



VT1682 Q & A

VT1682 Question and Answer

目录

1. 动画相关问题

1.1 PIC32 工具

Q1.1-001	动画数据转换的题。	5
Q1.1-002	如何将转换后的文件包含在程序里？	5
Q1.1-003	转换后的 PNT0 是什么意思？	5
Q1.1-004	如何使用 include 和 incbin？	5
Q1.1-005	动画数据转换的问题。	6
Q1.1-006	如何得固定的数据？	9

1.2 设置相关问题

Q1.2-001	DMA 传输字节大小的问题。	9
Q1.2-002	打开图层及关闭图层。	9
Q1.2-003	设置动画段号。	10
Q1.2-004	加载动画时所出现的问题。	10
Q1.2-005	如何使用动画的缓冲与堆栈？	11
Q1.2-006	在同一个位置如何设置动画的优先级？	12
Q1.2-007	_SpriteBuffer 和_SpriteStack 造成溢出的问题。	12
Q1.2-008	BG1, BG2, SP 是否共用两个 256 色盘？	12
Q1.2-009	如何设置动画深度 0,1,2,3？	12
Q1.2-010	如何把变量或数据放在该指定段？	13
Q1.2-011	动画在屏幕上为何只能显示 128 个(16*16)？	13
Q1.2-012	$X \times 8 + X \times 6 = \600 这个公式中 8 和 6 各表示什麼含义？	14
Q1.2-013	属性字节的问题。	14

2. 软件相关问题

2.1 程序编写

Q2.1-001	PRAM 中定义的 BUFFER 问题。	14
Q2.1-002	那里设置 code 段、RODATE 段、DATA 段以及 BSS 段？	15
Q2.1-003	程序中的 export,import 是对什麼而言？	15
Q2.1-004	程序编写中出现的错误提示及导致错误的原因。	15
Q2.1-005	改变 PROM 大小出现 PROM 溢出的问题。	15
Q2.1-006	承上提 2.1-005 所出现的问题。	15
Q2.1-007	改变 PROM 大小出现 PROM 溢出的问题。	16

2.2 DMA 常见问题

- Q2.2-001 进行 DMA 传输时所出现的问题。.....17
- Q2.2-002 进行 DMA 传输时所出现的问题。.....17
- Q2.2-003 DMA 传输到 sprite Vram 时，能否只传一个 PNT？.....18
- Q2.2-004 DMA 期间，CPU 停止工作,会不会被中断打断？.....18

2.3 编译器相关问题

- Q2.3-001 嵌入汇编语法的问题。.....18

2.4 乘除寄存器

- Q2.4-001 用乘，除寄存器时，该等多长时间才能读结果？.....19

2.5 结构体

- Q2.5-001 标准 C 结构体占用空间较多时，会出现的错误提示。.....19

2.6 函数 BUG

- Q2.6-001 下移函数的 BUG 问题。.....20

2.7 内存映像相关问题

- Q2.7-001 程序的保存空间的扩充问题。.....21

2.8 地址线

- Q2.8-001 PQ2EN、COMR6、EXT2421...各指的是那些地址线？.....21

2.9 VRAM

- Q2.9-001 通过\$2004,\$2007 读 VRAM 也要先忽略第一个字节？.....22

3. 硬件相关问题

3.1 XIOF

- Q3.1-001 XIOF3 能不能改变状态成为输入(input)方式？.....22
- Q3.1-002 使用 XIOF0 和 XIOF1 当作输出时的错误该如何解决?..... 22

3.2 DMA

- Q3.2-001 从内部的 RAM DMA 到外部的 RAM 时会出现错误?22
- Q3.2-002 可以跨 Bank(Bank crossing)吗?22

3.3 PPU

- Q3.3-001 在背景 2(BK2)256 色方式图形(Graphic)会出现错误吗?23

3.4 EXIT IRQ

- Q3.4-001 为何主 CPU 外部的 IRQ 是高准位触发而不是负的边缘触发?23

3.5 LCD

- Q3.5-001 在\$200C 的 LCD_CLK 置为 3 时,LCD 显示屏上会有黑点吗?23

3.6 AUDIO DAC

- Q3.6-001 播放 Audio 的时会有一些噪音, 该如何解决?23

3.7 ADC

- Q3.7-001 使用 XIOF0 和 XIOF1 当作 ADC 端口时的错误该如何改善?23

3.8 CCIR

- Q3.8-001 在 YUV 方式中, 有些高明亮度颜色是错误的颜色, 如何改善?23

1. 动画相关问题

1.1 PIC32 工具

Q1.1-001 用 VT168 的图形转换工具 PIC32 进行动画的转换时，转换出来的数据只有 3 列，而 Demo 中转出来的数据都是 5 列，这 5 列数据都代表什麼含义？转出来的 3 列数据正确吗？

答：数据是正确的，例如：

```
dw PNT0+$0001
```

```
db $00,$00,$00 ;Y,A,X,0
```

同样是 5 个字节

Q1.1-002 背景和动画图片使用 VT168 的图形转换工具 PIC32 进行转换,转换后的文件在程序中如何包含进来？

答：可以使用 ".include" 或 ".incbin" 偽指令

Q1.1-003 动画和背景同时使用 VT168 的图形转换工具 PIC32 进行转换后出现的 PNT0 是什麼意思？

如： $PNT0 = (.LOWORD(_GREENPIECE09_PGT) \& \$1FFF) / \$100$ 。（為什麼這麼做？）

_GREENPIECE09_PNT0:

```
db $03
```

```
db .LOBYTE(_GREENPIECE09_PNT0_AREA>>13)
```

```
dw (.LOWORD(_GREENPIECE09_PNT0_AREA) \& $1FFF) + $8000
```

```
dw PNT0+$0001
```

```
db $00, $00, $00 ;Y, A, X, 0
```

```
dw PNT0+$0002
```

```
db $10, $00, $00 ;Y, A, X, 1
```

```
dw PNT0+$0003
```

```
db $20, $00, $00 ;Y, A, X, 2
```

