

MSX Article

MARMSX

*Som de 8 bits no
PSG*

Resumo

Este artigo é baseado em um artigo publicado na MSX Assembly Page [1], onde iremos abordar como reproduzir sons de 8 bits do PC com boa qualidade no chip de som nativo do MSX, o PSG AY-3-8910-A.

1- Introdução

Foi visto no artigo “Digitalização de sons para o MSX” [2] duas formas de digitalização de sons para o MSX. A primeira é através do clique do teclado da PPI, no qual podemos discriminar apenas dois valores, ou seja, um formato de som de 1 bit. A segunda é através do controle do volume do PSG, onde podemos discriminar 16 níveis em um de seus canais, gerando sons de 4 bits. Entretanto, podemos combinar os três canais de PSG de forma a aumentar a discriminação dos níveis de som e sermos capazes de reproduzir até 816 níveis, aumentando e muito a qualidade do som. É isso que veremos nos próximos capítulos.

2- A forma logarítmica do volume do PSG

O controle de volume PSG do MSX opera na forma logarítmica. A cada dois níveis que descemos na escala nominal do volume, estamos na realidade diminuindo sua amplitude pela metade [1][3]. O datasheet da General Instrument [3] fornece um gráfico mostrando a relação entre o valor nominal do volume (digital) e o valor normalizado da voltagem produzida (analgico).

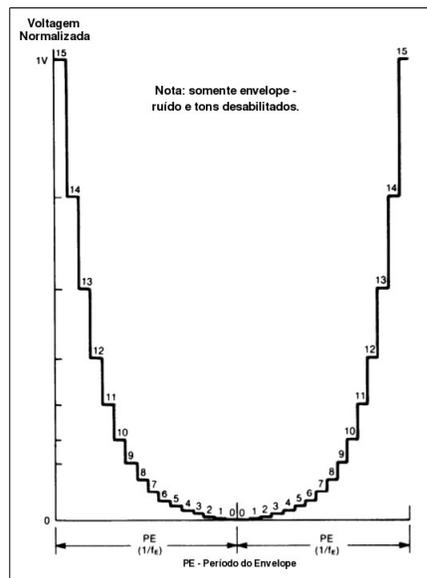


Figura 2.1 – Conversão dos valores digitais para analógico do PSG [3].

A fórmula que relaciona o valor digital e a resposta analógica (DAC) [1] é apresentada a seguir.

$$y=2^{\frac{-(15-n)}{2}} \quad \text{fórmula (1)}$$

Onde y é a resposta analógica DAC para o valor digital correspondente de 0 a 15.

A fórmula (1) pode ser entendida da seguinte maneira: suponha um conjunto de elementos que cresce a uma *razão* constante, de tantos em tantos *passos*. Para calcular o valor de um elemento y de um *índice* qualquer desse conjunto, temos:

$$y=\text{razão}^{\frac{\text{índice}}{\text{passo}}}$$

Por exemplo, seja a progressão exponencial na base 2:

1, 2, 4, 8, 16, 32, etc.

Onde a razão entre os membros anteriores e posteriores é constante, e o membro seguinte é sempre o dobro do anterior. Logo, a *razão* é 2 e o *passo* é 1.

$$y=2^{\frac{i}{1}}=2^i$$

O valor de i começa em 0. Ao utilizarmos a fórmula, encontramos para o quarto elemento do conjunto, cujo índice é igual a 3, o valor 8.

Agora suponha que nosso conjunto dobrasse de valor a cada 3 unidades. Então teríamos:

$$y=2^{\frac{i}{3}}$$

Os membros do conjunto são: 1; 1,26; 1,59; 2; 2,52; 3,18; 4; etc

Na expressão exponencial, o valor de y “explode” conforme o valor de i aumenta. No caso do PSG, o valor de y amortiza e não explode. Assim, deveremos substituir o valor de i por uma expressão que diminua o valor do numerador conforme i aumente. Além disso, sabemos que quando $i=15$, o valor do DAC é 1,0. Ora, qualquer número elevado a zero é igual a 1. Logo, a expressão do numerador deverá ser zero quando $i=15$. Assim, concluímos que a expressão do numerador é $15-i$.

