



4
2ej-

**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**APLICACION DE UN TRANSDUCTOR ULTRASONICO PARA
MEDICION DE VOLUMENES EN CONTENEDORES
DE SECCION CONOCIDA.**

TESIS PROFESIONAL
Que para obtener el Título de
INGENIERO MECANICO ELECTRICISTA
p r e s e n t a
EDUARDO JOEL AGUILAR SANCHEZ



Director: ING. NICOLAS CONTRERAS GOMEZ

México, D. F.

1992

**TESIS CON
FALLA DE ORIGEN**



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

| | | |
|------|---|-----|
| I. | Introducción. | I.1 |
| II. | El sonar. | 1 |
| | Características de las ondas sonoras. | 1 |
| | Historia del sonar. | 5 |
| III. | Circuitos del sonar. | 8 |
| | Características del circuito sintonizado. | 8 |
| | Circuito Transmisor-Receptor. | 12 |
| | Transductor Ultrasónico. | 16 |
| IV. | Arquitectura del Microprocesador. | 21 |
| | Microprocesador Z80. | 21 |
| | Decodificación de memoria y puertos. | 23 |
| | Circuito Multipuertos Z80 KIO. | 25 |
| | Display de cristal líquido LCD. | 30 |
| V. | Interfase del sistema analógico y digital. | 36 |
| | Conformación de la señal entre el CPU y el sonar. | 36 |
| | Sincronización del microprocesador y el sonar. | 38 |
| VI. | Software. | 41 |
| | Modulo de adquisición de datos. | 41 |
| | Modulo de proceso de información. | 42 |
| | Modulo de despliegue de información. | 45 |
| VII. | Disposición de los elementos. | 49 |
| | Tarjeta de circuito impreso CPU. | 49 |
| | Tarjeta de circuito impreso sonar. | 49 |

INDICE

| | |
|--|----|
| Conclusiones. | 56 |
| Apéndices. | 58 |
| A. Programa en lenguaje ensamblador. | 58 |
| B. Generación de tabla de distancias. | 85 |
| C. Generación de tabla de volúmenes. | 90 |
| D. Manual del MUV. | 93 |
| Bibliografía. | 97 |

Todos los aparatos electrónicos que cotidianamente sin pensarlo encendemos y apagamos, tienen consigo un gran trasfondo y no imaginamos la gran cantidad de ingenieros que estuvieron largas horas de su vida creando. Antes de entrar en materia quisiera hacer una breve justificación de lo que aquí trate de plasmar.

Esta tesis habla del diseño del medidor ultrasónico de volúmenes pero por supuesto no establece cada paso que se dio para llegar al diseño final, esto es que como cualquier investigador debe suponer, tuve que realizar muchos ensayos antes de llegar a esto y aunque el escrito por momentos parezca breve hay un gran trasfondo de investigación para la realización del sistema. La meta que nos fijamos fue desarrollar un sistema, es decir, hacerlo y que sea útil, ya que se creo la ingeniería para realizar cosas que ayuden a la sociedad y no solo dejarlas en la mente, por supuesto, apoyadas totalmente en la física y las matemáticas, pero explotando la parte creativa de cada ser.

Hasta ahora he convivido con alumnos, profesores, compañeros, y estoy convencido de que el ingeniero busca la realización de las cosas, pensar en que la obra de un grupo de personas esta ahí y se puede plasmar, da algún servicio y que poco a poco ha de mejorarse.

Es muy importante para todos nosotros universitarios que estuvimos o estamos en la Facultad de Ingeniería, pensar que con el esfuerzo potencial que cada uno va poniendo día a día, se realiza mas y mejor tecnología Mexicana y por ello la meta fundamental que nos fijamos es precisamente concretar, con lo cual debe entenderse todo un procedimiento metódico y ordenado que a la luz de la sociedad reditue un buen servicio.

Introducción

En la época actual en que el auge de la electrónica ha ocasionado una revolución completa a los sistemas cotidianos, es donde los sistemas de medición tienden a realizarse por otros medios, los cuales deben arrojar las características de facilidad de manejo, uso, economía y precisión.

Puede verse la revolución que han sufrido los automóviles. Por ejemplo cuando en la década de los sesenta creían que eran lo máximo con solo sistemas mecánicos y al hacer el motor mas grande se podía obtener mayor potencia, sin importar el rendimiento; es en la década de los ochentas cuando cobra mayor importancia el ahorro. Se generan nuevas alternativas siendo aquí donde la electrónica hace su aparición como algo ya no solo de lujo sino fundamental en el rendimiento y distribución de energía. Así un automóvil que cuenta con un microprocesador que gobierna la correcta inyección de combustible, paso de aire, sustancias auxiliares, y tantas señales que se pueden controlar, es capaz de hacer sumamente eficiente a un motor, generando maquinas pequeñas con potencias muy grandes.

Es ahora cuando el interés por hacer más eficiente el uso de muchos productos naturales como energéticos y agua entre otros, nos lleva a pensar en herramientas electrónicas, que fundamentadas en algún microprocesador tengan funciones inteligentes y de control.

Aquí es donde sistemas nuevos de medición permiten mayor eficiencia en el uso y una mejor distribución, consecuentemente aprovechamiento.

Debe ser muy grato pensar que México, una nación llamada país en vías de desarrollo pueda crear tecnología mejor que en otros países y sobre todo mas barata.

El medidor ultrasónico de volúmenes, cuyo diseño se establece en esta tesis, es un resultado de todo el pensamiento anterior y establece las bases para seguir con la investigación en esta área y quizá mas adelante no solo sea capaz de medir recipientes de sección conocida o de tipo tanque como las de los trailers, vacíos y llenos o distancias, sino que también pueda controlar por ejemplo: sistemas de canales abiertos, flujo en tuberías o una gama tan grande como la mente del investigador pueda abarcar.

Características de las ondas sonoras. Ecos y Rebotes

Las ondas sonoras en el aire o en el agua son ondas longitudinales de compresión y rarefacción, esto es en analogía con un resorte helicoidal, que la velocidad de las partículas de cada una de las espiras es paralela a la dirección de propagación, lo cual se conoce como onda longitudinal. Las ondas cortantes son aquellas que se producen dentro de una porción de un medio sólido como una barra de acero por ejemplo.

Solamente ondas longitudinales pueden propagarse dentro de un medio fluido; las ondas cortantes no pueden propagarse porque un fluido no puede transmitir esfuerzo transversal apreciable de una capa a la siguiente.

La velocidad de las diferentes clases de ondas depende, en general, de las propiedades del medio y de la magnitud o amplitud de la perturbación de la onda misma. Pulsos de presión muy grandes en el aire se propagan mas rápidamente que las perturbaciones pequeñas. Las perturbaciones pequeñas se propagan con una velocidad que depende solamente de las propiedades del medio.

Entonces el sonido se define haciendo referencia a ondas de pequeña amplitud y la velocidad del sonido en un fluido esta dada por: $v = (dp/d\rho)^{1/2}$; donde la derivada describe la variación de la presión con la densidad bajo condiciones tales que no hay suficiente tiempo para que ocurra en flujo de calor apreciable, de las regiones de ondas que están comprimidas a las que están en rarefacción.

Reflexión

Cuando una onda sonora llega a un medio de frontera muy diferente como una pared, un risco o por superficies acuosas, la onda que llega no puede reproducir la secuencia del movimiento de compresión rarefacción que venia teniendo, el pulso que llega "incidente" es regresado desde este extremo de la frontera como un pulso semejante invertido en fase y viajando en dirección opuesta. Al regreso de una onda incidente se le llama reflexión y es lo que comunmente conocemos como ecos.

Estos ecos se rigen por las leyes básicas de la reflexión y transmisión de las ondas vibratorias.

La ley de la reflexión nos dice que el ángulo de incidencia de la onda debe ser igual al ángulo de reflexión. Tal reflexión, debida a una superficie lisa se conoce frecuentemente como especular.

Una superficie lisa es aquella en la cual las dimensiones de las irregularidades de la superficie son mucho más pequeñas que la longitud de onda.

Si las dimensiones de las irregularidades son del mismo orden de magnitud que la longitud de onda, las ondas reflejadas son incoherentes, reflejadas en forma difusa o dispersa.

Superposición o Interferencia

Otro aspecto del comportamiento ondulatorio se presenta en la manera en que pulsos, viajando en diferentes direcciones, pueden literalmente pasar uno a través de otro sin ningún efecto apreciable en su forma y propagación, esto es lo que se denominaría superposición de las ondas. Las ondas sonoras también se superponen viajando una a través de la otra sin alteración.

Ahora bien si hacemos que dos trenes periódicos con la misma forma de onda y de la misma frecuencia se superpongan mientras viajan en la misma dirección puede ocurrir interferencia que básicamente puede ser de dos tipos:

Interferencia constructiva: Las señales se superponen cresta a cresta y valle con valle. La onda resultante tiene el doble de amplitud de los trenes de onda (señales) individuales.

Interferencia destructiva: Los trenes de onda son superpuestos cresta con valle, en fase opuesta. La onda resultante tiene amplitud nula.

Dispersión

El fenómeno de la variación de la velocidad de la propagación de ondas con la frecuencia de estas se llama dispersión, y un medio que presenta esta propiedad se conoce como un medio dispersor. El aire presenta muy poca dispersión de ondas sonoras para el amplio rango de frecuencias audibles.

Ondas audibles, Ultrasónicas e Infrasonicas

Las ondas sonoras están restringidas a los límites de frecuencias que puede estimular el oído y el cerebro humano, dando la sensación de sonido. Estos límites son aproximadamente de 20 Hz a 20 KHz y constituyen los límites audibles.

Una onda mecánica longitudinal, cuya frecuencia sea inferior al límite audible, se llama onda infrasonica, y una onda mecánica longitudinal cuya frecuencia sea superior al límite audible se denomina onda ultrasónica.

Las altas frecuencias correspondientes a las ondas ultrasónicas se pueden producir mediante vibraciones elásticas de un cristal de cuarzo inducidas por resonancia por un campo eléctrico alterno que se aplique (efecto piezoeléctrico). Es posible producir frecuencias ultrasónicas hasta 6×10^8 Hz por este procedimiento; la longitud de onda correspondiente en el aire es de 5×10^{-5} cm igual que la longitud de onda de las ondas luminosas visibles.

La velocidad de propagación de las ondas sonoras como se mencionó antes, es dependiente del medio en el que estén viajando; a continuación se presentará una pequeña tabla.

Velocidad con respecto al medio de propagación.

| Medio | Temperatura °C | Velocidad | | |
|------------------|----------------|-----------|---------|----------|
| | | Mts/seg | Pie/seg | |
| Aire | 0 | 331.3 | 1087 | |
| Hidrógeno | 0 | 1286 | 4220 | |
| Oxígeno | 0 | 317.2 | 1041 | |
| Agua | 15 | 1450 | 4760 | |
| Plomo | 20 | 1230 | 4030 | |
| Cobre | 20 | 3560 | 11 700 | |
| Hierro | 20 | 5130 | 16 800 | |
| Granito | | 6000 | 19 700 | Valores |
| Goma Vulcanizada | 0 | 54 | 177 | extremos |

Y esta velocidad es la misma para todas las frecuencias.

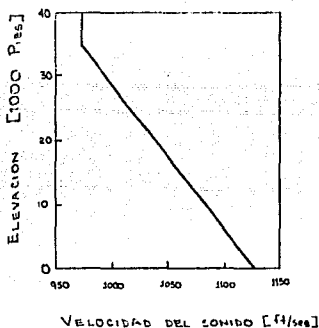
Supersónico y Subsónico

Cabe hacer mención de algunos vocablos más, relacionados con las ondas sonoras, solo a manera de aclaración.

Los términos supersónico y subsónico son vocablos familiares, se refieren a las velocidades que son mayor y menor que la velocidad del sonido, respectivamente. Aquí intervienen otros conceptos como el número de Mach = V/C , es decir que la razón de la velocidad local del flujo entre la velocidad local del sonido, sin tomar en cuenta la frecuencia de esta señal sonora.

En la siguiente figura aparecerá una gráfica de la velocidad del sonido con respecto a la elevación del nivel, es decir partiendo del nivel del mar (elevación cero).

Se puede observar que la velocidad del sonido en aire, bajo condiciones estandar, varia desde 1120 pies/seg al nivel del mar, hasta 968 pies/seg en el intervalo de altitudes que va de 36,000 a 40,000 pies



Para referir la velocidad del sonido de acuerdo a la temperatura podemos utilizar la siguiente ecuación.

$$C_t = C_0 (1 + (T/273))^{1/2}$$

Donde:

- Ct. Es la velocidad del sonido a una temperatura específica.
- C0. Es la velocidad del sonido a 0 °C
- T. Es la temperatura en °C.

Por ejemplo la velocidad del sonido a 25 C variará de 331.4 a 346 m/s.

La historia del sonar

La historia del sonar comienza (como muchas de las tecnologías actuales) en las aplicaciones bélicas, ya que se tenía la necesidad de detección y orientación de objetos y esta era fundamentalmente cubierta con el radar que utilizaba ondas electromagnéticas, solo que para naves (buques, barcos, etc) en el agua no es posible utilizar el radar pues las ondas electromagnéticas son propicias en el aire no así en el agua, sin embargo las señales sonoras como las que utiliza el sonar son buenas viajeras en un medio como del que hablamos.

La palabra sonar se deriva de la frase inglesa SOUND NAVIGATION AND RANGING que literalmente significa navegación y alcance por sonido, aunque una traducción mas apropiada a la utilización que tiene este dispositivo podría ser "ayuda a la navegación por determinación de distancias obtenidas por el sonido".

Así el sonar fue utilizado por primera vez durante la segunda guerra mundial para la localización de submarinos. La existencia del cono de silencio, situado en la vertical del proyector, fue uno de los grandes obstáculos con que contó en sus comienzos este equipo.

Efectivamente existe una zona bajo el proyector que no podía ser explorada. Posteriormente fue resuelto este problema con el sistema llamado del "haz abierto".

Este sistema utiliza una excitación del proyector que obliga a utilizar una pequeña superficie de radiación, con lo que el haz que se produce es muy abierto, permitiendo la exploración en la misma vertical del proyector.

Otros grandes problemas relacionados con el sonar son la velocidad de propagación de las ondas ultrasónicas en el agua, las reflexiones y las refracciones que experimentan estas ondas (que en general se van curvando hacia abajo), los ruidos debidos a los buques próximos.

Para el primero solo queda la solución de utilizar tablas de corrección que dan valores en función de la temperatura del agua y la salinidad, así como comprobar periódicamente la distancia eficaz real de alcance del sonar.

Esto último se hace en función de buques (objetos) de los que se sabe su distancia exacta al proyector.

Para evitar las perturbaciones debidas al ruido submarino se desarrolló el modulador de barrido, que hace variar linealmente la frecuencia de cada impulso emitido con lo que permite distinguirlo del ruido de fondo.

La velocidad del buque también impone limitaciones en los sectores de exploración del sonar, por ejemplo, a velocidades mas altas de 25 nudos aparece la imposibilidad de arriar el proyector.

A otras velocidades existen sectores donde la exploración es mala. la zona de proa solo se puede explorar a velocidades menores de 15 nudos, hasta 120 a velocidades mayores de 20 nudos.

Pero no solo se han utilizado a bordo de buques, también se ha utilizado el sonar en zonas cerca de la costa, depositado el proyector sobre el fondo marino para la localización de minas submarinas y buques enemigos, todo ello con bastante éxito.

La técnica utilizada para el ataque submarino con el sonar, consiste en obtener mediante este equipo los datos de distancia y demora del submarino, con lo que se puede calcular su velocidad y rumbo.

La aplicación mas importante del sonar en tiempo de paz ha sido en las actividades pesqueras, determinando la profundidad de un blanco submarino (bancos de peces etc).

Otro uso del sonar, en su forma de utilización "escucha" (solo recibiendo señales) permite el mantenimiento de comunicaciones, utilizando para ello el manipulador. La ventaja es que solo puede ser escuchado por otros buques con equipos similares.

Los microprocesadores han invadido también el campo del sonar. Su aplicación fundamental se halla en el análisis del entorno que permite determinar el nivel de ruido propio, así como las condiciones ambientales de temperatura, etc.

Otra de las facetas en las que interviene el microprocesador es en el almacenamiento de memoria de imágenes, su despliegue posterior en pantalla y la persistencia de las mismas.

También en el seguimiento de blancos interviene el microprocesador así como en la medición de los parámetros de los objetivos y la transmisión de estos a otros puntos del buque.

El procesamiento permite distinguir entre los ecos prácticamente puntuales de los submarinos y los ecos debidos a reverberaciones en el fondo marino o la superficie del mar, con lo que el operador queda emplazado a seguir únicamente los ecos "verdaderos" o reales.

La tendencia actual del uso del sonar es en la detección de niveles o medición de flujos de tuberías; esta última aplicación del sonar en el aire es la que se conocería como un radar sónico o "sonic radar" a lo que se nombraría de una manera mas específica como Sodar, que fundamentalmente es la medición de distancias en el aire.

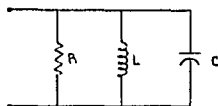
Características del circuito tanque LC

Es evidente que al hablar acerca de un circuito transmisor receptor de señales, se está íntimamente ligado con un circuito de tipo tanque LC que esté sincronizado a la frecuencia que se desee trabajar (en dependencia del transductor), es por esto que se hablará un poco, acerca de tal circuito.

Aquí el objetivo es realizar un oscilador lineal que genera ondas senoidales esencialmente por un fenómeno de resonancia.

Un circuito esta, o entra, en resonancia, cuando la tensión aplicada y la intensidad de corriente que circula están en fase. En resonancia, pues, la impedancia compleja del circuito, se reduce exclusivamente a una resistencia para R.

Si tenemos el siguiente circuito sintonizado:



En donde R puede ser la impedancia que presente el transductor a la frecuencia de sintonía (resonancia).

Entonces la admitancia compleja del circuito paralelo de la figura es:

$$Y = G + j (\omega C - 1/\omega L) = G + jB$$

$$\text{donde } B = B_C - B_L$$

Y las susceptancias son $B_C = \omega C$ y $B_L = 1/\omega L$

El circuito entra en resonancia cuando $B=0$ es decir:

$$\omega C = 1/\omega L$$

de donde se obtiene:

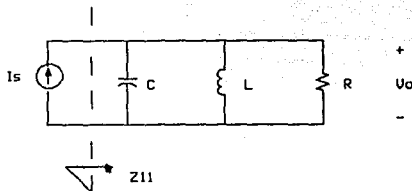
$$W = 1 / (LC)^{1/2} = W_0 \quad [\text{rad/seg}]$$

o bien

$$f_0 = 1 / [2 * \pi * (LC)^{1/2}] \quad [\text{Hz}]$$

Teoría del circuito sintonizado.

Sea el siguiente circuito.



Para encontrar la ecuación de transferencia.

$$Z_{11}(S) = V_o(S) / I_s(S) \quad ; \text{ impedancia de transferencia}$$

$$Z_{11}(S) = R \parallel SL \parallel 1/SC$$

$$Z_{11}(S) = [(R/SC) / (R + 1/SC)] \parallel SL = [(R/SCR+1)] \parallel SL$$

$$Z_{11}(S) = [SL(R/(SCR+1))] / [SL+R/(SCR+1)]$$

$$Z_{11}(S) = (1/C) S / (S^2 + S/RC + 1/LC)$$

Que nos representa a un circuito con un cero en el origen y 2 polos.

Ahora para revisar la estabilidad (recordemos que existe una oscilación permanente) veamos a los polos.

$$P_{1,2} = -(1/RC) \pm [(1/RC)^2 - 4/LC]^{1/2}$$

$$= -1/2RC \pm [((1/RC)^2 - 4/LC) / 4]^{1/2}$$

Entonces:

Reales si $(1/RC)^2 > 4/LC$; sobreamortiguado

$$R < [(L/C)^{1/2} / 2]$$

Y no existirá una frecuencia a la cual el circuito resuene, es decir, su comportamiento será mas bien pasivo.

Polos iguales.

Transición Si $2R = (L/C)^{1/2}$; críticamente amortiguado

$P_{1,2} = -1/2RC$ y tampoco existirá oscilación.

Polos complejos conjugados.

$$R > [(L/C)^{1/2}/2] ; \text{ subamortiguado}$$

Y en este caso si es posible obtener una frecuencia a la cual se anule la impedancia capacitiva y la inductiva para que el sistema se comporte e forma puramente resistiva.

Y partiendo del sistema

$$Z_{11}(S) = LS / [(S^2 / \omega_0^2) + (2\zeta / \omega_0)S + 1]$$

Tenemos que

$$\omega_0 = 1 / (LC)^{1/2} \quad \text{y es la frecuencia de resonancia}$$

$$\zeta = LC / 2R$$

Y como habíamos dicho

$\zeta > 1$ sobreamortiguado

$\zeta = 1$ críticamente amortiguado

$\zeta < 1$ subamortiguado

Y el factor de calidad Q que es la energía máxima almacenada entre la energía máxima disipada por periodo es

$$Q = R / \omega L = 1 / 2\zeta$$

Nos indica la selectividad del tanque, y el ancho de banda es:

$$BW = \omega_0 / Q$$

Nos indica que si es muy selectivo el ancho de banda debe ser pequeño.

Circuito Transmisor-Receptor

En los circuitos de transmisión y recepción de pulsos sonoros, se decidió utilizar como base al circuito integrado de national semiconductors LM1812, el cual proporciona todas las características deseadas.

De entre estas características la mas importante que se explotará para el diseño, es que dentro de este mismo circuito se realizan las dos funciones primordiales; es decir, es capaz de transmitir y después conmutarse para operar como receptor y así mediante algunos elementos (redes) externas obtener el circuito transmisor receptor.

El LM1812 es un transmisor ultrasónico de propósito general. Este chip contiene un transmisor clase C (modulado por pulsos). Un receptor de alta ganancia un detector de modulación por pulso y un circuito de rechazo de ruido.

Para la frecuencia a la cual se desea que trabaje el sistema, esto es, tanto el transmisor como el receptor, es necesario instalar una red de tipo LC sintonizada a dicha frecuencia.

Cabe destacar que debido a esta característica es necesario un solo transductor, el cual se utiliza para la transmisión y la recepción y que oscila a una sola frecuencia, el cual se detallará posteriormente.

Hasta frecuencias de 325 KHz el transmisor puede manejar salidas de hasta 1 Amper, es decir 12 Watts de pico.

La sensibilidad de la ganancia del receptor puede llegar a ser de 200 V_{p.p}. El circuito de detección incluye un chip que es capaz de rechazar impulsos de ruido.

El LM1812 es un circuito que se puede alimentar hasta con 18V y consume 40 mA, sin embargo nosotros lo polarizaremos con 12 Volts por ser mas común.

El consumo de corriente detectado en nuestras pruebas es de alrededor de 25 mA, es decir, muy pequeño, ya que se podrá esperar que la tarjeta digital consuma alrededor de 200 mA. Esto significa que del consumo total del sistema general esta sección tan solo necesitará alrededor del 10% .

El transmisor

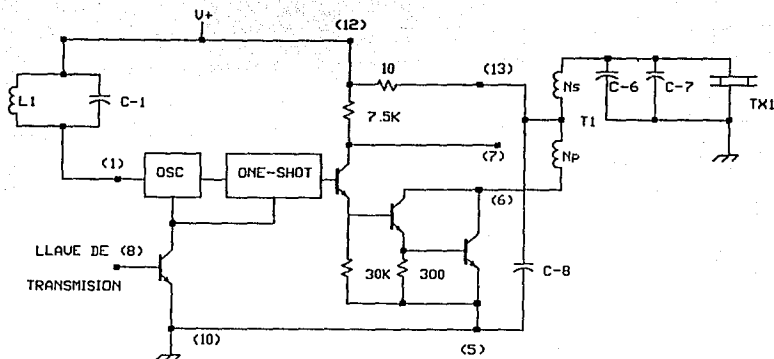


FIGURA 2.1

El transmisor (figura 2.1) consta de un oscilador, un one shot de $1\mu\text{s}$ y un amplificador de potencia.

El circuito es switchado en la forma de transmisor, cuando en el pin ocho se aplica un pulso positivo, el circuito tanque compuesto por L1 y C-1 es establecido en modo de oscilador.

Para el circuito de encendido, el one-shot de $1\mu\text{s}$ es disparado cada vez que el oscilador cubre un ciclo, y este circuito maneja al amplificador de potencia. El one-shot tiene un tiempo de reset de $2\mu\text{s}$ limitando con esto a una frecuencia máxima de alrededor de 325 KHz.

Un transformador acopla el transductor a la salida de esta etapa. La frecuencia de oscilación del tanque es fijada por L1 y C-1 como se estableció anteriormente.

$$f_o = 1 / 2\pi(L1C-1)^{1/2} \quad \dots (1)$$

Como sabemos, de entre estos dos componentes fundamentales, el menos comercial en cuanto a variedad de sus valores es la inductancia.

CIRCUITOS DEL SONAR

Entonces podemos conocer el capacitor a partir de algún valor fijo de inductancia como sigue:

$$C = [1 / (2\pi f_0)]^2 / L \quad \dots (2)$$

El transductor que se utilizará es uno de masa cuya frecuencia central de oscilación es de 215 KHz. Una inductancia comercial es por ejemplo la PCH-27-684 de Coilcraft, tiene un valor de $680\mu\text{H}$ por lo que entonces, utilizando la ecuación (2). $C-1=805\text{ pf}$, que se puede componer en forma practica por dos capacitores $C-1=330\text{pf}$ y $C-2=470\text{ pf}$.

Si comprobamos la frecuencia de oscilación del circuito tanque con estos valores reales de los componentes, con la ecuación (1) tenemos que $f_0=215.784\text{ KHz}$ pero el transductor nos presenta un ancho de banda de 2.5 kHz por lo que no se tienen problemas con esta etapa.

