

Diseño y aplicación de un Sonar de barrido

Alejandro Alonso Puig – mundobot.com
Diciembre 2003

Introducción	2
Diseño Hardware	2
Diseño software	5
Utilización de la aplicación.....	5
Mapas Polares	5
Mapas de Profundidad	6
Diseño de la aplicación	7
Conclusiones y líneas futuras.....	8
Bibliografía y enlaces de interés	9

Introducción

El siguiente informe técnico describe el diseño de un sonar de barrido y su aplicación para generar mapas polares y de profundidad

El objetivo de este trabajo ha sido mostrar de una forma palpable el modo en que se percibe el entorno mediante emisores-receptores de ultrasonidos.

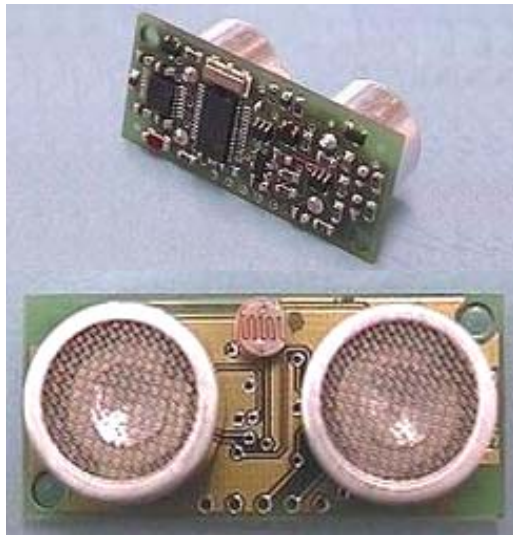
Los elementos fundamentales en los que se basa este trabajo son:

- Módulo Sonar
- Tarjeta para control de servos
- PC
- Tarjeta interface puerto paralelo - bus I²C
- Aplicación en Visual C++ 6.0 que controlará el conjunto y generará en pantalla los mapas polar y de profundidad.

En este trabajo se hará mención a todos los aspectos Hardware y Software, así como conclusiones de la experiencia.

Diseño Hardware

Como pieza clave se utilizará el módulo de ultrasonidos SRF08 de Devantech controlado por bus I²C, que permite detectar obstáculos alejados hasta 6 metros del sensor y obtener el valor de hasta 17 ecos.

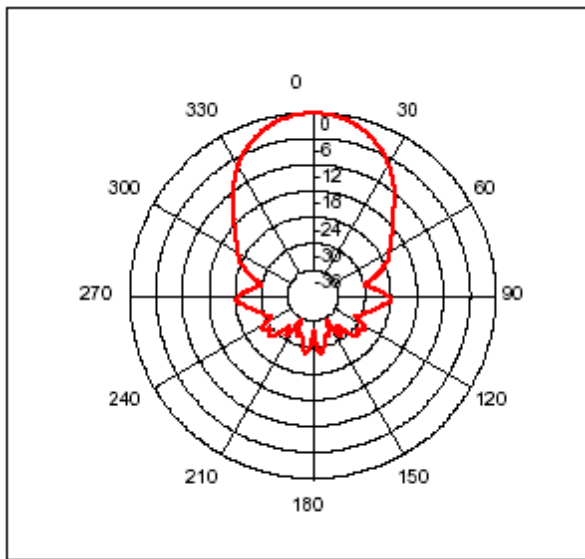


El haz emitido rebota en los objetos llegando al receptor. En función del tiempo que tarda en llegar el eco y del valor conocido de la velocidad del sonido, se calcula la distancia del objeto. El módulo SRF08 entrega directamente este valor en centímetros.

