

```

MODULE* BootLoadLine; (*NW 20.10.2013 / PR 4.2.2014 / AP 1.11.2017 for use in
Oberon0*)
  IMPORT SYSTEM, SysDef;
CONST VERSION* = "BootLoadLine AP 1.11.2017 Oberon0";
(* sw0: init SD; sw1: line|disk*)
MT      = SysDef.MT;          (* 12; *)
SP      = SysDef.SP;          (* 14; *)
MTOrg   = SysDef.MTOrg;      (* 20H; *)
MemLim  = SysDef.MEMLIM;     (* 0E7EF0H; *)
stackOrg= SysDef.STACKORG;   (* 80000H; *)
rsData  = SysDef.RSDATA;     (* -56; *)
rsCtrl  = SysDef.RSCTRL;     (* -52; *)
(* PO.Applications, chapter 14.1, page 74; RISC5.Update page 2 *)
SysStartAdr = SysDef.SysStartAdr; (* = 0; *)
IntVecAdr   = SysDef.IntVecAdr;   (* = 4; *)
MemLimAdr   = SysDef.MemLimAdr;   (* = 12; *)
ModAllocAdr = SysDef.ModAllocAdr; (* = 16; *)
ModRootAdr  = SysDef.ModRootAdr;  (* = 20; *)
ModLimAdr   = SysDef.ModLimAdr;   (* = 24; *)

PROCEDURE RecInt(VAR x: INTEGER);
  VAR z, y, i: INTEGER;
BEGIN z := 0; i := 4;
  REPEAT i := i-1;
    REPEAT UNTIL SYSTEM.BIT(rsCtrl, 0);
      SYSTEM.GET(rsData, y); z := ROR(z+y, 8)
    UNTIL i = 0;
  x := z
END RecInt;

PROCEDURE LoadFromLine*;
  VAR len, adr, dat: INTEGER;
BEGIN RecInt(len);
  WHILE len > 0 DO
    RecInt(adr);
    REPEAT RecInt(dat);
      SYSTEM.PUT(adr, dat); adr := adr + 4; len := len - 4
    UNTIL len = 0;
    RecInt(len)
  END
END LoadFromLine;

BEGIN
  SYSTEM.LDREG(SP, stackOrg);
  SYSTEM.LDREG(MT, MTOrg);
  LED(82H); LoadFromLine;
  SYSTEM.PUT(MemLimAdr, MemLim); (*12*)
  SYSTEM.PUT(ModLimAdr, stackOrg); (*24*)
  LED(84H)
END BootLoadLine.

```