

MSX Article

MARMSX

RGB to Gray

Resumo

Este artigo tem como objetivo mostrar como converter uma imagem colorida para tons de cinza no MSX 2.

1- Introdução

Uma imagem digital é composta de 3 componentes de cores primárias no sistema aditivo: vermelho (red), verde (green) e azul (blue). Esse sistema de cores é chamado de sistema RGB. Não confunda esse sistema com o sistema de cores subtrativo, que é aquele que é ensinado em escolas e o utilizado por impressoras, no qual é formado pelas cores primárias: ciano, amarelo e magenta. O primeiro é baseado na emissão de luz, enquanto que o segundo na reflexão da luz. A figura 1 apresenta os dois sistemas.



Figura 1. Sistemas de cores.

Cada cor primária do sistema aditivo varia sua intensidade do mais escuro para o mais claro. Os valores das intensidades das cores na imagem digital é composta por valores discretos, formando níveis de intensidade. A quantidade de níveis de intensidade depende do sistema de vídeo utilizado. No caso dos PCs, as placas de vídeo possuem 256 níveis por componente de cor, totalizando 16 milhões de cores. Já o MSX 2, o processador de vídeo possui 8 níveis por componente de cor, totalizando 512 cores.

Uma imagem em tons de cinza é aquela em que as componentes de cores primárias em cada pixel possuem a mesma intensidade. Dessa forma, podemos representar a cor através de uma componente apenas, fazendo-se uma média entre as todas as componentes de cor.

A intensidade de cinza é calculada utilizando-se a seguinte fórmula:

$$\text{GRAY} = (\text{RED} + \text{GREEN} + \text{BLUE}) / 3$$

Porém, existe uma outra fórmula que é utilizada em processamento de imagens, que leva em consideração a capacidade de absorção do olho humano da luz emitida por cada componente de cor. É uma média ponderada das componentes RGB.

$$\text{GRAY} = \text{RED} * 0,3 + \text{GREEN} * 0,59 + \text{BLUE} * 0,11$$

Podemos converter uma imagem colorida em uma imagem em tons de cinza, uma vez que está é um subconjunto do sistema colorido. O caminho inverso é complexo, uma vez que não há, *a priori*, informações quanto às cores originais.

2- Conversão para tons de cinza no MSX

2.1- Screens 2-7

As screens gráficas de 2 a 7 do MSX 2, possuem o sistema de cores indexado, onde cada pixel não possui diretamente o valor RGB da cor, em sim um índice para uma tabela que contém esse valor. Essa tabela é chamada de paleta de cores.

Para converter uma imagem colorida para tons de cinza nesses modos de tela, é necessário aplicar a transformação nessa paleta de cores.

O MSX 2 possui o comando COLOR=, que permite alterar as cores da paleta. Entretanto, não há um comando específico para ler informações dessa paleta. Assim, é necessário ir buscar a informação da paleta de cores diretamente na VRAM.

A área da VRAM que contém os dados da paleta de cores varia em cada modo de tela. A tabela 1 apresenta a faixa de endereços da paleta de cores para cada modo de tela.

Screen	Endereço inicial	Endereço final
0 / width 40	&H0400	&H041F
0 / width 80	&H0F00	&H0F1F
1	&H2020	&H203F
2	&H1B80	&H1B9F
3	&H2020	&H203F
4	&H1B80	&H1B9F
5	&H7680	&H769F
6	&H7680	&H769F
7	&HFA80	&HFA9F

Tabela 1. Endereço da paleta na VRAM.

Cada índice da paleta contém 2 bytes de informação, configurados da seguinte forma:

```
E - 0rrr0bbb  
E+1 - 00000ggg
```

O endereço para cada índice da tabela é calculado da seguinte forma:

```
E = endereco_inicial + índice x 2
```

O programa em Basic a seguir, altera os valores de qualquer paleta de cores para tons de cinza na screen 5. Desenhe qualquer coisa nessa tela, ou carregue uma imagem para ver os resultados.

```

10 SCREEN 5
20 FOR E=&H7680 TO &H769F STEP 2
30 R = FIX(VPEEK(E)/16)
40 G = VPEEK(E+1)
50 B = VPEEK(E) AND 7
60 C = FIX(R*0.3 + G*0.59 + B*0.11)
70 VPOKE E, C*16 + C
80 VPOKE E+1,C
90 NEXT E
100 COLOR=RESTORE
110 GOTO 110

```



Para os outros modos de tela, alterar a faixa de valores na linha 20.