答：PNT0 是根据 PGT 文件编译装载的位置自动计算出开始的序号，如果你将 PGT 文件按 \$2000 地址对齐放置，可以简单理解 $PNT0 = 0$ 。上面的那个公式表示取出标号 _GREENPIECE09_PG 的编译地址的低 16 位对于 \$2000 的偏移量，再用偏移量除以一个块的大小，得出第一个块的序号

Q1.1-004 背景和动画图片使用 VT168 的图形转换工具 PIC32 进行转换,转换后的文件在程序中如何使用 include 和 incbin 包含进来？都需要怎样设置？在 data.s 中如何包含？在 demo.h 中如何设置？在 demo.c 中如何包含？

答：只能在 data.s 中包含，不能在 C 的文件中包含，因為格式只适用于汇编的语法。在 data.s 中有很多这样的包含语法，你仔细看看。

Q1. 1-005 用 VT168 的图形转换工具 PIC32 进行动画的转换时，转出来的数据如下所示（这是一张分辨率为 96*16 的图片）图片如下所示：



```
. scope
PNT0=(. LOWORD(_SPRITE_PGT)&$1FFF)/$80
.segment "PGTDATA"
.align $80
.export _SPRITE_PGT
_SPRITE_PGT:
.incbin "Sprite_pgt.bin"
.RODATA
.export _SPRITE_PNT_PALETTE
_SPRITE_PNT_PALETTE:
    db $01,$09
    dw $5C1C,$7FFF,$39DF,$58DF,$6C5F,$003B,$025E,$22BF,$4B5F
.segment "DEFINE"
.export _SPRITE_PNT0
_SPRITE_PNT0:
    db $05
    db .LOBYTE(_SPRITE_PNT0_AREA>>13)
    dw (. LOWORD(_SPRITE_PNT0_AREA)& $1FFF)+$8000
    dw PNT0+$0001
    db $00,$00,$00 ;Y, A, X, 0
    dw PNT0+$0002
    db $00,$00,$10 ;Y, A, X, 1
    dw PNT0+$0003
    db $00,$00,$20 ;Y, A, X, 2
    dw PNT0+$0004
    db $00,$00,$30 ;Y, A, X, 3
    dw PNT0+$0005
    db $00,$00,$40 ;Y, A, X, 4
_SPRITE_PNT0_AREA:
    db $01
    db $00,$00,$50,$10
```

. endscope

请问：这段数据是否要转化成 Demo 中的 5 列数据：如下（这是一段 Demo 中的动画数据）

```
Sprite04:
    db $08
    db .LOBYTE(Sprite01Area>>13)
    dw (. LOWORD(Sprite01Area)& $1FFF)+$8000
    db $11,$10,$00,$00,$00
    db $12,$10,$10,$00,$00
```

```
db $13, $10, $20, $00, $00
db $14, $10, $30, $00, $00
db $15, $10, $40, $00, $00
db $16, $10, $50, $00, $00
db $17, $10, $60, $00, $00
db $18, $10, $70, $00, $00
```

在这段数据中，第 1 列表示索引序号，第 2 列表示的是调色板的序号，第 3 列表示的是 x 坐标，第 4 列表示的是动画的深度，第 5 列表示的是 y 坐标。那么转出来的数据是否也要转化成此形式，如果是要转化成此种形式，应该如何把

```
dw PNT0+$0001
    db $00, $00, $00 ;Y, A, X, 0
```

转化成 5 列数据？

答：PIC32 转出的数据其实也是 5 列的，其中 “dw PNT0+\$0001 ” 一个 WORD 表示索引序号和调色板序号。“db \$00, \$00, \$00 ;Y, A, X, 0 ” 表示 Y 坐标，属性，X 坐标。只是 Y 坐标和 X 坐标的位置跟以前的是相反的。

(2) 如果我把我用工具转化后的数据：(如下)

```
_SPRITE_PNT0:
db $05
db .LOBYTE(_SPRITE_PNT0_AREA>>13)
dw (.LOWORD(_SPRITE_PNT0_AREA)& $1FFF)+$8000
dw PNT0+$0001
db $00, $00, $00 ;Y, A, X, 0
dw PNT0+$0002
db $00, $00, $10 ;Y, A, X, 1
dw PNT0+$0003
db $00, $00, $20 ;Y, A, X, 2
dw PNT0+$0004
db $00, $00, $30 ;Y, A, X, 3
dw PNT0+$0005
db $00, $00, $40 ;Y, A, X, 4
```

_SPRITE_PNT0_AREA:

```
db $01
db $00, $00, $50, $10
```

把上面这段数据转化成以下形式，添加前两列，如下：

```
_SPRITE_PNT0:
db $05
db .LOBYTE(_SPRITE_PNT0_AREA>>13)
dw (.LOWORD(_SPRITE_PNT0_AREA)& $1FFF)+$8000
db $11, $10, $00, $00, $00
db $11, $10, $10, $00, $00
db $11, $10, $20, $00, $00
```

