



VT1682 Q & A

VT1682

Question and Answer

目錄

1. 動畫相關問題

1.1 PIC32 工具

Q1.1-001	動畫數據轉換的問題。	5
Q1.1-002	如何將轉換後的文件包含在程序裡？	5
Q1.1-003	轉換後的 PNT0 是什麼意思？	5
Q1.1-004	如何使用 include 和 incbin？	5
Q1.1-005	動畫數據轉換的問題。	6
Q1.1-006	如何得固定的數據？	9

1.2 設置相關問題

Q1.2-001	DMA 傳輸字節大小的問題。	9
Q1.2-002	打開圖層及關閉圖層。	9
Q1.2-003	設置動畫段號。	10
Q1.2-004	加載動畫時所出現的問題。	10
Q1.2-005	如何使用動畫的緩衝與堆棧？	11
Q1.2-006	在同一個位置如何設置動畫的優先級？	12
Q1.2-007	_SpriteBuffer 和 _SpriteStack 造成溢出的問題。	12
Q1.2-008	BG1、BG2、SP 是否共用兩個 256 色盤？	12
Q1.2-009	如何設置動畫深度 0,1,2,3？	12
Q1.2-010	如何把變量或數據放在該指定段？	13
Q1.2-011	動畫在屏幕上為何只能顯示 128 個(16*16)？	13
Q1.2-012	$X \times 8 + X \times 6 = \600 這個公式中 8 和 6 各表示什麼含義？	14
Q1.2-013	屬性字節的問題。	14

2. 軟件相關問題

2.1 程序編寫

Q2.1-001	PRAM 中定義的 BUFFER 問題。	14
Q2.1-002	哪裡設置 code 段、RODATE 段、DATA 段以及 BSS 段？	15
Q2.1-003	程序中的 export,import 是對什麼而言？	15
Q2.1-004	程序編寫中出現的錯誤提示及導致錯誤的原因。	15
Q2.1-005	改變 PROM 大小出現 PROM 溢出的問題。	15
Q2.1-006	承上提 2.1-005 所出現的問題。	15
Q2.1-007	改變 PROM 大小出現 PROM 溢出的問題。	16

2.2 DMA 常見問題

- Q2.2-001 進行DMA 傳輸時所出現的問題。.....17
- Q2.2-002 進行 DMA 傳輸時所出現的問題。.....17
- Q2.2-003 DMA 傳輸到 sprite Vram 時,能否只傳一個 PNT?18
- Q2.2-004 DMA 期間,CPU 停止工作,會不會被中斷打斷?18

2.3 編譯器相關問題

- Q2.3-001 嵌入彙編語法的問題。.....18

2.4 乘除寄存器

- Q2.4-001 用乘,除寄存器時,該等多長時間才能讀結果?19

2.5 結構體

- Q2.5-001 標準 C 結構體佔用空間較多時,會出現的錯誤提示。.....19

2.6 函數 BUG

- Q2.6-001 下移函數的 BUG 問題。.....20

2.7 內存映像相關問題

- Q2.7-001 程序的存貯空間的擴充問題。.....21

2.8 地址線

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

2.9 VRAM

- Q2.9-001 通過\$2004,\$2007 讀 VRAM,也要先忽略第一個字節?22

3. 硬件相關問題

3.1 XIOF

- Q3.1-001 XIOF3 能不能改變狀態成為輸入(input)模式?22
- Q3.1-002 使用 XIOF0 和 XIOF1 當作輸出時的錯誤該如何解決?.....22

3.2 DMA

- Q3.2-001 從內部的RAM DMA到外部的RAM時會出現錯誤？22
Q3.2-002 可以跨 Bank(Bank crossing)嗎？22

3.3 PPU

- Q3.3-001 在背景 2(BK2)256 色模式圖形(Graphic)會出現錯誤嗎？23

3.4 EXIT IRQ

- Q3.4-001 為何主CPU外部的IRQ是高準位觸發而不是負的邊緣觸發？23

3.5 LCD

- Q3.5-001 在\$200C的LCD_CLK置為3時,LCD顯示屏上會有黑點嗎？23

3.6 AUDIO DAC

- Q3.6-001 播放 Audio 的時後會有一些噪音,該如何解決？23

3.7 ADC

- Q3.7-001 使用XIOF0和XIOF1當作ADC端口時,的錯誤該如何改善？23

3.8 CCIR

- Q3.8-001 在 YUV 模式中,有些高明亮度顏色是錯誤的顏色,如何改善？23

1. 動畫相關問題

1.1 PIC32 工具

Q1.1-001 用 VT168 的圖形轉換工具 PIC32 進行動畫的轉換時，轉換出來的數據只有 3 列，而 Demo 中轉出來的數據都是 5 列，這 5 列數據都代表什麼含義？轉出來的 3 列數據正確嗎？

