

Escola Politécnica da
Universidade de São Paulo
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

7,5 (site e anexo)
wsm

PMC 581 – Projeto mecânico II

SISTEMA ACÚSTICO DE LOCALIZAÇÃO SUBMARINA

Alunos: Alex de Oliveira
Frederico Vines Faria de Lima

Orientador: Prof. Dr. Celso Massatoshi Furukawa

Data de entrega: 08/2/2001

1	Motivação	5
2	Introdução	6
3	Descrição do sistema proposto	8
4	Estudo De Acústica	11
4.1	Equação da onda acústica e soluções simples	11
4.1.1	Hipóteses:	11
4.1.2	Equação de estado	11
4.1.3	Equação da continuidade	12
4.1.4	A equação simples da força: Equação de Euler	13
4.1.5	Equação de onda linearizada	14
4.1.6	Densidade de energia, Intensidade acústica e Impedância acústica específica	14
4.1.7	Ondas Harmônicas Planas	16
4.1.8	Ondas esféricas	18
4.2	Radiação de Ondas Acústicas	21
4.2.1	Radiação de uma esfera pulsante	21
4.2.2	Força da fonte	22
4.2.3	Fontes simples	22
4.2.4	Radiação de uma fonte tipo linha	24
4.2.5	Radiação de um pistão plano	25
4.2.6	Impedância de Radiação	26
4.3	Absorção e Atenuação de Ondas Sonoras	28
4.3.1	Aproximação para a absorção	28
4.3.2	O fenômeno da absorção em água do mar	30
4.4	Comentários sobre o estudo de acústica	32
5	Estudo De Transdutores Piezelétricos	33
5.1	Transdutores piezelétricos	33
5.1.1	Modelo matemático simplificado	33
5.1.2	Transdutor de três camadas	39
5.1.3	Modelo do receptor	44
6	Estudo Da Viabilidade Do Projeto Com Transdutores Comerciais	47
7	Modelagem Matemática Desenvolvida	49
8	Estudo do PIC e Comunicação Serial	51
8.1	Estudo do PIC	51
8.1.1	Arquitetura	51
8.1.2	Periféricos	52
8.1.3	Programação	53
8.2	Estudo de comunicação serial – PC	54

9	Requisitos e decisões iniciais do projeto	55
10	Materiais e softwares utilizados na implementação do projeto	58
11	Construção e Testes de Transdutores na Água	60
11.1	A primeira tentativa de construção de um transdutor	60
11.2	Etapas de construção e decisões de projeto do protótipo final do transdutor	63
11.3	Curvas de Impedância Elétrica do Transdutor	66
11.4	Circuito Equivalente do Transdutor na água	67
11.5	Circuito de Teste do Transdutor na Água	68
12	Implementação do Software e da comunicação com o PC	69
12.1	Softwares.....	69
12.1.1	Programas dos PIC's	69
12.1.2	Programa do PC	71
12.1.3	Lista de Comandos	71
12.1.4	Regras de comunicação serial	72
12.2	Circuito de testes	73
13	Testes e Discussões dos softwares das Unidades de Comando, Móvel e Fixa	74
14	Estudo dos Circuitos Eletrônicos de Emissão	76
14.1	Breve Explicação do Circuito de Emissão.....	76
14.2	O Comparador LM311	76
14.3	O Power-Mosfet IRF840.....	77
14.4	Circuito para estabilizar a fonte	77
14.5	O Transformador.....	78
14.6	Teste do Circuito de Emissão	78
15	Considerações Finais	80
16	Bibliografia	81
17	Anexos	83
17.1	Convenção de sinais para o modelamento dos transdutores	83
17.2	Programas em MATLAB	85

17.3	Catálogo de transdutores esféricos	93
17.4	Exemplo de Circuito estudado	95
17.5	Código fonte da Unidade de comando.....	97
17.6	Código fonte da Unidade móvel	106
17.7	Código fonte da Unidade fixa.....	119
17.8	Esquema do circuito de teste	122

1 Motivação

A Petrobrás é uma das empresas que possuem melhor tecnologia em prospecção de petróleo, especialmente extração de petróleo de jazidas localizadas sob o fundo do mar. Para se extrair petróleo sob o fundo do mar, deve-se instalar uma série de equipamentos a uma profundidade muito grande. À profundidades superiores a 1000 metros é impossível que os trabalhos sejam feitos por seres humanos. Vários sistemas especiais de tubulações, acoplamento e outros são necessários. Um certo acoplamento de um equipamento é feito da seguinte forma: um equipamento está preso ao fundo do mar e outro equipamento deve ser acoplado a ele. Esse equipamento é deslocado para um local próximo do equipamento-alvo através de um cabo. No entanto, o posicionamento exato do equipamento é feito a partir do navio mediante o movimento das extremidades do cabo acoplado a um guindaste. O monitoramento é feito através de uma câmera,. Esse processo é muito difícil e demorado devido a interferências externas e a falta de precisão no posicionamento. Um projeto da Petrobrás em conjunto com a EPUSP pretende projetar um sistema de posicionamento que consiste de um veículo automático e independente que posicione o equipamento sobre o equipamento-alvo.

2 Introdução

O presente trabalho tem como objetivo projetar, construir e testar um sistema submarino de localização acústico capaz de localizar um veículo em relação a um alvo. O princípio de funcionamento é a cronometragem do tempo decorrido entre a emissão de um sinal acústico e a recepção de uma resposta por parte do veículo. Serão utilizados transdutores acústicos de ultra-som para a geração dos sinais acústicos.

A tarefa mais básica do sistema de localização é medir a distância entre o veículo e o alvo. Se a distância até o alvo puder ser medida em pontos diferentes do veículo com boa precisão, a posição e a orientação do veículo em relação ao alvo pode ser calculada. Sendo assim, a princípio decidiu-se que o protótipo final deste trabalho será capaz apenas de medir a distância entre o equipamento e o alvo, já que os princípios de projeto e implementação seriam os mesmos que os do sistema completo, havendo diferença apenas na quantidade de componentes e cálculos.

O sistema de localização será composto de duas unidades: a unidade móvel envia sinais acústicos, cronometra o tempo, identifica o sinal de resposta, calcula a distância e torna disponíveis os dados da distância para uma provável unidade de comando; a unidade fixa identifica o sinal enviado pelo veículo e procura responder imediatamente, enviando outro sinal.

As unidades móvel e fixa serão controladas por microcontroladores, que coordenarão o acionamento dos transdutores acústicos, a cronometragem do tempo, a recepção da resposta, o cálculo da distância e a disponibilização dos resultados. A unidade de comando, que no presente trabalho será simulada por um PC, fornecerá parâmetros de calibração, ordenará a localização e receberá o valor da distância.

No primeiro semestre foram feitos os estudos de acústica, dos transdutores piezoelétricos, além do estudo de viabilidade do uso de transdutores comerciais e a modelagem de um sistema de emissão-transmissão-recepção de ondas acústicas em ambiente marinho.

No segundo semestre parte do projeto consistiu do desenvolvimento e implementação de toda a parte de software, o que incluiu os programas dos microcontroladores e da unidade de comando.

Para isto, inicialmente foi feita a escolha do microcontrolador e o estudo de suas capacidades. Em seguida, foram desenvolvidos os programas dos microcontroladores. Seguiu-se o estudo de comunicação serial (método de comunicação escolhido entre a unidade de comando e a unidade móvel) e o desenvolvimento do software da unidade de comando. Foi construído então um circuito de testes para que todas as funções do sistema pudessem ser testadas na prática.

A seguir foram escolhidos os componentes e estes foram montados para a construção dos circuitos eletrônicos de emissão e recepção a serem acoplados nas unidades móveis e fixas para que sejam calculadas as distâncias necessárias que este sensor de posição fornecerá para a unidade de comando do sistema.

Também foi projetado e construído um transdutor piezoelétrico para a emissão e recepção dos sinais. Foram levantadas as curvas de impedância do transdutor na água e este foi inserido na parte eletrônica do circuito de emissão para efeito de teste.

3 Descrição do sistema proposto

O sistema de localização proposto deve funcionar como um sensor de posição para um veículo submarino que tem o objetivo de se movimentar até um alvo predeterminado. O veículo possui um comando central, que será chamado Unidade de Comando, que é responsável pelo controle de movimentação do veículo, recebendo dados, tomando decisões segundo uma estratégia de controle e acionando atuadores (motores).

A função do sistema de localização é fornecer a distância e a orientação do veículo (em relação ao alvo) à unidade de comando. O método escolhido para a determinação dessas informações é a medição da distância até o alvo de pelo menos três pontos diferentes do veículo utilizando ondas acústicas. Ondas acústicas se propagam em altas velocidades (da ordem de 1500 m/s) e possuem baixa atenuação na água. Se o intervalo de tempo entre a emissão de um sinal acústico por parte do veículo e a recepção por parte do mesmo de um sinal de resposta enviado pelo alvo for medido, a distância pode ser calculada, desde que seja conhecida a velocidade de propagação do som no meio.

No presente trabalho, será implementada a tarefa mais básica do sistema de localização, que consiste em se medir simplesmente a distância do veículo até o alvo. Para um sistema completo, as metodologias de projeto e implementação seriam as mesmas, havendo diferença apenas no número de componentes (para medir a distância de três pontos) e na quantidade de cálculos a partir dos dados de distância obtidos.

A figura a seguir ilustra o que foi dito e mostra as duas unidades básicas do sistema de localização:

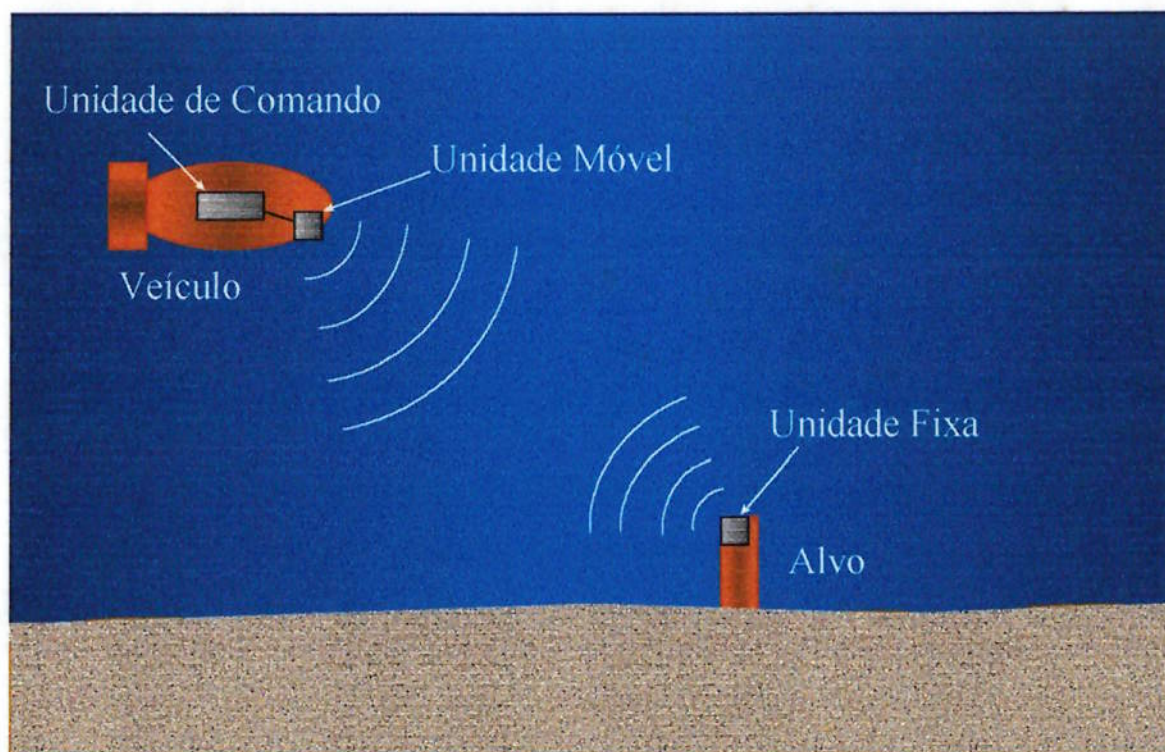


Figura 1 - Ilustração do sistema de localização e das unidades constituintes

O sistema proposto é formado por duas unidades: unidade móvel e unidade fixa. A unidade móvel possui comunicação com a unidade de comando, que não faz parte do sistema de localização mas envia comandos à unidade móvel e recebe respostas da mesma.

A unidade móvel tem a função de emitir sinais acústicos (que serão chamados sinais de busca), cronometrar o tempo, identificar uma resposta válida da unidade fixa, calcular a distância até a unidade fixa com base no tempo cronometrado entre a emissão do sinal e a recepção da resposta e disponibilizar essa resposta à unidade de comando. É constituída basicamente de microcontrolador, transdutor acústico de ultra-som, fonte de alta tensão, driver (para acionamento do transdutor) e circuito de recepção (composto por amplificador, filtro analógico e detetor de limiar - veja seção sobre as decisões de projeto).

A unidade fixa tem a função de identificar um sinal acústico vindo da unidade móvel e enviar outro sinal acústico (que será chamado sinal de resposta). Os componentes dessa unidade são basicamente os mesmos da unidade móvel: microcontrolador, transdutor acústico de ultra-som, fonte de alta tensão, driver e circuito de recepção.

Apresenta-se abaixo um esquema dos componentes de cada unidade, seguido da explicação do funcionamento do sistema (sobre os componentes de cada unidade, veja a seção sobre as decisões de projeto iniciais):

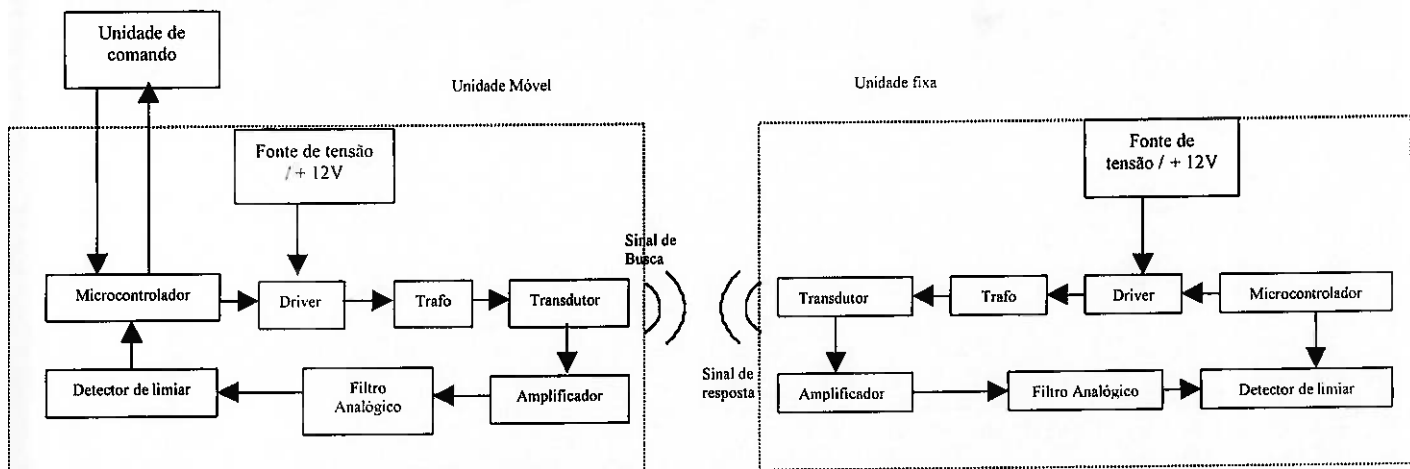


Figura 2 - Componentes das unidades móvel e fixa

- A unidade de comando envia um comando para a unidade móvel realizar a localização;
- O microcontrolador reconhece o comando e gera um trem de pulsos que comanda o acionamento de um transdutor acústico, que por sua vez emite um sinal acústico omnidirecional (chamado sinal de busca). Ao mesmo tempo, um cronômetro digital é acionado pelo microcontrolador;
- O sinal acústico chega até a unidade móvel e é por ela reconhecido através de amplificação, filtragem e detecção;
- A unidade móvel envia outro sinal acústico omnidirecional (chamado sinal de resposta);
- O sinal de resposta chega até a unidade móvel e é por ela reconhecido;
- O cronômetro é parado e a distância é calculada a partir da medida do tempo feita pelo cronômetro e do valor da velocidade de propagação do som previamente calibrado e armazenado na unidade móvel;
- A medida de distância é disponibilizada pela unidade móvel à unidade de comando.

4 Estudo De Acústica

Visando a compreensão e o modelamento dos fenômenos acústicos que ocorrem na geração, propagação e recepção de ondas acústicas, foi feito um estudo de acústica com base na referência bibliográfica [4].

4.1 Equação da onda acústica e soluções simples

4.1.1 Hipóteses:

- A densidade de equilíbrio do fluido e a pressão de equilíbrio têm valores uniformes em todo o fluido (efeitos de gravidade serão desprezados).
- Fluido homogêneo, isotrópico e perfeitamente elástico (não há dissipação).
- Mudanças na densidade do meio são pequenas se comparadas à densidade de equilíbrio.

4.1.2 Equação de estado

A equação isotérmica de estado para um gás perfeito é:

$$\frac{P}{P_0} = \left(\frac{\rho}{\rho_0} \right) \quad (\text{gás perfeito})$$

Experimentalmente, sabe-se que processos acústicos são aproximadamente adiabáticos (as trocas de calor entre as partículas do fluido são insignificantes). A equação adiabática de estado é:

$$\frac{P}{P_0} = \left(\frac{\rho}{\rho_0} \right)^\gamma \quad (\text{gás perfeito})$$

Para fluidos que não são gases perfeitos, os processos acústicos ainda podem ser considerados adiabáticos, mas a equação de estado deve ser determinada experimentalmente. Uma linearização em torno do ponto de equilíbrio (P_0, ρ_0) é:

$$P = P_0 + \left(\frac{\partial P}{\partial \rho} \right)_{\rho_0} (\rho - \rho_0) \Rightarrow P - P_0 = \left(\frac{\partial P}{\partial \rho} \right)_{\rho_0} \left(\frac{\rho - \rho_0}{\rho_0} \right)$$

O valor $\left(\frac{\partial P}{\partial \rho} \right)_{\rho_0}$ é chamado **módulo adiabático de volume**, e pode

ser determinado experimentalmente e tabelado. O valor $\left(\frac{\rho - \rho_0}{\rho_0} \right)$ é chamado **de condensação**. O excesso de pressão $P - P_0$ é chamado de **pressão acústica**. A equação adiabática de estado fica:

$$p = \beta s \quad (\text{válida para } |s| \ll 1)$$

Onde:

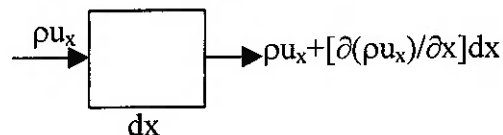
p: pressão acústica;

β : módulo acústico de volume;

s: condensação.

4.1.3 Equação da continuidade

A equação da continuidade proporciona uma relação entre a velocidade da partícula fluida \vec{u} e a densidade instantânea ρ .



O paralelepípedo acima representa um porção fixa do espaço (elemento de volume) e o fluido flui através dele. O fluxo de massa que entra menos o fluxo de massa que sai é igual a taxa de variação de massa dentro do elemento de volume na direção x. O fluxo líquido de massa é:

$$\rho u_x dydz - \left[\rho u_x + \frac{\partial(\rho u_x)}{\partial x} dx \right] dydz = -\frac{\partial(\rho u_x)}{\partial x} dx dydz$$

Analogamente, pode-se montar a expressão acima para as direções y e z. O fluxo mássico total no elemento (diferença entre a vazão mássica de entrada e a de saída) é:

$$-\left[\frac{\partial(\rho u_x)}{\partial x} + \frac{\partial(\rho u_y)}{\partial y} + \frac{\partial(\rho u_z)}{\partial z}\right]dV = -[\nabla \cdot (\rho \vec{u})]dV$$

A taxa de aumento de massa em relação ao tempo é $\frac{\partial \rho}{\partial t} dV$, que é igual ao fluxo líquido:

$$\frac{\partial \rho}{\partial t} dV = -[\nabla \cdot (\rho \vec{u})]dV \Rightarrow \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$$

Como $s = \frac{\rho - \rho_0}{\rho_0} \Rightarrow \frac{\partial s}{\partial t} = \frac{1}{\rho_0} \frac{\partial \rho}{\partial t}$ e $\rho \approx \rho_0$. Então:

$$\frac{\partial s}{\partial t} \rho_0 + \rho_0 \nabla \cdot \vec{u} = 0 \Rightarrow \frac{\partial s}{\partial t} + \nabla \cdot \vec{u} = 0 \quad (\text{Equação da continuidade linearizada})$$

4.1.4 A equação simples da força: Equação de Euler

Em fluidos reais, a existência de viscosidade e o fato dos processos acústicos não serem perfeitamente adiabáticos introduzem termos dissipativos. Na análise seguinte, considera-se o fluido invíscido (2ª. hipótese).

Num elemento fluido em movimento (paralelepípedo com lados dx , dy e dz), a força resultante causará uma aceleração segundo a lei de Newton: $d\vec{f} = dm\vec{a}$. Essa força é causada por uma diferença de pressão nas faces do elemento. A resultante na direção de x é:

$$P dydz - \left[P + \frac{\partial P}{\partial x} dx \right] dydz = -\frac{\partial P}{\partial x} dV = df_x$$

Para as direções y e z a equação é análoga. A força vetorial é:

$$d\vec{f} = df_x \vec{i} + df_y \vec{j} + df_z \vec{k} = -\nabla P dV$$

A velocidade é função do espaço e do tempo. Verifica-se (usando derivadas parciais) que a aceleração é dada por:

$$\vec{a} = \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u}$$

A massa dm é igual a ρdV , portanto a 2ª lei de Newton fica:

$$-\nabla P dV = \rho dV \left[\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \right] \Rightarrow -\nabla p = \rho \left[\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \right]$$

Foi usado acima que $\nabla P = \nabla p$.

Se $s \ll 1$ e $\left| \left(\vec{u} \cdot \nabla \right) \vec{u} \right| \ll \frac{\partial \vec{u}}{\partial t}$ (processo acústico de pequena amplitude),

temos:

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\nabla p \quad (\text{Equação de Euler})$$

4.1.5 Equação de onda linearizada

Tomando o divergente da equação anterior, temos:

$$\rho_0 \nabla \cdot \frac{\partial \vec{u}}{\partial t} = -\nabla \cdot \nabla p = -\nabla^2 p$$

Derivando a equação da continuidade linearizada, temos:

$$\frac{\partial^2 s}{\partial t^2} + \frac{\partial \left(\nabla \cdot \vec{u} \right)}{\partial t} = 0$$

Substituindo:

$$\rho_0 \left(\frac{-\partial^2 s}{\partial t^2} \right) = -\nabla^2 p \rightarrow \nabla^2 p = \rho_0 \frac{\partial^2 s}{\partial t^2}$$

Usando a equação de estado $p = \beta s$, podemos eliminar s :

$$\nabla^2 p = \frac{\rho_0}{\beta} \frac{\partial^2 p}{\partial t^2}$$

Definindo $c = \sqrt{\frac{\beta}{\rho_0}}$, a equação anterior fica:

$\nabla^2 p = \frac{\partial^2 p}{c^2 \partial t^2}$, que é a **equação linearizada para a propagação do som**

em fluidos (sem perdas)

A equação de estado pode ser reescrita da seguinte forma:

$$P = \rho_0 c^2 s$$

4.1.6 Densidade de energia, Intensidade acústica e Impedância acústica específica

Para que se possa prosseguir no estudo das ondas, convém que sejam explicados algumas variáveis acústicas importantes na descrição de ondas e meios de propagação.

A energia cinética de um elemento de volume é $E_k = \frac{1}{2} \rho_0 V_0 u^2$, onde $\rho_0 V_0$ é a massa do elemento calculada no fluido sem perturbação.

A variação de energia potencial associada com a variação de volume de V_0 para V é $E_p = - \int_{V_0}^V p dV$, onde o sinal negativo indica que a energia potencial vai aumentar se o fluido tiver seu volume diminuído pela ação de uma pressão acústica positiva p .

Da conservação de massa: $\rho V = \rho_0 V_0$.

$$\text{Daí: } \rho dV + d\rho V = 0 \Rightarrow dV = -\frac{V}{\rho} d\rho = -\left(\frac{V_0 + dV}{\rho_0 + d\rho}\right) d\rho = -\frac{V_0}{\rho_0 + d\rho} d\rho$$

$$\text{Como } s = \frac{\rho - \rho_0}{\rho_0} \ll 1 \rightarrow d\rho \ll \rho_0 \rightarrow dV = -\frac{V_0}{\rho_0} d\rho$$

$$\text{Como } p = \rho_0 c^2 s, \text{ então: } dp = \rho_0 c^2 \frac{d\rho}{\rho_0} \text{ Substituindo } d\rho \text{ tem-se:}$$

$$dV = -\frac{V_0}{\rho_0 c^2} dp$$

Pode-se agora calcular a variação de energia potencial, integrando-se de 0 a p .

$$E_p = \int_0^p p \frac{V_0}{\rho_0 c^2} dp = \frac{V_0}{\rho_0 c^2} \frac{p^2}{2}$$

A energia acústica total do elemento de volume é:

$$E = E_k + E_p = \frac{1}{2} \rho_0 V_0 u^2 + \frac{1}{2} \frac{V_0}{\rho_0 c^2} p^2$$

A **energia instantânea por unidade de volume** (densidade de energia) é:

$$\varepsilon_i = \frac{1}{2} \rho_0 \left(u^2 + \frac{p^2}{\rho_0^2 c^2} \right)$$

A média no tempo da energia instantânea dá a densidade de energia:

$$\varepsilon = \frac{1}{t} \int_0^t \varepsilon_1 dt$$

A **intensidade acústica I** de uma onda sonora é definida como a taxa média de fluxo de energia através de uma unidade de área normal à direção de propagação.

A taxa instantânea do trabalho que um elemento de fluido realiza sobre um elemento adjacente, por unidade de área, é $p \cdot u$. A intensidade é a média no tempo desta taxa: $I = \langle pu \rangle_t = \frac{1}{t} \int_0^t p u dt$, onde a integração é feita no período.

A razão entre a pressão acústica em um meio e a velocidade da partícula associada é chamada de **impedância acústica específica**: $Z = \frac{p}{u}$

Em geral, Z possui a forma complexa $Z = r + jx$, onde r é chamada de resistência acústica específica e x de reatância acústica específica.

4.1.7 Ondas Harmônicas Planas

Se todas as variáveis acústicas são funções de somente uma coordenada espacial, a fase de qualquer variável é uma constante em qualquer plano perpendicular a esta coordenada. Tais ondas são chamadas de ondas planas.

Se o sistema de coordenadas é escolhido de tal forma que a onda plana se propaga ao longo da direção x a equação da onda se reduz a:

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \rightarrow \frac{\partial^2 p}{\partial x^2} = \frac{\partial^2 p}{c^2 \partial t^2}, \text{ onde } p = p(x, t)$$

Esta equação é uma equação diferencial de segunda ordem. Verifica-se que sua solução mais geral é $p = y(x, t) = y_1(ct - x) + y_2(ct + x)$, onde y_1 e y_2 são funções completamente arbitrárias com argumentos $(ct - x)$ e $(ct + x)$.

Se p é função da posição e do tempo, vamos considerar uma onda plana se propagando na direção x.

No instante de tempo t_1 a pressão é $p(x, t_1) = y_1(ct_1 - x) + y_2(ct_1 + x)$. Num instante posterior t_2 a pressão será dada por $p(x, t_2) = y_1(ct_2 - x) + y_2(ct_2 + x)$.

A pressão $p(x_1, t_1)$ da onda no ponto x_1 quando $t=t_1$ deve ser encontrada no ponto x_2 no instante t_2 (pois a onda se propaga). Se os argumentos da função forem iguais as pressões serão iguais, logo, igualando-se os argumentos para o ponto x_2 no instante t_2 e para o ponto x_1 no instante t_1 , tem-se:

$$\begin{aligned} ct_1 - x_1 &= ct_2 - x_2 & \longrightarrow & \quad x_2 - x_1 = c(t_2 - t_1) \\ ct_1 + x_1 &= ct_2 + x_2 & \longrightarrow & \quad x_1 - x_2 = c(t_2 - t_1) \end{aligned}$$

Nota-se que, no caso acima, os argumentos só serão iguais se $t_1=t_2$. Portanto, neste caso, não se pode garantir que as pressões são iguais em instantes de tempo diferentes em lugares diferentes (isto pode acontecer, dependendo apenas das funções y_1 e y_2).

Por outro lado, se a onda plana é dada por $p(x, t) = y_1(ct - x)$, podemos usar a mesma discussão acima e

$$ct_1 - x_1 = ct_2 - x_2 \text{ e } x_2 - x_1 = c(t_2 - t_1)$$

Ora, se a pressão no ponto x_2 no instante t_2 é igual à pressão no ponto x_1 no instante t_1 , então a onda se propagou!

A velocidade de fase ou a velocidade de propagação da onda acústica é:

$$\frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1} = c!$$

A onda anterior se propaga no sentido positivo do eixo x . A forma complexa de uma solução harmônica para a pressão acústica de uma onda plana é:

$$p = Ae^{jk(ct-x)} + Be^{jk(ct+x)}, \text{ onde } k \text{ é o número de onda e vale } \omega/c.$$

$$\text{Portanto : } p = Ae^{j(\omega t - kx)} + Be^{j(\omega t + kx)}$$

A partir desta expressão acústica pode-se obter a expressão da velocidade das partículas (u) e da condensação (s).

A equação de Euler simplificada é:

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\nabla p = -\frac{\partial p}{\partial x} \vec{i} \rightarrow \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho_0} [A(-KJ)e^{j(\omega t - kx)} + BkJe^{j(\omega t + kx)}] \vec{i}$$

Integrando-se em t tem-se:

$$\vec{u} = \left[\frac{A}{\rho_0 c} e^{j(\omega t - kx)} - \frac{B}{\rho_0 c} e^{j(\omega t + kx)} \right] \vec{i}$$

Se $B=0$ a onda se propaga no sentido de x positivo

$$p = Ae^{j(\omega t - kx)} \text{ e } \vec{u} = \frac{A}{\rho_0 c} e^{j(\omega t - kx)} \rightarrow u = \frac{p}{\rho_0 c}$$

A energia instantânea por unidade de volume é:

$$\varepsilon_i = \frac{1}{2} \left(\frac{p^2}{\rho_0^2 c^2} + \frac{p^2}{\rho_0^2 c^2} \right) \rho_0 \rightarrow \varepsilon_i = \frac{p^2}{\rho_0 c^2} \rightarrow \varepsilon = \frac{1}{T} \int_0^T \varepsilon_i dt, \text{ onde } p \text{ é a parte real}$$

da pressão.

$$\text{Portanto } \rho_0 c^2 \varepsilon = \frac{A^2}{T} \int_0^T \cos^2(\omega t - kx) dt \rightarrow \varepsilon = \frac{A^2}{2\rho_0 c^2} = \frac{1}{2c} P_r U, \text{ onde } P_r = A \text{ e}$$

$U = A/\rho_0 c$ (amplitudes). Ou então $\varepsilon = \rho_0 U^2/2$.