Sus especificaciones técnicas según las publica el fabricante son las siguientes:

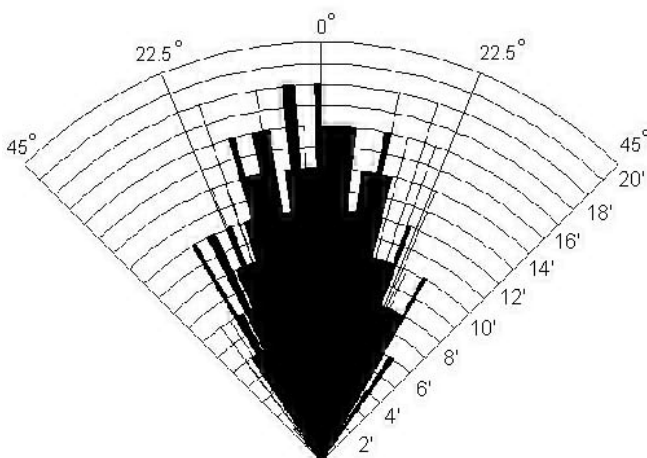
<i>Voltaje</i>	5v
<i>Consumo</i>	<ul style="list-style-type: none">▪ 275mA Max en encendido (llegada comando I2C) – 3uS▪ 40mA generación ultrasonido – 200us▪ 12mA Proceso de escucha – 65ms▪ 3mA Standby.
<i>Frecuencia</i>	40KHz
<i>Rango</i>	3 cm a 6 m
<i>Ganancia</i>	Variable. 94 a 1025 en 32 pasos
<i>Conexión</i>	Bus I ² C
<i>Sensor de luz</i>	En la parte frontal. Lectura por I ² C
<i>Timing</i>	Fully timed echo, freeing host computer of task
<i>Eco</i>	Múltiples ecos (17)
<i>Unidades</i>	Distancia reportada en uS, cm o pulgadas
<i>Peso</i>	11,34 gramos
<i>Dimensiones</i>	43 x 20 x 17 mm

El beam pattern del SRF08 es conico, siendo la apertura angular función de la superficie del transductor. El beam pattern del transductor SRF08, según especificaciones del fabricante es el que se muestra a continuación para una temperatura de 25°C y una frecuencia de trabajo de 40.0Khz (*información publicada por el fabricante del transductor en su datasheet: <http://www.robot-electronics.co.uk/datasheets/t400s16.pdf>*)



Los valores en la vertical están en decibelios. Así por ejemplo el ángulo de apertura del haz ultrasónico es de unos 55° típicos para unos -6dB.

Existe un modo más “visual” de mostrar el cono generado por el haz ultrasónico en este dispositivo y es el siguiente (*fuentes: Acroname [1]*):

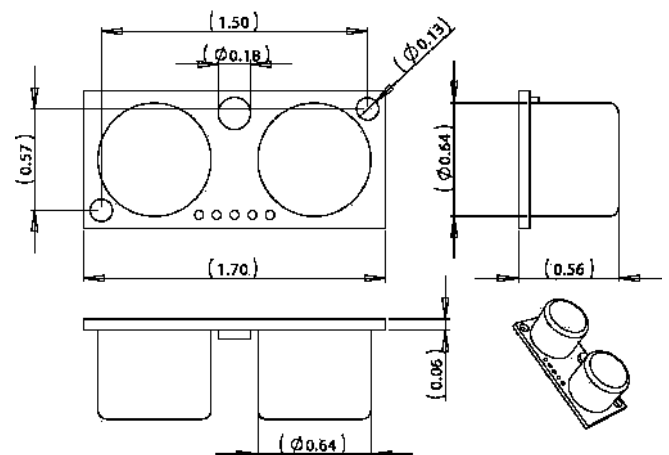


Las medidas de distancia (a la derecha) están en pies. 20 pies son aproximadamente 6 metros.

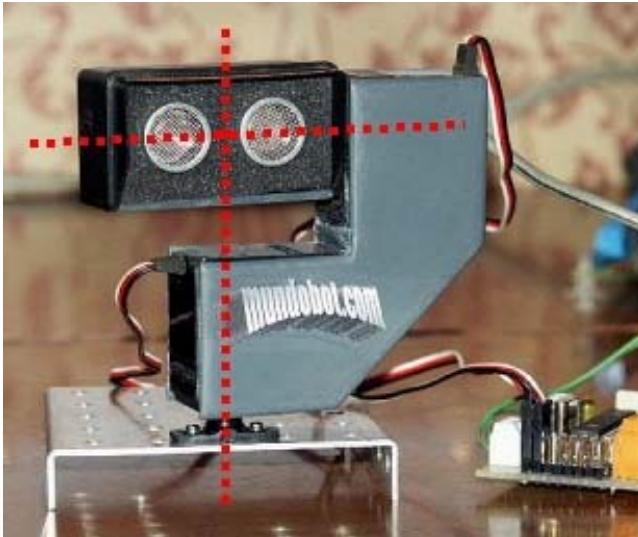
Como puede verse no forma un cono exacto, con lo que a distancias más cortas la anchura del haz es mayor que a distancias mayores.

