

SERIES-III PL/M-86 X19B COMPILATION OF MODULE TRANSMITTASK
 OBJECT MODULE PLACED IN :F1:XMITSK.OBJ
 COMPILER INVOKED BY: PLM86.86 :F1:XMITSK.P86 XREF

```

1      TRANSMIT$TASK:
      DO;
2      1      /*          ptr/selector redefinition schtick          */
      declare pointer_overlay literally 'structure(
          base selector, /* selector part */
          offset word /* offset part */
      )' ;

      /*          specific declarations          */

3      1      DECLARE DCL LITERALLY 'DECLARE';
4      1      DCL TOKEN LITERALLY 'SELECTOR';
5      1      DCL FOREVER LITERALLY 'WHILE 1';
6      1      DCL XMIT$PORT LITERALLY '0';
7      1      DCL RCV$PORT LITERALLY '1';
8      1      DCL ESOK LITERALLY '0';
9      1      DCL INFINITE LITERALLY 'OFFFHH';

10     1      DCL (MMX$RESPONSE$MAILBOX,
      CONNECTIONSID ) TOKEN;
11     1      dcl segment$p pointer ;
12     1      dcl segment$t pointer_overlay at(@segment$p) initial(0,0);

13     1      DCL (RESPONSE$PTR,
      EXCEPTION$PTR) POINTER;
14     1      DCL (EXCEPTION, FULL$DELVR) WORD;
15     1      DCL RESPONSE$WORD WORD;

16     1      DCL MESSAGE$TEXT BASED SEGMENT$p (16) BYTE;
17     1      DCL INDEX BYTE;
18     1      DCL MESS$BUFF (16) BYTE;

19     1      DCL TH$INPUT$MB (12) BYTE DATA (10, 'RQTHNORMIN', 0);
20     1      DCL TH$OUTPUT$MB (13) BYTE DATA (11, 'RQTHNORMOUT', 0);
21     1      DCL (CO, CI, ROOT$JOB) TOKEN;
22     1      dcl th$seg$ptr pointer ;
23     1      dcl th$seg$tokn pointer_overlay at(@th$seg$ptr) initial(0,0);

24     1      DCL TH$SEGMENT BASED TH$SEG$ptr STRUCTURE (
          FUNCTION WORD,
          COUNT WORD,
          XCEPT WORD,
          ACTUAL WORD,
          MESSAGE (16) BYTE);

25     1      DCL TH$WRITE LITERALLY '5';
26     1      DCL TH$READ LITERALLY '1';
27     1      DCL TH$RESPONSE$MAILBOX TOKEN;
28     1      DCL LF LITERALLY 'OAH';
29     1      DCL CR LITERALLY 'ODH';
30     1      DCL SIGNON (13) BYTE DATA (' MMX TESTER', LF, CR);
    
```

DCL Red\$off Literally '0'
 " Red\$on " '1'
 " Yellow\$off " '2'
 " Yellow\$on " '3'

```
/*  
*  
* TITLE:          MMX INTERFACE EXTERNAL DECLARATIONS:  
*                 FOR COMPILER WITH SELECTOR TYPE SUPPORT.  