Esse valor de densidade de energia é válido para ondas planas e para ondas progressivas quando as superfícies de fase constante se tornam praticamente planas (o raio de curvatura das superfícies é muito maior que o comprimento de onda).

A intensidade acústica é:

$$I = \frac{1}{T} \int_0^T p u dt \rightarrow I = \frac{A^2}{T\rho_0 c} \int_0^T \cos^2(\omega t - kx) dt = \frac{1}{2} \frac{P^2}{\rho_0 c} = \frac{1}{2} P U$$

Para ondas planas a impedância acústica Z é $Z = p/u = \rho_0 c$. Esse valor é chamado de impedância característica do meio.

4.1.8 Ondas esféricas

Ondas esféricas são ondas que têm simetria esférica, isto é, a pressão acústica é função de uma distância radial.

O operador Laplaciano (∇^2), expresso em coordenadas esféricas é:

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{2\partial}{r\partial r} + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \sin \theta \frac{\partial}{\partial \theta} + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2}$$

Se as derivadas parciais em relação a θ e ϕ são nulas (simetria esférica):

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{2\partial}{r\partial r}$$

Uma onda esférica longe da sua fonte pode ser aproximada para uma onda plana, na qual a relação $p = \rho_0 c u$ é válida. A intensidade acústica é dada

por $I = \frac{P^2}{2\rho_0 c}$ (ver item sobre ondas planas). A potência em toda a área irradiada

é $\Pi = 4\pi r^2 I$. Pela conservação de energia, pode-se esperar que a potência irradiada seja constante e, conseqüentemente, que o produto Pr seja constante, ou seja, que a amplitude da pressão acústica é inversamente proporcional ao raio. A função pr é uma função de amplitude constante em todo o espaço, assim como nas ondas planas p possui amplitude constante em todo o espaço. Pode-se tentar escrever a equação da onda esférica tratando pr como uma única variável:

$$\frac{\partial^2 p}{\partial r^2} + \frac{2}{r} \frac{\partial p}{\partial r} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \Leftrightarrow r \frac{\partial^2 p}{\partial r^2} + 2 \frac{\partial p}{\partial r} = \frac{r}{c^2} \frac{\partial^2 p}{\partial t^2} \Leftrightarrow \frac{\partial^2 (rp)}{\partial r^2} = \frac{1}{c^2} \frac{\partial^2 (rp)}{\partial t^2}$$

A equação acima tem a mesma forma que a equação de ondas planas.

Portanto, a solução geral é:

$$pr = f_1(ct-r) + f_2(ct+r) \text{ ou } p = \frac{1}{r} f_1(ct-r) + \frac{1}{r} f_2(ct+r)$$

O primeiro termo representa uma onda esférica divergindo de uma fonte na origem e o segundo termo uma onda convergindo na origem.

As ondas esféricas mais importantes são as ondas harmônicas divergentes. Tais ondas são representadas na forma complexa por:

$$p = \frac{A}{r} e^{j(\omega t - kr)}$$

A partir da equação de Euler, pode-se encontrar a expressão para a velocidade das partículas.

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\nabla p$$

$$\nabla p = \frac{\partial p}{\partial r} \hat{r} = \frac{A e^{j(\omega t - kr)} (-jk)r - A e^{j(\omega t - kr)}}{r^2} \text{ . Daí:}$$

$$u = \frac{-1}{\rho_0} \int \frac{A e^{j(\omega t - kr)} (jkr + 1)}{r^2} dt = \frac{(jkr + 1) A e^{j(\omega t - kr)}}{\rho_0 r^2 j\omega}$$

Como $\omega = kc$ tem-se que:

$$\vec{u} = \frac{(kr - j)}{kr} \frac{p}{\rho_0 c} \hat{r}$$

Desta equação pode-se observar que a velocidade da partícula não está em fase com a pressão acústica (a pressão é multiplicada por um número com parte imaginária não nula).

A impedância acústica específica é:

$$Z = \frac{p}{u} = \frac{\rho_0 c k r}{k r - j} \frac{k r + j}{k r + j} = \frac{\rho_0 c (k r)^2}{(k r)^2 + 1} + j \frac{\rho_0 c k r}{1 + (k r)^2} = \rho_0 c \cos \theta e^{j\theta}, \text{ sendo } \cot \theta = k r$$

(θ : diferença de fase entre u e p)

O termo real da impedância é a resistência acústica específica e o termo imaginário é a reatância acústica específica.

$k r = \frac{2\pi r}{\lambda}$ é função da razão entre a distância à fonte e o comprimento de onda. Quando a distância da fonte é pequena (uma fração do comprimento de onda), a diferença de fase entre a pressão complexa e a velocidade da partícula é grande. À distância de vários comprimentos de onda, por outro lado, p e u estão aproximadamente em fase e a onda esférica assume as características de uma onda plana, comportamento que já era esperado.

A intensidade acústica de uma onda esférica é dada por:

$$I = \frac{1}{T} \int_0^T P \cos(\omega t - k r) U \cos(\omega t - k r - \theta) dt = \frac{P U \cos \theta}{2} = \frac{P^2}{2 \rho_0 c}, \text{ sendo:}$$

$$P = \frac{A}{r} \text{ e } U = \frac{A}{r \rho_0 c \cos \theta}$$

Pode-se observar que o fator $\cos \theta$ é análogo ao fator de potência de um circuito de corrente alternada.

4.2 Radiação de Ondas Acústicas

4.2.1 Radiação de uma esfera pulsante

A fonte mais simples para gerar ondas acústicas é uma esfera pulsante, ou seja, uma esfera cujo raio varia harmonicamente com o tempo. Por simetria, este tipo de fonte produzirá uma onda esférica harmônica e divergente. A onda acústica deve ser da forma:

$$p(r,t) = \frac{A}{r} e^{j(\omega t - kr)}$$

A condição de contorno no infinito é que não há reflexão, então somente uma onda harmônica divergente existe. A amplitude "A" pode ser obtida da condição de contorno na superfície da esfera: a componente radial da velocidade das partículas deve ser igual à da superfície da esfera. Se a superfície de raio médio "a" está vibrando com amplitude de velocidade U_0 e frequência ω então:

$$u(a,t) = U_0 e^{j\omega t}$$

Essa velocidade é igual à velocidade das partículas de fluido a $r=a$. Para sabermos a pressão basta multiplicar u pela impedância específica em $r=a$.

$$Z(a) = \rho_0 c \cos \theta_a e^{j\theta_a}, \text{ onde } \cotg \theta_a = ka$$

A pressão na superfície será então:

$P(a,t) = \rho_0 c U_0 \cos \theta_a e^{j(\omega t + \theta_a)}$, que é o valor da equação da onda esférica para $r=a$. Pode-se encontrar a amplitude A comparando as duas equações para $r=a$.

$$\frac{A}{a} e^{j\omega t} e^{-jka} = \rho_0 c U_0 \cos \theta_a e^{j\omega t} e^{j\theta_a}$$

A pressão para $r \geq a$:

$$p(r,t) = \frac{\rho_0 c U_0 a \cos \theta_a}{r} e^{j(\omega t - k(r-a) + \theta_a)}$$

A velocidade é dada por:

$$u = \frac{p}{z} = U_0 \frac{a \cos \theta_a}{r \cos \theta} e^{j(\omega t - k(r-a) + \theta_a - \theta)}$$

A intensidade acústica é a seguinte:

$$I = \frac{1}{T} \int_0^T \text{Re}\{u\} \text{Re}\{p\} dt = \frac{1}{2} \rho_0 c U_0^2 \frac{a^2}{r^2} \cos^2 \theta_a$$

4.2.2 Força da fonte

Consideremos uma fonte sonora de tamanho e forma arbitrários cuja superfície oscila com uma frequência simples, mas de tal maneira que a amplitude da velocidade e a fase (U e ϕ) variam de ponto para ponto na superfície. A velocidade instantânea de um ponto na superfície é:

$$\vec{u} = \vec{U} e^{j(\omega t + \phi)}$$

Essa fonte desloca um volume de fluido à seguinte taxa:

$$Qe^{j\omega t} = \int_S \vec{u} \cdot \vec{n} dS$$

O número complexo Q indica a amplitude da taxa de volume deslocado pela fonte. Esta constante é a força complexa da fonte.

A importância da força da fonte reside no fato de que se as dimensões de uma fonte são muito menores que o comprimento de onda som sendo radiado, então os detalhes do movimento da superfície não são importantes e essa fonte radiará exatamente a mesma onda sonora que qualquer outra fonte com a mesma força de fonte, desde que essa outra fonte também tenha dimensões bem menores que o comprimento de onda da onda acústica radiada.

4.2.3 Fontes simples

Para uma onda esférica harmônica, a impedância acústica específica é:

$$Z(r) = \frac{\rho_0 c (kr)^2}{(kr)^2 + 1} + j \frac{\rho_0 c kr}{1 + (kr)^2}$$

Se uma esfera pulsante tem um raio a muito menor que o comprimento de onda da onda radiada tal que $ka \ll 1$ ($\frac{2\pi a}{\lambda} \ll 1$), então a impedância para $r=a$ será: $z(a) = \rho_0 c ka (ka + j)$

Essa impedância é altamente reativa (parte complexa bem maior). Esta reatância é um sintoma de alta divergência radial da onda acústica na fonte e representa o armazenamento e liberação de energia porque camadas

sucessivas de fluido devem se esticar e se encolher em forma de circunferência. Isto produz um efeito inercial semelhante ao efeito de uma massa.

A pressão quando $ka \ll 1$ é:

$$Ka \ll 1 \rightarrow \cos \theta_a \cong ka \rightarrow p(r, t) = j\rho_0 c U_0 \frac{a}{r} k a e^{j(\omega t - kr)} = j\rho_0 c U_0 \frac{a}{r} k a e^{j(\omega t - kr)},$$

pois $\theta_a \approx \pi/2$ rad

A defasagem entre a pressão e a velocidade da partícula é próxima de $\pi/2$. A fonte descrita acima é uma **fonte simples**. Uma fonte simples é uma superfície fechada, vibrando com distribuição de velocidades arbitrária, mas que tem todas as dimensões muito menores do que o comprimento de onda do som emitido.

A força desta fonte simples é dada por $Q=4\pi a^2 U_0$. Qualquer outra fonte simples, com a mesma força e vibrando na mesma frequência, produzirá uma onda acústica exatamente igual.

A expressão da intensidade pode ser reescrita para $ka \ll 1$ e em termos de Q . Também usando-se que $k=2\pi/\lambda$ tem-se que:

$$I = \frac{1}{8} \rho_0 c \left(\frac{Q}{r\lambda} \right)^2$$

É importante observar que essas expressões para pressão e intensidade são válidas para todas as fontes simples e independem das dimensões da fonte.

A integração da intensidade sobre a esfera centrada na fonte dá a potência radiada:

$$\Pi = \frac{\pi}{2} \rho_0 c \left(\frac{Q}{\lambda} \right)^2$$

Um outro tipo de fonte simples é uma fonte simples presa ou muito próxima a uma borda rígida plana. Se as dimensões da borda são muito maiores que o comprimento de onda do som então a borda pode ser considerada um plano de extensão infinita. Se essa superfície é perfeitamente rígida, 100% da onda incidente na superfície será refletida. O campo de pressão no meio-espaco ocupado pela fonte será duas vezes o gerado pela fonte no espaco livre (por causa da relexão de metade da onda no plano rígido).

$$p(r,t) = j\rho_0 c \frac{Qk}{2\pi r} e^{j(\omega t - kr)}$$

Pela equação de Euler, pode-se observar que, para uma pressão dobrada, a velocidade das partículas fluidas também dobrará. A intensidade para o caso de borda rígida será, portanto, quatro vezes maior que a intensidade no caso de espaço livre:

$$I = \frac{1}{2} \rho_0 c \left(\frac{Q}{r\lambda} \right)^2$$

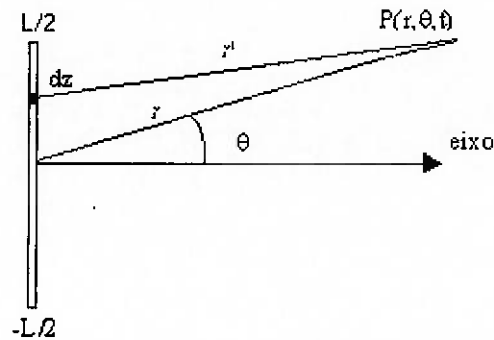
A integração da intensidade sobre um hemisfério é igual à potência radiada, que será duas vezes maior que o caso de espaço livre. Isso é explicado da seguinte forma: a superfície da fonte continua se movendo com a mesma velocidade, mas está trabalhando com uma força duas vezes maior; como consequência, a fonte deve gastar uma potência duas vezes maior para manter seu próprio movimento na presença de uma pressão duas vezes maior.

4.2.4 Radiação de uma fonte tipo linha

Consideremos um cilindro longo e fino funcionando como fonte de radiação acústica. Esse cilindro vibra radialmente com velocidade de partícula harmônica. Consideremos ainda que esse cilindro esteja preso a uma borda rígida. Se o cilindro for suficientemente fino (raio bem menor do que o comprimento de onda da onda acústica emitida), podemos dividir este cilindro em fontes simples de comprimento infinitesimal.

A onda acústica produzida por esta fonte será descrita pelo campo de pressão acústica. A pressão acústica em cada ponto do espaço é a somatória das pressões acústicas produzidas por cada fonte simples que forma a fonte.

Pode-se usar a seguinte configuração para encontrar o campo de pressões em qualquer ponto do espaço:



Para simplificar, a solução encontrada será para $r \gg L$. Essa aproximação é chamada de **aproximação de campo distante**.

Pode ser deduzido (pela somatória dos efeitos das fontes simples que formam a linha) que o valor da pressão num ponto qualquer é dada por:

$$p(r, \theta, t) = j\rho_0 c U_0 \frac{a}{r} k L e^{j(\omega t - kr)} \left[\frac{\sin\left(\frac{1}{2} k L \sin \theta\right)}{\frac{1}{2} k L \sin \theta} \right]$$

O fator $\frac{\sin\left(\frac{1}{2} k L \sin \theta\right)}{\frac{1}{2} k L \sin \theta} = H(\theta)$ é chamado de **fator direcional**. O

restante da expressão é função de r e t e corresponde ao campo de pressão no eixo perpendicular à fonte ($\theta=0$)

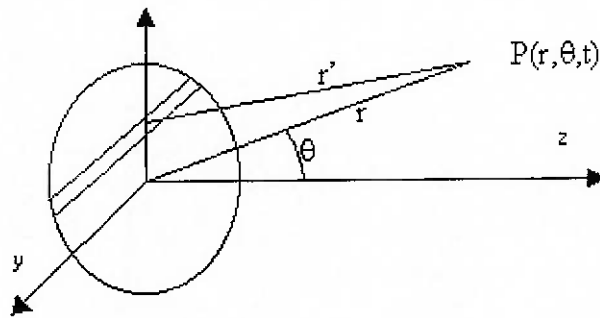
Este procedimento de separar a expressão para a pressão no campo distante em um fator que depende somente do ângulo (ou ângulos) e outro que depende somente da distância é uma prática comum na descrição de campos acústicos de fontes complicadas.

4.2.5 Radiação de um pistão plano

O tipo de fonte de importância especial para o projeto é o pistão plano montado sobre uma superfície rígida vibrando em movimento harmônico simples. A solução deste exemplo é aplicável, por exemplo, à face emissora de um transdutor piezoelétrico cilíndrico. Será calculado o campo de pressões no campo distante, ou seja, para $r \gg a$, sendo r a distância do ponto até o centro do pistão e a o raio do pistão.

Dada a posição (r, θ) no qual a pressão é desejada, pode-se adotar um sistema de coordenadas de forma que o eixo z seja o eixo do pistão e que o ponto (r, θ) fique no plano xz . Isto permite que a superfície do pistão seja quebrada em várias fontes tipo linha paralelas ao eixo y , de modo que o ponto fique no eixo de cada fonte tipo linha.

O esquema a seguir mostra o que foi explicado:



No campo distante, a contribuição de cada uma das fontes tipo linha é simplesmente o campo de pressão axial de cada fonte.

Pode ser mostrado (realizando a integração dos efeitos das fontes tipo linha) que o valor da pressão no campo distante é dado por:

$$p(r, \theta, t) = j\rho_0 c U_0 \frac{a}{r} k a e^{j(\omega t - kr)} \left(\frac{J_1(ka \sin \theta)}{ka \sin \theta} \right)$$

Como o valor máximo de $\frac{J_1(ka \sin \theta)}{ka \sin \theta}$ é 0,5 (quando $\theta=0$), para que o fator direcional seja igual a 1 quando $\theta=0$, podemos reescrever a expressão do campo de pressão do seguinte modo:

$$p(r, \theta, t) = j\rho_0 \frac{c}{2} U_0 \frac{a}{r} k a e^{j(\omega t - kr)} \left(\frac{2J_1(ka \sin \theta)}{ka \sin \theta} \right)$$

4.2.6 Impedância de Radiação

Consideremos um transmissor cuja face ativa é movida com uma velocidade que pode ser uma função da posição na face. Se df_s é a componente da força na direção do movimento que é feita sobre o fluido por um elemento da face ativa, a impedância de radiação é:

$$Z_r = \int \frac{df s}{u}$$

Z_r é um número complexo e pode ser dividido em parte real e imaginária:
 $Z_r = R_r + jX_r$, onde R_r é a resistência de radiação e X_r é a reatância de radiação.

O cálculo da impedância de radiação para um pistão plano montado numa borda rígida é muito complicado e pode ser encontrado na literatura (Kinster 1962). Os resultados são:

$$R_r = \Pi a^2 \rho_0 c R_1(2ka)$$

$$X_r = \Pi a^2 \rho_0 c X_1(2ka), \text{ sendo:}$$

$$R_1(x) = 1 - 2J_1(x)/x$$

$$X_1(x) = \frac{4}{\pi} \left(\frac{x}{3} - \frac{x^3}{3^2 5} + \frac{x^5}{3^2 5^2 7} - \dots \right)$$

Para uma esfera pulsante o cálculo é bem mais simples. Basta multiplicar a impedância acústica em $r=a$ pela área da superfície esférica ativa, pois a pressão e a velocidade são uniformes sobre a superfície da fonte:

$$Z_r = 4 \Pi a^2 \rho_0 c \cos \theta_a e^{j\theta_a}, \text{ sendo } \cotg \theta_a = ka$$

4.3 Absorção e Atenuação de Ondas Sonoras

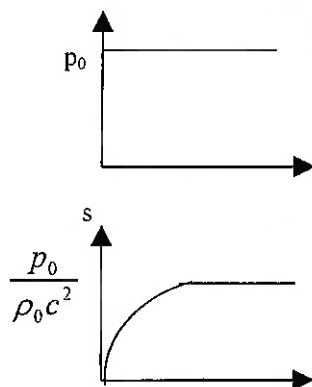
Nas discussões anteriores a dissipação de energia acústica foi considerada. Em muitas situações a dissipação é muito lenta e pode ser ignorada para pequenas distâncias a pequenos intervalos de tempo. Entretanto, em qualquer caso, toda energia acústica se transforma em energia térmica. Será abordada a seguir a dissipação relacionada ao meio no qual a onda se propaga.

4.3.1 Aproximação para a absorção

Uma consequência de ignorar todos os mecanismos de perda é que a pressão acústica e a condensação (variação percentual de densidade) estão em fase e sua relação é dada pela equação linear de estado:

$$p = \rho_0 c^2 s$$

Uma maneira de introduzir perdas é modificar esta equação de estado para permitir um atraso entre a aplicação de uma súbita mudança de pressão e a mudança de condensação até atingir o valor de equilíbrio expresso pela equação de estado.



A resposta de s em função do tempo mostrada ao lado pode ser aproximada por uma exponencial:

$$S(t) = \frac{p_0}{\rho_0 c^2} - \frac{p_0}{\rho_0 c^2} e^{-\frac{t}{\tau}}$$

A análise da equação mostra que, quando $t=0$, $s=0$; e, à medida que o tempo aumenta, s tende ao valor de equilíbrio $\frac{p_0}{\rho_0 c^2}$. A constante τ é a constante de tempo (instante de tempo no qual o expoente de "e" é -1 e o valor

de s é 63,2% do valor de equilíbrio). Essa constante é chamada de **tempo de relaxação**.

O sistema pode ser aproximado por um sistema de 1ª ordem com constante de tempo τ e ganho $\rho_0 c^2$:

$$\frac{S(s)}{P(s)} = \frac{1}{\tau s + 1} \rightarrow \left(s(t) + \tau \frac{\partial s(t)}{\partial t} \right) \rho_0 c^2 = p(t), \text{ que é a equação de estado}$$

modificada. Se $s(t)$ é proporcional a $\exp(j\omega t)$ (onda harmônica), a equação de estado com acréscimo de dissipação fica:

$$p(t) = \rho_0 c^2 (1 + \tau j\omega) s(t).$$

$$\text{Pode-se definir } \vec{c}^2 = c^2 (1 + \tau j\omega) \rightarrow p = \vec{c}^2 s \rho_0$$

Para deduzir a equação da onda acústica deve-se utilizar a equação da continuidade e a equação de Euler (desenvolvidas em itens anteriores):

$$\frac{\partial s}{\partial t} + \nabla \vec{u} = 0 \rightarrow \frac{\partial^2 s}{\partial t^2} + \frac{\partial(\nabla \vec{u})}{\partial t} = \frac{\partial^2 s}{\partial t^2} + \nabla \cdot \frac{\partial \vec{u}}{\partial t} = 0. \text{ Sabe-se também que:}$$

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\nabla p \rightarrow \rho_0 \nabla \cdot \frac{\partial \vec{u}}{\partial t} = -\nabla \cdot \nabla p = -\nabla^2 p. \text{ Substituindo } \nabla \cdot \frac{\partial \vec{u}}{\partial t} \text{ na primeira}$$

equação tem-se:

$$\frac{\partial^2 s}{\partial t^2} - \frac{\nabla^2 p}{\rho_0} = 0. \text{ Da equação de estado: } \frac{\partial^2 s}{\partial t^2} = \frac{1}{\rho_0 \vec{c}^2} \frac{\partial^2 p}{\partial t^2}. \text{ A equação de}$$

onda acústica é:

$$\frac{1}{\vec{c}^2} \frac{\partial^2 p}{\partial t^2} = \nabla^2 p, \text{ que é análoga à equação de onda sem atenuação. Por}$$

analogia, as soluções desta equação diferencial são funções com argumentos $(\vec{c} t - x)$ para ondas planas e $(\vec{c} t - r)$ para ondas esféricas.

Portanto, todas as soluções vistas para ondas esféricas e planas podem ser usadas, desde que se faça a substituição de c por \vec{c} e de k por $\vec{k} \left(\vec{k} = \frac{\omega}{\vec{c}} \right)$.

O número de onda complexo é: $\vec{k} = \frac{\omega}{c}$, que pode ser dividido em parte

real e imaginária. Assim:

$$\vec{k} = k - j\alpha = \frac{\omega}{c\sqrt{1+j\omega\tau}} = \frac{\omega\sqrt{1-j\omega\tau}}{c\sqrt{1+(\omega\tau)^2}}$$

$$k^2 - \alpha^2 = \frac{\omega^2}{c^2} \frac{1}{1+(\omega\tau)^2} \text{ (parte real)}$$

$$2k\alpha = \frac{\omega^2}{c^2} \frac{\omega\tau}{1+(\omega\tau)^2} \text{ (parte imaginária)}$$

Resolvendo o sistema tem-se:

$$\alpha = \frac{\omega}{c} \frac{1}{\sqrt{2}} \left[\frac{\sqrt{1+(\omega\tau)^2} - 1}{1+(\omega\tau)^2} \right]^{\frac{1}{2}} \text{ (coeficiente de absorção)}$$

$$k = \frac{\omega}{c} \frac{1}{\sqrt{2}} \left[\frac{\sqrt{1+(\omega\tau)^2} + 1}{1+(\omega\tau)^2} \right]^{\frac{1}{2}} \text{ (constante de propagação)}$$

A intensidade acústica para uma onda plana considerando-se a dissipação é:

$$I = \frac{P_0^2 e^{-2\alpha x}}{2\rho_0 c}$$

Observa-se que a intensidade diminui conforme a onda se propaga. A mudança da intensidade em decibéis da posição 0 até a posição x é:

$$10 \log \frac{I(0)}{I(x)} = 10 \log e^{2\alpha x} = 10 * 2\alpha x \log e = ax$$

O termo **a** do último membro da equação acima é o coeficiente de absorção em decibéis por metro e pode ser relacionado diretamente com α .

$$a = 8,686\alpha$$

4.3.2 O fenômeno da absorção em água do mar

Existem vários mecanismos macroscópicos e microscópicos que explicam a absorção acústica nos líquidos. Na água do mar os mecanismos

principais são a relaxação estrutural e a relaxação química . [kinsler] fornece uma fórmula para o cálculo do coeficiente de absorção em decibéis por metro na água do mar. Neste trabalho não serão explicados os mecanismos nem os detalhes para obtenção das equações.

O coeficiente de absorção da água do mar é (valores empíricos para pH=8 e salinidade=35 partículas por milhar):

$$\alpha = \frac{A f_1 f^2}{f_1^2 + f^2} + \frac{B f_2 f}{f_2^2 + f^2} + C f^2, \text{ onde:}$$

f: frequência da onda acústica em hertz;

$$f_1: 1,32 \cdot 10^3 (T+273) e^{-1700/(T+273)}$$

$$f_2: 1,55 \cdot 10^7 (T+273) e^{-3052/(T+273)}, T: \text{temperatura em } ^\circ\text{C}.$$

As constantes A,B e C são as seguintes (para temperaturas entre 0° e 30° C e para pressões entre 1 e 400 atm)

- $A=8,95 \cdot 10^{-8} (1+2,3 \cdot 10^{-2} T-5,1 \cdot 10^{-4} T^2)$
- $B=4,88 \cdot 10^{-7} (1+1,3 \cdot 10^{-2} T)(1-0,9 \cdot 10^{-3} P_0)$
- $C=4,76 \cdot 10^{-13} (1-4,0 \cdot 10^{-2} T+5,9 \cdot 10^{-4} T^2) * (1-3,8 \cdot 10^{-4} P_0)$

P_0 : pressão em atmosferas.

4.4 Comentários sobre o estudo de acústica

Foi estudada a equação da onda acústica e suas soluções mais simples. Este estudo foi importante para a compreensão dos fenômenos que envolvem a geração e propagação de ondas acústicas.

A seção sobre radiação de ondas acústicas possui uma relação mais estreita com o projeto, já que ondas acústicas serão geradas pelos transdutores emissores, sejam eles esféricos ou cilíndricos. Os modelos de onda gerada por fonte esférica e por pistão plano são particularmente úteis para o modelamento do comportamento do campo acústico produzido pelos transdutores.

No estudo da atenuação, foi observado que o coeficiente de absorção aumenta com a frequência. Para 1 MHz, o coeficiente de absorção é de 0,4 dB/m. Essa atenuação é considerável para longas distâncias. No modelamento da propagação de ondas acústicas, deve-se sempre calcular a atenuação, já que, para altas frequências, ela não pode ser desprezada.

Também foi feito um estudo sobre a transmissão e a reflexão de ondas sonoras, embora esse estudo não esteja documentado aqui. Esse estudo tenta modelar o comportamento de uma onda acústica quando a mesma atinge uma interface entre dois meios de impedâncias acústicas diferentes. Este estudo será útil particularmente no modelamento de transdutores receptores, pois os mesmos recebem uma onda acústica na sua face receptora; uma parte dessa onda é refletida e outra parte é responsável pela excitação do transdutor. Os cálculos das transmissões e reflexões são feitos com base nas condições de continuidade de pressão e velocidade nessas interfaces.

5 Estudo De Transdutores Piezelétricos

5.1 Transdutores piezelétricos

Os materiais piezelétricos são largamente utilizados na fabricação de transdutores acústicos. Esses transdutores convertem energia acústica (vibração mecânica) em energia elétrica e vice-versa. Os transdutores piezelétricos são usados especialmente em aplicações de frequência ultrassônica (onda acústica inaudível com frequência superior a 20 kHz).

Um transdutor piezelétrico é formado basicamente por uma cerâmica piezelétrica e 2 eletrodos.

Quando uma voltagem variável é aplicada entre os terminais, a cerâmica vibra e produz uma onda acústica.

O inverso também ocorre: quando uma onda acústica excita a cerâmica, uma diferença de potencial variável aparece entre os terminais.

Além da cerâmica piezelétrica também podem ser usados num transdutor outras duas camadas : a de matching e a de backing.

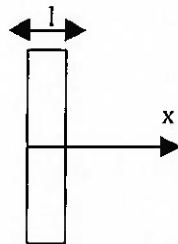
A camada de matching é utilizada para aumentar a largura de banda em torno da frequência de ressonância para que dessa forma o transdutor possa atuar numa faixa maior de frequência. Outro ponto importante é que o uso da camada de matching aumenta o coeficiente de transmissão da onda acústica entre a cerâmica e o meio.

A camada de backing pode ser usada para aumentar a eficiência, dependendo do meio de atuação do transdutor e também para aumentar a atenuação, ou seja, para que quando a cerâmica começa a vibrar ela pare de vibrar depois de um curto tempo.

Como referências para os fundamentos e o modelamento dos transdutores foi usada a referência [3] e também serviu de base um outro trabalho de formatura feito nesta área – referência [5].

5.1.1 Modelo matemático simplificado

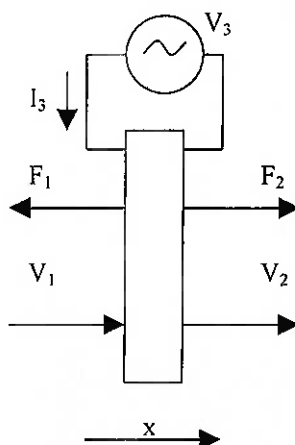
Será deduzido um modelo matemático simplificado de um transdutor com as dimensões da seção transversal de muitos comprimentos de onda que vibram no modo de espessura. Com essa hipótese, pode-se garantir que a onda acústica que percorre o transdutor é uma onda plana. Conforme o esquema abaixo, a direção de propagação da onda e de vibração do transdutor é a direção de x .