Por fim, o sinal de negativo em uma equação exponencial faz com que a razão seja invertida, ou seja, $\text{razão} = 1/\text{razão}$. Dessa forma, temos a seguinte equivalência:

$$y=2^{\frac{-(15-n)}{2}} = \frac{1}{2^{\frac{15-n}{2}}}$$

A partir da fórmula (1), podemos calcular os valores de DAC e PCM de 8 bits não sinalizados correspondente a cada valor de volume do PSG. Para calcular o valor PCM, basta apenas multiplicar o valor do DAC por 255. O valor do DAC é 0 quando o PSG é 0.

Volume PSG	DAC	PCM 8 bits
0	0	0
1	0,0078125	2
2	0,0110485435	3
3	0,015625	4
4	0,0220970869	6
5	0,03125	8
6	0,0441941738	11
7	0,0625	16
8	0,0883883476	23
9	0,125	32
10	0,1767766953	45
11	0,25	64
12	0,3535533906	90
13	0,5	128
14	0,7071067812	180
15	1	255

Tabela 2.1. Conversão dos valores de volume do PSG para DAC e PCM.

Para realizar o caminho inverso e encontrar o respectivo valor do volume do PSG através de um valor DAC, temos:

$$\log_a b = X \quad \text{e} \quad a^X = b$$

Então:

$$b = 2^{\frac{-(15-n)}{2}}$$

Onde:

- $a = 2$
- $X = -(15-n) / 2$
- $b = DAC \text{ ou } y$

A incógnita nesse caso é o valor n presente na expressão X .

Assim:

$$\log_2 y = \frac{-(15-n)}{2}$$

$$2 \times \log_2 y = -15 + n$$

$$2 \times \log_2 y + 15 = n \quad \text{fórmula (2)}$$

3- Combinando os três canais do PSG

O PSG permite a mistura dos seus três canais de som, fazendo com que a potência resultante seja a soma dos valores DAC de cada canal [1]. A figura 3.1 apresenta a resposta do PSG para o uso de um, dois e três canais ao mesmo tempo, a partir da nota dó (oitava 4) com volume igual a 15 em cada canal.

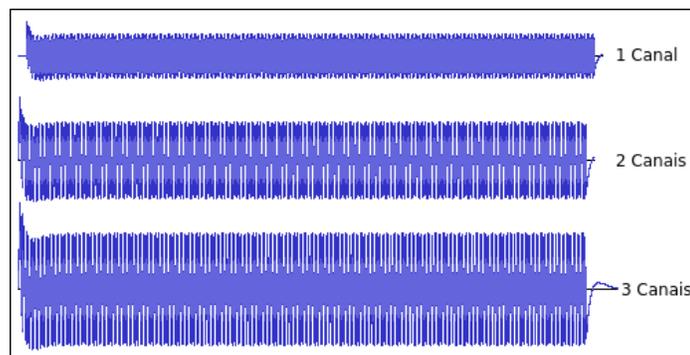


Figura 3.1 – Resposta do PSG para 1, 2, ou 3 canais de som para a nota dó.

Além disso, ao combinar diferentes canais, podemos obter mais valores DAC do que somente os 16 em cada canal. Por exemplo, para obter o valor DAC igual a 0,51, podemos somar o volume 13 (0,50) de um canal com o volume 2 (0,01) de outro [1].

Como cada canal possui 16 níveis ou 3 bits, juntando os três canais temos ao todo 4096 níveis ou 12 bits. Entretanto, ao eliminarmos os sons repetidos, obtemos 816 níveis. Isto dá e sobra para reproduzir os 256 níveis do formato de 8 bits do PC.

Nos formatos de sons do MSX de 1 e 4 bits [2], o arquivo de som a ser reproduzido no MSX já possui o valor direto do clique do teclado ou do volume do canal. No caso do arquivo PCM 8 bit não sinalizado, o valor ainda é do PCM e necessitamos convertê-lo para a combinação certa de cada volume dos três canais do PSG. Para isso, será utilizada uma tabela de conversão na memória do MSX, onde os índices correspondem aos valores PCM e cada linha da tabela contém três valores: o volume do canal A, o volume do canal B e o volume do canal C do PSG.

O desafio agora é encontrar um subconjunto de 256 combinações de volumes dos 816 possíveis, onde o erro seja o menor possível para os 256 valores do PCM.