答：數據是正確的，例如：

```
dw PNT0+$0001
```

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

同樣是 5 個字節

Q1.1-002 背景和動畫圖片使用 VT168 的圖形轉換工具 PIC32 進行轉換，轉換後的文件在程序中如何包含進來？

答：可以使用".include"或".incbin"偽指令

Q1.1-003 動畫和背景同時使用 VT168 的圖形轉換工具 PIC32 進行轉換後出現的 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 是根據 PGT 文件編譯裝載的位置自動計算出開始的序號，如果你將 PGT 文件按 \$2000 地址對齊放置，可以簡單理解 PNT0=0。上面的那個公式表示取出標號 _GREENPIECE09_PG 的編譯地址的低 16 位對於 \$2000 的偏移量，再用偏移量除以一個塊的大小，得出第一個塊的序號

Q1.1-004 背景和動畫圖片使用 VT168 的圖形轉換工具 PIC32 進行轉換，轉換後的文件在程序中如何使用 include 和 incbin 包含進來？都需要怎樣設置？在 data.s 中如何包含？在 demo.h 中如何設置？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，因為格式只適用於彙編的語法。在 data.s 中有很多這樣的包含語法，你仔細看看。

Q1.1-005 用 VT168 的圖形轉換工具 PIC32 進行動畫的轉換時，轉出來的數據如下所示（這是一張分辨率為 96*16 的圖片）圖片如下所示：



```
.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
```

請問：這段數據是否要轉化成 Demo 中的 5 列數據：如下（這是一段 Demo 中的動畫數據）

```
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

```

在這段數據中，第 1 列表示索引序號，第 2 列表示的是調色板的序號，第 3 列表示的是 x 坐標，第 4 列表示的是動畫的深度，第 5 列表示的是 y 坐標。那麼轉出來的數據是否也要轉化成此形式，如果是要轉化成此種形式，應該如何把

```

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

```

轉化成 5 列數據？

答：PIC32 轉出的數據其實也是 5 列的，其中“dw PNT0+\$0001” 一個 WORD 表示索引序號和調色板序號。“db \$00,\$00,\$00 ;Y,A,X,0” 表示 Y 坐標，屬性，X 坐標。只是 Y 坐標和 X 坐標的位置跟以前的是相反的。

(2) 如果我把我用工具轉化後的數據：(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

```

把上面這段數據轉化成以下形式，添加前兩列，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

```

前兩列是我添加的，然後把最後一列和第 3 列對換，因為最後一列 Demo 中是 y 坐標，而轉化出來的是 x 坐標，所以進行了調換。結果通過軟件模擬器模擬，動畫是出來了，但是 少一塊數據，少的數據正好是第一塊數據（如下圖所示）。這是為什麼？