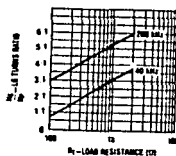
El circuito tanque $L1, C-1$ tiene que presentar una mínima resistencia R_p de $10\text{ k}\Omega$, donde $R_p = 2\pi f_0 Q L$ y Q es la Q sin carga del circuito tanque $L1, C-1$ y la podemos encontrar: $Q = \omega_0 L / r_{\text{ind}}$ (es la misma Q del inductor pues no tiene carga). De datos del manual del fabricante tenemos $Q=25$.

Entonces:

$$R_p = 2\pi(215\text{k})(25)(680\mu) = 22.9\text{ k}\Omega \text{ y sobrepasa a la } R \text{ mínima}$$

Interfase del transductor con el oscilador

A la salida, el diseño del transformador se puede establecer con ayuda de la siguiente gráfica.



Para una dada resistencia de carga se obtiene una cierta relación de vueltas del transformador.

Para no exceder las especificaciones de voltaje de pico a pico del transformador, este puede ser ajustado mediante la siguiente ecuación:

$$VR_p = 2V + (N_s/N_p) \quad \dots \quad (3)$$

De las hojas de especificaciones del transductor elegido, la impedancia que presenta este, a la frecuencia de 215 KHz.

Es de 0.4 K Ω entonces de la gráfica anterior para una $R_t=400\Omega$ tenemos una relación de vueltas $N_s/N_p = 4.5 : 1$ y el voltaje a la salida del transformador $V_{p,p}=28$ según la ecuación (3).

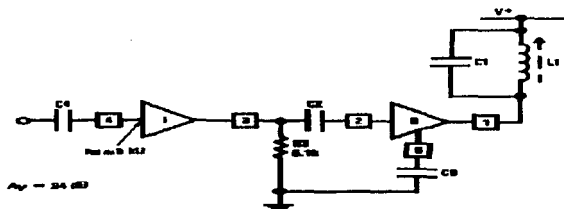
Para asegurar que la etapa de la salida no esta sobrecargada se tiene que hacer una medición de corriente en el pin 6. Los primeros pulsos quizá tengan picos de hasta 2 o 3 amperes pero en estado estable dichos picos no excederán 1 Amp. Los picos de corriente se pueden atenuar si se reduce la relación de vueltas del transformador T1.

El secundario de dicho transformador T1 se sintoniza junto con C-6 a la frecuencia de operación f_0 . De ahí que si elegimos al transformador T1 como el transformador f5672-A = SD250-4 de coilcraft cuya inductancia del secundario es de 4.5 mH, y si la capacitancia C-6 se descompone en la suma de dos, es decir, dos capacitancias en paralelo, entonces el valor de estas será de 100 pf correspondiente a C-6 y C-7 de 22 pf.

El control de circuito LM1812 como transmisor o receptor, es con el pin ocho. Cuando esta pata del C.I. está en alto el circuito se comporta en el modo transmisor, mientras que cuando está en bajo, dicho pin el modo receptor se establece.

Receptor

El receptor contiene dos etapas de ganancia como se ilustra



Es necesaria una protección para evitar los picos de corriente a que sean menores que 50 mA. Hay que recordar que en este caso es el mismo transductor el que se utiliza para la transmisión y para la recepción. Como el voltaje que aplicamos al transductor es menor que 200 Vp-p, un capacitor cuya reactancia a la frecuencia de operación sea de 5 k Ω es una protección adecuada.

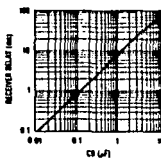
Para esto se utiliza C-4 de 220 pf.

$$X_c = 1/WC \quad ; \quad w = 2\pi f$$

$$\Rightarrow C_{min} = 1/WXc = 2/[1.35 \times 10^6 (5K)] = 148 \text{ pf}$$

El circuito tanque compuesto por L-1 y C-1, C-2 está compartido con el transmisor, estando siempre a la misma frecuencia de sintonía.

La ganancia de voltaje de la segunda etapa esta dada por $A_v = (Q/70) * (L1/C1)^{1/2}$; donde Q es nuevamente la Q sin carga de circuito tanque. (Esta ganancia es aproximadamente 900 para el caso). Esta segunda etapa está apagada cuando el circuito esta en el modo de transmisor. La segunda etapa no se enciende de inmediato cuando se conmuta al modo receptor, pero si se enciende después de un pequeño retardo introducido por el capacitor C-5. Este retardo da oportunidad de blanquear al receptor, dando el tiempo al transductor para que deje de resonar. Para calcular C-5 nos auxiliamos de la siguiente gráfica donde C-5 es una función del retardo.



C-5=100 nf, entonces el retardo es aproximadamente 1 ms.

Transductor ultrasónico

Un transductor es un elemento que interactúa con el medio ambiente es decir, es la parte que cambia de señal eléctrica a señal mecánica para así poder obtener un efecto físico palpable.

Comúnmente relacionamos ultrasonidos con altas frecuencias, y estas con la parte superior del espectro de audio por lo que relacionamos a un tweeter que es una bobina para agudos capaz de reproducir frecuencias acústicas muy "altas" (del orden de 30 KHZ). En forma practica si no se supera este limite de frecuencia estas bocinas (transductores) funcionan muy bien, pero si la frecuencia que se necesita generar es superior, la solución es con transductores cerámicos fabricados especialmente para ultrasonido.

Estos transductores cerámicos generalmente se conocen como piezotitas como a cualquier cristal piezoeléctrico. Tienen dos características muy bien definidas.

a) Si se someten a vibraciones mecánicas o sonoras, pueden generar en sus extremos una tensión (como los micrófonos piezoeléctricos).

b) Si en sus extremos se aplica una tensión, empiezan a vibrar, generando una frecuencia. Prácticamente se comportan como cualquier cuarzo de transmisión cuando es excitado por una etapa osciladora.

A diferencia de los tweeter, que funcionan en una banda ancha de frecuencia los transductores cerámicos funcionan en una banda muy estrecha.

Por lo tanto si tomamos un transductor capaz de funcionar a 215 KHZ funcionará también dentro de una gama de frecuencias comprendida entre 212 KHZ y 218 KHZ pero con un rendimiento distinto.

De acuerdo a la frecuencia de resonancia y a su aplicación, la composición química de la piezotita varia por ejemplo:

Titanato de bario

Titanato de plomo

Titanato de plomo + Titanato de circonio.

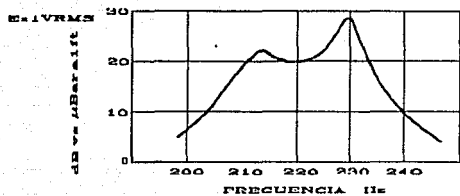
El transductor utilizado para este diseño es el modelo E-188/215 de Massa Products Corporation. Es un transductor ultrasónico para aire y tiene alta resolución. Opera a una frecuencia de 215 KHZ y además presenta la característica de ser sumamente direccional.

La característica de direccionalidad se refiere básicamente a que emite una señal que solo se esparce a unos cuantos grados (narrow beam) esto permite eliminar todos los lóbulos secundarios, de emisión, lo que lo hace ideal para detectar rebotes de señales hacia objetos específicos.

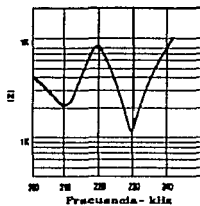
Tiene las siguientes características de respuesta.

CIRCUITOS DEL SONAR

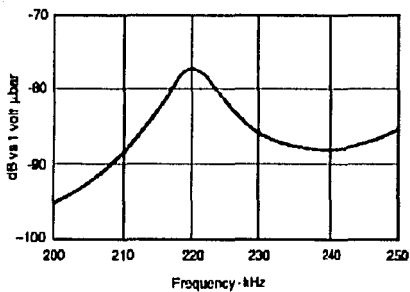
RESPUESTA DE TRANSMISION



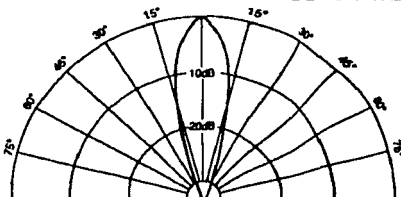
IMPEDANCIA



RESPUESTA DE RECEPCION



RESPUESTA DIRECCIONAL



Si se desea utilizar algún spot (cono) acoplado al transductor este va a depender directamente del ángulo de esparción de las ondas α (α = beam angle), y es definido de manera similar a lo que sería la frecuencia de corte en un filtro ya que es el ángulo total entre los puntos a los cuales, la potencia del sonido ha decaído a la mitad de su valor y es equivalente a 3 dB abajo del máximo.

Para calcular el diámetro del spot se puede utilizar la fórmula:

$$D=2*R*\tan(0.5\alpha) \text{ , donde}$$

D=diámetro del spot en plg

R=rango de la distancia del blanco

α = ángulo total de incidencia en grados.

La resolución de sistemas ultrasónicos es mejorada si la longitud de onda es más pequeña. Como sabemos la longitud de onda decrementa cuando la frecuencia es más alta $\lambda = C/f$; donde λ = longitud de onda, C = velocidad del sonido, f = frecuencia. De aquí que para una aplicación de alta resolución es bueno utilizar transductores con la más alta frecuencia posible como en este caso.

Como el sonido viaja a través del aire, la energía se atenúa más rápidamente si la frecuencia es aumentada. La máxima atenuación teórica para ultrasonido (arriba de 200 KHz) se puede calcular con la fórmula: $a_{\text{máx}} = f*10^{-2}$; donde $a_{\text{máx}}$ = atenuación máxima en dB/ft y f = frecuencia en KHz.

Entonces utilizando esta ecuación, la energía del sonido para el transductor de 215 KHz se atenuara a una máxima de 2.15 dB/ft, cuando viaje a través del aire. La atenuación quizá sea menor, dependiendo de la humedad, pero, no es sencillo calcularla.

Para la alta frecuencia de trabajo también presenta otra ventaja y es que la presencia de ruido es menor. Por lo tanto el transductor ultrasónico de alta frecuencia puede trabajar mejor aun en ambientes contaminados con ruido.

El ángulo de dispersión (Beam angle) también ayuda en la lucha contra el ruido, pues el área de sensibilidad del transductor queda definida por dicho ángulo.

La función del transductor se ve afectada también por factores ambientales, tales como temperatura, la cual ya fue comentada anteriormente, y fundamentalmente tiene un efecto de aumentar la velocidad de propagación del sonido, causando con esto malas lecturas en el tiempo de viaje de la señal por aire.

Las turbulencias del aire y las corrientes de convección también afectan, pues varían la temperatura. La presión atmosférica, se vera reflejada como pequeños efectos en la precisión de la medición.

La humedad sin embargo, no afecta significativamente la operación de un sistema de medición ultrasónico. La interferencia acústica que existe en el ambiente también puede afectar, sin embargo, como es un transductor de una frecuencia mas bien alta no es tan sensible al ruido. La interferencia por radio frecuencia (RFI) puede evitarse con una buena soldadura del transductor al sistema.

Los Blancos

Todos los blancos cercanos a la fuente de ultrasonido producen eco para ser detectado. Algunas texturas de materiales producirán ecos de menor tamaño. La reflexión generalmente es función de la frecuencia, así a altas frecuencias objetos porosos tienen buena reflexión, mientras que no es así a bajas frecuencias.

Un blanco que en forma relativa sea mas grande que la longitud de onda del sonido, producirá un eco mayor que un blanco mas pequeño. El blanco debe ser mas grande que aproximadamente 10 longitudes de onda transversalmente. Por lo tanto pequeños blancos son mejores detectados con sonidos de altas frecuencias. El sensor debe estar apuntando al objeto para tener un mejor eco.

Microprocesador Z80

Para la etapa inteligente del sistema medidor ultrasónico de volúmenes, se utiliza el microprocesador Z80, como parte fundamental, auxiliado de sus periféricos para lo cual contamos con el circuito multipuertos Z80 KIO del que hablaremos mas adelante.

Algunas de las razones que se tienen para usar este microprocesador son:

- Es un μP muy conocido y de fácil adquisición en el mercado.
- La capacidad de direccionamiento es de 64K bytes de memoria y 256 puertos (capacidad mas que suficiente para este sistema)
- El costo es bajo y además es un μP muy versátil.

El microprocesador Z80 de Zilog, presenta las siguientes características.

Tamaño de la palabra 8 bits. Posee 22 registros disponibles para usos en programas, 18 registros de 8 bits y 6 registros de 16 Bits, los que se pueden dividir en dos grandes grupos:

Registros de propósito general: Es un doble set de registros. Registros principales y registros alternativos. Los seis registros principales son B,C,D,E,H y L. Son los de propósito general porque se pueden usar de cualquier manera, o también como tres registros pares BC,DF y HL para datos o información de 16 bits.

Registros de propósito especial: Son registros especiales para ser usados como acumuladores de resultados de operaciones aritméticas y lógicas, para direccionamiento de memoria y también en forma interna. Estos registros son:

- A -Acumulador
- F -El registro de banderas (flag)
- SP-Stack Pointer
- PC-Program Counter
- IX-Registro índice
- IY-Registro índice
- R -Refresco de memoria
- I -Vector de interrupciones.

En el Z80 CPU (Central Processing Unit) las manipulaciones aritméticas y las operaciones lógicas se tratan como 8 bits simultáneos en la unidad lógica Aritmética (ALU). La ALU se comunica internamente con los registros del procesador central. La unidad lógica aritmética realiza las siguientes operaciones:

ARQUITECTURA DEL MICROPROCESADOR

Desplazamiento, a izquierda o derecha, prueba de bits, incremento y decremento, operaciones lógicas y suma y resta.

El registro de instrucciones y control del procesador central, retiene el contenido de la posición de memoria direccionada en la memoria por el program counter (PC) y se carga durante el ciclo de búsqueda de cada instrucción. La unidad de control del procesador central ejecuta las funciones definidas por la instrucción en el registrador de instrucciones y genera todas las señales de control necesarias para transmitir los resultados a los registros adecuados.

Este microprocesador posee un bus de direcciones de 16 bits de A_0-A_{15} , que proporcionan la dirección correspondiente a intercambios de datos de memoria hasta 64 K bytes; y a los intercambios de datos de dispositivos de entrada-salida (puertos). En el direccionamiento de puertos se utilizan los ocho bits de dirección inferiores para permitir al usuario seleccionar directamente hasta 256 puertos de entrada o salida.

El bus de datos bidireccional en este microprocesador es de 8 bits (D_0-D_7) y se utiliza para intercambio de datos con la memoria y los puertos.

El Z80 CPU puede ejecutar 158 instrucciones independientes incluyendo las 78 instrucciones del 8080A, las cuales están divididas en las siguientes categorías:

-Cargas de 8 bits -Carga de 16 bits -Cambios de transferencia de bloques y búsqueda -Aritméticas de 8 bits y operaciones lógicas -Aritmética de propósito general y control del CPU - Operaciones aritméticas de 16 bits -Rotaciones y desplazamientos -Pruebas de bits y operaciones -Saltos -Llamadas y retornos -Operaciones de entrada y salida.

El microprocesador ejecuta las instrucciones mediante un proceso que tiene una secuencia específica de operaciones. Lectura o escritura de memoria. -Entrada o salida a puertos, lectura o escritura. -Interrupt acknowledge. El CPU muestra la señal de interrupción con el flanco del último ciclo del reloj, al final de cualquier instrucción.

Cuando una interrupción es aceptada un ciclo especial $M1$ es generado. Durante este ciclo $M1$, $IORQ$ va a activa (en vez de $MREQ$) para indicar que el dispositivo de interrupción se puede establecer en un vector de 8 bits en el bus de datos. El CPU automáticamente adiciona dos ciclos de espera a este ciclo.

Decodificación de Memoria y Puertos

Para la parte digital del sistema, es evidente la necesidad de la utilización de una memoria RAM y una ROM, así como de puertos de entrada y salida al sistema. Cabe mencionar la utilización de un circuito de reloj (temporizador) para la correcta sincronía de todos estos elementos.

Es muy importante el correcto manejo del reloj y su ajuste pues es parte fundamental en el conteo de tiempos de terminación y recepción. Este reloj se basa en el uso de un cristal cuyo valor es de 3.579545 MHz y para ello se exploto la característica que presenta el circuito multipuertos Z80-KIO del que se comentara mas ampliamente en la siguiente sección.

MEMORIA EPROM

En este tipo de memoria está residente el software que maneja el Hardware.

Para conectar una memoria EPROM directamente al bus de control multiplexado, debe de tener habilitación de las salidas. La habilitación de salida es controlada por la señal de lectura de memoria del sistema. La selección del dispositivo o decodificador de memoria para habilitarlo o deshabilitarlo se logra usando la señal MREQ del bus de control.

Cuatro parámetros deben ser evaluados cuando se determina la compatibilidad de las Eproms en un sistema con microprocesador.

TOE.- Tiempo de habilitación de salida para tener datos validos.

TACC.-Tiempo de direccionamiento par accesar un dato valido.

TCE.-Tiempo de selección del circuito integrado para obtener un dato valido.

TOF.-Señal de habilitación de salida alta para flotación de las salidas o desconexión. Lógica del circuito del bus común.

La memoria EPROM seleccionada para este sistema con reloj de 3.57 Mhz presenta un tiempo de acceso de 420 ns. En el sistema para poder realizar una lectura deben transcurrir alrededor de 3 ciclos de reloj con lo cual podemos asegurar que estamos dentro de los limites de acceso, sin necesidad de utilizar un circuito de retardo para lectura de la memoria EPROM.

Memoria RAM

De igual manera para conectar la memoria ram, debe de existir alguna habilitación en su pata de chip select para lo cual es importante la señal de solicitud de memoria del CPU MREQ.

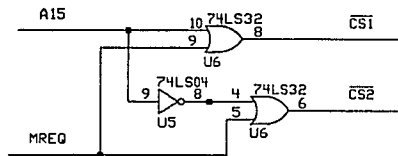
Los parámetros típicos de las memorias ram estáticas son:

- TW- Duración del pulso de escritura
- TWR-Levantamiento de la señal de escritura (tiempo que se sostiene la dirección desde que terminó la escritura)
- TOW-Tiempo de traslape de los pulsos de lectura y escritura.
- TDH-Dato sostenido desde el termino de la escritura
- TAW-Dirección valida al final de la escritura.
- TCW-Selección del circuito final de la escritura.
- TASW-Dirección valida al inicio de la escritura.

La memoria RAM utilizada es una 6116, que es un arreglo de memoria de 2K bytes, cada Byte por supuesto de 8 bits. Una memoria relativamente pequeña y que cumple con los requerimientos del sistema.

Cabe mencionar que el acceso a las memorias es también fundamentado de acuerdo al estado de la señal de Read (RD) o de Write (WR) del CPU.

Como solo son necesarias dos memorias, la implementación de la decodificación se hace con compuertas, apoyado en las señales de dirección del CPU como se muestra:

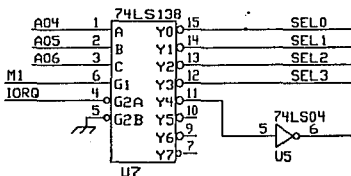


Contando cada señal de CS con un margen de hasta 32 Kbytes.

Para la decodificación de puertos se utiliza la configuración propuesta por el Z80-KIO, debido a que como es un circuito multipuertos sigue una decodificación interna que debemos respetar; El display de cristal liquido de cuarzo se utiliza como otro puerto, habilitado con la pata de enable.

ARQUITECTURA DEL MICROPROCESADOR

Para este caso si se utiliza un circuito decodificador por que otorga la posibilidad de dejar mas lineas para otros puertos posibles, es por esto, que se utilice el 74LS138 de la siguiente manera:



Es importante mencionar que la línea de control del μP IORQ junto con la señal de M1 nos dan el parámetro para saber si el microprocesador desea hacer una entrada de los datos de algún puerto o bien alguna salida hacia este.

El circuito multipuertos Z80 KIO

Como ya se hizo mención, la parte digital del sistema además de contar con el microprocesador Z80 CPU como base, incorpora la aplicación del nuevo circuito multipuertos Z80 KIO. Es un circuito de alta escala de integración que cuenta con todos los periféricos compatibles con el Z80 CPU y de poco uso hasta ahora.

La razón fundamental de utilizar este circuito como base para realizar la comunicación del CPU con el ambiente del usuario, es que basta con él, para poder contar con todos los periféricos necesarios.

Después de analizar la problemática que nos depara el medidor ultrasónico de volúmenes, se entiende que serán fundamentales dos puertos para el correcto funcionamiento del sistema. Estos puertos deben ser el puerto paralelo de entrada y salida Z80 PIO para adquisición de información tanto del usuario como del sistema mismo (parte analógica); y el circuito temporizador contador Z80 CTC para la coordinación de tiempo, en intervalos de transmisión y recepción, así como la parte mas importante, que es el conteo del tiempo que tarde la señal en realizar su viaje por el aire, entre el transductor y el contacto con el objetivo, de lo cual se tomara la distancia. También del Z80 KIO se explotara su característica de tener su propio generador de reloj.

El Z80 KIO es un circuito que tiene la presentación de paquete PLCC con 84 pins. Utiliza 15 registros internos para datos e información de programación.

Unidad lógica Temporizador-Contador. CTC.

Esta unidad provee al usuario de cuatro canales de ocho bits. Los temporizadores o contadores pueden ser programados por el CPU. Este circuito se utiliza básicamente como contador y por su característica de ser de ocho bits cada canal cuenta hasta 256, por lo que se ponen dos contadores en cascada para así tener un conteo de hasta 65535 numero suficiente de acuerdo al reloj, con que se cuenta para arrojar mediciones de tiempo que se transforman en distancia.

Cada canal-temporizador o contador (según programación) posee un byte de preescala (cuando se usa en modo timer). Cada canal tiene también su propia entrada de disparo o reloj, para cuantificar el proceso, el cual se utiliza para realizar la conexión en cascada que se menciona arriba, tiene también una salida para indicar el cruce por cero de la cuenta.

Basta un solo vector de interrupción para esta unidad pues de manera automática internamente genera su vector de interrupción para cada canal.

La lógica de control de canal, recibe una palabra de control de canal de 1 byte para programar a los canales contador/temporizador. La lógica de control del canal decodifica la palabra de control y establece las siguientes condiciones de operación.

- Habilita o deshabilita interrupciones.
- Modo de operación (timer/counter).
- Factor de preescala en modo timer (16 o 256).
- La forma de la entrada para el CLK/TRG. (alto o bajo).
- Disparo de modo temporizador (automático o entrada por CLK/TRG).
- Sigue palabra de constante de tiempo.
- Reset por software.
- Palabra de control o vector.

Cuando el canal contador/temporizador es programado recibe un registro de constante de tiempo en un byte el cual, puede ser cualquiera de 1 hasta 256 (el cero es el 256). Esta constante es cargada automáticamente en un contador a cero (decrementador), cuando el canal contador-temporizador se inicializa, y subsecuentemente después de cada cuenta por cero.

ARQUITECTURA DEL MICROPROCESADOR

En el modo temporizador (timer) se usa una preescala, que divide la frecuencia del sistema de reloj por un factor de 16 ó 256. Esta preescala da un efecto de multiplicación del período de reloj por 16 o 256.

El contador puede ser decrementado de dos formas dependiendo del modo de operación: mediante la salida de la preescala (modo timer) o mediante un pulso de disparo en la entrada CLK/Trg (modo contador).

Sin interrumpir la cuenta el CPU se puede leer la cuenta que permanece en cualquier momento, tal como se hace para el sistema MUV (medidor ultrasónico de volumen).

Como se puede observar en el apéndice uno, el canal uno y el dos se programan como contadores.

En ambos modos contador o temporizador (timer) la salida es un pulso activo alto, cuando el contador decremента a cero.

En el modo contador, el pulso de entrada en CLK/TRG decremента el contador, el disparador es asíncromo, pero el conteo es sincronizado con el CLK. Para que un decrememento ocurra en el siguiente flanco de reloj, el disparador (trigger) tiene que preceder al CLK por un tiempo mínimo. Si este tiempo de mando, es menor que el especificado, el conteo se retrasa por un ciclo de reloj. El pulso de disparo, tiene que tener un ancho mínimo, y el período de disparo tiene que ser un poco menos que dos veces el período de reloj.

Para realizar la tabla fundamental de distancia del sistema MUV, se deben tener las siguientes consideraciones:

Para la máxima resolución, posible del sistema hay que tomar en cuenta el período mínimo de reloj que se tiene y que pueda ejecutar sus instrucciones el sistema.

$CK=3.579545$ MHz con el flip-flop que da forma a la señal, nos queda 1789772.5 Hz, que es la frecuencia de trabajo del sistema.

Entonces el período $T=558.7302297$ nS ≈ 0.558702 μ S. Ahora, si se pretende una resolución de 0.1 cm es decir $1\text{mm} = 1 \times 10^{-3}$ m, el tiempo debe ser $t = d/V = 3.02$ μ S tomando como base la velocidad del sonido a condiciones normales.

De este resultado se puede ver que la necesidad estriba en que el CTC cuente cada 3 μ S.

ARQUITECTURA DEL MICROPROCESADOR

Ya que el reloj del sistema que se diseña aquí presenta un período, bastante menor como se demostró hace unos instantes, se puede soportar como garantía de trabajo hasta cinco ciclos de reloj. Pero como se mencionó anteriormente el CTC, no siempre puede contar a la velocidad de trabajo, así, demostrar si dicho circuito, alcanza una cuenta cada $3\mu\text{s}$ que correspondería a una resolución de 1mm , es de la siguiente manera:

$$\text{CK} \times \text{P} \times \text{T} = \text{Intervalo de cuenta}$$

donde

CK=clock
P=preescala 16,256
T=cte de tiempo de 1 a 256

Por lo que el intervalo mínimo de cuenta, utilizando la preescala y la constante de tiempo señalados es de $8.9396 \mu\text{s}$ que nos arroja una resolución de 2.5mm

Unidad lógica de entrada-salida en paralelo. PIO .

El circuito multipuertos KIO, contiene en su haber un circuito semejante al Z80 PIO el cual es un puerto de entrada o salida de datos en paralelo.

