

在调色板内有 256 色。在 16 色方式，它可以分割成 16 个储存空间；在 64 色方式下，它可以分割成 4 个存储空间。

背景:

以图片 Color16.bmp 为例，在 16 色与 8*8 字符大小的方式下转换，导出部分 PALETTE、PGT 和 PNT 数据:

PALETTE 数据: 根据字符方式，图片分成数量和大小不同的字符块。按字符块顺序读取颜色，保存第一次出现的颜色值。如果当前调色板储存空间（如 0 号）已经被其他图片占用，就需要改变颜色的储存空间。不然将无法正确显示图片。

\$00: 颜色从0号储存空间开始储存, \$10: 调色板内有16个颜色值;

```

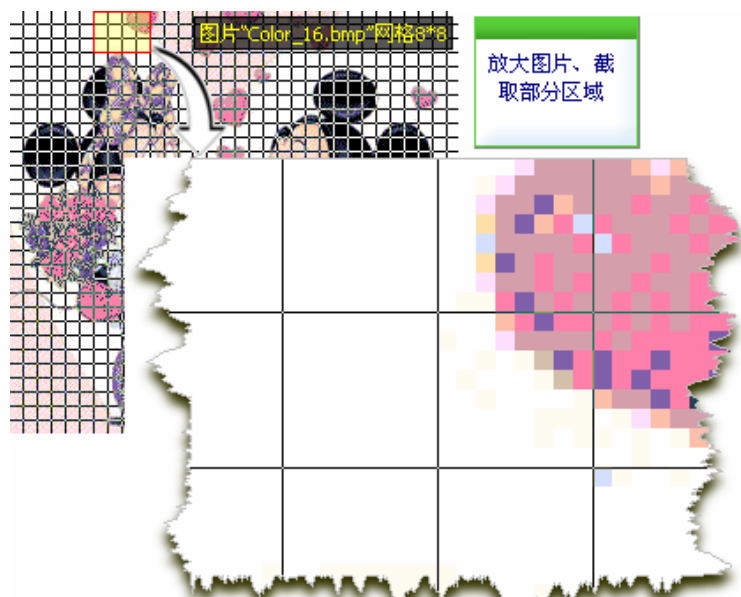
COLOR16_PNT_PALETTE:
db  $00,$10
dw  $7FFF,$7F7F,$6A75,$7FFE,$3D75,$7EF5,$7F75,$7DF5,$6B7F,$6AF5,$14EA,$6378,$146A,$0000,$14F5
    
```

图片PAL数据

PNT 数据: 保存图片的字符定义，每个字符需要两个字节来定义。两个字节包括 12bits 的字符序号和 4bits 的调色板储存空间参数。字符序号为 0 表示透明，则显示背景颜色。图片根据字符定义显示相应的字符块，就可以显示图片。