Cualquier objeto situado en dicho cono producirá un rebote y por tanto un valor de distancia, aunque no se encuentre exactamente delante del sensor. Según la amplitud del cono se obtendrá una mayor o menor precisión selectiva de objetos.

Puede obtenerse información completa sobre el módulo SRF08 de Devantech en la dirección <http://www.robot-electronics.co.uk/html/srf08tech.shtml> donde también está publicado el diseño electrónico y programa en C utilizado en el mismo [1].



Dicho módulo se ha instalado adecuadamente en una carcasa rellena de espuma plástica para reducir vibraciones que puedan alterar la medición. Dicha carcasa está acoplada a una estructura de doble servo reforzada con planchas de PVC, para permitir giros horizontales y verticales del módulo. La posición de los servos en dicha estructura se ha dispuesto para que ambos ejes de giro pasen por el centro del conjunto sensor. De esta manera el sensor podrá llevar a cabo barridos unidimensionales y bidimensionales precisos.

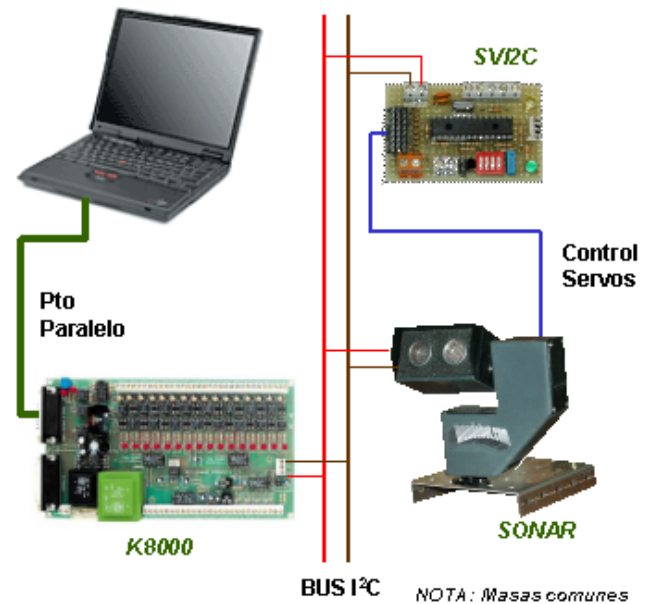


Los servos utilizados son modelos económicos de 3kgcm de par y libertad de giro de 180°, manejados por la tarjeta SVI2C desarrollada por el autor. Dicha tarjeta permite el control mediante bus I²C de hasta 8 servos y 5 entradas analógica (conversión A/D de 8 bits). Para información completa sobre esta tarjeta puede accederse a la dirección: <http://mundobot.com/tecnic/Svi2c/Svi2c.htm>

El conjunto está conectado al bus I²C de la tarjeta Interface K8000 de Velleman que permite acceder al bus I²C a través del puerto paralelo de un PC. Para información adicional sobre esta tarjeta y su manejo puede accederse al trabajo presentado por el autor en la dirección:

<http://mundobot.com/tecnic/ecoi2C/ecoi2c.htm>

Adicionalmente, en sustitución de dicha tarjeta puede utilizarse la tarjeta no comercial KI²C de bajo coste sobre la que pueden encontrarse todos los detalles de fabricación y funcionamiento en la dirección: <http://mundobot.com/tecnic/KI2C/KI2C.htm>



Es muy importante tener en cuenta que además de las dos líneas del bus I²C, han de estar conectadas las masas de los diferentes módulos para asegurar las mismas referencias en las señales generadas.

