled, pp. 225-230.
- [4] Alfonsi A., Binoto L. y García J. (2000): "Sistema de instrumentación virtual para el procesamiento y generación de señales". In: Troyani N & Cerrolaza M (ed). V Cong. Int. de Métodos Numéricos en Ingeniería y Cs. Aplicadas, Puerto La Cruz, Venezuela, pp. EC-11 – EC-19.
- [5] Pressman R. (2002): "Ingeniería del software". McGraw-Hill. 5ta. ed., Madrid.
- [6] Rumbaugh J., Jacobson I. y Booch G. (2000): "El lenguaje Unificado de Modelado". Addison Wesley, Madrid.
- [7] Elfes A. (1989): "Using occupancy grids for mobile robot perception and navigation", Proc. IEEE Computer, Málaga.
- [8] ACROMED INC (2001): "The Devanthech SRF04 Ultrasonic Range Finder". ACROMED INC, USA.
- [9] Microchip Technology Inc (2002): "Microcontroller PIC". 05 de mayo 2002. <http://www.microchip.com/1010/picmicro/category/digictrl/>.
- [10] Ritcher J. (1997): "Programación Avanzada en Windows". McGraw Hill, Madrid.
- [11] Joyanes L. (1999): "Borland Delphi. Iniciación y Referencia". McGraw Hill, Madrid
- [12] Pérez A. (1996): "Técnicas de graficación de superficies". Rep. Interno Dpto. Computación y Sistemas, UDO, Barcelona.