*  
* DATE:           07-12-81  
*  
* REVISION:       1.0  
*  
* LANGUAGE:       PLM/86  
*  
***/
```

```
31  1  CQFIND: PROCEDURE (system$port$name,  
                    status$pointer)  
        SELECTOR  
        EXTERNAL;  
32  2  DECLARE system$port$name WORD;  
33  2  DECLARE status$pointer POINTER;  
34  2  END CQFIND;  
  
35  1  CQCLOSE: PROCEDURE (connection$id,  
                        status$pointer)  
        EXTERNAL;  
36  2  DECLARE connection$id SELECTOR;  
37  2  DECLARE status$pointer POINTER;  
38  2  END CQCLOSE;  
  
39  1  CQACTV: PROCEDURE (system$port$name,  
                        status$pointer)  
        SELECTOR  
        EXTERNAL;  
40  2  DECLARE system$port$name WORD;  
41  2  DECLARE status$pointer POINTER;  
42  2  END CQACTV;  
  
43  1  CQDACT: PROCEDURE (system$port$name,  
                        status$pointer)  
        EXTERNAL;  
44  2  DECLARE system$port$name WORD;  
45  2  DECLARE status$pointer POINTER;  
46  2  END CQDACT;  
  
47  1  CQXFER: PROCEDURE (connection$id,  
                        message$token,  
                        xfer$flag,  
                        message$length,  
                        status$pointer)  
        EXTERNAL;  
48  2  DECLARE connection$id SELECTOR;  
49  2  DECLARE message$token SELECTOR;  
50  2  DECLARE xfer$flag WORD;  
51  2  DECLARE message$length WORD;  
52  2  DECLARE status$pointer POINTER;  
53  2  END CQXFER;
```

```
/*      rmx86 primitive calls      */

54  1  RQ$CREATE$SEGMENT: PROCEDURE (size,excpt$ptr)
      SELECTOR EXTERNAL;
55  2      DECLARE size WORD;
56  2      DECLARE excpt$ptr POINTER;
57  2      END RQ$CREATE$SEGMENT;

58  1  RQ$DELETE$SEGMENT: PROCEDURE (segment,excpt$ptr)
      EXTERNAL;
59  2      DECLARE segment SELECTOR;
60  2      DECLARE excpt$ptr POINTER;
61  2      END RQ$DELETE$SEGMENT;

62  1  RQ$CREATE$MAILBOX: PROCEDURE (mailbox$flags,excpt$ptr)
      SELECTOR EXTERNAL;
63  2      DECLARE mailbox$flags WORD;
64  2      DECLARE excpt$ptr POINTER;
65  2      END RQ$CREATE$MAILBOX;

66  1  RQ$DELETE$MAILBOX: PROCEDURE (mailbox,excpt$ptr)
      EXTERNAL;
67  2      DECLARE mailbox SELECTOR;
68  2      DECLARE excpt$ptr POINTER;
69  2      END RQ$DELETE$MAILBOX;

70  1  RQ$SENDS$MESSAGE: PROCEDURE (mailbox, object, response, excpt$ptr)
      EXTERNAL;
71  2      DECLARE (mailbox,object,response) SELECTOR;
72  2      DECLARE excpt$ptr POINTER;
73  2      END RQ$SENDS$MESSAGE;

74  1  RQ$RECEIVES$MESSAGE: PROCEDURE (mailbox,time$limit,response$ptr,excpt$ptr)
      SELECTOR EXTERNAL;
75  2      DECLARE mailbox SELECTOR;
76  2      DECLARE time$limit WORD;
77  2      DECLARE (response$ptr,excpt$ptr) POINTER;
78  2      END RQ$RECEIVES$MESSAGE;

79  1  RQ$GET$SIZE: PROCEDURE (segment,excpt$ptr)
      WORD EXTERNAL;
80  2      DECLARE segment SELECTOR;
81  2      DECLARE excpt$ptr POINTER;
82  2      END RQ$GET$SIZE;

83  1  RQ$END$INIT$TASK: PROCEDURE
      EXTERNAL;
84  2      END RQ$END$INIT$TASK;

85  1  RQ$LOOKUP$OBJECT: PROCEDURE (job,
                                name,
                                time$limit,
                                excpt$ptr).