A continuación se muestra el conjunto completo montado en una base para pruebas de campo.

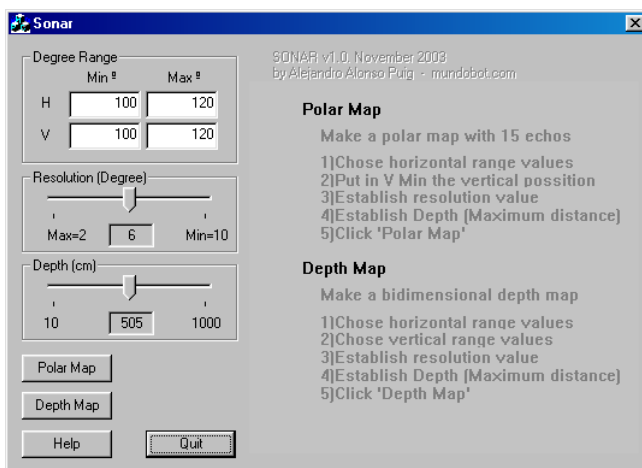


Diseño software

Para el control del conjunto completo y producción de mapas polares y de profundidad se ha generado la aplicación “Sonar”.

Utilización de la aplicación

El interface con el usuario consiste básicamente en un cuadro de diálogo como el que se muestra a continuación.



Mapas Polares

Un mapa polar se genera mediante el barrido del sonar en una única dimensión. El sensor se posicionará verticalmente en la posición en grados especificada en V_{min} y recorrerá horizontalmente las posiciones en grados de H_{min} a H_{max} . El factor de resolución indicará la amplitud de los saltos en grados que se realizarán en el desplazamiento. Por ejemplo, si se elige

$$V_{min} = 90$$

$$H_{min} = 20$$

$$H_{max} = 50$$

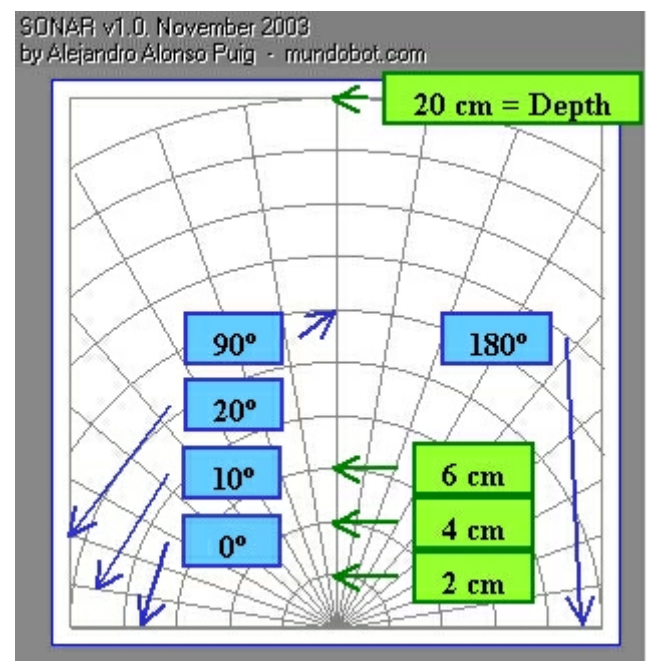
$$Resolution = 10$$

El sensor se colocará en 90° de posición vertical (haz horizontal a la base) y recorrerá horizontalmente las posiciones 20° , 30° , 40° y 50° , tomando medidas de distancia en cada una de esas posiciones.

El valor de profundidad (*Depth*) indica la distancia máxima que se desea medir. En el caso de mapas polares este valor será el que se establezca como límite del área de trazado.

Así mismo, el área de trazado posee divisiones radiales y concéntricas. Las divisiones radiales indican las decenas de grado, desde los 0° a los 180° . Cada división concéntrica representa la décima parte de la profundidad (*Depth*) especificada.