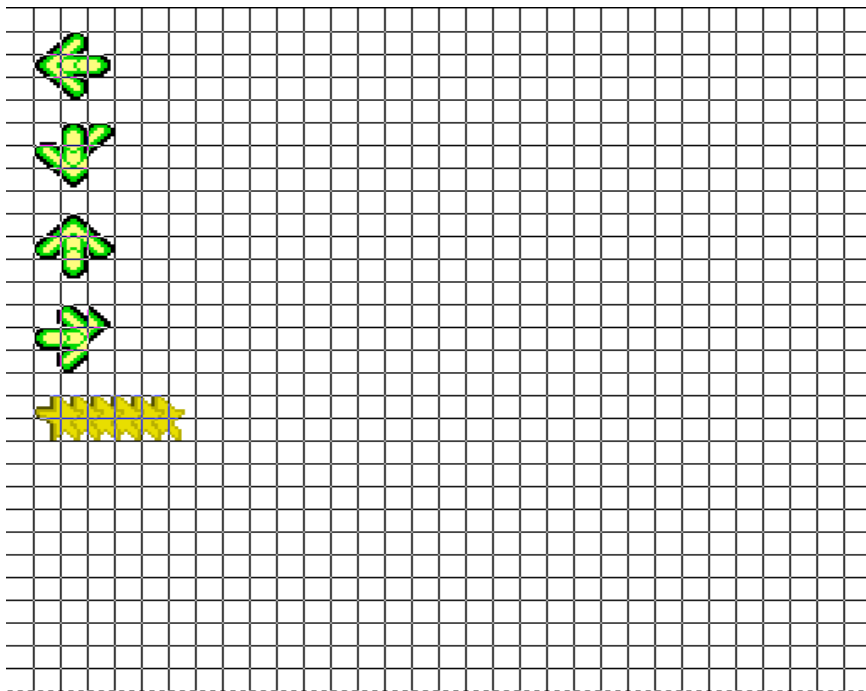


答：你的序號為什麼都是\$11?請注意如果序號為 00 的卡通是不會被顯示的。

(3) 上面是一張分辨率為 96*16 的圖片，按 16*16 進行分塊，應該是 6 塊。可是轉出來的數據是 5 塊，也是少一塊，這是為什麼呢？

答：卡通的作圖是有規則的，請參考下圖：

每一個卡通組，四周必須留空，如果按照 8×8 轉換四周需要留空 8 點（1 格），如果按照 16×16 轉換，四周需要留空 16 點（2 格）



(4) 一組動畫是否可以用不同的色盤？如果可以，如何設置及調用？

答：可以的，如果使用了不同的色盤，轉換出的數據會包含此信息

Q1.1-006 PIC32 轉圖工具轉出的背景(16/64/256 色)數據為浮動的,那麼 DEMO 中有的背景數據為固定的，固定的數據是怎麼得到的？

答:固定的數據可以把 PNT0 設置為 0，或者刪除掉"PNT0+"

1.2 設置相關問題

Q1.2-001 下面的(1)和(2)有何區別？

(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 改為 0xFF80，有什麼區別？

答：FF00 表示傳送 512 字節，FF80 表示傳送 256 字節

Q1.2-002 為什麼打開圖層要和 80 或，如"ora #80\n"；而關閉圖層與 7F 與，如 "and #7F\n"？

答：因為最高位 1 代表打開圖層，為 0 代表關閉圖層

Q1.2-003 在設置動畫的段號時，如下：

```
// 設置 Sprite Segment
//
// 參數 :   DataLable       對應 PGT 的標號
//
#define     SetSprSegment(DataLable) \
           __asm__("lda  #.LOBYTE(%s>>13)\n" \
           "sta _P_Spr_SegmnetL\n" \
           "lda #.LOBYTE(%s>>21)\n" \
           "sta _P_Spr_SegmnetH\n" \
           ,DataLable,DataLable \
           );
```

在這段代碼中，為什麼 s 右移 13 和 21 是何含義？

答：第一個 %s 表示第一個參數“DataLable”，第二個 %s 表示第二個參數“DataLable”，有關嵌入彙編語法請參考編譯器相關文檔

Q1.2-004 加載動畫時，動畫的大小是 16*16。每行加載的數目為什麼達不到 16 個。(只能加載 8 個，超過 8 個就不能顯示出來，如圖 1 (本應加載 12 個，却只加載 8 個，後面 4 個紅色的表示沒有填充))。而每列却可以加載到 15 個，這是為什麼？ 每屏加載的動畫數目而且不能超過某個數目 (<240)，當超過這個數目後就會把前面的動畫充斥掉，如圖 2 (此圖本應上 48 個 16*16 的動畫，而只上了 24 個)。這是什麼原因？應怎麼解決？



