

Assembler Label Directory LBLDIR.

Dependencies: LOOP (Internal)@116 MLOCB@ xD00
 Routines Used: ERRNE@ 2E0 RSTKB@ 0098 CLLCDE@ 2CF0
 MSG105@ 1C80 RSTMS0@ 038E OFF@ 11C8
 RSTI0@ 009D

Output: Labels and addresses in buffer order to LCD and any printer in TRACE mode.

Input: ON turns calculator OFF midway through listing.
 RIS terminates listing and returns control to the user.
 SST holds the current label until the key is released.
 Any other key causes the listing to speed up whilst it is held.
 Buffer 1 must contain DAVID Assembler labels.

Errors: NONEXISTENT if Buffer 1 is not there or empty.

x100		092	R:	
		009	I	
		004	D	
		00C	L	
		002	B	
		00C	L	
x106	LBLDIR:	A=0	MS	; Entry point
		A=A+1	MS	; load buffer ID for MLOCB
		GOSUB	MLOCB	; Find buffer 1
		JNC+03	CONT	; If found continue below
	ELBL:	NC GO	ERRNE	; Else give NONEXISTENT error & stop
	CONT:	RCR	12	; Get buffer size byte from header
		R=11		
		C=0	R←-	; and clear rest
		RCR	9	; Put size in C[M]
		A<>C	S&X	; Get addr in C[S&X]
		C=C-1	M	; If size = 0 then
		JC-	ELBL	; give NONEXISTENT error
		M=C		; keep address & size in M during loop
x116	LOOP:	C=M		; Retrieve address & size
		C=C+1	S&X	; Increment to next register
		C=C-1	M	; Decrement size
		C GO	RSTKB	; If carry then all done de-bounce keyboard
		M=C		; Save new addr & size in M
		C=0	ALL	; Select nonexistent peripheral
		PRPH	SLCT	
		C=M		; Select register
		RAM	SLCT	
		READ	DATA	; Read label
		N=C		; Store it in N during calculations
		DSPTOG		; Turn off display (to stop flicker)
		NC XQ	CLLCDE	; Clear & enable display
		C=N		; Get back label
		RCR	4	; Get address ready for output
		R=4		; Use pointer as count for 4 digits
x128	DLP:	C=0	S&X	; Clear unused digits
		RCR	13	; Get next hex digit for output
		A=C	S&X	; Save it in A[S&X] (A[2:1]=00)
		LDI	00A	; Compare it with 10 decimal

	?A < C	S&X	; If greater than 9 then
	JNC+	GT9	; jump to hex (A..F) part
	LDI	030	; Else load prefix of 03 in front
	C = C + A	S&X	; of digit to make it a number
	JNC+	CLP	; Jump down & output number
GT9:	C = C - 1	S&X	; Subtract 9 from (A..F) to give
	C = A - C	S&X	; a character code
CLP:	WRABCI R	(3E8)	; Output digit (character
	R = R - 1		; Decrement counter
	?R = 0		
	JNC-	DLP	; Do 4 times
	LDI	020	; Then output two spaces
	WRABCI R	(3E8)	
	WRABCI R	(3E8)	
	C = N		; Get label text ready for output
	RCL R	4	
	R = 6		; Use pointer to count out 6 digits
x140 DL2:	A = C	ALL	; Save in A temporarily
	C <> ST		
	CLRF	6	; Clear bits 6 & 7
	CLRF	7	
	C <> ST		
	C = 0	XS	; Clear bits 11-8
	?C # 0	S&X	; If this leaves all zeroes (NULL)
	JC+ 03		
	LDI	020	; change it to a space character
	WRABCI R	(3E8)	; Output character in bits 11-0
	A <> C	ALL	; Get back original
	C = C + C	ALL	; Shift right by 6 bits
	C = C + C	ALL	
	RCL R	2	
	R = R - 1		; Decrement counter
	?R = 0		
	JNC-	DL2	; Do 6 times
	DSPTOG		; Turn on display
	SETF	8	
	NC XQ	MSG105	; Output display to TRACEing printer
	CLRF	5	
	LDI	3FF	; Load time out constant
KLP:	C = C - 1	S&X	
	JCL+	CL3	; Timed out
	?KEY		
	JNC-	KLP	; No key down, continue waiting
	C = KEY		; Key down, get its value
	RCL R	3	
	A = C	S&X	; Put keycode in A[S&X]
	LDI	087	; Check if R/S pressed
	?A # C	S&X	
	JCL+	TSTON	
	NC XQ	RSTMS0	; If so reset flags
	NC GO	RSTKB	; and wait for key to come up before exiting
x168 TSTON:	LDI	018	; Check if ON pressed
	?A # C	S&X	
	NC GO	OFF	; Turn off immediately if it was

x172 CL3:

x178/9/A

```
LDI    φC2
?A#C   S&X
NC XQ   RSTK/B
NC XQ   RSTMSφ
LDI     18φ
NC XQ   RST1φ
GOTO    LOOP
```

; Check if SST pressed

; If so, wait for it to come up before continuing

; Otherwise reset flags

; Wait for a little while

; and try clearing KEY flag

; Go back for next label