

# CONTROL SIMULATION FOR TAPE-DECK

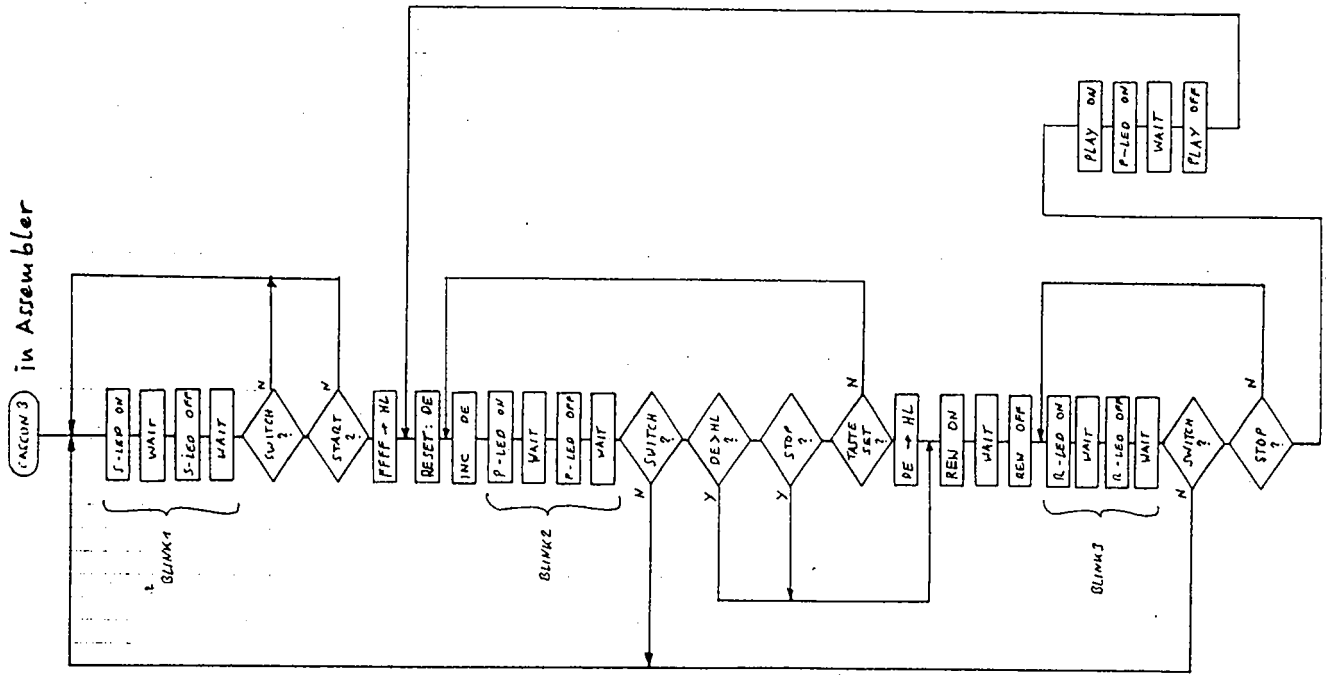
## Screen # 0

0 The following screens show a simple way to control a stereo-  
1 tapedeck or something like this. In spring '84 I fixed an  
2 auto-repeat-control into an electronic tapedeck and wrote the  
3 program in assembler because I had no Target-compiler in FORTH.  
4 But this year I got One and to check out the differences of  
5 the two approaches it was new stiled in FORTH.  
6  
7 First I had to build a new structure because the assembler-  
8 version had no. The next difference is that the asm-code is  
9 very short but if I wish to change or to add something - a bad  
10 job. On the other hand, if an application like this is developed  
11 to control a machine in industry - asm-code needs only a 2Kbyte  
12 EPROM and no RAM. So I found out that these screens are more a  
13 lesson than useful in practice but it was fun to construct the  
14 'virtual tapedeck' for to run the FORTH-version. A lesson first  
15 in factoring and second in simulating a process. ok