Este circuito se utiliza básicamente para enlazar al CPU con el sistema físico que le rodea. En este caso se utiliza el puerto A, para que sea quien gobierne al oscilador del sonar y el puerto B, para utilizarse como teclado y recibir los datos del usuario. Además es también este puerto quien da el reporte al CPU de haber alcanzado un rebote el sonar.

Esta unidad de KIO provee interfases compatibles con circuitos TTL o CMOS y el CPU através del uso de un puerto paralelo de 8 bits. El CPU configura la lógica para interfasar un ancho rango de dispositivos periféricos sin lógica externa.

Es común interfasar con dispositivos típicos como lo son impresoras, memorias PAL y teclados, tal como se le da uso aquí.

En realidad tiene dos puertos paralelos, el puerto A y el puerto B de 8 bits cada uno tal como el Z80PIO. Estos dos puertos tienen varios modos de operación. Modo entrada, salida, Bidireccional y modo bit. Cada puerto tiene 2 señales de protocolo RDY y STB, las cuales pueden ser utilizadas para la transferencia de datos. La señal RDY (ready) indica que los puertos están listos para la transferencia de datos, mientras que la señal STB (strobe). Es una entrada al puerto que indica cuando la transferencia de datos ha ocurrido.

ARQUITECTURA DEL MICROPROCESADOR

Cada uno de los puertos puede también ser programado para interrumpir al CPU, por encima de las condiciones generadas de status y genera un vector único de interrupción cuando el CPU responde.

Para programar estos puertos paralelos (PIO), de manera semejante, nuevamente hay que enviar una o mas palabras de control, así, como un vector de interrupciones si es lo que se desea.

La palabra del vector de interrupción. El PIO es diseñado para trabajar con el CPU en el modo de interrupción 2. Esta palabra tiene que ser programada si se van a utilizar interrupciones, y el bit cero tiene que ser cero.

El usuario provee al $V_7V_6V_5V_4V_3V_2V_1V_0$ vector de interrupciones. identifica al vector de interrupciones.

La palabra de control de modo. Selecciona el modo de operación del puerto. Esta palabra se puede escribir en cualquier momento.

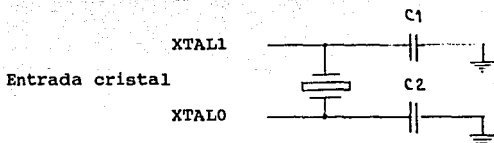
| | | | |
|--------------------|----------------|---------|--|
| Selecciona el modo | $D_7D_6D_5D_4$ | 1 1 1 1 | Identifica la palabra de control de modo |
| 0 0 | modo 0 | | |
| 0 1 | modo 1 | | |
| 1 0 | modo 2 | | No importa |
| 1 1 | modo 3 | | |

Cuando se selecciona el modo de trabajo, Modo 3, la palabra de control tiene que ser seguida por la palabra de control de registro de entrada-salida. Esta palabra configura los registros de entrada-salida, que definen que líneas del puerto son de entrada y cuales de salida. Un uno lógico indica entrada, mientras que un cero indica salida.

Para explotar estas características del PIO, se programa al PIO-A en el modo cero de salida y el PIO-B en el modo 3 modo Bit.

Unidad Lógica del Controlador Del Reloj Oscilador.

Esta unidad es otra característica mas del circuito multipuertos KIO, esta etapa proporciona el reloj a todo el sistema. Dicha unidad aceptara un cristal, o un resonador cerámico o un reloj compatible con TTL, y genera una señal de reloj de salida. Para el circuito MUV se utiliza un cristal cuya frecuencia de oscilación es de 3.57 MHz como se hizo mención en la sección del CTC, ya que es lo que se recomienda, así como el valor de 2 capacitores de 33 pf cada uno.



Finalmente se puede ver que las direcciones de los registros internos en el KIO están dadas por la siguiente tabla:

| | |
|-------------|-------------------------|
| Registro 0 | Pio puerto A de datos |
| Registro 1 | Pio puerto A de control |
| Registro 2 | Pio puerto B de datos |
| Registro 3 | Pio puerto B de control |
| Registro 4 | CTC canal 0 |
| Registro 5 | CTC canal 1 |
| Registro 6 | CTC canal 2 |
| Registro 7 | CTC canal 3 |
| Registro 8 | SIO canal A datos |
| Registro 9 | SIO canal A control |
| Registro 10 | SIO canal B datos |
| Registro 11 | SIO canal B control |
| Registro 12 | PIA puerto C datos |
| Registro 13 | PIA puerto C control |
| Registro 14 | KIO comandos |
| Registro 15 | Reservado |

El registro de comando del KIO se usa para programar los reset por software y para configurar la prioridad interna del daisy chain este registro se programara antes que todos. Los bits de control del reset que están momentáneamente en 1, pulsaran una señal interna de reset para el dispositivo correcto.

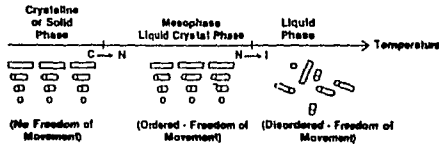
Display de cristal liquido de cuarzo. LCD.

Se comentara ahora un poco, acerca de los displays supertwisted-nemáticos.

El termino nemático, se refiere al estado liquido intermedio entre el estado cristalizado y el estado amorfo en el cual las moléculas están orientadas en una dirección dada.

Los display de cristal liquido "Twisted Nematic", hacen uso de las propiedades eléctricas y ópticas de un grupo específico de moléculas orgánicas, las cuales exhiben un estado intermedio "mesofase".

Mientras en la mesofase estas moléculas tienen amplio rango de propiedades de ordenamiento, similar a aquellas que existen en cristalinos de estado sólido, también exhiben algunas libertades de movimiento como ocurre en los líquidos isotrópicos.



El rango de esta mesofase o "cristal líquido", es comprendido por el estado, que es definido por temperaturas de transición: la temperatura de transición de cristalina a nemático y la temperatura de transición de nemático a isotrópico.

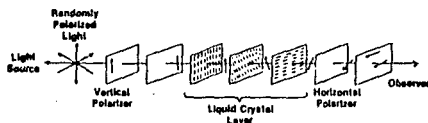
El término "Twisted Nematic" se refiere a una organización molecular específica, donde todas las moléculas dentro de un plano particular, están con el eje mayor apuntando en la misma dirección. La dirección de la moléculas puede ser inducida a girar, por la adición de compuestos colestéricos.

La transición de nemático a isotrópico es muy drástica y dan rangos de 30 °C a 90 °C, dependiendo del tipo de material de cristal líquido usado.

La temperatura de transición de cristalino a nemático, define la temperatura de fundición teórica del material, pero no necesariamente determina el límite mas bajo de temperatura de operación. Esto es debido a que hay muchos materiales de cristal líquido "superfríos", es decir, como la temperatura decrece, baja la temperatura de transición de cristalino a nemático, y la transición no ocurre. Una vez "congelado" en el estado cristalino, (el cual generalmente requiere exposición a temperaturas menores de -60 °C). La temperatura de transición de cristalino a nemático, tiene que ser rebasada, antes de que el material regresara al estado de cristal líquido. Es este hecho el que permite que opere bien bajo su temperatura de transición de cristalino a nemático.

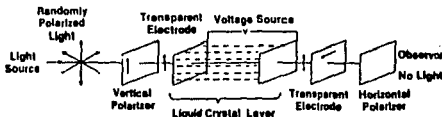
Puesto que la viscosidad del material aumenta, la temperatura disminuye, por lo que las moléculas son menos libres al movimiento a temperaturas bajas. Esto es un factor, el cual usualmente determina el límite inferior para operación práctica.

Los cristales líquidos, Twisted Nematic, tienen la habilidad de rotar la polarización de la luz, cuando pasa a través de los niveles moleculares. Muchos displays de Cristal líquido hacen uso de un "virador" de 90° posicionado entre los electrodos conductores y las líneas de polaridad cruzadas tal como se muestra.



La luz aleatoriamente polarizada entra al dispositivo, solamente la luz de polarización vertical es habilitada para pasar a través del frente polarizador. La polarización de la luz es rotada 90° cuando pasa a través del espejo horizontal polarizador. Si una fuente reflectora se establece atrás del espejo polarizador, la luz pasará atrás, a través de la celda, con su polarización obtenida, siendo rotada de nuevo 90° por la capa de cristal líquido.

Si un voltaje es aplicado a los electrodos transparentes, un campo eléctrico es establecido en la capa de cristal líquido. Puesto que las moléculas tienen algún grado de libertad de movimiento y un dipolo de momento eléctrico, esto permite la estructura twisted-nematic (nemático-virado) y una alineación con el campo como se muestra.



En este estado, la luz entrante pasa las capas de cristal líquido, pero no se polarizarán. La polarización vertical de la luz no es transmitida mediante el espejo polarizador horizontal, y así la extinción de la luz ocurre.

El reflejo límite de la extinción es definido muy abruptamente por la muestra del electrodo patrón y el resultado visual es un área oscura con un fondo de reflexión de luz. Mediante el uso de polarizaciones paralelas en vez de cruzadas, el formato inverso se obtiene; es decir, áreas activas aparecen con luz y el fondo negro.

En un display típico de efecto de campo twisted-nematic, los electrodos transparentes de óxido de indio-titanio, son modelados en substratos de vidrio, usando técnicas fotolitográficas estándares, y el electrodo común está cubierto por una capa delgada para encadenar la realización del dispositivo.

Cada plato, es cubierto con un polímero de revestimiento alineado, el cual es entonces estirado para proveer el efecto propio de la orientación de la superficie, es decir los 90° de giro de la estructura del cristal líquido. Un conductor por encima provee una señal de ruta entre el electrodo común y la unión de contacto común, la cual elimina la necesidad de hacer contacto de dos superficies diferentes.

Un sello de polímero, contiene el material de cristal líquido y mantiene los substratos de vidrio juntos. La polarización y reflexión está limitada al ensamblaje.

El display utilizado en este desarrollo, es de la marca Shelly (SLM 11601) el cual es del tamaño de una línea y 16 caracteres ya que con esto es posible mandar al usuario suficiente información, tanto para la entrada de datos, como para el despliegue de los resultados.

Para el control de display es necesario conectarlo al bus correspondiente y mandar una señal de R/W (read,write) una de control para instrucción o datos, y por supuesto habilitar al display E (enable) que para el caso está decodificado como un puerto mas, es decir, directamente se accesa a él con una simple escritura a su dirección específica (salida del 74138); solo que a través de un inversor, pues es con activo alto.

Este display que es de tipo super twisted, necesita una polarización negativa pequeña por lo que es conveniente introducir el potenciómetro marcado como POT1 de 20KΩ el cual se conecta a la fuente de -5V como se puede ver en el diagrama esquemático general.

ARQUITECTURA DEL MICROPROCESADOR

Para la programación del LCD existe una secuencia específica que da el fabricante. Dicha programación se basa en introducir una rutina de espera de la señal de bandera del display pues este trabaja a frecuencia bastante menor que el CPU (270 KHz). Esta programación como se puede ver en el apéndice A (en el archivo son-disp.asm) prepara dos rutinas, una "Transf-aviso" que pone cada uno de los bits del letrero deseado al buffer y la otra Refresh-LCD que manda todo lo deseado a que aparezca en pantalla.

Estas dos rutinas están preparadas de esta manera por que se utilizan cada vez que sea necesario mandar algún letrero. Para mandar letreros sin información mayor que la del mismo mensaje (por ejemplo "espere" "oprime" etc). Se creó la subrutina pinta, que contiene a las dos ya mencionadas todo esto con el fin de hacer al programa mas amigable tanto para el usuario, así como también, para quien programa y sobre todo para ahorrar espacio en memoria.

Se presentan aquí algunas instrucciones del LCD. Siempre estarán en el mismo orden es decir, primero RS y R/W y posteriormente los ocho datos.

RS, R/W, DB7, DB6, DB5, DB4, DB3, DB2, DB1, DB0,
Clear Display 0 0 0 0 0 0 0 0 0 0 1

Esta instrucción limpia todo el display y regresa el cursor a la posición de inicio (dirección cero)

Entry Mode Set 0 0 0 0 0 0 0 1 1/D S

Establece la dirección del movimiento del cursor y especifica o no el corrimiento del display.

1/D = 1 Incrementa 1/D = 0 Decremento
 S = 1 Corrimiento del display

Display ON/OFF 0 0 0 0 0 0 1 D C B

Establece el control de encendido o apagado del display (D), encendido o apagado del cursor (C) y parpadeo del cursor en la posición del caracter (B).

D = 0 display apagado D = 1 Display encendido
 C = 0 cursor apagado C = 1 cursor encendido
 B = 0 parpadeo apagado B = 1 parpadeo encendido

Function Set. 0 0 0 0 1 DL N F * *

Establece la interfase de la longitud de los datos (DL), el numero de líneas del display (N), y la fuente del caracter (F)

ARQUITECTURA DEL MICROPROCESADOR

| | | | |
|--------|---------------|--------|---------------------|
| DL = 1 | 8 bits | DL = 0 | 4 bits |
| N = 1 | 2 líneas | N = 0 | 1 línea |
| F = 1 | 5 X 10 puntos | F = 0 | 5 x 7 puntos (dots) |
| * | = No importa | | |

Set DD RAM Address 0 0 1 A_{DD}

Establece la dirección de los datos de la ram del display. El dato de la DDRAM es enviado y recibido, después de este establecimiento.

Read Busy y Address 0 1 BF AC

Lee la señal de Busy flag, indicando si una operación interna esta siendo rendida, y lee el contenido de la dirección del contador (busy flag es la señal de que ya terminó alguna operación).

BF = 1 Operando internamente BF = 0 puede aceptar una instrucción

Write Data to CG or DDRAM 1 0 Datos escritos

Escribe datos en la ram del display o en la ram de generación de caracteres.

Para la generación de letreros el display cuenta con una ram de generación CG-RAM ; la que se puede explotar mandando en un byte el número en hexadecimal que corresponda al carácter que se desea exhibir, todos los cuales están en la tabla ascii que se conoce, haciendo incapié de que son hexadecimales, así por ejemplo el 0 (cero) es 48 en la tabla ascii el cuál corresponde al 30 en hexadecimal.

Aquí se explota una característica más del paquete ensamblador de avocet, que es el que se utiliza para todo el software del medidor ultrasónico de volúmenes, esta característica es que basta escribir el letrero como se quiera mostrar para que cuando se ensamble haga la conversión de tal letrero a hexadecimal y lo ponga consecutivamente en una serie de bytes de la EPROM.

Conformación de la señal entre el sonar y el microprocesador.

El principio básico general del sistema es, enviar un pulso de una señal ultrasónica, en dirección de un blanco específico que puede ser el fondo del recipiente, esperar el rebote que causa el choque de ambos y recibirlo nuevamente a través del transductor, mientras tanto el microprocesador estará ejecutando un conteo, que posteriormente será equivalente a una cierta distancia, la cual al multiplicarse por el área predefinida del recipiente podrá arrojar la lectura de volumen, lleno o vacío, es decir, de líquido en el recipiente o el total del volumen de este cuando no guarde ninguna substancia.

Es aquí donde se explicara como se ejecuta la interfase entre la señal analógica de 12 volts de amplitud y que genera un pulso de cerca de 50 volts a 215 KHz, con la señal digital de 5 volts de amplitud y cuyo reloj esta en 1.7897 MHz.

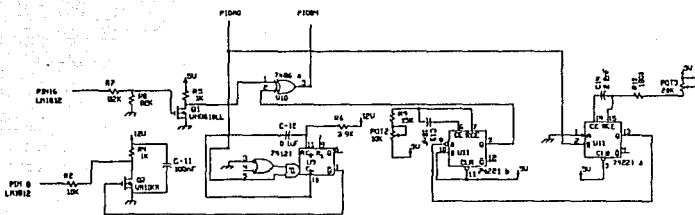
Para transmitir un pulso en el circuito LM1812, hay que abrir la llave de transmisión con una señal en la pata 8. Para ejecutar un pulso de transmisión acorde al tiempo de viaje y respectiva atenuación, se escoge de 270 microsegundos, el cual se genera con un circuito one-shot 74121 cuyo tiempo en alto del pulso de salida es:

$$tW_{(OUT)} = C - 12(R6) \ln(2)$$

de aquí que $R6 = 3.9K$ y $C = 0.1 \mu f$. Este pulso es efecto de una señal que proviene del Bit cero del PIO A, la se generara cuando el usuario del sistema pide se ejecute la lectura.

La salida del one-shot es de tipo TTL como la señal del CPU sin embargo la pata ocho del LM1812, denotada como chip enable (CE) en el diagrama general, se energiza con 12 volts, para ser homogénea con el resto del sistema analógico, por lo que se introduce un transistor de tipo FET Q2 polarizado con 12 Volts y además tiene por su configuración el comportamiento de inversor, de ahí que la señal proveniente del one-shot tiene que ser de la salida negada Q ya que, al ser esta inversa y reinvertirse a través de FET, llega con la lógica correcta para activar la transmisión del pulso ultrasónico.

INTERFASE ANALOGICO-DIGITAL



Una vez realizada la transmisión y capturado el rebote como se menciona en el capítulo dos, el CI LM1812 tiene una salida digital de unos y ceros en el pin 16, solo que es de niveles CMOS es decir, con una amplitud de 12 Volts. Para atenuar la señal, puesto que entrará al KIO (nivel TTL), la manera mas sencilla es con un arreglo de resistencias, tal que funcione como divisor de voltaje, lo cual nos lleva a una relación de $R7 = 1.402 R8$, pero estas resistencias causan una carga al circuito; entonces existe una caída de voltaje de entrada al divisor, esto se puede superar introduciendo valores altos de resistencia. Tomando en cuenta, que la entrada se va a 10 volts puede establecerse que $R7 = R8 = 82K\Omega$, además la señal resultante puede reconfigurarse a través de un transistor FET que servirá como pull-up, además actuará como inversor ya que la salida del pin 16 presenta lógica inversa, este FET es el presentado como Q1 en el diagrama. Como se puede observar se polariza con solo 5 volts, a diferencia de Q2 que necesitaba elevar el voltaje. Esta señal finalmente debe entrar al puerto B del Pío en el Bit cuatro para de ella obtener la información pertinente y ser procesada.

Un nuevo problema surge al entender que en la salida del sonar se nota un pulso cuando hace la recepción pero, también cuando hace la transmisión, es decir, la señal mencionada arriba, es procesada y llega a P10B4, pero puede ser el resultado de la transmisión y no del rebote deseado.

Esto quiere decir, siempre que exista una transmisión se presentara un pulso de inmediato en la entrada del Pío.

Esto debe solucionarse eliminando siempre el pulso que corresponda a la excitación, para ello es evidente, se hecha mano de la señal de salida en el PIOA. Dicha señal sabe es la que dice cuando existe una transmisión.

De aquí que, si se hace pasar a la señal de la salida (drain) de Q1 junto con esta señal de PIO A0 a través de una compuerta OR exclusiva podemos seleccionar la señal de información real pues:

| A + B | | SALIDA | |
|----------------------|------|--------|---|
| PIO _{A0} =0 | Q1=0 | 0 | |
| PIO _{A0} =0 | Q1=1 | 1 | salida deseada |
| PIO _{A0} =1 | Q1=0 | 1 | estado que no existe |
| PIO _{A0} =1 | Q1=1 | 0 | no puede haber excitación en una oscilación en Q1 |

De esta manera basta sincronizar las señales del PIOA0 y Q1 (drain) para que entren exactamente al mismo tiempo a la compuerta exor SN7486.

Sincronización del microprocesador con el sonar.

Esta sincronización es debido a que el CI LM1812 tiene un periodo de retardo al responder, de ahí que la señal proveniente del PIOA0 se extienda a través de otro circuito one-shot, esta vez el SN74221 puesto que se desea obtener un periodo de defasamiento equivalente al retardo que introduce el sonar, así como la generación de un pulso de igual tamaño exactamente como el de la salida del pin 16 del sonar.

El primer one-shot del circuito SN74221 produce un pequeño pulso de 30 μ s de duración y se conecta en cascada con el segundo one-shot del mismo SN74221 solo que activándolo de manera inversa, es decir alimentándose por la entrada A y de duración 270 μ s aproximadamente.

El tiempo de duración de los pulsos es de nuevo calculado con la ecuación $t_w = C_{ext} R_t \ln 2$, con lo que se puede encontrar el valor de las resistencias R9 y R10, así como el de los capacitores C13 y C14.

La incorporación de dos potenciómetros más el POT2 y el POT3, que van conectados con las resistencias R9 y R10 en serie respectivamente, permiten ajustar con precisión el valor de tales resistencias y de esta forma generar pulsos de manera exacta como se necesitan y que por lo tanto, a la salida de la compuerta or exclusiva SN7486 no existan picos cuando es transmisión, pero después de la recepción se pueda obtener una señal acorde para ser procesada tal y como se desea.

Cabe mencionar que en cuanto ingrese esta señal al PIOB4, el CPU a su vez mandara una instrucción para que el CTC se detenga en el conteo iniciado y arroje una lectura equivalente ala distancia que viajo la señal en el aire. De este modo, el CPU queda sincronizado con el sonar tanto para transmitir como para recibir.

Eliminación de ruidos.

En este circuito puede haber dos tipos de ruidos, el que se introduzca ala sección de 5 volts y que altere al control del CPU y el que imite a la resonancia del sonar creando confusiones.

La parte digital se puede proteger con una correcta fijación y distribución de los elementos en la tableta de circuito impreso para asegurar que en donde deban existir 5 volts efectivamente hay 5 volts y que en donde deba haber cero que es el nivel de referencia denotado como tierra (GND) existan cero volts. Con la frecuencia de trabajo de reloj superior a un megahertz pueden existir problemas de este tipo. Una buena solución para ayudar a estabilizar contra ruidos es la introducción de capacitores en los circuitos TTL, denotados como C4, C5, C6, C7, C15, y C16 los cuales protegen contra la inserción de picos de voltaje y rizados no deseados.

Para la parte analógica deben tenerse en consideración todas las precauciones establecidas en el capitulo dos. Se pueden utilizar también capacitores como los mencionados, así por ejemplo el C-11. La frecuencia de oscilación es relativamente alta para ser transmitida en aire, sin embargo protege contra posibles armónicas provenientes de señales audibles, pues se nota que para una señal de 215 KHz como primer armónica la fundamental debe ser de 107 KHz y una señal sonora así es bastante rara.

También la disposición de los elementos es sumamente importante pues se recomiendan pistas mas o menos gruesas para evitar el efecto antena, y otra recomendación muy importante es la buena aterrización de los elementos es decir correctas soldaduras y protección en las patas para evitar señales indeseadas y canalizarlas a tierra correctamente. Es importante debido a la alta ganancia del receptor, eliminar oscilaciones para esto es importante separar los componentes del circuito tanque L1, C1 de los componentes que están a la salida con el transductor C-6, C-7 y T1, y con esto evitar posibles acoplamientos inductivos. El transductor debe estar protegido con malla de alambre tipo, para asegurar un correcto aislamiento y aterrizado.

Modulo de adquisición de datos.

El modulo de adquisición de datos tiene por objetivo captar la información proveniente del sonar después de que este ha recibido el eco de una señal que se direccionó a algún blanco (por ejemplo el fondo de un barril).

El PIOA manda una señal y se activa el CTC, mientras se espera el rebote de la señal sonora el CPU activa al PIO B para recibir la señal de información. Esta activación se lleva a cabo por medio de una rutina de poleo como se le conoce, la cual consiste básicamente en sensar la información del BIT cuatro de puerto B del pio. Esta rutina se llama Lee en el apéndice A .Tiene una estructura que básicamente funciona bajo la figura de un IF THEN ELSE de cualquier lenguaje estructurado de programación, y trabaja de la siguiente manera.

Llama los datos del puerto B, del PIO los deposita en el acumulador, a este registro lo máscara y se prueba el Bit cuatro, si hay un uno, se sale del bucle para ir a detener al CTC, si no de nuevo llama a los datos del PIO B.

Una vez que sale el bucle, antes de detener al CTC, se recuperan los datos que hay en el, tanto en el puerto uno como en el dos, que son los que realmente nos arrojan la información de la medición necesaria para procesarla adecuadamente.

De esta manera se ha mencionado, la adquisición de los datos provenientes del sonar, pero también existirán datos provenientes del usuario.

Estos datos serán ingresados al CPU por medio del PIOB en sus primeros cuatro Bits PBO,PB1,PB2,PB3, para esto se incorpora una rutina que atiende al valor que tiene el PIOB ,cuando se necesite un dato que ingrese el usuario, tal rutina funciona nuevamente con el método de poleo y hace una máscara de los 4 bits de interés para que regrese con la información requerida, esta rutina se denomina tecla.

Al retorno de esta rutina, cada vez que se le llama, posee en el acumulador el valor correspondiente a la tecla oprimida, con lo cuál se obtiene información. Este valor de información puede ser cualquiera de entre 0EH, 0DH, 0BH o 07H, correspondiendo cada uno en su caso a una función específica de acuerdo al momento en que se le llama.

Así por ejemplo cuando se ingresen "numeros" para realizar el calculo del volumen del registro A tiene que ser 07H y cada vez que este "numero" corresponda al deseado por el usuario, se pedirá que el registro A tenga el valor de 0EH.

Como se habrá entendido, la rutina tecla se llama en el momento que sea necesaria información complementaria por el usuario, tanto para cálculos como para designar la ruta que ha de seguir el programa pues depende del valor del registro acumulador si este es 0B o 0D por ejemplo seguirá hacia un recipiente horizontal predefinido o a un cilindro vertical respectivamente. En el primer caso 0B indicara accesar directamente a la tabla de volúmenes cuando regrese la señal del sonar, mientras que el segundo preguntara si contiene liquido o no y de acuerdo a la respuesta nuevamente 0B o 0D pedirá dos datos numéricos o sólo uno y con estos datos calculara los volúmenes respectivos.

Adelante se incluirá un diagrama de flujo en el que entiende claramente que cada vez que se hace una pregunta se hace una nueva adquisición de datos a través del miniteclado incorporado.