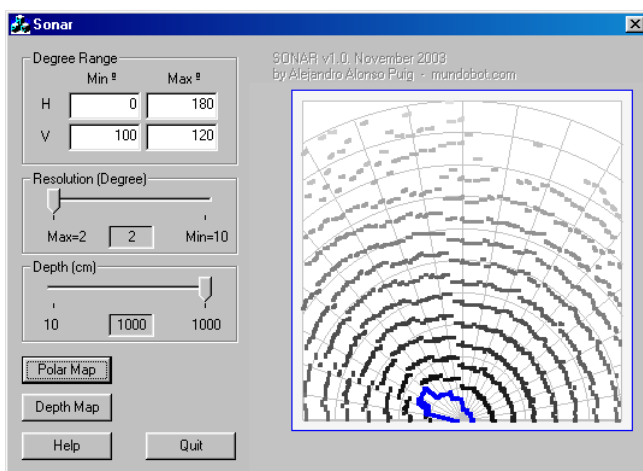
Así por ejemplo, para un *Depth*=20cm se tendrían los valores que se muestran en el siguiente gráfico para las líneas de división.



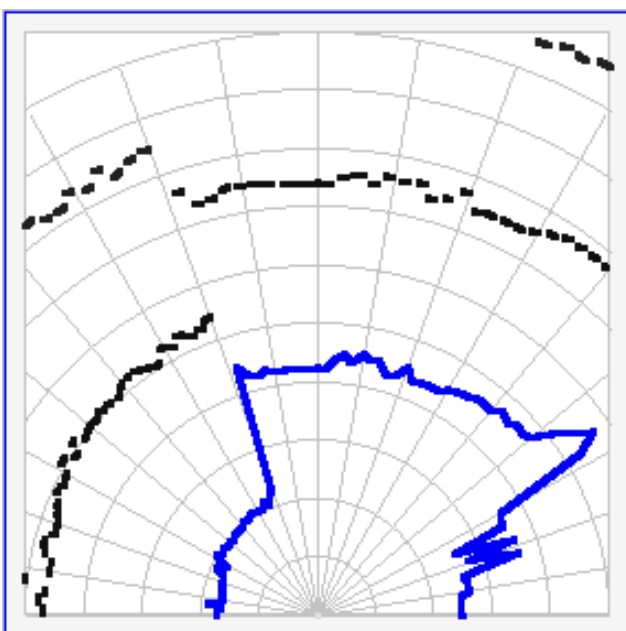
Se puede considerar como un zoom o como una indicación de que no nos parece relevante aquello que se encuentre más allá del valor de *Depth*.

Al indicar a la aplicación que genere el Mapa Polar lo irá trazando a la vez que toma los valores del sonar. Mostrará en rojo el primer eco recibido y en negro-gris los siguientes ecos recibidos dentro de la distancia límite indicada por *Depth*. Los ecos serán de un tono grisáceo más claro cuanto más lejanos sean.

En el siguiente gráfico se muestra un mapa polar obtenido en una habitación cerrada llena de múltiples objetos que producen ecos múltiples. Se ha seleccionado un *Depth* máximo para poder mostrar el máximo número de ecos.



A continuación se muestra una imagen más detallada de una habitación con algunos objetos (muebles) a los lados.