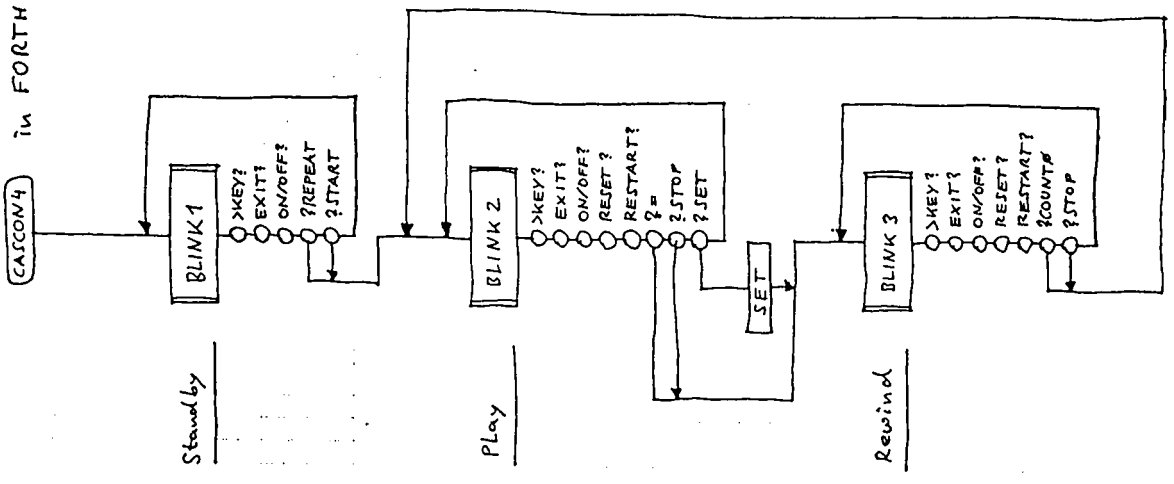
## Screen # 1

```
0 \ Load-Screen cassette-control-program          LRA 24.10.85
1
2 2 LOAD                                          \ variables etc.
3 3 LOAD                                          \ check key-code ?xxxx
4 4 LOAD      5 LOAD                            \ .places: .xxxx
5 6 LOAD      7 LOAD                            \ xxxx-on-off x-blink's
6 8 LOAD                                          \ on/off? exit?
7 9 LOAD                                          \ .menu .mode
8 10 LOAD                                         \ start
9 11 LOAD                                         \ play-rewind
10 13 LOAD                                        \ cascon-main
11
12
13 TURNKEY  CASCON5  CASCS
14
15
```

in Assembler



in FORTH



Screen # 2

```
\ cascon vars at hom >key? set reset reset?   LRA 24.10.85

VARIABLE COUNT VARIABLE VALUE VARIABLE SWITCH VARIABLE (KEY)

: AT      (-) GOTOXY ;      : CLS      (-) CLEARSCREEN ;
: ON      (-) -1 SWAP ! ;    : OFF      (-) 0 SWAP ! ;
: HOM     (-) 1 1 AT ;

: >KEY?   (-)
          ?TERMINAL IF KEY (KEY) ! THEN ;
: SET     (-)
          COUNT @ VALUE ! ;
: CHECK-RESET (-)
          30000 VALUE ! ;
: RESET?  (-)
          (KEY) @ ASCII R = IF (KEY) OFF CHECK-RESET THEN ;
```

Screen # 3

```
\ cascon hardware simulation ?xxxxxx          LRA 20.03.85
: ?START (- f)
          (KEY) @ ASCII T = IF (KEY) OFF -1 ELSE 0 THEN ;
: ?STOP  (- f)
          (KEY) @ ASCII P = IF (KEY) OFF -1 ELSE 0 THEN ;
: ?SET   (- f)
          (KEY) @ ASCII S = IF (KEY) OFF SET -1 ELSE 0 THEN ;
: ?REPEAT (- f)
          SWITCH @ 0 = NOT ;
: ?=     (- f)
          COUNT @ VALUE @ = ;
: ?COUNT0 (- f)
          COUNT @ 0 = ;
```

Screen # 4

```
\ cascon hardware simulation .xxxxxx-boxes    LRA 20.03.85

: .REPEAT-BX (-) 8 3 AT ." Repeat----"
              8 5 AT ." -----" ;
: .REWD-REL-BX (-) 8 8 AT ." Rewind-Rel"
              8 10 AT ." -----" ;
: .PLAY-REL-BX (-) 8 13 AT ." Play-Rel--"
              8 15 AT ." -----" ;

: .LED-BX     (-) 55 3 AT ." ---LED----"
              55 5 AT ." -----" ;
: .CHECK-BX   (-) 55 8 AT ." Countcheck"
              55 10 AT ." -----" ;
: .COUNT-BX (-) 55 13 AT ." Counter---"
              55 15 AT ." -----" ;
```

