

SERIES-III PL/M-86 X193 COMPILATION OF MODULE RMXON
 OBJECT MODULE PLACED IN :F1:LEDON.OBJ
 COMPILER INVOKED BY: PLM86.86 :F1:LEDON.P86 XREF

```

1      rmxon:
      DO;
      /*
      RMX88 TASK TO TURN ON CXU LED
      */

2      1      DECLARE LOCATION LITERALLY 'POINTER';

3      1      rq$wait: PROCEDURE( exchange$address, time$limit ) LOCATION EXTERNAL;
4      2      DECLARE exchange$address LOCATION;
5      2      DECLARE time$limit WORD;
6      2      END rq$wait;

7      1      rq$send: PROCEDURE( exchange$address, message$address ) EXTERNAL;
8      2      DECLARE exchange$address LOCATION;
9      2      DECLARE message$address LOCATION;
10     2      END rq$send;

11     1      rq$ctsk: PROCEDURE( std$pointer ) EXTERNAL;
12     2      DECLARE std$pointer POINTER;
13     2      END rq$ctsk;

14     1      rq$cxch: PROCEDURE( exchange$address ) EXTERNAL;
15     2      DECLARE exchange$address LOCATION;
16     2      END rq$cxch;

17     1      DECLARE task$links LITERALLY
      'link$forward      LOCATION,
      link$back          LOCATION,
      thread             LOCATION';

18     1      DECLARE task$links$a LITERALLY
      'link$forward      LOCATION,
      link$back          LOCATION,
      thread             LOCATION,
      delay              WORD';

19     1      DECLARE task$descriptor LITERALLY 'STRUCTURE (
      task$links,
      delay              WORD,
      unused             WORD,
      exchange$address   LOCATION,
      sp                 LOCATION,
      marker             ADDRESS,
      priority           BYTE,
      status             BYTE,
      name$ptr           POINTER,
      ios$links          LOCATION,
      task$link          LOCATION,
      mask               BYTE)';

```

```

20 1 DECLARE task$desc$overlay LITERALLY 'STRUCTURE (
    task$links,
    message$address LOCATION,
    exchange$address LOCATION,
    sp LOCATION,
    marker ADDRESS,
    priority BYTE,
    status BYTE,
    name$ptr POINTER,
    ios$links LOCATION,
    task$link LOCATION,
    mask BYTE)';

21 1 DECLARE task$descriptor$length LITERALLY '41';

22 1 DECLARE running LITERALLY '00010000B';
23 1 DECLARE ndp$task LITERALLY '00001000B';
24 1 DECLARE mask$int LITERALLY '00000100B';
25 1 DECLARE suspended LITERALLY '00000010B';
26 1 DECLARE delayed LITERALLY '00000001B';

27 1 DECLARE bottom$flag LITERALLY '0c7c7h';
28 1 DECLARE unused$flag LITERALLY '0c7h';

29 1 DECLARE static$task$descriptor LITERALLY
    'STRUCTURE (
        name(6) BYTE,
        pc POINTER,
        sp LOCATION,
        stklen WORD,
        ds POINTER,
        priority BYTE,
        exchange$address LOCATION,
        task$ptr LOCATION,
        task$ndp BYTE)';

30 1 DECLARE ndp$used LITERALLY '1';

31 1 DECLARE msg$hdr LITERALLY
    'link LOCATION,
    length WORD,
    type BYTE,
    home$exchange LOCATION,
    response$exchange LOCATION';

32 1 DECLARE msg$descriptor LITERALLY 'STRUCTURE(
    msg$hdr,
    remainder(1) BYTE)';

33 1 DECLARE message MSG$DESCRIPTOR;

34 1 DECLARE exchange$descriptor LITERALLY 'STRUCTURE (
    message$head LOCATION,
    message$tail LOCATION,
    task$head LOCATION,
    task$tail LOCATION,