Modulo de proceso de información

La ruta elegida para obtener información a partir de los pulsos de transmisión y recepción, es utilizando al CTC como circuito base.

Cuando el Software llega al nivel en el que, el usuario da la orden de realizar la medición, ocurren dos cosas de manera prácticamente simultanea.

La primera es mandar una señal de salida al PIOAO, la cual activará sonar. La segunda situación es mandar activar al CTC para que inicie un conteo de manera que cuando el rebote sea sensado, se adquiera la información existente en el CTC.

La razón de utilizar 2 canales del CTC, es porque si se acomodan correctamente estas 2 lecturas se pueden tener en cuenta de hasta FFFF H o sea 36.61 mSeg a la frecuencia de reloj, y con este tiempo tomando la velocidad standard del sonido, se pueden alcanzar recorridos de un poco más de 12 mt. En este caso se debe tener en cuenta que el número que se obtiene es debido a la distancia total del recorrido, es decir, ida y vuelta, lo que significa que con FFFFH puede la señal viajar aproximadamente 6 metros en el aire.

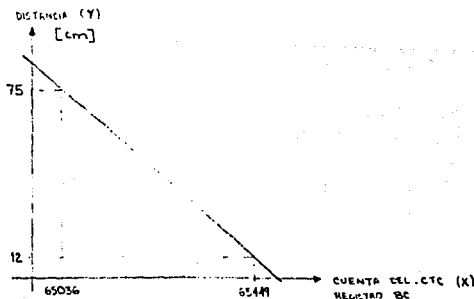
Cabe hacer la aclaración de que como el CTC cuenta en forma regresiva y es mas correcto referirse a esto si el CTC tiene en ambos canales 00, como que transcurrieron 65535 pulsos de reloj del sistema.

Para la correcta interpretación de las cuentas obtenidas por el CTC se elabora un método experimental, el cual consiste primordialmente en comprobar la linealidad de las lecturas. Esta linealidad esperada es debido a, la velocidad es prácticamente constante y por lo tanto variara la cuenta del CTC en forma proporcional a la distancia recorrida por la señal ultrasónica.

El método consiste en tomar lecturas de cuentas y medir las distancias, con esto observar el correcto funcionamiento del sistema y almacenar las lecturas en un archivo ascii, el cual servirá para alimentar al programa desarrollado en pascal que se muestra en el apéndice B.

Este programa en turbo pascal, se vale del archivo que contienen las lecturas (JJ2.TAB) para que con ellas realice una interpolación lineal y pueda encontrar todos los puntos de la recta generada en forma experimental. Cada uno de estos puntos, corresponde a una cuenta con su respectiva distancia. Los datos generados a causa de la interpolación se mandan a un nuevo archivo en formato ascii, el cual se incluye en el programa general de la EPROM, a través del comando INCLUDE del cross assembler de avocet. El programa que se menciona es muy sencillo, y lo que hace es realizar las interpolaciones correspondiente de acuerdo al siguiente principio matemático.

La tabla generada es de la forma.



A cada punto le corresponden sus coordenadas (x,y), y podemos partir de una forma general de interpolación conocida, de la que podemos despejar la correspondiente interpolación o extrapolación como sigue:

$$Y_1 = (Y_3 A - Y_2) / (A - 1)$$

$$Y_2 = A (Y_3 - Y_1) + Y_1$$

$$Y_3 = [(Y_2 - Y_1) / A] + Y_1$$

Donde A es un número conocido para cada Y_1 , Y_3 extrapolación o Y_2 interpolación y se encuentra de la siguiente manera:

$$A = (X_2 - X_1) / (X_3 - X_1)$$

Una vez obtenida esta tabla, queda anexada en el programa de la EPROM. Puede accederse por medio de su primer byte, que esta contenido en la dirección TAB-DIST, a cada distancia le corresponden 2 Bytes, en el apéndice A esta la tabla, aparece editada y solo se ve una muestra. Basta recorrer el apuntador (HL) el doble de las n veces que el desplazamiento indique para que se posicionen en el lugar correspondiente a la distancia equivalente. Este desplazamiento es el resultado de restar la cuenta que arrojó el CTC, contenida en los registros BC, al valor inicial de la cuenta contenida en la dirección VAR1, de esta manera, se produce que aunque el CTC cuente regresivamente, se tenga un desplazamiento mayor para una distancia mayor. Esta rutina se llama POS-TABLA.

Una vez obtenido el valor numérico en BCD, se guarda en direcciones generadas en ram cuyo apuntador es distancia y es fundamental para obtener los volúmenes deseados.

Hasta aquí solo se ha hablado del procesamiento de la información adquirida del sonar, pero existe otra información más. Es muy importante para que el sistema arroje resultados palpables. Esta información de que se menciona, es aquella que introduce el usuario y hace al medidor versátil.

A través del PIOB se reciben datos, para que auxiliados con la distancia se obtengan volúmenes de objetos cuya forma geométrica sea cilíndrica.

Los datos son básicamente el diámetro del cilindro, y la altura máxima de este, dependiendo del nivel del programa al que se hubiera ingresado. De esto hablaremos ampliamente en la siguiente sección.

Cuando se ingresa el dato del diámetro, el CPU está preparado para calcular el área de la cara del círculo del cilindro, y se mantiene contenida en una dirección de ram, hasta que se obtiene la medición de distancia y con esta el volumen. Es evidente que estos cálculos son solo multiplicaciones en BCD, el primero es resultado de realizar $s = \pi D^2 / 4$ y el segundo $h * s$; donde h es la distancia.

En cuanto a la forma en la que se pide también ingresar la altura máxima el procedimiento es similar, solo que aquí se espera la información de la distancia proveniente del sonar, y esta se resta de la altura máxima para así obtener una altura de líquido con lo cual se procede a calcular de la misma manera que antes el volumen de dicho líquido.

La información que se ha comentado es la que implica un cierto proceso, la demás información que entra al sistema por parte del usuario es mas bien complementaria, que solo decide rutas del mismo.

Modulo de atención al usuario

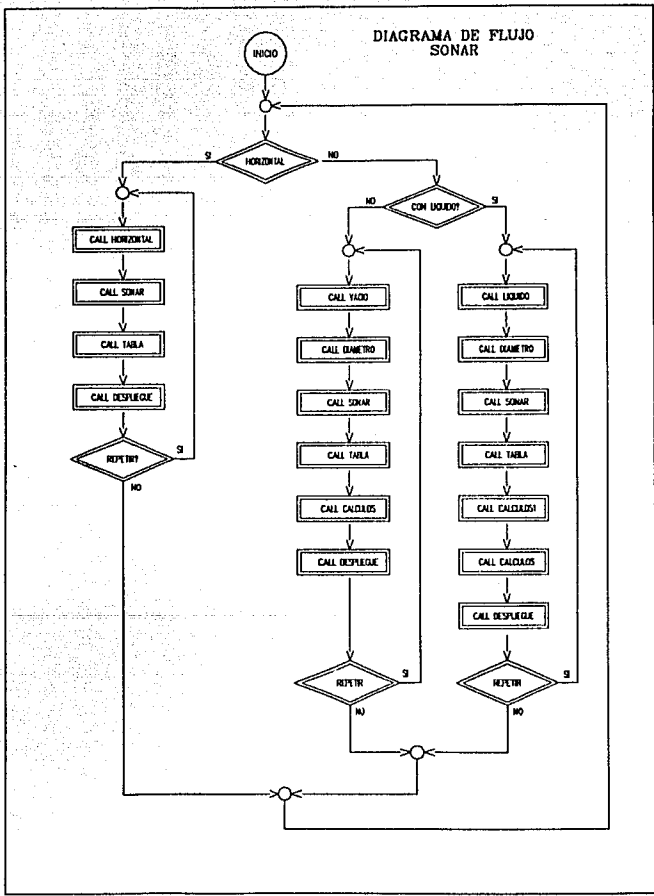
Se incorpora un miniteclado a través del PIO B con el cual el usuario tendrá acceso al sistema y podrá ordenar su funcionamiento.

Es esta parte de atención al usuario, la que realmente da la estructura básica al sistema, ya que establece niveles de petición de información por parte de quien lo maneja. Pretendiendo que sea fácil de usar y atractivo, conociéndose como amigable. Es necesario entonces, incorporar teclas de uso específico para que se decida la ruta a seguir y el usuario tenga el manejo del sistema.

De esta manera se crea entonces una rutina para efectuar la medición de algún tanque horizontal cuyas dimensiones son predeterminadas, o bien otra medir algún barril vertical del cual se tendrá que ingresar la información pertinente, o una mas, que incluye la posibilidad de que este barril tenga algún líquido y se desee medir la cantidad de tal substancia.

En el sistema final no se contempla el solo medir distancias, sin embargo, puede crearse una subrutina más, muy sencilla para que se tengan estos datos, o bien, utilizando este mismo sistema se puede preparar para que al desplegar la lectura final se obtenga una distancia.

DIAGRAMA DE FLUJO SONAR



Para esto es necesario elegir la opción de recipiente vacío, y posteriormente introducir el valor del diámetro igual a 1.1284, con lo que se generara un área cuyo valor es la unidad y uno multiplicado por la distancia arroja como resultado la distancia medida.

A continuación se muestra el diagrama de flujo del software del sistema y en el apéndice D se encuentra la designación de las teclas, así como su uso.

Modulo de despliegue de información.

Es evidente que al incorporar el LCD como un puerto más, toda la información tanto de resultados como de entrada de datos, se desplegará a través de este.

Se han creado algunas rutinas que permiten el uso mas sencillo y dinámico del display. Es en esta etapa en la que se elaboran las rutinas para acomodar los datos adecuadamente, ya que en algunos casos existirán despliegues especiales, por partes y no como se despliega un letrero, que es corrido y basta escribirlo.

Como ejemplo la rutina SET_DIAM, en la que se ve que el despliegue de datos tiene que ser por zona y no de todo el display. Pensar que se obtenga la información, se acomode un letrero en alguna zona de ram y posteriormente mandarlo al LCD es problematico, es mejor ir escribiendo la información directamente al display, auxiliado de algunas rutinas como lo CAR_DISP.

Puede verse entonces en la rutina SET_DIAM que primero se manda un letrero como los anteriores, pero en el cuál queda predefinida la posición de los caracteres numéricos que se desea aparezcan, entendiéndose que el último caracter (de izquierda a derecha) está en la posición 8B del display ,la que se guarda en una dirección llamada POS_CAR. Al contenido de esta dirección se manda el numero 30H que corresponde al cero en ascii y se hace la incursión de una subrutina llamada MUESTRA.

Esta subrutina muestra, llama a la mencionada CAR_DISP ,que hace que el caracter en uso este subrayado por el cursor, posteriormente se llama a TECLA que por supuesto pide al usuario si el cero (30H) que aparece desplegado, es el que se desea o es necesario correr a otro número, si la respuesta es buscar otro número, manda al acumulador el valor que tiene el bit "caracter" y lo compara con 39H que es el 9 en ascii, si es igual se va a cero el valor de la dirección caracter y si no, aumenta el valor del acumulador para de esta manera tener ahora el uno es ascii.

Nuevamente lo despliega con `CAR_DISP`, hace solicitud de la tecla que el usuario maneje, de esta forma el usuario puede ver pasar en la misma posición del display, todos los números del cero al nueve en forma cíclica hasta que decida si el número es el adecuado mediante la tecla OK. Al sensar esta señal resta 30H para obtener el número en ascii en el acumulador y se sale de la subrutina nuestra para volver a la de `SET_DIAM`, en esta se guarda el valor del registro A en una variable denominada `DIAMETRO`. El siguiente paso es cambiar el contenido de la posición del carácter por el valor de 8A, y realizar el mismo procedimiento anterior. Notese que el siguiente valor de la posición del carácter no es 89H sino 88H, esto se debe a que el valor de 89 se respeta, pues ahí es en donde está almacenado el punto decimal. Se nota que el valor del diámetro queda contenido en 4 bytes.

Usando la rutina `DESP_DIAM` puede mandarse todo el valor del diámetro junto con su respectivo letrero, de la siguiente manera.

Se usa el registro DE como apuntador de la variable `Aviso_Display`, la cual se utiliza cuando se hace el refresco del LCD, entonces si se indexa una dirección con `Aviso_Display+13` por ejemplo, se refiere a que en la posición número 13 del display (cuenta de la 0 a la 15) se va a escribir un dato. Notase la suma de 30H a cada valor por que así realiza la conversión en ascii. De esta forma se está escribiendo lugar a lugar el valor del diámetro y se despliega todo mediante un `REFRESH_LCD`.

Finalmente como complemento de la rutina `SET_DIAM`, existe la rutina `GUARD_DIAM`, que junta los dos primeros bytes leídos y los pone en una nueva variable de RAM llamada `DIAM1`. A los dos siguientes bytes los pone en `DIAM1+1` asegurando que en la dirección `DIAM` se tiene el valor del diámetro leído y desplegado, en formato BCD.

De manera similar se lee la altura y se despliegan los volúmenes resultantes de las lecturas.

Referente a la opción que se incluye de un recipiente horizontal, el despliegue de resultados es directo de una tabla que se genera muy similar a la comentada. Esta tabla se obtiene con el programa en turbo pascal del apéndice C, el cual queda diseñado para que el usuario, solo cambie el valor del diámetro y la longitud del recipiente y con esto se genera la tabla correspondiente, de manera que el MUV pueda obtener las lecturas de volúmenes del recipiente deseado.

El programa del apéndice C, se basa en el cálculo de la integral de una circunferencia delimitada por una recta,

Se envía una señal que arroja una distancia, misma que se resta al valor del diámetro ,con esto obtenemos el área del recipiente hasta donde quede el nivel de agua.

Esta área junto con la longitud del cilindro permite calcular el volumen correspondiente. Generando de esta forma una tabla de volúmenes en función de la distancia recorrida por la señal ultrasónica . El acceso a la tabla de volumen en ensamblador es completamente análogo al acceso de la tabla de distancia.

Tarjeta de circuito impreso para el sistema de microprocesador.

Es aquí, donde se integra todo el sistema, se trata de hacer lo más compacto posible.

Como se sabe la introducción del circuito KIO ayuda bastante a la compresión de los elementos, por que es un circuito multipuertos de alta escala de integración, pero sin embargo es fundamental que se monte sobre un circuito impreso, y muy preferentemente de dos caras, además de que las pistas tienen que ser muy delgadas para así aprovechar todas las salidas del KIO. Es por todo esto que un auxilio muy práctico son los paquetes computacionales. En este caso se utiliza el Hiwire de Wintek Corporation, para realizar el circuito impreso. El Hiwire presenta la opción del autorutin, después de haber generado un circuito esquemático y acomodar los elementos de una manera adecuada.

En la siguiente página se anexa el diagrama de la distribución de los elementos de todo el circuito impreso.

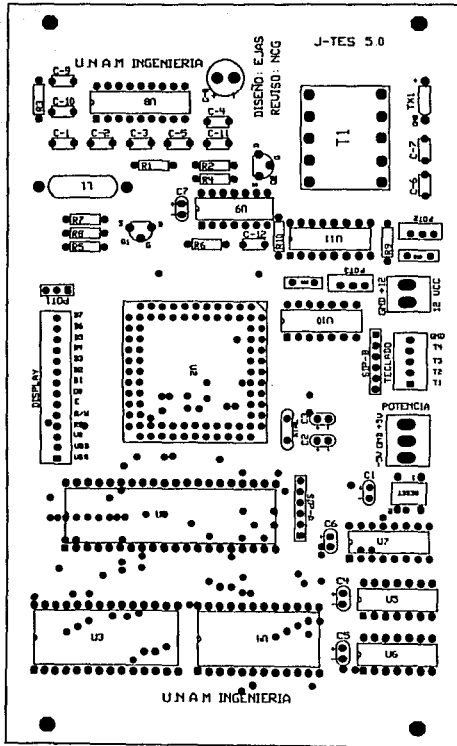
Como se puede observar se incluyen conectores de tipo panduit para manejar el ensamble del display y del teclado, así de esta forma el montaje final se simplifica. También se nota que para la economía de espacio y ensamble, las resistencias marcadas como R1, R2, R3, R4, R5 de pull-up utilizadas para señales del microprocesador y del teclado, se han comprimido en dos arreglos de tipo SIP-A y SIP-B.

Tarjeta de circuito impreso para el sistema de sonar.

Podemos pensar en realizar dos tarjetas por separado una que conserve al sistema digital y otra al analógico, pero como el complemento de la parte analógica es con circuitos TTL, realizar una tarjeta comprendida por el sonar implicaría tener dos polarizaciones, una a 5 volts y otra de 12 volts.

Debido a que ya se cuenta con una tarjeta polarizada con 5 volts sería solo complicación el doble alambrado de la misma fuente, por lo que se crea entonces una tarjeta general que incluya tanto la parte analógica como la digital. Esta tarjeta de cualquier manera se polariza con dos fuentes, pero se economizan muchas conexiones externas entre ambas.

DISPOSICIÓN DE LOS ELEMENTOS



DISPOSICIÓN DE LOS ELEMENTOS

El diagrama esquemático general es el ilustrado en la página 52.

A partir de este diagrama, la tarjeta finalmente queda de la manera como se muestra en la página siguiente (53).

La lista de las partes de esta tarjeta, es decir sin incluir la fuente de alimentación es:

Resistencias:

| | | | |
|---------------------|---------------------|--------------------|----------------------|
| R1 = 5.1 K Ω | R4 = 1 K Ω | R7 = 82 K Ω | R10 = 1.8 K Ω |
| R2 = 10 K Ω | R5 = 1 K Ω | R8 = 82 K Ω | |
| R3 = 47 K Ω | R6 = 3.9 K Ω | R9 = 15 K Ω | |

Capacitores:

| | | |
|------------------|-----------------------|--------------------|
| C1 = 10 μ f | C-1 = 330 pf | C-8 = 470 μ f |
| C2 = 33 pf | C-2 = 470 pf(varicap) | C-9 = 10 nf |
| C3 = 33 pf | C-3 = 1 nf | C-10 = 1 nf |
| C4 = 0.1 μ f | C-4 = 220 pf | C-11 = 100 nf |
| C5 = 0.1 μ f | C-5 = 0.1 μ f | C-12 = 0.1 μ f |
| C6 = 1.1 μ f | C-6 = 100 pf | C13 = 10 nf |
| C7 = 0.1 μ f | C-7 = 22 pf(varicap) | C14 = 2.2 nf |

Potenciómetros:

| |
|-----------------------|
| POT 1 = 20 K Ω |
| POT 2 = 20 K Ω |
| POT 3 = 20 K Ω |

Transistores FET:

| |
|---------------------------|
| Q ₁ = VN0610LL |
| Q ₂ = VN10KM |

Cristal:

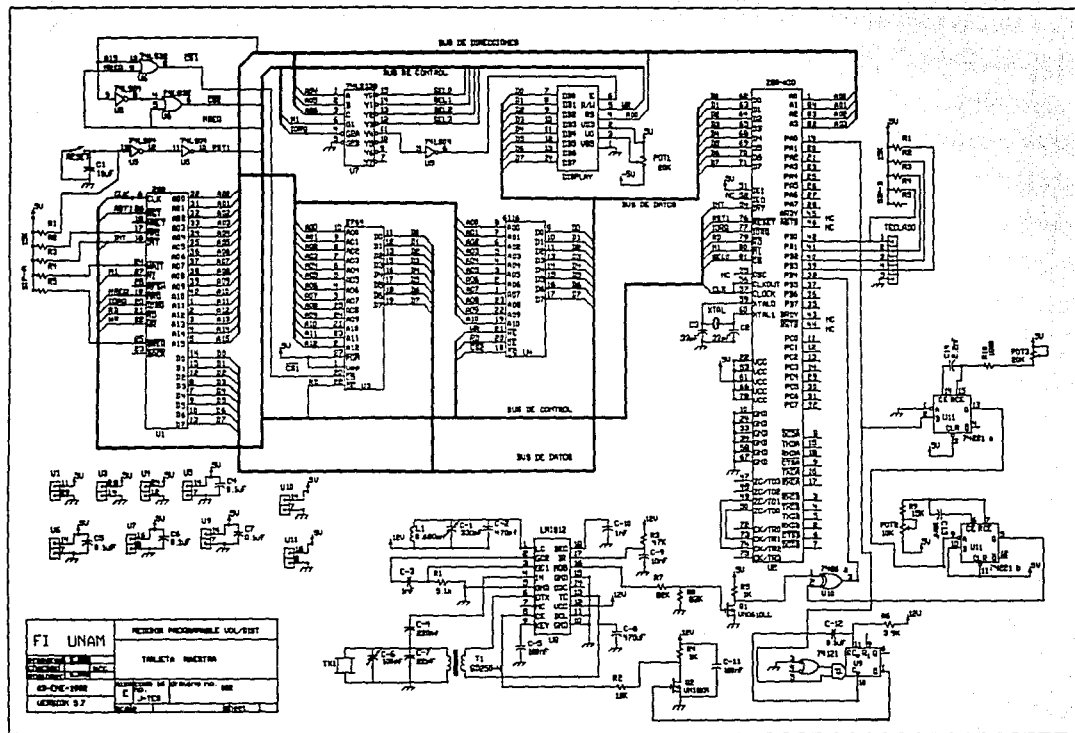
XTAL = 3.579545 MHz

Display:

Display = 11603 Shelly

Transformador:

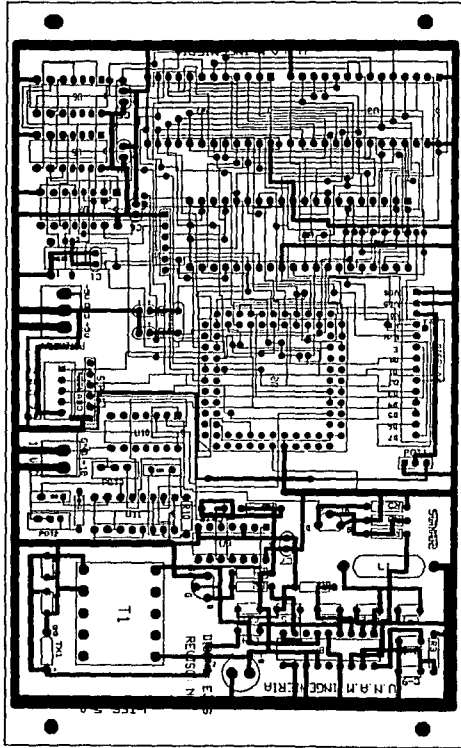
T1 = SD250-4 = 75672-A Coilcraft



| | |
|-------------|--------------------------------|
| FI UNAM | RECORDADOR PROGRAMABLE VOL/EST |
| | TRILISTA INACTIVA |
| REVISOR 10 | |
| REVISOR 11 | |
| REVISOR 12 | |
| REVISOR 13 | |
| REVISOR 14 | |
| REVISOR 15 | |
| REVISOR 16 | |
| REVISOR 17 | |
| REVISOR 18 | |
| REVISOR 19 | |
| REVISOR 20 | |
| REVISOR 21 | |
| REVISOR 22 | |
| REVISOR 23 | |
| REVISOR 24 | |
| REVISOR 25 | |
| REVISOR 26 | |
| REVISOR 27 | |
| REVISOR 28 | |
| REVISOR 29 | |
| REVISOR 30 | |
| REVISOR 31 | |
| REVISOR 32 | |
| REVISOR 33 | |
| REVISOR 34 | |
| REVISOR 35 | |
| REVISOR 36 | |
| REVISOR 37 | |
| REVISOR 38 | |
| REVISOR 39 | |
| REVISOR 40 | |
| REVISOR 41 | |
| REVISOR 42 | |
| REVISOR 43 | |
| REVISOR 44 | |
| REVISOR 45 | |
| REVISOR 46 | |
| REVISOR 47 | |
| REVISOR 48 | |
| REVISOR 49 | |
| REVISOR 50 | |
| REVISOR 51 | |
| REVISOR 52 | |
| REVISOR 53 | |
| REVISOR 54 | |
| REVISOR 55 | |
| REVISOR 56 | |
| REVISOR 57 | |
| REVISOR 58 | |
| REVISOR 59 | |
| REVISOR 60 | |
| REVISOR 61 | |
| REVISOR 62 | |
| REVISOR 63 | |
| REVISOR 64 | |
| REVISOR 65 | |
| REVISOR 66 | |
| REVISOR 67 | |
| REVISOR 68 | |
| REVISOR 69 | |
| REVISOR 70 | |
| REVISOR 71 | |
| REVISOR 72 | |
| REVISOR 73 | |
| REVISOR 74 | |
| REVISOR 75 | |
| REVISOR 76 | |
| REVISOR 77 | |
| REVISOR 78 | |
| REVISOR 79 | |
| REVISOR 80 | |
| REVISOR 81 | |
| REVISOR 82 | |
| REVISOR 83 | |
| REVISOR 84 | |
| REVISOR 85 | |
| REVISOR 86 | |
| REVISOR 87 | |
| REVISOR 88 | |
| REVISOR 89 | |
| REVISOR 90 | |
| REVISOR 91 | |
| REVISOR 92 | |
| REVISOR 93 | |
| REVISOR 94 | |
| REVISOR 95 | |
| REVISOR 96 | |
| REVISOR 97 | |
| REVISOR 98 | |
| REVISOR 99 | |
| REVISOR 100 | |

DISPOSICIÓN DE LOS ELEMENTOS

DISPOSICIÓN DE LOS ELEMENTOS



Transductor:

TX1 = E-188/215 Massa

Inductancia:L1 = 680 μ H = PCH-27-684 Coilcraft**Circuitos integrados:**

| | |
|---------------|-----------------------|
| U1 = 280 CPU | Microprocesador |
| U2 = 280 KIO | Circuito multipuertos |
| U3 = 2764 | Memoria EPROM |
| U4 = 6116 | Memoria RAM |
| U5 = 74LS04 | Inversores |
| U6 = 74LS32 | Compuertas OR |
| U7 = 74LS138 | Decodificador |
| U8 = IM1812 | Sonar |
| U9 = SN74121 | One-shot |
| U10 = 74LS86 | Compuertas EXOR |
| U11 = SN74221 | One-shot |

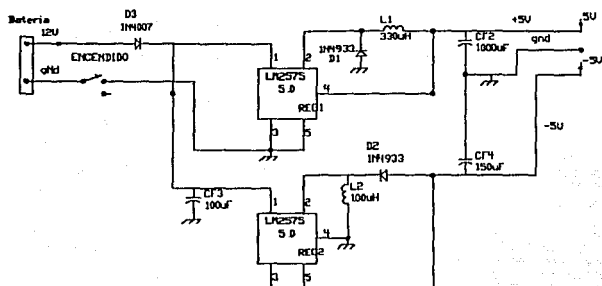
Fuente de alimentación.

