REU Detect

```
;------------------------------------------------------------------------
----------------------------------------
; REUDETECT v1.0a ( REU : 1700, 1764, 1750, 1024Kb and 2048Kb )
; ----------------------------------------------------------------------
----------------------------------------
; Overview : Detection of REC and RAM-Type : 0 = 1700 (64Kbx1) or 16 =
1764/50 or bigger(256Kbx1).
;            Write to Registers 2-5 and compare.
;            Write 33 banks with "messy datas".
;            Fetch bank, inc bank-counter (banks $1500) if own dummy-bytes
not found.
;            Stash bank, write own dummy bytes for later comparing and fetch
next bank.
;            Skip bankcheck if dummy byte-chain found. Detection is
finished.
;            Read available banks, evaluate and drop some text on the
screen.
; ----------------------------------------------------------------------
----------------------------------------
; 21. Januar 2005 M. Sachse (cbmhardware/People of Liberty)
;
; E-Mail : info(at)cbmhardware.de
;
; 22. Januar 2005 : Bugfix, add ram-type detection 1764, Vice and C64
compatibility, messy code reworked;
;
; ----------------------------------------------------------------------
----------------------------------------
; GPL
;
; This program is free software; you can redistribute it and/or modify it
under the terms of the
; GNU General Public License as published by the Free Software Foundation;
either version 2 of the
; License, or (at your option) any later version.
;
;------------------------------------------------------------------------
----------------------------------------
; Source Code for ACME Cross-Assembler :
;------------------------------------------------------------------------
----------------------------------------
!to "reudetect.prg"

*= $0800

!byte $00,$0c,$08,$0a,$00,$9e,$32,$30,$36,$32,$00,$00,$00,$00
```

```
*= $080e

reubase      = $df00
reucommand   = $df01
banks        = $1500     ; to store found banks and count


;--------------------------------------------------
; Detect REU 1700,1764/50 or 1/2MB
;--------------------------------------------------
reudetect    lda #1
             sta $0286           ; textcolor white
             lda #00
             sta $d020
             sta $d021
             sta banks
             sta reubase
             cmp reubase
             beq noreu
             bne unsafe
unsafe       lda reubase
             and #16          ; check bit 4 for REU mem
             cmp #16          ; 16 = 256Kbx1
             beq regcheck     ; yes, touch registers
             bne l1           ; no, 1700 ?
l1           lda reubase
             and #16          ; check bit 4 for REU mem
             cmp #0
             beq r1700        ; reu 1700 found
             bne noreu        ; no ram-type, no reu, no fun ...
regcheck     lda reubase
             ldx #2
loop1        txa
             sta $df00,x      ; write to registers 2-5
             inx
             cpx #5
             bne loop1
             ldx #02
loop2        txa
             cmp $df00,x
             bne noreu
             inx
             cpx #5
             bne loop2
             jmp rinit
r1700        lda #<reutext
             ldy #>reutext
             jmp $ab1e
;--------------------------------------------------
rinit        ldx #00          ; 1764 wake up
rinit2       lda #128         ; stash
```

```
             sta config
             lda #$12          ; write some crap in ...
             sta c64hi+1
             stx bank+1
             jsr main
             inx
             cpx #33           ;  ... 33 banks into somewhere
             bne rinit2
             jmp action
noreu        lda #<notext
             ldy #>notext
             jmp $ab1e
;--------------------------------------------------
; Count banks
;--------------------------------------------------
action       lda reubase
             ldx #$00
             stx bank+1        ; reset bank counter
check        lda #129          ; fetch : transfer to C64 : $1300
             sta config
             lda #$13          ; C64 : $1300
             sta c64hi+1
             jsr main
             lda #128          ; stash
             sta config
             lda #$0A          ; write dummy bytes from $0900
             sta c64hi+1
             jsr main
             jsr bankcheck     ; check for existing ram banks
             inx
             cpx #33           ; try 33
             stx bank+1
             bne check
;--------------------------------------------------
             lda #0            ; restore bordercolor
             sta $d020
             lda banks         ; banks found ?
             cmp #4
             beq r1764
             bne j1
r1764        lda #4
             sta banks
             lda #<text1764
             ldy #>text1764
             jmp $ab1e
j1           cmp #8
             beq r512
             bne j2
r512         lda #<reut512
             ldy #>reut512
             jmp $ab1e
```

```
j2          cmp #16
            beq r1024
            bne j3
r1024       lda #<reut1024
            ldy #>reut1024
            jmp $ab1e
j3          cmp #20
            beq r1764
            bne j4
j4          cmp #32
            beq r2048
            bne j6
r2048       lda #<reut2048
            ldy #>reut2048
            jmp $ab1e
j6          lda #<reuunk
            ldy #>reuunk
            jmp $ab1e
;------------------------------------------------
; Bank Check
;------------------------------------------------
bankcheck   ldy #$00
l2          lda $0A00,y
l6          cmp $1300,y
            bne l4            ; bank found ?
            beq l3            ; no, check 16 Bytes
l3          iny
            cpy #16
            bne l2            ; loop
end         ldy #00
            lda #00
delete      sta $1300,y       ; delete buffer
            iny
            cpy #16
            bne delete
            rts
l4          jmp l5
l5          inc banks         ; bank found (inc), border color change and
exit
            inc $d020
            rts
config !byte 252
;------------------------------------------------
; Bytes and text
;------------------------------------------------
*=$0A00

!scr "---C64-RULEZ!---"         ;  ;-)
reutext:  !text "REU 1700 : 128KB DETECTED",0
text1764: !text "REU 1764 : 256KB DETECTED",0
reut512:  !text "REU 1750 : 512KB DETECTED",0
```

```
reut1024: !text "REU 1024KB DETECTED",0
reut2048: !text "REU 2048KB DETECTED",0
reuunk:   !text "REU PORT DETECTED - BANK ERROR",0
notext:   !text "NO REU",0

;------------------------------------------------
; REU TRANSFER ROUTINE
;------------------------------------------------

main
           lda config
           sta reubase+1
           lda #$00
           sta reubase+2
c64hi      lda #$09
           sta reubase+3
           lda #$00
           sta reubase+4
           sta reubase+5
bank       lda #0
           sta reubase+6      ; Bank
rbytes     lda #16
           sta reubase+7      ; 16 Bytes
           lda #$00
           sta reubase+8
irq        lda #$00
           sta reubase+9
           lda #$00
           sta reubase+10
           lda $1
           pha
           lda #$30           ; RAM
           sei
           sta $1
           sta $ff00
           pla
           sta $1
           cli
           rts
```