A cerâmica piezelétrica pode ser estudada como um sistema tri-porto. Cada porto é um local (físico ou conceitual) através do qual um elemento troca energia.

Um transdutor pode ser estudado como uma “caixa preta” com dois portos mecânicos (pontos onde ocorrem fluxo de energia mecânica) e um porto elétrico.

Isto pode ser representado como no esquema a seguir:



- F_1 e F_2 são forças externas aplicadas ao transdutor.

A convenção de sinais adotada encontra-se no apêndice “Convenção de Sinais”

Portanto, as condições de contorno dos portos acústicos são:

$$F_1 = -AT(0)$$

$$F_2 = AT(l)$$

$$V_1 = v(0)$$

$$V_2 = v(l)$$

Onde $T(x_1, t)$ é a tensão mecânica dos elementos no plano $x=x_1$ em função de t e $V(x_1, t)$ é a velocidade dos elementos no plano $x=x_1$ em função de t .

Neste modelo, será assumido que todas as grandezas variam harmonicamente (grandezas proporcionais a $\exp(j\omega t)$). Assim, as seguintes grandezas serão expressas como números complexos (possuem módulo e fase) que são proporcionais a $\exp(j\omega t)$:

$$S(x, t) = S_0 e^{j\omega t} = S - \text{deformação}$$

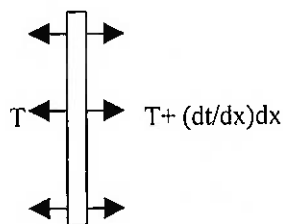
$$D(x, t) = D_0 e^{j\omega t} = D - \text{deslocamento elétrico (carga/área)}$$

$$U(x, t) = u_0 e^{j\omega t} = u - \text{deslocamento da partícula}$$

$$V(t) = V_0 e^{j\omega t} = V_3 - \text{voltagem}$$

$$I(t) = I_0 e^{j\omega t} = I_3 - \text{corrente elétrica.}$$

A fim de encontrar a equação de movimento, pode-se separar um elemento de volume



A força resultante sobre o elemento é: $(T + (dT/dx)dx) - T$.

E a equação de movimento será:

$$\frac{dT}{dx} dx dA = dm \frac{dv}{dt} = \rho_0 (dx dA) \frac{dv}{dt} \rightarrow \frac{dT}{dx} = \rho_0 \frac{dv}{dt}$$

A deformação é definida como:

$$S = \frac{\Delta l}{l_0} = \frac{u + \frac{\partial u}{\partial x} \Delta x - u}{\Delta x} = \frac{du}{dx}$$

Pode-se derivar em relação ao tempo:

$$\frac{dS}{dt} = \frac{\frac{du}{dt}(t + \Delta t) - \frac{du}{dt}(t)}{\Delta t} = \frac{dv}{dx}$$

Como v e S são harmônicas, pode-se escrever:

$$\frac{dv}{dx} = j\omega S \text{ e } \frac{dT}{dx} = \rho_0 j\omega v$$

A corrente elétrica é a taxa de variação da carga armazenada em relação ao tempo:

$$I = dQ/dt = A j \omega D$$

Pela convenção adotada na figura: $V_3 = V(0) - V(l)$

$$\text{Sabe-se que } V(l) - V(0) = - \int_0^l E dx \rightarrow V_3 = \int_0^l E dx$$

Conforme visto anteriormente as relações constitutivas do material são:

$$D = \epsilon E + eS$$

$$T = c^E S - eE \rightarrow E = \frac{c^E S - T}{e} . \text{ Assim:}$$

$$D = \frac{\epsilon}{e} (c^E S - T) + eS . \text{ Definindo } h = e/\epsilon \text{ e } c^D = c^E + e^2/\epsilon, \text{ tem-se:}$$

$$T = c^D S - hD$$

Derivando em relação a x :

$$\frac{dT}{dx} = c^D \frac{dS}{dx} - h \frac{dD}{dx} . \text{ Como não há cargas livres no interior da cerâmica,}$$

$$Dd/dx = 0.$$

Sabe-se das equações anteriores que:

$$\frac{dS}{dx} = \frac{d^2 v}{dx^2} \frac{1}{j\omega} \text{ e } \frac{dT}{dx} = \rho_0 j\omega v . \text{ Portanto, tem-se:}$$

$$\rho_0 j\omega v - c^D \frac{d^2 v}{dx^2} \frac{1}{j\omega} = 0 . \text{ Assim:}$$

$$\frac{d^2 v}{dx^2} + \frac{\omega^2 \rho_0 v}{c^D} = 0$$

A equação acima é a equação da onda acústica harmônica em termos da velocidade de partícula. A solução geral é da forma:

$V = Ae^{-j\beta x} + Be^{j\beta x}$, onde A e B são constantes complexas a serem determinadas pelas condições de contorno.

$$\text{Sendo } \beta = \omega \sqrt{\frac{\rho_0}{c^D}}.$$

Pode-se também determinar T(x), sabendo-se que:

$$T = c^D S - hD \text{ e que } S = \frac{dv}{dx} \frac{1}{j\omega}.$$

Portanto:

$$T = -c^D \frac{\beta}{\omega} A e^{-j\beta x} + c^D \frac{\beta}{\omega} B e^{j\beta x} - hD$$

Vamos agora determinar as constantes A e B. Usando as condições de contorno, temos:

$$v(0) = v_1 = A + B$$

$$v(l) = v_2 = A e^{-j\beta l} + B e^{j\beta l}.$$

Resolvendo o sistema, tem-se:

$$A = (v_2 - v_1 e^{j\beta l}) / (e^{-j\beta l} - e^{j\beta l}) \text{ e}$$

$$B = (v_1 e^{-j\beta l} - v_2) / (e^{-j\beta l} - e^{j\beta l})$$

Pode-se então determinar v(x). Assim:

$$v(x) = [(v_2 - v_1 e^{j\beta l}) / (e^{-j\beta l} - e^{j\beta l})]^* e^{-j\beta x} + [(v_1 e^{-j\beta l} - v_2) / (e^{-j\beta l} - e^{j\beta l})]^* e^{j\beta x}$$

Desenvolvendo a expressão acima tem-se:

$$v(x) = \frac{v_1 \sin[\beta(l-x)] + v_2 \sin(\beta x)}{\sin(\beta l)}$$

Também desta forma determina-se que:

$$T = -\frac{c^D \beta}{\omega} \frac{v_2 - v_1 e^{j\beta l}}{-2j \sin(\beta l)} e^{-j\beta x} + \frac{c^D \beta}{\omega} \frac{v_1 e^{-j\beta l} - v_2}{-2j \sin(\beta l)} e^{j\beta x} - h \frac{I_3}{j\omega A}$$

$$T(0) = -\frac{F_1}{A} \Rightarrow F_1 = -j(-v_2 \cos ec(\beta l) \frac{c^D \beta A}{\omega} + v_1 \cot g(\beta l) \frac{c^D \beta A}{\omega} + \frac{h}{\omega} I_3)$$

$$T(l) = \frac{F_2}{A} \Rightarrow F_2 = j(v_1 \cos ec(\beta l) \frac{c^D \beta A}{\omega} + v_2 \cot g(\beta l) \frac{c^D \beta A}{\omega} + \frac{h}{\omega} I_3)$$

Vamos integrar em x a seguinte relação constitutiva:

$$D = \epsilon E + eS \rightarrow \int_0^l D dx = \epsilon \int_0^l E dx + e \int_0^l S dx$$

Das equações anteriores pode-se determinar que:

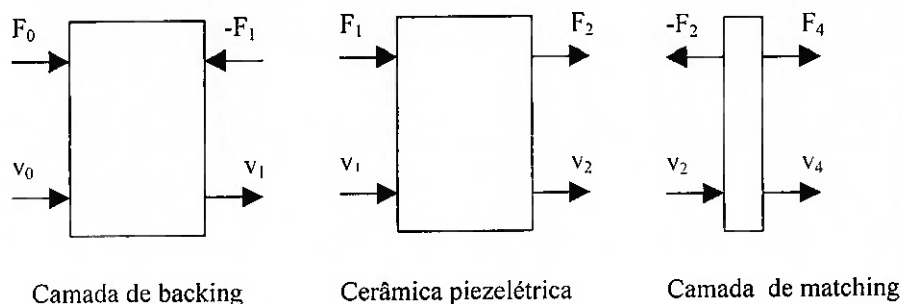
$$\frac{1}{j\omega A} \int_0^l I_3 dx = \epsilon V_3 + \frac{e}{j\omega} \int_0^l \frac{dv}{dx} dx \Rightarrow V_3 = -j \left(\frac{h}{\omega} v_1 - \frac{h}{\omega} v_2 + I_3 \frac{l}{\omega A \epsilon} \right)$$

5.1.2 Transdutor de três camadas

Um transdutor piezelétrico cilíndrico normalmente é constituído de três camadas: camada de matching, cerâmica piezelétrica e camada de backing.

Deseja-se calcular a amplitude de velocidade de vibração da face do transdutor que está em contato com o fluido quando uma tensão harmônica de entrada V_{in} é imposta.

O equacionamento visto no item anterior pode ser feito para as três camadas tomando-se os devidos cuidados com as condições de contorno

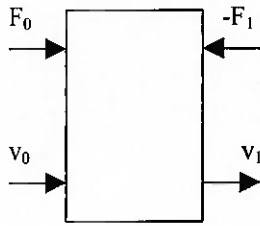


A face traseira da camada de backing e a face dianteira da camada de matching estão em contato com o meio fluido, que será considerado infinito. A razão entre a força sobre o fluido e a velocidade de vibração da força ativa é a impedância de radiação. Se considerarmos as superfícies ativas como pistões planos, já temos calculada a impedância de radiação em uma seção anterior. Assim:

$$Z_r = -\frac{F_0}{v_0} = -\frac{F_4}{v_4}$$

Para determinar as equações da camada de backing primeiramente será feita a introdução de absorção nesta camada. Dessa forma tem-se:

Dado o coeficiente de absorção de um material em nepers/m (α), vamos determinar as equações para as forças que agem em ambos os lados da camada em função das velocidades das faces.



Conforme visto na seção sobre atenuação, a atenuação pode ser aproximada com a introdução do número de onda complexo e da velocidade de propagação complexa. O número de onda complexo é: $\vec{k} = k - j\alpha$, onde α é o coeficiente de atenuação em nepers/metro. Para o sólido da camada de backing, vamos fazer a seguinte aproximação:

$$\vec{k} = \frac{\omega}{c} - j\alpha$$

Pode-se observar do equacionamento para uma camada do transdutor que:

$$\beta = \omega \sqrt{\frac{\rho_0}{c_D}} = \omega \sqrt{\frac{\rho_0}{c^E + \frac{e^2}{\varepsilon}}} = \frac{\omega}{c}$$

Para introduzir a dissipação basta substituir β pelo número de onda complexo $\vec{\beta} = \beta - j\alpha$ e resolver a equação da onda acústica:

$$\frac{d^2 v}{dx^2} + \vec{\beta}^2 v = 0. \text{ A solução geral é:}$$

$$v = Ae^{-j\vec{\beta}x} + Be^{j\vec{\beta}x} \Rightarrow v = Ae^{-\alpha x} e^{-j\beta x} + Be^{\alpha x} e^{j\beta x}$$

As condições de contorno são:

$$v(0) = v_0 = A + B \Rightarrow B = v_0 - A$$

$$v(l) = Ae^{-\alpha l} e^{-j\beta l} + Be^{\alpha l} e^{j\beta l} = v_1$$

Substituindo a primeira condição de contorno na segunda tem-se:

$$Ae^{-\alpha l} e^{-j\beta l} + (v_0 - A)e^{\alpha l} e^{j\beta l} = v_1 \Rightarrow A = \frac{v_1 - v_0 e^{\alpha l} e^{j\beta l}}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}}$$

$$B = v_0 - A = \frac{v_0 e^{-\alpha l} e^{-j\beta l} - v_1}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}}$$

Como o material da camada de backing não é piezoelétrico temos:

$$T = c^E S = c^E \frac{dv}{dx} \frac{1}{j\omega}. \text{ Sabe-se que:}$$

$$\frac{dv}{dx} = -j\vec{\beta} A e^{-j\vec{\beta}x} + j\vec{\beta} B e^{j\vec{\beta}x}. \text{ Também sabe-se que:}$$

$$S = \frac{dv}{dx} \frac{1}{j\omega} = \frac{-\vec{\beta}}{\omega} A e^{-j\vec{\beta}x} + \frac{\vec{\beta}}{\omega} B e^{j\vec{\beta}x}. \text{ Dessa forma tem-se:}$$

$$T = -c^E \frac{\vec{\beta}}{\omega} A e^{-j\vec{\beta}x} + c^E \frac{\vec{\beta}}{\omega} B e^{j\vec{\beta}x}$$

Pelas condições de contorno sabemos que:

$$F_0 = -AT(0)$$

$$F_1 = -AT(l)$$

Assim:

$$F_0 = -A \left[-c^E \frac{\vec{\beta}}{\omega} \left(\frac{-v_0 e^{\alpha l} e^{j\beta l} + v_1}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) + c^E \frac{\vec{\beta}}{\omega} \left(\frac{v_0 e^{-\alpha l} e^{-j\beta l} - v_1}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) \right]. \quad \text{Dessa}$$

forma tem-se:

$$F_0 = v_0 \left[-Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{e^{\alpha l} e^{j\beta l} + e^{-\alpha l} e^{-j\beta l}}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) \right] + v_1 \left[Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{2}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) \right]$$

Temos também que:

$$F_1 = -A \left[-c^E \frac{\vec{\beta}}{\omega} \left(\frac{-v_0 e^{\alpha l} e^{j\beta l} + v_1}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) e^{-j\vec{\beta}l} + c^E \frac{\vec{\beta}}{\omega} \left(\frac{v_0 e^{-\alpha l} e^{-j\beta l} - v_1}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) e^{j\vec{\beta}l} \right]$$

Assim tem-se:

$$F_1 = v_0 \left[-Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{e^{\alpha l} e^{j\beta l} e^{-j\vec{\beta}l} + e^{-\alpha l} e^{-j\beta l} e^{j\vec{\beta}l}}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) \right] + v_1 \left[Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{e^{-j\vec{\beta}l} + e^{j\vec{\beta}l}}{e^{-\alpha l} e^{-j\beta l} - e^{\alpha l} e^{j\beta l}} \right) \right]$$

Podemos reescrever as expressões notando que:

$$e^{j\vec{\beta}l} = e^{j(\beta - j\alpha)l} = e^{\alpha l} e^{j\beta l}$$

$$F_0 = v_0 \left[-Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{e^{j\vec{\beta}l} + e^{-j\vec{\beta}l}}{e^{-j\vec{\beta}l} - e^{j\vec{\beta}l}} \right) \right] + v_1 \left[Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{2}{e^{-j\vec{\beta}l} - e^{j\vec{\beta}l}} \right) \right]$$

$$F_1 = v_0 \left[-Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{2}{e^{-j\vec{\beta}l} - e^{j\vec{\beta}l}} \right) \right] + v_1 \left[Ac^E \frac{\vec{\beta}}{\omega} \left(\frac{e^{-j\vec{\beta}l} + e^{j\vec{\beta}l}}{e^{-j\vec{\beta}l} - e^{j\vec{\beta}l}} \right) \right]$$

Para o caso em que a camada de backing não existir temos:

$$F_0 = F_1 \text{ e como } F_0 = -Z_r v_0 \Rightarrow 0 = F_1 + Z_r v_0$$

$$v_0 = v_1 \Rightarrow 0 = v_1 - v_0$$

As equações para a cerâmica piezelétrica são:

$$F_1 = -j(-v_2 A \frac{c_c^D \beta_c}{\omega} \operatorname{cosec}(\beta_c l_c) + v_1 A \frac{c_c^D \beta_c}{\omega} \cot g(\beta_c l_c) + \frac{h}{\omega} I_3)$$

$$F_2 = j(v_1 A \frac{c_c^D \beta_c}{\omega} \operatorname{cosec}(\beta_c l_c) - v_2 A \frac{c_c^D \beta_c}{\omega} \cot g(\beta_c l_c) + \frac{h}{\omega} I_3)$$

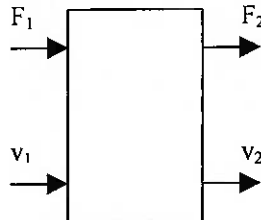
$$V_{in} = -j(\frac{h}{\omega} v_1 - \frac{h}{\omega} v_2 + \frac{l}{A\omega e} I_3)$$

Deve-se notar que a colocação de uma camada de matching entre a cerâmica piezelétrica e o meio é opcional. Sendo assim, o modelo deve ser capaz de resolver o transdutor com ou sem camada de matching. Esta camada também é modelada como um disco de comprimento l.

Se $l=0$ (não há camada de matching) as equações para esta camada são:

$$F_1 = -F_2$$

$$v_1 = v_2$$



(A força externa aplicada à camada à direita é igual à força aplicada pela camada no corpo à esquerda)

Se $l>0$ as equações para esta camada são as seguintes:

$$F_1 = v_1 \left(-j \cot g(\beta l) \frac{c^E \beta A}{\omega} \right) + v_2 \left(j \operatorname{cosec}(\beta l) \frac{c^E \beta A}{\omega} \right)$$

$$F_2 = v_1 \left(j \operatorname{cosec}(\beta l) \frac{c^E \beta A}{\omega} \right) + v_2 \left(-j \cot g(\beta l) \frac{c^E \beta A}{\omega} \right)$$

Como $F_4 = -Z_r v_4$ e se $l_m=0$ (ausência da camada de matching) então:

$$F_4 = F_2 \Rightarrow 0 = F_2 + Z_r v_4$$

$$v_4 = v_2 \Rightarrow 0 = v_2 - v_4$$

Tem-se, portanto, 7 equações e 7 incógnitas. Pode-se montar um sistema linear 7x7. A representação matricial deste sistema é dada logo a seguir:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_m \end{bmatrix} = \begin{bmatrix} 0 & D(1,2) & 0 & 0 & D(1,5) & 0 & 0 \\ 1 & D(2,2) & 0 & 0 & D(2,5) & 0 & 0 \\ 0 & 0 & -1 & D(3,4) & 0 & D(3,6) & 0 \\ 0 & 0 & 0 & D(4,4) & 0 & D(4,6) & 0 \\ 1 & D(5,2) & 0 & D(5,4) & 0 & 0 & D(5,7) \\ 0 & D(6,2) & -1 & D(6,4) & 0 & 0 & D(6,7) \\ 0 & D(7,2) & 0 & D(7,4) & 0 & 0 & D(7,7) \end{bmatrix} \begin{bmatrix} F_1 \\ v_1 \\ F_2 \\ v_2 \\ v_0 \\ v_4 \\ I_3 \end{bmatrix}$$

Onde:

$$D(1,2) = -j \operatorname{cosec}(\beta_B l_B) c_B^D \beta_B A / \omega$$

$$D(1,5) = -Z_r + j \cotg(\beta_B l_B) c_B^D \beta_B A / \omega$$

$$D(2,2) = -j(A_{CB}^D / \omega) \beta_B \cotg(\beta_B l_B)$$

$$D(2,5) = j(A_{CB}^D / \omega) \beta_B \operatorname{cosec}(\beta_B l_B)$$

$$D(3,4) = j \cotg(\beta_m l_m) c_m^D \beta_m A / \omega$$

$$D(3,6) = -j \operatorname{cosec}(\beta_m l_m) c_m^D \beta_m A / \omega$$

$$D(4,4) = j \operatorname{cosec}(\beta_m l_m) c_m^D \beta_m A / \omega$$

$$D(4,6) = Z_r - j \cotg(\beta_m l_m) c_m^D \beta_m A / \omega$$

$$D(5,2) = j \cotg(\beta_c l_c) c_c^D \beta_c A / \omega$$

$$D(5,4) = -j \operatorname{cosec}(\beta_c l_c) c_c^D \beta_c A / \omega$$

$$D(5,7) = j e / \epsilon \omega$$

$$D(6,2) = j \operatorname{cosec}(\beta_c l_c) c_c^D \beta_c A / \omega$$

$$D(6,4) = -j \cotg(\beta_c l_c) c_c^D \beta_c A / \omega$$

$$D(6,7) = j e / \epsilon \omega$$

$$D(7,2) = -j e / \epsilon \omega$$

$$D(7,4) = j e / \epsilon \omega$$

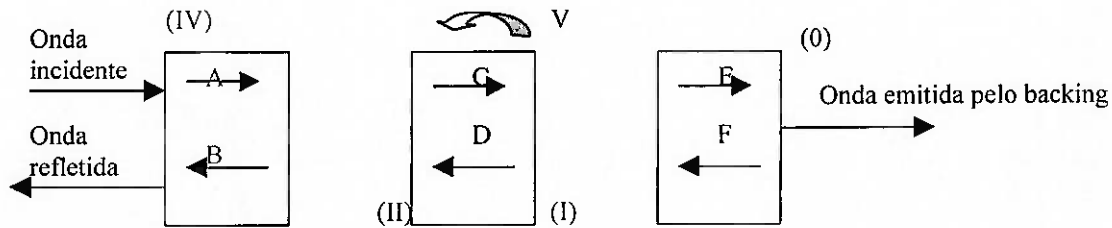
$$D(7,7) = -j l_c / A \omega \epsilon$$

5.1.3 Modelo do receptor

O modelo matemático obtido anteriormente é para um emissor, isto é, a entrada é uma voltagem e a saída é a onda acústica produzida.

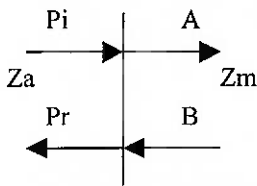
Num receptor, uma onda acústica excita a face do transdutor e uma tensão aparece entre os terminais.

O esquema abaixo ilustra a situação acima descrita:



Realizando-se a condição de continuidade em cada face, pode-se encontrar o coeficiente de reflexão (R) para o transdutor.

Continuidade na face (IV):



$$P_i + P_r = A + B \text{ (Pressões)}$$

$$u_i + u_r = u_A + u_B \text{ (velocidades)}$$

Assim:

$$\frac{P_i + P_r}{u_i + u_r} = \frac{A + B}{u_A + u_B}$$

$$\text{Para a água: } P_i = Z_A u_i \Rightarrow u_i = \frac{P_i}{Z_A}$$

$$P_r = -Z_A u_r \Rightarrow u_r = -\frac{P_r}{Z_A}$$

$$\text{Para o transdutor: } A = -Z_m u_A \Rightarrow u_A = -\frac{P_A}{Z_m}$$

$$B = Z_m u_B \Rightarrow u_B = \frac{P_B}{Z_m}$$

Portanto:

$$Z_A \frac{P_i + P_r}{P_i - P_r} = \frac{A + B}{A - B} Z_m$$

Para a face (II):

$$Z_m \frac{Ae^{-jk_m l_m} + Be^{jk_m l_m}}{Ae^{-jk_m l_m} - Be^{jk_m l_m}} = Z_c \frac{C + D + j \frac{h}{\omega A} I}{C - D}$$

Para a face (I):

$$Z_c \frac{Ce^{-jk_c l_c} + De^{jk_c l_c} j \frac{h}{\omega A} I}{Ce^{-jk_c l_c} - De^{jk_c l_c}} = Z_B \frac{E + F}{E - F}$$

Para a face (0):

$$Z_B \frac{Ee^{-jk_b l_b} + Fe^{jk_b l_b}}{Ee^{-jk_b l_b} - Fe^{jk_b l_b}} = Z_r. \text{ Desenvolvendo-se esta última equação tem-se:}$$

$$\frac{E}{F} = \frac{-\frac{Z_r}{Z_b} \exp(jk_b l_b) - \exp(jk_b l_b)}{\exp(-jk_b l_b) - \frac{Z_r}{Z_b} \exp(-jk_b l_b)} = k_1$$

Substituindo-se a equação acima na face (I) tem-se:

$$\frac{Ce^{-jk_c l_c} + De^{jk_c l_c} + j \frac{h}{\omega A} I}{Ce^{-jk_c l_c} - De^{jk_c l_c}} = \frac{Z_B}{Z_c} \frac{k_1 + 1}{k_1 - 1} = k_2$$

Desenvolvendo-se esta equação tem-se:

$$C(e^{-jk_c l_c} - k_2 e^{-jk_c l_c}) + D(e^{jk_c l_c} + k_2 e^{jk_c l_c}) + j \frac{h}{\omega A} I = 0$$

Para $l=0$

$$\frac{C}{D} = \frac{(-e^{-jk_c l_c} - k_2 e^{-jk_c l_c})}{(e^{-jk_c l_c} - k_2 e^{-jk_c l_c})} = k_3$$

Substituindo-se agora na face (II) e com $l = 0$, tem-se:

$$\frac{Ae^{-jk_m l_m} + Be^{jk_m l_m}}{Ae^{-jk_m l_m} - Be^{jk_m l_m}} = \frac{Z_c}{Z_m} \frac{1 + k_3}{k_3 - 1} = k_4$$

Desenvolvendo tem-se:

$$\frac{A}{B} = \frac{-k_4 e^{jk_m l_m} - e^{jk_m l_m}}{e^{-jk_m l_m} - k_4 e^{-jk_m l_m}} = k_5$$

Sabendo-se que:

$$P_r = RP_i$$

$$\frac{P_i(1+R)}{P_i(1-R)} = \frac{Z_m B(k_5+1)}{Z_A B(k_5-1)} = k_6 \Rightarrow R = \frac{k_6-1}{1+k_6}$$

Calculado R, pode-se calcular a força e a velocidade da face receptora:

$$P_4 = P_i + P_r \Rightarrow F_4 = AP_i(1+R)$$

$$v_4 = \frac{P_i}{\rho c} - \frac{P_r}{\rho c} = \frac{P_i}{\rho c}(1-R)$$

Equações para a camada de matching:

$$0 = -F_2 + jZ_m A \cot g(k_m l_m) v_2 - jZ_m A \operatorname{cosec}(k_m l_m) v_4$$

$$0 = -F_4 + jZ_m A \operatorname{cosec}(k_m l_m) v_2 - jZ_m A \cot g(k_m l_m) v_4$$

$$v_2 = \frac{F_4 + jZ_m A \cot g(k_m l_m) v_4}{jZ_m A \operatorname{cosec}(k_m l_m)}$$

$$F_2 = jZ_m A \cot g(k_m l_m) v_2 - jZ_m A \operatorname{cosec}(k_m l_m) v_4$$

Equações para a cerâmica (l=0):

$$0 = F_1 + jZ_c A \cot g(k_c l_c) v_1 - jZ_c A \operatorname{cosec}(k_c l_c) v_2$$

$$0 = -F_2 + jZ_c A \operatorname{cosec}(k_c l_c) v_1 - jZ_c A \cot g(k_c l_c) v_2$$

$$V = \frac{je}{\epsilon \rho \omega} (v_2 - v_1)$$

$$v_1 = \frac{F_2 + jZ_c A \cot g(k_c l_c) v_2}{jZ_c A \operatorname{cosec}(k_c l_c)}$$

$$F_1 = -jZ_c A \cot g(k_c l_c) v_1 + jZ_c A \operatorname{cosec}(k_c l_c) v_2$$

Equações para a camada de backing (com dissipação):

$$0 = D_1 v_1 + D_2 v_0 \Rightarrow v_0 = \frac{-D_1 v_1}{D_2}$$

$$0 = F_1 + D_3 v_1 + D_4 v_0$$

$$F_0 = -Z_r A v_0$$

Os coeficientes D_1 , D_2 , D_3 e D_4 são:

$$D_1 = D_4 = A c^E \frac{\vec{\beta}}{\omega} \left(\frac{2}{e^{-j\vec{\beta}l} - e^{j\vec{\beta}l}} \right)$$

$$D_2 = Z_r - A c^E \frac{\vec{\beta}}{\omega} \left(\frac{1 + e^{2j\vec{\beta}l}}{1 - e^{2j\vec{\beta}l}} \right)$$

$$D_3 = -A c^E \frac{\vec{\beta}}{\omega} \left(\frac{1 + e^{2j\vec{\beta}l}}{1 - e^{2j\vec{\beta}l}} \right)$$

6 Estudo Da Viabilidade Do Projeto Com Transdutores Comerciais

No projeto final serão utilizados transdutores esféricos omnidirecionais, que serão selecionados e comprados dentre os existentes no mercado. No estudo abaixo, foi utilizado o catálogo de transdutores esféricos da EDO, que se encontra em anexo.

O método mais simples para a detecção de um sinal recebido é a detecção por limiar. Nesse método, um sinal periódico de tensão recebida é comparado a uma tensão de referência. O sinal é reconhecido quando ele ultrapassa o valor da tensão de referência. Esse método analógico é de implementação simples, porém o erro previsto na detecção do sinal é de 0,5 a 1 comprimento de onda do sinal. Portanto, quanto menor o comprimento de onda do sinal acústico enviado pelo veículo e do sinal enviado pelo alvo, menor será o erro na medida da distância. Quanto maior, porém, a frequência do sinal acústico, maior será a atenuação do mesmo pela água, e menor será a potência elétrica máxima que pode ser fornecida ao transdutor por uma fonte de tensão.

O transdutor com máxima frequência de ressonância do catálogo pesquisado é o modelo OM6829-320. Suas características relevantes são:

- Frequência de ressonância: 320 kHz
- Raio: 5,46 mm
- Voltagem máxima de operação: 40 Vrms
- Ganho na emissão na frequência de ressonância: 142 dB ($\mu\text{Pa/V}$ a 1m de distância)
- Ganho na recepção na frequência de ressonância: -220 dB ($\text{V}/\mu\text{Pa}$)

Como o sinal de 320 kHz deve ser detectado uma vez pelo alvo e uma vez pelo veículo, o erro previsto para a medição da distância percorrida pela onda é de 1 comprimento de onda. Essa distância, no entanto, é o dobro da distância entre o alvo e o veículo. O erro no cálculo dessa distância devido à detecção pode ser previsto como meio comprimento de onda, ou seja, em torno de 2,5 milímetros. Considerando-se as especificações do projeto, esse erro é permissível.

Supondo que dois transdutores 6829-320 estão separados por 50 metros, um deles funcionará como emissor e o outro como receptor, e é aplicada a tensão máxima de operação sobre o emissor, deve-se verificar se o sinal recebido no detector é detectável.