Para polarizar el circuito se necesitan 3 fuentes de poder una de 5 volts que debe suministrar una corriente de 220mA, otra de -5 volts de la cual el consumo de corriente es mínimo 2 mA, pues solo se utiliza para dar intensidad al display, y por ultimo una de 12 volts para polarizar al sonar y que sea capaz de drenar 25 mA.

Para hacer del MUV un sistema versátil de fácil manejo, que aquí se pretende, resulta ideal pensar en incluir una batería. Esta batería debe ser de 12 volts para que pueda polarizar al circuito que necesita más voltaje. Para obtener ± 5 volts a partir de esta batería se hacen dos fuentes conmutadas (switcheadas), se prefieren de tipo conmutado y no lineales para eficientizar el uso de la corriente y no introducir una batería muy grande.

Para la elaboración de estas dos fuentes una de tipo Boost y otra Buck se utilizan los circuitos LM2575 de Motorola, los cuales proveen las características de realizar el trabajo de conmutación, solo basta introducir una bobina, un capacitor y un diodo como componentes externos. El diagrama esquemático de estas 2 fuentes es el siguiente:

DISPOSICIÓN DE LOS ELEMENTOS



Conclusiones

Muchas conclusiones fueron tomadas durante el desarrollo de este sistema. Es evidente que el diseño del mismo, tuvo que sufrir modificaciones y adecuarse a los problemas que fueron surgiendo.

El sistema presentado es el resultado de muchas mejoras técnicas a prototipos desarrollados probando cada sección por separado y posteriormente en un sistema general.

Se generó un sistema amigable con el usuario, práctico y portátil, capaz de realizar mediciones en el rango de distancias de 13 a 90 centímetros, procesar los resultados y obtener volúmenes de recipientes con líquido o completamente vacíos.

Queda también establecido un respaldo de software para que se puedan manejar recipientes horizontales de tipo tanque "pipa" muy comunes. Con este software el dispositivo se puede adecuar a las necesidades de muchos usuarios.

Este método de medición establece las bases para realizar sistemas que sean capaces de realizar control de nivel de líquidos, medición de flujo en canales abiertos, medición de volúmenes en salones, entre otros.

Desde luego se pueden realizar mejoras al diseño presentado, pues se ha creado la base para la investigación en este tipo de dispositivos: Medidores ultrasónicos.

Algunas de las mejoras posibles pueden ser:

- Realizar una compresión de componentes, utilizando circuitos "Surface mounted", integrar la fuente a la tarjeta maestra para reducir espacio y conectores.

- Cambiar por un display que solo utilice polarización positiva de +5 volts, eliminando una fuente conmutada.

- Mejorar la presentación del sistema, cambiando el chasis final.

- Introducir algún supervisor para carga de batería.

- Cambios en software para otras mediciones.

- Ajustes en la etapa de salida para obtener mayor potencia y con ello alcance.

El sistema en general es equiparable con muchos, que muy recientemente han llegado al mercado mexicano. Este diseño es competitivo con sistemas de características parecidas, con la ventaja de ser un diseño nacional.

CONCLUSIONES

Más aún es un diseño universitario que demuestra que en México, y en especial en la Facultad de Ingeniería de la UNAM, los alumnos y profesores desean realizar nueva tecnología, poseen las bases teóricas y que con solo hacer un ligero intercambio con las industrias privadas, es posible el desarrollo de nuevas opciones tecnológicas.

El costo total del sistema es el siguiente:

| | |
|-----------------------------|--------------|
| Circuitos integrados TTL. | 2.82 |
| CPU, RAM, EPROM. | 8.25 |
| KIO. | 23.75 |
| Bobina | 3.00 |
| Transistores | 1.40 |
| Cristal | 1.19 |
| Display | 23.20 |
| Transformador | 30.00 |
| Transductor | 47.25 |
| Resistencias | 0.20 |
| Dip | 1.24 |
| Capacitores | 8.44 |
| Interruptores | 2.00 |
| Conectores | 4.79 |
| Tarjeta de circuito impreso | 35.00 |
| Bases CI | <u>15.49</u> |
| | 208.02 |

Todas las cantidades desplegadas hasta aquí están en dólares Estadounidenses.

Dicho costo total es calculado, tomando como base costos unitarios de ahí que se puede abatir grandemente, en función de la comercialización del sistema.

Como punto de comparación podemos pensar en los sistemas comerciales LV400 series de Omega o series 4200 de BadgerMeter, Inc; los cuales son para medición de flujo en canales abiertos. Funcionan de manera similar al sistema desarrollado y con características muy semejantes. El LV404 de Omega solo mide distancias, el costo de estos sistemas asciende a 350 dólares más gastos de importación. Pero sin software, el valor del software apropiado para estos sistemas es variable. Notese que a pesar de que el sistema desarrollado no es en primera instancia tan económico, en comparación con los equipos comerciales, sí resulta serlo y en gran escala. Si tomamos en cuenta además en este caso el ahorro de líquido, vemos que efectivamente el sistema es de un costo muy bajo.

APENDICE A

PROGRAMA EN LENGUAJE ENSAMBLADOR PARA Z80

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```
0000          "Z80"
0000          ; PROGRAM SONAR;
0000          ; {PROGRAMACION DE UN SONAR PARA MEDIR VOLUMEN);
0000          ; PUERTOS
0000          PIO_A      EQU    00H      ; pio_a_datos:=00H
0002          PIO_B      EQU    02H      ; pio_b_datos:=01H
0001          PIO_AC     EQU    01H      ; pio_a_comandos:=02H
0003          PIO_BC     EQU    03H      ; pio_b_comandos:=03H
0004          CTC_0       EQU    04H      ; ctc_0:=04H
0005          CTC_1       EQU    05H      ; ctc_1:=05H
0006          CTC_2       EQU    06H      ; ctc_2:=06H
0007          CTC_3       EQU    07H      ; ctc_3:=07H
000F          KIO_C      EQU    15D      ; LCD_comandos := 10H
0040          LCD_C      EQU    40H      ; LCD_comandos := 10H
0041          LCD        EQU    41H      ; LCD_datos:= 11H
0000          ;
8000          ORG      8000H          ; VAR
8000 00          RAM      DEFB 00          ; RAM:=BYTE
8001          AVISO_DISPL DEFS 16D          ;
8011 00          CARAC    DEFB 00          ;
8012 00          POS_CAR  DEFB 00          ;
8013          VECES      DEFS 03H          ;
8016 00          TECLA1   DEFB 00          ; Byte para guardar la lectura
8017 00          DIG_1    DEFB 00          ; Byte para separar digito BCD
8018 00          DIG_2    DEFB 00          ; Byte para separar digito BCD
8019 00          DIGIT_A  DEFB 00          ;
801A 00          DIGIT_B  DEFB 00          ; RUTINA BN2BCD
801B          DESPLZ     DEFS 02          ; DESPLAZAMIENTO:BYTE;
801D 00          DESP_N   DEFB 00          ;
801E          DISTANCIA  DEFS 10H          ;
802E          DIAMETRO   DEFS 10H          ; DIAMETRO:SEGMENTO (LUGAR PAR
803E          DIAM1      DEFS 10H          ; DIAMETRO PARA LA MULTIPLICAC
804E          AV1        DEFS 10H          ;
805E          VOL_VAC    DEFS 10H          ;
806E          ALTURA    DEFS 10H          ;
807E          ALT1       DEFS 10H          ;
808E          VOL_PIPA   DEFS 10H          ;
809E          CUENTA     DEFS 05H          ;
80A3          LEN        DEFS 01          ; {DE LA MULTIPLICACION} LONGI
80A4          DCNT       DEFS 01          ; DIGIT COUNTER FOR BYTES
80A5          LPCNT      DEFS 01          ; CONTADOR DE MAYA
80A6          OVRFLW     DEFS 01          ; BYTE DE OVERFLOW
80A7          MCADR      DEFS 02          ; SIGUIENTE PARA SER GRABADO
80A9          MPADR      DEFS 02          ; DIRECCION DEL MULTIPLICADOR
80AB          NBYTE      DEFS 02          ; SIGUIENTE DIGITO DEL MULTIPL
80AD          PROD       DEFS 20H          ; BUFFER DEL PRODUCTO
80CD          MCAND      DEFS 20H          ; BUFFER DEL MULTIPLICANDO
80ED          MINUENDO   DEFS 10H          ; MINUENDO DE LA RESTA DE LA T
80FD          ;
8500          ORG      8500H          ;
8500 00          PILA     DEFB 00H          ; PILA:=BYTE
8501          ; DATA ENDS
8501          ;
```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0000                ORG      0000H      ;
0000 F3            INICIO    DI         ;
0001 310085        LD        SP,PILA   ; STACK POINTER:=34048;(ESPACI
0004 C3B711        JP        BEGIN    ;
0007 00            DEFB     00         ;
0008 0000         DEFW     00         ; interrupcion_ctc0
000A 0000         DEFW     00         ; ctc1
000C 0000         DEFW     00         ; ctc2
000E 0000         DEFW     00         ;
0010 01            N1       DEFB     01 ;
0011 00            DEFB     00         ;
0012 00            DEFB     00         ;
0013 EA            VAR1    DB      0EAH ; VALOR DE FFFFH PARA POSICION
0014 FF            DB      0FFH       ; EN LA TABLA DE DISTANCIAS
0015 00            DB      00H        ;
0016 AB            VAR2    DB      0ABH ; VALOR DE FF5BH PARA POSICION
0017 FF            DB      0FFH       ; EN LA TABLA DE DISTANCIAS
0018 00            DB      00H        ;
0019 54            PI_4    DB      54H  ;
001A 78            DB      78H        ;
001B 00            DB      00H        ;
001C 00            DB      00H        ;
001D 00            DB      00H        ;
001E 00            DB      00H        ;
001F 00            DB      00H        ;
0020 00            DB      00H        ;
0021 00            DB      00H        ;
0022 00            DB      00H        ;
0023

```

INCLUDE SON-INI.ASM ; (***** EXTERNAL *****)

```

0023            INICIA_PTOS ;PROCEDURE INICIA PTOS
0023            ; BEGIN
0023 3E0F          LD        A,0FH      ; PIO A
0025 D301        OUT      (PIO_AC),A    ; MODO 0 SALIDA
0027 3ECF        LD        A,0CFH     ; PIO B
0029 D303        OUT      (PIO_BC),A    ; MODO 3 BIT(IN/OUT)
002B 3E7F        LD        A,01111111B ; I/O REGISTRO CONTROL WORD
002D D303        OUT      (PIO_BC),A    ; PIO_B0=IN
002F            ; PIO_B1=IN
002F            ; PIO_B2=IN
002F            ; PIO_B3=IN
002F            ; PIO_B4=IN
002F            ; PIO_B5=IN
002F            ; PIO_B6=IN
002F            ; PIO_B7=OUT
002F 3E07        LD        A,00000111B ; CONTROL INTERRUPTACIONES
0031 D303        OUT      (PIO_BC),A    ;
0033 3E3F        LD        A,0FFH     ;
0035 D300        OUT      (PIO_A),A    ;
0037 C9         RET                ; END
0038            CTC0_START
0038 3E05        LD        A,00000101B ; PROCEDURE start_ctc0

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0000                ORG      0000H      ;
0000 F3             INICIO    DI        ;
0001 310085        LD        SP,PILA    ; STACK POINTER:=34048;(ESPACI
0004 C3B711        JP        BEGIN    ;
0007 00            DEFB     00        ;
0008 0000        DEFW     00        ;
000A 0000        DEFW     00        ; interrupcion_ctc0
000C 0000        DEFW     00        ; ctc1
000E 0000        DEFW     00        ; ctc2
0010 01            DEFB     01        ;
0011 00            DEFB     00        ;
0012 00            DEFB     00        ;
0013 EA            VAR1     DB     0EAH ; VALOR DE FFFFH PARA POSICION
0014 FF            DB     0FFH        ; EN LA TABLA DE DISTANCIAS
0015 00            DB     00H        ;
0016 A8            VAR2     DB     0A8H ; VALOR DE FF8BH PARA POSICION
0017 FF            DB     0FFH        ; EN LA TABLA DE DISTANCIAS
0018 00            DB     00H        ;
0019 54            PI_4     DB     54H  ;
001A 78            DB     78H        ;
001B 00            DB     00H        ;
001C 00            DB     00H        ;
001D 00            DB     00H        ;
001E 00            DB     00H        ;
001F 00            DB     00H        ;
0020 00            DB     00H        ;
0021 00            DB     00H        ;
0022 00            DB     00H        ;
0023                ;
                INCLUDE SON-INI.ASM ; (***** EXTERNAL *****)

0023                INICIA_PTOS        ;PROCEDURE INICIA PTOS
0023                ; BEGIN
0023 3E0F          LD        A,0FH      ; PIO A
0025 D301          OUT       (PIO_AC),A ; MODO 0 SALIDA
0027 3E0F          LD        A,0CFH      ; PIO B
0029 D303          OUT       (PIO_BC),A ; MODO 3 BIT(IN/OUT)
002B 3E7F          LD        A,01111111B ; I/O REGISTRO CONTROL WORD
002D D303          OUT       (PIO_BC),A ; PIO_B0=IN
002F              ; PIO_B1=IN
002F              ; PIO_B2=IN
002F              ; PIO_B3=IN
002F              ; PIO_B4=IN
002F              ; PIO_B5=IN
002F              ; PIO_B6=IN
002F              ; PIO_B7=OUT
002F              ; CONTROL INTERRUPTACIONES
002F 3E07          LD        A,00000111B ;
0031 D303          OUT       (PIO_BC),A ;
0033 3E0F          LD        A,0FFH      ;
0035 D300          OUT       (PIO_A),A  ;
0037 C9            RET                ; END
0038              ;
0038 3E05          LD        A,00000101B ; PROCEDURE start_ctc0

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0000                ORG      0000H      ;
0000 F3             INICIO   DI         ;
0001 310085        LD        SP,PILA   ; STACK POINTER:=34048;(ESPACI
0004 C3B711        JP        BEGIN    ;
0007 00            DEFB     00         ;
0008 0000          DEFW     00         ;      interrupcion_ctc0
000A 0000          DEFW     00         ;      ctc1
000C 0000          DEFW     00         ;      ctc2
000E 0000          DEFW     00         ;
0010 01            N1       DEFB     01         ;
0011 00            DEFB     00         ;
0012 00            DEFB     00         ;
0013 EA            VAR1    DB        0EAH   ; VALOR DE FFFFH PARA POSICION
0014 FF            DB        0FFH   ; EN LA TABLA DE DISTANCIAS
0015 00            DB        00H         ;
0016 A8            VAR2    DB        0A8H   ; VALOR DE FFASH PARA POSICION
0017 FF            DB        0FFH   ; EN LA TABLA DE DISTANCIAS
0018 00            DB        00H         ;
0019 54            PI_4    DB        54H   ;
001A 78            DB        78H         ;
001B 00            DB        00H         ;
001C 00            DB        00H         ;
001D 00            DB        00H         ;
001E 00            DB        00H         ;
001F 00            DB        00H         ;
0020 00            DB        00H         ;
0021 00            DB        00H         ;
0022 00            DB        00H         ;
0023

```

INCLUDE SON-INI.ASM ; (***** EXTERNAL *****)

```

0023                INICIA_PTOS      ;PROCEDURE INICIA PTOS
0023                ; BEGIN
0023 3E0F            LD        A,0FH     ; PIO A
0025 D301            OUT     (PIO AC),A ; MOD0 0 SALIDA
0027 3ECF            LD        A,0CFH   ; PIO B
0029 D303            OUT     (PIO BC),A ; MOD0 3 BIT(IN/OUT)
002B 3E7F            LD        A,01111111B ; I/O REGISTRO CONTROL WORD
002D D303            OUT     (PIO_BC),A ; PIO_B0=IN
002F                ; PIO_B1=IN
002F                ; PIO_B2=IN
002F                ; PIO_B3=IN
002F                ; PIO_B4=IN
002F                ; PIO_B5=IN
002F                ; PIO_B6=IN
002F                ; PIO_B7=OUT
002F 3E07            LD        A,00000111B ; CONTROL INTERRUPTIONES
0031 D303            OUT     (PIO BC),A ;
0033 3EFF            LD        A,0FFH   ;
0035 D300            OUT     (PIO_A),A  ;
0037 C9             RET                ; END
0038                CTC0_START
0038 3E05            LD        A,00000101B ; PROCEDURE start_ctc0

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

003A D304          OUT   (CTC_0),A      ; BEGIN
003C 3E01          LD    A,01H          ; cte_tiempo:=CLK/256/1(dep
003E D304          OUT   (CTC_0),A      ;
0040 C9            RET                      ; END
0041
0041
0041              STOP_CTC0
0041 3E03          LD    A,00000011B   ; PROCEDURE stop_ctc0
0043 D304          OUT   (CTC_0),A      ; BEGIN
0045 C9            RET                      ; END;
0046
0046              CTC1_START
0046 3E55          LD    A,01010101B   ; PROCEDURE start_ctc1 ;
0048 D305          OUT   (CTC_1),A      ; BEGIN
004A 3EFF          LD    A,0FFH        ;
004C D305          OUT   (CTC_1),A      ;
004E C9            RET                      ; END;
004F
004F              CTC2_START
004F 3E55          LD    A,01010101B   ; PROCEDURE start_ctc2
0051 D306          OUT   (CTC_2),A      ; BEGIN
0053 3EFF          LD    A,0FFH        ; vector_interruccion_ctc
0055 D306          OUT   (CTC_2),A      ;
0057 C9            RET                      ; END;
0058
0058              BLK_MEMORIA
0058 210080         LD    HL,8000H        ; PROCEDURE inicializa_buffer
005B 010001         LD    BC,0100H        ; i:=0100H;
005E 3600         MEM_00  LD    (HL),000H      ; FOR i=BC DOWNT0 0 DO
0060 23            INC    HL            ;
0061 0B            DEC    BC            ; CELDA_MEMORIA[I]:=0;
0062 79            LD    A,C          ;
0063 FE00         CP    00H          ;
0065 20F7         JR    NZ,MEM_00     ;
0067 78            LD    A,B          ;
0068 FE00         CP    00H          ;
006A 20F2         JR    NZ,MEM_00     ;
006C C9            RET                      ; END;
006D
006D              MENSAJE_INI
006D 214B02         LD    HL,AVISO1      ;
0070 CD4402         CALL  PINTA          ; WRITELN('INICIO ');
0073 CDD004         CALL  ESPERA_3        ;
0076 215B02         LD    HL,AVISO2      ;
0079 CD4402         CALL  PINTA          ;
007C CDD004         CALL  ESPERA_3        ;
007F C9            RET                      ;
0080
0080
0080              PREG1
0080 214B03         LD    HL,AVISO26     ; PROCEDURE PREGUNTA_PIPA_?
0083 CD4402         CALL  PINTA          ; BEGIN
0086 CDD004         CALL  ESPERA_3        ; "ES UNA PIPA ?"
0089 215B03         LD    HL,AVISO27     ;
008C CD4402         CALL  PINTA          ; "POR FAVOR SI o NO"
008F CD4B05         MONI  CALL  TECLA          ; READLN(tecla);

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0092 321680      LD      (TECLA1),A      ;
0095 CD2905      CALL    SUELTA TECLA  ;
0098 3A1680      LD      A,(TECLA1)  ;
009B FE0B        CP      OBH          ;
009D 2807        JR      Z,SI          ;           IF TECLA=SI THEN HORIZO
009F FE0D        CP      ODH          ;
00A1 2832        JR      Z,NO          ;           ELSE RET
00A3 C38F00      JP      MONI          ;
00A6             SI          ;
00A6 216B03      LD      HL,AVISO28  ;
00A9 CD4402      CALL    PINTA          ;           " HORIZONTAL "
00AC CD6205      CALL    PERMISO         ;
00AF CD7605      CALL    PULSOS         ;           MANDA_PULSOS_AL_SONAR;
00B2 CD0001      CALL    POS VOL         ;
00B5 CDE605      CALL    LIMBIA CTC      ;           WRITELN( VOLUMEN );
00BB CD4B05      CALL    TECLA          ;
00BB 321680      LD      (TECLA1),A      ;
00BE CD2905      CALL    SUELTA TECLA  ;
00C1 3A1680      LD      A,(TECLA1)  ;
00C4 FE0E        CP      0EH          ;           READLN(tecla);
00C6 2807        JR      Z,OT_MEDIDA  ;
00C8 FE07        CP      07H          ;           IF TECLA=NUEVO THEN INICIA
00CA 2806        JR      Z,NUEVO       ;
00CC C3B800      JP      Y_AHORA        ;           ELSE OTRA_MEDIDA
00CF C3A600      OT MEDIDA  JP      SI          ;
00D2 C38000      NUEVO    JP      PREG1         ;
00D5 1800        NO       JP      FI_FIN        ;
00D7 C9          FI_FIN  RET          ;
00DB             PREG2          ;           PROCEDURE PREGUNTA ESTADO D
00DB 217B03      LD      HL,AVISO29  ;           BEGIN
00DB CD4402      CALL    PINTA          ;
00DE CD4B05      VAC_LLE  CALL    TECLA          ;           VACIO_o_CON_LIQUIDO
00E1 321680      LD      (TECLA1),A      ;
00E4 CD2905      CALL    SUELTA TECLA  ;
00E7 3A1680      LD      A,(TECLA1)  ;
00EA FE0B        CP      OBH          ;
00EC 2807        JR      Z,C_VACIO     ;
00EE FE0D        CP      ODH          ;           READLN(TECLA);
00F0 CAFA00      JP      Z,C_LLENO     ;
00F3 18E9        JR      VAC_LLE      ;           IF TECLA=VACIO THEN VACIO
00F5 CD710D      C_VACIO  CALL    VACIO         ;
00F8 1805        JR      PREG2_FIN     ;           ELSE LLENO
00FA CDA70D      C_LLENO  CALL    LLENO         ;
00FB 1800        JR      PREG2_FIN     ;
00FF C9          PREG2_FIN  RET          ;           End;
0100             POS_VOL          ;           PROCEDURE POSICION DEL VOLU
0100             LD      A,C          ;
0101 321B80      LD      (DESPLZ),A      ;           BEGIN
0104 78          LD      A,B          ;
0105 321C80      LD      (DESPLZ+1),A    ;
0108 211B80      LD      HL,DESPLZ     ;
010B E5          PUSH   HL          ;

```


AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

010C 211600      LD      HL,VAR2      ;
010F CDA404      CALL   RESTA      ;   DESPLZ:=FFA8-DESPLZ;
0112 E1         POP     HL      ;
0113 3A1B80      LD      A,(DESPLZ) ;
0116 6F         LD      L,A      ;
0117 3A1C80      LD      A,(DESPLZ+1);
011A 67         LD      H,A      ;   HL:=DESPLZ;
011B 23         INC     HL      ;
011C 29         ADD     HL,HL     ;
011D 29         ADD     HL,HL     ;   HL:=HL*4;
011E 11AB0E      LD      DE,TAB_VOL  ;   DE:=^TABLA DE VOLUMENES;
0121 19         ADD     HL,DE     ;
0122 7E         LD      A,(HL)    ;   HL:=^VOLUMENES;
0123 32BE80      LD      (VOL_PIPA),A ;   GUARDA EL VOLUMEN DE LA P
0126 E5         PUSH    HL      ;
0127 218B03      LD      HL,AVISO30   ;   "VOL = 99.99 cm"
012A CDCB01      CALL   TRANSF_AVISO ;
012D E1         POP     HL      ;   HL := ^NUM_CUENTA_BCD;
012E CDAB03      CALL   BCDA2B     ;
0131 110C80      LD      DE,AVISO_DISPL+11; WRITELN(VOLUMEN);
0134 CD0702      CALL   EN_AV      ;
0137 23         INC     HL      ;
0138 7E         LD      A,(HL)    ;
0139 328F80      LD      (VOL_PIPA+1),A; GUARDA EL VOLUMEN
013C CDAB03      CALL   BCDA2B     ;
013F 110980      LD      DE,AVISO_DISPL+8;
0142 CD0702      CALL   EN_AV      ;
0145 23         INC     HL      ;
0146 7E         LD      A,(HL)    ;
0147 329080      LD      (VOL_PIPA+2),A; GUARDA EL VOLUMEN
014A CDAB03      CALL   BCDA2B     ;
014D 110780      LD      DE,AVISO_DISPL+6;
0150 CD0702      CALL   EN_AV      ;
0153 23         INC     HL      ;
0154 7E         LD      A,(HL)    ;
0155 329180      LD      (VOL_PIPA+3),A; GUARDA EL VOLUMEN
0158 CDAB03      CALL   BCDA2B     ;
015B 110580      LD      DE,AVISO_DISPL+4;
015E CD0702      CALL   EN_AV      ;
0161 CDA01      CALL   REFRESH_LCD ;
0164 0605      LD      B,05H     ;
0166 DD219280   LD      IX,VOL_PIPA+4 ;
016A 3E00      LD      A,00H     ;
016C DD23      INC     IX      ;   (CEROS EN EL RESTO DEL
016E DD7700      LD      (IX),A    ;   VECTOR DE VOL_PIPA)
0171 10F9      DJNZ   TODOS_CE   ;
0173 C9      RET      ;   END; (POS_TABLA)
0174

                INCLUDE SON-DISP.ASM ; (***** EXTERNAL *****)

0174      INIC_DISPLAY ;
0174 3E30      LD      A,00110000B ; Procedure Inicia_Display
0176 D340      OUT     (LCD_C),A ; Begin

