MSX Article

Cores do MSX 2+

Resumo

O objetivo deste artigo é demonstrar como se chegar ao cálculo do total de cores das screens 10, 11 e 12 do MSX 2+.

1- Introdução

O MSX 2+ possui um pardão para representar as cores chamado YJK. Esse padrão trabalha com pixels agrupados na forma 4x1 (4 colunas e uma linha), onde cada grupo possui a seguinte configuração em bits:

Bit	7	6	5	4	3	2	1	0
Pixel 0	Y4	Y3	Y2	Y1	Y0	K2	K1	K0
Pixel 1	Y4	Y3	Y2	Y1	Y0	K5	K4	K3
Pixel 2	Y4	Y3	Y2	Y1	Y0	J2	J1	J0
Pixel 3	Y4	Y3	Y2	Y1	Y0	J5	J4	J3

Tabela 1. Sistema YJK

Os componentes K e J controlam a cor do grupo, variando de -32 a 31 cada componente e totalizando 4096 combinações diferentes (2^{12}). Quando positivos, o J corresponde à componente de cor vermelha e o K ao verde. Quando ambos são negativos, formam o azul. A combinação de cores de J e K pode ser vista na figura 1.

O elemento Y indica o brilho de cada pixel do grupo. Varia de 0 a 31, possuindo assim 32 intensidades.

A combinação da componente Y de cada pixel com as componentes JK do grupo formam um número de 17 bits, resultando em uma combinação de 131072 valores (2^{17}).

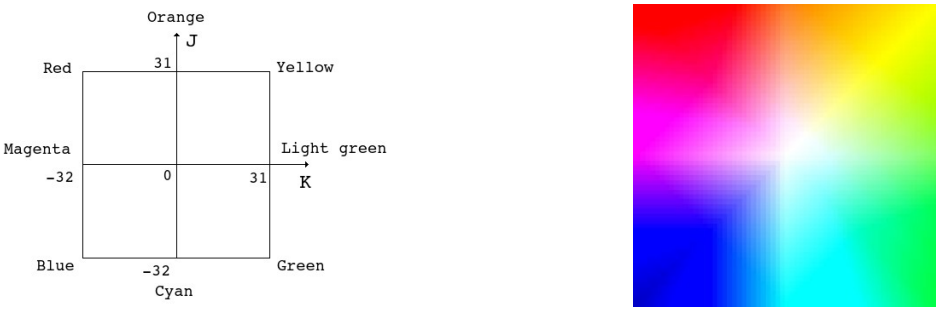


Figura 1. A distribuição de cores JK.

De acordo com o manual do processador gráfico v9958 [1], os valores YJK armazenados na VRAM são convertidos para o sistema RGB, para que os pixels sejam exibidos na tela. Então, cada componente RGB gerado nessa conversão possui 5 bits de dados. Desse modo, seria possível gerar até 32768 (2^{15}) cores distintas. Entretanto, por quê a screen 12 somente é capaz de gerar 19268 cores? E por quê as screens 10 e 11 somente 12499?

2- Análise das screens 10, 11 e 12

A diferença entre as screens 10, 11 e 12 é que as screens 10 e 11 possuem um bit seletor de modo RGB / YJK. Quando esse bit possui valor igual a 0, o pixel se comporta como YJK. Quando o bit possui valor igual a 1, o pixel se comporta como RGB, igual às screens 2, 5 e 7.

Conforme pode ser visto na tabela 2, o bit número 3 seleciona o modo de cada pixel. Se A=0, a cor é formada por Y0~Y3 + J + K. Entretanto, se A=1, a cor é formada pelos bits Y0~Y3, variando de 0 a 15, correspondendo a um valor da paleta de cores do MSX 2.

Bit	7	6	5	4	3	2	1	0
Pixel 0	Y3	Y2	Y1	Y0	A	K2	K1	K0
Pixel 1	Y3	Y2	Y1	Y0	A	K5	K4	K3
Pixel 2	Y3	Y2	Y1	Y0	A	J2	J1	J0
Pixel 3	Y3	Y2	Y1	Y0	A	J5	J4	J3

Tabela 2. Screens 10 e 11.

Já a screen 12 utiliza mais um bit para a componente Y, não possuindo a escolha entre sistemas, conforme pode ser visto na tabela 3.