- Voltagem de entrada (amplitude) : 56,57 V

- Pressão acústica a 1 metro: $20 \log \frac{P_0}{V} = 142 \Rightarrow P_0 = 7,12 \cdot 10^8 \mu Pa$

- Coeficiente de atenuação da água do mar para 320 kHz: 0,0082 np/m

- Pressão acústica a 50 metros do emissor:

$$P = P_0 \frac{1}{50} e^{-0,0082 \cdot 50} = 9,45 \cdot 10^6 \mu Pa$$

- Tensão de saída no receptor (amplitude):

$$20 \log \frac{V_{out}}{P} = -220 \Rightarrow V_{out} = 9,45 \cdot 10^{-5} V$$

- Tensão mínima para detecção: 0,5 Vrms = 0,707 Vp

Portanto, a amplificação necessária é de: $20 \log \frac{0,707}{9,45 \cdot 10^{-5}} = 77 dB$

Logo, o sinal acústico enviado pelo emissor operando na voltagem máxima é detectável pelo sistema a 50 m de distância se o circuito amplificador na recepção realizar uma amplificação da ordem de 77 dB. É possível se obter tal amplificação com um circuito analógico, logo, é possível a construção de um sistema emissor-receptor com tais transdutores comerciais.

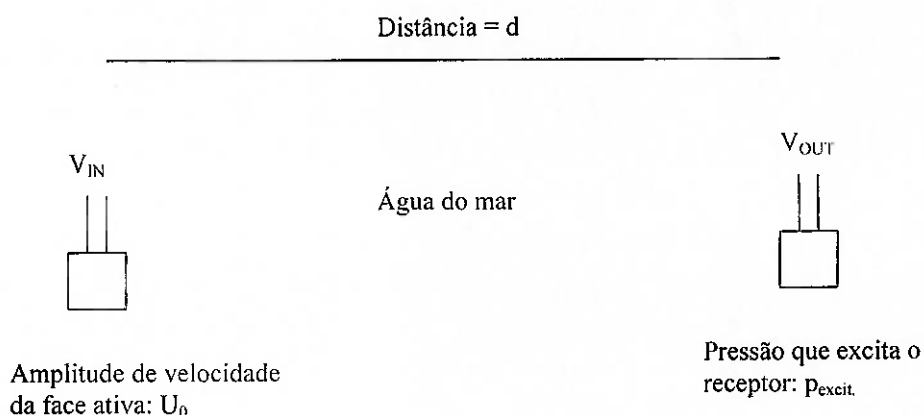
Cabe aqui um comentário acerca das tensões máximas de operação contidas no catálogo. Esses valores são para operação contínua do transdutor. No presente projeto, entretanto, os transdutores vão operar durante alguns ciclos de onda periódica alternados com tempo de espera. Esse modo de operação é chamado modo "burst". Provavelmente, será possível operar os transdutores a uma voltagem maior no modo burst e futuros contatos com o fabricante podem esclarecer esse ponto.

7 Modelagem Matemática Desenvolvida

Com os conhecimentos adquiridos sobre acústica e transdutores pode-se modelar um sistema de emissão-transmissão-recepção de ondas acústicas em ambiente marinho.

O sistema todo consiste de um transdutor-emissor, do meio fluido e de um transdutor-receptor. O emissor é acionado por uma tensão aplicada entre os terminais e produz a onda acústica. O meio fluido é água do mar, que permite a propagação e também atenua a onda acústica. O receptor é excitado pela onda acústica e produz uma tensão. O objetivo principal do modelamento destes 3 elementos é calcular a tensão de saída no receptor em função da tensão de entrada aplicada no emissor e de vários parâmetros.

O esquema abaixo ilustra o que foi dito:



A modelagem do emissor se resume basicamente à resolução do sistema de equações apresentado na seção anterior. Foi feito um programa em Matlab cuja entrada é um vetor composto de vários parâmetros do transdutor (dimensões geométricas, propriedades físicas etc) e da amplitude da tensão de entrada aplicada. A saída deste programa é composta pelas forças aplicadas em cada face das camadas do transdutor, pelas velocidades de cada superfície das camadas e pela corrente elétrica que entra no transdutor. O vetor de saída é composto de números complexos, isto é, que possuem módulo e fase. O módulo representa a amplitude da grandeza em unidades do Sistema Internacional. A fase representa a fase relativa entre essa grandeza e a tensão de entrada, isto é, se a fase da tensão for a referência (fase de $V_{IN} = 0^0$) as fases

de todas as grandezas de saída representarão a fase relativa em relação à tensão. As entradas e as saídas são mostradas no comentário do programa (programa “emissor.m” em anexo).

O modelo do emissor foi feito conforme a descrição feita anteriormente. O programa “recept.m”, que se encontra em anexo, contém o modelo do receptor.

Para a análise dos modelos e o estudo de transdutores, foram feitos programas que analisam a resposta em frequência de várias variáveis, tanto para o emissor como para o receptor. Esses programas são os seguintes: “respxf.m” e “recepfxf.m”.

Foi feito também o modelamento da propagação da onda acústica gerada por um pistão plano e um programa para o cálculo do coeficiente de absorção na água do mar segundo a equação empírica apresentada. Esses programas se encontram em anexo (“abscfdB.m” e “propag.m”)

Os objetivos da construção desses modelos são: Projetar transdutores adequados para o protótipo a ser construído e prever o comportamento desse protótipo para várias condições de uso, auxiliando no projeto do mesmo.

8 Estudo do PIC e Comunicação Serial

Para a execução da primeira parte do projeto, os principais tópicos estudados foram o PIC e a comunicação serial.

8.1 Estudo do PIC

O modelo de microcontrolador estudado e usado foi o PIC16C73B, principalmente por ser esse o microcontrolador disponível aos alunos. Foram estudadas suas características visando a sua programação e uso no projeto.

O estudo feito pode ser dividido em três partes principais: arquitetura interna do microcontrolador, periféricos internos e programação.

O estudo baseou-se nas referências [2], [6] e [12] contidas na bibliografia deste trabalho.

8.1.1 Arquitetura

O esquema da arquitetura interna do PIC é mostrado na figura 3.

O PIC possui algumas características de arquitetura que o tornam rápido e eficiente. Ele possui arquitetura Harvard, ou seja, o programa e os dados são acessados de memórias diferentes e por vias diferentes. Isso permite que ciclos de busca e execução sejam feitos simultaneamente (enquanto uma instrução é executada, é feito o ciclo de busca da próxima instrução). Como as vias de programa e dados são diferentes, as instruções podem ter um número de bits diferente dos 8 bits de dados. As instruções possuem 14 bits de tamanho e podem ser executadas em um ciclo de instrução simples, exceto os casos de saltos de programa.

O PIC possui uma ULA (unidade lógica e aritmética) de 8 bits e um registrador de trabalho (*working register* - W). Permitem a execução de funções aritméticas e lógicas entre os dados no registrador de trabalho e qualquer outro dado na memória de dados. A ULA pode executar operações de soma, subtração e operações lógicas.

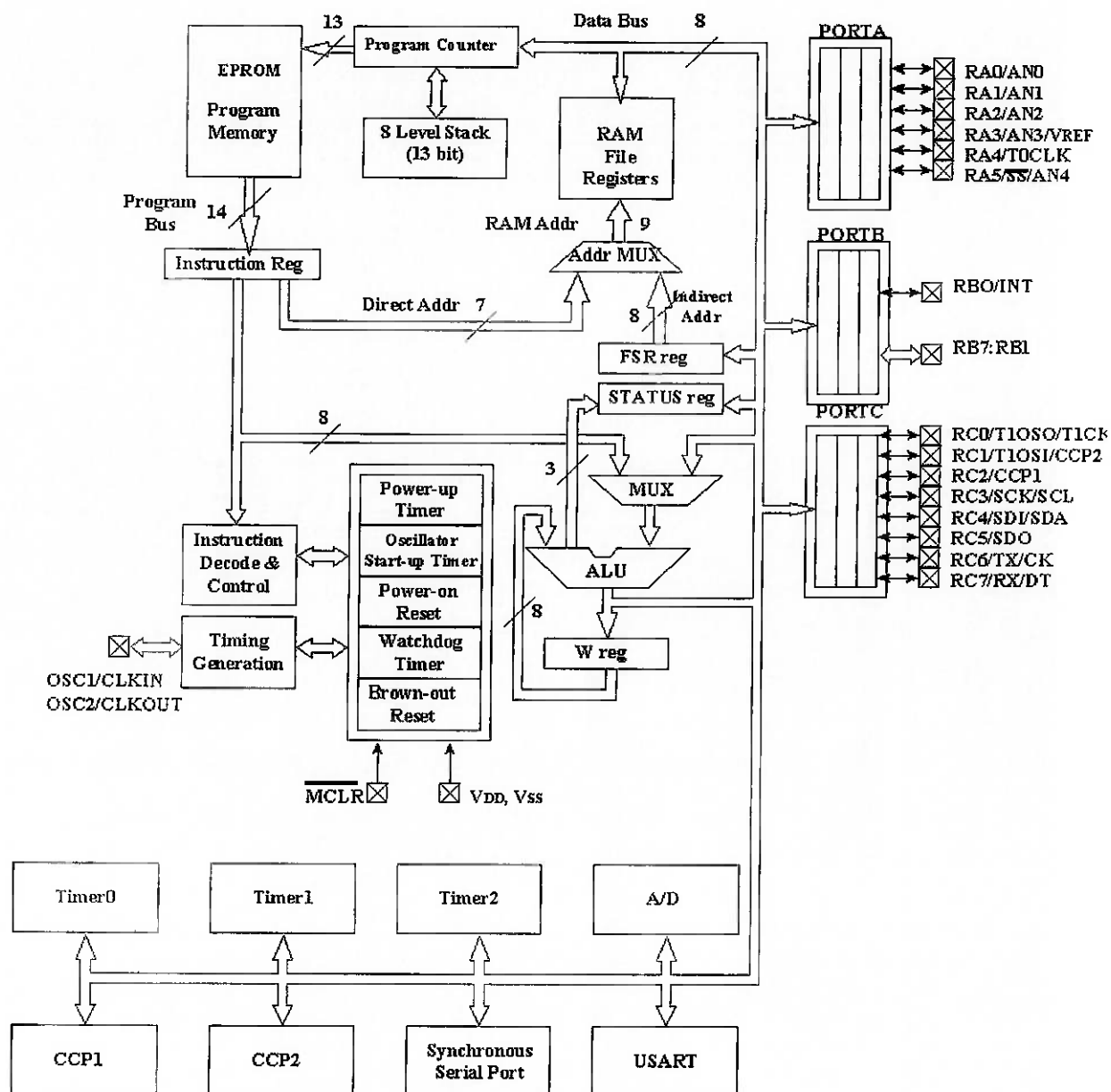


Figura 3 - Arquitetura do microcontrolador PIC16C73B

8.1.2 Periféricos

Os periféricos do PIC16C73B são mostrados na parte inferior da figura

3. As características e usos dos periféricos são os seguintes:

- Timer0: um timer de 8 bits. Pode ser programado para dividir a frequência do clock por 1, 2, 4, 8, 16, 32, 64, 128 e 256. O clock pode ser externo ou interno. É capaz de gerar uma interrupção ou setar um flag quando há estouro da contagem.
- Timer1: um timer/contador de 18 bits. Pode funcionar como contador, com clock externo e como timer, com clock interno. Pode ser

programado para dividir a frequência do clock por 1, 2, 4 e 8. É capaz de gerar uma interrupção ou setar um flag quando há estouro da contagem.

- Timer 2: um timer de 8 bits com período programável. Pode ser programado para dividir a frequência do clock por 1, 4 e 16. É capaz de gerar uma interrupção ou setar um flag quando há um determinado número de estouros (1 a 16).

- CCP1 e CCP2: os módulos CCP podem operar em três modos: modo Capture, modo Compare e modo PWM. No modo Capture, a contagem do Timer1 é capturada quando ocorre um evento no pino CCP1 ou CCP2. No modo Compare, a contagem do Timer1 é comparada a um valor pré-programado e é feita uma sinalização quando a contagem é atingida. No modo PWM (*Pulse Width Modulation* - Modulação de Largura de Pulso), são gerados pulsos num dos pinos CCP. O Período e a largura dos pulsos pode ser programado. O modo PWM é controlado pelo Timer2.

- A/D: um conversor A/D que possui uma frequência de amostragem máxima de aproximadamente 65 kHz.

- SSP: uma porte de comunicação serial síncrona. É útil para comunicações com outros periféricos ou microcontroladores.

- USART: abreviação de Universal Synchronous Assynchronous Receiver Transmitter (Receptor e transmissor síncrono e assíncrono universal). É uma porta serial que pode operar tanto no modo síncrono quanto no modo assíncrono. Pode ser usado para comunicação com PC's ou outros periféricos.

8.1.3 Programação

A programação do PIC é geralmente feita escrevendo-se o programa numa linguagem de programação (Assembler, Linguagem C etc), compilando-se o programa e gravando o programa no microcontrolador. É usado um sistema de desenvolvimento que faz a compilação, simulação e gravação.

Neste trabalho, a linguagem de programação utilizada foi Assembler, por se tratar da linguagem disponível no sistema de desenvolvimento usado.

A memória de programa do PIC16C73B é uma EPROM, ou seja, é uma memória apagável por meio de luz UV. Isso permite o desenvolvimento de programas com segurança e permite a realização de testes.

Para o desenvolvimento dos programas, foram estudadas as instruções do PIC , linguagem Asssembler e o manual de instruções do sistema de desenvolvimento MPLAB – referência [9].

8.2 *Estudo de comunicação serial – PC*

A implementação de comunicação serial do PC via software é basicamente a programação de endereços da memória definidos, que definem as características da comunicação e controlam o trânsito de dados.

Foram usadas as referências [7] e [10] como um guia na programação e utilização das portas seriais do PC.

9 Requisitos e decisões iniciais do projeto

Para que o sistema de localização forneça medidas úteis a um comando do veículo, a implementação deve possuir algumas características. A relação seguinte lista os principais requisitos do sistema:

- **Exatidão da distância:** a distância calculada deve ter uma exatidão da ordem de poucos milímetros (3 a 4 mm). Essa exatidão é limitada principalmente pela detecção por limiar e por pequenos atrasos no processo de localização.

- **Resposta rápida:** a informação da distância entre o veículo e o alvo deve ser utilizada com fins de posicionamento do veículo. Portanto, o atraso entre a ordem de localização e disponibilização da resposta deve ser o mínimo possível. Considerando que a distância a ser medida pode ser de até 50 m, teria-se um atraso mínimo da ordem de 70 ms, que é o tempo aproximado que uma onda sonora leva para percorrer 100 m (ida e volta) na água.

- **Cronometragem precisa:** como ondas sonoras se propagam na água em velocidades da ordem de 1500 m/s, para que haja uma precisão da ordem de milímetros no cálculo da distância, a resolução da cronometragem deve ser inferior a $1\mu\text{s}$ (que corresponde a um avanço da frente de onda de 1,5 mm).

- **Freqüência, largura de pulso e duração do trem programáveis:** O acionamento dos transdutores será feito através de trens de pulsos gerados pelos microcontroladores. Se a freqüência, a largura de pulso e a duração do trem de pulsos forem programáveis, os valores ideais desses parâmetros podem ser determinados experimentalmente. Além disso, poderia ser interessante a variação da largura de pulso e da duração do trem de pulsos de acordo com a variação da distância (para menores distâncias podem-se utilizar larguras de pulso e durações menores)

- **Velocidade do som programável e com boa resolução:** A velocidade do som varia principalmente com o meio, temperatura, salinidade e pressão. Para que possam ser feitos testes e para que possam ser feitas calibrações do sistema durante o seu funcionamento, a unidade móvel deve permitir a reprogramação da velocidade do som gravada. A resolução da

velocidade do som deve ser da ordem de 0,1 mm, para que a precisão na medição da distância não seja prejudicada.

As principais decisões preliminares foram as seguintes:

- **Uso de transdutores piezelétricos:** os transdutores acústicos piezelétricos são muito utilizados na geração e recepção de ondas acústicas, principalmente ultra-sônicas. Foram feitos estudos durante o primeiro semestre a respeito do funcionamento, construção e uso desses transdutores. São de fácil construção e todos os materiais encontram-se à disposição dos alunos.

- **Uso de freqüências diferentes para o sinal de busca e de resposta:** a unidade móvel envia o sinal de busca e espera pelo sinal de resposta enviado pela unidade fixa. O sinal de busca muito provavelmente produzirá um sinal de eco, que deve atingir a unidade móvel antes do sinal de resposta. A unidade móvel, portanto, deve ser capaz de diferenciar o sinal de resposta do eco do seu próprio sinal. Isso pode ser feito se forem usadas freqüências diferentes para os sinais de busca e resposta e se a identificação dos sinais recebidos for feita através de um tratamento dos sinais recebidos.

- **Uso do microcontrolador PIC:** o PIC é conhecido por ser um microcontrolador rápido, pois possui arquitetura de Harvard (veja seção sobre o PIC). Foi escolhido para o projeto principalmente pela sua velocidade e pela sua disponibilidade.

- **Uso de comunicação serial:** o microcontrolador deve possuir uma interface com a unidade de comando, a qual nesse caso será simulada por um PC. A comunicação serial foi escolhida para a implementação dessa interface por ser um tipo padrão de comunicação e porque vários modelos de PIC possuem porta de comunicação serial

- **Uso de linguagem C:** para a implementação da interface de comando do sistema de localização e interface com o usuário, foi decidida a utilização da linguagem C, que é uma linguagem de uso geral, altamente divulgada, utilizada e bem conhecida dos alunos. Parte do estudo para programação em comunicação serial foi baseada na referência [8].

- **Tratamento analógico dos sinais recebidos:** os sinais recebidos, tanto pela unidade fixa como pela unidade móvel, devem receber um tratamento para que o sinal de excitação desejado seja reconhecido

corretamente. Após um estudo inicial sobre a amostragem e o tratamento digital de sinais, decidiu-se que o tratamento analógico, constituído por filtragem, amplificação e detecção, em comparação com o tratamento digital; seria mais rápido (produzindo menos erro na cronometragem do tempo) e de implementação mais simples. A filtragem e a amplificação dos sinais, no entanto, devem ser muito bem feitas de modo a evitar uma detecção errada, o que produziria um resultado espúrio.

10 Materiais e softwares utilizados na implementação do projeto

Para a implementação do projeto, foram usados os seguintes materiais e softwares:

- Matlab 4.0 – versão do estudante
- Compilador Pacific C – versão freeware
- Sistema de desenvolvimento MPLAB 5.0
- Proto-board
- Placa de protótipo para construção do circuito
- Microcontrolador PIC16C73B
- Conversor MAX232
- Cristal TXC 20 MHz
- Comparador LM311
- Power-Mosfet IRF 840
- Núcleo de ferrite RM 6S
- Fios de transformador com bitolas 28 e 32 AWG
- Cerâmicas piezelétricas
- Matching de acrílico
- Conector BNC
- Barras de latão e PVC para construção dos corpos dos transdutores
- Eletrodo de cobre
- Fios de cobre para conexões gerais
- Materiais para solda
- Cabo serial DB-25 - conector de três pinos
- Cabos BNC
- Fonte de Tensão Estabilizada FSCC – 1503D - Dawer
- Osciloscópio digital Kikusui
- Osciloscópio 54112D – Hewlett Packard
- PC 486 100MHz 8 megabytes de RAM
- Gerador de Funções Pantec 3414

- Capacitores eletrolíticos
- Capacitores cerâmicos
- Resistores
- Analisador de impedância 4194A – Hewlett Packard
- Tanque de água

11 Construção e Testes de Transdutores na Água

11.1 A primeira tentativa de construção de um transdutor

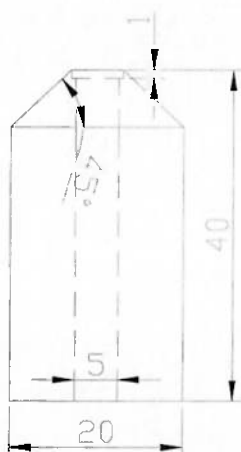
No primeiro semestre foi construído um transdutor que não apresentou resultados satisfatórios, porém os conhecimentos adquiridos foram importantes para a construção do protótipo definitivo no segundo semestre. Por este motivo as etapas de construção deste transdutor, assim como a escolha da frequência de ressonância e as curvas de impedância elétrica do transdutor na água também são apresentados.

As etapas da construção do transdutor foram:

1. Foi definida a diretividade e a frequência de ressonância do transdutor. O ideal é que ele tenha uma baixa diretividade. Isto implica em um pequeno diâmetro, pois: $Dir \cong \frac{1}{4} K^2 D_{transd}^2$, sendo K o número de onda. A frequência de ressonância é ditada pela espessura da cerâmica, pois $c = \frac{2l}{\Delta t} = 2lf \Rightarrow f = \frac{c}{2l}$, sendo l a espessura da cerâmica, c a velocidade da luz na água e f a frequência de ressonância.
2. Foi escolhido um transdutor de 6 mm de diâmetro e 1 MHz de frequência. Deve-se notar que quanto maior a frequência menor o erro de medição, porém a atenuação é maior.

O material da cerâmica foi o PZT5A e a sua espessura de aproximadamente 2,5 mm. O seu diâmetro inicial era de 12 mm, mas por meio de um corte no esmeril ele foi reduzido para os 6 mm desejados.

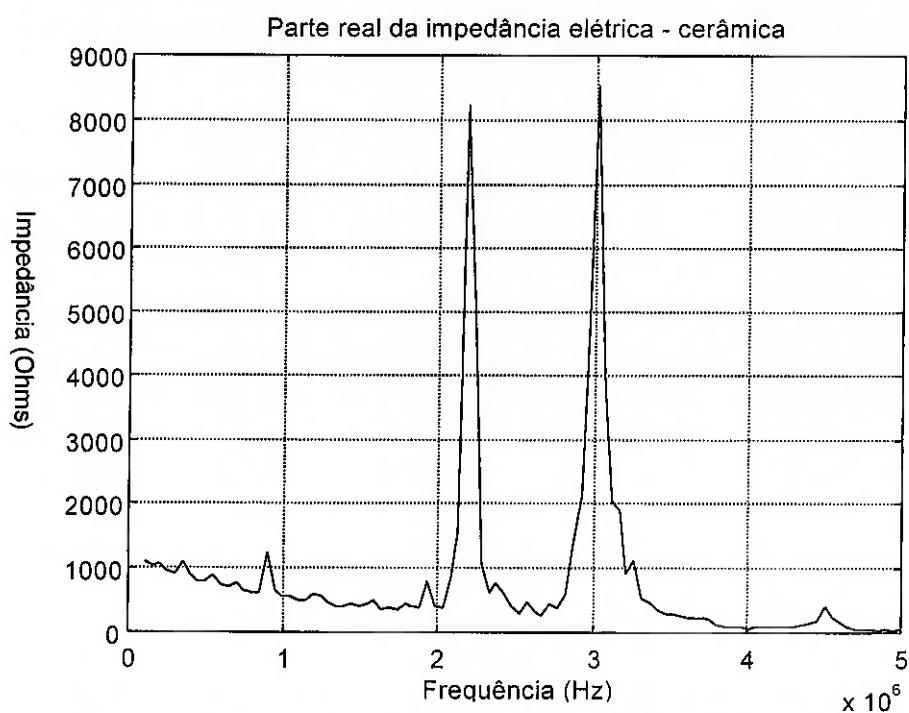
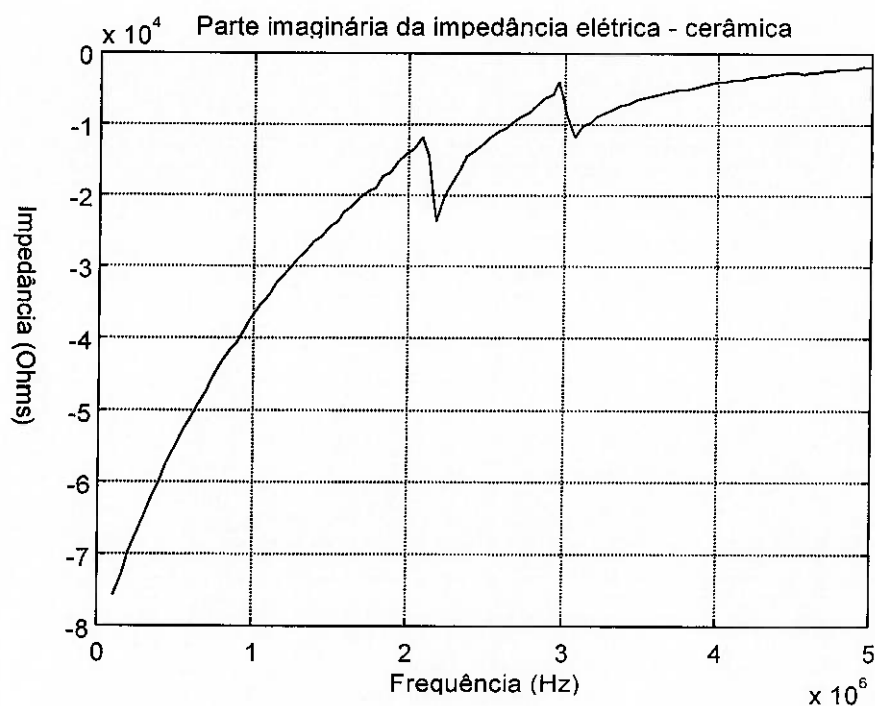
3. A próxima etapa foi a construção do corpo do transdutor, que tem o seguinte formato:



O desenho acima é o de fabricação. O material do corpo é de PVC e adotou-se como tolerância geral 0,1 mm.

4. Fez-se depois uma soldagem do fio e a colagem da chapa na cerâmica. A colagem é feita com uma resina condutiva para que ela não se comporte como isolante
5. Logo depois realizam-se os testes e resultados através do impedômetro e por fim a avaliação final.

Os resultados da análise no impedômetro são os seguintes gráficos de impedância:



Os gráficos obtidos são bem diferentes dos esperados: A cerâmica não apresenta ressonância próxima de 1 MHz. A ressonância e a anti-ressonância são encontradas entre 2 e 3 MHz.

O motivo mais provável destas distorções é que podem ter havido alterações sérias nas propriedades acústicas ou piezelétricas da cerâmica durante a usinagem. Um aquecimento excessivo pode ter ocasionado a

despolarização da cerâmica, ou os choques durante a usinagem podem ter causado o aparecimento de trincas internas na cerâmica.

11.2 Etapas de construção e decisões de projeto do protótipo final do transdutor

1. Considerações iniciais

Por motivo de disponibilidade foi utilizada uma cerâmica de 10 mm de espessura. O material desta cerâmica disponível para o uso dos alunos era o PZT-4, que propicia uma velocidade de aproximadamente 3000 m/s neste material. Desta forma a frequência de ressonância seria de 150 kHz, conforme o cálculo já explicado nas etapas de construção do transdutor do primeiro semestre.

Na verdade, inicialmente tinha-se escolhido um valor de frequência de ressonância de 500 kHz como ideal para que se tenha um baixo erro e ao mesmo tempo uma atenuação não muito grande, mas como a disponibilidade de cerâmicas piezelétricas é limitada resolveu-se utilizar o PZT-4 de 10 mm de espessura. Outro ponto importante aprendido no primeiro semestre é que não se deve tentar usinar a cerâmica, sob o risco dela se despolarizar, o que ocorreu no primeiro semestre. Este fator limitou muito a escolha de valores da frequência de ressonância, já que a partir de uma cerâmica de uma certa espessura esta não poderia ter as suas dimensões alteradas por meio de usinagem.

2. O matching

O material do matching foi o acrílico e ele foi utilizado para aumentar a largura de banda em torno da frequência de ressonância para que dessa forma o transdutor possa atuar numa faixa maior de frequência. Outro ponto importante é que o uso da camada de matching aumenta o coeficiente de transmissão da onda acústica entre a cerâmica e o meio. Foi projetado um matching com espessura de 4,2 mm.

O projeto da espessura do matching é baseado nas seguintes considerações:

- Considerando-se que a velocidade do som no acrílico é no máximo 2700 m/s

- $v = \lambda * f$, $f=150$ kHz
- espessura = $\lambda/4$

3. Etapas de construção

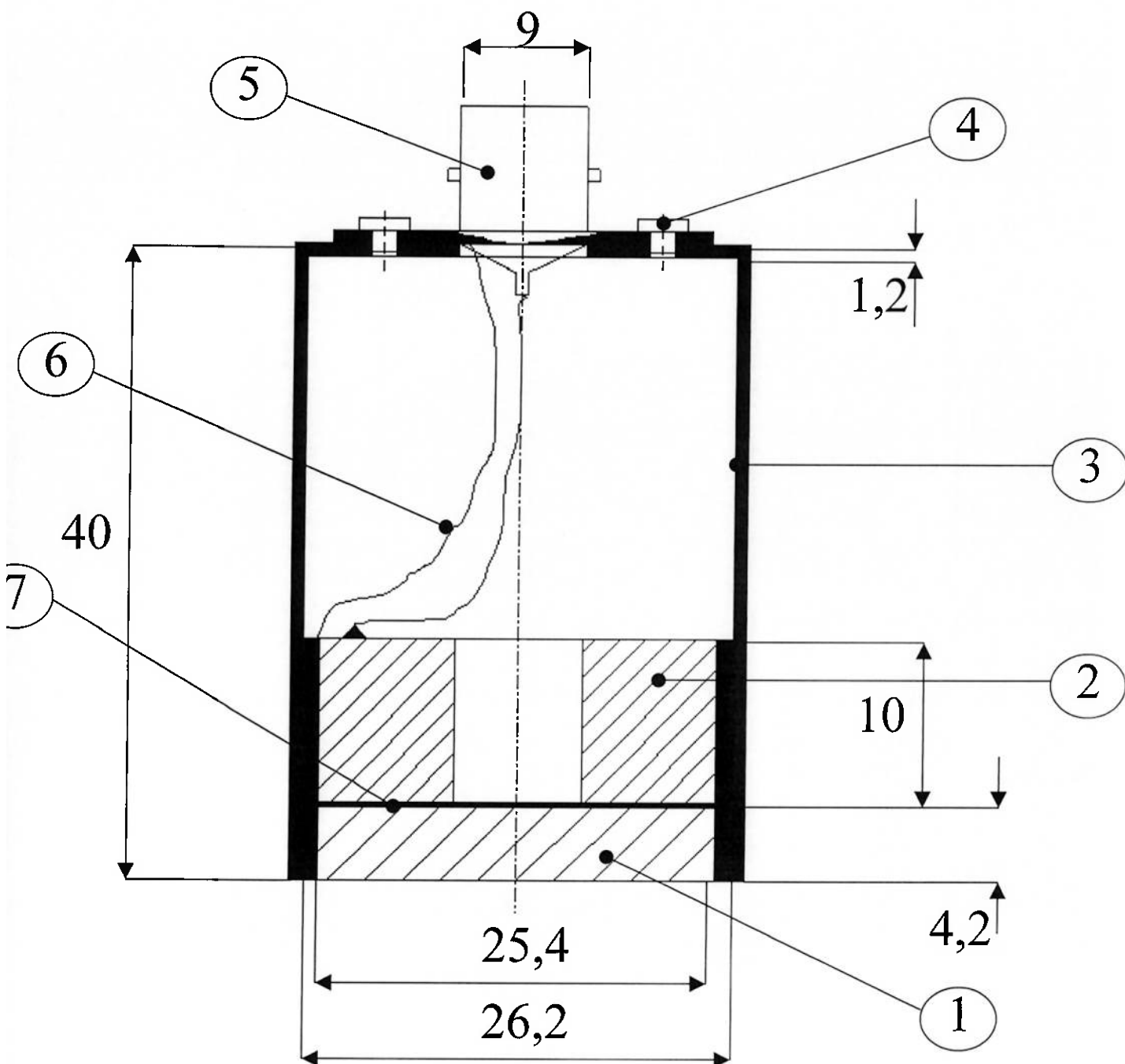
Soldaram-se 2 fios na cerâmica piezelétrica escolhida e foi verificado que a frequência de ressonância era a esperada. Em seguida colou-se o matching na cerâmica. Entre o matching e a cerâmica foi colocado com uma resina condutiva um eletrodo de cobre para que pudesse ser feito o contato elétrico.