```

```
exchange$link LOCATION)';

35 1 DECLARE sleep EXCHANGE$DESCRIPTOR PUBLIC;
36 1 DECLARE msgex EXCHANGE$DESCRIPTOR PUBLIC;

37 1 DECLARE switches BYTE PUBLIC;

38 1 ledon: PROCEDURE PUBLIC;
39 2 declare i byte;
40 2 declare msg$ptr pointer;
41 2 DECLARE mess BASED msg$ptr MSG$DESCRIPTOR;

42 2 declare time$out literally '3';
43 2 declare TRUE literally '1';
44 2 declare FALSE literally '0';
45 2 declare PPI literally '262h';
46 2 declare IN literally '0'; /* ppi input channel */
47 2 declare OUT literally '2'; /* ppi output channel */
48 2 declare RED literally '20h'; /* bit to turn red led on */
49 2 declare YELLOW literally '40h'; /* bit to turn yellow led on */
50 2 declare CLEAR literally '0h'; /* no led's on */
51 2 declare STATE byte; /* led states */

52 2 state = clear; /* initially no led's on */
53 2 OUTPUT(PPI+out) = state;

54 2 CALL rqcxch(@sleep);
55 2 CALL rqcxch(@msgex);

56 2 MESSAGE.TYPE= 250;
57 2 CALL rqsend(@msgex,@message);

58 2 DO WHILE 1;
59 3 do case SWITCHES;
60 4 state = state and not red ; /* turn red led off */
61 4 state = state or red ; /* turn red led on */
62 4 state = state and not yellow ; /* turn yellow led off */
63 4 state = state or yellow ; /* turn yellow led on */
64 4 end /* cases*/ ;

65 3 OUTPUT(PPI+out) = state; /* actual write */

66 3 msg$ptr = rq$wait(@sleep,0); /* wait forever for msg*/
67 3 CALL rqsend(@msgex,@message); /* give mmxtst hi-sign */

68 3 end;
69 2 end ledon;

70 1 end;
```

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
27			BOTTOMFLAG LITERALLY '0c7c7h'
50			CLEAR. LITERALLY '0h' IN PROC (LEDON) 52
26			DELAYED. LITERALLY '00000001B'
15	0000H	2	EXCHANGEADDRESS. POINTER IN PROC (RQCXCH) PARAMETER 15
8	0000H	2	EXCHANGEADDRESS. POINTER IN PROC (RQSEND) PARAMETER 8
4	0000H	2	EXCHANGEADDRESS. POINTER IN PROC (RQWAIT) PARAMETER 4
34			EXCHANGEDESRIPTOR LITERALLY 'STRUCTURE (message\$headLOCATION,message\$tailLOCATION,t -ask\$headLOCATION,task\$tailLOCATION,exchange\$linkLOCATION)' 35 36
44			FALSE. LITERALLY '0' IN PROC (LEDON)
39	0021H	1	I. BYTE IN PROC (LEDON)
46			IN LITERALLY '0' IN PROC (LEDON)
38	0000H	135	LEDON. PROCEDURE PUBLIC STACK=0008H
2			LOCATION LITERALLY 'POINTER' 3 4 8 9 15 33 35 36 41
24			MASKINT. LITERALLY '00000100B'
41	0000H	10	MESS STRUCTURE BASED(MSGPTR) IN PROC (LEDON)
	0000H	2	LINK POINTER
	0002H	2	LENGTH WORD
	0004H	1	TYPE BYTE
	0005H	2	HOMEECHANGE POINTER
	0007H	2	RESPONSEEXCHANGE POINTER
	0009H	1	REMAINDER. BYTE ARRAY(1)
33	0000H	10	MESSAGE. STRUCTURE 57 67
	0000H	2	LINK POINTER
	0002H	2	LENGTH WORD
	0004H	1	TYPE BYTE 56*
	0005H	2	HOMEECHANGE POINTER
	0007H	2	RESPONSEEXCHANGE POINTER
	0009H	1	REMAINDER. BYTE ARRAY(1)
3	0000H	2	MESSAGEADDRESS POINTER IN PROC (RQSEND) PARAMETER 9
32			MSGDESCRIPTOR. LITERALLY 'STRUCTURE(msg\$hdr,remainder(1)BYTE)' 33 41
36	0014H	10	MSGEX. STRUCTURE PUBLIC 55 57 67
	0000H	2	MESSAGEHEAD. POINTER
	0002H	2	MESSAGETAIL. POINTER
	0004H	2	TASKHEAD POINTER
	0006H	2	TASKTAIL POINTER
	0008H	2	EXCHANGELINK POINTER
31			MSGHDR LITERALLY 'linkLOCATION,lengthWORD,typeBYTE,home\$exchan -geLOCATION,response\$exchangeLOCATION' 33 41
40	001EH	2	MSGPTR POINTER IN PROC (LEDON) 66*
23			NDPTASK. LITERALLY '00001000B'
30			NDPUSED. LITERALLY '1'
47			OUT. LITERALLY '2' IN PROC (LEDON) 53 65
			OUTPUT BUILTIN 53* 65*
45			PPI. LITERALLY '262h' IN PROC (LEDON) 53 65
48			RED. LITERALLY '20h' IN PROC (LEDON) 60 61
	0000H		RMXON. PROCEDURE STACK=0000H
11	0000H		RQCTSK PROCEDURE EXTERNAL(2) STACK=0000H
14	0000H		RQCXCH PROCEDURE EXTERNAL(3) STACK=0000H 54 55
7	0000H		RQSEND PROCEDURE EXTERNAL(1) STACK=0000H 57 67
3	0000H		RQWAIT PROCEDURE POINTER EXTERNAL(0) STACK=0000H 66