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0178 D340      OUT      (LCD_C),A      ;
017A D340      OUT      (LCD_C),A      ;      function_set;
017C CDA301    CALL     BUSY_FLAG      ;
017F 3E30      LD       A,30H          ;
0181 D340      OUT      (LCD_C),A      ;
0183 CDA301    CALL     BUSY_FLAG      ;
0186 3E08      LD       A,00001000B    ;      display_off;
0188 D340      OUT      (LCD_C),A      ;
018A CDA301    CALL     BUSY_FLAG      ;
018D 3E01      LD       A,00000001B    ;      clear_display;
018F D340      OUT      (LCD_C),A      ;
0191 CDA301    CALL     BUSY_FLAG      ;
0194 3E06      LD       A,00000110B    ;      entry_mode_set
0196 D340      OUT      (LCD_C),A      ;
0198 CDA301    CALL     BUSY_FLAG      ;
019B 3E0E      LD       A,0000110B    ;      display_on
019D D340      OUT      (LCD_C),A      ;
019F CDA301    CALL     BUSY_FLAG      ;
01A2 C9        RET              ;      END;
01A3          BUSY_FLAG
01A3 DB40      IN       A,(LCD_C)      ;
01A5 CB7F      BIT      7,A          ;
01A7 20FA      JR       NZ,BUSY_FLAG  ;
01A9 C9        RET              ;
01AA          REFRESH_LCD
01AA          ;      PROCEDURE refresh_display
01AA C5        PUSH     BC          ;      VAR
01AB 210180    LD       HL,AVISO_DISP;      HL:= aviso_display;
01AE 3E80      LD       A,80H          ;
01B0 D340      OUT      (LCD_C),A      ;
01B2 060F      LD       B,15          ;
01B4 10FE      JGF      DJNZ     JGF      ;
01B6 1610      LD       D,16          ;      contador:=0;(tamaño de di
01B8 CDA301    MANDA    CALL     BUSY_FLAG  ;      ESPERA busy flag
01BB 7E        LD       A,(HL)      ;      FOR contador:= 1 TO D DO
01BC D341      OUT      (LCD),A      ;      manda_caracter_al_dipla
01BE 060F      LD       B,15          ;
01C0 10FE      AGENCIA  DJNZ     AGENCIA  ;
01C2 23        INC      HL          ;      aviso_display:=aviso_disp
01C3 15        DEC      D          ;
01C4 20F2      JR       NZ,MANDA    ;      contador:= contador -1;
01C6 CDA301    CALL     BUSY_FLAG      ;
01C9 C1        POP      BC          ;
01CA C9        RET              ;      END;
01CB          TRANSF_AVISO;
01CB          ;      PROCEDURE transfere_aviso_a
01CB C5        PUSH     BC          ;
01CC 110180    LD       DE,AVISO_DISP;      BEGIN
01CF 01000     LD       BC,16D          ;      contador:=BC;
01D2 EDB0      LDIR     ;      FOR contador=BC DOWNT0 0
01D4 C1        POP      BC          ;
01D5 C9        RET              ;      END;
01D6          CURSOR    ;      PROCEDURE cursor_posicion (a

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

01D6 C5          PUSH    BC          ; BEGIN
01D7 47          LD      B,A          ; localidad_cursor:=B ;
01D8 CDA301     CALL    BUSY_FLAG      ; REPEAT_ leo_status_disp
01DB 78          LD      A,B          ;
01DC D340     OUT    (LCD_C),A      ;
01DE 060F     LD      B,15        ; escribe_en_display_locali
01E0 10FE     DJNZ   IHV          ;
01E2 CDA301     CALL    BUSY_FLAG      ;
01E5 C1        POP     BC          ;
01E6 C9        RET          ; UNTIL (dato_mandado);
01E7          ; END;
01E7          CAR_DISP          ;
01E7 C5          PUSH    BC          ; PROCEDURE despliega_caracter
01E8 3A1280     LD      A,(POS_CAR)    ; despliega caracter en LCD a
01EB CDD601     CALL    CURSOR      ; cambiar de posicion el curs
01EE CDA301     CALL    BUSY_FLAG      ;
01F1 3A1180     LD      A,(CARAC)    ;
01F4 D341     OUT    (LCD),A      ;
01F6 060F     LD      B,15        ;
01F8 10FE     DJNZ   TFX          ;
01FA CDA301     CALL    BUSY_FLAG      ;
01FD DB40     IN      A,(LCD_C)    ;
01FF CBFF     SET     7,A          ;
0201 3D        DEC     A          ;
0202 CDD601     CALL    CURSOR      ;
0205 C1        POP     BC          ;
0206 C9        RET          ;
0207          ;
0207          EN_AV          ; Procedure inserta_datos_en_a
0207 3A1780     LD      A,(DIG_1)    ; DIG1:= DIGITO_BCD_ALTO
020A 12        LD      (DE),A      ;
020B 13        INC     DE          ;
020C 3A1880     LD      A,(DIG_2)    ; DIG_2:= DIGITO_BCD_BAJO
020F 12        LD      (DE),A      ;
0210 13        INC     DE          ;
0211 C9        RET          ;
0212          ;
0212          EN_AV1          ; Procedure inserta_datos_avis
0212 3A1880     LD      A,(DIG_2)    ; DIG_2:= DIGITO_BCD_ALTO
0215 12        LD      (DE),A      ;
0216 1B        DEC     DE          ;
0217 3A1780     LD      A,(DIG_1)    ; DIG_1:= DIGITO_BCD_BAJO
021A 12        LD      (DE),A      ;
021B C9        RET          ;
021C          ;
021C          DISP_VECES      ; PROCEDURE DISPLAY_VECES;
021C 216B02     LD      HL,AVISO4    ; BEGIN
021F CDCB01     CALL    TRANSF_AVISO    ;
0222 110380     LD      DE,AVISO_DISP+2; AVISO:= '#999999';
0225 211580     LD      HL,VECES+2    ;
0228 CDAB03     CALL    BCDA2B      ;
022B CD0702     CALL    EN_AV        ; DE=AVISO_DISP+2 ;
022E 211480     LD      HL,VECES+1    ;
0231 CDAB03     CALL    BCDA2B      ;

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0234 CD0702          CALL EN_AV          ; DE=AVISO_DISP+4 ;
0237 211380         LD HL,VECES        ;
023A CDAB03         CALL BCDA2B        ;
023D CD0702         CALL EN_AV          ; DE=AVISO_DISP+6 ;
0240 CDA A01        CALL REFRESH_LCD   ;
0243 C9             RET                                ; END: ( DISPLAY VECES )
0244
0244
0244 PINTA
0244 CDCB01         CALL TRANSF_AVISO   ;
0247 CDA A01        CALL REFRESH_LCD   ;
024A C9             RET                                ;
024B
; LETREROS
024B 20202A2A AVISO1: DB " *** JAS *** " ;
025B 20534F4E AVISO2: DB " SONAR MIDIENDO " ;
026B 43544120 AVISO4: DB "CTA = " ;
027B 44495354 AVISO7: DB "DIST= ___cm" ;
028B 20202044 AVISO8: DB " D= ???.?cm" ;
029B 4449414D AVISO9: DB "DIAM CILINDRO ?" ;
02AB 4449414D AVISO12: DB "DIAMETRO ___cm" ;
02BB 445E323D AVISO13: DB "D^2= ___cm^2" ;
02CB 413D205F AVISO14: DB "A= ___cm^2" ;
02DB 564F4C3D AVISO16: DB "VOL= ___cm" ;
02EB 52455354 AVISO17: DB "RESTA ___cm" ;
02FB 20202020 AVISO21: DB " VACIO " ;
030B 2020434F AVISO22: DB " CON LIQUIDO ?" ;
031B 414C5455 AVISO23: DB "ALTURA TOTAL ?" ;
032B 20202068 AVISO24: DB " h= ???.?cm" ;
033B 414C5455 AVISO25: DB "ALTURA= ___cm" ;
034B 20455320 AVISO26: DB " ES HORIZONTAL ?" ;
035B 504F5220 AVISO27: DB "POR FAVOR SI-NO" ;
036B 54415155 AVISO28: DB "TAQUE HORIZONTAL" ;
037B 20564143 AVISO29: DB " VACIO o LLENO" ;
038B 564F4C3D AVISO30: DB "VOL= ___cm^3" ;
039B 4E4F2050 AVISO31: DB "NO PUEDO HACERLO" ;
03AB
;
INCLUDE SON-ARI.ASM; (*****EXTERNAL*****)
03AB
03AB BCDA2B ; PROCEDURE BCD-ASCII-EN-2-BYT
03AB ; (convertir dos numeros BCD c
03AB ; en un byte a formato ASCII
03AB ; bytes )
03AB ; VAR (ENTRADA)
03AB ; HL:= ^numeros_bcd_en_byte;
03AB ; (SALIDA)
03AB ; DIG_1:= ASCII_BCD_ALTO;
03AB ; DIG_2:= ASCII_BCD_BAJO;
03AB
03AB D5 PUSH DE ;
03AC 111780 LO DE,DIG_1 ; DE:= ^DIG 1;
03AF 7E LD A,(HL) ; A:= ^NUMEROS_BCD_EN_BYTE;
03B0 4F LD C,A ;
03B1 E6F0 AND OFOH ;
03B3 CB3F SRL A ;
03B5 CB3F SRL A ;

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

03B7 CB3F      SRL      A      ;
03B9 CB3F      SRL      A      ;
03BB C630      ADD      A,30H  ;
03BD 12        LD       (DE),A  ;
03BE 13        INC      DE      ;      DE:= ^DIG_2;
03BF 79        LD       A,C      ;
03C0 E60F      AND      0FH     ;
03C2 C630      ADD      A,30H  ;
03C4 12        LD       (DE),A  ;
03C5 D1        POP      DE      ;
03C6 C9        RET      ;
03C7           ;
03C7           ;      PROCEDURE BINARIO A BCD;
03C7           ; { convierte un byte Binario
03C7           ; a dos bytes dedatos en BCD
03C7           ; reg A:= datos binario;
03C7           ; reg H:= dato BCD byte alto;
03C7           ; reg L:= dato BCD byte bajo;
03C7           ; Calcular digitos de 100's -d
03C7           ; H= Cociente A-residuo
03C7           ; iniciar cociente = -1
03C7 26FF      LD       H,0FFH  ;
03C9 24        INC      H      ;      sumar 1 a cociente
03CA D664      SUB      100     ;      restar 100
03CC 30FB      JR       NC,D100LP ;      saltar si diferencia todavia
03CE C664      ADD      A,100   ;      sumar 100 nuevamente
03D0           ;      Calcular digitos de 10's y d
03D0           ; dividir el residuo de los 10
03D0           ; entre 10..
03D0           ; L= digitos 10's A=digitos 1
03D0 2EFF      LD       L,0FFH  ;      iniciar cociente en -1.
03D2 2C        INC      L      ;      sumar 1 al cociente.
03D3 D60A      SUB      10      ;      restar 10
03D5 30FB      JR       NC,D10LP ;      saltar si la diferencia toda
03D7 C60A      ADD      A,10   ;      sumar nuevamente los ultimos
03D9           ;      Combinar digitos de 1's y 10
03D9 4F        LD       C,A   ;      guardar digitos de 1's en C
03DA 7D        LD       A,L   ;
03DB 07        RLCA      ;      mover 10's al nibble alto de
03DC 07        RLCA      ;
03DD 07        RLCA      ;
03DE 07        RLCA      ;
03DF B1        OR       C      ;      sumar el digito de 1's.
03E0 6F        LD       L,A   ;
03E1 321A80    LD       (DIGIT_B),A ;      PARTE_BAJA
03E4 7C        LD       A,H   ;
03E5 321980    LD       (DIGIT_A),A ;      PARTE_ALTA
03E8 C9        RET      ;
03E9           ;
03E9 78        LD       A,B   ;      PROCEDURE MULTIPLE PRECISION
03EA B7        OR       A      ;      PRUEBA LA LONGITUD DE LOS OP
03EB CA9104    JP       Z,F,MPDMUL ;      SALE SI LA LONGITUD ES CERO
03EE 32A380    LD       (LEN),A ;      GRABA LA LONGITUD
03F1 32A580    LD       (LPCNT),A ;      MALLA DEL CONTADOR=LONGITUD

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

03F4 22A780          LD      (MCADR),HL      ; GRABA LA DIRECCION DEL MULTI
03F7 ED53A980       LD      (MPADR),DE      ; GRABA LA DIRECCION DEL MULTI
03FB 11CD80         LD      DE,MCAND        ; MULTIPLICANDO EN UN BUFFER T
03FE ED53AB80       LD      (NBYTE),DE     ; DE, APUNTA AL MULTIPLICANDO
0402 48             LD      C,B            ; HL, APUNTA AL MULTIPLICANDO
0403 0600           LD      B,0            ; BC=LONGITUD
0405 EDB0           LDIR                     ; MUEVE EL MULTIPLICANDO AL BU
0407                ; LIMPIA EL PRODUCTO PARCIAL,
0407                ; QUE CONSISTE DE LOS BYTES SU
0407                ; COMENZANDO EL PRODUCTO Y LOS
0407                ; BAJOS REMPLAZAN AL MULTIPLIC
0407                ;
0407 2AA780          LD      HL,(MCADR)      ;
040A 3AA380         LD      A,(LEN)        ;
040D CD9204        CALL   ZEROBUF        ; MULTIPLICANDO CERO
0410                ; PRODUCTO CERO
0410 21AD80         LD      HL,PROD        ;
0413 CD9204        CALL   ZEROBUF        ; ARREGLO DE PRODUCTO CERO
0416 LPMUL          ; MALLA DE TODOS LOS BYTES DEL
0416 3E01          LD      A,1            ;
0418 32A480        LD      (DCNT),A        ;
041B                ; INICIA CON EL DIGITO MAS BAJ
041B                ; MALLA DE DOS DIGITOS POR BYT
041B                ; DURANTE EL DIGITO MAS BAJO D
041B                ; DURANTE EL DIGITO MAS ALTO D
041B                ;
041B DLOOPMUL      SUB      A            ; A=0
041B 97           LD      (OVRFLW),A      ; LIMPIA EL BYTE DE OVERFLOW
041C 32A680        LD      A,(DCNT)        ;
041F 3AA480        OR      A            ;
0422 B7           OR      A            ; PRUEBA EL DIGITO MAS BAJO (Z
0423 2AAB80        LD      HL,(NBYTE)      ; TOMA EL SIGUIENTE BYTE
0426 7E           LD      A,(HL)        ;
0427 2004         JR      NZ,DLOOP1M      ; SALTA SI ES EL DIGITO MAS BA
0429 0F           RRCA                     ; ROTA A LA DERECHA 4 BITS EL
042A 0F           RRCA                     ;
042B 0F           RRCA                     ;
042C 0F           RRCA                     ;
042D                ;
042D DLOOP1M      AND      0FH          ; SOLAMENTE GUARDA EL DIGITO C
042D E60F         JR      Z,SDIGIT        ; ROMPE SI EL DIGITO ES CERO
042F 281D         LD      C,A            ; C=DIGITO
0431 4F           LD      A,C            ;
0432                ;
0432 ADDLP          ; SUMA EL MULTIPLICADOR AL PRO
0432 2AA980        LD      HL,(MPADR)      ; HL= DIRECCION DEL MULTIPLICA
0435 11AD80        LD      DE,PROD        ; DE= DIRECCION DEL PRODUCTO
043B 3AA380        LD      A,(LEN)        ;
043B 47           LD      B,A            ; B=LONGITUD
043C B7           OR      A            ; LIMPIA EL CARRY INICIALMENTE
043D                ;
043D 1A           LD      A,(DE)        ; TOMA EL SIGUIENTE BYTE DEL P
043E BE           ADC      A,(HL)        ; SUMA EL SIGUIENTE BYTE DEL M
043F 27           DAA                     ; AJUSTE DECIMAL
0440 12           LD      (DE),A        ; GRABA LA SUMA EN EL PRODUCTO
0441 23           INC      HL            ;
0442 13           INC      DE            ;
0443 10F8         DJNZ   INNERM        ; CONTINUA HASTA QUE TODOS LOS

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0445 3004          JR      NC,DECND      ; SALTA SI NO HAY OVERFLOW DE
0447 21A680       LD      HL,OVRPLW    ; O SI NO INCREMENTA EL BYTE D
044A 34           INC      (HL)        ;
044B             DECND                ;
044B 0D           DEC      C          ;
044C 20E4       JR      NZ,ADDLP    ; CONTINUA HASTA QUE DIGITO=0
044E             ; GRABA EL ULTIMO DIGITO SIGNI
044E             ; PRODUCTO COMO EL SIGUIENTE D
044E             ; DEL MULTIPLICANDO
044E             ;
044E 3AAB80       SDIGIT  LD      A,(PROD)  ; TOMA EL BYTE BAJO DEL PRODUC
0451 E60F       AND      OFH          ;
0453 47         LD      B,A          ; GRABA EN B
0454 3AA480       LD      A,(DCNT)    ;
0457 B7         OR      A           ; PRUEBA PARA EL MAS BAJO DIGI
0458 78         LD      A,B          ; A = SIGUIENTE DIGITO
0459 2004       JR      NZ,SD1      ; SALTA SI ESTA TRABAJANDO EN
045B 0F         RRCA                ; O SI NO MUEVE EL DIGITO A LO
045C 0F         RRCA                ;
045D 0F         RRCA                ;
045E 0F         RRCA                ;
045F             SD1                 ;
045F 2AA780       LD      HL,(MCADR)  ; LUGAR DEL SIGUIENTE DIGITO D
0462 B6         OR      (HL)        ;
0463 77         LD      (HL),A      ;
0464             ; CORRE EL PRODUCTO A LA DERECH
0464 3AA380       LD      A,(LEN)    ;
0467 47         LD      B,A          ; B = LONGITUD
0468 5F         LD      E,A          ;
0469 1600       LD      D,0         ;
046B 21AD80       LD      HL,PROD    ;
046E 19         ADD     HL,DE        ;
046F 3AA680       LD      A,(OVRPLW) ; HL APUNTA ADELANTE DEL FINAL
0472             SHFTLP              ;
0472 2B         DEC     HL           ; DECREMENTA APUNTA AL SIGUIEN
0473 ED67       RRD                ; ROTA EL BYTE DEL PRODUCTO AL
0475 10FB       DJNZ    SHFTLP      ; CONTINUA HASTA TERMINAR
0477             ; CHECA SI ESTA HECHO CON AMBO
0477 21A480       LD      HL,DCNT    ; ESTAMOS EN EL DIGITO MAS BAJ
047A 35         DEC     (HL)        ;
047B 289E       JR      Z,DLOOPMUL  ; SI, HACE EL DIGITO MAS ALTO
047D             ; INCREMENTA AL SIGUIENTE BYTE
047D 2AAB80       LD      HL,(NBYTE) ; INCREMENTA AL SIGUIENTE BYTE
0480 23         INC     HL           ;
0481 22AB80       LD      HL,(NBYTE),HL ;
0484 2AA780       LD      HL,(MCADR) ; INCREMENTA PARA EL SIGUIENTE
0487 23         INC     HL           ;
0488 22A780       LD      HL,(MCADR),HL ;
048B 21A580       LD      HL,LPCNT   ; DECREMENTA EL CONTADOR DE MA
048E 35         DEC     (HL)        ;
048F 2085       JR      NZ,LPMUL    ;
0491 C9         F MPMUL  RET        ;
0492             ZEROBUF             ; RUTINA ZEROBUF .DA CEROS AL