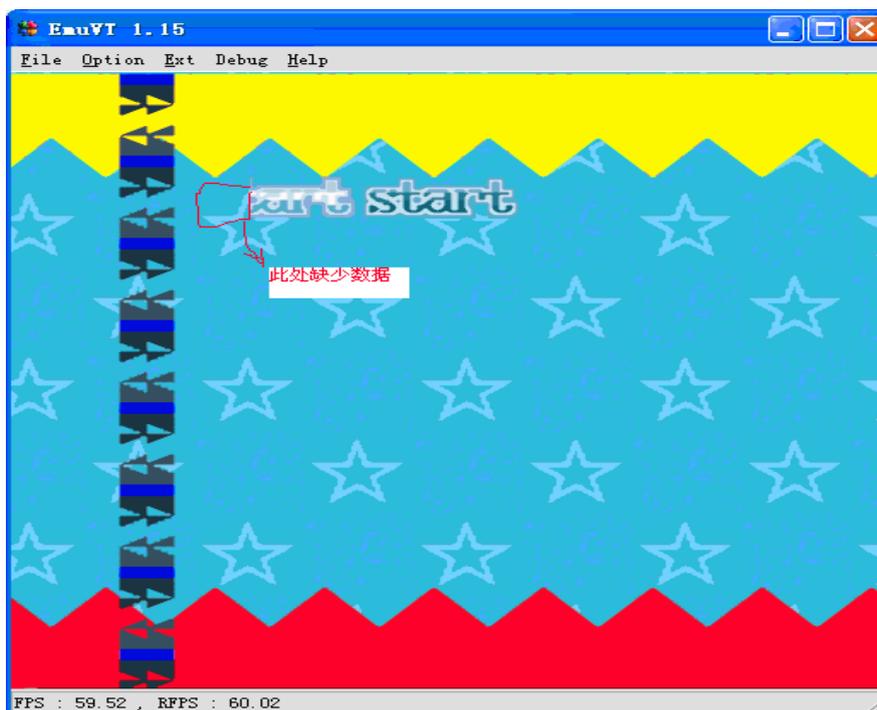
```

db $11, $10, $30, $00, $00
db $11, $10, $40, $00, $00
_SPRITE_PNT0_AREA:
db $01

db $00, $00, $50, $10

```

前两列是我添加的，然后把最后一列和第 3 列对换，因为最后一列 Demo 中是 y 坐标，而转化出来的是 x 坐标，所以进行了调换。结果通过软件模拟器模拟，动画是出来了，但是少一块数据，少的数据正好是第一块数据（如下图所示）。这是为什么？

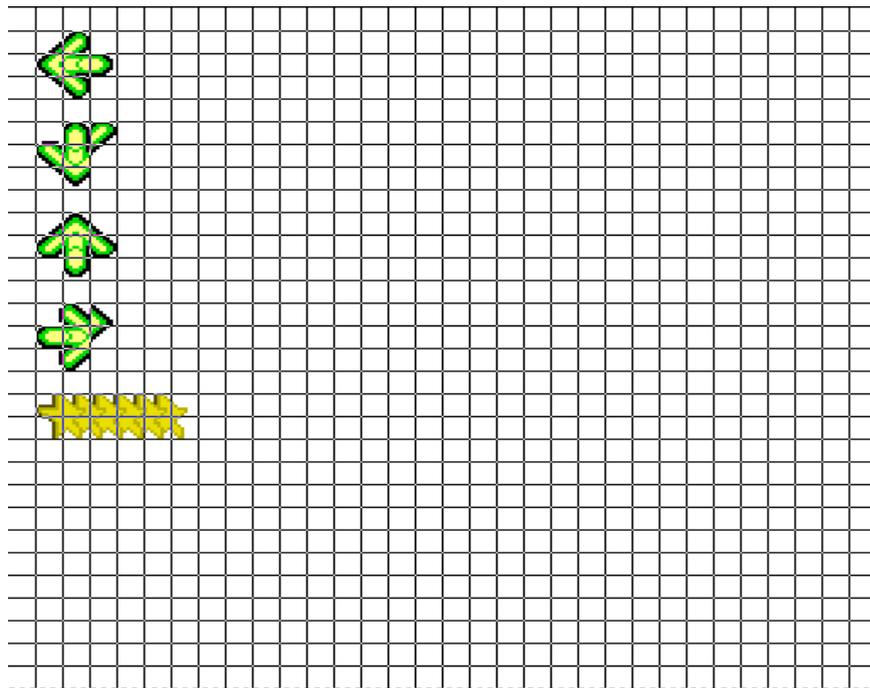


答：你的序号为什么都是\$11?请注意如果序号为00的卡通是不会被显示的。

(3) 上面是一张分辨率为 96*16 的图片，按 16*16 进行分块，应该是 6 块。可是转出来的数据是 5 块，也是少一块，这是为什么呢？

答：卡通的作图是有规则的，请参考下图：

每一个卡通组，四周必须留空，如果按照 8*8 转换四周需要留空 8 点（1 格），如果按照 16*16 转换，四周需要留空 16 点（2 格）



(4) 一组动画是否可以用不同的色盘？如果可以，如何设置及调用？

答：可以的，如果使用了不同的色盘，转换出的数据会包含此信息

Q1.1-006 PIC32 转图工具转出的背景（16/64/256 色）数据为浮动的，那么 DEMO 中有的背景数据为固定的，固定的数据是怎样得到的？

答:固定的数据可以把 PNT0 设置为 0，或者删除掉"PNT0+"

1.2 设置相关问题

Q1.2-001 下面的(1)和(2)有何区别？

```
(1)DAMTransmitTable[0] = 0x1E00;
    DAMTransmitTable[1] = (int)PaletteBuffer;
    DAMTransmitTable[2] = 0;
    DAMTransmitTable[3] = 0xFF00;
(2)DAMTransmitTable[0] = 0x1E00;
    DAMTransmitTable[1] = (int)PaletteBuffer;
    DAMTransmitTable[2] = 0;
    DAMTransmitTable[3] = 0xFF80;
```

在 DAMTransmitTable[3]由 0xFF00 改为 0xFF80，有什么区别？

答：FF00 表示传送 512 字节，FF80 表示传送 256 字节

Q1.2-002 为什么打开图层要和 80 或，如"ora #\$80\n"；而关闭图层与 7F 与，如"and #\$7F\n"？

答：因為最高位 1 代表打开图层，為 0 代表关闭图层

Q1.2-003 在设置动画的段号时，如下：

```
// 设置 Sprite Segment
//
// 参数：   DataLabel      对应 PGT 的标号
//
#define     SetSprSegment(DataLabel)   \
    __asm__("lda    #.LOBYTE(%s>>13)\n" \
    "sta    _P_Spr_SegmnetL\n" \
    "lda    #.LOBYTE(%s>>21)\n" \
    "sta    _P_Spr_SegmnetH\n" \
    ,DataLabel,DataLabel   \
    );
```

