MSX Article

MARMSX

MSX 2+ Colors

Summary

This article presents the MSX 2+ YJK color system and describes how to calculate screens 10, 11 and 12 total colors.

1- Introduction

The MSX 2+ makes uses of a color system called YJK to store screens 10, 11 and 12 data. This system clusters pixels on a 4x1 pattern (4 columns and 1 row), where each cluster has the following configuration in bits:

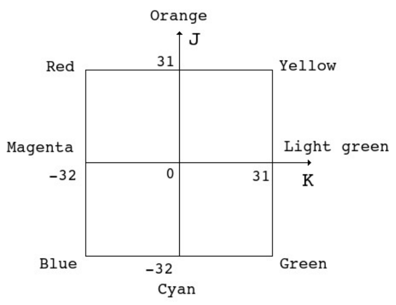
Bit	7	6	5	4	3	2	1	0
Pixel 0	Y4	Y3	Y2	Y1	Y0	K2	K1	K0
Pixel 1	Y4	Y3	Y2	Y1	Y0	K5	K4	K3
Pixel 2	Y4	Y3	Y2	Y1	Y0	J2	J1	J0
Pixel 3	Y4	Y3	Y2	Y1	Y0	J5	J4	J3

Table 1. YJK System.

The K and J components control the color of the whole group, ranging from -32 to 31 each one and resulting on 4096 possible combinations (2^{12}). When the value is positive, the J corresponds to the red color and the K corresponds to the green. When both are negative, they control the blue color. The J and K combinations are seen on figure 1.

The Y component controls individually the brightness of each pixel, ranging from 0 to 31. In this case, each pixel may have 32 different brightness intensities.

The combination of each pixel Y component with the group JK components compose a 17-bit number, resulting on 131072 possible values (2^{17}).



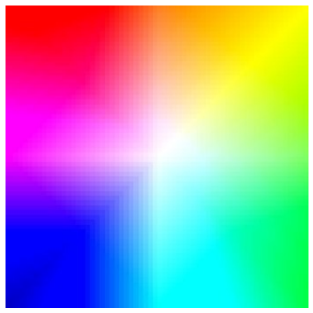


Figure 1. JK color space.

According to the v9958 graphic processor manual [1], the YJK values stored in the VRAM are converted to the RGB system in order to display the pixels. Each color component R,G,B resulted from this conversion has 5 bits, which results on 32768 (2^{15}) distinct colors. Then, why screen 12 is able to display only 19268 colors? And why screens 10 and 11 display only 12499 colors?

2- MSX 2+ screens 10, 11 e 12

The difference between screens 10, 11 and 12 lies on the fact that screens 10 and 11 use one bit from each pixel to select between RGB and YJK color system. When this pixel is valued 0, the pixel works as YJK. Otherwise, when this pixel is valued 1, the pixel works as RGB in the same way of screens 2, 5 and 7.

As seen on table 2, the bit 3 selects the color mode from each pixel. If A=0, the color is formed by Y0~Y3 + J + K. In the other hand, if A=1, the color is formed by Y0~Y3 bits, ranging from 0 to 15, and corresponding to a color from the MSX 2 palette system.

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

Table 2. Screens 10 and 11.

The screen 12 uses bit 3 as a part of Y component. In that case, this screen is not able to select the color system mode. Table 3 shows the screen 12 color configuration.

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

Table 3. Screen 12.

The v9958 conversion circuit [1] is seen on figure 2, as well as the conversion formulas between YJK and RGB systems on table 4.

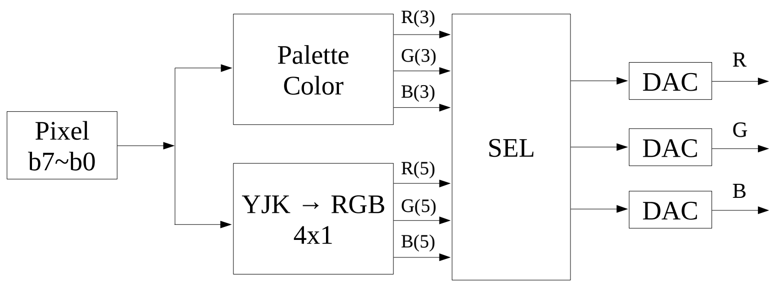


Figure 2. The MSX 2+ color conversion circuit.