3.1. A construção da tabela de conversão entre PCM e PSG

A tabela de conversão entre PCM e PSG a ser utilizada no MSX possui 256 entradas. Entretanto, o primeiro passo é construir uma tabela com as 816 combinações do PSG, de forma com que possamos avaliar qual é o melhor sub-conjunto de 256 níveis dos 816 para os valores PCM. O número 816 vem do cálculo do número de combinações com repetição, que é feito de acordo com a fórmula:

$$C_{n,p} = \frac{(n+p-1)!}{k!(n-1)!} \quad \text{fórmula (3)}$$

Assim:

$$C_{16,3} = \frac{(16+3-1)!}{3!(16-1)!} = \frac{18!}{3!15!} = \frac{18 \times 17 \times 16 \times 15!}{3!15!} = \frac{18 \times 17 \times 16}{3 \times 2 \times 1} = \frac{4896}{6} = 816$$

O programa a seguir é uma adaptação em C de um programa encontrado em [1]. Ele cria uma tabela com as 816 combinações possíveis dos volumes dos três canais do PSG.

```
#include <math.h>

double psg_dac[16];
double psg_table[816][4];
unsigned int map[256];

void psg_values() {
    int n;
    psg_dac[0] = 0;
    for (n=1; n<16; n++)
        psg_dac[n] = pow(2.0, -(15-n)/2.0);
}

void create_psg_table()
{
    int p=0, i, j, k;
    for (i=0; i<16; i++) {
        for (j=i; j<16; j++) {
            for (k=j; k<16; k++) {
                psg_table[p][0] = (double) i;
                psg_table[p][1] = (double) j;
                psg_table[p][2] = (double) k;
                psg_table[p][3] = psg_dac[i] + psg_dac[j] + psg_dac[k];
                p++;
            }
        }
    }
}
```

A tabela *psg_table* possui o volume de cada um dos três canais (colunas 0 a 2) mais os valores DAC do PSG (coluna 3), que variam de 0,0 a 3,0. Para cada valor PCM em análise, devemos encontrar a posição da tabela cujo valor DAC desse PCM é o mais próximo do valor DAC do PSG.

O código a seguir é uma adaptação de [1], que calcula a melhor combinação de volumes do PSG para os valores PCM.

```
void get_best_combinations(double scale) {
    int i, p, best_idx = 0;
    double total_error = 0, dac, diff, min_diff;
    for (i=0; i<256; i++) {
        dac = scale * (i / 255.0);

        best_idx = 0;
        min_diff = fabs(dac - psg_table[0][3]);
        for (p=1; p<816; p++) {
            diff = fabs(dac - psg_table[p][3]);
            if (diff < min_diff) {
                min_diff = diff;
                best_idx = p;
            }
        }

        map[i] = best_idx;
        total_error += min_diff;
    }

    printf(" Total error: %f\n", total_error);
    printf(" Total error normalized: %f\n", total_error / scale);
}
```

O valor do DAC correspondente ao valor PCM é calculado da seguinte maneira:

$$DAC = \frac{PCM}{255} \quad \text{fórmula (4)}$$

O erro total é calculado para as 256 entradas de i . Este erro servirá de referência para um truque que será apresentado a seguir para se conseguir a melhor combinação de valores da tabela *psg_table*.

O artigo em [1] sugere que ateremos a escala dos valores de PCM entre 0,0 e 3,0, pois assim espalharíamos esses valores no espectro dos valores da tabela *psg_table*, encontrando os valores da tabela que resultariam no menor erro total. Assim:

$$DAC = \frac{PCM}{255} \times scale \quad \text{fórmula (5)}$$

É necessário também normalizar o erro total, de acordo com a escala utilizada.

```
// Normaliza erro
total_error = total_error / scale;
```

Após alguns testes [1], o artigo sugere que os melhores valores estão entre 1,0 e 1,5, onde o valor igual 1,328 é o valor de escala onde o erro encontrado é o menor.

4. Análise do desempenho do player de 3 canais

Neste capítulo iremos realizar a análise do player de 3 canais proposto pelos autores da MSX Assembly Page - MAP [1], no qual chamaremos aqui de MAP Player.

4.1. Análise gráfica dos erros obtidos em relação à escala

Variando os valores de escala entre 0,001 e 3,000 e com passo de 0,001, foram gerados três gráficos apresentados na figura 3.2. O gráfico em (a) representa toda faixa. A partir do cotovelo da curva, observamos que valores de escala abaixo de 0,25 poderiam ser descartados para uma análise mais detalhada. Assim, traçamos um novo gráfico em (b), para poder visualizar os erros com melhor resolução. Nesse novo gráfico, observamos que os melhores valores estão entre 0,8 e 1,5. Por fim, traçamos um novo gráfico em (c) para melhor visualização. O valor de escala igual 1,3281 apresentou o menor erro (1,16969).