在这段代码中，為什麼 s 右移 13 和 21 是何含义？

答：第一个 %s 表示第一个参数“DataLabel”，第二个 %s 表示第二个参数“DataLabel”，有关嵌入汇编语法请参考编译器相关文件

Q1.2-004 加载动画时，动画的大小是 16*16。每行加载的数目為什麼达不到 16 个。（只能加载 8 个，超过 8 个就不能显示出来，如图 1（本应加载 12 个，却只加载 8 个，后面 4 个红色的表示没有填充）。而每列却可以加载到 15 个，这是為什麼？ 每屏加载的动画数目而且不能超过某个数目（<240），当超过这个数目后就会把前面的动画充斥掉，如图 2（此图本应上 48 个 16*16 的动画，而只上了 24 个）。这是什麼原因？应怎麼解决？



图 1

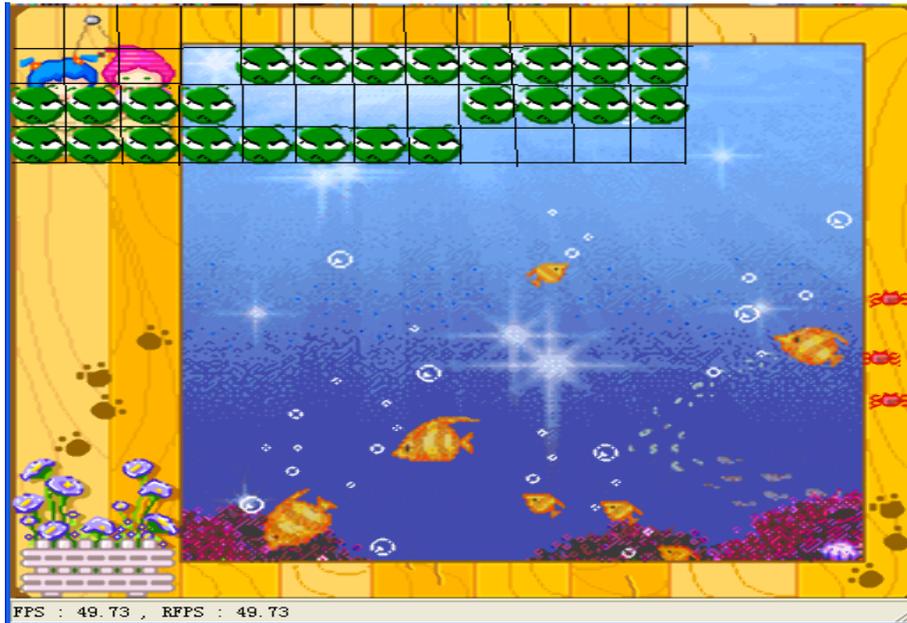
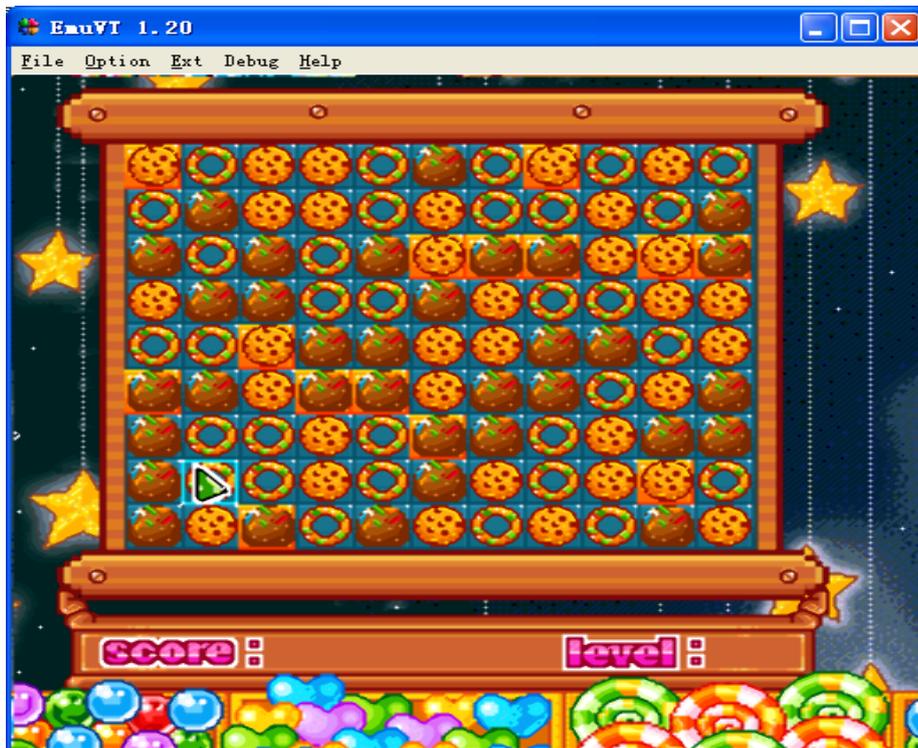


图 2

答：每行最多是显示 16 个 Sprite,每列没有限制显示多少个 Sprite，Sprite 总数不能超过 240 个。

Q1.2-005 现在使用的是多个卡通组，因此按照上面所计算的，我们的卡通块肯定达不到 240 个，按照我的理解，动画是按堆栈的顺序排序的，即首先显示 0xFF，然后是 0xFE，依次递减的？我现在要实现一系列动画的下移，如下图所示：



我现在要实现如下功能：

(1) 例如：我现在在第一列，下面有四个相同的，我已经做了删除的函数，完全可以删除，现在的问题是：删除这四个以后，上面的 5 个要依次落下来，在上面空出四个空块？

我使用的函数是 `SetSpriteXY(jj*16+32,(ii+a)*16+24,0xFD-transfer)`;用它来设置动画的位置。但是问题是，用了这个函数后，第一次还可以，但是以后就出现了动画被重叠覆盖的现象？有时还会花屏？这是为什么啊？