A espessura do matching de acrílico foi usinada com 1 ou 2 milímetros a mais do que o necessário e, para que realmente a largura de banda fosse aumentada, foi preciso que se lixasse o acrílico até que se observasse no impedômetro o aumento da largura de banda na curva de impedância em torno de 150 kHz.

O próximo passo seria então a construção do corpo do transdutor, por meio de usinagem feita nos laboratórios de máquinas operatrizes da Escola Politécnica. Um desenho de conjunto a seguir detalha os materiais e dimensões utilizados no projeto do transdutor, como por exemplo o uso de latão no corpo do transdutor pela facilidade de usinagem deste material e o uso do conector BNC para o acionamento do transdutor.

Após a montagem do conjunto o transdutor foi selado com resina épxi para a proteção contra a água no seu interior. As decisões sobre algumas dimensões como espessura da parede do corpo do transdutor ou comprimento do mesmo foram tomadas tendo como base de que o transdutor não poderia ser pesado e deveria comportar a cerâmica, o matching e o conector BNC.

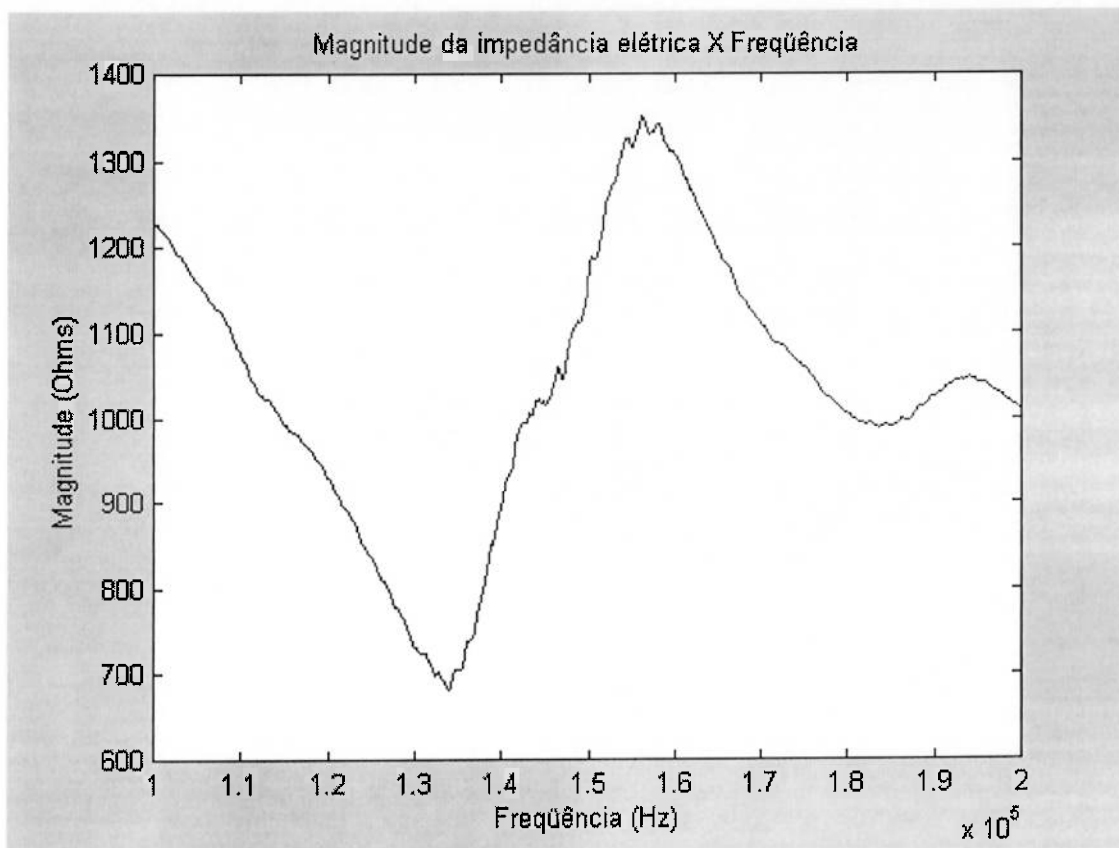
O desenho de conjunto a seguir mostra melhor o que foi escrito acima:

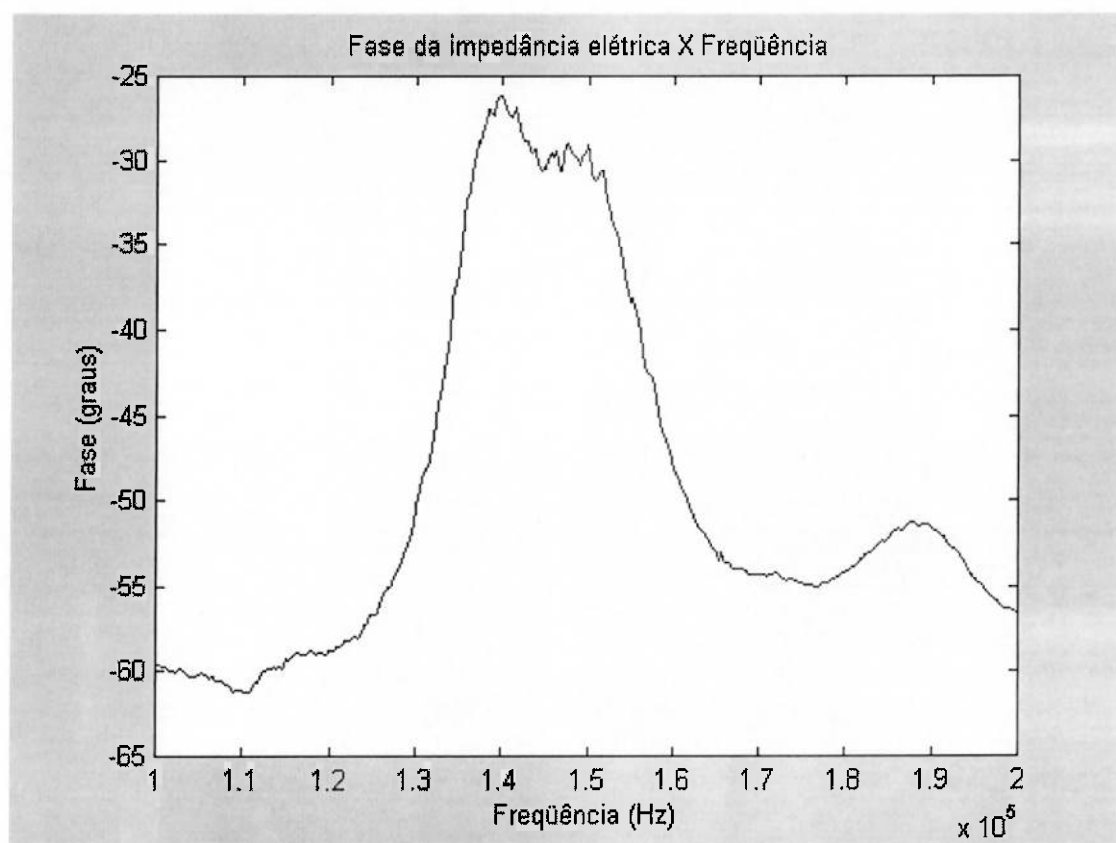


7	Eletrodo de cobre	1	Cobre
6	Fio de cobre	2	Cobre
5	Conector BNC	1	
4	Parafuso tamanho M4	2	
3	Corpo do transdutor	1	Latão
2	Cerâmica piezelétrica	1	Cerâmica PZT4
1	Matching	1	Acrílico
Peca	Denominação	Quantidade	Material
Escala : 1:4	Titulo: Desenho de conjunto de um transdutor		PMC-581

11.3 Curvas de Impedância Elétrica do Transdutor

As curvas de magnitude e fase abaixo mostram que a frequência de ressonância, conforme o esperado, está entre 150 kHz e 160 kHz. Estas curvas foram obtidas por meio de um impedômetro com o transdutor imerso em água (em um tanque). Como a largura de banda foi aumentada com o uso do matching de acrílico, as faixas de frequência entre 130 kHz e 170 kHz são adequadas para o acionamento do transdutor.

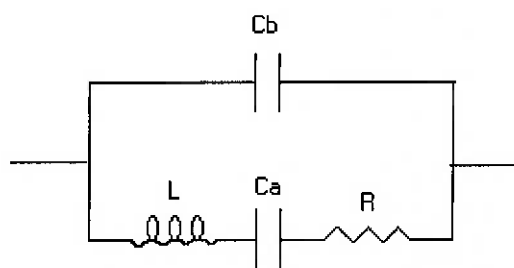




11.4 Circuito Equivalente do Transdutor na água

O circuito equivalente de um transdutor é o circuito elétrico que simula o comportamento real do transdutor no ambiente em que ele atua, no caso a água. Os modelos de circuitos podem ser escolhidos em um impedômetro pelo usuário. Após a escolha do modelo os valores dos resistores, capacitores e indutores são fornecidos pelo impedômetro.

O esquema do circuito equivalente é dado abaixo:



Tendo como valor central 150 kHz e para uma faixa de freqüência de 100 kHz a 200 kHz os valores encontrados foram:

$$R = 1,2 \text{ k}\Omega$$

$$L = 7,8 \text{ mH}$$

$$C_a = 175,641 \text{ pF}$$

$$C_b = 795,552 \text{ pF}$$

Já tendo uma faixa de frequência mais estreita, os valores do modelo fazem com que ele se aproxime mais do transdutor real, no presente caso. Para a faixa de 143 kHz a 163 kHz e tendo como o valor central 153 kHz, tem-se:

$$R = 1,02 \text{ k}\Omega$$

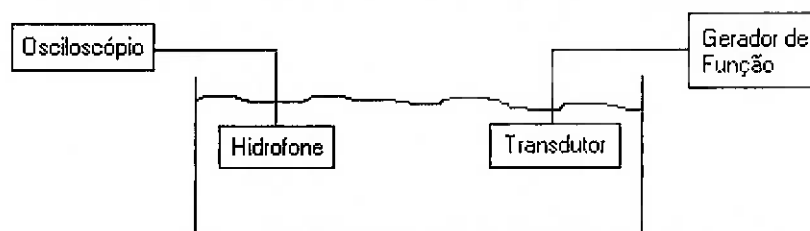
$$L = 8,04 \text{ mH}$$

$$C_a = 169,599 \text{ pF}$$

$$C_b = 808,236 \text{ pF}$$

11.5 Circuito de Teste do Transdutor na Água

Para verificar o funcionamento do transdutor foi montado o circuito de testes mostrado abaixo:



O gerador de função produz um trem de pulsos, o que aciona o transdutor. O hidrofone identifica este sinal e este é lido no osciloscópio. Este simples procedimento mostrou o resultado satisfatório da frequência de ressonância do transdutor e uma razoável largura de banda (aproximadamente de 130 kHz a 170 kHz).

12 Implementação do Software e da comunicação com o PC

12.1 Softwares

12.1.1 Programas dos PIC's

Foram feitos dois programas: unidade móvel e unidade fixa. O programa da unidade móvel é muito mais complexo, pois abrange comunicação, cronometragem do tempo e multiplicação. O programa da unidade fixa está quase completamente contido no programa da unidade móvel. Ambos os programas são explicados a seguir.

Em primeiro lugar, foram escolhidos os periféricos do microcontrolador que serão usados:

- Para a cronometragem do tempo, foi escolhido o Timer1, já que possui 16 bits. A resolução da contagem escolhida foi de 1 ciclo de instrução, que é de 200 ns. Além do Timer1, para se obter uma faixa de medida de tempo adequada, foi decidido utilizar uma variável (registrador da memória) para contar o número de estouros do Timer1. A medida do tempo, portanto, tem um total de 24 bits (3 bytes).
- Para a geração do trem de pulsos, verificou-se que o módulo PWM era ideal para a geração de pulsos com período e largura programáveis. Foi utilizado um período de 8 bits e uma largura de pulso de 8 bits, ambos com resolução de 1 ciclo de instrução. Foi usado o módulo CCP1 operando no modo PWM.
- O Timer0 foi escolhido para a medição da duração do trem de pulsos, ou seja, para indicar durante quantos ciclos o módulo PWM deve gerar os pulsos. A duração do trem pode ser programada e o módulo PWM pode ser desligado por software quando determinada contagem é atingida.
- A contagem do Timer1 deve ser lida imediatamente após a identificação do sinal de resposta, para que sejam evitados erros de cronometragem. Para isso, foi usado o módulo CCP2, que captura a contagem do Timer1 quando o pino CCP2 é excitado. O sinal de identificação da resposta deve, portanto, ser ligado ao pino CCP2.

- Como foi decidido pela comunicação serial assíncrona entre o PIC e o PC, o periférico USART foi usado para realizar essa comunicação.

Foi decidido usar 3 bytes para armazenar a velocidade do som. Pode ser programada uma velocidade de até 1670 m/s com precisão de 0,1 mm/s.

Depois de medido o tempo, deve ser feita uma multiplicação entre o tempo medido e a velocidade do som programada. Como o PIC não possui instrução de multiplicação, essa operação deve ser implementada por software, foi utilizada uma macro que realiza multiplicação de 8 bits por 8 bits. Como a multiplicação é de 24 bits por 24 bits, os números são decompostos em bytes, as operações são feitas e a resposta de 48 bits é composta.

Foram implementadas macros para a comunicação serial: uma macro para receber dado do PC e armazenar na memória, uma macro para enviar um byte para o PC e uma macro para enviar o conteúdo de um endereço para o PC.

O programa foi desenvolvido com o auxílio do sistema de desenvolvimento MPLAB. Com o MPLAB é possível realizar simulações discretas do microcontrolador, o que é muito útil durante a etapa de desenvolvimento. Durante a programação foi feita uma constante consulta ao datasheet do PIC, que contém as instruções de programação de todos os módulos do microcontrolador.

O programa inicialmente faz a inicialização das variáveis e dos módulos periféricos utilizados. Em seguida, é feita uma inicialização da comunicação, na qual o PIC espera um determinado dado vindo do PC e responde um outro determinado dado. Isso é feito para que o programa em si só continue se a comunicação com o PC estiver funcionando corretamente. Em seguida é esperado um comando vindo do PC. Quando um comando chega, ele é reconhecido e é executada a sub-rotina correspondente (Veja seção *Lista de Comandos*).

Quanto ao programa da unidade fixa, o que ele faz é responder a um sinal de entrada, gerando um trem de pulsos com período, largura de pulso e duração previamente programados. Propositadamente, é gerado um atraso entre a recepção do sinal e a geração da resposta. Esse artifício é utilizado para que a unidade fixa não reconheça duas entradas durante o mesmo sinal de busca e para que a resposta não seja tão rápida a ponto de atingir a

unidade móvel antes que ela esteja preparada para recebê-la. Esse atraso é corrigido pela unidade móvel antes do cálculo da distância.

Os programas completos das unidades móvel e fixa se encontram nos anexos.

12.1.2 Programa do PC

O programa do PC é um programa simples. A função dele é receber comandos do usuário, verificar sua validade, enviar os comandos ao PIC e receber respostas. Está preparado para enviar/receber dados das portas COM1, COM2, COM3 e COM4.

O programa do PC e da unidade móvel foram desenvolvidos em conjunto e ambos só funcionam se estiverem rodando ao mesmo tempo e se a comunicação serial estiver ligada corretamente.

A listagem do código fonte do programa (escrito em linguagem C) encontra-se em um dos anexos.

12.1.3 Lista de Comandos

Os comandos reconhecidos pelo PIC são todos enviados pelo PC. A seguir, encontra-se uma lista dos comandos disponíveis ao usuário:

- **Realizar localização:** o PC envia um comando ao PIC. O comando é reconhecido e o microcontrolador gera o trem de pulsos para acionamento do transdutor. A cronometragem do tempo começa e a unidade móvel se prepara para receber o sinal de resposta. Se a resposta da unidade fixa é recebida, a distância é calculada e enviada ao PC, que imprime a resposta na tela. Se a resposta não chega

- **Modo de teste:** o PC é mantido em estado de espera por mensagens vindas do PIC. O PIC, por sua vez, deve ser excitado com sinais de entrada no pino CCP2, o que, segundo implementado na versão de testes do programa da unidade móvel, deve enviar respostas ao PC proporcionais ao intervalo de tempo entre duas bordas de subida consecutivas.

- **Reinicializar comunicação:** se por algum motivo o PIC é resetado durante a execução do programa da unidade de comando ou o programa da unidade de comando é reiniciado quando o programa do PIC já está

rodando, esse comando permite o reestabelecimento da comunicação entre o PIC e o PC.

- **Verificar Parâmetros programados:** esse comando é enviado ao PIC, que envia os valores programados na sua memória de velocidade, largura de pulso, período dos pulsos e duração do trem de pulsos.

- **Programar velocidade:** esse comando permite que um novo valor de velocidade do som seja programado no PIC. Quando o PIC é inicializado, o valor default de 1500m/s é armazenado na memória. Esse valor pode ser modificado através desse comando.

- **Programar largura de pulso:** esse comando permite que um novo valor de largura de pulso seja programado no PIC.

- **Programar período dos pulsos:** esse comando permite que um novo valor de período seja programado no PIC.

- **Programar duração do trem de pulsos:** esse comando permite que um novo valor de duração do trem de pulsos seja programado no PIC.

- **Ajuda:** esse comando exibe uma lista dos comandos e sua sintaxe.

- **Finalizar programa:** termina a execução do programa da unidade de comando. O programa da unidade móvel, no entanto, permanece rodando.

12.1.4 Regras de comunicação serial

Para a comunicação serial entre o PIC e o PC, foram estabelecidas algumas regras para que as mensagens possam ser entendidas e tratadas corretamente:

- **Dados enviados pelo PIC pelo PC:** os dados podem ser de dois tipos: strings e números. Os Strings devem ser precedidos pelo código 0x50 (código ASCII da letra 'P') e devem ser finalizados com o código 0x00 (código ASCII para 'NULL'). Os números devem ser precedidos pelo código 0x4E (código ASCII da letra 'N') e pelo número de bytes do número a ser enviado. Essas regras foram criadas para que o PC possa receber mensagens genéricas do PIC e para que mensagens inválidas possam ser identificadas.

- **Dados enviados pelo PC ao PIC:** os dados enviados pelo PC ao PIC são simplesmente seqüências de bytes. Para cada comando um número bem definido de bytes é enviado. O PIC se encarrega de reconhecer os comandos e interpretar os dados.

12.2 Circuito de testes

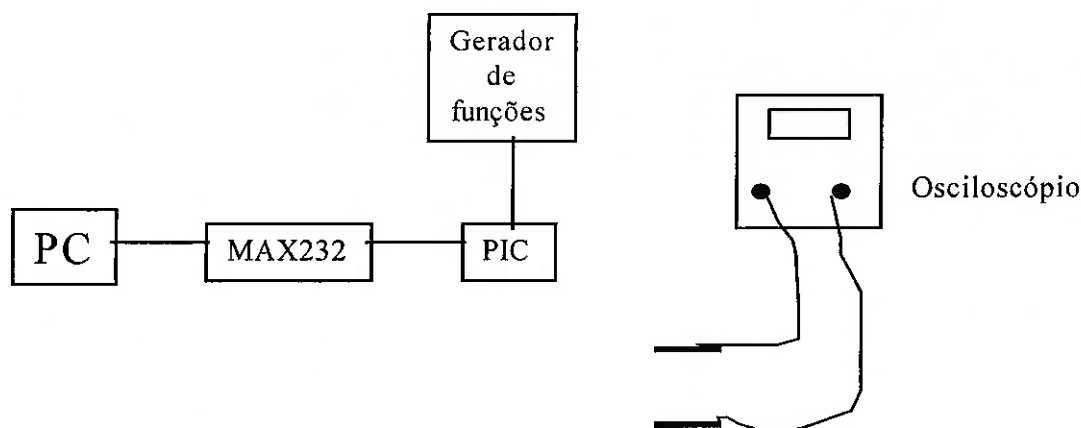
Foi construído um circuito de testes numa proto-board para que as funções do sistema pudessem ser testadas. Para a comunicação serial, foi necessário o uso de um conversor, pois as voltagens padrão de comunicação são diferentes para o PIC e para o PC (o PC usa o padrão RS-232 e o PIC usa tensões TTL). Foi usado o conversor MAX232.

O cabo serial utilizado foi de 25 pinos e foi ligado na porta COM1 do PC.

O circuito foi montado com todos os componentes necessários, segundo os datasheets do PIC e do MAX232. O esquema do circuito de testes montado se encontra em anexo.

Um gerador de funções com largura de pulso regulável foi utilizado para a geração de sinais de teste, simulando medições de distância.

A seguir, encontra-se um esquema dos equipamentos utilizados:



13 Testes e Discussões dos softwares das Unidades de Comando, Móvel e Fixa

Foram realizados testes de comunicação entre o PC e o PIC. Os programas das unidades de comando e móvel foram testados em conjunto e a comunicação foi implementada com sucesso. Os comandos podem ser enviados ao PIC, o qual reconhece-os, executa as tarefas correspondentes e envia uma resposta. Os comandos de programação da velocidade, largura de pulso, período e duração do trem puderam ser verificados graças à função de verificação implementada.

Para o teste da função de localização, foi feita uma versão de testes do programa da unidade móvel, a qual permite que a ordem de localização e o sinal de resposta sejam reconhecidos no mesmo pino (CCP2). Desse modo, se um clock de período conhecido for ligado à esse pino, haverá a simulação de uma medida de distância, sendo que a localização é iniciada numa borda de subida do clock de entrada e o sinal de resposta é reconhecido na próxima borda de subida.

Para essa função era esperada a obtenção de respostas (distâncias) iguais para um sinal de clock com pequena largura de pulso ($\sim 10 \mu s$) na entrada de testes. Foram obtidas, no entanto, respostas variadas.

A simulação por software dessa função no MPLAB, com uma entrada de clock simulada, no entanto, mostrou as respostas esperadas.

A comunicação serial assíncrona foi implementada com sucesso e possui uma velocidade compatível com o sistema de localização, que necessita de respostas rápidas.

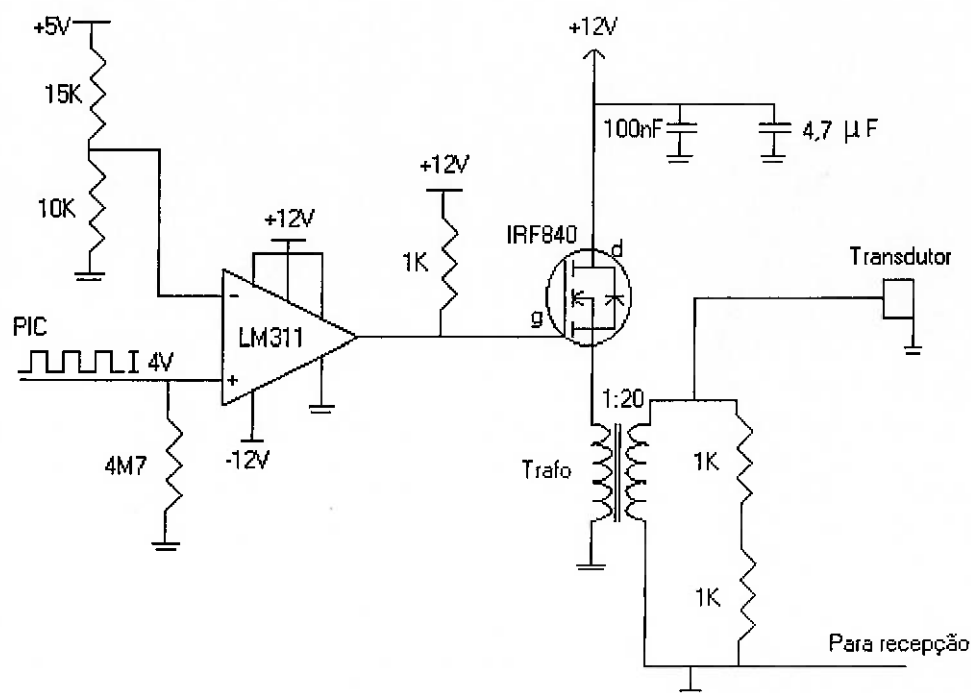
Todas as funções puderam ser testadas e se mostraram funcionais, exceto a função de localização, que apresentou respostas bem diferentes quando um trem de pulsos de período constante foi introduzido na entrada de testes. Era esperado que vários resultados iguais fossem apresentados. Testes posteriores mostraram que esses valores diferentes provavelmente vêm de excitações indesejadas que são produzidas durante a introdução dos pulsos. Nas simulações do MPLAB, entretanto, a função de localização cronometrou o

tempo e calculou as respostas da maneira correta. Para essa função, portanto, são necessários testes melhores para comprovar a sua funcionalidade prática.

Portanto, foi projetada e implementada a parte de software do sistema de localização. O que foi implementado mostrou-se em sua maioria funcional e pronto para a o acionamento e controle do sistema de localização completo.

14 Estudo dos Circuitos Eletrônicos de Emissão

O circuito de emissão montado neste semestre é o esquematizado abaixo. Serão a seguir detalhadas as funções de cada componente no circuito de emissão.



14.1 Breve Explicação do Circuito de Emissão

O PIC envia uma onda quadrada (trem de pulsos) de amplitude no mínimo de 4V. O comparador LM311 gera na saída uma voltagem de 12V. Com este valor de tensão o Power-Mosfet IRF840 chaveará o primário do transformador, que por sua vez transformará a tensão pico a pico num valor considerável para o acionamento do transdutor.

14.2 O Comparador LM311

A entrada positiva do comparador é de 4V. A entrada negativa do comparador tem o valor da tensão de referência, que no caso é de 2V. O valor

da tensão de referência é de 2V porque há uma queda de tensão de 3V no resistor de 15K Ω (basta observar que: $5V - \frac{5V}{10K\Omega + 15K\Omega} * 15K\Omega = 2V$).

O comparador funciona da seguinte forma: se o valor da tensão de entrada positiva for maior que o da tensão de referência (no caso 4V > 2V) a saída torna-se, no presente caso, +12V. Quando a entrada positiva é zero a saída é zero.

Isso acontece porque o comparador tem saída open-collector, no qual deve-se colocar uma alimentação na saída (neste caso de 12V).

Porém, a principal questão que se faz sobre o uso de um comparador neste circuito é: Por que não usar somente um Power-Mosfet para chavear o trem de pulsos enviado pelo PIC? A resposta é simples: o Power-Mosfet considerado neste projeto não consegue chavear com uma entrada de tensão de apenas 4V, ou seja, ele precisa de um valor maior de tensão para executar sua função.

14.3 O Power-Mosfet IRF840

O Power-Mosfet chaveia o primário do transformador em 12V e 0V. Uma das questões que se faz sobre o uso deste tipo de chave é: Por que não usar um transistor NPN por exemplo? Isto se deve porque a entrada no gate do Power-Mosfet é uma tensão e não corrente elétrica. Isto facilita muito o projeto do circuito já que dessa forma não é necessário calcular a corrente de base, a saturação etc.

Outro ponto que se levanta sobre o uso do Power-Mosfet é: Se o comparador já, de certa forma, faz um chaveamento, por que usar então outra chave? O fato é que o comparador não fornece tanta corrente na saída e, sem o Power-Mosfet, o comparador “queimaria”.

14.4 Circuito para estabilizar a fonte

Os capacitores de 4,7 μ F e 100 nF são usados para estabilizar a fonte em 12 V, já que a fonte possui uma resistência interna e pode ocorrer uma queda momentânea de tensão quando altas correntes são exigidas.

14.5 O Transformador

Foi usada uma relação de 1:20 neste transformador para que a tensão pico a pico no secundário fosse de pelo menos em torno de 200V (valor razoável para o acionamento do transdutor). Para o projeto do transformador foi usado como referência um projeto base - referência [1]. Foi usado um núcleo de ferrite modelo RM-6S da fabricante Thornton, o fio do primário de bitola 28 AWG e do secundário de 32 AWG.