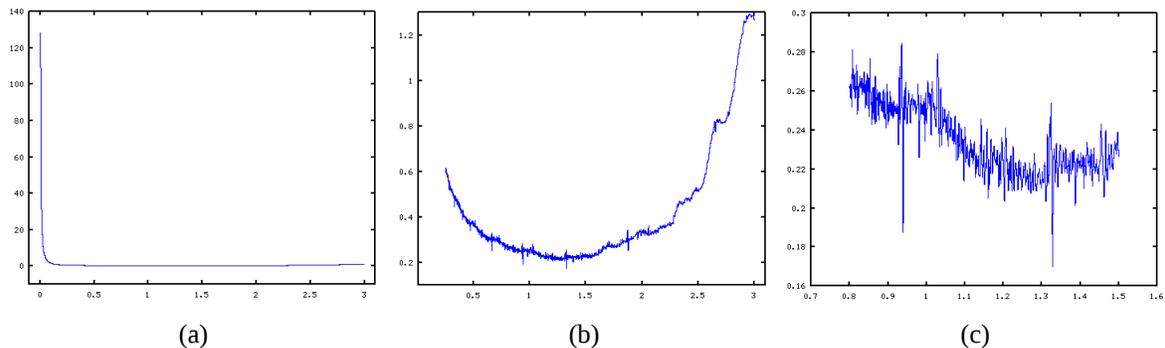


Figura 4.1 – Análise gráfica dos erros obtidos em relação à escala utilizada.

A figura a seguir apresenta a distribuição dos 256 níveis do PCM nas 816 combinações possíveis dos três canais do PSG (círculos azul escuro), com escala igual a 1,3821. Observe que os níveis de PSG praticamente formam uma reta, e exatamente no local antes de começar a curva de distorção acentuada.

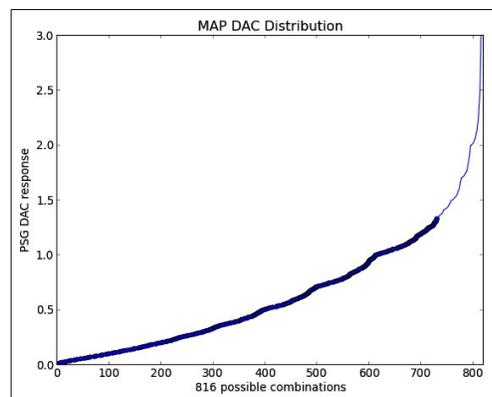


Figura 4.2. - Distribuição dos níveis de PSG.

Caso a escala seja maior que 1,3281, a distribuição de pontos avança para a direita, subindo a curva de distorção (figura 4.2). Caso seja menor, os dados caminham para o lado esquerdo do gráfico. Na figura a seguir são apresentadas as respostas DAC do PSG para cada valor de entrada PCM de 8 bits não sinalizado em diferentes escalas. Observe o que acontece com valores de escala mais altos, quando o PCM é maior que 170.

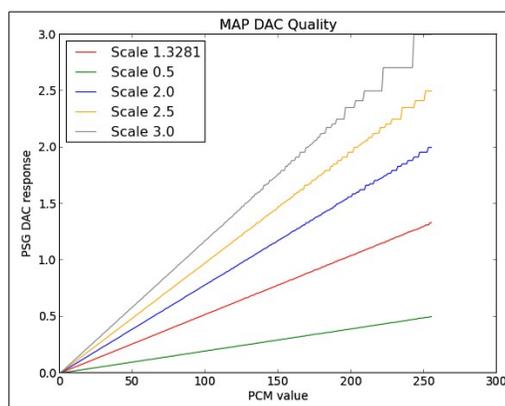


Figura 4.3. - Respostas do PSG para valores em PCM 8 bits em diversas escalas.

Quando aumentamos a escala, o volume também aumenta. Entretanto, há distorção no som em valores PCM mais altos. Porém, em todos os casos apresentados na figura 4.3, a resposta do PSG para valores de PCM foi sempre linear, o que prova que o MAP player cumpre bem o papel de corrigir a curva logarítmica do PSG.

Observe que o valor ótimo de escala 1,3281 (linha vermelha) possui volume bem inferior ao da escala 3.0 (linha cinza). Será que vale a pena sacrificar um som com volume mais alto em relação à máxima qualidade possível? É o que veremos na próxima seção.