答：删除的函数会使序号重新排布，比如序号为 **FC** 的卡通，当删除了 **FE** 号卡通后，他的操作序号就变为 **FD** 了。建议不要频繁使用删除卡通的操作，而尽量使用 **DisableSprite** 和 **SetSprite** 进行覆盖操作。

(2) 请问如何使用动画的缓冲和堆栈？

答：动画的缓冲是根据 **VT1682 Datasheet** 所描述的数据结构相一致的。而堆栈是通过 **InsertSprite** 入栈，他负责描述一个卡通组的坐标和对应的卡通组缓冲数据的地址。现在我只能笼统的讲一下，更详细的需要你看代码，把看不懂的地方提出来。你可以用模拟器 **DEBUG** 并观看内存变量有助你理解。

Q1.2-006 现在要在同一位置，比如在我的程序里，我现在要在同一位置显示四个动画，例如最下层是砖块，砖块上是蛋糕，蛋糕上是光标，光标上要显示其它东西，那我该如何设置它们的优先级呢？

答：有两种方式：

1. 请修改属性字节，即 **A** 那个字节
2. 同一属性的卡通，在 **Buffer** 位置越靠前优先级越高

Q1.2-007 在本程序中要显示 `showheart()` ; `showmoli()` 这两个动画，`showheart()` 是显示心的动画，`showmoli()` 是显示魔力棒的动画，但是现在和我前面的动画一同显示，就出现花屏，但是如果单独显示，一切正常，这是为什么呢？

答：可能是 **_SpriteBuffer** 和 **_SpriteStack** 溢出造成的

Q1.2-008 **BG1,BG2,SP** 是否共用两个 **256** 色盘？如果 **BG1** 用了 **256** 色，**BG2** 也用了 **256** 色，那么动画的颜色在哪儿呢？透明色由哪一个色盘决定？

答:如果 **BG1** 用了 **256** 色，**BG2** 也用了 **256** 色，那么动画的颜色也必须在这 **256** 中。两个调色板的第一个颜色是透明色，如果两个叠加的背景都是透明色，则显示级别低的背景对应的透明色的颜色。

Q1.2-009 请问动画的深度应该如何控制，假如我们现在在屏幕同一位置需要显示四个动画，怎么样控制它们的显示顺序？根据资料介绍，是设置深度 **0, 1, 2, 3**，**0** 在最上层，可是我们如何设置此深度呢？是通过动画的数据吗？如果是，我们应该如

何设置此数据呢？

答：卡通	D7	D6	D5	D4	D3	D2	D1	D0
0x2018					SPALSEL	SP_EN	SP_SIZE	

Q1.2-010 如果在变量前加 **const** 修饰符，是否可以节省内存？如何把自己的数据或变量放在指定的段中，如 **code** 段、**RODATE** 段、**DATA** 段以及 **BSS** 段？

答：变量前加 **const** 修饰符是不占用 **RAM** 内存的。在汇编中用段修饰符可以把自己的数据或变量放在指定的段中。例如：

```
.segment "DEFINE"
.export _PaletteTable
_PaletteTable:
    dd VRTLOGO_palette ;00
    dd BACK1_PNT_palette ;01
    dd PALETTE_PNT_palette ;02
    dd TEST256_1_PNT_palette ;03
    dd TEST256_2_PNT_palette ;04
    dd TEST256_3_PNT_palette ;05
    dd TEST256_4_PNT_palette ;06
    dd _TEST4SCR_PNT_PALETTE ;07
```

是把数据放到 **DEFINE** 段。

```
.zeropage
sp: .res 2 ; Stack pointer
sreg: .res 2 ; Secondary register/high 16 bit for longs
regsave: .res 4 ; slot to save/restore (E)AX into
ptr1: .res 2
ptr2: .res 2
ptr3: .res 2
ptr4: .res 2
tmp1: .res 1
tmp2: .res 1
tmp3: .res 1
tmp4: .res 1
```

是把变量放到 **ZEROPAGE** 段

Q1.2-011 动画一屏幕也只能显示 128 个（按 16*16），这是为什么？

答：您使用了 **InsertSprite** 函数，此函数在设计的时候是按照以下思路设计的：

```
_SpriteBuffer = $0200 ;Sprite Buffer $5A0
_SpriteStack = $07A0 ;$60
```

一共的 **BUFFER** 是 \$600,由 **SpriteBuffer** 和 **SpriteStack** 共同使用

SpriteStack 存放的是卡通组的索引数据，每一组数据由 8 个字节表示，以栈的方式存储而 **SpriteBuffer** 存放的是单个卡通块的数据，跟 **Datasheet** 上所讲的结构是一致

的。如果你简单的使用 **16×16** 的卡通，我们算一下理论的最大卡通数：
 $X \times 8 + X \times 6 = \600 $X=109$, 所以以缓存不溢出的情况下最大显示 109 个卡通
 建议您使用卡通组的数据结构

Q1.2-012 关于上面这个问题所衍生的小问题如下：

- (1) 如何使用卡通组的数据结构？
- (2) $X \times 8 + X \times 6 = \600 这个公式中 8 和 6 各表示什麼含义？
- (3) 目前我的代码可以上到 128 个动画？

请问这个动画的缓冲和堆栈可否自己设置？

答：按照以前说的 **Sprite** 的作图方法作图，通过 **PIC32** 转换，按照 **Sprite** 格式输出的就是卡通组的数据结构。**8** 代表卡通组数据由 **8** 个字节表示，**6** 代表每一个卡通块由 **6** 个字节表示。

使用动画的个数是要看你怎麼样用，不是绝对的，比如你只有一个卡通组，这个卡通组有 **240** 个卡通块，那麼你完全可以使用这 **240** 个卡通块，如果你有 **N** 个卡通组，每个卡通组有 **1** 个卡通块，那麼 $n \times 8 + n \times 6 = \600 ， $n=109$,就是说最多 **109** 个卡通可以使用，超过了就出错了。