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0492 3600          LD      (HL),0          ; CERO FIRST BYTE
0494 3AA380        LD      A,(LEN)         ;
0497 3D           DEC      A              ;
0498 CAA304        JP      Z,FIN_ZEROBUF ; REGRESA SI ONLY ONE BYTE
049B 54           LD      D,H            ;
049C 5D           LD      E,L            ;
049D 13           INC      DE            ; DE=SEGUNDO BYTE
049E 4F           LD      C,A            ; BC=LONGITUD DEL ARREGLO
049F 0600        LD      B,0            ;
04A1 EDB0        LDIR                     ; LIMPIA EL RESTO DEL BUFER ME
04A3                                     ; PROPAGACION DE CEROS DE UN B
04A3 C9          FIN_ZEROBUF RET          ;
04A4 RESTA                                     ;
04A4 D5          PUSH   DE                ; [RESTA ***H AL VALOR DE LA
04A5 11ED80      LD      DE,MINUENDO     ;
04A8 010200      LD      BC,02H         ;
04AB EDB0        LDIR                     ;
04AD 21ED80      LD      HL,MINUENDO    ;
04B0 111B80      LD      DE,DESPLZ     ;
04B3 0602        LD      B,02H         ;
04B5 7B          LD      A,B            ; [ENTRAR A LA TABLA]
04B6 A7          AND      A              ;
04B7 2808        JR      Z,RESTA_FIN    ; LIMPIA CARRY,PRUEBA EL ACUMU
04B9 EB          EX      DE,HL          ; CAMBIA LOS ARREGLOS .HL ES
04BA LOOP_RES                                     ;
04BA 1A          LD      A,(DE)          ; TOMA EL SIGUIENTE BYTE DEL M
04BB 9E          SBC      A,(HL)        ; SUSTRAE BYTES
04BC 12          LD      (DE),A         ; GRABA DIFERENCIA
04BD 13          INC      DE            ; INCREMENTA EL APUNTADOR DEL
04BE 23          INC      HL            ; INCREMENTA EL APUNTADOR DEL
04BF 10F9        DJNZ   LOOP_RES        ;
04C1 RESTA_FIN                                     ;
04C1 21ED80      LD      HL,MINUENDO    ;
04C4 111B80      LD      DE,DESPLZ     ;
04C7 010200      LD      BC,02H         ;
04CA EDB0        LDIR                     ;
04CC D1          POP      DE            ;
04CD C9          RET                      ;
04CE MPDSUB                                     ;
04CE                                     ; MULTIPLE-PRECISION DECIMAL SU
04CE                                     ; PROCEDURE MULTIPLE-PRECISION
04CE                                     ; { RESTAR 2 ARREGLOS DE BYTES
04CE                                     ; VAR (ENTRADA)
04CE                                     ; HL:= ^MINUENDO;
04CE                                     ; DE:= ^SUSTRAYENDO;
04CE                                     ; B:= Longitud en bytes de ar
04CE                                     ; { array[0]:= byte menos sign
04CE                                     ; { array[B-1]:= byte mas sign
04CE                                     ; (SALIDA)
04CE                                     ; (MINUENDO:= MINUENDO - SUSTR
04CE                                     ; ^HL := ^HL - ^DE
04CE 7B          LD      A,B            ;
04CF B7          OR      A              ; IF LENGTH:= 0 THEN

```


AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

04D0 CADCO4      JP      Z,SAL_SUB      ; ERROR
04D3 EB         EX      DE,HL         ; HL:= SUSTRAYENDO
04D4           ; DE:= MINUENDO
04D4           ; RESTAR 2 DIGITOS A LA VEZ
04D4           ; CARRY:= 0 ; (INICIALMENTE)
04D4           ; B:= LENGTH;
04D4           ; REPEAT
04D4 1A         RESTA2  LD      A,(DE)      ; A:= MINUENDO[DE];
04D5 9E         SBC      A,(HL)         ; A:= MINUENDO[DE] - SUSTRAE
04D6 27         DAA           ; CAMBIA A DECIMAL;
04D7 12         LD      (DE),A        ; MINUENDO[DE]:= A;
04D8 23         INC      HL           ; HL:= HL+1;
04D9 13         INC      DE           ; DE:= DE+1;
04DA           ; B:= B - 1;
04DA 10FB      DJNZ     RESTA2        ; UNTIL B=0;
04DC C9         SAL_SUB  RET          ;
04DD           ;

                INCLUDE SON-SOP.ASM ;(*****EXTERNAL *****)
04DD           ; PROCEDIMIENTOS PARA SOPORTE
04DD D5         ESPERA_3  PUSH  DE      ; PROCEDURE espera_3
04DE E5         PUSH  HL          ; BEGIN
04DF 11FF7F    LD      DE,07FFH     ; cuenta:=ffffh
04E2 2602     LD      H,02H        ; cuenta2:=2
04E4 1B         VUELTA1  DEC      DE          ; REPEAT
04E5 7A         LD      A,D         ; cuenta:=cuenta-1
04E6 FE00     CP      00H          ;
04E8 20FA     JR      NZ,VUELTA1   ;
04EA 7B         LD      A,E         ;
04EB FE00     CP      00H          ;
04ED 20F5     JR      NZ,VUELTA1   ;
04EF 25         DEC      H          ;
04F0 20F2     JR      NZ,VUELTA1   ;
04F2 E1         POP      HL         ; cuenta2:=cuenta2-1
04F3 D1         POP      DE         ; UNTIL cuenta2:=0
04F4 C9         RET              ;
04F5           ;
04F5           ;
04F5 C5         MEDIO_SEG  PUSH  BC      ;
04F6 0E7F    LD      C,07FH       ;
04F8 06FF    LD      B,0FFH       ;
04FA 00         MNB          NOP          ;
04FB 00         MNB          NOP          ;
04FC 10FC    DJNZ     MNB         ;
04FE 0B         DEC      C          ;
04FF 20F7     JR      NZ,LKJ       ;
0501 C1         POP      BC         ;
0502 C9         RET              ;
0503           ;
0503           ;
0503 C5         CUARTO_SEG  PUSH  BC      ;
0504 0E40    LD      C,40H        ;
0506 06FF    LD      B,0FFH       ;
0508 00         MNB          NOP          ;

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0509 00                NOP                ;
050A 10FC             DJNZ             MNB1        ;
050C 0D              DEC              C          ;
050D 20F7           JR              NZ,LKJ1      ;
050F C1             POP              BC         ;
0510 C9             RET                ;
0511                ;
0511                ; PROCEDURE DELAY_33 ms;
0511                ;   CLOCK = 3.072 MHz
0511                ; T = t1+ t2*10+ t3*255*30 + t
0511                ; T = 33 ms;
0511 C5             PUSH             BC         ;
0512 0E1E           LD              C,30        ; t1= 7 ciclos
0514 06FF           LD              B,255      ; t2= 7 ciclos
0516 10FE           DJNZ             D_30      ; t3= 13 ciclos
0518 0D              DEC              C         ; t4= 6 ciclos
0519 20F9           JR              NZ, CARGA_255 ; t5= 12 ciclos
051B C1             POP              BC         ;
051C C9             RET                ;
051D                ;   END; (retardo 30 ms)
051D                ;
051D                ; PROCEDURE DELAY_10 ms;
051D                ;   CLOCK = 3.072 MHz
051D                ; T = t1+ t2*10+ t3*255*10 + t
051D                ; T = 11 ms;
051D C5             PUSH             BC         ;
051E 0E0A           LD              C,10        ; t1= 7 ciclos
0520 06FF           LD              B,255      ; t2= 7 ciclos
0522 10FE           DJNZ             D_10      ; t3= 13 ciclos
0524 0D              DEC              C         ; t4= 6 ciclos
0525 20F9           JR              NZ, CAR_255  ; t5= 12 ciclos
0527 C1             POP              BC         ;
0528 C9             RET                ;
0529                ;   END; (retardo 10 ms)
0529                ;
0529 DB02           SUELTA_TECLA IN A,(PIO_B) ;
052B E60F           AND              OFH          ;
052D FE0F           CP              OFH          ;
052F 20F8           JR              NZ,SUELTA_TECLA ;
0531 CD1105        CALL             DEBOUNCE ;
0534 C9             RET                ;
0535                ;
0535 DB02           LEE                IN A,(PIO_B) ; PROCEDURE LEE
0537 E610           AND              00010000B ; IF A:=1 THEN
0539 2002           JR              NZ,DETENTE ; STOP CTC
053B 18F8           JR              LEE          ; ELSE
053D C9             RET                ; SIGUE CONTANDO
053E DB05           VE_CTA           IN A,(CTC_1) ; PROCEDURE VE LA CUENTA
0540 4F             LD              C,A          ; BEGIN
0541 DB06           IN              A,(CTC_2) ; BC:=CTA
0543 47             LD              B,A          ;
0544 CDE205        CALL             CONVIERTE ; DESPLIEGA BC
0547 CD4100        CALL             STOP_CTCO ;
054A C9             RET                ; END
054B DB02           TECLA           IN A,(PIO_B) ; Read(kbd,tecla);

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

054D E60F          AND    OFH          ;
054F FE0F          CP      OFH          ;
0551 28F8          JR      Z,TECLA     ;
0553 CD1105        CALL   DEBOUNCE     ;
0556 DB02          IN      A,(PIO_B)   ;
0558 E60F          AND    OFH          ;
055A FE0F          CP      OFH          ;
055C 28ED          JR      Z,TECLA     ;
055E CD1105        CALL   DEBOUNCE     ;
0561 C9            RET                    ;
0562                                     ;
0562             PERMISO                                     ;
0562                                     ;
0562             PROCEDURE PERMISO                             ;
0562             ( ESPERA_SENAL_PARA_CONTI                       ;
0562             REPEAT                                           ;
0562 CD4B05          CALL   TECLA                               ;
0565 321680        LD      (TECLA1),A   ;
0568 CD2905        CALL   SUELTA TECLA   ;
056B JA1680        LD      A,(TECLA1)   ;
056E FE0E          CP      0EH          ;
0570 2803          JR      Z,YA SIGUE   ;
0572 C36205        JP      PERMISO     ;
0575 C9            RET                    ;
0576             YA_SIGUE                                     ;
0576             UNTIL PIO_B3=0;{LEE ORDEN                       ;
0576             ;
0576             PULSOS                                       ;
0576 CD4600        CALL   CTC1_START   ;
0579 CD4F00        CALL   CTC2_START   ;
057C 3E00          LD      A,00H          ;
057E D300          OUT   (PIO_A),A   ;
0580 3E01          LD      A,01H          ;
0582 D300          OUT   (PIO_A),A   ;
0584 3E00          LD      A,00H          ;
0586 D300          OUT   (PIO_A),A   ;
0588 CD3800        CALL   CTC0_START   ;
058B CD3505        CALL   LEE -          ;
058E CD3E05        CALL   VE CTA       ;
0591 CD6205        CALL   PERMISO     ;
0594 C9            RET                    ;
0595             ;
0595             MUESTRA                                     ;
0595 CDE701        CALL   CAR DISP     ;
0598 CD4B05        CALL   TECLA       ;
059B 321680        LD      (TECLA1),A   ;
059E CD2905        CALL   SUELTA TECLA   ;
05A1 JA1680        LD      A,(TECLA1)   ;
05A4 FE07          CP      07H          ;
05A6 201B          JR      NZ,ENTER   ;
05A8 JA1180        LD      A,(CARAC)   ;
05AB FE39          CP      39H          ;
05AD 2008          JR      NZ,AUMENTA  ;
05AF 3E30          LD      A,30H          ;
05B1 321180        LD      (CARAC),A   ;
05B4 C39505        JP      MUESTRA     ;
05B7 JA1180        LD      A,(CARAC)   ;
05BA 47            CP      B,A         ;
05BB 04            INC     B           ;
05BB             ;
0595             despliega numero                             ;
0598             lee pio_b                                   ;
0595             ;
0595             if dato_pio=sube then                       ;
0595             ;
0595             if numero=9 then                             ;
0595             ;
0595             numero:=0                                   ;
0595             ;
0595             else                                         ;
0595             ;
0595             numero:numero+

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

05BC 78          LD      A,B          ;
05BD 321180     LD      (CARAC),A        ;
05C0 C39505     JP      MUESTRA          ;           (END IF)
05C3 FE0E     ENTER    CP      0EH          ;
05C5 20CE     JR      NZ,MUESTRA      ;
05C7 CD2905     CALL    SUELTA TECLA      ;
05CA CDE701     CALL    CAR DISP          ;
05CD CD1D05     CALL    D10MS          ;
05D0 JA1180     LD      A,(CARAC)        ;           else {tecla de correc
05D3 D630     SUB     30H          ;           bien:=true;
05D5 C9          RET          ;           until bien;
05D6 3E03     LIMPIA_CTC LD      A,00000011B      ;
05D8 D304     OUT     (CTC_0),A        ;
05DA D305     OUT     (CTC_1),A        ;
05DC D306     OUT     (CTC_2),A        ;
05DE CD5800     CALL    BLK MEMORIA      ;
05E1 C9          RET          ;
05E2 C5          CONVIERTE PUSH   BC          ; PROCEDURE DESPLIEGA_CUENTA
05E3 216B02     LD      HL,AVISO4        ; BEGIN
05E6 110180     LD      DE,AVISO DISP   ;
05E9 CDCB01     CALL    TRANSF AVISO    ; LETRERO = 'CTA-NNNN';
05EC 110A80     LD      DE,AVISO_DISP+9;
05EF 79          LD      A,C          ;
05F0 E60F     AND     A,OFH          ;
05F2 CD3006     CALL    ASCII          ; BUSCA EL EQUIVALENTE EN A
05F5 1B          DEC     DE          ;
05F6 79          LD      A,C          ;
05F7 E6F0     AND     A,OF0H          ; BUSCA NIBBLE ALTO DE C
05F9 CB3F     SRL     A          ;
05FB CB3F     SRL     A          ;
05FD CB3F     SRL     A          ;
05FF CB3F     SRL     A          ;
0601 CD3006     CALL    ASCII          ; BUSCA EL EQUIVALENTE EN A
0604 1B          DEC     DE          ;
0605 78          LD      A,B          ;
0606 E60F     AND     A,OFH          ; BUSCA NIBBLE BAJO DE B
0608 CD3006     CALL    ASCII          ;
060B 1B          DEC     DE          ; BUSCA EL EQUIVALENTE EN A
060C 78          LD      A,B          ;
060D E6F0     AND     A,OF0H          ;
060F CB3F     SRL     A          ;
0611 CB3F     SRL     A          ; BUSCA NIBBLE ALTO DE B
0613 CB3F     SRL     A          ;
0615 CB3F     SRL     A          ;
0617 CD3006     CALL    ASCII          ; BUSCA EL EQUIVALENTE EN A
061A CDAA01     CALL    REFRESH_LCD   ; WRITE EL NUMERO EN HEXADC
061D C1          POP     BC          ;
061E C9          RET          ;
061F 30     TABLA  DB     "0"          ;
0620 31     DB     "1"          ;
0621 32     DB     "2"          ;
0622 33     DB     "3"          ;

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```
0623 34          DB      "A"          ;
0624 35          DB      "S"          ;
0625 36          DB      "6"          ;
0626 37          DB      "7"          ;
0627 38          DB      "8"          ;
0628 39          DB      "9"          ;
0629 41          DB      "A"          ;
062A 42          DB      "B"          ;
062B 43          DB      "C"          ;
062C 44          DB      "D"          ;
062D 45          DB      "E"          ;
062E 46          DB      "F"          ;
062F 58          DB      "X"          ;
0630 C5          ASCII    PUSH      BC          ; PROCEDURE CAMBIA A ASCII
0631 E5          ASCII    PUSH      HL          ;
0632 211F06      LD      HL, TABLA          ;
0635 FE00        CP      00H              ;
0637 2B04        JR      Z, NUM_ASCII       ;
0639 47          LD      B, A              ;
063A 23          REGRESA INC      HL          ; POSICIONA EN TABLA
063B 10FD        DJNZ    REGRESA          ;
063D 7E          NUM_ASCII LD      A, (HL)   ; BUSCA SU CORRESPONDIENTE E
063E 12          LD      (DE), A          ;
063F E1          POP     HL              ;
0640 C1          POP     BC              ;
0641 C9          RET                     ;
0642 CDFD0B      PRI_DIAM CALL    SET_DIAM          ; BEGIN
0645 CD500C      CALL    DESP_DIAM         ;
0648 CD6205      CALL    PERMISO          ; READ(DIAMETRO)
064B CD7A0C      CALL    GUARD_DIAM        ; WRITE(DIAMETRO)
064E C9          RET                     ; END;

064F                                     ;
                                INCLUDE JJ2.ASM ; (*****EXTERNAL *****)
064F                                     ;
TAB_DIST                               ; Inicio de Tabla
064F 00          DB      00H              ; Cta=65514 X= 0.00 cm
0650 00          DB      00H              ;
0651 20          DB      20H              ; Cta=65513 X= 0.20 cm
0652 00          DB      00H              ;
0653 40          DB      40H              ; Cta=65512 X= 0.40 cm
0654 00          DB      00H              ;
0655 60          DB      60H              ; Cta=65511 X= 0.60 cm
0656 00          DB      00H              ;
0657 80          DB      80H              ; Cta=65510 X= 0.80 cm
0658 00          DB      00H              ;
0659 00          DB      00H              ; Cta=65509 X= 1.00 cm
065A 01          DB      01H              ;
065B 20          DB      20H              ; Cta=65508 X= 1.20 cm
065C 01          DB      01H              ;
065D 40          DB      40H              ; Cta=65507 X= 1.40 cm
065E 01          DB      01H              ;
065F 60          DB      60H              ; Cta=65506 X= 1.60 cm
0660 01          DB      01H              ;
```

AVOCET SYSTEMS 280 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0B59 86          DB      86H          ; Cta=64869   X= 98.86 cm
0B5A 98          DB      98H          ;
0B5B 00          DB      00H          ; Cta=64868   X= 99.00 cm
0B5C 99          DB      99H          ;
0B5D 14          DB      14H          ; Cta=64867   X= 99.14 cm
0B5E 99          DB      99H          ;
0B5F 29          DB      29H          ; Cta=64866   X= 99.29 cm
0B60 99          DB      99H          ;
0B61 43          DB      43H          ; Cta=64865   X= 99.43 cm
0B62 99          DB      99H          ;
0B63 57          DB      57H          ; Cta=64864   X= 99.57 cm
0B64 99          DB      99H          ;
0B65 71          DB      71H          ; Cta=64863   X= 99.71 cm
0B66 99          DB      99H          ;
0B67 86          DB      86H          ; Cta=64862   X= 99.86 cm
0B68 99          DB      99H          ;
0B69 00          DB      00H          ; Cta=64861   X=100.00 cm
0B6A 00          DB      00H          ;

```

0B6B

```

                                ;
                                ; *****EXTERNAL *****
                                ;
0B6B          INCLUDE SON-TAMA.ASM ;(*****EXTERNAL *****)
POS_TABLA    LD      A,C          ; PROCEDURE POSICIONA_EN_TABLA
0B6B 79        LD      (DESPLZ),A  ; BEGIN
0B6C 321B80   LD      A,B          ;
0B6F 78        LD      (DESPLZ+1),A ;
0B70 321C80   LD      HL,DESPLZ    ;
0B73 211B80   LD      HL,DESPLZ    ;
0B76 E5       PUSH   HL           ;
0B77 211300   LD      HL,VARI      ;
0B7A CDA404   CALL   RESTA        ; DESPLZ:=FFEA-DESPLZ;
0B7D E1       POP    HL           ;
0B7E 3A1B80   LD      A,(DESPLZ)    ;
0B81 6F       LD      L,A          ;
0B82 3A1C80   LD      A,(DESPLZ+1) ;
0B85 67       LD      H,A          ; HL:=DESPLZ;
0B86 29       ADD   HL,HL          ; HL:=HL*2;
0B87 114F06   LD      DE,TAB_DIST    ; DE:=^TABLA DE DISTANCIAS;
0B8A 19       ADD   HL,DE          ;
0B8B 7E       LD      A,(HL)      ; HL:=^DISTANCIAS;
0B8C 321E80   LD      (DISTANCIA),A      ; GUARDA LA DISTANCIA
0B8F E5       PUSH   HL           ;
0B90 217B02   LD      HL,AVISO7       ; "DIST= 99.99 cm"
0B93 CDCB01   CALL   TRANSF_AVISO  ;
0B96 E1       POP    HL           ; HL := ^NUM_CUENTA_BCD;
0B97 CDAB03   CALL   BCDA2B         ;
0B9A 110C80   LD      DE,AVISO_DISPL+11;  Writeln(DISTANCIA);
0B9D CD0702   CALL   EN_AV          ;
0BA0 23       INC   HL            ;
0BA1 7E       LD      A,(HL)      ;
0BA2 321F80   LD      (DISTANCIA+1),A ; GUARDA LA DISTANCIA
0BA5 CDAB03   CALL   BCDA2B         ;
0BAB 110980   LD      DE,AVISO_DISPL+8;
0BAB CD0702   CALL   EN_AV          ;

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

OBAB CDA01          CALL REFRESH_LCD ;
OBBI 0605          LD B,05H ;
OBBJ DD211F80      LD IX,DISTANCIA+1;
OBBI 3E00          LD A,00H ;
OBBI DD23          DIS_CEROS INC IX ; (CEROS EN EL RESTO DEL
OBBI DD7700        LD (IX),A ; VECTOR DE DISTANCIA)
OBBI 10F9          DPNZ DIS_CEROS ;
OBBI C9           RET ; END;(POS_TABLA)

OBC1              ANDROME ; PROCEDURE OPERACIONES_SIGUI
OBC1 213E80        LD HL,DIAM1 ; BEGIN
OBC4 114E80        LD DE,AY1 ;
OBC7 010700        LD BC,07H ;
OBBA EDB0          LDIR ; (REALIZA TODAS LAS OPERAC
OBCB 213E80        LD HL,DIAM1 ; (PARA EL CALCULO DEL VOLU
OBCF 114E80        LD DE,AY1 ;
OBD2 0608          LD B,08H ;
OBD4 CDE903        CALL MPDMUL ;
OBD7 CDA10C        CALL D_CUADRADO ; D^2:=D*D;
OBDA CD6205        CALL PERMISO ;
OBD8 213E80        LD HL,DIAM1 ;
OBE0 111900        LD DE,PI_4 ;
OBE3 0609          LD B,09H ;
OBE5 CDE903        CALL MPDMUL ;
OBE8 CDD70C        CALL AREA ; AREA:=(D^2)*(PI/4);
OBEB CD6205        CALL PERMISO ;
OBEF 213E80        LD HL,DIAM1 ;
OBF1 111E80        LD DE,DISTANCIA ;
OBF4 060A          LD B,0AH ;
OBF6 CDE903        CALL MPDMUL ; VOLUMEN:=AREA*DISTANCIA;
OBF9 CD0D0D        CALL VOLUMEN ;
OBF C9           RET ;
OBF0              INCLUDE SON-DIAM.ASM ;(*****EXTERNAL *****)

OBF0              SET_DIAM ; PROCEDURE DIAMETRO
OBF0              LD HL,AVISO9 ; BEGIN
OC00 CD4402        CALL PINTA ;
OC03 CD6205        CALL PERMISO ; Writeln('DIAM CILINDRO?');
OC06 218B02        LD HL,AVISO8 ; Writeln(' ?? ?? cm ');
OC09 CD4402        CALL PINTA ;
OC0C CDF504        CALL MEDIO_SEG ;
OC0F 3E8B          LD A,8BH ;
OC11 321280        LD (POS_CAR),A ;
OC14 3E30          LD A,30H ;
OC16 321180        LD (CARAC),A ;
OC19 CD9505        CALL MUESTRA ; Readln( Diametro[0] );
OC1C 322E80        LD (DIAMETRO),A ;
OC1F 3E8A          LD A,8AH ;
OC21 321280        LD (POS_CAR),A ;
OC24 3E30          LD A,30H ;
OC26 321180        LD (CARAC),A ;
OC29 CD9505        CALL MUESTRA ; Readln( Diametro[1] );

```


ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```
OC9D 323F80          LD      (DIAM1+1),A      ;
OCA0 C9             RET      ;
OCA1                D_CUADRADO
OCA1 21BB02          LD      HL,AVISO13      ;
OCA4 CDCB01          CALL     TRANSF_AVI      ; writeln('D*2=_____.____')
OCA7 11080          LD      DE,AVISO_DISP+15;
OCAA 213E80          LD      HL,DIAM1      ;
OCAD CDAB03          CALL     BCDA2B      ;
OCB0 CD1202          CALL     EN_AV1      ;
OCB3 213F80          LD      HL,DIAM1+1      ; writeln(diam1_cuad:10:4)
OCB6 CDAB03          CALL     BCDA2B      ;
OCB9 1B             DEC     DE      ; (*DESPLIEGA EL NUMER
OCBA CD1202          CALL     EN_AV1      ; CORRESPONDIENTE AL
OCBD 110B80          LD      DE,AVISO_DISP+10; OPERACION REALIZAD
OCC0 214080          LD      HL,DIAM1+2      ;
OCC3 CDAB03          CALL     BCDA2B      ;
OCC6 CD1202          CALL     EN_AV1      ;
OCC9 214180          LD      HL,DIAM1+3      ;
OCCC CDAB03          CALL     BCDA2B      ;
OCCF 1B             DEC     DE      ;
OCD0 CD1202          CALL     EN_AV1      ;
OCD3 CDAA01          CALL     REFRESH_LCD   ;
OCD6 C9             RET      ;
OCD7                AREA
OCD7 21CB02          LD      HL,AVISO14      ;
OCDA CDCB01          CALL     TRANSF_AVI      ; writeln('A =____.____cm*2')
OCDD 110C80          LD      DE,AVISO_DISP+11;
OCE0 214080          LD      HL,DIAM1+2      ;
OCE3 CDAB03          CALL     BCDA2B      ;
OCE6 CD1202          CALL     EN_AV1      ;
OCE9 214180          LD      HL,DIAM1+3      ;
OCEC CDAB03          CALL     BCDA2B      ; writeln(área:8:4);
OCEF 1B             DEC     DE      ;
OCF0 CD1202          CALL     EN_AV1      ;
OCF3 110780          LD      DE,AVISO_DISP+6;
OCF6 214280          LD      HL,DIAM1+4      ;
OCF9 CDAB03          CALL     BCDA2B      ; (*DESPLIEGA EL NUME
OCFC CD1202          CALL     EN_AV1      ; CORRESPONDIENTE A
OCFF 214380          LD      HL,DIAM1+5      ; OPERACION REALIZA
OD02 CDAB03          CALL     BCDA2B      ;
OD05 1B             DEC     DE      ;
OD06 CD1202          CALL     EN_AV1      ;
OD09 CDAA01          CALL     REFRESH_LCD   ;
OD0C C9             RET      ;
OD0D                VOLUMEN
OD0D 21DB02          LD      HL,AVISO16      ;
OD10 CDCB01          CALL     TRANSF_AVI      ; writeln('V=____.____cm*3')
OD13 111080          LD      DE,AVISO_DISP+15;
OD16 214180          LD      HL,DIAM1+3      ;
OD19 CDAB03          CALL     BCDA2B      ;
OD1C CD1202          CALL     EN_AV1      ;
OD1F 214280          LD      HL,DIAM1+4      ; writeln(volumen:8:4);
OD22 CDAB03          CALL     BCDA2B      ;
```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

0D25 1B          DEC      DE          ;
0D26 CD1202     CALL    EN_AV1      ;
0D29 110B80     LD      DE,AVISO_DISP+10; (*DESPLIEGA EL NUME
0D2C 214380     LD      HL,DIAM1+5  ;      CORRESPONDIENTE A
0D2F CDAB03     CALL    BCDA2B     ;      OPERACION REALIZA
0D32 CD1202     CALL    EN_AV1     ;
0D35 214480     LD      HL,DIAM1+6  ;
0D38 CDAB03     CALL    BCDA2B     ;
0D3B 1B         DEC      DE          ;
0D3C CD1202     CALL    EN_AV1     ;
0D3F 110780     LD      DE,AVISO_DISP+6;
0D42 214580     LD      HL,DIAM1+7  ;
0D45 CDAB03     CALL    BCDA2B     ;
0D48 CD1202     CALL    EN_AV1     ;
0D4B CDAA01     CALL    REFRESH_LCD ;
0D4E C9         RET          ;
0D4F           ;
0D50           ;
0D51           ;
0D52 21EB02     LD      HL,AVISO17  ;
0D55 110D80     CALL    TRANSF_AVISO ; writeln('RESTA _____.cm
0D58 211E80     LD      HL,DISTANCIA ;
0D5B CDAB03     CALL    BCDA2B     ;
0D5E CD1202     CALL    EN_AV1     ;
0D61 110A80     LD      DE,AVISO_DISP+9;
0D64 211F80     LD      HL,DISTANCIA+1; writeln(resta:6:2);
0D67 CDAB03     CALL    BCDA2B     ;
0D6A CD1202     CALL    EN_AV1     ;
0D6D CDAA01     CALL    REFRESH_LCD ;
0D70 C9         RET          ;
0D71           ;
0D72           ;
0D73           ;
0D74 21FB02     LD      HL,AVISO21  ; PROCEDURE RECIPIENTE VACIO
0D77 CD4402     CALL    PINTA      ; Begin
0D7A CD6205     CALL    PERMISO    ; writeln(' VACIO ');
0D7D CD7605     CALL    PRI_DIAM   ; readln(diametro);
0D80 CD6B0B     CALL    PULSOS     ; envia_pulsos_a_sonar;
0D83 CD6205     CALL    POS_TABLA  ; encuentra_distancias;
0D86 CDC10B     CALL    PERMISO    ;
0D89 CDD605     CALL    ANDROME    ; realiza_operaciones;
0D8C CD4B05     CALL    LIMPIA_CTC ; limpia_sistema;
0D8F 321680     LD      (TECLA1),A ;
0D92 CD2905     CALL    SUELTA_TECLA ;
0D95 3A1680     LD      A,(TECLA1) ;
0D98 FE0E       CP      0EH        ; readln;
0D9A 2807       JR      Z,MIDE  ;
0D9C FE07       CP      07H        ;
0D9E 2806       JR      Z,NEW  ;
0DA0 C38COD     JP      QUE_MAS   ;
0DA3 C3710D     JP      VACIO     ;
0DA6 C9         NEW     RET          ;
0DA7           ;
0DA8           ;
0DA9           ;
0DA7           ;
0DA7           ;
0DA7 210B03     LD      HL,AVISO22  ; PROCEDURE RECIPIENTE LLENO
                                ; * TIENE LIQUIDO *

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASH

```
ODAA CD4402      CALL PINTA      ;
ODAD CD6205      CALL PERMISO     ;
ODBO 211B03      LD HL,AVISO23  ; "ENTRE ALTURA MAXIMA"
ODB3 CD4402      CALL PINTA      ;
ODB6 CD6205      CALL PERMISO     ;
ODB9 212B03      LD HL,AVISO24  ; " h= ?? ?? cm  "
ODBC CD4402      CALL PINTA      ;
ODBF CDF504      CALL MEDIO SEG ;
ODC2 3E8B        LD A,8BH      ; LEE_LA _MAXIMA _ALTURA
ODC4 321280      LD (POS CAR),A ;
ODC7 3E30        LD A,30H      ;
ODC9 321180      LD (CARAC),A  ;
ODCC CD9505      CALL MUESTRA   ;
ODCF 326E80      LD (ALTURA),A ; ALMACENA ALTURA
ODD2 3E8A        LD A,8AH      ;
ODD4 321280      LD (POS CAR),A ; CADA UNO DE LOS 4 DIGI
ODD7 3E30        LD A,30H      ;
ODD9 321180      LD (CARAC),A  ; DE LA ALTURA
ODDC CD9505      CALL MUESTRA   ;
ODDF 326F80      LD (ALTURA+1),A ;
ODE2 3E88        LD A,88H      ;
ODE4 321280      LD (POS CAR),A ;
ODE7 3E30        LD A,30H      ;
ODE9 321180      LD (CARAC),A  ;
ODEC CD9505      CALL MUESTRA   ;
ODEF 327080      LD (ALTURA+2),A ;
ODF2 3E87        LD A,87H      ;
ODF4 321280      LD (POS CAR),A ;
ODF7 3E30        LD A,30H      ;
ODF9 321180      LD (CARAC),A  ;
ODFC CD9505      CALL MUESTRA   ;
ODFF 327180      LD (ALTURA+3),A ;
OE02 213B03      LD HL,AVISO25  ; "ALTURA= __. __cm"
OE05 CDCB01      CALL TRANSF AVISO ;
OE08 110E80      LD DE,AVISO_DISP+13;
OE0B 216E80      LD HL,ALTURA  ; DESPLIEGA LA ALTURA
OE0E 7E          LD A,(HL)     ; QUE SE INTRODUJO,
OE0F C630        ADD 30H        ; PARA CORROBORAR
OE11 12          LD (DE),A     ;
OE12 1B          DEC DE        ;
OE13 23          INC HL        ;
OE14 7E          LD A,(HL)     ;
OE15 C630        ADD 30H        ;
OE17 12          LD (DE),A     ;
OE18 23          INC HL        ;
OE19 110B80      LD DE,AVISO_DISP+10;
OE1C 7E          LD A,(HL)     ;
OE1D C630        ADD 30H        ;
OE1F 12          LD (DE),A     ;
OE20 1B          DEC DE        ;
OE21 23          INC HL        ;
OE22 7E          LD A,(HL)     ;
OE23 C630        ADD 30H        ;
```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```