Bit	7	6	5	4	3	2	1	0
Pixel 0	Y4	Y3	Y2	Y1	Y0	K2	K1	K0
Pixel 1	Y4	Y3	Y2	Y1	Y0	K5	K4	K3
Pixel 2	Y4	Y3	Y2	Y1	Y0	J2	J1	J0
Pixel 3	Y4	Y3	Y2	Y1	Y0	J5	J4	J3

Tabela 3. Screen 12.

O circuito conversor do v9958 [1] pode ser visto na figura 2, bem como as fórmulas de conversão entre os sistemas YJK e RGB na tabela 4.

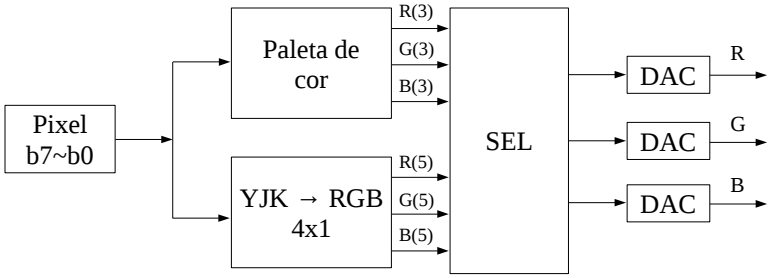


Figura 2. O circuito conversor do MSX 2+.

YJK → RGB	RGB → YJK
R = Y + J G = Y + K B = 5/4*Y – J/2 – K/4	Y = B/2 + R/4 + G/8 J = R – Y K = G – Y

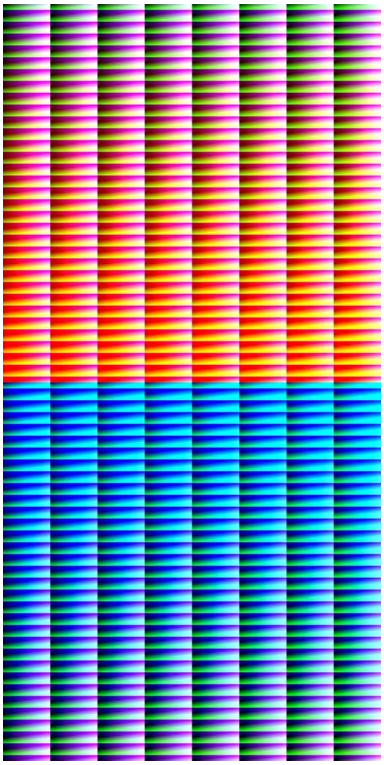
Tabela 4. Conversão entre os sistemas RGB e YJK [1].

Os valores de Y, R, G e B variam de 0 a 31. Já J e K variam de -32 a 31. Dessa forma, a conversão entre sistemas deverá garantir que essas faixas de valores estão sendo respeitadas.

Observe que J e K são armazenados na VRAM com valores entre 0 e 63. Assim, devemos fazer o seguinte ajuste para calcular o valor RGB:

```
IF K>31 THEN K=K-64
IF J>31 THEN J=J-64
```

A figura 3 apresenta todas as combinações YJK possíveis, bem como o programa em Basic para gerar essa imagem.



```
10 SCREEN 12
20 Y=0 : J=0 : K=0
30 FOR E=0 TO 256*212-1 STEP 4
40 VPOKE E,Y*8 + (K AND 7)
50 VPOKE E+1,(Y+1)*8 + FIX(K / 8)
60 VPOKE E+2,(Y+2)*8 + (J AND 7)
70 VPOKE E+3,(Y+3)*8 + FIX(J / 8)
80 Y=Y+4
90 IF Y>31 THEN K=K+1 : Y=0
100 IF K>63 THEN J=J+1 : K=0
110 NEXT E
```

Programa tem que ser rodado em 3 telas e depois deve-se juntar as telas verticalmente.

Tela 1:
```
20 Y=0 : J=0 : K=0
```

Tela 2:
```
20 Y=0 : J=26 : K=32
```

Tela 3:
```
20 Y=0 : J=53 : K=0
```

Figura 3. Todas as combinações YJK possíveis geradas na tela do MSX 2+

3- Descobrindo as combinações possíveis de cores RGB

Conforme visto na seção anterior, o YJK é capaz de gerar 131072 combinações para cada pixel. Entretanto, a saída RGB é capaz de gerar apenas 32768 cores. Além disso, a conversão entre YJK e RGB é feita através de uma fómula apresentada na tabela 4.

O problema está na fórmula de conversão de YJK para RGB, que não é capaz de gerar todas as 32768 combinações de cores. Além disso, as screens 10 e 11 possuem 1 bit a menos para gerar as cores.