                                SELECTOR
                                EXTERNAL;
86  2      DECLARE job SELECTOR;
87  2      DECLARE name POINTER;
```

```
88 2      DECLARE time$limit WORD;
89 2      DECLARE excpt$ptr POINTER;
90 2      END RQ$LOOKUP$OBJECT;

91 1      RQ$GET$TASK$TOKENS: PROCEDURE (selection,
                                     excpt$ptr)
                                     SELECTOR
                                     EXTERNAL;
92 2      DECLARE selection BYTE;
93 2      DECLARE excpt$ptr POINTER;
94 2      END RQ$GET$TASK$TOKENS;

95 1      RQ$CREATE$REGION: PROCEDURE (flag,
                                     excpt$ptr)
                                     SELECTOR
                                     EXTERNAL;
96 2      DECLARE flag WORD;
97 2      DECLARE excpt$ptr POINTER;
98 2      END RQ$CREATE$REGION;

99 1      RQ$RECEIVE$CONTROL: PROCEDURE (region,
                                     excpt$ptr)
                                     EXTERNAL;
100 2     DECLARE region SELECTOR;
101 2     DECLARE excpt$ptr POINTER;
102 2     END RQ$RECEIVE$CONTROL;

103 1     RQ$SEND$CONTROL: PROCEDURE (excpt$ptr)
                                     EXTERNAL;
104 2     DECLARE excpt$ptr POINTER;
105 2     END RQ$SEND$CONTROL;
SEJECT
```

```
/*
  CALL ENDSINITSTASK TO RESTART ROOT JOB
*/
106 1 CALL RQSENDSINITSTASK;

/*
  GET TOKENS FOR TERMINAL HANDLER MAILBOXES
*/
107 1 ROOT$JOB = RQ$GETSTASKSTOKENS (3,@EXCEPTION);

108 1 CO = RQ$LOOKUP$OBJECT (ROOT$JOB, @TH$OUTPUT$MB, INFINITE, @EXCEPTION);
109 1 CI = RQ$LOOKUP$OBJECT (ROOT$JOB, @TH$INPUT$MB, INFINITE, @EXCEPTION);
/*
  CREATE TH MESSAGE SEGMENT AND RESPONSE MAILBOX
*/
110 1 th$seg$tokn.base = RQ$CREATE$SEGMENT (32, @EXCEPTION);
111 1 TH$RESPONSE$MAILBOX = RQ$CREATE$MAILBOX (0, @EXCEPTION);

/*
  SEND SIGNON MESSAGE
*/
112 1 TH$SEGMENT.FUNCTION = TH$WRITE;
113 1 TH$SEGMENT.COUNT = 13;
114 1 CALL MOVB (@SIGNON, @th$segment.message, 13);

115 1 CALL RQ$SEND$MESSAGE (CO, th$seg$tokn.base, TH$RESPONSE$MAILBOX, @EXCEPTION);

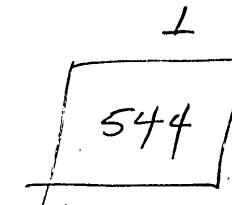
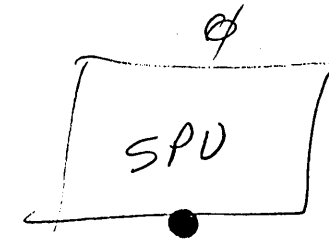
/*
  FIND THE PORT FOR THE DEVICE ON WHICH THE RCVR TASK RUNS
*/
116 1 CONNECTION$ID = CQ$FIND ( RCV$PORT, @EXCEPTION);

/*
  ACTIVE THIS DEVICES PORT
*/
117 1 MMX$RESPONSE$MAILBOX = CQ$ACTV (XMIT$PORT, @EXCEPTION);

/*
  CREATE MMX MESSAGE SEGMENT
*/