OE25 12          LD      (DE),A      ;
OE26 CDA01      CALL    REFRESH_LCD ;
OE29 3A6E80     LD      A,(ALTURA) ; SEPARA Y GUARDA LA ALTUR
OE2C 47         LD      B,A      ;
OE2D 3A6F80     LD      A,(ALTURA+1) ; EN DOS BYTES
OE30 CB27      SLA     A      ;
OE32 CB27      SLA     A      ; EN BCD
OE34 CB27      SLA     A      ;
OE36 CB27      SLA     A      ;
OE38 80        ADD     A,B      ;
OE39 327E80     LD      (ALT1),A ;
OE3C 3A7080     LD      A,(ALTURA+2) ;
OE3F 47         LD      B,A      ; FOR I:=1 TO 2 DO
OE40 3A7180     LD      A,(ALTURA+3) ;
OE43 CB27      SLA     A      ;
OE45 CB27      SLA     A      ; ALTURA(I):=ALT(I)
OE47 CB27      SLA     A      ;
OE49 CB27      SLA     A      ;
OE4B 80        ADD     A,B      ;
OE4C 327F80     LD      (ALT1+1),A ;
OE4F D60B      SUB     0BH      ; IF ALTURA<=10 THEN ENGA0;
OE51 FA00E     JP      M,ENGANO ;
OE54 CD6205     CALL    PERMISO ;
OE57 CD4206     CALL    PRI DIAM ; ESTABLECE EL DIAMETRO
OE5A CD7605     CALL    PULSOS ; MANDA SENAL AL SONAR
OE5D CD6B0B     CALL    POS TABLA ; BUSCA LA DISTANCIA
OE60 CD6205     CALL    PERMISO ;
OE63 217E80     LD      HL,ALT1 ;
OE66 111E80     LD      DE,DISTANCIA ; RESTA LA DISTANCIA
OE69 0608      LD      B,0BH ; A LA ALTURA
OE6B CDCE04     CALL    MPDSUB ;
OE6E 217E80     LD      HL,ALT1 ;
OE71 111E80     LD      DE,DISTANCIA ; DIST:=DIST-ALT1
OE74 010700     LD      BC,07H ;
OE77 EDB0      LDIR ;
OE79 CD4F0D     CALL    RESULTADO ; DESPLIEGA LA RESTA
OE7C CD6205     CALL    PERMISO ;
OE7F CDC10B     CALL    ANDROME ; REALIZA CALCULOS
OE82 CDD605     CALL    LIMPIA CTC ;
OE85 CD4B05     CALL    TECLA ;
OE88 321680     LD      (TECLA1),A ;
OE8B CD9P05     CALL    SUELTA TECLA ;
OE8E 3A1680     LD      A,(TECLA1) ; READLN(TECLA);
OE91 FE0E      CP      0EH ;
OE93 2807      JR      Z,MIKI ; IF TECLA:=0E THEN OTRA M
OE95 FE07      CP      07H ;
OE97 2806      JR      Z,F_LLENO ; ELSE
OE99 C38800     JP      Y AHORA ;
OE9C C3A70D     JP      MIKI ; GOTO INICIO
OE9F C9        RET ;
OEAO           ;
OEAO 219B03     ENGANO LD      HL,AVISO31 ; MANDA LETRERO DE
OEAA 3          CALL    PINTA ;

```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```
OEAG CDDD04          CALL  ESPERA 3      ;   OPERACION IRREALIZABLE.
OEAG 18F4          JR    F_LLENO      ;
OEAB

                INCLUDE JJV.ASM ;(*****EXTERNAL *****)
OEAB                TAB_VOL          ; Tabla de valores para
OEAB                DEFB 70H          ;
OEAC 06          DEFB 06H          ;   CTA=65448 VOL= 28706.
OEAD 87          DEFB 87H          ;
OEAE 02          DEFB 02H          ;
OEAF 56          DEFB 56H          ;   CTA=65449 VOL= 28562.
OEB0 62          DEFB 62H          ;
OEB1 85          DEFB 85H          ;
OEB2 02          DEFB 02H          ;
OEB3 60          DEFB 60H          ;   CTA=65448 VOL= 28506.
OEB4 06          DEFB 06H          ;
OEB5 85          DEFB 85H          ;
OEB6 02          DEFB 02H          ;
OEB7 96          DEFB 96H          ;   CTA=65447 VOL= 28444.
OEB8 44          DEFB 44H          ;
OEB9 84          DEFB 84H          ;
OEBA 02          DEFB 02H          ;
OEBB 19          DEFB 19H          ;   CTA=65446 VOL= 28378.
OEBC 78          DEFB 78H          ;
OEBD 83          DEFB 83H          ;
OEBE 02          DEFB 02H          ;
OEBF 70          DEFB 70H          ;   CTA=65445 VOL= 28306.
OEC0 06          DEFB 06H          ;
OEC1 83          DEFB 83H          ;
OEC2 02          DEFB 02H          ;
OEC3 83          DEFB 83H          ;   CTA=65444 VOL= 28230.
OEC4 30          DEFB 30H          ;
11AC 35          DEFB 35H          ;
11AD 00          DEFB 00H          ;
11AE 00          DEFB 00H          ;
11AF 15          DEFB 15H          ;   CTA=65257 VOL= 8.
11B0 08          DEFB 08H          ;
11B1 00          DEFB 00H          ;
11B2 00          DEFB 00H          ;
11B3 00          DEFB 00H          ;   CTA=65256 VOL= 0.
11B4 00          DEFB 00H          ;
11B5 00          DEFB 00H          ;
11B6 00          DEFB 00H          ;

11B7                ;
11B7                ;   ; (**** PROGRAMA PRINCIPAL ****)
11B7                ;
11B7 3E00          BEGIN          LD  A,00H          ; BEGIN
11B9 ED47          LD  I,A          ; parte alta_vector_int = 00
11BB ED5E          IM  2          ; modo InterRupcion = 2;
11BD CD2300        CALL INICIA_PTOS ; inicializacion_sistema
```

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #01002

SOURCE FILE NAME: SON-BU.ASM

```
11C0 CD7401          CALL INIC_DISPLAY ; inicializa_display_LCD
11C1 CD5800          CALL BLK_MEMORIA  ; inicializa_buffer
11C6 CD6000          CALL MENSAJE_INI ; letrero de inicio
11C9 CD8000  PRINCIPIO CALL PREG1         ;
11CC CDD800          CALL PREG2         ;
11CF 18F8           JR   PRINCIPIO      ;
0000                END                ;
```

***** NO ERRORS DETECTED *****

APENDICE B

PROGRAMA EN LENGUAJE TURBOPASCAL PARA

GENERAR TABLA DE DISTANCIAS

0**(*

Programa para generar una tabla de valores de distancia en cm interpolando a partir de otra tabla obtenida en laboratorio para calibrar el sonar.

Los resultados se envían a un archivo con formato para ser incluido en programas para el cross-assembler Z80 de Avocet Inc.

Archivo de entrada: BJ1.TAB u otro con formato similar

Formato:

```

                                65515  99.99
                                ^      ^
Columna 1---|                    |---Columna 12
                                [DECIMAL] [cm]
```

Archivo de salida : PONER: NOMBRE.ASM

Nota : el archivo de salida debe ser editado con WORDSTAR para eliminar una línea de caracteres de control que agrega al final del archivo el programa de PASCAL. Es un error generado por TURBOPASCAL al procesar archivos en formato texto.

*)

```
program distancias_con_sonar;
uses printer;
```

TYPE

```
ST2 = STRING[2];
```

var

```

arch1      : Text; { Archivo con formato en forma de tabla }
arch2      : Text; { Archivo con formato en ensamblador }
i          : integer;
línea      : string[80];
Vec_cuenta : array [0..1000] of longint;
Vec_dist   : array [0..1000] of real;
x_vec      : array [0..1000] of real;
cta_arr    : array [0..1000] of longint;
distancia_car : string[5];
dist_x     : real;
cuenta_car : string[6];
j,i_fin    : integer;
nom_arch   : string[30];
nom_arch2  : string[30];
puntos     : integer;
primer_dato_car : string[5];
dato_inicial : longint;
ok         : boolean;
error      : integer;
p_c       : string[1];
lis       : char;
```

```
FUNCTION HEXA2(X:REAL;POS:INTEGER):ST2;
```

```
{
  { Extrae 2 dígitos de la variable real para convertirla en formato BCD }
  { Primero se llama para extraer 2 decimales, después los dígitos de }
  { unidades y decenas y por último de centenas y unidades de millar }
}
```

var

```

hexa : string[7];
h_bajo : string[1];
h_alto : string[1];
```



```

begin
  STR(X:7:2,HEXA);
  H_ALTO:=COPY(HEXA,POS,1);
  H_BAJO:=COPY(HEXA,POS+1,1);
  IF H_ALTO=' ' THEN H_ALTO:='0';
  IF H_BAJO=' ' THEN H_BAJO:='0';
  HEXA2:=H_ALTO+H_BAJO;
end; (HEXA2)

PROCEDURE LEE_ARCHIVO;
{
  Procedimiento para leer archivo de Tabla de Conversion
}
BEGIN
  i:=-1;
  while not eof(arch1) do
  begin
    readln(arch1,linea);
    p_c:=copy(linea,1,1);
    if p_c<>';' then
      begin
        if p_c='+' then
          begin
            primer_dato_car := copy(linea,4,5);
            Val(primer_dato_car,dato_inicial,error);
            if error<>0 then writeln('Error en valor de dato inicial !!!');
          end
        else
          begin
            i:=i+1;
            cuenta_car := copy(linea,1,6);
            distancia_car:= copy(linea,9,5);
            Val(cuenta_car,vec_cuenta[i],error);
            if error<>0 then writeln('Error en datos de entrada en la linea ',i:3);
            Val(distancia_car,vec_dist[i],error);
            if error<>0 then writeln('Error en datos de entrada en la linea ',i:3);
          end;
        end;
      end;
    i_fin:=i;
    close(arch1);
  END; (LEE_ARCHIVOS)

PROCEDURE ESCRIBE_EN_ARCHIVO;
{
  Procedimiento para escribir datos de conversion en archivos
}
var ll : integer;
BEGIN
  assign(arch2,nom_arch2);
  rewrite(arch2);
  writeln(arch2,'TAB DIST ', ' ':29,'; Inicio de Tabla ');
  for i:=1 to puntos-1 do
  begin
    dist_x:=x_vec[i];
    writeln(ARCH2,' ':16,'DB ',HEXA2(dist_x,6),'H',
    ' ':10,'; cta= ',cta_arr[i]:5,' X=',dist_x:6:2,' cm');
    writeln(ARCH2,' ':16,'DB ',HEXA2(dist_x,3),'H', ' ':10,';');
  end;
  close(arch2);
  writeln;
  writeln;
  writeln(' Archivo de conversion en: ',nom_arch2);
  WRITELN;
  WRITE('Oprima RETURN para continuar');
  readln;

```

```

END;

procedure realiza_calculos;
var
  cx      : longint;
  x1,x2,x3 : longint;
  y1,y2,y3 : real;
  a       : real;
begin
  ( extrapolacion )
  puntos := 1;
  x2 := vec_cuenta[0];
  x1 := vec_cuenta[1];
  y2 := vec_dist[0];
  y1 := vec_dist[1];
  for cx:= dato_inicial downto x2+1 do
    begin
      x3 := cx;
      cta_arr[puntos]:=cx;
      a:=(x1-x2)/(x1-x3);
      y3:=y1-((y1-y2)/a);
      x_vec[puntos]:=y3;
      puntos:=puntos+1;
    end;
  cta_arr[puntos]:=x2;
  x_vec[puntos]:=y2;

  ( interpolacion )
  puntos:=puntos+1;
  j :=0;
  x3 := vec_cuenta[j];
  x1 := vec_cuenta[j+1];
  y3 := vec_dist[j];
  y1 := vec_dist[j+1];
  for cx:= vec_cuenta[0] downto vec_cuenta[i_fin]+1 do
    begin
      x2 := cx-1;
      cta_arr[puntos]:=cx-1;
      if x2 = x1 then
        begin
          x_vec[puntos]:=y1;
          j := j+1;
          x3 := vec_cuenta[j];
          x1 := vec_cuenta[j+1];
          y3 := vec_dist[j];
          y1 := vec_dist[j+1];
        end
      else
        begin
          a:= (x1-x2)/(x1-x3);
          y2:=y1-(a*(y1-y3));
          x_vec[puntos]:=y2;
        end;
      (end if)
      puntos:=puntos+1;
    end; ( for )

  (extrapolacion )
  x3 := vec_cuenta[i_fin-1];
  x2 := vec_cuenta[i_fin];
  y3 := vec_dist[i_fin-1];
  y2 := vec_dist[i_fin];
  x1 := vec_cuenta[i_fin]-1;
  repeat

```

```

a:= (x1-x2)/(x1-x3);
y1:=((y3*a)-y2)/(a-1);
x_vec[puntos]:=y1;
cfa_arr[puntos]:=x1;
x1 := x1-1;
puntos:=puntos+1;
until ( (y1>=100.00) or (x1<10000) );
end; ( procedure realiza_calculos )

begin ( PROGRAMA PRINCIPAL )

writeln;
writeln('          Generacion de Tabla:  Distancia = F( Cuenta ) .');
writeln;
write(' Archivo con lecturas del sonar >> ');
readln(nom_arch);
writeln;
write(' Archivo de salida >> ');
readln(nom_arch2);
assign(arch1,nom_arch);
($I-) reset(arch1) ($I+);
ok:= (ioresult=0);
if not ok then writeln('No se encontro el archivo ',nom_arch)
else
begin
  lee_archivo;
  writeln;
  writeln(' Realizando calculos !!!!!');
  writeln;
  writeln(' Espere por favor ...');
  realiza_calculos;
  escribe_en_archivo;
end;
end.

```

APENDICE C

PROGRAMA EN LENGUAJE TURBOPASCAL PARA

GENERAR TABLA DE VOLUMENES

(* Programa para generar una relación entre valores de rebotes de señales sonoras y volúmenes.
 Los resultados se envían a un archivo con formato para ser incluidos en programas en ensamblador de Z80.

*)

program sonar_distancias;

CONST

maxima_distancia= 100; { Puede enviar a una distancia de 1 metro }
 m=-0.1500003;
 b=9817.98;
 radio=14.75; { cm } (*dimensiones del barrilito experimental*)
 longitud=42; { cm }

TYPE

ST2 = STRING[2];
 ST3 = STRING[3];

var

cm : real;
 arch1 : Text; { Archivo con formato en ensamblador }
 i : integer;
 cta : longint;
 x,y,z : real;
 h,a,v : real;
 r_cuad: real;
 j,k,l : real;
 arcsin: real;
 ang : real;
 area_incognita: real;

FUNCTION SEG (X:REAL;POS:INTEGER):ST3;

{
 Extrae 2 dígitos de la variable real para convertirla en formato BCD }
 var
 hexa : string[9];
 h : string[2];

begin

STR(X:9:2,HEXA);
 H:=COPY(HEXA,POS,2);
 SEG:=H;
 if h=' 0' then seg:='00';
 if h=' 1' then seg:='01';
 if h=' 2' then seg:='02';
 if h=' 3' then seg:='03';
 if h=' 4' then seg:='04';
 if h=' 5' then seg:='05';
 if h=' 6' then seg:='06';
 if h=' 7' then seg:='07';
 if h=' 8' then seg:='08';
 if h=' 9' then seg:='09';
 if h=' ' then seg:='00';

end;

PROCEDURE CALCULO_AREA_BAJA;

{Procedimiento para calculo de area}

Begin

j:=pi*r_cuad/2;
 k:=h*(sqrt(r_cuad-(h*h)));
 ang:=h/radio;
 arcsin:=arctan(ang/(sqrt(-ang*ang+1)));
 l:=r_cuad*arcsin;
 a:=j-k-l;

end;

PROCEDURE ABRE_ARCHIVO;

{

}

```

{ Procedimiento para abrir archivos de Tablas de Conversion }
BEGIN
  assign(arch1,'jjv.asm');
  rewrite(arch1);
  writeln(arch1,' :40,; Tabla de valores para volúmenes');
  writeln(arch1,'TAB_VOL');
END; (ABRE_ARCHIVOS)

PROCEDURE ESCRIBE_EN_ARCHIVO;
{
  { Procedimiento para escribir datos de conversión en archivos }
  {
  BEGIN
    writeln(ARCH1,' :16,DEFB ',SEG(v,8),'H', ' :13,
      :; CTA=',CTA:5,' VOL=',V:9:2,'');
    writeln(ARCH1,' :16,DEFB ',SEG(v,5),'H', ' :13,;');
    writeln(ARCH1,' :16,DEFB ',SEG(v,3),'H', ' :13,;');
    writeln(ARCH1,' :16,DEFB ',SEG(v,1),'H', ' :13,;');
  END;

begin (* programa principal*)
  writeln(' Tabla de datos para volúmenes');
  writeln;
  r_cuad:=radio*radio;
  aRe_archivo;
  v:=pi*r_cuad*longitud;
  cta:=65448;
  cm:=0;
  writeln('CTA=',CTA:5,' VOL=',v:5:2,' cm= ',cm:5:2);
  escribe_en_archivo;
  cta:=cta+1;
  y:=0;
  while (y<(2*radio-0.15)) do
  begin
    y:=(m*CTA+b);
    cm:=y;
    if (y>radio) and (y<2*radio) then
    begin
      h:=y-radio;
      calculo_area_baja;
      area_incognita:=a;
      v:=area_incognita*longitud;
      writeln(' Y mayor CTA=',CTA:5,' VOL=',v:5:2,' cm= ',cm:5:2);
      escribe_en_archivo;
    end
  else
    if y=radio then
    begin
      area_incognita:=(pi*(sqr(radio)))/2;
      v:=area_incognita*longitud;
      writeln(' Y igual CTA=',CTA:5,' VOL=',v:5:2,' cm= ',cm:5:2);
      escribe_en_archivo;
    end
  else
    begin
      h:=radio-y;
      calculo_area_baja;
      area_incognita:= pi*r_cuad - a ;
      v:=area_incognita*longitud;
      writeln(' Y menor CTA=',CTA:5,' VOL=',v:5:2,' cm= ',cm:5:2);
      escribe_en_archivo;
    end;
  cta:= cta-1;
  end;
  v:=0;

```

```
cm:=radio*2;
writeln(' Y menor CTA=',CTA:5,' VOL=',v:5:2,' cm= ',cm:5:2);
escribe_en_archivo;
WRITELN(ARCH1,^z);
close(arch1);
write(' Oprima RETURN para continuar. ');
readln;
end.
```

APENDICE D

MANUAL DE USO DEL MUV

Manual de uso del "MUV"

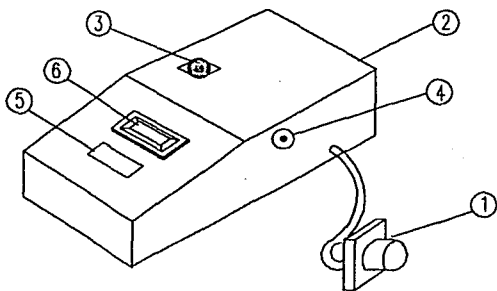
Introducción

El medidor ultrasónico de volúmenes "MUV" es un circuito electrónico, el cual permite obtener la medición de volúmenes de recipientes cuya sección sea conocida, con gran versatilidad.

Es un dispositivo portátil y compacto que puede ser utilizado en una gran variedad de aplicaciones. Presenta además la característica de realizar mediciones en cilindros horizontales comúnmente utilizados.

Presentación

El ensamble que presenta es el siguiente:



1) Transductor

El transductor es el dispositivo que se debe utilizar para realizar las mediciones. Tiene que ser dirigido al objeto a medir, tomando en cuenta que la medición obtenida, es de la base del transductor al objeto.

2) El chasis

3) Botón de encendido o apagado del sistema

4) Botón reset

5) Teclado

Está provisto de cuatro botones con dos funciones diferentes cada uno. De izquierda a derecha como sigue:

Superior. Rep = Repetir
Liq = Recipiente con contenido
Vac = Recipiente vacío
Ini = Seguir a otra medición desde el inicio

Inferior. OK = Enter o correcto
No = Respuesta no
Si = Respuesta si
Num = Cambio de número

6) Display de LCD

Pantalla en la que aparecerán los datos de entrada y la información final.

USO

1) Enciendase el sistema con el botón correspondiente. Si no aparece nada en el display, oprima una vez el botón de reset. Una vez encendido aparecerá el mensaje "sonar medidor" durante tres segundos, posteriormente desplegará una pregunta que usted debe contestar a través del teclado.

2) Recipiente horizontal SI. Si usted ha elegido la ruta si, esto significa que el recipiente es de tipo tanque (pipa). No. si usted oprime esta tecla pase al punto número cinco.

3) En el display aparecerá la confirmación de esto mediante un mensaje. Ahora usted debe establecer el transductor en el lugar de la toma, una vez hecho lo anterior, debe oprimir la tecla "OK" con lo que el dispositivo realizará las mediciones pertinentes.

En este momento, el volumen del líquido del recipiente, el cuál usted desea medir, es desplegado. Si usted desea seguir con el uso del aparato, teclee "OK".

4) Existen aquí dos posibles alternativas. Si usted desea confirmar su lectura oprima. "Rep" y regrese al punto tres. De no ser así usted puede dar "ini" y con esto regresará de nuevo al inicio del sistema como si apenas hubiera encendido, punto dos.

5) Si después de la pregunta recipiente horizontal? Usted eligió la opción "no". Significa que es un cilindro en forma vertical lo cuál se corrobora con el mensaje correspondiente. Oprima OK, ahora se despliega el mensaje "vacío o con líquido?" a lo cual debe

responder con las teclas "vac" o "liq".

6) **VAC.** Significa entonces que el recipiente está vacío, aparecerá la señal de confirmación pase al punto ocho.

LIQ. Si oprime "liq", significa que el recipiente posee algún contenido dentro del mismo. Por lo que usted debe después del mensaje de confirmación introducir el valor de la máxima altura del cilindro, dígito por dígito cambiando con la tecla "num" al número deseado y confirmando este con la tecla de OK. Una vez introducida la altura máxima se desplegará el mensaje correspondiente.

7) Aparecerá ahora el mensaje de petición de diámetro a lo que debe contestar con la tecla numérica, indicando el valor de este, tal y como se hizo con la altura. Después aparecerá el mensaje de confirmación. Ponga el transductor en la boca del recipiente, oprima nuevamente "OK" para que el "MOV" realice las mediciones y cálculos respectivos. En el display se desplegará el volumen.

Usted se encuentra en la posición de solicitar, una nueva medida de un recipiente con liquido tecleando "rep", o iniciar el sistema y regresar al punto dos con "ini".

8) Si eligió la opción del recipiente vacío debe entrar el valor del diámetro, dígito a dígito con la tecla "num" y confirmando cada uno con "ok". Terminado el valor del diámetro oprimir "ok", esperar el mensaje de confirmación. Establecer el transductor en la parte superior del recipiente y entrar "ok" cuando este listo. Aparece el valor del volumen del recipiente completo.

Si desea solicitar una nueva medición del recipiente vacío dar "rep", o si desea otro tipo de medición oprimir "ini", regresando al punto dos .

NOTA: Si quiere realizar alguna medición de distancia debe ingresar a la ruta de recipiente vacío y cuando se pida el valor de diámetro favor de teclear 1.13, con lo que el calor final que despliega el dispositivo corresponde a la distancia entre el transductor y el blanco deseado.

BIBLIOGRAFÍA

- Evolución de los Conceptos de la Física
Arons Arnold. B.
Trillas

- Física parte I
Resnick Robert, Halliday David
CECSA

- Introducción a la Mecánica de Fluidos
Fox R.W - Mc Donald A.T.
Interamericana

- Biblioteca de Electrónica-Infomática
Tomo 53 Radar - Sonar
Mompin Poblet. José
Orbis s.a (Marcombo)

- Dispositivos Electrónicos y Amplificación de señales
Sedra. A. -Smith. K.C
Mc Graw Hill

- Circuitos Electrónicos
Edminister. Joseph A.
Mc Graw Hill. Schaum

- Ingeniería de Control Moderna
Ogata. Katsuhiku
Prentice Hall

- Lineal Circuits Amplifiers, Comparators and Special functions
Data book
Volumen 1
Texas instruments

- Revista Nueva electrónica
número 87 febrero 1991
Comercial electrónica RTE

- Revista Sensors
The Journal of Machine Perception
Noviembre 1989 Vol 6 No. 11
Helmert Publishing Inc.

- The Z80 Microprocessor
Hardware, software Programming and Interfacing
Brey. Barry B.
Prentice Hall 1988

- Build your own Z80 computer
Ciarcia. Steve
Mc Graw Hill USA 1984

- Z80 Family Data Book
Zilog January 1989 USA

- Cálculo Diferencial e Integral
Ayres. Frank Jr
Mc Graw Hill Schaum 1982