图片"Color_16.bmp"网格8*8

放大图片、截取部分区域



截取区域的PNT数据

→

图片"Color16.bmp"网格大小8*8

```

$0000, PNT0+$0000, PNT0+$0000, PNT0+$0001, PNT0+$0002, PNT0+$0003, PNT0+$0004, PNT0+$0005, PNT0+$0006, PNT0+$0007, PNT0+$0008, PNT0+$0009, PNT0+$000A, PNT0+$000B, PNT0+$000C, PNT0+$000D, PNT0+$000E, PNT0+$000F, PNT0+$0010, PNT0+$0011, PNT0+$0012, PNT0+$0013, PNT0+$0014, PNT0+$0015, PNT0+$0016, PNT0+$0017, PNT0+$0018, PNT0+$0019, PNT0+$001A, PNT0+$001B, PNT0+$001C, PNT0+$001D, PNT0+$001E, PNT0+$001F, PNT0+$0020, PNT0+$0021, PNT0+$0022, PNT0+$0023, PNT0+$0024, PNT0+$0025, PNT0+$0026, PNT0+$0027, PNT0+$0028, PNT0+$0029, PNT0+$002A, PNT0+$002B, PNT0+$002C, PNT0+$002D, PNT0+$002E, PNT0+$002F, PNT0+$0030, PNT0+$0031, PNT0+$0032, PNT0+$0033, PNT0+$0034, PNT0+$0035, PNT0+$0036, PNT0+$0037, PNT0+$0038, PNT0+$0039, PNT0+$003A, PNT0+$003B, PNT0+$003C, PNT0+$003D, PNT0+$003E, PNT0+$003F, PNT0+$0040, PNT0+$0041, PNT0+$0042, PNT0+$0043, PNT0+$0044, PNT0+$0045, PNT0+$0046, PNT0+$0047, PNT0+$0048, PNT0+$0049, PNT0+$004A, PNT0+$004B, PNT0+$004C, PNT0+$004D, PNT0+$004E, PNT0+$004F, PNT0+$0050, PNT0+$0051, PNT0+$0052, PNT0+$0053, PNT0+$0054, PNT0+$0055, PNT0+$0056, PNT0+$0057, PNT0+$0058, PNT0+$0059, PNT0+$005A, PNT0+$005B, PNT0+$005C, PNT0+$005D, PNT0+$005E, PNT0+$005F, PNT0+$0060, PNT0+$0061, PNT0+$0062, PNT0+$0063, PNT0+$0064, PNT0+$0065, PNT0+$0066, PNT0+$0067, PNT0+$0068, PNT0+$0069, PNT0+$006A, PNT0+$006B, PNT0+$006C, PNT0+$006D, PNT0+$006E, PNT0+$006F, PNT0+$0070, PNT0+$0071, PNT0+$0072, PNT0+$0073, PNT0+$0074, PNT0+$0075, PNT0+$0076, PNT0+$0077, PNT0+$0078, PNT0+$0079, PNT0+$007A, PNT0+$007B, PNT0+$007C, PNT0+$007D, PNT0+$007E, PNT0+$007F, PNT0+$0080, PNT0+$0081, PNT0+$0082, PNT0+$0083, PNT0+$0084, PNT0+$0085, PNT0+$0086, PNT0+$0087, PNT0+$0088, PNT0+$0089, PNT0+$008A, PNT0+$008B, PNT0+$008C, PNT0+$008D, PNT0+$008E, PNT0+$008F, PNT0+$0090, PNT0+$0091, PNT0+$0092, PNT0+$0093, PNT0+$0094, PNT0+$0095, PNT0+$0096, PNT0+$0097, PNT0+$0098, PNT0+$0099, PNT0+$009A, PNT0+$009B, PNT0+$009C, PNT0+$009D, PNT0+$009E, PNT0+$009F, PNT0+$00A0, PNT0+$00A1, PNT0+$00A2, PNT0+$00A3, PNT0+$00A4, PNT0+$00A5, PNT0+$00A6, PNT0+$00A7, PNT0+$00A8, PNT0+$00A9, PNT0+$00AA, PNT0+$00AB, PNT0+$00AC, PNT0+$00AD, PNT0+$00AE, PNT0+$00AF, PNT0+$00B0, PNT0+$00B1, PNT0+$00B2, PNT0+$00B3, PNT0+$00B4, PNT0+$00B5, PNT0+$00B6, PNT0+$00B7, PNT0+$00B8, PNT0+$00B9, PNT0+$00BA, PNT0+$00BB, PNT0+$00BC, PNT0+$00BD, PNT0+$00BE, PNT0+$00BF, PNT0+$00C0, PNT0+$00C1, PNT0+$00C2, PNT0+$00C3, PNT0+$00C4, PNT0+$00C5, PNT0+$00C6, PNT0+$00C7, PNT0+$00C8, PNT0+$00C9, PNT0+$00CA, PNT0+$00CB, PNT0+$00CC, PNT0+$00CD, PNT0+$00CE, PNT0+$00CF, PNT0+$00D0, PNT0+$00D1, PNT0+$00D2, PNT0+$00D3, PNT0+$00D4, PNT0+$00D5, PNT0+$00D6, PNT0+$00D7, PNT0+$00D8, PNT0+$00D9, PNT0+$00DA, PNT0+$00DB, PNT0+$00DC, PNT0+$00DD, PNT0+$00DE, PNT0+$00DF, PNT0+$00E0, PNT0+$00E1, PNT0+$00E2, PNT0+$00E3, PNT0+$00E4, PNT0+$00E5, PNT0+$00E6, PNT0+$00E7, PNT0+$00E8, PNT0+$00E9, PNT0+$00EA, PNT0+$00EB, PNT0+$00EC, PNT0+$00ED, PNT0+$00EE, PNT0+$00EF, PNT0+$00F0, PNT0+$00F1, PNT0+$00F2, PNT0+$00F3, PNT0+$00F4, PNT0+$00F5, PNT0+$00F6, PNT0+$00F7, PNT0+$00F8, PNT0+$00F9, PNT0+$00FA, PNT0+$00FB, PNT0+$00FC, PNT0+$00FD, PNT0+$00FE, PNT0+$00FF
            
```