这个动画的缓冲和堆栈当然可以自己设置，因為他是程序实现的，你修改我的源代码程序就可以了，具体怎麼修改需要你看懂我的那段源代码。

Q1.2-013 关于上面这个问题所衍生的小问题如下：

请问修改这个属性，即 **A** 那个字节，是不是这样设置，它不是有四个深度吗？是不是 **\$00** 表示第 0 层；**\$10** 表示第 1 层；**\$20** 表示第 2 层；**\$30** 表示第 3 层呢？

我现在采用的是你的第二种方法，即同一属性的卡通，在 **Buffer** 位置越靠前优先级越高，但是当动画多时)，就出现了花屏，可能是 **_SpriteBuffer** 和 **_SpriteStack** 溢出造成的，但我现在最大只能显示 128 个呀？

答：是这样的，属性字节最高两位没用，由高到低依次爲：
Depth[1:0],Flip[1:0],VRCH,PalSel

按照我上面的计算，超过 **109** 个卡通就会不正常了

2. 软件相关问题

2.1 程序编写

Q2.1-001

```
.export _SpriteBuffer
_SpriteBuffer = $0200 ;Sprite Buffer $5A0
.export _SpriteStack
_SpriteStack = $07A0 ;$60
.export _PaletteBuffer
_PaletteBuffer = $0800 ;Palette Buffer
```

请问：上段代码中\$0200、\$07A0、\$0800 三个地址是 VRAM 中的吗？若不是，那请问它们是哪个 ROM 或 RAM 中的地址？

答：不是的，是 PRAM 中的定义的 BUFFER

Q2.1-002 请问在哪里设置 code 段、RODATE 段、DATA 段以及 BSS 段？

答：在 CFG 文件中设置这些数据段

Q2.1-003 程序中的 export,import 是对什麼而言的？

答：export 是声明一个标号為外部的可供在另一个文件中调用这个标号，import 是引用一个声明為外部的标号。详细资料请参考相关文件，在 www.vrt.com.tw 有下载

Q2.1-004 在程序编写过程中会出现此错误提示：ld65.exe:Error: Memory area overflow in 'PROM',segment 'STARTUP'<743bytes0>,请问如何解决此错误？或者是导致这种错误的原因是什麼？当前，代码还不到 700 行，就发生溢出，当我们删除一些程序或变量，这个问题就可以解决，但是随着程序的增加，此问题又出现了？是不是要在哪里设置？还是要切 BANK？如果是要切 BANK，那麼应该如何切呢？根据资料介绍，它的物理地址纔是 25 根，那麼至少应该寻址 32 M 的空间，那麼為什麼现在这麼少的程序，就会发生溢出呢？

答：你所说的 32M 是物理地址，对于 VT168 来说他的綫性寻址空间是 48K (\$4000-\$FFFF)。对于物理地址的操作需要映射页面 (switch bank)，映射页面需要通过设置相关寄存器,参见 datasheet 有关 Program_Bank_Register 的描述。在 cfg 文件中定义了 'PROM' 的大小，DEMO CODE 中定义的是 start = \$C000, size = \$3FFA，DEMO CODE 中的 CODE，DATA，STARTUP 段都是装载在这一空间，如果是 C 写的 CODE，700 行有可能溢出。

Q2.1-005 在 cfg 文件中改变了 'PROM' 的大小，我把它改為 start=\$C000,size=\$5FFA; 结果出现了此错误提示：ld65.exe:Error:Range error in module 'Demo.s',line 353.我现在已经把 showheart(); showmoli();这两段代码屏蔽了，还有 if((a>2) || (b>2) || (c>2) || (d>2))这个条件检测也屏蔽了，以及关于 a,b,c,d 的定义和赋值都屏蔽了。目前这段代码是没有错误的，只有几个警告。现在只要把屏蔽打开，就会出现 'PROM' 的溢出？

答：还是设置有问题，应该设置為：

```
DATAS: start = $0, size = $7A000, file = %O ,fill = yes, define = yes;  
PROM: start = $A000, size = $5FFA, file = %O ,fill = yes, define = yes;  
这样 PROM 的大小就增加了$2000
```

Q2.1-006 承上个问题：

a (1) 按上面的设置，修改了 CFG 文件中的设置，'PROM' 不会发生溢出了。但是当代码增多的时候，试过再增加一倍的代码，原来是 700 行代码，当让它增加到 1400 行时候，又出现了 'PROM' 溢出，这时候又在 CFG 文件中进行了设置，增大了它的

空间，

```
DATAS: start = $0, size = $71000, file = %O ,fill = yes, define = yes;
```

```
PROM: start = $1000, size = $CFFA, file = %O ,fill = yes, define = yes;
```

这时不会出现 ‘PROM’ 溢出了，但是把 rom.bin 加到模拟器上没有任何效果，出现了黑屏，什麼都没有，这是什麼原因啊？

答：PROM: start = \$1000, size = \$CFFA, file = %O ,fill = yes, define = yes;

表示标号地址从\$1000 开始，size 是\$CFFA,如果程序按照这样编译，编译后的运行地址就不对了，程序当然就不可能运行正确了。

按照我们以前的设想，常用的代码是使用 CODE 段，装载在 PROM 空间，即将物理地址\$7FC00-\$7FFFA 映射到\$C000-\$FFFA 运行，CODE 段在 CFG 文件中的定义如下：

```
CODE:      load = PROM,      type = wprot, define = yes;
```

另外再定义若干个 CODE0，CODE1...这样的段

CODE0 段在 CFG 文件中的定义如下：

```
CODE00:   load = DATAS,      run = CODES, type = wprot, define = yes, align = $2000;
```

这个定义的意思是，编译按 CODES (\$A000-\$BFFF) 所定义的地址编译，编译后的数据装载到 DATAS 定义的空间，数据的起始是按照\$2000 对齐的。

然后在 C CODE 中用 #pragma codeseg ("CODE0") 强定义把不常用的代码编译到 CODE0 段，在调用前需要通过置\$2108 把物理地址映射到\$A000-\$BFFF 然后再调用函数。