CROSS-REFERENCE LISTING

```

22          RUNNING. . . . . LITERALLY '00C10000B'
35 000AH   10 SLEEP. . . . . STRUCTURE PUBLIC      54   66
    0000H   2  MESSAGEHEAD. . . . . POINTER
    0002H   2  MESSAGETAIL. . . . . POINTER
    0004H   2  TASKHEAD . . . . . POINTER
    0006H   2  TASKTAIL . . . . . POINTER
    0008H   2  EXCHANGELINK . . . . . POINTER
51 0022H   1  STATE. . . . . BYTE IN PROC (LEDON)      52*  53  60*  60  61*  61  62*
    62  63*  63  65
29          STATICTASKDESCRIPTOR LITERALLY 'STRUCTURE (name(6)BYTE,pcPOINTER,spLOCATIO
    -N,stklenWORD,dsPOINTER,priorityBYTE,exchange$addr
    -essLOCATION,task$ptrLOCATION,task$ndpBYTE)'
12 0000H   2  STOPPOINTER . . . . . POINTER IN PROC (RQCTSK) PARAMETER      12
25          SUSPENDED. . . . . LITERALLY '00000010B'
37 0020H   1  SWITCHES . . . . . BYTE PUBLIC      59
OCATI00,markerADDRESS,task$ptrLOCATION,task$ndpBYTE,task$exchange$addressLOCATION,spL
TION,markerADDRESS,priorityBYTE,name$PTRLOCATION,task$ndpBYTE,task$exchange$addressLOCATION,spLOCA
21          TASKDESCRIPTORLENGTH LITERALLY '41'
17          TASKLINKS. . . . . LITERALLY 'link$forwardLOCATION, link$backLOCATION, threadLO
    -CATION'
18          TASKLINKSA . . . . . LITERALLY 'link$forwardLOCATION, link$backLOCATION, threadLO
    -CATION, delayWORD'
 4 0000H   2  TIMELIMIT. . . . . WORD IN PROC (RQWAIT) PARAMETER      5
42          TIMEOUT. . . . . LITERALLY '3' IN PROC (LEDON)
43          TRUE . . . . . LITERALLY '1' IN PROC (LEDON)
28          UNUSEDFLAG . . . . . LITERALLY '0c7h'
49          YELLOW . . . . . LITERALLY '40h' IN PROC (LEDON)      62  63

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0087H      135D
CONSTANT AREA SIZE = 0000H       0D
VARIABLE AREA SIZE = 0023H      35D
MAXIMUM STACK SIZE = 0008H       8D
158 LINES READ
0 PROGRAM WARNINGS
0 PROGRAM ERRORS

```

END OF PL/M-86 COMPILATION