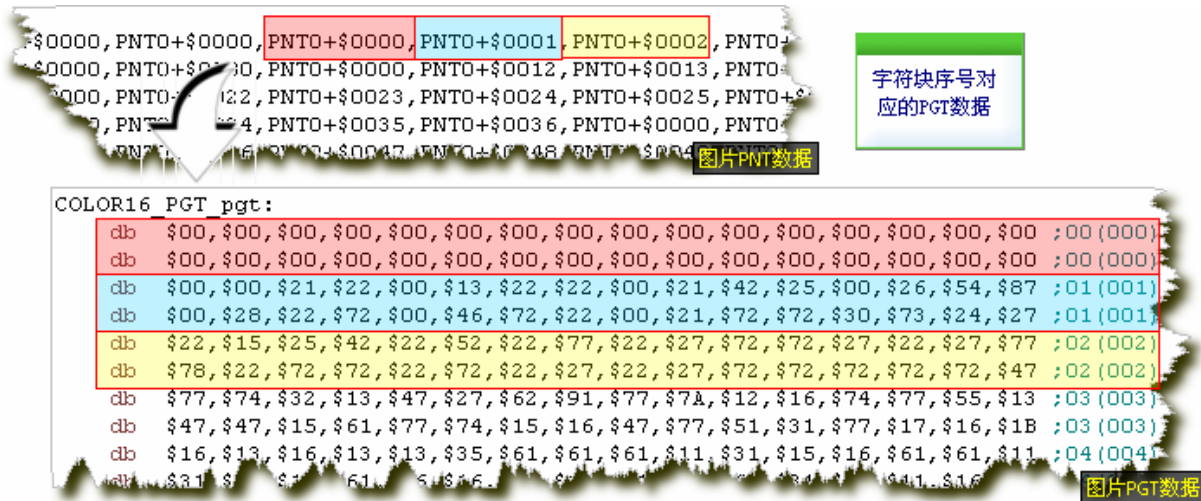
图片PNT数据

©V.R.Technology Co.,Ltd.
Proprietary & Confidential

1

2006.12.05
Version:1.0

PNT 数据对应的 PGT 数据:



字符块序号对应的PGT数据

```

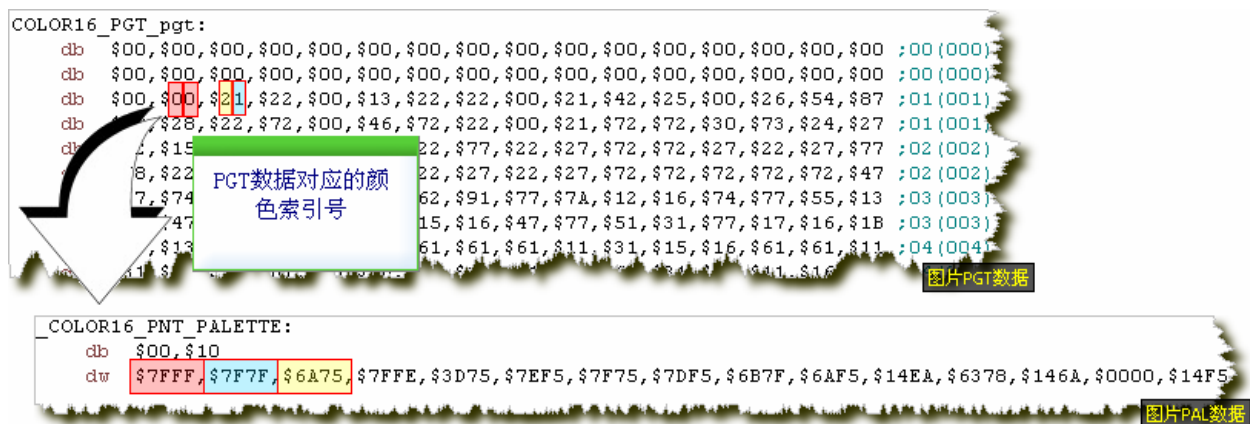
COLOR16_PGT_pgt:
db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00 ;00(000)
db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00 ;00(000)
db $00,$00,$21,$22,$00,$13,$22,$22,$00,$21,$42,$25,$00,$26,$54,$87 ;01(001)
db $00,$28,$22,$72,$00,$46,$72,$22,$00,$21,$72,$72,$30,$73,$24,$27 ;01(001)
db $22,$15,$25,$42,$22,$52,$22,$77,$22,$27,$72,$72,$27,$22,$27,$77 ;02(002)
db $78,$22,$72,$72,$22,$72,$22,$27,$22,$27,$72,$72,$72,$72,$72,$47 ;02(002)
db $77,$74,$32,$13,$47,$27,$62,$91,$77,$7A,$12,$16,$74,$77,$55,$13 ;03(003)
db $47,$47,$15,$61,$77,$74,$15,$16,$47,$77,$51,$31,$77,$17,$16,$1B ;03(003)
db $16,$13,$16,$13,$13,$35,$61,$61,$61,$11,$31,$15,$16,$61,$61,$11 ;04(004)
db $31,$15,$16,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61
    
```

图片PNT数据

图片PGT数据

PGT 数据: 按字符块保存图片的颜色索引号（即颜色在调色板中的顺序），只保存第一次出现的数据。不同的颜色方式和字符方式，保存的格式和大小也不同。（具体见 VT1682 指南 7.4）

1. 在 16 色方式，保存 4 个像素的颜色索引号需要 2 个字节（每个索引号使用 4bits 保存），保存一个 8*8 的字符块需要 32 个字节。Byte0 的低 4 位保存第一个索引号，Byte0 的高 4 位保存第二个索引号；Byte1 的低 4 位保存第三个索引号，byte1 的高 4 位保存第四个索引号。
2. 在 64 色方式，保存 4 个像素的颜色索引号需要 3 个字节（每个索引号使用 6bits 保存），保存一个 8*8 的字符块需要 48 个字节。Byte0 的低 6 位保存第一个索引号，Byte0 的高 2 位和 Byte1 的低 4 位保存第二个索引号，Byte1 的高 4 位和 Byte2 的低 2 位保存第三个索引号，Byte2 的高 6 位保存第 4 个索引号。
3. 在 256 色方式，保存 4 个像素的颜色索引号需要 4 个字节（每个索引号使用 8bits 保存），保存一个 8*8 的字符块需要 64 个字节。Byte0、Byte1、Byte2、Byte3 顺序各保存一个索引号。



