MSX Article

MARMSX

MSX 2+ Colors

Abstract

The goal of this article is to show why MSX 2+ has 19268 colors at screen 12 mode and 12499 colors at screens 10 and 11 modes.

1- Introduction

MSX 2+ has a new color pattern called YJK. YJK works on 4 pixels groups, as shown on the following table:

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

K and J elements corresponds to RGB factor, totalizing 4096 different color combinations (2^{12}). Y-K value is set for all 4 pixels group.

Y element indicates individual pixel intensity, together with J-K group value.

So, each pixel can manipulate 131072 different colors (2^{17}).

Ohterwise, each RGB component manipulates 5 bits (R, G and B), totalizing 15 bits. So, we can have only 32768 (2^{15}) distinct colors.

But ... screen 12 has 19268 colors. Where are the other colors?

2- Finding out MSX 2+ colors

Yamaha v9958 video chip has a built in circuit that converts YJK to RGB, where is the video out pattern (TV), which has the following conversion formula:

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

J and K are 6 bits signed numbers each, and Y is a 5 bit unsigned number.

To insert signal on J or K, we have:

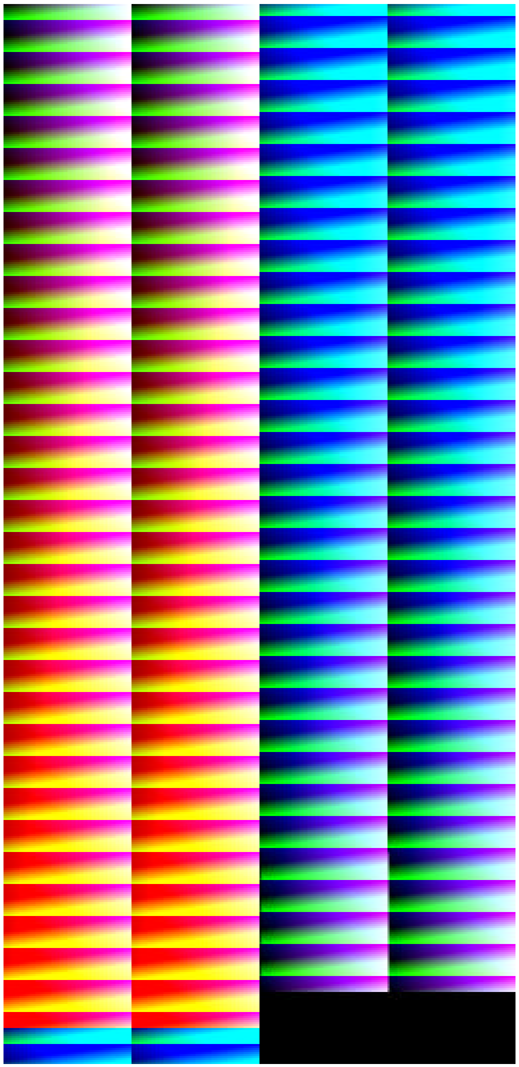
IF K>31 THEN K=K-64
IF J>31 THEN J=J-64

R, G and B results must be inside of 0 to 31 value range. So, we have:

```
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
```

According to the conversion formula, plus these restrictions, for sure we will have many repeated colors.

Picture 2.1 shows a spectrum generated with all 131072 YJK possible combinations. By counting different colors, we have exactly 19268 colors.



Picture 2.1 - Screen 12 spectrum.

We can create a program who stores new colors in a table, in order to eliminate repeated colors.

Screen 12:

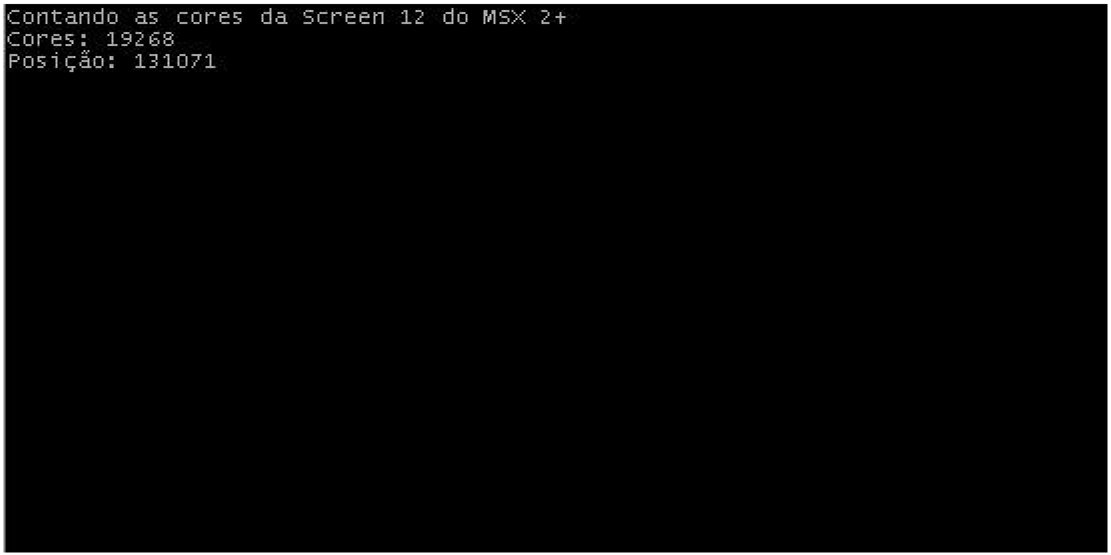
```
#include <stdio.h>
#include <conio.h>

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("Counting  MSX 2+ Screen 12 colors ...");
  while ((j<64) && (c<20000))
  {
    gotoxy(1,3);
    printf("Position: %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("Colors: %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++;
     }
   }
 }
```

Program results can be seen on Picture 2.2.



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

Picture 2.2 - Screen 12 program results.

And what about screens 10 and 11? Why they have only 12499 colors?

Y0 bit in these screens works as YJK and RGB mode selection. If Y0 worths 0, this pixel works as YJK system. If Y0 worths 1, works as RGB (only 16 colors).

So, Y has only 4 bits to control intensity. In fact, we use all 5 bits, but, to work as YJK, Y0 must ever be 0, where we can conclude that Y steps from 2 to 2.

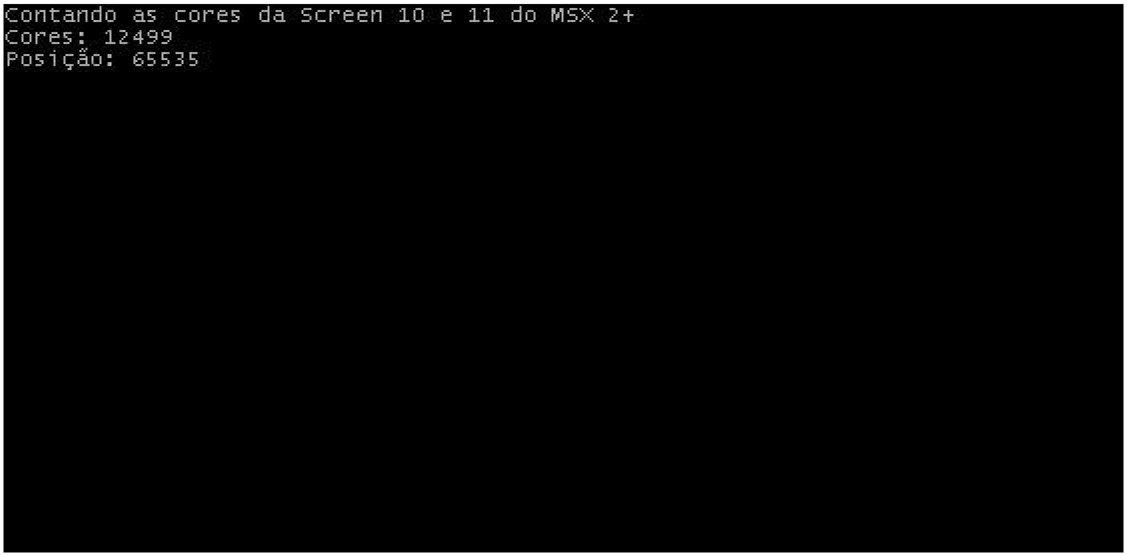
Screens 10 and 11:

```
#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("Counting MSX 2+ Screens 10 and 11 colors ... ");
  while ((j<64) && (c<20000))
  {
    gotoxy(1,3);
    printf("Position: %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("Colors: %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++;
    }
  }
}
```

This program output can be seen on Picture 2.3.



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

Picture 2.3 - Screens 10 and 11 pogram results.

3- Credits and references

This article was written by Marcelo Silveira, Systems and Computing Engeneer, graduated on UERJ, in 2002.

The theory and programs to count MSX programs were made by Marcelo Silveira.

Convertion data were took from Yamaha v9958 chip manual.

Home page: www.marmsx.cjb.net
Email: flamar98@hotmail.com