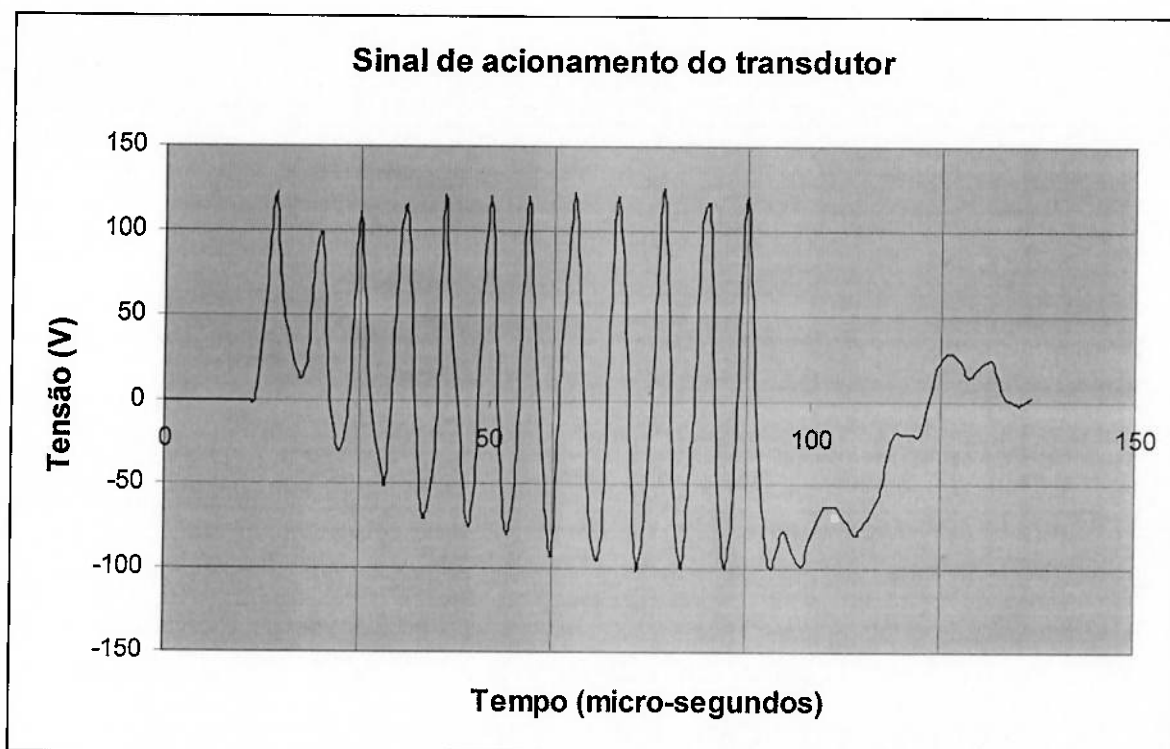
Os resistores de 1 k Ω colocados em paralelo com o transformador são usados para eliminar oscilações na frequência natural que aparecem no sinal de acionamento do transdutor.

14.6 Teste do Circuito de Emissão

Foi testado o transdutor em conjunto com o circuito eletrônico de emissão (feito numa placa protótipo). Este teste foi feito com pulso-eco, ou seja, o transdutor emitia um sinal que refletia na parede de um tanque e voltava para o transdutor (logicamente o transdutor deveria estar posicionado de forma normal à parede do tanque). Foram acrescentadas funções de teste aos programas com a finalidade de gerar trens de pulso com uma determinada taxa de repetição.

Foi enviado, por exemplo, um sinal pelo PIC com frequência de 130 KHz e observou-se que a frequência do eco era realmente a mesma de 130 KHz, porém também foi observada uma frequência em torno de 150 KHz.

Foi também analisado o formato do sinal de acionamento do transdutor. Um exemplo de sinal de acionamento é mostrado no gráfico abaixo, para frequência de aproximadamente 150 kHz, comprimento de 12 pulsos (aproximadamente 80 μ s) e ciclo de trabalho de 50 %:



Pode-se observar que o sinal apresenta um transiente nos seus instantes iniciais e apresenta um comportamento oscilatório aproximadamente senoidal no restante do sinal.

À medida que se aumenta o comprimento do trem de pulsos, o comportamento transiente no início do sinal permanece o mesmo, ou seja, tem o mesmo tempo de duração. Porém, a parte do sinal no qual o comportamento é aproximadamente senoidal terá um tempo maior de duração.

Um ponto a ser observado no teste do circuito de emissão é que se o transdutor for substituído por um resistor como carga o comportamento transiente no início do sinal desaparece.

15 Considerações Finais

No primeiro semestre foram feitos estudos teóricos, estudos de projetos anteriores e estudos de viabilidade do projeto.

Essa primeira parte do projeto teve muitas contribuições, principalmente relacionadas ao entendimento e discussão dos conceitos físicos e do conhecimento técnico relacionado ao projeto do sistema de localização. Foi também desenvolvido um modelamento matemático completo da geração e recepção de ondas acústicas com transdutores cilíndricos, bem como da propagação da onda acústica em meio fluido. Esses modelos podem ser usados no projeto de transdutores e na previsão de alguns comportamentos do protótipo.

No segundo semestre foram feitos o projeto, testes e construção de transdutores piezelétricos, foram elaborados os softwares do microcontrolador, da unidade fixa e da unidade móvel, foram feitos o circuito de emissão e os testes deste circuito com o transdutor.

Os testes do circuito de emissão com o transdutor tiveram pleno êxito, mostrando que o transdutor construído, o circuito de emissão e os softwares do microcontrolador, da unidade fixa e da unidade móvel podem ser usados para o intuito no qual foram propostos.

Não foram realizados na prática o circuito de recepção e em apenas uma das duas 2 unidades foi feito o protótipo, embora elas sejam muito semelhantes e tenham o mesmo projeto. Provavelmente, no início deste trabalho a sua dificuldade não foi bem avaliada, o que fez que ele não se completasse em toda a sua extensão inicialmente proposta.

16 Bibliografia

1. Furukawa, Celso Massatoshi – ***Navegação de um veículo autônomo por ultra-som em ambiente estruturado*** – Dissertação de Mestrado – São Paulo, 1992
2. James, M. R. - ***Microcontroller Cookbook*** – Newnes, 1997 - Contém informações básicas a respeito do PIC.
3. Kino, Gordon S. – ***Acoustic waves: Devices, imaging and analog signal processing*** - Prentice-Hall – 1987
4. Kinsler, Lawrence E. [et al] - ***Fundamentals of Acoustics*** - 3rd ed. - New York - Wiley - 1982
5. Shikata, F.M. - ***Transdutor piezelétrico de ultra-som*** - Trabalho de Formatura - 1997
6. ***Datasheet da família pic16cXX*** – fornecido pela Microchip Technology - Contém detalhes do uso do PIC e de sua programação.
7. ***Revista Saber Eletrônica*** – Ano 33, nº303 - Editora Saber - Informações sobre comunicação serial.
8. ***Manual de instruções do compilador Pacific C*** – download do site da internet www.htsoft.com
9. ***Manual de instruções do MPLAB*** – fornecido pela Microchip Technology
10. ***www.beyondlogic.org*** - Contém várias informações sobre porta serial, porta paralela e outras interfaces usando PC.

11. www.htsoft.com - Fornece uma versão freeware de um compilador C, entre outros downloads.

12. www.microchip.com - Fabricante do PIC. Contém muitos datasheets, application notes, softwares e informações sobre os microcontroladores e sistemas de desenvolvimento.

17 Anexos

17.1 Convenção de sinais para o modelamento dos transdutores

1 – Deslocamento da partícula: o deslocamento é positivo se é feito no sentido positivo do eixo x .

2 – Velocidade da partícula: velocidade positiva se o deslocamento é feito no sentido positivo de x .

3 – Aceleração da partícula: é positiva se a velocidade da partícula é positiva e aumenta com o tempo ou se a velocidade da partícula é negativa e diminui com o tempo.

4 – Força: é positiva aquela que provoca uma aceleração positiva, ou seja, uma variação positiva de velocidade. Força positiva, portanto, é uma força no sentido positivo de x e força negativa é no sentido negativo de x .

5 – Campo elétrico: campo elétrico positivo é aquele que é orientado no sentido positivo do eixo x (é orientado de um local carregado positivamente para um local carregado negativamente)

6 – Deformação: deformação é a razão entre a variação do tamanho e o tamanho inicial. Portanto, deformação positiva significa aumento de tamanho e deformação negativa diminuição de tamanho.

7- Tensão mecânica: tensão é a força por unidade de área que age sobre um elemento. A tensão é positiva quando causa uma deformação positiva, isto é, quando a tensão é de tração. Explicando em termos do eixo x , a tensão é positiva quando a força sobre o elemento é positiva na face da direita e negativa na face da esquerda.

8 – Deslocamento elétrico: deslocamento elétrico de um dielétrico pode ser definido como a carga acumulada por unidade de área no capacitor. É orientado da placa carregada positivamente para a placa carregada negativamente.

9 – Voltagem: A voltagem é a diferença de potencial elétrico entre dois pontos. Para definir o sinal em termos do sentido do eixo x e do campo elétrico, pode-se convencionar que a voltagem é positiva quando o potencial elétrico da

direita é menor que o da esquerda (sendo que o eixo x é orientado da direita para a esquerda).

10 – Corrente elétrica: pode-se convencionar o sinal da corrente elétrica que flui por um capacitor. Corrente elétrica positiva é aquela que flui no sentido de aumentar o deslocamento elétrico no sentido positivo ou aumentar o campo elétrico no sentido positivo.

17.2 Programas em MATLAB

Programa: 'emissor.m'

```
function Y=emissor(entrada)
% 'emissor(entrada)' calcula o modelo básico de um transdutor
% piezoelétrico funcionando como emissor que é composto de três camadas:
% cerâmica piezoelétrica, camada de backing e camada de matching.
% Este modelo inclui dissipação na camada de backing e não inclui
% no cálculo a presença de um indutor em paralelo.
%
% 'entrada' é um vetor que contém todos os parâmetros necessários
% para o modelamento do transdutor e o valor da tensão de entrada.
% Essa função calcula as tensões e as velocidades harmônicas de cada
% face das camadas do transdutor e também a corrente elétrica de entrada.
% Esse modelo é válido para grandezas harmônicas.
% Como saída, é criado um vetor composto da seguinte forma:
%
% x(1)= força aplicada na cerâmica pelo backing
% x(2)= veloc. da face traseira da cerâmica
% x(3)= força aplicada na cerâmica pelo matching
% x(4)= veloc. da face dianteira da cerâmica
% x(5)= veloc. da face traseira do backing
% x(6)= veloc. da face dianteira do matching
% x(7)= corrente elétrica de entrada
% x(8)= força aplicada pelo meio à face do backing
% x(9)= força aplicada pelo meio à face do matching
%
% O vetor de entrada deve ser da seguinte forma:
% entrada(1) = tensão de entrada em volts           - Vin
% entrada(2) = raio do transdutor em metros         - a
% entrada(3) = densidade da cerâmica piezoelétrica em kg/m^2 - Pc
% entrada(4) = densidade do material de backing em kg/m^2   - Pb
% entrada(5) = densidade do material de matching em kg/m^2  - Pm
% entrada(6) = velocidade de prop. do som na cerâmica em m/s - cc
% entrada(7) = velocidade de prop. do som no backing em m/s - cb
% entrada(8) = velocidade de prop. do som no matching em m/s - cm
% entrada(9) = comprimento da cerâmica em metros        - Lc
% entrada(10)= comprimento da camada de backing em metros - Lb
% entrada(11)= comprimento da camada de matching em metros - Lm
% entrada(12)= constante piezoelétrica da cerâmica em C/m^2 - ec
% entrada(13)= constante dielétrica relativa da cerâmica   - epsrel
% entrada(14)= frequência da onda acústica em Hz          - freq
% entrada(15)= dens. do meio em contato com o backing em kg/m^3 - P0
% entrada(16)= veloc. som no meio em contato com o backing em m/s - c0
% entrada(17)= dens. do meio em contato com o matching em kg/m^3 - P4
% entrada(18)= veloc. som no meio em contato com o matching em m/s - c4
% entrada(19)= coeficiente de absorção do backing em nepers/m - alpha
%
% Se entrada(11)=0 (comprimento da camada de matching nulo), o programa
% interpretará esse dado como ausência da camada de matching.
% Se entrada(10)=0 (comprimento da camada de backing nulo), o programa
% interpretará esse dado como ausência da camada de backing.

% Leitura dos dados
Vin = entrada(1);
a = entrada(2);
Pc = entrada(3);
Pb = entrada(4);
Pm = entrada(5);
cc = entrada(6);
cb = entrada(7);
cm = entrada(8);
Lc = entrada(9);
Lb = entrada(10);
Lm = entrada(11);
e = entrada(12);
```

```

eprel = entrada(13);
freq = entrada(14);
P0 = entrada(15);
c0 = entrada(16);
P4 = entrada(17);
c4 = entrada(18);
alpha = entrada(19);

A=pi*a^2; % área
w=2*pi*freq; % frequência angular
eps=eprel*8.85e-12; % constante dielétrica absoluta
k4=w/c4; % número de onda no meio dianteiro
k0=w/c0; % número de onda no meio traseiro
cdb=cb^2*Pb; % constante elástica do backing

% Cálculo das impedâncias de radiação
Zr0=A*P0*c0;
Zr4=A*P4*c4;

% Cálculo do número de onda complexo no material de backing
alpha=alpha*freq/5e6; % encontra alpha para uma frequência dif. de 5MHz
% baseado no pressuposto que o coef. de atenuação
% é diretamente prop. à frequência.
kb=w/cb-alpha*j;

% Montagem da matriz

D=zeros(7);

% equações para a camada de backing (incluem atenuação)
if Lb==0 % Se o backing não existe
    D(1,2)=1;
    D(1,5)=-1;
    D(2,1)=1;
    D(2,5)=Zr0;
end
if Lb~=0 % Se o backing existe
    D(1,2)=A*cdb*kb/w^2/(exp(-j*kb*Lb)-exp(j*kb*Lb));
    D(1,5)=Zr0-A*cdb*kb/w*(1+exp(j*2*kb*Lb))/(1-exp(j*2*kb*Lb));
    D(2,1)=1;
    D(2,2)=-A*cdb*kb/w*(1+exp(j*2*kb*Lb))/(1-exp(j*2*kb*Lb));
    D(2,5)=A*cdb*kb/w^2/(exp(-j*kb*Lb)-exp(j*kb*Lb));
end
% equações para a camada de matching
if Lm==0 % Se o matching não existe
    D(3,4)=1;
    D(3,6)=-1;
    D(4,3)=1;
    D(4,6)=Zr4;
end
if Lm~=0 % Se o matching existe
    D(3,3)=-1;
    D(3,4)=j*Pm*cm*A*cot(w/cm*Lm);
    D(3,6)=-j*Pm*cm*A*csc(w/cm*Lm);
    D(4,4)=j*Pm*cm*A*csc(w/cm*Lm);
    D(4,6)=Zr4-j*Pm*cm*A*cot(w/cm*Lm);
end
% equações para a cerâmica piezelétrica
D(5,1)=1;
D(5,2)=j*Pc*cc*A*cot(w/cc*Lc);
D(5,4)=-j*Pc*cc*A*csc(w/cc*Lc);
D(5,7)=j*e/(eps*w);
D(6,2)=j*Pc*cc*A*csc(w/cc*Lc);
D(6,3)=-1;
D(6,4)=-j*Pc*cc*A*cot(w/cc*Lc);
D(6,7)=j*e/(eps*w);
D(7,2)=-j*e/(eps*w);
D(7,4)=j*e/(eps*w);

```

```

D(7,7)=-j*Lc/(w*eps*A);

B2=[0;0;0;0;0;0;Vin];

x=D\B2;

x(8)=-Zr0*x(5); % Força externa aplicada sobre a camada de backing
x(9)=-Zr4*x(6); % Força externa aplicada sobre a camada de matching

Y=x;

```

Programa: 'respfx.m'

```

function respfx(en,R,L,fi,ff)
% 'respfx(en,R,L,fi,ff)' plota os gráficos da eficiência,
% da impedância, da potência elétrica, da potência acústica e
% a amplitude de corrente elétrica de um circuito formado por
% um transdutor calculado pelo programa 'emissors.m', um indutor
% em paralelo e uma resistência interna da fonte de tensão.
% 'en' é o vetor de entrada do programa 'emissor.m'.
% 'R' é a resistência interna da fonte.
% 'L' é a indutância do indutor.
% 'fi' é a frequência inicial dos gráficos e 'ff' é
% a frequência final.
% Se L=0, o programa interpretará esse dado como a
% ausência de indutor

cont=1;
div=100; % número de divisões
freq=[fi:(ff-fi)/div:ff];
while cont<=length(freq)
    en(14)=freq(cont);
    Y=emissor(en); % resposta do transdutor (volt. qquer)
    Z=en(1)/Y(7); % impedância do transdutor
    if L~=0
        % impedância total com indutor
        Zt(cont)=R+(Z*j*2*pi*en(14)*L)/(Z+j*L*2*pi*en(14));
    else
        Zt(cont)=R+Z; % impedância total sem indutor
    end
    Iin=en(1)/Zt(cont); % corrente de entrada
    V=en(1); % tensão de entrada do circuito
    Vin=en(1)-R*Iin; % tensão no transdutor
    en(1)=Vin;
    X=emissor(en); % resposta do transdutor
    % eficiência
    efic(cont)=abs(real(X(9)*conj(X(6)))/real(V*conj(Iin)));
    corr(cont)=abs(X(7)); % amplitude da corrente elétrica
    pot(cont)=real(V*conj(Iin)); % potência elétrica de entrada
    % potência acústica
    potacust(cont)=abs(real(X(9)*conj(X(6))));
    cont=cont+1;
end

plot(freq,efic*100,'b')
title('Eficiência do circuito transdutor+indutor+fonte')
xlabel('Frequência (Hz)')
ylabel('Eficiência (%)')
cinvert
grid
figure(2)
plot(freq,corr,'b')
title('Amplitude da corrente elétrica de entrada')
xlabel('Frequência (Hz)')
ylabel('Corrente (A)')
cinvert
grid
figure(3)
plot(freq,pot,'b')

```

```

title('Potência elétrica de entrada')
xlabel('Frequência (Hz)')
ylabel('Potência (W)')
cinvert
grid
figure(4)
plot(freq,real(Zt),'b',freq,imag(Zt),'w--')
title('Impedância total do circuito (partes real e imaginária)')
xlabel('Frequência (Hz)')
ylabel('Impedância (ohms)')
cinvert
grid
figure(5)
plot(freq,potacust,'b')
title('Potência acústica emitida')
xlabel('Frequência (Hz)')
ylabel('Potência (W)')
cinvert
grid

```

Programa: 'recept.m'

```

function Y=recept(entrada)
% 'recept(entrada)' calcula o modelo básico de um transdutor
% piezoelétrico funcionando como receptor em meio fluido e que é
% composto de três camadas: cerâmica piezoelétrica, camada de
% backing e camada de matching.
% Este modelo inclui dissipação na camada de backing e não inclui
% no cálculo a presença de um indutor em paralelo.
%
% 'entrada' é um vetor que contém todos os parâmetros necessários
% para o modelamento do transdutor e o valor da pressão acústica no
% ponto no qual se encontra o transdutor.
% Essa função calcula as tensões e as velocidades harmônicas de cada
% face das camadas do transdutor e também a tensão elétrica de saída.
% A tensão de saída calculada é a tensão em circuito aberto (corrente
% nula). Esse modelo é válido para grandezas harmônicas.
% Como saída, é criado um vetor composto da seguinte forma:
%
% x(1)= força aplicada na cerâmica pelo backing
% x(2)= veloc. face traseira da cerâmica
% x(3)= força aplicada na cerâmica pelo matching
% x(4)= veloc. da face dianteira da cerâmica
% x(5)= veloc. da face traseira do backing
% x(6)= veloc. da face dianteira do matching
% x(7)= corrente elétrica de saída (nesse caso, é nula)
% x(8)= força aplicada pelo meio à face do backing
% x(9)= força aplicada pelo meio à face do matching
% x(10)=tensão de saída para circuito aberto
%
% O vetor de entrada deve ser da seguinte forma:
% entrada(1) = pressão acústica onde se encontra o transd. em Pa - Pinc
% entrada(2) = raio do transdutor em metros - a
% entrada(3) = densidade da cerâmica piezoelétrica em kg/m^2 - Pc
% entrada(4) = densidade do material de backing em kg/m^2 - Pb
% entrada(5) = densidade do material de matching em kg/m^2 - Pm
% entrada(6) = velocidade de prop. do som na cerâmica em m/s - cc
% entrada(7) = velocidade de prop. do som no backing em m/s - cb
% entrada(8) = velocidade de prop. do som no matching em m/s - cm
% entrada(9) = comprimento da cerâmica em metros - Lc
% entrada(10)= comprimento da camada de backing em metros - Lb
% entrada(11)= comprimento da camada de matching em metros - Lm
% entrada(12)= constante piezoelétrica da cerâmica em C/m^2 - ec
% entrada(13)= constante dielétrica relativa da cerâmica - epsrel
% entrada(14)= frequência da onda acústica em Hz - freq
% entrada(15)= dens. do meio em contato com o backing em kg/m^3 - P0
% entrada(16)= veloc. som no meio em contato com o backing em m/s - c0
% entrada(17)= dens. do meio em contato com o matching em kg/m^3 - P4
% entrada(18)= veloc. som no meio em contato com o matching em m/s- c4

```



```

% entrada(19)= coef. de absorção do backing em nepers/m (a 5MHz) - alpha
%
% Se entrada(11)=0 (comprimento da camada de matching nulo), o programa
% interpretará esse dado como ausência da camada de matching.
% Se entrada(10)=0 (comprimento da camada de backing nulo), o programa
% interpretará esse dado como ausência da camada de backing.

% Leitura dos dados
Pinc = entrada(1);
a     = entrada(2);
Pc    = entrada(3);
Pb    = entrada(4);
Pm    = entrada(5);
cc    = entrada(6);
cb    = entrada(7);
cm    = entrada(8);
Lc    = entrada(9);
Lb    = entrada(10);
Lm    = entrada(11);
e     = entrada(12);
eprel= entrada(13);
freq  = entrada(14);
P0    = entrada(15);
c0    = entrada(16);
P4    = entrada(17);
c4    = entrada(18);
alpha= entrada(19);

A=pi*a^2;           % área
w=2*pi*freq;        % frequência angular
eps=eprel*8.85e-12; % constante dielétrica absoluta
k0=w/c0;            % número de onda no meio traseiro
k4=w/c4;            % número de onda no meio dianteiro
km=w/cm;            % número de onda no matching
kc=w/cc;            % número de onda na cerâmica

% O valor de 'alpha' é para 5MHz. Assumindo que esse coef. é diretamente
% proporcional à frequência, o cálculo é o seguinte:
alpha=alpha*freq/5e6;

kb=w/cb-alpha*j;    % número de onda complexo no backing
z4=P4*c4;           % impedância caract. do meio dianteiro
if Lm~=0
    zm=Pm*cm;        % impedância caract. do matching
else
    zm=z4;           % se Lm=0, o matching não existe
end

zc=Pc*cc;           % impedância caract. da cerâmica
zr=P0*c0;           % impedância caract. do meio traseiro
if Lb~=0
    zb=Pb*cb;        % impedância caract. do backing
else
    zb=zr;           % se Lb=0, o backing não existe
end

cdb=cb^2*Pb;        % constante elástica do backing

% Cálculo do coeficiente de reflexão
if Lb~=0
    K1=exp(2*j*kb*Lb)*(-zr/zb-1)/(-zr/zb+1);
else
    K1=-1e30;        % para eliminar a divisão por zero
end
K2=zb/zc*(K1+1)/(K1-1);
K3=exp(2*j*kc*Lc)*(-K2-1)/(-K2+1);
K4=zc/zm*(1+K3)/(K3-1);
K5=exp(2*j*km*Lm)*(-K4-1)/(-K4+1);

```

```

K6=zm/z4*(1+K5)/(K5-1);

R=(K6-1)/(K6+1); % coeficiente de reflexão

% Pressão acústica e velocidade na face receptora
F4=Pinc*(1+R)*A; % força aplicada na face receptora
v4=Pinc/z4*(1-R); % velocidade da face receptora
x(6)=v4;
x(9)=F4;

% Resolução das equações do matching
if Lm~=0
    v2=(F4+j*zm*A*cot(km*Lm)*v4)/(j*zm*A*csc(km*Lm));
    F2=j*zm*A*cot(km*Lm)*v2-j*zm*A*csc(km*Lm)*v4;
else
    v2=v4;
    F2=F4;
end
x(4)=v2;
x(3)=F2;

% Resolução das equações da cerâmica
v1=(F2+j*zc*A*cot(kc*Lc)*v2)/(j*zc*A*csc(kc*Lc));
F1=-j*zc*A*cot(kc*Lc)*v1+j*zc*A*csc(kc*Lc)*v2;
V=j*e/(eps*w)*(v2-v1);

x(2)=v1; % velocidade da face traseira
x(1)=F1; % força aplicada à face traseira
x(10)=V; % tensão de saída
x(7)=0; % corrente

% Resolução das equações do backing
if Lb~=0
    D1=A*cdb*kb/w*2/(exp(-j*kb*Lb)-exp(j*kb*Lb));
    D2=zr*A-A*cdb*kb/w*(exp(j*2*kb*Lb)+1)/(1-exp(j*2*kb*Lb));
    v0=-D1*v1/D2;
    F0=-zr*A*v0;
else
    v0=v1;
    F0=F1;
end
x(5)=v0;
x(8)=F0;

% Resposta final
Y=x;

```

Programa: 'recepxf.m'

```

function recepxf(en,fi,ff)
% 'recepxf(en,fi,ff)' plota os gráficos em função da frequência
% da potência acústica recebida, da potência acústica absorvida,
% da potência refletida, da voltagem de saída, da potência
% elétrica de saída e do ganho de um transdutor-receptor calculado
% pelo programa 'recept.m'.
% 'en' é o vetor de entrada do programa 'recept.m'.
% 'fi' é a frequência inicial dos gráficos e 'ff' é
% a frequência final.

cont=1;
div=100; % número de divisões
freq=[fi:(ff-fi)/div:ff];
while cont<=length(freq)
    en(14)=freq(cont);
    Y=recept(en); % resposta do transdutor
    Prefl=Y(9)/(en(2)^2*pi)-en(1); % pressão refletida
    potrefl(cont)=abs(Prefl)^2/(en(17)*en(18))*pi*en(2)^2;
    potel(cont)=abs(Y(10)*Y(7)); % potência elétrica de saída
    potacust(cont)=abs(real(Y(9)*conj(Y(6)))); % potência acústica absorvida
    cont=cont+1;
end

```

```

Vout(cont)=abs(Y(10));
potrec(cont)=en(1)^2/(en(17)*en(18))*pi*en(2)^2;% potência recebida
ganhodB(cont)=20*log10(Vout(cont)/(en(1)*1e6)); % ganho em dB
cont=cont+1;
end

% Gráficos
figure(1)
plot(freq,Vout,'b')
title('Voltagem de saída')
xlabel('Frequência (Hz)')
ylabel('Voltagem (V)')
cinvert
grid
figure(2)
plot(freq,potel,'b')
title('Potência elétrica de saída')
xlabel('Frequência (Hz)')
ylabel('Potência (W)')
cinvert
grid
figure(3)
plot(freq,potrec,'b')
axis([fi ff 0 2*potrec(1)])
title('Potência acústica recebida')
xlabel('Frequência (Hz)')
ylabel('Potência (W)')
cinvert
grid
figure(4)
plot(freq,potacust,'b')
title('Potência acústica absorvida')
xlabel('Frequência (Hz)')
ylabel('Potência (W)')
cinvert
grid
figure(5)
plot(freq,potrefl,'b')
title('Potência acústica refletida')
xlabel('Frequência (Hz)')
ylabel('Potência (W)')
cinvert
grid
figure(6)
semilogx(freq,ganhodB,'b')
title('Resposta em frequência do receptor')
xlabel('Frequência (Hz)')
ylabel('Ganho para circuito aberto em dB (volts/micropascal)')
cinvert
grid

```

Programa: 'propag.m'

```

function P=propag(rho,c,a,U0,theta,d,f,P0,T)
% 'P=propag(rho,c,a,U0,theta,d,f,P0,T)' calcula a amplitude
% da pressão acústica de uma onda gerada por um pistão
% plano a um ponto a distância 'd' (em metros) e a um
% ângulo 'theta' (em graus) do eixo do pistão.
% O cálculo é válido para o campo distante.
% As entradas são:
% 'rho'- densidade da água do mar em kg/m^3
% 'c' - velocidade de propagação do som na água do mar em m/s
% 'a' - raio do pistão em metros
% 'U0'- amplitude de veloc. de vibração do pistão em m/s
% 'theta'- ângulo com o eixo do pistão em graus
% 'd' - distância do emissor em metros
% 'f' - frequência da onda em Hz
% 'P0'- pressão em atm
% 'T' - temperatura em graus Celcius

```

```

alpha=abscfdB(f,T,P0)/8.686; % coeficiente de absorção em np/m
k=2*pi*f/c; % número de onda
theta=theta/360*2*pi; % transformação para radianos
% Cálculo da pressão acústica em módulo
% Os valores absolutos de k_vetorial e c_vetorial foram aproximados para
% os valores de 'k' e 'c'.
pressao=rho*c*k*U0*a^2/d*exp(-alpha*d)*abs(J1(k*a*sin(theta),0.01));
P=pressao;

```

Programa: 'abscfdB.m'

```

function Y=abscfdB(f,T,P0)
% 'Y=abscfdB(f,T,P0)' calcula o coeficiente de absorção para a
% água do mar em dB/m. 'f' é a frequência da onda, 'T' é a
% temperatura em graus Celcius e 'P0' é a pressão local em
% atmosferas.

f1=1.32e3*(T+273)*exp(-1700/(T+273));
f2=1.55e7*(T+273)*exp(-3052/(T+273));

A=8.95e-8*(1+2.3e-2*T-5.1e-4*T^2);
B=4.88e-7*(1+1.3e-2*T)*(1-0.9e-3*P0);
C=4.76e-13*(1-4e-2*T+5.9e-4*T^2)*(1-3.8e-4*P0);

Y=A*f1*f^2/(f1^2+f^2)+B*f2*f^2/(f2^2+f^2)+C*f^2;

```

Programa: 'J1.m'

```

function Y=J1(z, intervalo)
% 'Y=J1(z, intervalo)', calcula uma aproximação para J1(z)/z.
% 'intervalo' é o intervalo usado na integração.

fi=0; % argumento da função
n=1; % contador
while fi<=pi
    func(n)=cos(z*cos(fi))*(sin(fi))^2;
    n=n+1;
    fi=fi+intervalo;
end

fi=[0:intervalo:pi];
Integral=trapz(fi,func); % calcula o valor da integral

Y=Integral/pi; % valor de J1(z)/z

```

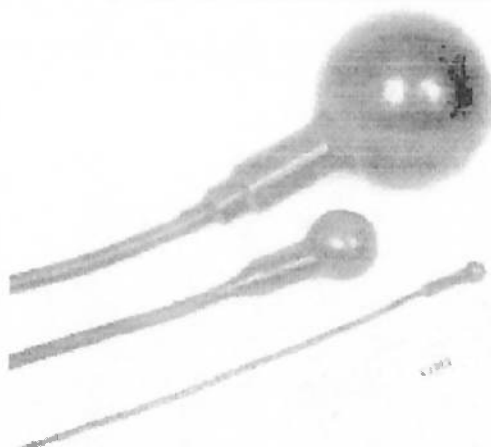
17.3 Catálogo de transductores esféricos

SPHERICAL TRANSDUCERS

Omnidirectional
Beam Transducer

Model 6829 Series
OM 6829

EDO



The Model 6829 series of reciprocal transducers offer the system designer a choice of devices providing an omnidirectional beam pattern over a broad range of operating frequencies. These highly efficient transducers utilize spherical lead - titanate-zirconate ceramic elements which allow a high power output over broad frequency ranges. The active element of each transducer is encapsulated with a stem mount to minimize the adverse acoustic effects (caused by the cable).