4.2. Comparação do MAP com diferentes sample players

Nesta seção iremos comparar a qualidade do MAP player em relação a outros players, como o S4b player (MarMSX) e o PSG Sampler. O S4b player utiliza apenas um canal do PSG, enquanto que o PSG Sampler utiliza os três canais, assim como o MAP player.

A figura 4.4. compara o MAP player (escala 1,3281) com os demais players, quanto à resposta do PSG para valores PCM de 8 bits não sinalizados. Observe que o MAP Player é o único a ter resposta linear.

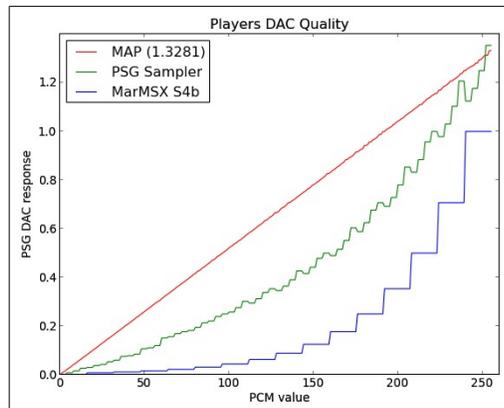


Figura 4.4. - Respostas do PSG para valores em PCM 8 bits de diferentes players.

A seguir, é feita uma comparação entre as quatro escalas do MAP player e os demais players, utilizando-se diferentes critérios. Registra-se que para o PSG Sampler foi feita uma medição de erro com base em uma aproximação poligonal feita na curva de respostas DAC para o PSG Sampler (curva verde na figura 4.4).

Player	Escala	Erro Total	Erro em Escala	Níveis Únicos	Resposta PSG
MAP Player	1,3281	0,2253	0,1696408403	244	Linear
MAP Player	0,5	0,66705	1,3341	221	Linear
MAP Player	2,5	1,27542	0,510168	193	Linear
MAP Player	3	3,78722	1,2624066667	170	Linear
PSG Sampler	1,3536	3,4309	2,5346483452	64	Exponencial
S4b Player	1	6,3228	6,3228	16	Exponencial

Tabela 4.1. Comparação entres diferentes sample players de MSX.

Mesmo com a escala em 3.0, o MAP player é melhor que os outros dois testados, em relação ao erro em escala, níveis únicos e resposta PSG.

A seguir, iremos comparar as ondas sonoras geradas pelo PSG do MSX (emulador openMSX) com o MAP player em escala 3.0 e PSG Sampler para um determinado arquivo de som PCM.

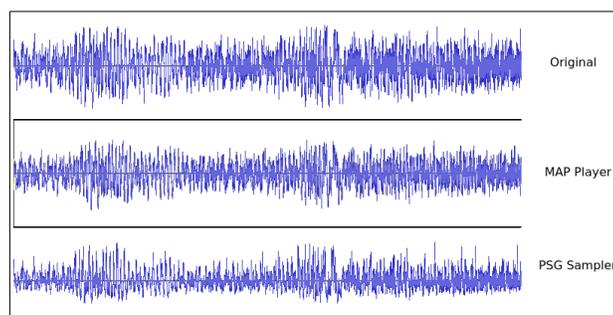


Figura 4.5. - Ondas sonoras geradas pelo Map player e PSG Sampler.

A onda gerada pelo MAP player é bem mais parecida com a onda original do que a do PSG Sampler. Observe que para valores próximos à linha central da onda (valor de PCM igual a 128) ou abaixo dela, as distorções no PSG Sampler são maiores.

Créditos e Bibliografia

Este artigo foi escrito por Marcelo Silveira em Janeiro de 2022, com base no artigo publicado na MSX Assembly Page [1].

E-mail: flamar98@hotmail.com

Homepage: <http://marmsx.msxall.com>

Referências Bibliográficas:

[1] – Playing Samples on the PSG. MSX Assembly Page, em:

http://map.grauw.nl/articles/psg_sample.php (mais atual)

[2] - Digitalização de sons para o MSX . MarMSX Development, artigos em:

<http://marmsx.msxall.com>

[3] – General Instrument AY-3-8910, AY-3-8912, AY-3-8913 datasheet. Encontrado em: MSX Assembly Page - <http://map.grauw.nl/>.