PGT数据对应的颜色索引号

```

COLOR16_PGT_pgt:
db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00 ;00(000)
db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00 ;00(000)
db $00,$00,$21,$22,$00,$13,$22,$22,$00,$21,$42,$25,$00,$26,$54,$87 ;01(001)
db $00,$28,$22,$72,$00,$46,$72,$22,$00,$21,$72,$72,$30,$73,$24,$27 ;01(001)
db $22,$15,$25,$42,$22,$52,$22,$77,$22,$27,$72,$72,$27,$22,$27,$77 ;02(002)
db $78,$22,$72,$72,$22,$72,$22,$27,$22,$27,$72,$72,$72,$72,$72,$47 ;02(002)
db $77,$74,$32,$13,$47,$27,$62,$91,$77,$7A,$12,$16,$74,$77,$55,$13 ;03(003)
db $47,$47,$15,$61,$77,$74,$15,$16,$47,$77,$51,$31,$77,$17,$16,$1B ;03(003)
db $16,$13,$16,$13,$13,$35,$61,$61,$61,$11,$31,$15,$16,$61,$61,$11 ;04(004)
db $31,$15,$16,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61,$61
    
```

```

COLOR16_PNT_PALETTE:
db $00,$10
dw $7FFF,$7F7F,$6A75,$7FFE,$3D75,$7EF5,$7F75,$7DF5,$6B7F,$6AF5,$14EA,$6378,$146A,$0000,$14F5
    
```