Each model is available in a variety of encapsulants, which include polyurethane, neoprene, epoxy, etc. For added flexibility, the spherical ceramic elements may be manufactured with customized electrode and mechanical configurations. Each transducer is also available with various cable types and lengths plus a low noise integral pre-amplifier and transmit/receive switch.



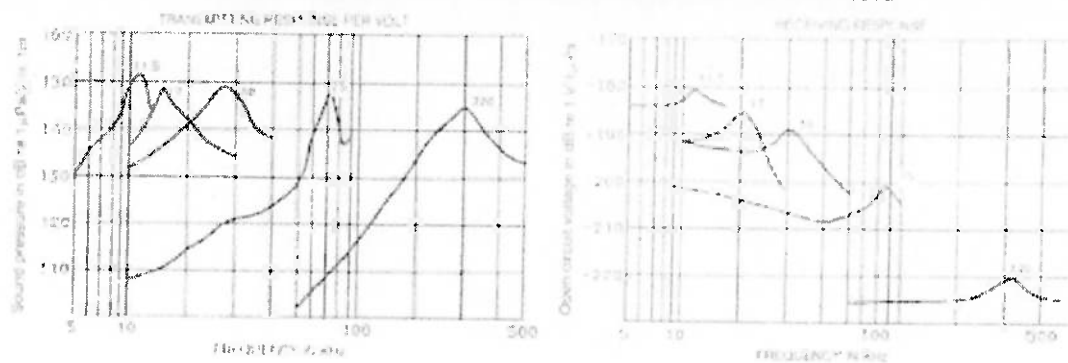
Table 1. Mechanical Characteristics

Model	Part No.	Dim "A" (inches)	Dim "B" (inches)	Dim "C" (inches)	Cable Type	Max. Operating Depth (meters)
6829-11.0	49337	1.0	0.5	1.0	205-2	1500 (487)
6829-17	46475	0.5	0.55	0.7	205-2	2000 (639)
6829-24	46074	0.75	0.57	1.0	205-3	1200 (365)
6829-27	45366	0.93	0.64	0.5	410-1	900 (275)
6829-30	45188	2.7	0.45	0.2	205-1	4000 (1220)
6829-75	46558	1.14	2.01	0.2	585-125	2700 (820)
6829-129	27550	0.75	1.04	1	585-129-1	2500 (762)
6829-17.1	11537	0.53	1.38	3	205-1-304	2500 (762)
6829-320	54550	0.14	1.243	3	205-1-304	2500 (762)

Table 2. Typical Performance Characteristics

Model	Maximum Input Voltage V _{IN}	70 Hz Sensitivity dB re 1 μ V @ 1 Hz	Pressure Response dB re 1 μ V @ 1 Hz	Capacitance pF	Impedance at Resonance	Open Circuit Voltage	Max. Output Power Ratio approx.
6829-15.5	15.5	150	-184	21,614	11.4 \pm 2%	140V	4,000
6829-17	17	144	-187	24,060	9.5 \pm 4%	170V	4,000
6829-21	21	170	-195	23,543	12.5 \pm 3%	180V	2,000
6829-27	27	140	-191	50,220	11.5 \pm 3%	190V	1,000
6829-30	30	160	-179	38,000	9.5 \pm 4%	221V	1,000
6829-75	75	140	-201	9,129	10.5 \pm 3%	230V	300
6829-129	129	144	-207	3,408	21.6 \pm 4%	25V	25
6829-170	170	135	-215	1,400	25.6 \pm 3%	60V	10
6829-320	320	112	-212	1,000	31.4 \pm 4%	40V	3

Typical Response Curves for Model 6829 Series Transducers



Performance Characteristics

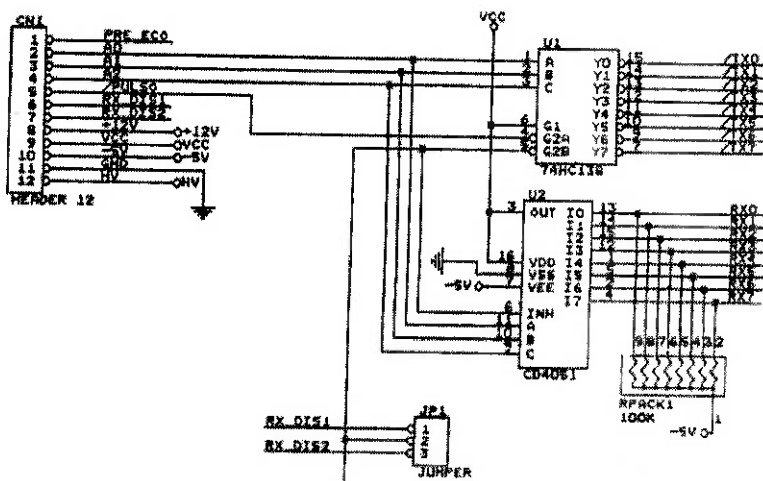
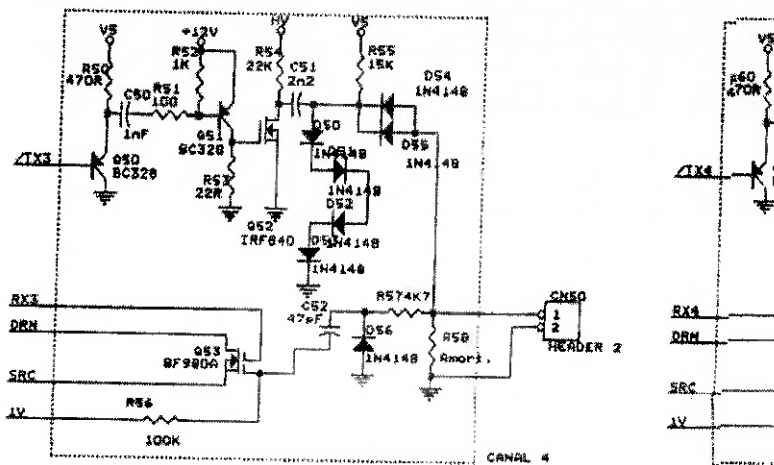
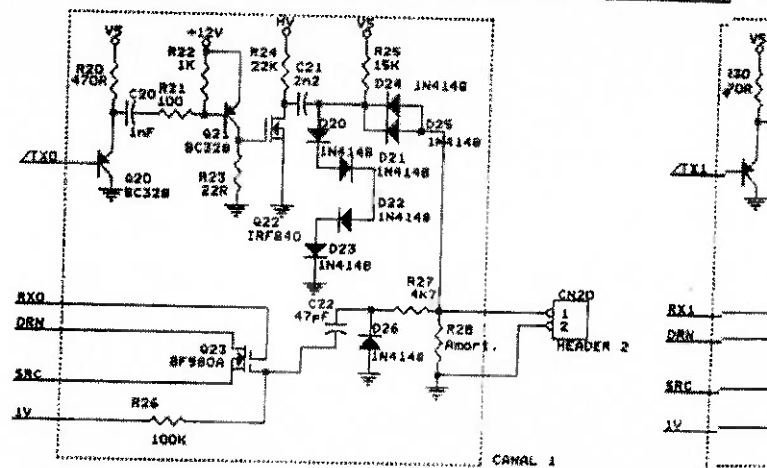
Active material	Navy Type 1
Capacitance	(See Table 2)
Impedance	(See Table 2)
Efficiency	50% Typical
Maximum Input Voltage	(See Table 2)
Maximum Operating Depth	(See Table 1)
Beam Patterns	
Horizontal	Omnidirectional within \pm 2 dB
Vertical (Cable Plane)	Omnidirectional within \pm 3 dB
Cable Type	DSS-2 Typical (Other encapsulants available upon request)
Acoustic Characteristics	(See Table 2)
Duty Cycle	10% Typical

Specifications subject to change
without notice.
Revised February '98

EDO Electro-Ceramic
Products

2645 South 300 West, Salt Lake City, Utah 84115
Telephone (801) 486-7481 • Fax (801) 484-3301
E-mail: sales@edoceramic.com • www.edocorp.com

17.4 Exemplo de Circuito estudado





17.5 Código fonte da Unidade de comando

/* Programa de interface com o PIC - Unidade de Comando (PC)

* Data : Dezembro de 2000

* Compilador : Pacific C

* Programador: Frederico Vines Faria de Lima

* A unidade de comando envia comandos ao PIC e espera por
* confirmações ou respostas.

* ☐ verificada a validade de cada comando. São enviados ao PIC
* somente comandos válidos.

* A sintaxe de cada comando , a seguinte:

* (letra do comando)<argumento - se necessário> (sem espaço em branco)

* Para o comando G (ordem para localização), a resposta esperada

* , a distância em milímetros entre a unidade móvel e a fixa.

* Para o comando S (verificar parâmetros), a resposta esperada

* uma lista de parâmetros numa ordem pre-estabelecida.

* O comando I (inicialização) realiza a reinicialização da comunicação em caso de
* reset do PIC.

* O comando T (modo de teste), aciona o modo de teste, isto é, gera

* trens de pulso intermitentes com período de repetição de aprox. 3ms.

* Para os demais comandos, , esperada a confirmação 'OK'.

* O comando , enviado para o PIC como uma palavra de até 32 bits,

* que consiste de comando (8 bits - código ASCII) e argumento (até 24 bits).

* Quando a maioria dos comandos é executada, uma resposta do PIC é

* esperada. Se a resposta não chega num certo intervalo de tempo,

* o programa julga que ocorreu um erro.

*/

#include<stdio.h>

#include<string.h>

#include<dos.h>

#include<conio.h>

/* DEFINICOES */

#define ESPERA 2600000/* No. limite de iterações de espera */

/* VARIÁVEIS GLOBAIS */

int PORT; /* Endereço da porta serial escolhida */

/* ROTINAS */

void InicializaPorta(void)

/* Prepara a porta serial para comunicação */

{

outp(PORT + 3 , 0x00); /* Limpa DLAB - Acesso ao IER */

outp(PORT + 1 , 0x00); /* Desabilita interrupções */

outp(PORT + 3 , 0x80); /* Seta DLAB - Acesso ao Divisor Latch */

outp(PORT + 0 , 0x0C); /* Set Baud rate - Divisor Latch Low Byte */

/* 0x03 = 38,400 BPS */

/* 0x01 = 115,200 BPS */

/* 0x02 = 57,600 BPS */

/* 0x06 = 19,200 BPS */

/* -> 0x0C = 9,600 BPS */

/* 0x18 = 4,800 BPS */

```

        /*      0x30 = 2,400 BPS      */
        outp(PORT + 1, 0x00); /* Set Baud rate - Divisor Latch High Byte */
        outp(PORT + 3, 0x03); /* 8 Bits, Sem paridade, 1 Stop Bit */
        outp(PORT + 2, 0xC7); /* FIFO Control Register */
        outp(PORT + 4, 0x0B); /* Seta DTR, RTS, e OUT2 */
    }

```

```

int Espera_Transmissor(void)
/* Espera o buffer de transmissao esvaziar */
{
    long int i;          /* Contador */

    /* Espera ate que o flag Data Ready esteja setado */
    for(i=0;i<ESPERA && ((inp(PORT+5)/32)%2)==0;i++);

    if(i<ESPERA)
        return 1; /* Transmissor preparado */
    else
        return 0; /* Algum erro ocorreu */
}

```

```

int Espera_Dado(void)
/* Espera que o dado esteja disponivel para leitura da porta */
{
    long int i;          /* Contador */

    for(i=0;i<ESPERA && (inp(PORT+5)%2)==0;i++);

    if (i<ESPERA)
        return 1; /* Dado preparado */
    else
        return 0; /* time out - dado nao chegou */
}

```

```

void Recebe_Resposta(void)
/* Recebe uma resposta do PIC segundo as seguintes regras:
   - Uma resposta em ASCII deve ser precedida pelo codigo 0x50 e deve
   se encerrar com o codigo 0x00
   - Uma resposta numerica deve ser precedida pelo codigo 0x4E e pelo
   numero de bytes da resposta
*/
{
    int i;
    int flagR;
    short word;
    int cont;
    short resposta[30];

    flagR=Espera_Dado();
    if(flagR==0)
        printf("\nNao ha resposta\n");
    else
    {
        word=inp(PORT);
        if(word==0x50)
        {
            for(i=0;word!=0x00&&flagR==1;i++)
            {
                flagR=Espera_Dado();
                word=inp(PORT);
                resposta[i]=word;
            }
        }
    }
}

```

```

    }
    if(flagR==1)
    {
        i=0;
        printf("\n");
        while (resposta[i]!=0x00)
        {
            printf("%c",resposta[i]);
            i++;
        }
        printf("\n");
    }
    else
        printf("\nFormato dos dados recebidos e invalido\n");
}/* Do if */
else
    if(word==0x4E)
    {
        flagR=Espera_Dado();
        if(flagR==0)
            printf("\nFormato dos dados recebidos e invalido\n");
        else
        {
            cont=inp(PORT);
            for(i=0;i<cont&&flagR==1;i++)
            {
                flagR=Espera_Dado();
                resposta[i]=inp(PORT);
            }
            if(flagR==0)
                printf("\nFormato dos dados recebidos e invalido\n");
            else
            {
                printf("\n");
                for(i=0;i<cont;i++)
                {
                    if(resposta[i]<16)
                        printf("0"); /* Correcao da impressao */
                    printf("%x",resposta[i]);
                }
                printf("\n");
            }
        }
    }
}/* Do if */
else
    printf("\nFormato dos dados recebidos e invalido\n");
}/* Do else */
}/* Da funcao */

```

```

void Envia_Mensagem(char mensagem[])
{
    /* Envia um vetor de tamanho definido ao PIC.
       O tamanho do vetor esta contido na posicao '0'.
       O PIC deve interpretar os dados.
    */

```

```

    int tamanho;
    int i;
    int flagR;

    flagR=1;
    tamanho=mensagem[0];

```

```

i=1;
while((i<=tamanho)&&(flagR==1))
{
    flagR=Espera_Transmissor();
    outp(PORT,mensagem[i]);
    i++;
}
if(tamanho<1)
    printf("\nFormato de dados invalido\n");
else
    if(flagR==0)
        printf("\nProblemas na transmissao de dados. Transmissor nao libera\n");
}

void locate(void)
/* Envia ordem de realizar a localizacao e espera pelo valor da distancia */
{
    char mensagem[30];

    mensagem[0]=1; /* Tamanho da mensagem */
    mensagem[1]=0x47; /* Codigo da letra de comando - G */

    Envia_Mensagem(mensagem);

    printf("\nDistancia (em 10e-8mm em hexadecimal): ");
    Recebe_Resposta();
}

void help(void)
/* Exibe lista de comandos */
{
    printf("\nLISTA DE COMANDOS:");
    printf("\nComando\t\t\tSintaxe\t\tUnidade\n");
    printf("-----");
    printf("\nLocalizar\t\tG\t\t");
    printf("\nProgramar velocidade\tV<velocidade>\t10e-1mm/s");
    printf("\nProgramar periodo\tP<periodo>\tNo. de ciclos");
    printf("\nProgramar largura\tW<largura>\tNo. de ciclos");
    printf("\nProgramar comprimento\tL<comprimento>\tNo. de ciclos");
    printf("\nReiniciar comunicacao\tI\t\t");
    printf("\nVerificar parametros\tS\t\t");
    printf("\nModo de teste\t\tT\t\t");
    printf("\nAjuda\t\t\t?\t\t");
    printf("\nSair do programa\tQ\t\t");
    printf("\n\nOBSERVAÇ&ES:\n");
    printf("1. N&o use espaço em branco\n");
    printf("2. Use letras mai&sculas ou min&sculas\n");
    printf("3. N&o use ponto flutuante\n");
}

void setvelocity(char comando[])
/* Programa velocidade do som (em 10e-1mm/s) */
{
    int i ; /* contador */
    long int V ; /* velocidade do som em 10^(-1)mm/s */
    char mensagem[30];

    mensagem[0]=4; /* Tamanho da mensagem */
    for(i=1,V=0;i<strlen(comando);i++)
        V=(V)*10+(comando[i]-48);
    if((V<3000000)||(V>16777215))
        printf("\nArgumento fora da faixa permitida");
}

```

```

else
{
    mensagem[1]=0x56; /* codigo da letra de comando */
    mensagem[2]=V%(0x100); /* LSB da velocidade */
    V=V/(0x100);
    mensagem[3]=V%(0x100);
    V=V/(0x100);
    mensagem[4]=V%(0x100); /* MSB da velocidade */

    Envia_Mensagem(mensagem);

    Recebe_Resposta();

} /* end else */
}

void setperiod(char comando[])
{ /* Programa periodo dos pulsos (em numero de ciclos) */
    char mensagem[30];
    char P;
    int i;

    mensagem[0]=2; /* Tamanho da mensagem */
    mensagem[1]=0x50; /* Codigo da letra de comando - P */

    for(i=1,P=0;i<strlen(comando);i++)
        P=P*10+(comando[i]-48);
    if(P<1||P>100)
        printf("\nArgumento fora da faixa especificada\n");
    else
    {
        mensagem[2]=P; /* Periodo dos pulsos */
        Envia_Mensagem(mensagem);
        Recebe_Resposta();
    }
}

void setwidth(char comando[])
{ /* Programa a Largura de pulso (em numero de ciclos) */
    char mensagem[30];
    int W;
    int i;

    mensagem[0]=2; /* Tamanho da mensagem */
    mensagem[1]=0x57; /* Codigo da letra de comando - W */

    for(i=1,W=0;i<strlen(comando);i++)
        W=W*10+(comando[i]-48);
    if(W<1||W>50)
        printf("\nArgumento fora da faixa especificada\n");
    else
    {
        mensagem[2]=W; /* Largura de pulso */
        Envia_Mensagem(mensagem);
        Recebe_Resposta();
    }
}

void setlength(char comando[])
{ /* Programa comprimento do trem de pulsos (em numero de ciclos) */

```

```

int i;
int L;
char mensagem[30];

mensagem[0]=2;      /* Tamanho da mensagem */
mensagem[1]=0x4C;    /* Codigo da letra de comando - L */

for(i=1,L=0;i<strlen(comando);i++)
    L=L*10+(comando[i]-48);
if(L<2||L>255)
    printf("\nArgumento fora da faixa especificada\n");
else
{
    mensagem[2]=L; /* Comprimento do trem de pulsos */
    Envia_Mensagem(mensagem);
    Recebe_Resposta();
}
}

void Verifica_Parametros(void)
{ /* Imprime velocidade do som, periodo, largura de pulso e comprimento do
trem programados no PIC
*/
    char    mensagem[30];

    mensagem[0]=1; /* Tamanho da mensagem */
    mensagem[1]=0x53; /* Codigo da letra de comando - S */

    Envia_Mensagem(mensagem);

    printf("\nVelocidade: ");
    Recebe_Resposta();

    printf("\nPeriodo: ");
    Recebe_Resposta();

    printf("\nLargura de pulso: ");
    Recebe_Resposta();

    printf("\nComprimento do trem: ");
    Recebe_Resposta();
}

void Teste(void)
{ /* Modo de teste de emissao */
    char    ch;
    char    mensagem[30];

    printf("\nModo de teste. Pressione ESC para encerrar o teste...\n");

    mensagem[0]=1; /*Tamanho da mensagem*/
    mensagem[1]=0x54; /*Codigo do comando de teste - T*/
    Envia_Mensagem(mensagem);
    Recebe_Resposta();

    ch=0;
    for(;;ch!=27;)
    {
        if(kbhit()) ch=getch();
        if(ch==27)

```

```

        {
            mensagem[0]=1; /*Tamanho da mensagem*/
            mensagem[1]=0x5A; /*Codigo do comando parar - Z */
            Envia_Mensagem(mensagem);
            Recebe_Resposta();
        }
    }
}
/***** PROGRAMA PRINCIPAL *****/
void main (void)
{
    /* Variaveis */
    char comando[10]; /* Comando do usuario */
    int flag; /* Indicador geral */
    int p; /* No. da porta de comunicacão */
    int inicio; /* Codigos de inicio */
    int inicializado; /* Indica inicializacao da comunicacao */
    char ch;

    /* Inicializacoes das variaveis */
    comando[0]=00;
    inicio=0;
    ch=0;
    inicializado=0;

    printf("\n\n Programa de comando do PIC - Comunicacao Serial\n\n");

    for(;inicializado==0;) /* Loop de inicializacao */
    {
        p=0;
        for(;p!=1 && p!=2 && p!=3 && p!=4;)
        {
            printf("\nPorta de comunicacão (1,2,3,4): ");
            scanf("%d",&p);
        }
        if(p==1) PORT=0x3f8;
        if(p==2) PORT=0x2f8;
        if(p==3) PORT=0x3e8;
        if(p==4) PORT=0x2e8;

        printf("\nEndereço: %x\n",PORT);

        InicializaPorta(); /* Preparacao da porta serial */

        /* Inicializacao da comunicacao */
        Espera_Transmissor();
        outp(PORT,0xAA); /* Envia codigo de inicio */
        flag=Espera_Dado();
        inicio=inp(PORT);
        if(flag==1&&inicio==0xBB)
        {
            printf("\nComunicacao inicializada com sucesso\n");
            inicializado=1;
        }
        else
            printf("\nFalha na inicializacao da comunicacao\n");
    } /* Fim do loop de inicializacao */

    /* Loop Principal */
    while((comando[0]!='q')&&(comando[0]!='Q'))

```

```

{
    outp(PORT + 2 , 0xC7);/* Limpa os FIFO's da porta serial */
    flag=0;
    printf("\nReady: "); /* Prompt */
    scanf("%s",comando);
    if((comando[0]=='G')||(comando[0]=='g'))
    {
        locate(); /* Ordem para localizacao */
        flag=1; /* Seta flag de passagem */
    }
    if((comando[0]=='S')||(comando[0]=='s'))
    {
        Verifica_Parametros();
        flag=1; /* Seta flag de passagem */
    }
    if((comando[0]=='T')||(comando[0]=='t'))
    {
        Teste(); /* Inicia modo de teste */
        flag=1; /* Seta flag de passagem */
    }

    if((comando[0]=='I')||(comando[0]=='i'))
    {
        Espera_Transmissor();
        outp(PORT,0xAA); /* Envia codigo de inicio */
        flag=Espera_Dado();
        inicio=inp(PORT);
        if(flag==1&&inicio==0xBB)
        {
            printf("\nComunicacao reinicializada\n");
            inicializado=1;
        }
        else
            printf("\nFalha na comunicacao\n");
        flag=1; /* Seta flag de passagem */
    }

    if (comando[0]=='?')
    {
        help(); /* Mostra lista de comandos */
        flag=1;
    }
    if((comando[0]=='V')||(comando[0]=='v'))
    {
        setvelocity(comando);/* Programa veloc. do som */
        flag=1;
    }
    if((comando[0]=='P')||(comando[0]=='p'))
    {
        setperiod(comando);/* Programa periodo */
        flag=1;
    }
    if((comando[0]=='W')||(comando[0]=='w'))
    {
        setwidth(comando); /* Programa largura de pulso */
        flag=1;
    }
    if((comando[0]=='L')||(comando[0]=='l'))
    {
        setlength(comando); /* Programa comprimento do trem */

```



```
        flag=1;
    }
    if((comando[0]=='Q')||(comando[0]=='q'))
        flag=1;

    if (flag==0) /* Se nao houve nenhum comando valido */
        printf("\nComando inexistente. Digite ? para ajuda\n");

} /* Fim do while */

} /* Fim do main */
```

17.6 Código fonte da Unidade móvel

```
;
;Programa do PIC - sistema de localização - Unidade Móvel
;
;Versão de testes
;
;O Programa inclui: interface serial com a unidade de comando (PC),
; geração do trem de pulsos, cronometragem do tempo, cálculo da distância
; e envio de respostas
;Foi desenvolvido conjuntamente com o programa COMMUNIT.EXE e só funciona
; se o mesmo estiver rodando.
;
;Pino de saída do trem de pulsos          - RC1(CCP1) - pino 12

    list p=16c73b
    include <p16c73b.inc>

;***** MACROS *****
;Multiplicação - 1 byte x 1 byte

mpy    macro    multplr,multpld,rlo,high,count,temp

    clrf rlo
    clrf rhigh
    movf    multplr,0
    movwf   temp
    movlw   0x8
    movwf   count
    movf    multpld,0
    bcf STATUS, C    ; Clear carry bit
    ; Loop de multiplicação
    rrf     temp,1
    btfsc   STATUS, C
    addwf   rhigh,1
    rrf     rhigh,1
    rrf     rlo,1
    decfsz  count,1
    goto    $-6

    endm

;*****
;Envia byte para o PC

sndbt   macro    byte

    bsf     STATUS,RP0; Bank 1
    btfss   TXSTA,TRMT
    goto    $-1    ; Espera até liberação de TXREG
    bcf     STATUS,RP0; Bank 0
    movlw   byte
    movwf   TXREG   ; Envia byte
    bsf     STATUS,RP0; Bank 1
    btfsc   TXSTA,TRMT
    goto    $-1    ; Espera até limpar o flag
    bcf     STATUS,RP0; Bank 0

    endm

;*****
;Envia conteúdo de um registrador para o PC
```

```

sndrg    macro    register

        bsf      STATUS,RP0; Bank 1
        btfss    TXSTA,TRMT
        goto     $-1      ; Espera até liberação de TXREG
        bcf      STATUS,RP0; Bank 0
        movf     register,0
        movwf    TXREG    ; Envia registrador
        bsf      STATUS,RP0; Bank 1
        btfsc    TXSTA,TRMT
        goto     $-1      ; Espera até limpar o flag
        bcf      STATUS,RP0; Bank 0

        endm

;*****
;Recebe byte do PC

rcvbt    macro    endereço

        btfss    PIR1,RCIF ; Espera dado
        goto     $-1

        movf     RCREG,0
        movwf    endereço ; Salva byte recebido no endereço

        endm

;***** Registradores Utilizados - Bank 0 - 20h a 7Fh *****
;Para o cálculo da distância (3 bytes X 3 bytes)
VSOUND0      equ 0x30 ;
VSOUND1      equ 0x31 ;
VSOUND2      equ 0x32 ;Vel. do som com resolução de 24 bytes (0.1 mm/s)

TIME0        equ 0x40 ;LSB do tempo cron. no Timer1 (resolução de 200 ns)
TIME1        equ 0x41 ;MSB do tempo cron. no Timer1
TIME2        equ 0x42 ;Número de overflows do Timer1

RESP0        equ 0x50
RESP1        equ 0x51
RESP2        equ 0x52
RESP3        equ 0x53
RESP4        equ 0x54
RESP5        equ 0x55 ;Resposta da multiplicação (6 bytes)

X0           equ 0x60
X1           equ 0x61
Y0           equ 0x62
Y1           equ 0x63
Y2           equ 0x64
Z0           equ 0x65
Z1           equ 0x66
Z2           equ 0x67
W0           equ 0x68
W1           equ 0x69
W2           equ 0x6A
P0           equ 0x6B
P1           equ 0x6C ;Termos da soma final (registradores auxiliares)

TEMP0        equ 0x6D

```

```

TEMP1      equ 0x6E
TEMP2      equ 0x6F
TEMP3      equ 0x45
TEMP       equ 0x46
CONT       equ 0x47 ;Variáveis temporárias e contadores

```

;Para a geração do trem de pulsos

```

PWIDTH      equ 0x70 ;Largura do Pulso (em ciclos)
PERIOD      equ 0x71 ;Período do Pulso (em ciclos)
TRAINL      equ 0x72 ;Comprimento do trem (em ciclos)

```

***** PROGRAMA PRINCIPAL *****

```

;
org 0x00 ; Define endereço base no vetor de início 0x00

```

reset

```

goto start ; Vai para o início do programa

```

```

org 0x04 ; Endereço de início do código do usuário

```

start

***** Constantes - valores default *****

```

bcf STATUS,RP0 ; Bank 0

```

```

movlw 0x3
movwf PWIDTH ; Largura do pulso em ciclos

```

```

movlw 0x6
movwf PERIOD ; Período dos pulsos em ciclos
decf PERIOD,1; Decremento para correção do período

```

```

movlw 0x3C ; (60d)
movwf TRAINL ; Duração do trem de pulsos

```

```

movlw 0xC0
movwf VSOUND0

```

```

movlw 0xE1
movwf VSOUND1

```

```

movlw 0xE4
movwf VSOUND2 ; Velocidade do som (1500.0000 m/s)

```

***** Programações Default dos timers e módulos CCP *****

;Programação do Timer2 - Mede o período do pulso

```

bcf STATUS,RP0 ; Bank 0
movf PERIOD,0 ; Número de ciclos
bsf STATUS,RP0 ; Bank 1
movwf PR2 ; Programa reg. de período
bcf PIE1,TMR2IE; Desarma interrupção
bcf STATUS,RP0 ; Bank 0
bcf PIR1,TMR2IF; Limpa o flag de overflow
movlw 0x00 ; Palavra de controle
movwf T2CON ; Programa Timer2 (sem ligar)
clrf TMR2 ; Limpa Timer2

```

;Programação do módulo CCP2 - Controla a largura do pulso

```

bcf STATUS,RP0 ; Bank 0
movf PWIDTH,0 ; Largura do pulso
movwf CCPR2L ; Programa a largura do pulso
bsf STATUS,RP0 ; Bank 1

```

```

bcf    TRISC,1          ; Programa porta de saída do sinal
bcf    STATUS,RP0 ; Bank 0
bcf    PORTC,1          ; Limpa a saída

```

;Programação do Timer1 - Cronometragem do tempo

```

bcf    STATUS,RP0 ; Bank 0
clrf   TMR1L  ; Limpa Timer1
clrf   TMR1H  ; Limpa Timer1
clrf   TIME2   ; Limpa o contador de overflows
bcf    PIR1,TMR1IF; Limpa flag de overflow
movlw  0x00    ; Palavra de controle
movwf  T1CON   ; Programa palavra de controle (sem ligar)
bsf    STATUS,RP0 ; Bank 1
bcf    PIE1,TMR1IE; Desarma interrupção