118 1 segment$t.base = RQ$CREATE$SEGMENT (32, @EXCEPTION);

/*
  LOOP READING CONSOL AND SENDING MESSAGES TO THE RECEIVER TASK
*/
119 1 DO FOREVER;
/*
  WAIT FOR CONSOL MESSAGE RESRONSE
*/
120 2 th$seg$tokn.base = RQ$RECEIVE$MESSAGE (TH$RESPONSE$MAILBOX, INFINITE,
      @RESPONSE$WORD, @EXCEPTION);

/*
  READ CONSOL FOR MESSAGE TO SEND
*/
121 2 TH$SEGMENT.FUNCTION = TH$READ;
122 2 TH$SEGMENT.COUNT = 12;
```



```
123 2 CALL RQ$SEND$MESSAGE (CI, th$seg$tokn.base, TH$RESPONSE$MAILBOX, @EXCEPTION);
124 2 th$seg$tokn.base = RQ$RECEIVE$MESSAGE (TH$RESPONSE$MAILBOX, INFINITE,
      @RESPONSE$WORD, @EXCEPTION);
125 2 CALL MOVBC(@TH$SEGMENT.MESSAGE, @MESSAGE$TEXT, TH$SEGMENT.ACTUAL);
      /*
      SEND CONSOL MESSAGE TO RCV TASK AS MMX MESSAGE
      */
126 2 FULL$DELVR = 0;
127 2 CALL CQXFER (CONNECTION$ID, segment$t.base, FULL$DELVR, OFFFFH, @EXCEPTION);
      /*
      WAIT FOR RETURNING MMX MESSAGE
      */
128 2 segment$t.base = RQ$RECEIVE$MESSAGE (MMX$RESPONSE$MAILBOX, INFINITE,
      @RESPONSE$WORD, @EXCEPTION);
      /*
      ECHO RETURNING MMX MESSAGE TO CONSOL
      */
129 2 CALL MOVBC(@MESSAGE$TEXT, @TH$SEGMENT.MESSAGE, 16);
130 2 TH$SEGMENT.FUNCTION = TH$WRITE;
131 2 TH$SEGMENT.COUNT = 16;
132 2 CALL RQ$SEND$MESSAGE (CO, th$seg$tokn.base, TH$RESPONSE$MAILBOX, @EXCEPTION);
133 2 th$seg$tokn.base = RQ$RECEIVE$MESSAGE (TH$RESPONSE$MAILBOX, INFINITE,
      @RESPONSE$WORD, @EXCEPTION);
      /*
      SEND LF,CR AND THEN READ NEXT LINE
      */
134 2 TH$SEGMENT.FUNCTION = TH$WRITE;
135 2 TH$SEGMENT.COUNT = 2;
136 2 TH$SEGMENT.MESSAGE(0) = CR;
137 2 TH$SEGMENT.MESSAGE(1) = LF;
138 2 CALL RQ$SEND$MESSAGE (CO, th$seg$tokn.base, TH$RESPONSE$MAILBOX, @EXCEPTION);
139 2 END; /* FOREVER LOOP */
140 1 END TRANSMIT$TASK;
```

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
21	0012H	2	CI SELECTOR 109* 123
21	0010H	2	CO SELECTOR 108* 115 132 138
48	0000H	2	CONNECTIONID SELECTOR IN PROC (CQXFER) PARAMETER 48
36	0000H	2	CONNECTIONID SELECTOR IN PROC (CQLOSE) PARAMETER 36
10	0002H	2	CONNECTIONID SELECTOR 116* 127
39	0000H		CQACTV PROCEDURE SELECTOR EXTERNAL(2) STACK=0000H 117
43	0000H		CQDACT PROCEDURE EXTERNAL(3) STACK=0000H
31	0000H		CQFIND PROCEDURE SELECTOR EXTERNAL(0) STACK=0000H 116
35	0000H		CQLOSE PROCEDURE EXTERNAL(1) STACK=0000H
47	0000H		CQXFER PROCEDURE EXTERNAL(4) STACK=0000H 127
29			CR LITERALLY 'ODH' 30 136