圖 1

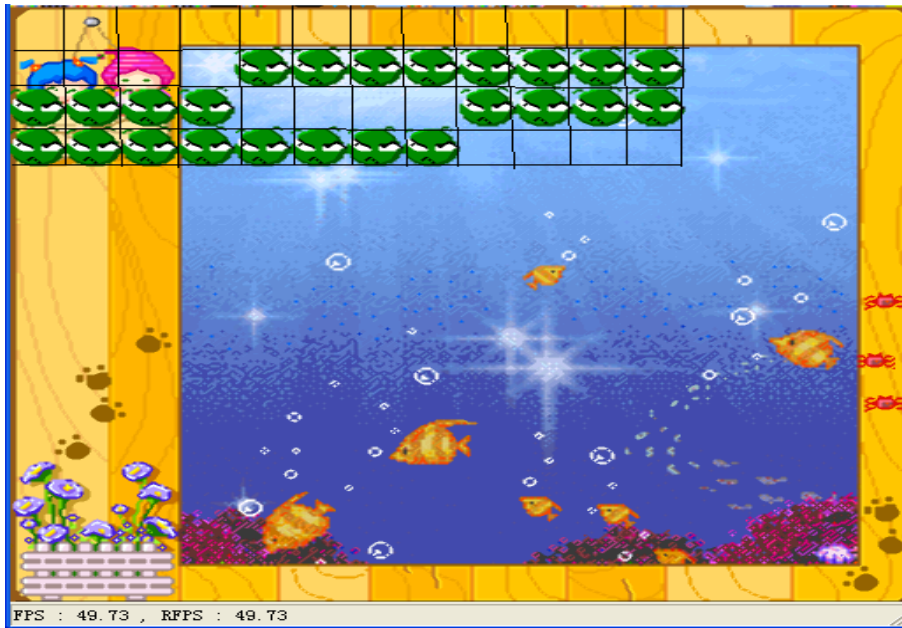


圖 2

答：每行最多是顯示 16 個 Sprite,每列沒有限制顯示多少個 Sprite，Sprite 總數不能超過 240 個。

Q1.2-005 現在使用的是多個卡通組，因此按照上面所計算的，我們的卡通塊肯定達不到 240 個，按照我的理解，動畫是按堆棧的順序排序的，即首先顯示 0xFF，然後是 0xFE，依次遞減的？我現在要實現一列動畫的下移，如下圖所示：



我現在要實現如下功能：

(1) 例如：我現在在第一列，下面有四個相同的，我已經做了刪除的函數，完全可以刪除，現在的問題是：刪除這四個以後，上面的 5 個要依次落下來，在上面空出四個空塊？

我使用的函數是 `SetSpriteXY(jj*16+32,(ii+a)*16+24,0xFD-transfer)`;用它來設置動畫的位置。但是問題是，用了這個函數後，第一次還可以，但是以後就出現了動畫被重疊覆蓋的現象？有時還會花屏？這是為什麼啊？

答：刪除的函數會使序號重新排布，比如序號為 **FC** 的卡通，當刪除了 **FE** 號卡通後，他的操作序號就變為 **FD** 了。建議不要頻繁使用刪除卡通的操作，而盡量使用 **DisableSprite** 和 **SetSprite** 進行覆蓋操作。

(2) 請問如何使用動畫的緩衝和堆棧？

答：動畫的緩衝是根據 **VT1682 Datasheet** 所描述的數據結構相一致的。而堆棧是通過 **InsertSprite** 入棧，他負責描述一個卡通組的坐標和對應的卡通組緩衝數據的地址。現在我只能籠統的講一下，更詳細的需要你看代碼，把看不懂的地方提出來。你可以用模擬器 **DEBUG** 并觀看內存變量有助你理解。

Q1.2-006 現在要在同一位置，比如在我的程序裏，我現在要在同一位置顯示四個動畫，例如最下層是磚塊，磚塊上是蛋糕，蛋糕上是光標，光標上要顯示其它東西，那麼我該如何設置它們的優先級呢？

答：有兩種方式：

- 1.請修改屬性字節，即 **A** 那個字節
- 2.同一屬性的卡通，在 **Buffer** 位置越靠前優先級越高

Q1.2-007 在本程序中要顯示 `showheart()`；`showmoli()`這兩個動畫，`showheart()`是顯示心的動畫，`showmoli()`是顯示魔力棒的動畫，但是現在和我前面的動畫一同顯示，就出現花屏，但是如果單獨顯示，一切正常，這是為什麼呢？

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

Q1.2-008 **BG1,BG2,SP** 是否共用兩個 256 色盤？如果 **BG1** 用了 256 色，**BG2** 也用了 256 色，那麼動畫的顏色在哪兒呢？透明色由哪一個色盤決定？

答:如果 **BG1** 用了 256 色，**BG2** 也用了 256 色，那麼動畫的顏色也必須在這 256 中。兩個調色板的第一個顏色是透明色，如果兩個疊加的背景都是透明色，則顯示級別低的背景對應的透明色的顏色。