Screen # 5

```
\ cascon hardware simulation .xxxxxxx        LRA 20.03.85
: .BOXES     (-) .REPEAT-BX .LED-BX
                .REWD-REL-BX .CHECK-BX
                .PLAY-REL-BX .COUNT-BX HOM ;

: .REPEAT    (-) 8 4 AT ; \ point into boxes
: .REWD-REL  (-) 8 9 AT ; \
: .PLAY-REL  (-) 8 14 AT ; \
: .LED       (-) 55 4 AT ; \
: .CHECK-VAL (-) 55 9 AT ; \
: .COUNT-VAL (-) 55 14 AT ; \

: .MODE      (-) 33 4 AT ; \ pointer for text
: .CHECKER   (-) VALUE @ .CHECK-VAL 9 .R HOM ;
: .COUNTER   (-) COUNT @ .COUNT-VAL 9 .R HOM ;
```

Screen # 6

```
\ cascon hardware simulation xxxx-on/off      LRA 24.10.85
( words for to display the status of tape-deck and control )
: STARS-ON   (-) 10 0 DO 42 EMIT LOOP HOM ;
: STARS-OFF  (-) 10 SPACES HOM ;

: LED-ON     (-) .LED STARS-ON ;
: LED-OFF    (-) .LED STARS-OFF ;

: PLAY-REL-ON (-) .PLAY-REL STARS-ON ;
: PLAY-REL-OFF (-) .PLAY-REL STARS-OFF ;

: REWD-REL-ON (-) .REWD-REL STARS-ON ;
: REWD-REL-OFF (-) .REWD-REL STARS-OFF ;

: REPEAT-DISP-ON (-) .REPEAT STARS-ON ;
: REPEAT-DISP-OFF (-) .REPEAT STARS-OFF ;
```

Screen # 7

```
\ cascon hardware simulation count.x blinkx LRA 24.10.85
( the word 'msec' provides a wait-loop in milliseconds - the )
( inner value is dependend on the CPU-timing - here 6MHz 280 )
: MSEC      ( n - ) 0 DO 36 0 DO LOOP LOOP ;

: COUNT-RESET (-) COUNT OFF ;
: DEC-COUNTER (-) -1 COUNT +! ;
: INC-COUNTER (-) 1 COUNT +! ;

: STBY-BLINK (-) LED-ON 800 MSEC LED-OFF 800 MSEC ;
: PLAY-BLINK (-) LED-ON 400 MSEC LED-OFF 400 MSEC ;
: REWD-BLINK (-) LED-ON 50 MSEC LED-OFF 50 MSEC ;

: PLAY-ON    (-) PLAY-REL-ON 500 MSEC PLAY-REL-OFF ;
: REWD-ON    (-) REWD-REL-ON 500 MSEC REWD-REL-OFF ;
```

Screen # 8

```
\ cascon hardware simulation on/off? exit?   LRA 24.10.85
  ( on/off checks and toggles the simulated switch )
: ON/OFF? ( - )
  (KEY) @ ASCII 0 =
    IF (KEY) OFF SWITCH @ 0=
      IF SWITCH ON REPEAT-DISP-ON
      ELSE SWITCH OFF REPEAT-DISP-OFF
      THEN
    THEN ;
      ( with 'x' to cp/m )
: EXIT? ( - )
  (KEY) @ ASCII X = IF BYE THEN ;
: RESTART? ( - ) SWITCH @ 0= \ Repeat off
  IF ABORT THEN ; \ exit curr. def.
```

Screen # 10

```
\ cascon hard sim   stby   LRA 24.10.85
: STBY ( - )
  .BOXES .MENU .COUNTER .CHECKER .STBY-MODE
  CHECK-RESET COUNT-RESET
  BEGIN STBY-BLINK
  >KEY? \ if key then store in var
  EXIT? \ to cp/m
  ON/OFF? \ check if Repeat is still on
  ?REPEAT
  ?START AND
  UNTIL ;
```