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

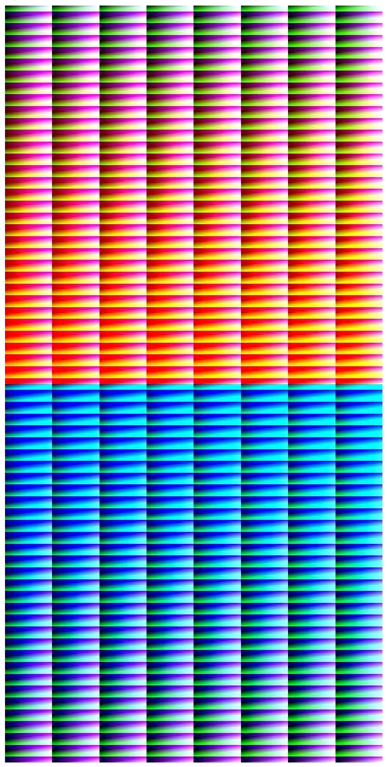
Table 4. Conversion between RGB and YJK color systems [1].

The Y, R, G and B components range from 0 to 31. In the other hand, J and K range from -32 to 31. So, we must keep in mind that the conversion formula will not violate the JK range values, once they are stored in the VRAM using the values between 0 and 63.

Before calculating the RGB values, we have to adjust the JK values using the following formula:

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

Figure 3 shows all possible YJK color combinations and also the Basic program used to create the image.



```
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
```

This program must be run using 3 screens and then concatenating vertically the screens.

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

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

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

Figure 3. All possible MSX 2+ YJK combinations.

3- Finding out the possible RGB colors combinations

As seen on the previous section, the YJK is able to generate up to 131072 combinations for each pixel and the RGB color output is able to generate only 32768 colors. Thus, the conversion between YJK and RGB is performed using a formula shown on table 4.

The problem lies on that conversion formula, which cannot generate all the 32768 possible colors. The screen 10 and 11 also loose one bit, dedicated to select the color mode.

The following C program [2] will create all YJK combinations, converting each one to the 15-bit RGB system (5 bit for each component) and store the result in a RGB histogram matrix. Once the same color is resulted many times, we must consider only the different colors that YJK system can generate.

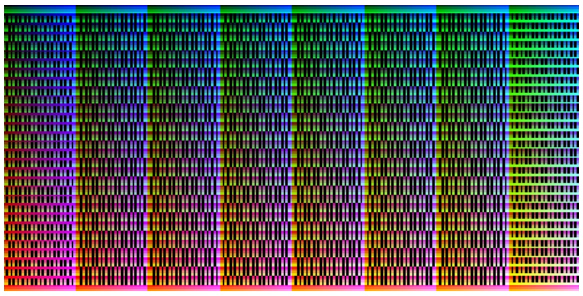
```
#include <stdio.h>
#include <math.h>

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);
}
```

Output:

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

Figure 4a display all the 19268 colors generated from YJK color system. The black "holes" are the missing colors that could not be generated using YJK to RGB conversion formula, which correspond to 41% from 32768 possible colors. The histogram on figure 4b shows the frequency of RGB colors generated. The most white is, the most frequent is.


a) RGB colors generated from YJK.

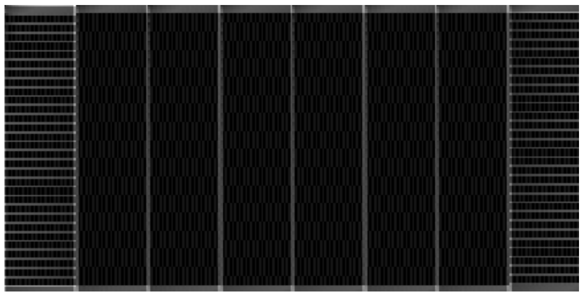

b) Resulting RGB histogram.

Figure 4. Possible RGB colors through YJK → RGB conversion.

4- Credits and References

This article was written originally in portuguese and translated by Marcelo Silveira, Computer Engineer, in 2002 e revised on July 2017.

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

References:

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