

# MSX Article

**MARMSX**

*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.

## 1- Introdução

O MSX 2+ possui um padrão novo de cores, chamado de YJK. O YJK trabalha em grupos de 4 pixels, da seguinte forma:

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

Os elementos K e J são relativos aos componentes RGB, totalizando 4096 combinações diferentes ( $2^{12}$ ). O valor J-K vale para todo o grupo de 4 pixels.

O elemento Y indica a intensidade individual do pixel, junto com o valor J-K do grupo.

Cada pixel então, pode controlar 131072 combinações de cores distintas ( $2^{17}$ )!

Porém, cada componente do RGB têm 5 bits, totalizando 15 bits. Então, poderíamos ter no máximo 32768 cores ( $2^{15}$ ).

Mas a screen 12 do MSX 2+ só tem 19268 cores. Cadê o resto?

## 2- Descobrimo as cores do MSX 2+

O chip de vídeo 9958, da Yamaha, do MSX 2+ tem um circuito que converte YJK para RGB, onde este é o padrão de saída para o vídeo (TV), que tem a seguinte fórmula de conversão:

$$R = Y + J$$

$$G = Y + K$$

$$B = 5/4*Y - J/2 - K/4$$

J e K são números de 6 bits sinalizados e y é um número de 5 bits não sinalizado.

Para inserir sinal em J e K, fazemos:

$$\text{IF } K > 31 \text{ THEN } K = K - 64$$

$$\text{IF } J > 31 \text{ THEN } J = J - 64$$

O resultado de r, g e b devem estar dentro da faixa de valores 0 a 31. Portanto:

```
IF R<0 THEN R=0
IF G<0 THEN G=0
IF B<0 THEN B=0
IF R>31 THEN R=31
IF G>31 THEN G=31
IF B>31 THEN B=31
```

Com esta fórmula de conversão e mais estas restrições, obviamente, teremos várias cores repetidas.

A figura 2.1 mostra um espectro gerado, com as 131072 combinações YJK possíveis. Contando-se as cores diferentes na imagem, temos 19268 cores!

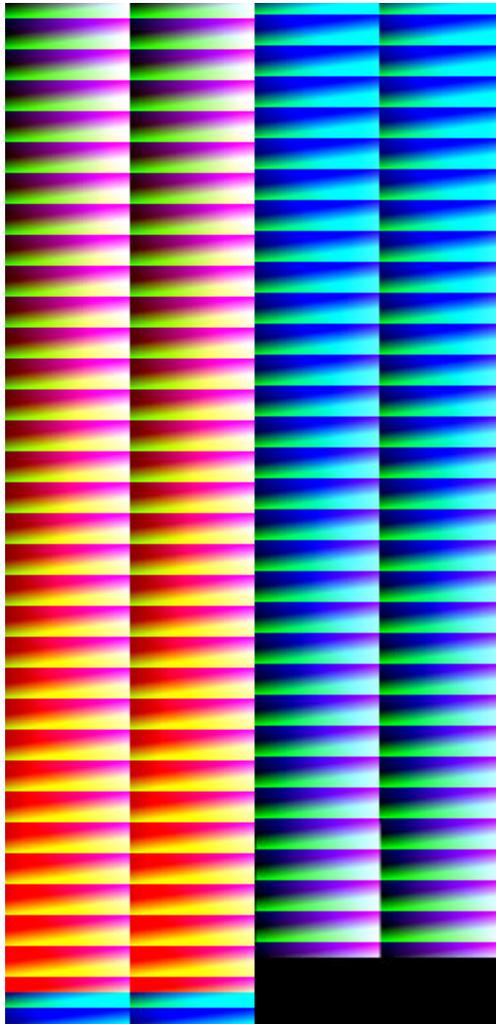


Fig 2.1 - Espectro da Screen 12.

Podemos criar um programa que armazena em uma tabela as cores inéditas, para eliminar as repetições.

Screen 12:

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

unsigned char cor[20000][3];
int c=0, m=0, j=0, k=0, y=0, j1, k1;
int r,g,b,f,flag;
long int cont=0;

void main(void)
{
    clrscr();
    printf("Contando as cores da Screen 12 do MSX 2+");
    while ((j<64) && (c<20000))
    {
        gotoxy(1,3);
        printf("Posiç.Æo: %ld",cont);
        cont++;
        j1=j;
        if (j1>31)
            j1=j1-64;
        k1=k;
        if (k1>31)
            k1=k1-64;
        r=y+j1;
        g=y+k1;
        b=(5*y-2*j1-k1)/4;
        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;
        flag=0;
        for (f=0; f<c; f++)
        {
```

```

if ((cor[f][0]==r) && (cor[f][1]==g) && (cor[f][2]==b))
{
flag=1;
break;
}
}
if (flag==0)
{
gotoxy(1,2);
printf("Cores: %d\n",c+1);
cor[c][0]=r;
cor[c][1]=g;
cor[c][2]=b;
c++;
}
m++;
y++;
if (y>31)
{
y=0;
k++;
}
if (k>63)
{
k=0;
j++;
}
}
}
}

```

O resultado do programa pode ser conferido na fig 2.2.