3			DCL. LITERALLY 'DECLARE' 3 4 5 6 7 8 9
			10 11 12 13 14 15 16 17 18 19 20 21
			22 23 24 25 26 27 28 29
8			EOK. LITERALLY '0'
14	000AH	2	EXCEPTION. WORD 107 108 109 110 111 115 116 117 118 120
			123 124 127 128 132 133 138
13	0008H	2	EXCEPTIONPTR POINTER
104	0000H	2	EXCPTPTR POINTER IN PROC (RQSENDCONTROL) PARAMETER 104
63	0000H	2	EXCPTPTR POINTER IN PROC (RQCREATEMAILBOX) PARAMETER 64
86	0000H	2	EXCPTPTR POINTER IN PROC (RQLOOKUPOBJECT) PARAMETER 89
75	0000H	2	EXCPTPTR POINTER IN PROC (RQRECEIVEMESSAGE) PARAMETER 77
67	0000H	2	EXCPTPTR POINTER IN PROC (RQDELETEMAILBOX) PARAMETER 68
55	0000H	2	EXCPTPTR POINTER IN PROC (RQCREATESEGMENT) PARAMETER 56
96	0000H	2	EXCPTPTR POINTER IN PROC (RQCREATERECTION) PARAMETER 97
100	0000H	2	EXCPTPTR POINTER IN PROC (RQRECEIVECONTROL) PARAMETER 101
59	0000H	2	EXCPTPTR POINTER IN PROC (RQDELETESEGMENT) PARAMETER 60
92	0000H	2	EXCPTPTR POINTER IN PROC (RQGETTASKTOKENS) PARAMETER 93
80	0000H	2	EXCPTPTR POINTER IN PROC (RQGETSIZE) PARAMETER 81
71	0000H	2	EXCPTPTR POINTER IN PROC (RQSENDMESSAGE) PARAMETER 72
96	0000H	2	FLAG WORD IN PROC (RQCREATERECTION) PARAMETER 96
5			FOREVER. LITERALLY 'WHILE 1' 119
14	000CH	2	FULLDELVR. WORD 126* 127
17	001AH	1	INDEX. BYTE
9			INFINITE LITERALLY 'OFFFFH' 108 109 120 124 128 133
86	0000H	2	JOB. SELECTOR IN PROC (RQLOOKUPOBJECT) PARAMETER 86
28			LF LITERALLY '0AH' 30 137
75	0000H	2	MAILBOX. SELECTOR IN PROC (RQRECEIVEMESSAGE) PARAMETER 75
71	0000H	2	MAILBOX. SELECTOR IN PROC (RQSENDMESSAGE) PARAMETER 71
67	0000H	2	MAILBOX. SELECTOR IN PROC (RQDELETEMAILBOX) PARAMETER 67
63	0000H	2	MAILBOXFLAGS WORD IN PROC (RQCREATEMAILBOX) PARAMETER 63
48	0000H	2	MESSAGELENGTH. WORD IN PROC (CQXFER) PARAMETER 51
16	0000H	16	MESSAGETEXT. BYTE BASED(SEGMENTP) ARRAY(16) 125 129
48	0000H	2	MESSAGETOKEN SELECTOR IN PROC (CQXFER) PARAMETER 49
18	001BH	16	MESSBUFF BYTE ARRAY(16)
10	0000H	2	MMXRESPONSEMAILBOX MOV. SELECTOR 117* 128
			BUILTIN 114 125 129
86	0000H	2	NAME POINTER IN PROC (RQLOOKUPOBJECT) PARAMETER 87
71	0000H	2	OBJECT SELECTOR IN PROC (RQSENDMESSAGE) PARAMETER 71
2			POINTER_OVERLAY. LITERALLY 'structure(base selector, /* selector part */
			-ffset word /* offset part */)' 12 23
7			RCVPORT. LITERALLY '1' 116

100	0000H	2	REGION	SELECTOR IN PROC (RQRECEIVECONTROL) PARAMETER	100			
71	0000H	2	RESPONSE	SELECTOR IN PROC (RQSENDMESSAGE) PARAMETER	71			
13	0006H	2	RESPONSEPTR	POINTER				