```

;Programação do Timer0 - Controla o comprimento do trem

```

bcf    STATUS,RP0 ; Bank 0
movf   TRAINL,0  ; Comprimento do trem
sublw  0xFF      ; Valor que deve ser carregado no Timer0
movwf  TMR0      ; Valor carregado
bcf    INTCON,T0IE; Desarma interrupção
bcf    INTCON,T0IF; Limpa flag de overflow
bsf    STATUS,RP0 ; Bank 1
movlw  0xE1      ; Palavra de programação
movwf  OPTION_REG; Programa palavra de controle (sem começar contagem)

```

;Programação do módulo CCP1 - Captura da contagem do Timer1

```

bcf    STATUS,RP0 ; Bank 0
bcf    PIR2,CCP1IF; Limpa flag de overflow
movlw  0x05      ; Palavra de controle
movwf  CCP1CON   ; Programa palavra de controle
clrf   CCPR1L    ; Limpa reg
clrf   CCPR1H    ; Limpa reg
bsf    STATUS,RP0 ; Bank 1
bcf    PIE2,CCP1IE; Desarma interrupção
bsf    TRISC,2    ; Prepara a entrada
bcf    STATUS,RP0 ; Bank 0
bcf    PORTC,2    ; Limpa a entrada

```

;***** Inicialização da USART *****

```

bsf    STATUS,RP0; Bank1
movlw  0x20      ; Baud rate = 9600 bps
movwf  SPBRG
bcf    TXSTA,TX9 ; Transmissão de 8 bits
bsf    TXSTA,TXEN; Habilita transmissão
bcf    TXSTA,SYNC; Modo Assíncrono
bcf    TXSTA,BRGH; Baixa velocidade
bcf    PIE1,TXIE ; Desarma interrupções de transmissão
bcf    PIE1,RCIE ; Desarma interrupções de recepção
bcf    STATUS,RP0; Bank 0
bsf    RCSTA,SPEN; Habilita porta serial
bcf    RCSTA,RX9 ; Recepção de 8 bits
bsf    RCSTA,CREN; Habilita recepção

```

;***** Inicializacao da Comunicacao *****

```

bcf    STATUS,RP0; Bank 0

```

;Sequencia de inicio de comunicacao

```

In_com bcf    RCSTA,CREN; Desabilita recepção
bsf    RCSTA,CREN; Habilita recepção (P/ evitar travamento da recepcao)

```

```

btfss   PIR1,RCIF ; Espera pelo sinal do PC
goto    $-1       ; Se o sinal não chegou, testa novamente

movf    RCREG,0    ; Leitura do sinal
movwf   TEMP0     ; Armazenamento temporário

movlw   0xAA       ; Sinal de inicializacao esperado
subwf   TEMP0,0    ; Realiza comparação por subtração
btfss   STATUS,Z   ; Segue em frente se recebeu o sinal correto
goto    In_com     ; Se o sinal não é o esperado, continua esperando

;Resposta ao PC se recebeu sinal correto (no inicio ou no reinicio)
Recom   btfss   PIR1,TXIF
        goto    $-1 ; Espera até liberação de TXREG
        movlw   0xBB ; Sinal de resposta
        movwf   TXREG ; Envia resposta ao PC

;***** Loop principal - leitura de comandos *****
        bcf     STATUS,RP0; Bank 0

Loop     bcf     RCSTA,CREN; Desabilita recepção
        bsf     RCSTA,CREN; Habilita recepção (P/ evitar travamento da recepcao)

w_com    btfss   PIR1,RCIF ; Loop de espera por um comando
        goto    w_com     ; Se um comando não chegou, testa novamente

        movf    RCREG,0    ; Leitura do comando
        movwf   TEMP0     ; Armazenamento temporário

        movlw   0x47       ; Código ASCII do comando G (realizar localização)
        subwf   TEMP0,0    ; Realiza comparação por subtração
        btfss   STATUS,Z   ; Testa o resultado da operação
        goto    $+3        ; Se o result. não é nulo, verf. o próximo comando
        call    LOCATE     ; Se o resultado é nulo, executa localização
        goto    Loop       ; Volta ao início do Loop e espera novo comando

        movlw   0x53       ; Código ASCII do comando S (verificar parametros)
        subwf   TEMP0,0    ; Realiza comparação por subtração
        btfss   STATUS,Z   ; Testa o resultado da operação
        goto    $+3        ; Se o result. não é nulo, verf. o próximo comando
        call    VERIFY     ; Se o resultado é nulo, envia parametros
        goto    Loop       ; Volta ao início do Loop e espera novo comando

        movlw   0x54       ; Código ASCII do comando T (teste de emissão)
        subwf   TEMP0,0    ; Realiza comparação por subtração
        btfss   STATUS,Z   ; Testa o resultado da operação
        goto    $+3        ; Se o result. não é nulo, verf. o próximo comando
        call    TESTE     ; Se o resultado é nulo, inicia teste
        goto    Loop       ; Volta ao início do Loop e espera novo comando

        movlw   0xAA       ; Código de reinicializacao
        subwf   TEMP0,0    ; Realiza comparação por subtração
        btfss   STATUS,Z   ; Testa o resultado da operação
        goto    $+2        ; Se o result. não é nulo, verf. o próximo comando
        goto    Recom     ; Envia resposta de reinicializacao

        movlw   0x56       ; Código ASCII do comando V (prog. velocidade)
        subwf   TEMP0,0    ; Realiza comparação por subtração
        btfss   STATUS,Z   ; Testa o resultado da operação

```

```

goto    $+3      ; Se o result. não é nulo, verif. o próximo comando
call    SET_V    ; Se o resultado é nulo, programa velocidade
goto    Loop     ; Volta ao início do Loop e espera novo comando

movlw   0x50     ; Código ASCII do comando P (prog. periodo do trem)
subwf   TEMP0,0  ; Realiza comparação por subtração
btfss   STATUS,Z ; Testa o resultado da operação
goto    $+3      ; Se o result. não é nulo, verif. o próximo comando
call    SET_P    ; Se o resultado é nulo, programa periodo
goto    Loop     ; Volta ao início do Loop e espera novo comando

movlw   0x57     ; Código ASCII do comando W (prog. largura de pulso)
subwf   TEMP0,0  ; Realiza comparação por subtração
btfss   STATUS,Z ; Testa o resultado da operação
goto    $+3      ; Se o result. não é nulo, verif. o próximo comando
call    SET_W    ; Se o resultado é nulo, programa largura
goto    Loop     ; Volta ao início do Loop e espera novo comando

movlw   0x4C     ; Código ASCII do comando L (prog. comp. do trem)
subwf   TEMP0,0  ; Realiza comparação por subtração
btfsc   STATUS,Z ; Testa o resultado da operação
call    SET_L    ; Se o resultado é nulo, programa comprimento

goto    Loop     ; Volta ao início do Loop e espera novo comando

```

***** ROTINAS DE EXECUÇÃO DOS COMANDOS *****
LOCATE ; Realiza localização e envia resposta

```

bcf     STATUS,C      ; O bit C será usado para acusar erro
movlw   0x0F          ; Palavra de controle
bsf     T2CON,2        ; Início do Timer2
movwf   CCP2CON        ; Liga o módulo PWM
bsf     T1CON,TMR1ON   ; Início da Cronometragem
bsf     STATUS,RP0     ; Bank 1
bcf     OPTION_REG,T0CS ; Início da medição do comprimento do trem
bcf     STATUS,RP0     ; Bank 0

```

;Espera pelo fim do trem e pára os pulsos

```

btfss   INTCON,T0IF    ;
goto    $-1            ; Se Timer0 não estourar, continua no loop

```

```

movlw   0x00          ; Palavra de controle
movwf   CCP2CON        ; Desliga o modulo PWM
bcf     T2CON,2        ; Desliga Timer2
bcf     PORTC,1        ; Limpa a saída de sinal
bsf     STATUS,RP0     ; Bank 1
bsf     OPTION_REG,T0CS ; Desliga Timer0
bcf     STATUS,RP0     ; Bank 0
movlw   0x01          ; Carrega constante

```

;Monitora cronometragem, espera pela resposta para capturar contagem

```

cont    btfss   PIR1,TMR1IF ; Teste de overflow
goto    $+6              ; Se não overflow, não incrementa TIME2
btfss   CCP1H,7          ; Se overflow, testa se houve captura antes
addwf   TIME2,1          ; Se bit=0, não houve captura -> incrementa
btfsc   STATUS,C         ; Se houver overflow de TIME2,
goto    erro             ; considera-se que houve erro
bcf     PIR1,TMR1IF      ; Limpa flag de overflow
btfss   PIR2,CCP1IF      ; Teste de captura
goto    cont             ; Se não houve captura, continua no loop

```

```

btfss   PIR1,TMR1IF ; Teste de overflow
goto    $+5          ; Se não overflow, não incrementa TIME2
btfss   CCPR1H,7     ; Se overflow, testa se houve captura antes
addwf   TIME2,1       ; Se bit=0, captura foi depois -> incrementa
btfsc   STATUS,C      ; Se houver overflow de TIME2,
goto    erro          ; considera-se que houve erro
bcf     PIR1,TMR1IF   ; Limpa flag de overflow

```

```

movf    CCPR1L,0
movwf   TIME0
movf    CCPR1H,0
movwf   TIME1          ; Guarda o valor da cronometragem

```

```

decf    TIME1,1        ; Correção do tempo cronometrado
movlw   0xFF           ; devido ao atraso na unidade móvel
subwf   TIME1,0
btfsc   STATUS,Z
decf    TIME2,1

```

;Executa multiplicação

```

;***** A multiplicação é feita pela decomposição de cada número em bytes *
;***** Os números a serem multiplicados (VSOUND e TIME) possuem 3 bytes cada*
;***** O resultado possui 6 bytes *
;***** Vários testes foram feitos. Multiplicação OK *
;***** No de ciclos : ~729 ciclos (145,8 us) *
;***** A resposta da multiplicação é em 10e-8 mm e é a distância entre *
;***** as unidades *

```

```

mpy     VSOUND2,TIME2,X0,X1,CONT,TEMP

```

```

mpy     VSOUND0,TIME0,P0,P1,CONT,TEMP

```

```

clrf    Y1
clrf    Y2
mpy     VSOUND2,TIME1,TEMP0,TEMP1,CONT,TEMP
mpy     TIME2,VSOUND1,TEMP2,TEMP3,CONT,TEMP
movf    TEMP0,0
addwf   TEMP2,0
movwf   Y0
btfsc   STATUS,C; testa o bit carry
incf    Y1,1
movf    TEMP1,0
addwf   TEMP3,0
btfsc   STATUS,C; testa o bit carry
incf    Y2,1
addwf   Y1,1

```

```

clrf    W1
clrf    W2
mpy     VSOUND1,TIME0,TEMP0,TEMP1,CONT,TEMP
mpy     VSOUND0,TIME1,TEMP2,TEMP3,CONT,TEMP
movf    TEMP0,0
addwf   TEMP2,0
movwf   W0
btfsc   STATUS,C; testa o bit carry
incf    W1,1
movf    TEMP1,0
addwf   TEMP3,0
btfsc   STATUS,C; testa o bit carry

```



```

incf    W2,1
addwf   W1,1

clrf    Z1
clrf    Z2
mpy     VSOUND2,TIME0,TEMP0,TEMP1,CONT,TEMP
mpy     VSOUND0,TIME2,TEMP2,TEMP3,CONT,TEMP
movf    TEMP0,0
addwf   TEMP2,0
movwf   Z0
btfsc   STATUS,C; testa o bit carry
incf    Z1,1
movf    TEMP1,0
addwf   TEMP3,0
btfsc   STATUS,C; testa o bit carry
incf    Z2,1
addwf   Z1,1
mpy     VSOUND1,TIME1,TEMP0,TEMP1,CONT,TEMP
movf    TEMP0,0
addwf   Z0,1
btfss   STATUS,C; pula se o carry for 1
goto    $+5      ; se o carry=0, não incrementar os outros bytes
movlw   0x01
addwf   Z1,1
btfsc   STATUS,C
incf    Z2,1
movf    TEMP1,0
addwf   Z1,1
btfsc   STATUS,C
incf    Z2,1

```

;Soma dos termos encontrados

```

clrf    RESP2
clrf    RESP3
clrf    RESP4
clrf    RESP5 ; Limpa os registradores de resposta

```

```

movf    P0,0
movwf   RESP0 ; Primeiro byte do resultado

```

```

movf    W0,0
addwf   P1,0
movwf   RESP1 ; Segundo byte do resultado
btfsc   STATUS,C; testa o bit carry
incf    RESP2,1 ; incrementa o terceiro byte

```

```

movf    W1,0
addwf   Z0,0
addwf   RESP2,1 ; terceiro byte do resultado
btfsc   STATUS,C; testa o bit carry
incf    RESP3,1 ; incrementa o quarto byte

```

```

movf    W2,0
addwf   Z1,0
addwf   RESP3,1
btfsc   STATUS,C; testa o bit carry
incf    RESP4,1 ; incrementa o quinto byte
movf    Y0,0
addwf   RESP3,1 ; completa a soma do quarto byte
btfsc   STATUS,C; testa o bit carry

```

```

incf    RESP4,1      ; incrementa o quinto byte

movf    Z2,0
addwf   Y1,0
addwf   RESP4,1
btfsc   STATUS,C; testa o bit carry
incf    RESP5,1      ; incrementa o sexto byte
movf    X0,0
addwf   RESP4,1      ; completa a soma do quinto byte
btfsc   STATUS,C; testa o bit carry
incf    RESP5,1      ; incrementa o sexto byte

movf    X1,0
addwf   RESP5,1
movf    Y2,0
addwf   RESP5,1      ; completa a soma do sexto byte (operação completa)

;Envia resposta via canal serial
sndbt   0x4E      ; código de número
sndbt   0x06      ; número de bytes do número
sndrg   RESP0     ; primeiro byte da resposta
sndrg   RESP1     ; segundo byte da resposta
sndrg   RESP2     ; terceiro byte da resposta
sndrg   RESP3     ; quarto byte da resposta
sndrg   RESP4     ; quinto byte da resposta
sndrg   RESP5     ; sexto byte da resposta

;Prepara timers para o próximo ciclo de emissão-recepção
;Timer0
prp_tm  movf    TRAINL,0 ; Comprimento do trem
sublw   0xFF      ; Valor que deve ser carregado no Timer0
movwf   TMR0      ; Valor carregado
bcf     INTCON,T0IF; Limpa flag de overflow
;Timer1
bcf     T1CON,TMR1ON; Desliga Timer1
clrf    TMR1L
clrf    TMR1H     ; Limpa Timer1
clrf    TIME2     ; Limpa o contador de overflows
bcf     PIR1,TMR1IF; Limpa flag de overflow
;Timer2
bcf     PIR1,TMR2IF; Limpa o flag de overflow
clrf    TMR2      ; Limpa Timer2
;Capture
bcf     PIR2,CCP2IF; Limpa flag de overflow
clrf    CCPR2L     ; Limpa reg
clrf    CCPR2H     ; Limpa reg
bcf     PORTC,1    ; Limpa a entrada

;Retorna ao Loop principal
return

;Mensagem de erro - resposta não recebida
erro    sndbt   0x50      ; código de palavra
sndbt   0x45      ; 'E'
sndbt   0x72      ; 'r'
sndbt   0x72      ; 'r'
sndbt   0x6F      ; 'o'
sndbt   0x00      ; 'null'

;Retorna a preparação dos timers

```

```

        goto    prp_tm
;*****
VERIFY    ;Envia os parametros programados

        ;Velocidade do som
sndbt    0x4E
sndbt    0x03
sndrg    VSOUND2
sndrg    VSOUND1
sndrg    VSOUND0

        ;Periodo
sndbt    0x4E
sndbt    0x01
sndrg    PERIOD

        ;Largura de pulso
sndbt    0x4E
sndbt    0x01
sndrg    PWIDTH

        ;Comprimento do trem
sndbt    0x4E
sndbt    0x01
sndrg    TRAINL

        ;Retorna ao Loop principal
return
;*****
TESTE    ;Realiza teste de emissão

        sndbt    0x50    ; código de palavra
        sndbt    0x54    ; 'T'
        sndbt    0x65    ; 'e'
        sndbt    0x73    ; 's'
        sndbt    0x74    ; 't'
        sndbt    0x65    ; 'e'
        sndbt    0x20    ; ' '
        sndbt    0x49    ; 'I'
        sndbt    0x6E    ; 'n'
        sndbt    0x69    ; 'i'
        sndbt    0x63    ; 'c'
        sndbt    0x69    ; 'i'
        sndbt    0x61    ; 'a'
        sndbt    0x64    ; 'd'
        sndbt    0x6F    ; 'o'
        sndbt    0x00    ; 'null'

tstcon    nop
        ;Programação do Timer0 para medição do período de repetição
bsf       STATUS,RP0; Bank 1
movlw    0xE5        ; Palavra de programação
movwf    OPTION_REG ; Programa palavra de controle
bcf       OPTION_REG,T0CS; Liga Timer0
bcf       STATUS,RP0; Bank 0

        ;Medição do intervalo
clrf     TMR0        ; Limpa Timer0
bcf       INTCON,T0IF ; Limpa flag de overflow
btfss    INTCON,T0IF ;

```

```

goto    $-1                ; Espera Timer0 estourar

;Reprogramação para medição do comp. do trem
bsf     STATUS,RP0; Bank 1
movlw   0xE1              ; Palavra de programação
movwf   OPTION_REG ; Programa palavra de controle
bcf     OPTION_REG,T0CS; Liga Timer0
bcf     STATUS,RP0; Bank 0

movlw   0x0F              ; Palavra de controle
bsf     T2CON,2           ; Início do Timer2
movwf   CCP2CON           ; Liga o modulo PWM
movf    TRAINL,0          ; Comprimento do trem
sublw   0xFF              ; Valor que deve ser carregado no Timer0
movwf   TMR0              ; Valor carregado
bcf     INTCON,T0IF       ; Limpa o flag de overflow

;Espera pelo fim do trem e pára os pulsos
btfss   INTCON,T0IF       ;
goto    $-1                ; Se Timer0 não estourar, continua no loop

movlw   0x00              ; Palavra de controle
movwf   CCP2CON           ; Desliga o modulo PWM
bcf     T2CON,2           ; Desliga Timer2
bcf     PORTC,1           ; Limpa a saída de sinal

;Prepara timer para o próximo ciclo de recepção-emissão
;Timer2
bcf     PIR1,TMR2IF; Limpa o flag de overflow
clrf    TMR2              ; Limpa Timer2

btfss   PIR1,RCIF ; Testa se recebeu comando de parada
goto    tstcon            ; Continua no teste se não recebeu nenhum comando

movf    RCREG,0           ; Leitura do comando
movwf   TEMP0             ; Armazenamento temporário

movlw   0x5A              ; Código ASCII do comando Z (Encerrar teste)
subwf   TEMP0,0           ; Realiza comparação por subtração
btfss   STATUS,Z          ; Testa o resultado da operação
goto    tstcon            ; Continua com o teste se não recebeu comando de parada

sndbt   0x50              ; código de palavra
sndbt   0x54              ; 'T'
sndbt   0x65              ; 'e'
sndbt   0x73              ; 's'
sndbt   0x74              ; 't'
sndbt   0x65              ; 'e'
sndbt   0x20              ; ' '
sndbt   0x45              ; 'E'
sndbt   0x6E              ; 'n'
sndbt   0x63              ; 'c'
sndbt   0x65              ; 'e'
sndbt   0x72              ; 'r'
sndbt   0x72              ; 'r'
sndbt   0x61              ; 'a'
sndbt   0x64              ; 'd'
sndbt   0x6F              ; 'o'
sndbt   0x00              ; 'null'

```

```

    return          ; Encerra o teste
;*****
SET_V ;Programa nova velocidade do som

    rcvbt    VSOUND0
    rcvbt    VSOUND1
    rcvbt    VSOUND2

;Confirmação à unidade de comando
    sndbt    0x50    ; código de palavra
    sndbt    0x4F    ; 'O'
    sndbt    0x4B    ; 'K'
    sndbt    0x00    ; 'null'

    return
;*****
SET_P ;Programa novo periodo do trem de pulsos

    rcvbt    PERIOD
    decf     PERIOD,1 ; Correção do período
    movf     PERIOD,0
    movwf    TMR2    ; Carrega Timer2 com valor máximo

    bsf      STATUS,RP0 ; Bank 1
    movwf    PR2      ; Programa reg. de período
    bcf      STATUS,RP0 ; Bank 0

;Confirmação à unidade de comando
    sndbt    0x50    ; código de palavra
    sndbt    0x4F    ; 'O'
    sndbt    0x4B    ; 'K'
    sndbt    0x00    ; 'null'

    return
;*****
SET_W ;Programa nova largura de pulso

    rcvbt    PWIDTH

    movf     PWIDTH,0
    movwf    CCPR2L    ; Programa a largura do pulso

;Confirmação à unidade de comando
    sndbt    0x50    ; código de palavra
    sndbt    0x4F    ; 'O'
    sndbt    0x4B    ; 'K'
    sndbt    0x00    ; 'null'

    return
;*****
SET_L ;Programa novo comprimento do trem

    rcvbt    TRAINL

    movf     TRAINL,0 ; Comprimento do trem
    sublw    0xFF     ; Valor que deve ser carregado no Timer0
    movwf    TMR0     ; Valor carregado

;Confirmação à unidade de comando

```

```
sndbt 0x50 ; código de palavra
sndbt 0x4F ; 'O'
sndbt 0x4B ; 'K'
sndbt 0x00 ; 'null'
```

```
return
```

```
,*****
```

```
end
```

17.7 Código fonte da Unidade fixa

```
;
;Programa do PIC - sistema de localização - Unidade Fixa
;
;Pino de entrada: RB1 - pino 22
;O PIC responde quando o pino de entrada = HIGH
;
;A resposta é composta de um tempo de espera determinado e
; um trem de pulsos com período, largura de pulso e duração
; programados
;
;Pino de saída do trem: RC2(CCP1) - pino 13
;
;Valores de período, largura de pulso e duração do trem devem ser
; modificados no programa

    list p=16c73b
    include <p16c73b.inc>

;***** Registradores Utilizados - Bank 0 - 20h a 7Fh *****
;Para a geração do trem de pulsos
PWIDTH      equ 0x70 ;Largura do Pulso (em ciclos)
PERIOD       equ 0x71 ;Período do Pulso (em ciclos)
TRAINL      equ 0x72 ;Duração do trem (em ciclos)

;***** PROGRAMA PRINCIPAL *****
org 0x00 ; Define endereço base no vetor de início 0x00
reset
    goto start ; Vai para o início do programa

    org 0x04 ; Endereço de início do código do usuário

start
;***** Constantes - valores programados *****
    bcf    STATUS,RP0 ; Bank 0

    movlw  0x2
    movwf  PWIDTH      ; Largura do pulso em ciclos

    movlw  0x6
    movwf  PERIOD      ; Período dos pulsos em ciclos
    decf   PERIOD,1; Decremento para correção do período

    movlw  0x24 ; (36d) - 6 pulsos
    movwf  TRAINL      ; Duração do trem de pulsos

;***** Programações Default dos timers e módulo CCP1 *****
;Programação do Timer2 - Mede o período do pulso
    bcf    STATUS,RP0 ; Bank 0
    movf   PERIOD,0 ; Número de ciclos
    bsf    STATUS,RP0 ; Bank 1
    movwf  PR2 ; Programa reg. de período
    bcf    PIE1,TMR2IE; Desarma interrupção
    bcf    STATUS,RP0 ; Bank 0
    bcf    PIR1,TMR2IF; Limpa o flag de overflow
    movlw  0x00 ; Palavra de controle
    movwf  T2CON ; Programa Timer2 (sem ligar)
    clrf   TMR2 ; Limpa Timer2
```

```

;Programação do módulo CCP1 - Controla a largura do pulso
bcf STATUS,RP0 ; Bank 0
movf PWIDTH,0 ; Largura do pulso
movwf CCP1L ; Programa a largura do pulso
bsf STATUS,RP0 ; Bank 1
bcf TRISC,2 ; Programa porta de saída do sinal
bcf STATUS,RP0 ; Bank 0
bcf PORTC,2 ; Limpa a saída

```

```

;Programação do Timer0 - Controla o comprimento do trem
bcf STATUS,RP0 ; Bank 0
bcf INTCON,T0IE; Desarma interrupção
bcf INTCON,T0IF; Limpa flag de overflow
bsf STATUS,RP0 ; Bank 1
movlw 0xE8 ; Palavra de programação
movwf OPTION_REG ; Programa palavra de controle
bcf OPTION_REG,T0CS; Liga Timer0

```

```

;***** Loop de Espera *****

```

```

bsf STATUS,RP0; Bank 1
bsf TRISB,1 ; Prepara a entrada do sinal

bcf STATUS,RP0; Bank 0
bcf PORTB,1 ; Limpa a entrada

```

```

Loop btfsc PORTB,1 ; Testa a entrada de sinal
call RESPOSTA ; Envia sinal se entrada = HIGH

```

```

goto Loop ; Continua no Loop de espera

```

```

;***** ROTINA DE EXECUÇÃO DO COMANDO *****

```

```

RESPOSTA; Espera um tempo determinado (51 us) e envia o sinal de resposta

```

```

clrf TMR0 ; Limpa Timer0
bcf INTCON,T0IF ; Limpa flag de overflow
btfss INTCON,T0IF ;
goto $-1 ; Espera Timer0 estourar

movlw 0x0F ; Palavra de controle
bsf T2CON,2 ; Início do Timer2
movwf CCP1CON ; Liga o modulo PWM
movf TRAINL,0 ; Comprimento do trem
sublw 0xFF ; Valor que deve ser carregado no Timer0
movwf TMR0 ; Valor carregado
bcf INTCON,T0IF ; Limpa o flag de overflow

```

```

;Espera pelo fim do trem e pára os pulsos
btfss INTCON,T0IF ;
goto $-1 ; Se Timer0 não estourar, continua no loop

```

```

movlw 0x00 ; Palavra de controle
movwf CCP1CON ; Desliga o modulo PWM
bcf T2CON,2 ; Desliga Timer2
bcf PORTC,2 ; Limpa a saída de sinal

```

```

;Prepara timer para o próximo ciclo de recepção-emissão
;Timer2
bcf PIR1,TMR2IF; Limpa o flag de overflow
clrf TMR2 ; Limpa Timer2

```

```

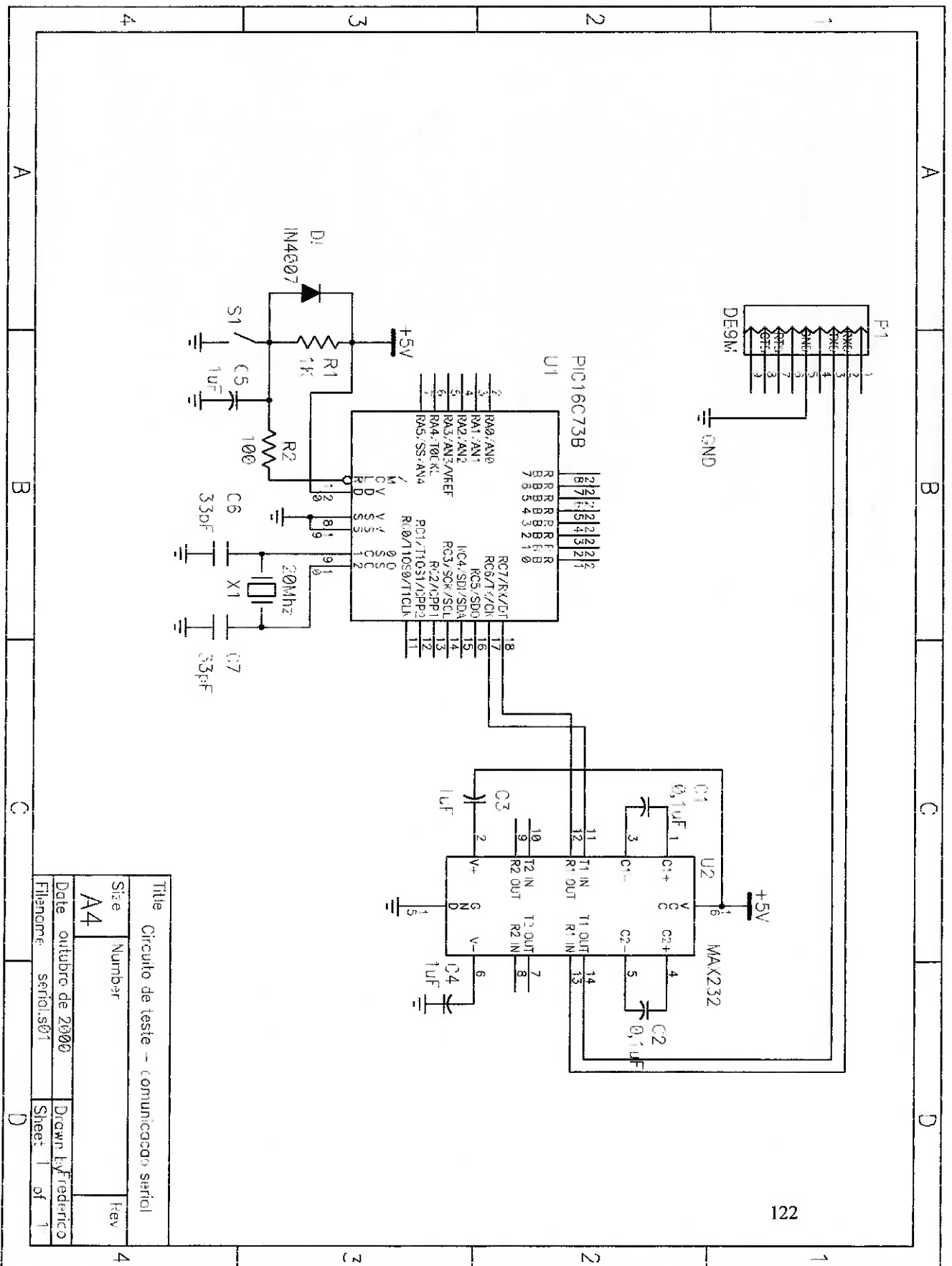
;Retorna ao Loop de espera

```



```
        return
    .*****
    ,
    end
```

17.8 Esquema do circuito de teste



Title			
Circuito de teste - (comunicação serial)			
Size	Number	Rev	
A4			
Date	outubro de 2000	Drawn by: Fredenico	
Filename	serial.s01	Sheet	1 of 1