Screen # 12

```
\ cascon hard sim   rewind   LRA 24.10.85
  REWD-ON .REWD-MODE
  BEGIN REWD-BLINK DEC-COUNTER .COUNTER
  >KEY? \ if key then store in var
  EXIT? \ to cp/m
  ON/OFF? \ check if Repeat is still on
  RESET? \ checker = high-value
  RESTART? \ abort to restart
  ?COUNT0 ?STOP OR
  UNTIL
  PLAY-ON ;
```

Screen # 9

```
\ cascon hardware simulation .menu .x-modes   LRA 24.10.85
: .MENU ( - )
  30 9 AT ." 0 = Repeat ON/OFF"
  30 11 AT ." T = Start Tape"
  30 12 AT ." P = Stop Tape"
  30 14 AT ." S = Set Checker"
  30 15 AT ." R = Reset Checker"
  60 18 AT ." X = Exit to CP/M"
  25 4 AT ." mode : "
  5 20 AT ." FORTH-Demo Tape-Deck Control-Program with hardware-simulation"
  25 22 AT ." for function 'Auto-Repeat' " ;
: .STBY-MODE ( - ) .MODE ." STAND-BY" ;
: .PLAY-MODE ( - ) .MODE ." PLAY " ;
: .REWD-MODE ( - ) .MODE ." REWIND " ;
```

Screen # 11

```
\ cascon hard sim   play   LRA 24.10.85
: PLAY-REWIND ( - )
  .COUNTER .PLAY-MODE
  BEGIN INC-COUNTER PLAY-BLINK .COUNTER
  >KEY? \ if key then store in var
  EXIT? \ to cp/m
  ON/OFF? \ check if Repeat is still on
  RESET? \ checker = high-value
  RESTART? \ abort to restart
  ?= ?STOP OR ?SET OR
  .CHECKER
  UNTIL
  -->
```

Screen # 13

```
\ cascon hard sim   CASCON5 main   LRA 24.10.85
: CASCON5 CLS STBY
  BEGIN PLAY-REWIND
  AGAIN ;
;S
( The application has to be compiled as a Turnkey-Program, so )
( that Key '0' [=repeat-off] leads to ABORT and this leads )
( the interpreter to execute the application again and again. )
( ?START has Tape started ? )
( ?STOP has Tape stopped ? )
( ?SET was SET-key pressed ? )
( ?REPEAT is Repeat-function still selected ? )
( ?= is Counter equals [or greater] than Checker ? )
( ?COUNT0 is Counter already decremented to Zero ? )
```

## CONTROL SIMULATION FOR TAPE-DECK

### Screen # 0

0 The following screens show a simple way to control a stereo-  
1 tapedeck or something like this. In spring '84 I fixed an  
2 auto-repeat-control into an electronic tapedeck and wrote the  
3 program in assembler because I had no Target-compiler in FORTH.  
4 But this year I got One and to check out the differences of  
5 the two approaches it was new stiled in FORTH.  
6  
7 First I had to build a new structure because the assembler-  
8 version had no. The next difference is that the asm-code is  
9 very short but if I wish to change or to add something - a bad  
10 job. On the other hand, if an application like this is developed  
11 to control a machine in industry - asm-code needs only a 2Kbyte  
12 EPROM and no RAM. So I found out that these screens are more a  
13 lesson than useful in practice but it was fun to construct the  
14 'virtual tapedeck' for to run the FORTH-version. A lesson first  
15 in factoring and second in simulating a process. ok

### Screen # 1

```
0 \ Load-Screen cassette-control-program          LRA 24.10.85
1
2 2 LOAD                                          \ variables etc.
3 3 LOAD                                          \ check key-code ?xxxx
4 4 LOAD      5 LOAD                             \ .places: .xxxx
5 6 LOAD      7 LOAD                             \ xxxx-on-off x-blink's
6 8 LOAD                                          \ on/off? exit?
7 9 LOAD                                          \ .menu .mode
8 10 LOAD                                         \ start
9 11 LOAD                                         \ play-rewind
10 13 LOAD                                        \ cascon-main
11
12
13 TURNKEY  CASCON5  CASCO5
14
15
```