Q1.2-009 請問動畫的深度應該如何控制，假如我們現在在屏幕同一位置需要顯示四個動畫，怎麼樣控制它們的顯示順序？根據資料介紹，是設置深度 0，1，2，3，0 在最上層，可是我們如何設置此深度呢？是通過動畫的數據嗎？如果是，我們應該如何設置此數據呢？

答：卡通的深度是通過卡通數據 **Layer** 的這兩位來控制的

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

Q1.2-010 如果在變量前加 **const** 修飾符，是否可以節省內存？如何把自己的數據或變量放在指定的段中，如 **code** 段、**RODATE** 段、**DATA** 段以及 **BSS** 段？

答：變量前加 **const** 修飾符是不占用 **RAM** 內存的。在彙編中用段修飾符可以把自己的數據或變量放在指定的段中。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
```

是把數據放到 **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
```

是把變量放到 **ZEROPAGE** 段

Q1.2-011 動畫一屏幕也只能顯示 128 個（按 16*16），這是為什麼？

答：您使用了 **InsertSprite** 函數，此函數在設計的時候是按照以下思路設計的：

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

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

SpriteStack 存放的是卡通組的索引數據，每一組數據由 8 個字節表示，以棧的方式存儲而 **SpriteBuffer** 存放的是單個卡通塊的數據，跟 **Datasheet** 上所講的結構是一致的。如果你簡單的使用 16*16 的卡通，我們算一下理論的最大卡通數：

$X \times 8 + X \times 6 = \600 $X = 109$, 所以以緩存不溢出的情況下最大顯示 109 個卡通

建議您使用卡通組的數據結構

Q1.2-012 關於上面這個問題所衍生的小問題如下：

- (1) 如何使用卡通組的數據結構？
- (2) $X \times 8 + X \times 6 = \600 這個公式中 8 和 6 各表示什麼含義？
- (3) 目前我的代碼可以上到 128 個動畫？
- (4) 請問這個動畫的緩衝和堆棧可否自己設置？

答：按照以前說的 **Sprite** 的作圖方法作圖，通過 **PIC32** 轉換，按照 **Sprite** 格式輸出的就是卡通組的數據結構。8 代表卡通組數據由 8 個字節表示，6 代表每一個卡通塊由 6 個字節表示。

使用動畫的個數是要看你怎麼樣用，不是絕對的，比如你只有一個卡通組，這個卡通組有 240 個卡通塊，那麼你完全可以使用這 240 個卡通塊，如果你有 N 個卡通組，每個卡通組有 1 個卡通塊，那麼 $n \times 8 + n \times 6 = \600 ， $n=109$ ，就是說最多 109 個卡通可以使用，超過了就出錯了。

這個動畫的緩衝和堆棧當然可以自己設置，因為他是程序實現的，你修改我的源代碼程序就可以了，具體怎麼修改需要你看懂我的那段源代碼。

Q1.2-013 關於上面這個問題所衍生的小問題如下：

- (1) 請問修改這個屬性，即 A 那個字節，是不是這樣設置，它不是有四個深度嗎？是不是 \$00 表示第 0 層；\$10 表示第 1 層；\$20 表示第 2 層；\$30 表示第 3 層呢？
- (2) 我現在采用的是你的第二種方法，即同一屬性的卡通，在 Buffer 位置越靠前優先級越高，但是當動畫多時，就出現了花屏，可能是 `_SpriteBuffer` 和 `_SpriteStack` 溢出造成的，但我現在最大只能顯示 128 個呀？