当初我们的设想是基于：

\$C000-\$FFFA 作為公用程序运行空间,上电自动将物理地址\$7FC00-\$7FFFA 映射到\$C000-\$FFFA

\$A000-\$BFFF 作為调用子程序运行空间

\$8000-\$9FFF 作為访问数据空间

现在 VT1682 又扩充了\$4000-\$7FFF 可以直接访问的地址空间，所以你可以按照你的思维自行安排你的程序空间结构

Q2.1-007 在 cfg 文件中改变了 ‘PROM’ 的大小，按照上面的回答，即设置為

```
DATAS: start = $0, size = $7A000, file = %O ,fill = yes, define = yes;
```

```
PROM: start = $A000, size = $5FFA, file = %O ,fill = yes, define = yes;
```

但是现在增加了一些程序后，又出现了 ‘PROM’ 溢出？

一开始，cfg 文件中的设置為

```
DATAS: start = $0, size = $7C000, file = %O ,fill = yes, define = yes;
```

```
PROM: start = $C000, size = $3FFA, file = %O ,fill = yes, define = yes;
```

但是在代码快到 700 行时，出现了 ‘PROM’ 溢出。于是按照上面的回答，把 PROM 的大小增加了\$2000，结果就解决了 ‘PROM’ 的溢出。但是当现在代码编写到快 1000 行时，又出现了 ‘PROM’ 的溢出？于是又在 cfg 文件中进行了设置，设置如下

```
DATAS: start = $0, size = $78000, file = %O ,fill = yes, define = yes;
```

```
PROM: start = $8000, size = $7FFA, file = %O ,fill = yes, define = yes;
```

这时不会出现 ‘PROM’ 溢出了，但是把 rom.bin 加到模拟器上没有任何效果，出现

了黑屏，什麼都沒有，這是什麼原因啊？

答：我們 **C-demo** 的程序架構是用**\$8000-\$9FFF** 作為數據區的，此程序架構需要用這一區域訪問 **PNT**，**PAL** 數據，這樣勢必會把**\$8000-\$9FFF** 切換到對應得物理地址，而使對這一區域程序調用造成破壞。

2.2 DMA 常見問題

Q2.2-001 在進行 DMA 傳輸時，如下：

```
SetupDMA((void *)(((int)TEST1_PNT_pnt_DMAINDEX & 0x1FFF)|0x8000))
```

為什麼要和 0x1FFF 與並且還要和 0x8000 或？

答：因為所有的 **DMA Table** 都被安排到邏輯地址**\$8000-\$9FFF**，C 的編譯器編譯得到的是物理地址，所以我們要把物理地址轉換為邏輯地址。

Q2.2-002 macro PROMDMATable Destination,Source,Length

```
dw Destination ;目的地址
dw .LOWORD(Source)|$8000 ;源數據地址
db .LOBYTE(.HIWORD(Source+$200)<<1+.HIBYTE(Source+$200)>>7)
db .LOBYTE(.HIWORD(Source+$200)>>7)
db Length
```

.endmacro

.macro DMAPNTTable Destination,TableName

```
PROMDMATable Destination,TableName,0
PROMDMATable Destination+$200,(TableName+$200),0
PROMDMATable Destination+$400,(TableName+$400),0
PROMDMATable Destination+$600,(TableName+$600),0
db $FF
```

.endmacro

請問：

(1) 在語句 “PROMDMATable Destination+\$200,(TableName+\$200),0” 中，Destination 為什麼要加\$200？為什麼要將給 Length 賦 0？語句 “db \$FF” 為何意？

答：0 表示傳送 256 WORD，請看 VT168 Datasheet 關於 DMA 長度的說明。“db \$FF” 表示該數據表結束

(2) 在語句 “db .LOBYTE(.HIWORD(Source+\$200)<<1+.HIBYTE(Source+\$200)>>7)” 中，

HIWORD(Source+\$200)<<1 和.HIBYTE(Source+\$200)>>7 中(Source+\$200)是什麼意思？整個語句實現了什麼操作？

答：加\$200 偏移

Q2.2-003 用\$2122，DMA 傳輸到 sprite Vram 時，能否用 DMA 只傳一個 PNT

;假設\$1000 存放 sprite PNT 1

```

lda #<$2004
sta $2122
lda #>$2004
sta $2123
lda #08
sta $2003
lda #00
sta $2004
lda #<$1000
sta $2122
lda #>$1000
sta $2123
lda #00
sta $2128
lda #03      ;6 bytes 还是 8 bytes?
sta $2127

```

答：DMA 可以只传一个 PNT,最小单位是一个 Word(两个字节)

```
lda #03      是 6 bytes
```

Q2.2-004 DMA 期间，CPU 停止工作,会不会被中断打断？

答：不会

2.3 编译器相关问题

Q2.3-001 请问，如下：

```
#defineSetText1Attibe(TextSwitch,Color,Size,Bitmap,PALSel,Depth,Scroll,PALSelect)
```

```

if(Color)
{
    __asm__("lda _Tx1_Scroll_Control_Temp\n"
            "and  #03\n"
            "ora #b\n"
            "sta _Tx1_Scroll_Control_Temp\n"
            "lda #b\n"
            "sta _Tx1_Control_Temp\n" \
            "lda _Pal_Select_Control_Temp\n" \
            "and  #0C\n" \
            "ora #b\n" \
            "sta _Pal_Select_Control_Temp\n" \
            ,Scroll,(unsigned)(TextSwitch+Depth+Color+Bitmap+Size),PALSelect \

```

在这段代码中,为甚么要和 03 与后,又和 b 或呢？

,Scroll,(unsigned)(TextSwitch+Depth+Color+Bitmap+Size),PALSelect 这句是何含

义?