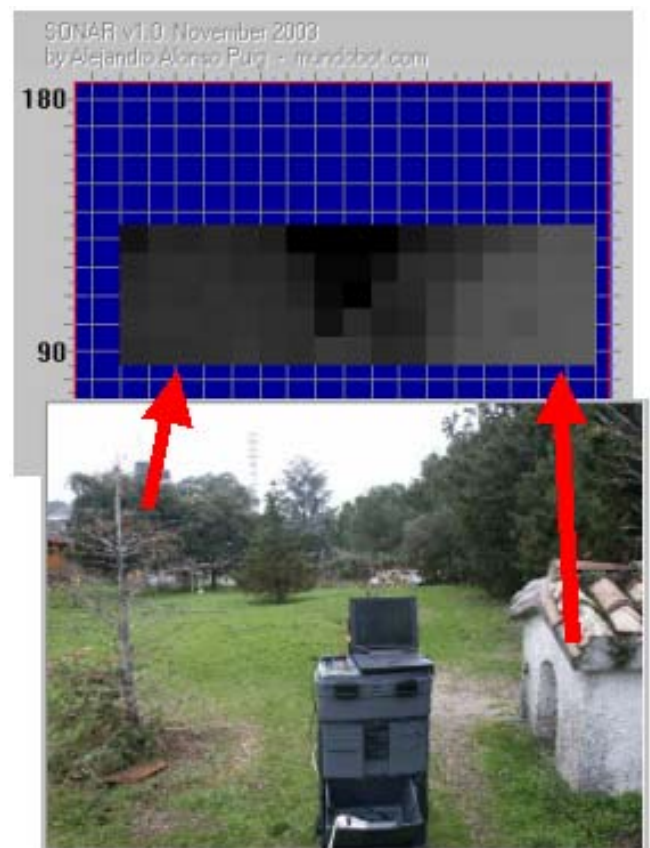
Mapas de Profundidad

Los mapas de profundidad son “fotografías” bidimensionales de un área de barrido. Para generar dicho mapa el sensor va recorriendo la coordenadas comprendidas entre ($Hmin$, $Vmin$) y ($Hmax$, $Vmax$). Las medidas tomadas se van representando en diferentes tonalidades de gris para mostrar la distancia al obstáculo. Tonos más claros son objetos más cercanos y tonos más oscuros son objetos más lejanos.

El valor de profundidad (*Depth*) indica la distancia máxima que se desea medir. En el caso de mapas de profundidad este valor será el que se establezca como límite de distancia y por tanto las medidas mayores que este valor se representarán con color negro.

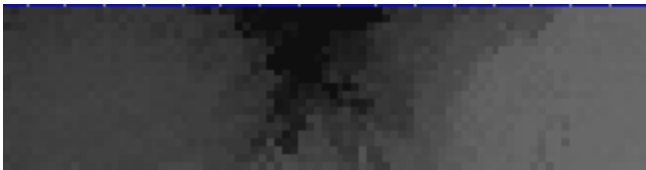
El factor de resolución sigue el mismo concepto explicado en el caso de mapas polares.

A continuación se muestra un ejemplo de mapa generado mediante un barrido horizontal de 160° y vertical de 40° correspondiente a un exterior.



Como puede verse, cualquier objeto es representado como un borrón debido a la amplitud del haz ultrasónico. La caseta es reconocida cercana y por ello la tonalidad del borrón es más clara. El árbol seco (a la izda.) por otra parte parece más lejano puesto que su carencia de hojas provoca un reflejo menor del haz ultrasónico.

A continuación se muestra un mapa de profundidad obtenido ante el mismo entorno, pero con una mejor resolución (Factor Resolución=2).



El pequeño borrón en la parte central inferior representa reflejos del suelo.

Diseño de la aplicación

La aplicación se ha generado en Visual C++ 6.0. Utiliza una serie de librerías para el control de la tarjeta K8000 de Velleman y poder así utilizar su bus I²C, al que se encuentran conectados los demás elementos del trabajo.

Para más información sobre el manejo de la mencionada librería puede consultarse el trabajo del autor existente en la dirección <http://mundobot.com/tecnic/ecoi2C/ecoi2c.htm>

El interface con el usuario consiste básicamente en un cuadro de diálogo asociado a una clase CsonarDlg dentro de la cual se encuentran todas las funciones y variables utilizadas para la toma de datos y presentación gráfica.

Los ficheros fuente de la aplicación contienen numerosos comentarios explicativos y pueden ser accedidos en la dirección: <http://mundobot.com/tecnic/Sonar/Sonar.zip>

No obstante a continuación explicaremos la estructura general de la aplicación.

Las direcciones que se han utilizado para los dispositivos I²C se han establecido en las siguientes variables:

```
cSRF08Add=224; //I2C address SRF08 module  
cSVI2CAdd=118; // I2C address Servo controller device
```

En el caso del SRF08, esa es su dirección I²C por defecto. En el caso del SVI2C, ha sido necesario situar los switches dip adecuadamente para obtener esta dirección, para que no entrase en conflicto con las direcciones de otros componentes que forman parte de la tarjeta K8000.