答：是這樣的，屬性字節最高兩位沒用，由高到低依次為：
Depth[1:0],Flip[1:0],VRCH,PalSel

按照我上面的計算，超過 109 個卡通就會不正常了

2. 軟件相關問題

2.1 程序編寫

Q2.1-001

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

_PaletteBuffer   =  $0800    ;Palette Buffer
```

請問：上段代碼中\$0200、\$07A0、\$0800 三個地址是 VRAM 中的嗎？若不是，那請問它們是哪個 ROM 或 RAM 中的地址？

答：不是的，是 PRAM 中的定義的 BUFFER

Q2.1-002 請問在哪里設置 code 段、RODATE 段、DATA 段以及 BSS 段？

答：在 CFG 文件中設置這些數據段

Q2.1-003 程序中的 export,import 是對什麼而言的？

答：export 是聲明一個標號為外部的可供在另一個文件中調用這個標號，import 是引用一個聲明為外部的標號。詳細資料請參考相關文檔，在 www.vrt.com.tw 有下載

Q2.1-004 在程序編寫過程中會出現此錯誤提示：ld65.exe:Error: Memory area overflow in 'PROM',segment 'STARTUP'<743bytes0> ,請問如何解決此錯誤？或者是導致這種錯誤的原因是什麼？當前，代碼還不到 700 行，就發生溢出，當我們刪除一些程序或變量，這個問題就可以解決，但是隨著程序的增加，此問題又出現了？是不是要在哪里設置？還是要切 BANK？如果是要切 BANK，那麼應該如何切呢？根據資料介紹，它的物理地址綫是 25 根，那麼至少應該尋址 32 M 的空間，那麼為什麼現在這麼少的程序，就會發生溢出呢？

答：你所說的 32M 是物理地址，對於 VT168 來說他的綫性尋址空間是 48K (\$4000-\$FFFF) .對於物理地址的操作需要映射頁面 (switch bank)，映射頁面需要通過設置相關寄存器，參見 datasheet 有關 Program_Bank_Register 的描述。在 cfg 文件中定義了 'PROM' 的大小，DEMO CODE 中定義的是 start = \$C000, size = \$3FFA，DEMO CODE 中的 CODE，DATA，STARTUP 段都是裝載在這一空間，如果是 C 寫的 CODE，700 行有可能溢出。

Q2.1-005 在 cfg 文件中改變了 'PROM' 的大小，我把它改爲 start=\$C000,size=\$5FFA；結果出現了此錯誤提示：ld65.exe:Error:Range error in module'Demo.s',line 353.我現在已經把 showheart(); showmoli();這兩段代碼屏蔽了，還有 if((a>2) || (b>2) || (c>2) || (d>2))這個條件檢測也屏蔽了，以及關於 a,b,c,d 的定義和賦值都屏蔽了。目前這段代碼是沒有錯誤的，只有幾個警告。現在只要把屏蔽打開，就會出現 'PROM' 的溢出？

答：還是設置有問題，應該設置為：

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

Q2.1-006 承上個問題：

a (1) 按上面的設置，修改了 CFG 文件中的設置，'PROM' 不會發生溢出了。但

是當代碼增多的時候，試過再增加一倍的代碼，原來是 700 行代碼，當讓它增加到 1400 行時候，又出現了‘PROM’溢出，這時候又在 CFG 文件中進行了設置，增大了它的空間，

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

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

這時不會出現‘PROM’溢出了，但是把 rom.bin 加到模擬器上沒有任何效果，出現了黑屏，什麼都沒有，這是什麼原因啊？

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

表示標號地址從\$1000 開始，size 是\$CFFA,如果程序按照這樣編譯，編譯後的運行地址就不對了，程序當然就不可能運行正確了。

按照我們以前的設想，常用的代碼是使用 CODE 段，裝載在 PROM 空間，即將物理地址\$7FC00-\$7FFFA 映射到\$C000-\$FFFA 運行，CODE 段在 CFG 文件中的定義如下：

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

另外再定義若干個 CODE0，CODE1...這樣的段

CODE0 段在 CFG 文件中的定義如下：

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

這個定義的意思是，編譯按 CODES (\$A000-\$BFFF) 所定義的地址編譯，編譯後的數據裝載到 DATAS 定義的空間，數據的起始是按照\$2000 對齊的。

然後在 C CODE 中用 #pragma codeseg ("CODE0") 強定義把不常用的代碼編譯到 CODE0 段，在調用前需要通過置\$2108 把物理地址映射到\$A000-\$BFFF 然後再調用函數。

當初我們的設想是基于：

\$C000-\$FFFA 作為公用程序運行空間,上電自動將物理地址\$7FC00-\$7FFFA 映射到\$C000-\$FFFA

\$A000-\$BFFF 作為調用子程序運行空間

\$8000-\$9FFF 作為訪問數據空間

現在 VT1682 又擴充了\$4000-\$7FFF 可以直接訪問的地址空間，所以你可以按照你的思維自行安排你的程序空間架構

Q2.1-007 在 cfg 文件中改變了‘PROM’的大小，按照上面的回答，即設置為

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

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

但是現在增加了一些程序後，又出現了‘PROM’溢出？

一開始，cfg 文件中的設置為

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

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

但是在代碼快到 700 行時，出現了‘PROM’溢出。於是按照上面的回答，把 PROM 的大小增加了\$2000，結果就解決了‘PROM’的溢出。但是當現在代碼編寫到快 1000 行時，又出現了‘PROM’的溢出？於是又在 cfg 文件中進行了設置，設置如下

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

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

這時不會出現‘PROM’溢出了，但是把 rom.bin 加到模擬器上沒有任何效果，出現了黑屏，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 與并且還要和 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 還是 8 bytes?
sta $2127