```

Contando as cores da Screen 12 do MSX 2+
Cores: 19268
Posição: 131071

```

Fig 2.2- Saída do programa para screen 12.

E a screen 10 e 11? Por quê só tem 12499 cores?

Estas screens possuem em Y0 um bit de seleção entre os sistemas YJK ou RGB. Se Y0 for 0, é YJK. Se y=1, então é RGB (16 cores).

Com isso, Y tem somente 4 bits. Na verdade, usamos os 5, mas Y0 é sempre 0. Com isso, o passo de Y vai de 2 em 2.

Screens 10 e 11:

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

unsigned char cor[20000][3];
int c=0, m=0, j=0, k=0, y=0, j1, k1;
int r,g,b,f,flag;
long int cont=0;

void main(void)
{
  clrscr();
  printf("Contando as cores da Screen 10 e 11 do MSX 2+");
  while ((j<64) && (c<20000))
  {
    gotoxy(1,3);
    printf("Posiçãõ: %ld",cont);
    cont++;
    j1=j;
    if (j1>31)
      j1=j1-64;
    k1=k;
    if (k1>31)
      k1=k1-64;
    r=y+j1;
    g=y+k1;
    b=(5*y-2*j1-k1)/4;
    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;
```

```

flag=0;
for (f=0; f<c; f++)
{
    if ((cor[f][0]==r) && (cor[f][1]==g) && (cor[f][2]==b))
    {
        flag=1;
        break;
    }
}
if (flag==0)
{
    gotoxy(1,2);
    printf("Cores: %d\n",c+1);
    cor[c][0]=r;
    cor[c][1]=g;
    cor[c][2]=b;
    c++;
}
m++;
y=y+2;
if (y>31)
{
    y=0;
    k++;
}
if (k>63)
{
    k=0;
    j++;
}
}
}

```

A diferença entre os programas está na linha em que y++ passa a ser y=y+2.  
A saída do programa pode ser vista na figura 2.3.

```
Contando as cores da Screen 10 e 11 do MSX 2+  
Cores: 12499  
Posição: 65535
```

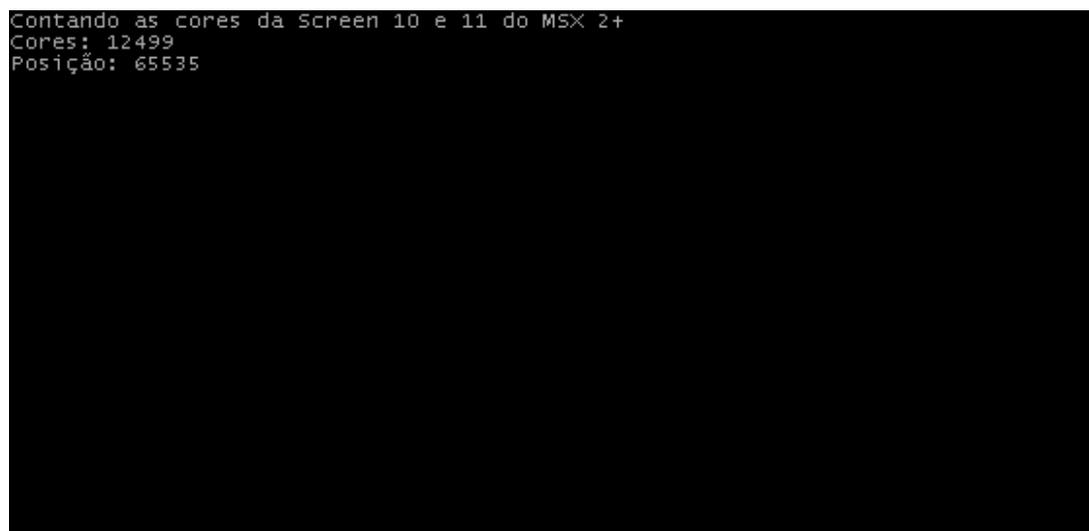


Fig 2.3 - Saída do programa das screens 10 e 11.

### **3- Créditos e bibliografia**

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

Dados de conversão foi retirado do manual do chip v9958 da Yamaha.

Home page: [www.marmsx.cjb.net](http://www.marmsx.cjb.net)

Email: [flamar98@hotmail.com](mailto:flamar98@hotmail.com)