

C= 1351 Standard Mouse Driver

The code is modified from it's original format to KickAssembler format by TWW

The routine alters \$d000 & \$d001. It shuldn't be too hard to modify it.

There are some enhancments possible as well. Self calibration and even limiting the returned alteration to a signed byte since the movements never exceeds 6 bits...

The rest of the routine should speak for itself...

```
.var iirq    = $0314
.var vic     = $d000
.var sid     = $d400
.var potx    = sid+$19
.var poty    = sid+$1a
.var vicdata = $d000      // basics copy of vic register image
.var xpos    = vicdata+$00 // low order xposition
.var ypos    = vicdata+$01 // y position
.var xposmsb = vicdata+$10 // bit 0 is high order x position

    lda potx // get delta values for x
    lsr //CCS64 fix (remove for other emus/real HW)
    ldy opotx
    jsr movchk
    sty opotx

    clc // modify low order xposition
    adc xpos
    sta xpos
    txa
    adc #$00
    and #%00000001
    eor xposmsb
    sta xposmsb

    lda poty // get delta value for y
    lsr //CCS64 fix (remove for other emus/real HW)
    ldy opoty
    jsr movchk
    sty opoty

    sec // modify y position ( decrease y for increase in pot )
    eor #$ff
    adc ypos
    sta ypos
    dec $01
    rts

//=====
```

```
// movchk
//      entry   y = old value of pot register
//              a = current value of pot register
//      exit    y = value to use for old value
//              x,a = delta value for position
//=====
movchk:
    sty oldvalue
    sta NewValue
    ldx #0

    sec
    sbc oldvalue
    and #%01111111
    cmp #%01000000
    bcs !+
    lsr
    beq !Slutt+
    ldy NewValue
    rts

!: ora #%11000000
   cmp #$ff
   beq !Slutt+
   sec
   ror
   ldx #$ff
   ldy NewValue
   rts

!Slutt:
    lda #0
    rts

opotx: .byte $00
opoty: .byte $00
```

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

Permanent link: https://codebase64.org/doku.php?id=base:c_1351_standard_mouse_routine

Last update: **2015-04-17 04:30**

