SilverSurfer polling mode driver

```
;------------------------------------------------------------------------
------------------
; SilverSurfer polling mode driver                                    (w)
2001 Groepaz/Hitmen
; - this one is optimized for size and reliability (rather than maximum
speed or efficiency)
;------------------------------------------------------------------------
------------------
; ss_detect             detect silversurfer, returns CARRY=1 on error
; ss_init               init uart, call with baudrate divisor lowbyte in A
; ss_getchr             get byte (blocking) - returns next byte in A
; ss_getchr2            get byte (non blocking, returns byte in A or CARRY=1
- error)
; ss_putchr             sends byte in A
;------------------------------------------------------------------------
------------------
; all functions except ss_detect and ss_init use/modify AKKU only.
;------------------------------------------------------------------------
------------------

;.include "silversurfer.inc"  ; fifo defines

                        .export ss_detect,ss_init
                        .export ss_getchr,ss_getchr2,ss_putchr

                        .export ss_stat_reciever_ready
                        .export ss_stat_data_available
                        .export ss_stat_transmit_ready
                        .export ss_pause,ss_unpause

;------------------------------------------------------------------------
------------------
; detect silversurfer, returns CARRY=1 on error
;------------------------------------------------------------------------
------------------

ss_detect:

        ;
        ; enable ssurfer-port
        ;

        lda $de01
        ora #$01
        sta $de01

        ;
```

```
         ; check if scratch-register available
         ;

         ldx #$00
@l1:
         stx fifo_scratch
         cpx fifo_scratch
         bne @sk1

         inx
         bne @l1

         ; ok
         clc
         rts

@sk1:
         ; error
         sec
         rts

;---------------------------------------------------------------------------
-------------------
; init uart, call with baudrate divisor lowbyte in A
;---------------------------------------------------------------------------
-------------------
;
;   9600      $30
;  19200      $18
;  38400      $0c
;  57600      $08
; 115000      $04
;
;---------------------------------------------------------------------------
-------------------

ss_init:
         tax

         ;
         ; enable ssurfer-port
         ;

         lda $de01
         ora #$01
         sta $de01

         ;
         ; init and set baudrate
         ;
```

```
         lda #%10000111
         sta fifo_fcr

   ; that delay thing really needed ?!
   ; (original datasheet mentions a delay here)
   ;      ldy #$00
   ;      dey
   ;      bny *-1

         ; set dlab
         lda #%10000011
         sta fifo_lcr

         ; set baudrate

         stx fifo_dll
         lda #$00
         sta fifo_dlm

         ; reset dlab
         lda #%00000011
         sta fifo_lcr

         ; disable nmi's from ssurfer
         lda #%00000000
         sta fifo_ier

         ; activate dtr
         jmp ss_pause

;-------------------------------------------------------------------------------
-------------------
; check if data available
; no data:   AKKU=0
; data:      AKKU!=0
;-------------------------------------------------------------------------------
-------------------
ss_stat_data_available:
         lda fifo_lsr
         and #%00000001
         rts
;-------------------------------------------------------------------------------
-------------------
; check if transmit register empty
; busy:     AKKU=0
; empty:    AKKU!=0
;-------------------------------------------------------------------------------
-------------------
ss_stat_transmit_ready:
         lda fifo_lsr
         and #%00100000
```

```
         rts
;-------------------------------------------------------------------------
------------------
; check if reciever ready
; busy:    AKKU=0
; ready:   AKKU!=0
;-------------------------------------------------------------------------
------------------
ss_stat_reciever_ready:
         ; check cts
         lda fifo_msr
         and #%00010000
         rts
;-------------------------------------------------------------------------
------------------
; Flow-Control
;-------------------------------------------------------------------------
------------------
ss_unpause:
         ; activate rts
         lda #%00000011
         sta fifo_mcr
         rts
ss_pause:
         ; deactivate rts
         lda #%00000001
         sta fifo_mcr
         rts
;-------------------------------------------------------------------------
------------------
; get byte (non blocking, returns byte in A or CARRY=1 - error)
;-------------------------------------------------------------------------
------------------
ss_getchr2:
         ; check if byte available
         jsr ss_stat_data_available
         bne sk32 ; yes

         ; release flow-control
         jsr ss_unpause
sk32:
         ; set flow-control
         jsr ss_pause

         ; check if byte available
         jsr ss_stat_data_available
         bne sk33 ; yes
         sec
         rts
sk33:
         ; get byte
```

```
         lda fifo_rxd
         clc
         rts

;------------------------------------------------------------------------
------------------
; get byte (blocking) - returns next byte in A
;------------------------------------------------------------------------
------------------

ss_getchr:

@lp1:
         jsr ss_getchr2
         bcs @lp1 ; no byte read
         rts

;------------------------------------------------------------------------
------------------
; send byte in A (blocking) - waits until byte can be send
;------------------------------------------------------------------------
------------------

ss_putchr:
         pha

         ; wait until reciever is ready to recieve
l2:
         jsr ss_stat_reciever_ready
         beq l2

         ; wait until send buffer is empty
l1:
         jsr ss_stat_transmit_ready
         beq l1

         ; send byte
         pla
         sta fifo_txd
         rts

;------------------------------------------------------------------------
------------------
```