```

答：DMA 可以只傳一個 PNT,最小單位是一個 Word(兩個字節)
lda #\$03 是 6 bytes

Q2.2-004 DMA 期間, CPU 停止工作,會不會被中斷打斷？

答：不會

2.3 編譯器相關問題

Q2.3-001 請問, 如下：

```

#defineSetText1Attibe(TextSwitch,Color,Size,Bitmap,PALSel,Depth,Scroll,PALS
elect)
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 \
```

在這段代碼中,為甚麼要和 03 與後,又和 b 或呢?
,Scroll,(unsigned)(TextSwitch+Depth+Color+Bitmap+Size),PALSelect 這句是何含義?

答：與上 03 表示保留低兩位，%b 表示參數值，第一個 %b 表示“Scroll”，第二個 %b 表示“(unsigned)(TextSwitch+Depth+Color+Bitmap+Size)”，第三個 %b 表示“PALSelect”，有關嵌入彙編語法請參考編譯器相關文檔

2.4 乘除寄存器

Q2.4-001 用乘，除寄存器時，該等多長時間才能讀結果？

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

2.5 結構體

Q2.5-001 在使用標準 C 的結構體時,結構體佔用空間較多時會出現此錯誤提示:
 ld65.exe:Error: Memory area overflow in 'RAM',segment 'BSS'<743bytes0> ,請問如何解決此錯誤?或者是導致這種錯誤的原因是什麼?

答：你的結構應該定義到 **RODATE**，**DATA** 或 **CODE** 段，不應該定義到 **BSS** 段，即在 **C** 中加 **const** 修飾符。例如：
char const MenuTitle[] = "1000 in 1";

按照了上面的說法，在結構體前加 **const** 關鍵字，但是結構體較大時，仍然報錯，例如定義了此結構：如下所示：

```
typedef struct                //定義方塊的結構體
{
    int fill;                  //是否填充，也就是是否有動畫，1 為有，0 為沒有
    int statu;                 //方塊的狀態
}box;
box screen[9][11];
```

當定義為 **box screen[12][12]**，它就報錯。出現的錯誤提示為：**ld65.exe:Error:**

Memory area overflow in 'RAM',segment 'BSS'<1333bytes0> ,如何解決此錯誤?或者是導致這種錯誤的原因是什麼?為什麼結構體較大時,就會出現此錯誤?如何避免和解決呢?

答：因為聲明的結構數據是放在 RAM 中的，而 RAM 的分配的空間很小，所以會出現溢出錯誤。你應該這樣：

```
typedef struct                //定義方塊的結構體
{
    int fill;                 //是否填充，也就是是否有動畫，1 為有，0 為沒有
    int statu;               //方塊的狀態
}box;
box const screen[2] = {
    {0,0},
    {1,1}
};
```

表示放在 ROM 的區域，你可以觀察 C 文件編譯對應的.S 文件，編譯的結果是：

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

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

2.6 函數 BUG

Q2.6-001 底下是一個下移函數。這個函數有什麼 BUG？

```
//下移的函數
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;
```

VT1682 Q & A

```
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
{
    a=0;
    goto kk;
}
}
}
}
```

a 是要下移的行數。

(2) 這個遊戲是，首先按任意鍵進入 logo，然後按方向鍵進行選擇，按 start 鍵進行第一次選擇，在按一次 start 鍵進行第二次選擇，這時這兩個就實現了交換（只能是相鄰的才能進行選擇和交換）只要行和列相同的個數大于 2 就消除，消除後下移。