图片PGT数据

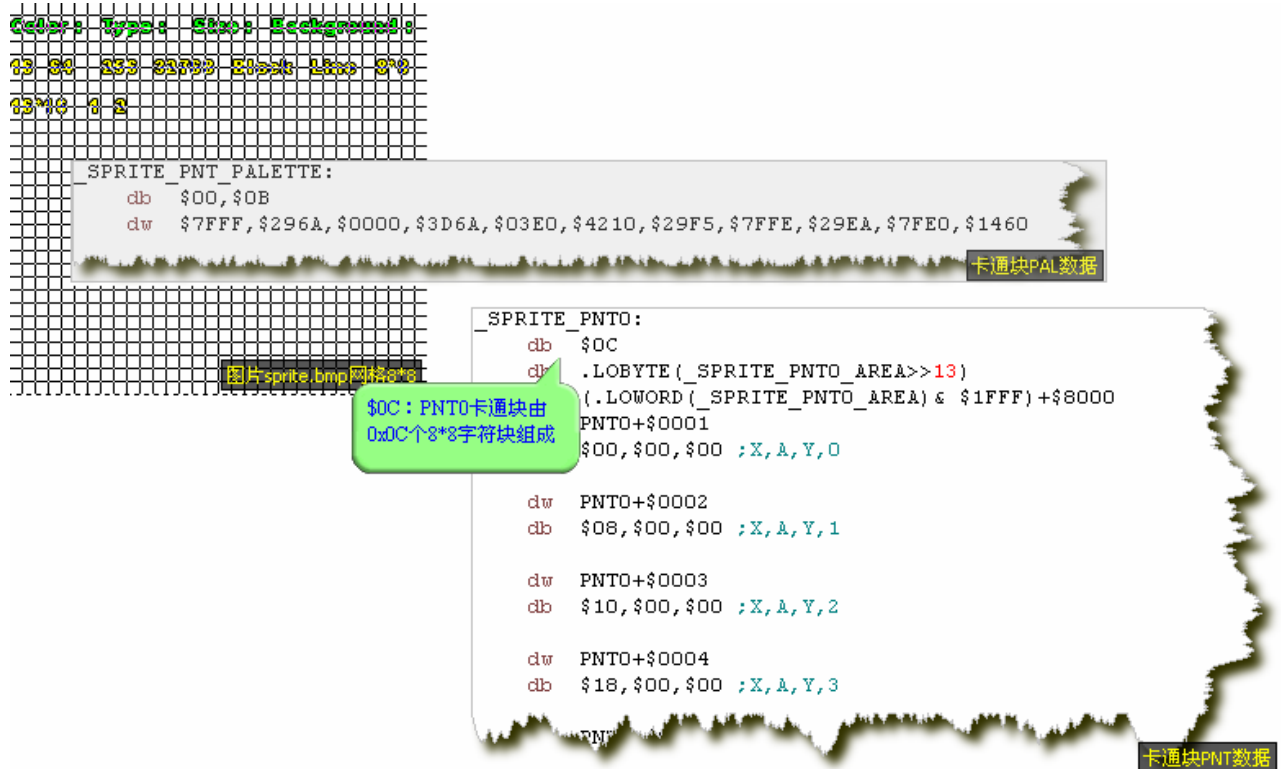
图片PAL数据

如上图：截取的 PGT 数据为 000 号字符块、001 号字符块和 002 号字符块。一个颜色索引号代表一个像素点的颜色值。

卡通块:

以图片 sprite.bmp 为例, 在 16 色与 8*8 字符大小的方式下转换, 导出部分 PALETTE、PGT 和 PNT 数据:

卡通块的 PALETTE、PGT 数据与背景的格式类似。(说明见背景部分)



Code Type: Sprite Background
 13 00 003 23732 0000: Line 8*8
 03*0 0 2

```

_SPRITE_PNT_PALETTE:
  db  $00,$0B
  dw  $7FFF,$296A,$0000,$3D6A,$03E0,$4210,$29F5,$7FFE,$29EA,$7FEO,$1460
  
```

卡通块PAL数据

```

_SPRITE_PNT0:
  db  $0C
  dw  .LOBYTE(_SPRITE_PNT0_AREA)>>13)
      (.LOWORD(_SPRITE_PNT0_AREA) & $1FFF)+$8000
      PNT0+$0001
      $00,$00,$00 ;X,A,Y,0

  dw  PNT0+$0002
  db  $08,$00,$00 ;X,A,Y,1

  dw  PNT0+$0003
  db  $10,$00,$00 ;X,A,Y,2

  dw  PNT0+$0004
  db  $18,$00,$00 ;X,A,Y,3
  
```

图片sprite.bmp网格8*8

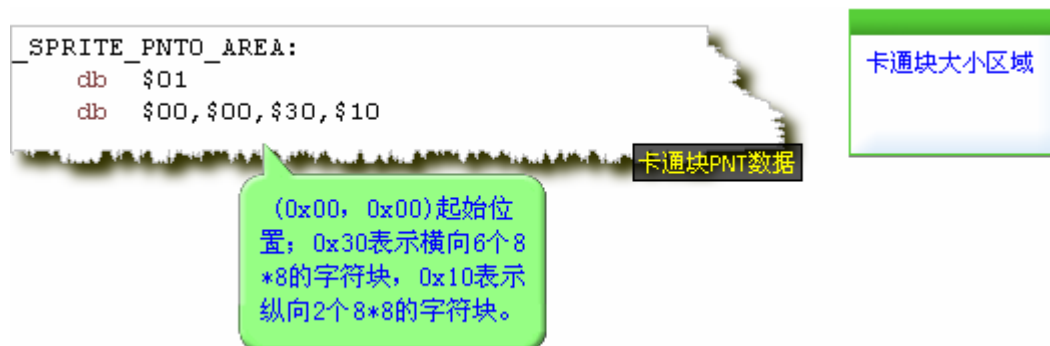
\$0C: PNT0卡通块由 0x0C个8*8字符块组成

卡通块PNT数据

卡通块的区域大小:

第一行数据: 0x01 表示卡通块区域(矩形范围)个数。

第二行数据: (0x00, 0x00)起始位置; 0x30 表示横向 6 个 8*8 的字符块, 0x10 表示纵向 2 个 8*8 的字符块。



```

_SPRITE_PNT0_AREA:
  db  $01
  db  $00,$00,$30,$10
  
```

卡通块大小区域

卡通块PNT数据

(0x00, 0x00)起始位置; 0x30表示横向6个8*8的字符块, 0x10表示纵向2个8*8的字符块。

第一行数据 0x01 表示卡通块区域（矩形范围）个数为 1 个；