答：与上 03 表示保留低两位，%b 表示参数值，第一个%b 表示“Scroll”，第二个%b 表示“(unsigned)(TextSwitch+Depth+Color+Bitmap+Size)”，第三个%b 表示“PALSelect”，有关嵌入汇编语法请参考编译器相关文件

2.4 乘除寄存器

Q2.4-001 用乘，除寄存器时，该等多长时间才能读结果？

答：乘法 **16 CPU CLOCK**
除法 **32 CPU CLOCK**

2.5 结构体

Q2.5-001 在使用标准 C 的结构体时,结构体占用空间较多时会出现此错误提示:
ld65.exe:Error: Memory area overflow in 'RAM',segment 'BSS'<743bytes0>,请问如何
解决此错误?或者是导致这种错误的原因是什麼?

答：你的结构应该定义到 **RODATA**，**DATA** 或 **CODE** 段，不应该定义到 **BSS** 段，即在 **C** 中加 **const** 修饰符。例如：

```
char const MenuTitle[] = "1000 in 1";
```

按照了上面的说法，在结构体前加 **const** 关键字，但是结构体较大时，仍然报错，例如定义了此结构：如下所示：

```
typedef struct                //定义方块的结构体
{
    int fill;                //是否填充，也就是是否有动画，1 为有，0 为没有
    int statu;              //方块的状态
}box;
box screen[9][11];
```

当定义为 `box screen[12][12]`，它就报错。出现的错误提示为：`ld65.exe:Error: Memory area overflow in 'RAM',segment 'BSS'<1333bytes0>`，如何解决此错误？或者是导致这种错误的原因是什麼？为什麼结构体较大时，就会出现此错误？如何避免和解决呢？

答：因为声明的结构数据是放在 **RAM** 中的，而 **RAM** 的分配的空间很小，所以会出现溢出错误。你应该这样：

```
typedef struct                //定义方块的结构体
{
    int fill;                //是否填充，也就是是否有动画，1 为有，0 为没有
    int statu;              //方块的状态
}box;
box const screen[2] = {
    {0,0},
    {1,1}
```

};

表示放在 ROM 的区域，你可以观察 C 文件编译对应的.S 文件，编译的结果是：

```
.....
.segment "RODATA"

_screen:
.word $0000
.word $0000
.word $0001
.word $0001
.....
```

2.6 函数 BUG

Q2.6-001 底下是一个下移函数。这个函数有什么 BUG ？

//下移的函数

```
void movecake()
{
    int ii,jj,kk;
    int transfer1=0,transfer2=0,transfer3=0,transfer=0;
    for(jj=0;jj<11;jj++) //coloun
    {
        kk:    for(ii=0;ii<9;ii++)    //row
        {
            transfer1=ii*11+jj;
            transfer2=transfer1/16;//////////high 4 bits
            transfer3=transfer1%16;//////////low 4 bits

            if(transfer3==10)transfer3=0x0a;
            else if(transfer3==11)transfer3=0x0b;
            else if(transfer3==12)transfer3=0x0c;
            else if(transfer3==13)transfer3=0x0d;
            else if(transfer3==14)transfer3=0x0e;
            else if(transfer3==15)transfer3=0x0f;

            transfer2=transfer2<<4;
            transfer=transfer2+transfer3;

            if(screen[ii][jj].statu!=1)
            {
                SetSpriteXY(jj*16+32,(ii+a)*16+24,0xFD-transfer); //(ii*11+jj)
            }
            else
            {
```


答：VT1682 无法做 VRAM / Sprite RAM 的读取

3. 硬件相关问题

3.1 XIOF

Q3.1-001 XIOF3 能不能改变状态成为输入(input)方式？

答：不能。

Q3.1-002 当使用 XIOF0 和 XIOF1 当作输出(output)时的错误该如何解决？

答：它们被接到此开发板的 FPGA .如果输入(input)方式是需要, 请将 FPGA 和 VT1682 的这两条连线切断即可。

3.2 DMA

Q3.2-001 为何从内部的 RAM(internal RAM)DMA 到外部的 RAM(externalRAM)时会出现错误？到内部的 RAM(internal RAM)时的错误，该如何处理？

答：当 DMA 路径是从内部的 RAM 到外部的 RAM(external RAM)时，在地址线是有一个限制的。这个目标和来源字节地址线 A[12]，必须是相同的。从内部的 RAM(\$0000~\$1FFF) DMA 到内部的 RAM(\$0000~\$1FFF) 是不允许的。

Q3.2-002 可以跨 Bank(Bank crossing)吗？

答：DMA bank 大小是 64K 字节。DMA 地址不可以跨过这个 Bank。

3.3 PPU

Q3.3-001 在背景 2(BK2)256 色方式图形(Graphic)会出现错误吗？水平方向的(横向)位置两个点的飘移(two pixels shift)，会发生吗？卡通块(Sprite)水平方向的横向跳动功能是无效的？

答：不会。不会。无。

3.4 EXIT IRQ

Q3.4-001 为何主 CPU(Main CPU)外部的 IRQ 是高准位触发而不是负的边缘触发 (negative edge trigger) ?

答：使用一个外加的 **74xx74** 或是使用 **SCPU** 外部的 **IRQ(SCPU_UIOB[7])**。

3.5 LCD

Q3.5-001 当在\$200C 的 LCD_CLK 置为 3 时,在 LCD 显示屏上会有一些黑点吗？

答：不会。

3.6 AUDIO DAC

Q3.6-001 当播放 Audio 的时会有一些噪音(noise)，该如何解决？

答：避免从 **Audio DAC** 的最高位(MSB)开始使用。

3.7 ADC

Q3.7-001 在 ADC 端口的 XIOF3 为何会是无效的？当使用 XIOF0 和 XIOF1 当作 ADC 端口时的错误该如何改善？

答：使用 **IOE3** 来取代。它们被接到此开发板的 **FPGA**。如果 **ADC** 方式是需要。请将 **FPGA** 和 **VT1682** 的这两条连线切断即可。

3.8 CCIR

Q3.8-001 在 YUV 方式,有一些高明亮度的(High luminance)颜色是错误的颜色，改善的方式？

答：使用 **CCIR RGB** 介面。