請問：當循環與條件語句嵌套較多時，會不會出現死機現象？什麼情況下會發生死機現象？

答：當循環與條件語句嵌套較多時，這個“較多”是什麼概念？如果你做 100 次的循環，那麼嵌套一次就是 100*100 次循環，如果嵌套兩次就是 100*100*100 次循環，如果有 10 次嵌套的話我想就跟死機的效果差不多了，程序運行跳出這個循環嵌套大概需要幾十分鐘了。

2.7 內存映像相關問題

Q2.7-001 內存映像部分

\$4000--\$FFFF " 是映像到 6 個程序存貯空間去擴充地址綫到 8M 字節的外部內存 " 用的是哪種映像方式？ 每個 BANK 是多大空間？

答：每個 BANK 為\$2000

2.8 地址線常見問題

Q2.8-001 在 VT1682 使用手冊中 4.2 節 地址方式中

PQ2EN COMR6 EXT2421 PA20--PA13 TP20--TP13
PQ37--PQ30 各指的是那些地址綫？

答：作為初學者一開始不需要瞭解這麼深，只需要用默認的地址方式就可以了，這些地址方式是兼容 VT03 的 C3 模式，只有合卡時才會採用。

2.9 VRAM 相關問題

Q2.9-001 通過\$2004,\$2007 讀 VRAM，是否也要先忽略第一個字節？

```
lda #$08
sta $2006
lda #$00
sta $2005
lda $2007 ;ignore???
lda $2007 ;$800,first byte????
lda $2007 ;$801,????
```

答：VT1682 無法做 VRAM / Sprite RAM 的讀取

3. 硬件相關問題

3.1 XIOF

Q3.1-001 XIOF3 能不能改變狀態成為輸入(input)模式？

答：不能。

Q3.1-002 當使用 XIOF0 和 XIOF1 當作輸出(output)時的錯誤該如何解決？

答：它們被接到此開發板的 FPGA。如果輸入(input)模式是需要的，請將 FPGA 和 VT1682 的這兩條連線切斷開即可。

3.2 DMA

Q3.2-001 為何從內部的 RAM(internal RAM)DMA 到外部的 RAM(external RAM)時會出現錯誤？到內部的 RAM(internal RAM)時的錯誤，該如何處理？

答：當 DMA 路徑是從內部的 RAM 到外部的 RAM(external RAM)時，在地址線是有一個限制的。這個目標和來源字節地址線 A[12]，必須是相同的。從內部的 RAM(\$0000~\$1FFF) DMA 到內部的 RAM(\$0000~\$1FFF) 是不允許的。

Q3.2-002 可以跨 Bank(Bank crossing)嗎？

答：DMA bank 大小是 64K 字節。DMA 地址不可以跨過這個 Bank。

3.3 PPU

Q3.3-001 在背景 2(BK2)256 色模式圖形(Graphic)會出現錯誤嗎？水平方向的(橫向)位置兩個點的飄移(two pixels shift)，會發生嗎？卡通塊(Sprite)水平方向的橫向跳動功能是無效的？

答：不會。不會。無。

3.4 EXIT IRQ

Q3.4-001 為何主 CPU(Main CPU)外部的 IRQ 是高準位觸發而不是負的邊緣觸發(negative edge trigger)？

答：使用一個外加的 74xx74 或是使用 SCPU 外部的 IRQ(SCPU_UIOB[7])。

3.5 LCD

Q3.5-001 當在\$200C 的 LCD_CLK 置為 3 時,在 LCD 顯示屏上會有一些黑點嗎？

答：不會。

3.6 AUDIO DAC

Q3.6-001 當播放 Audio 的時後會有一些噪音(noise)，該如何解決？

答：避免從 Audio DAC 的最高位(MSB)開始使用。

3.7 ADC

Q3.7-001 在 ADC 端口的 XIOF3 為何會是無效的？當使用 XIOF0 和 XIOF1 當作 ADC 端口時的錯誤該如何改善？

答：使用 IOE3 來取代。它們被接到此開發板的 FPGA。如果 ADC 模式是需要的，請將 FPGA 和 VT1682 的這兩條連線切斷開即可。

3.8 CCIR

Q3.8-001 在 YUV 模式,有一些高明亮度的(High luminance)顏色是錯誤的顏色，改善的方式？

答：使用 CCIR RGB 介面。