2.2- Screen 8

A screen 8 é um modo de tela em que o pixel é representado diretamente pelo sistema de cores RGB. Como cada componente de cor no MSX 2 possui 3 bits ($2^3 = 8$ níveis), seriam necessários 9 bits para representar um pixel. Como a unidade de memória é o byte, e cada byte possui 8 bits, decidiu-se eliminar um bit da componente de cor azul para o encaixe, uma vez que essa cor é a menos percebida pelo sistema visual humano.

Cada pixel da screen 8 possui a seguinte configuração em bits:

Bit	7	6	5	4	3	2	1	0
Cor	G	G	G	R	R	R	B	B

Serão apresentados dois algoritmos para a conversão de uma imagem colorida na screen 8 para tons de cinza. Um em Basic e outro em Assembly. O programa em Basic leva bastante tempo para rodar, enquanto que o programa em Assembly leva em torno de 1 minuto. O trabalho de cada programa é separar as componentes de cor para cada ponto, converter o valor de azul para 3 bits, calcular a média e finalmente fazer arranjo dos bits de volta. A média aritmética foi escolhida pela maior simplicidade nos cálculos.

```

10 SCREEN 8
20 BLOAD"imagem.pic",S
30 FOR Y=0 TO 211
40 FOR X=0 TO 255
50 C=POINT(X,Y)
60 B=(C AND &B00000011)*2
70 R=(C AND &B00011100)/4
80 G=(C AND &B11100000)/32
90 C=INT((R+G+B)/3)
100 B=C/2
110 R=C*4
120 G=C*32
130 C=R+G+B
140 PSET(X,Y),C
150 NEXT X,Y
160 GOTO 160

```

Programa em Assembly correspondente:

```
ORG &HC000
LD D,&HD4          ; Fim VRAM Screen 8
LD E,0            ;
LD HL,0           ; Inicio VRAM Sc8
s1: LD IX,&H10D    ; RDVRM (Lê VRAM) Cor em A
CALL &H15F        ; Call subrom
PUSH DE           ; Guarda DE
LD D,0            ; Zera contador da média
LD E,A            ; Salva cor
AND &B00000011    ; Separa o azul
SLA A             ; Converte 2 bits para 3 bits (por causa do R e G)
LD D,A            ; Acumula
LD A,E            ; Recupera cor
AND &B00011100    ; Separa o vermelho
SRL A             ; Passa de 000RRR00 para
SRL A             ; 00000RRR
ADD A,D           ; Soma vermelho com azul
LD D,A            ; Guarda em D
LD A,E            ; Recupera a cor
AND &B11100000    ; Separa o verde
LD B,5            ;
e1: SRL A         ; Faz 00000YYY (Y = GRAY)
DJNZ e1           ; Loop 5x
ADD A,D           ; Soma: Agora temos a soma das 3 cores
LD D,FF          ; Zera contador (- 1, por causa do algoritmo)
LD B,3            ; Média por 3 elementos
e2: SUB B         ; A = A - B
INC D             ; D = D + 1 (D é o nível de cinza!)
JR NC,e2         ; Se não negativo, continue
LD A,D           ;
SLA A             ; Faz 00YYY000
SLA A             ;
SLA A             ;
ADD A,D           ; Faz 00YYYYYY
SLA A             ;
SLA A             ; Faz YYYYYY00
SRL D             ; Converte Gray em 2 bits
ADD A,D           ; Finalmente YYYYYYYY
LD IX,&H109       ; WRTVRM (Escreve nova cor)
CALL &H15F        ; Chama subrom
INC HL           ; Proximo ponto
POP DE           ; Retorna o valor de DE para fim de memória
LD A,D           ;
CP H             ;
JR NZ,s1         ; Verifica se chegou ao final da memória
LD A,E           ;
CP L             ;
JR NZ,s1         ;
RET              ; Retorna
```

3- Créditos

Artigo escrito por Marcelo Teixeira Silveira, originalmente para a Fazine do encontro da MSX Rio de 2008.

Escrito em: 18 de março de 2008.

Revisado em: julho de 2017.

E-mail: flamar98@hotmail.com

Homepage: marmsx.mxsall.com