Las funciones de esta aplicación para el manejo de los dispositivos I²C son:

void CSonarDlg::SonarInit()

Inicializa el módulo svi2c para que los servos funcionen con valores adecuados de ángulos. Esto se lleva a cabo enviando al módulo mencionado los comandos I²C adecuados.

void CSonarDlg::SonarPos(int iHpos, int iVpos)

Envía los comandos I²C adecuados al módulo svi2c para mover los servos a las posiciones especificadas por *iHpos* e *iVpos* en grados.

void CSonarDlg::DistCheck(int iDist[16])

Ordena por I²C al módulo SRF08 hacer una medición de distancia tomando el valor de los 17 ecos y dejándolo en la matriz de parámetro. Un valor de cero indica que el objeto estaba muy alejado y el eco no regresó.

Basado en estas funciones y dependiendo de los valores de las variables introducidas por el formulario se generan los gráficos con las funciones siguientes:

void CSonarDlg::DrawPolarGrid()

Presenta el área de trazado llamando a la función *CSonarDlg::DrawPolarGrid()* y genera el mapa polar a base de ir moviendo los servos en los rangos especificados mediante llamadas a la función *CSonarDlg::SonarPos(int iHpos,*

int iVpos), tomando las medidas de todos los ecos y almacenándolas en memoria dinámica con punteros a estructuras mediante llamadas a la función *CSonarDlg::DistCheck(int iDist[16])*. Según va tomando y almacenando estos datos los va mostrando en pantalla.

void CSonarDlg::DrawDepthGrid()

Presenta el área de trazado llamando a la función *CSonarDlg::DrawDepthGrid()* y genera el mapa de profundidad a base de ir moviendo los servos en los rangos especificados mediante llamadas a la función *CSonarDlg::SonarPos(int iHpos, int iVpos)*, tomando las medidas de todos los ecos y almacenándolas en memoria dinámica con punteros a estructuras mediante llamadas a la función *CSonarDlg::DistCheck(int iDist[16])*. Según va tomando y almacenando estos datos los va mostrando en pantalla.

En realidad aunque tome todos los valores de todos los ecos, solo representará en

pantalla el valor del primer eco (objeto más cercano).

La estructura dinámica de memoria está basada en punteros a estructuras declarados de la siguiente forma:

```
struct stMeasure
{ int iMeasure[16];
};
struct stMeasure *pstDataRange,
                *pstDataPoint,
                *pstPrevMeasure;
```

La generación del espacio de memoria dinámica se hace en la función llamada en el momento de pulsar los botones de generación de mapa polar "*CSonarDlg::OnPolarMap()*" o de profundidad "*CSonarDlg::OnDepthMap()*". Previamente en las mismas funciones se destruye el espacio de memoria que se hubiese generado anteriormente mediante una llamada a la función *CSonarDlg::FreeMemory()*. También se llamara a esta función en el momento de salir del cuadro de diálogo y por tanto del programa mediante la inclusión de una llamada en el destructor de la clase "*CSonarDlg::~CSonarDlg()*"

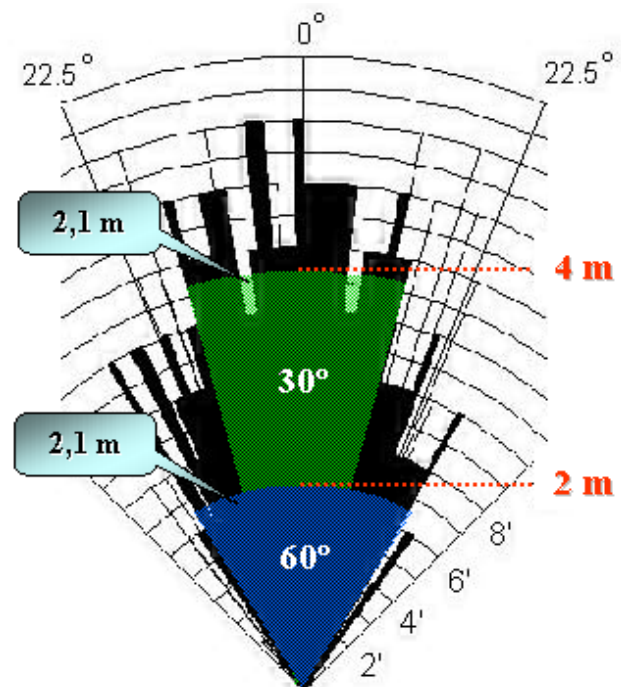
Conclusiones y líneas futuras

El trabajo desarrollado no ha pretendido más que ser una introducción al manejo de las mediciones ultrasónicas de distancia.

