

```

; raw frame for generic 64KB cartridge images
; v 1.0 enthusi 06/2012
; this 64 KB Cartridge framework was written for http://www.rgcd.co.uk
; feel free to use/change this code and give credits :)
; you will find this document also at http://codebase64.org
; this is a VERY simple but efficient approach, you can make more
; sophisticated usage of ROM using an own depacker routine etc....
; sources are in XA format but no special features are used
; I strongly recommend the usage of cartconv which comes with vice
;-----
; recent builds of VICE support the RGCD cartridge format,
; grab it at http://vice.pokefinder.org
;-----
; example usage:
; xa -M frame64kb.asm -o game_raw.bin
; cartconv -t rgcd -i game_raw.bin -o game.crt
; x64sc -cartcrt frame.crt
;-----

```

```

;no load-address for bin-file, so no header here
*=$8000

.word launcher ;cold start
.word launcher ;warm start
.byte $c3      ;c
.byte $c2      ;b
.byte $cd      ;m
.byte $38      ;8
.byte $30      ;0

launcher
    stx $d016
    jsr $fda3    ;prepare irq
    jsr $fd50    ;init memory
    jsr $fd15    ;init i/o
    jsr $ff5b    ;init video
                ;make sure this sets up everything you need,
                ;the calls above are probably sufficient

    ldx #$fb
    txs

;clear screen and set to black
    lda #0
    ldx #250
clearloop
    sta $d800-1+250*0,x
    sta $d800-1+250*1,x
    sta $d800-1+250*2,x
    sta $d800-1+250*3,x
    dex
    bne clearloop

```

```
sei
lda #$00
sta $d020
sta $d021
    lda #$37
    sta $01
;-----
;prepare cart-loader in stack
init_mover
    ldx #$00
loop1  lda mover,x
    sta $0100,x
    inx
    cpx #(mover_end-mover)
    bne loop1
    ;display RGCD-logo
    jsr setup_logo
    jmp $0100
;=====
mover
    lda #$00
    sta $fc
    lda #$89
    sta $fd

    lda #$01
    sta $fe
    lda #$08
    sta $ff

    ldx #$17
loop2  ldy #$00
loop3  lda ($fc),y ;8900
    sta ($fe),y ;0800
    iny
    bne loop3
    inc $fd
    inc $ff
    dex
    bne loop2

;continue on bank 1
    lda #$01 ;de00 =1
    sta $fb
;-----
loop4  sta $de00

    lda #$00
    sta $fc
```

```

    lda #$80
    sta $fd

    ldx #$20
loop5  ldy #$00
loop6  lda ($fc),y ;8000 (start of bank)
    inc $01
    sta ($fe),y ;continue from 0800 on...
    dec $01
    iny
    bne loop6 ;1 page

    inc $fd
    inc $ff
    dex
    bne loop5 ;32 pages = 8 KB

    inc $fb ;0-7
    lda $fb
    cmp #$08
    bne loop4

    sei
    lda #$08 ;kill cart
    sta $de00
    ldx #$ff
    txs
    lda #$00
    tax
    tay
    lda $d011
    and #%11101111
    sta $d011

    ;launch main program, configure this adress here and make sure
    ;to have ZP, I/O, VIC-REGS properly set up!
    jmp $80d

mover_end

;-----
#define OFF 7
setup_logo
.(
    ldx #0
sll1
    lda text,x
    sta $f000+40*(1+OFF),x
    inx
    bne sll1

```

```
sll2
  lda text+$100,x
  sta $f000+40*(1+OFF)+$100,x
  inx
  bne sll2
```

```
;charset
```

```
csl
```

```
  lda charset+$100*0,x
  sta $f800 +$100*0,x
```

```
  lda charset+$100*1,x
  sta $f800 +$100*1,x
```

```
  lda charset+$100*2,x
  sta $f800 +$100*2,x
```

```
  lda charset+$100*3,x
  sta $f800 +$100*3,x
```

```
  lda charset+$100*4,x
  sta $f800 +$100*4,x
```

```
  lda charset+$100*5,x
  sta $f800 +$100*5,x
```

```
  lda charset+$100*6,x
  sta $f800 +$100*6,x
```

```
  dex
  bne csl
```

```
;colors
```

```
  ldx #$28
```

```
cl2
```

```
  lda #9
  sta $d800+40*(1+OFF),x
```

```
  lda #2
  sta $d800+40*(2+OFF),x
```

```
  lda #4
  sta $d800+40*(3+OFF),x
```

```
  lda #14
  sta $d800+40*(4+OFF),x
```

```
  lda #3
  sta $d800+40*(5+OFF),x
```

```
  lda #13
  sta $d800+40*(6+OFF),x
```

```
  lda #1
  sta $d800+40*(7+OFF),x
```

```
  lda #6
```

```
    sta $d800+40*(8+0FF),x
    dex
    bpl cl2

    lda #6
    sta $d800+40*(7+0FF)+18
    sta $d800+40*(7+0FF)+19
    sta $d800+40*(7+0FF)+20
    sta $d800+40*(7+0FF)+21
    sta $d800+40*(6+0FF)+19
    sta $d800+40*(6+0FF)+20

    lda #%11001110
        sta $d018
    lda #%00010100
    sta $dd00
    rts

text
    .bin 2,0,"logo.screen"
charset
    .bin 2,0,"logo.bitmap"
after_charset
.)
;-----
.dsb $8900-*,0
.bin 2,0,"heartlight.prg"
e1
.dsb $18000-*,0 ;fill up to complete 65535 Bytes
e2
```

You find this file and the 3 addition files "logo.screen" "logo.bitmap" "heartlight.prg" In this zip-file:
[frame64kb.zip](#)

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
https://codebase64.org/doku.php?id=base:code_frame_for_64_kb crt-images_i.e._for_rgcd

Last update: **2015-04-22 19:55**