O programa em C a seguir [2] irá gerar todas as combinações possíveis de YJK, convertendo cada uma para o sistema RGB de 15 bits (5 bits para cada componente) e

armazenando o resultado em uma matriz. Como são geradas muitas vezes a mesma cor RGB, deve-se apenas considerar o número de cores diferentes que o YJK consegue gerar.

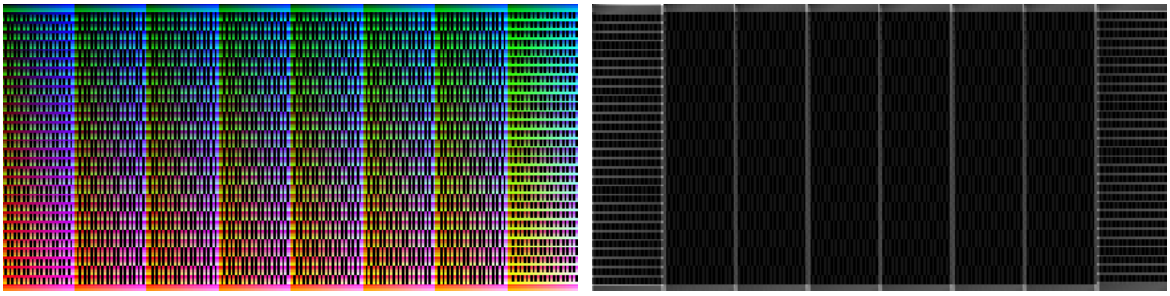
```
#include <stdio.h>
#include <math.h>

int RGB[32][32][32];

void yjk2rgb(int Y, int J, int K, int *R, int *G, int *B)
{
  *R = Y + J;
  *G = Y + K;
  *B = floor(5*Y/4.0 - J/2.0 - K/4.0);

  if (*R<0) *R = 0;
  if (*G<0) *G = 0;
  if (*B<0) *B = 0;

  if (*R>31) *R = 31;
  if (*G>31) *G = 31;
  if (*B>31) *B = 31;
}

void count_colors()
{
  int R, G, B, RGB_count=0;

  for (R=0; R<=31; R++)
  {
    for (G=0; G<=31; G++)
    {
      for (B=0; B<=31; B++)
        (RGB[R][G][B] != 0) ? RGB_count++ : RGB_count;
    }
  }

  printf("Total RGB colors: %d\n", RGB_count);
}


void calculate_YJK(int scr)
{
  int R, G, B, Y, J, K;
  int YJK_count=0;

  for (Y=0; Y<=31; Y++)
  {
    for (J=-32; J<=31; J++)
    {
      for (K=-32; K<=31; K++)
      {
        if (scr != 12 && (Y&1 == 1))
          continue;

        YJK_count++;

        yjk2rgb(Y, J, K, &R, &G, &B);
        RGB[R][G][B]++;
      }
    }
  }

  printf("Total YJK combinations: %d\n", YJK_count);
  count_colors();
}

main(void)
{
  printf("MSX 2+ - screens 10 and 11:\n");
  calculate_YJK(10);
  printf("\nMSX 2+ - screen 12:\n");
  calculate_YJK(12);
}
```

Saída:

MSX 2+ - screens 10 and 11:
Total YJK combinations: 65536
Total RGB colors: 12499

MSX 2+ - screen 12:
Total YJK combinations: 131072
Total RGB colors: 19268

A figura 4 apresenta todas as 19268 cores RGB geradas a partir do sistema YJK. Os “buracos” em preto são as cores não atingidas pela fórmula de conversão do YJK, no qual correspondem a 41% das cores RGB possíveis com 15 bits. O histograma da figura 4b indica a maior ou menor ocorrência das cores RGB geradas, em tons de cinza. Quanto mais claro for, mais freqüênte é.



a) Cores RGB geradas a partir do YJK. b) Histograma RGB resultante.

Figura 4. Cores RGB possíveis através da conversão YJK → RGB.

4- Créditos e bibliografia

O artigo foi escrito por Marcelo Silveira, Engenheiro de Sistemas e Computação, formado pela UERJ em 2002 e revisado em Julho de 2017.

Home page: marmsx.msxall.com
Email: flamar98@hotmail.com

Referências:

[1] – Yamaha v9958 MSX-Video Technical Data Book, 1988.
[2] – MSX2PCOL, Projeto Tools, MarMSX em http://marmsx.msxall.com