Las medidas tomadas y los mapas mostrados son simples, esto es, sin tratamientos avanzados de la señal para aumentar la precisión de los mismos.

La forma del haz (beam pattern) permite hacernos una idea del área detectada por el haz. Al no generar un haz cónico perfecto, podemos encontrarnos con situaciones como la de la figura siguiente, en la que la anchura del haz a cuatro metros es la mitad de la anchura del haz a dos metros, por lo que la longitud del arco (zona detectada) en ambos casos es la misma (2,1m) según la fórmula trigonométrica:

$$\text{Arco} = \text{ángulo}^\circ \cdot (\pi/180) \cdot \text{distancia}$$



Esto significa que si el sonar se encuentra instalado en un robot en posición de 90°, el haz incidirá contra el suelo y si se inclina por encima de los 90°, detectará objetos por encima del robot, lo cual puede ser un problema.

En cualquier caso se ha verificado que colocando espuma plástica en la base del transductor se reduce la emisión del haz hacia la zona en sombra. Utilizando este método se puede evitar la detección del suelo en caso deseado.

Como ejemplo de dificultad de detección se tiene el caso de una puerta abierta. El marco a un metro

puede impedir que “veamos” que la puerta está abierta ya que el haz (de unos 1,4 m de anchura) es más ancho que el paso de la puerta y por tanto se verá rebotado en el marco.

Existen transductores con un haz más estrecho que el utilizado y que por tanto permitirán mayor precisión selectiva.


Por otra parte en ciertos casos se analizan los ecos producidos más allá del primero para la detección de puertas abiertas.

Bibliografía y enlaces de interés



[1] Módulo ultrasónico SRF08. Características técnicas:

-  Superrobotica: <http://www.superrobotica.com/S320112.htm>
-  Devantech: <http://www.robot-electronics.co.uk/htm/srf08tech.shtml>
-  Acroname: <http://acroname.com/robotics/parts/R145-SRF08.html>
-  FAQs: http://www.robot-electronics.co.uk/htm/sonar_faq.htm
-  Module Schematic: <http://www.robot-electronics.co.uk/images/srf08schematic.gif>
-  Module Software in C: <http://www.robot-electronics.co.uk/files/srf08.c>
-  Transducer datasheet: <http://www.robot-electronics.co.uk/datasheets/t400s16.pdf>

[2] Módulo SVI2C para control de servos

-  Diseño de un módulo para control de 8 servos y 5 canales analógicos por bus I2C. Alejandro Alonso Puig: <http://mundobot.com/tecnica/Svi2c/Svi2c.htm>



[3] Comunicaciones I²C desde PC

-  Comunicación bidireccional por bus I2C entre un PC y un microcontrolador PIC de gama media actuando como servidor de eco. Alejandro Alonso Puig: <http://mundobot.com/tecnica/ecoI2C/ecoi2c.htm>
-  Tarjeta interface KI2C para comunicaciones entre Puerto Paralelo y Bus I2C: <http://mundobot.com/tecnica/KI2C/KI2C.htm>

[4] Comunicaciones I²C

-  Página de Philips: <http://www.semiconductors.philips.com/buses/i2c/>

[5] Otros trabajos

-  Ultrasonic Sonar Rangers: Some Practical problems and how to overcome them:
<http://didactical.physics.helsinki.fi/~aohamala/FP2002/poster.pdf>
-  Sonar imaging using a single SRF08 sonar mounted on a servo arm. Torfi Fran:
<http://this.is/torfi/robotblog/index.php?getInfo=1&projectID=32>