75	0000H	2	RESPONSEPTR	POINTER IN PROC (RQRECEIVEMESSAGE) PARAMETER		77		
15	000EH	2	RESPONSEWORD	WORD 120 124 128 133				
21	0014H	2	ROOTJOB	SELECTOR 107* 108 109				
62	0000H		RQCREATEMAILBOX	PROCEDURE SELECTOR EXTERNAL(7) STACK=0000H	111			
95	0000H		RQCREATEREGION	PROCEDURE SELECTOR EXTERNAL(15) STACK=0000H				
54	0000H		RQCREATESEGMENT	PROCEDURE SELECTOR EXTERNAL(5) STACK=0000H	110	118		
66	0000H		RQDELETEMAILBOX	PROCEDURE EXTERNAL(8) STACK=0000H				
58	0000H		RQDELETESEGMENT	PROCEDURE EXTERNAL(6) STACK=0000H				
83	0000H		RQENDINITTASK	PROCEDURE EXTERNAL(12) STACK=0000H	106			
79	0000H		RQGETSIZE	PROCEDURE WORD EXTERNAL(11) STACK=0000H				
91	0000H		RQGETTASKTOKENS	PROCEDURE SELECTOR EXTERNAL(14) STACK=0000H		107		
85	0000H		RQLOOKUPOBJECT	PROCEDURE SELECTOR EXTERNAL(13) STACK=0000H		108	109	
99	0000H		RQRECEIVECONTROL	PROCEDURE EXTERNAL(16) STACK=0000H				
74	0000H		RQRECEIVEMESSAGE	PROCEDURE SELECTOR EXTERNAL(10) STACK=0000H		120	124	128
				133				
103	0000H		RQSENDCONTROL	PROCEDURE EXTERNAL(17) STACK=0000H				
70	0000H		RQSENDMESSAGE	PROCEDURE EXTERNAL(9) STACK=0000H	115	123	132	138
59	0000H	2	SEGMENT	SELECTOR IN PROC (RQDELETESEGMENT) PARAMETER			59	
80	0000H	2	SEGMENT	SELECTOR IN PROC (RQGETSIZE) PARAMETER		80		
11	0004H	2	SEGMENTP	POINTER 12 125 129				
12	0004H	4	SEGMENTT	STRUCTURE AT INITIAL				
	0000H	2	BASE	SELECTOR 118* 127 128*				
	0002H	2	OFFSET	WORD				
92	0000H	1	SELECTION	BYTE IN PROC (RQGETTASKTOKENS) PARAMETER	92			
30	0019H	13	SIGNON	BYTE ARRAY(13) DATA 114				
55	0000H	2	SIZE	WORD IN PROC (RQCREATESEGMENT) PARAMETER	55			
48	0000H	2	STATUSPOINTER	POINTER IN PROC (CQXFER) PARAMETER	52			
44	0000H	2	STATUSPOINTER	POINTER IN PROC (CQDACT) PARAMETER	45			
40	0000H	2	STATUSPOINTER	POINTER IN PROC (CQACTV) PARAMETER	41			
36	0000H	2	STATUSPOINTER	POINTER IN PROC (CQCLOSE) PARAMETER	37			
32	0000H	2	STATUSPOINTER	POINTER IN PROC (CQFIND) PARAMETER	33			
44	0000H	2	SYSTEMPORTNAME	WORD IN PROC (CQDACT) PARAMETER	44			
40	0000H	2	SYSTEMPORTNAME	WORD IN PROC (CQACTV) PARAMETER	40			
32	0000H	2	SYSTEMPORTNAME	WORD IN PROC (CQFIND) PARAMETER	32			
19	0000H	12	THINPUTMB	BYTE ARRAY(12) DATA 109				
20	000CH	13	THOUTPUTMB	BYTE ARRAY(13) DATA 108				
26			THREAD	LITERALLY '1' 121				
27	0018H	2	THRESPONSEMAILBOX	SELECTOR 111* 115 120 123 124 132 133 138				
24	0000H	24	THSEGMENT	STRUCTURE BASED(THSEGPTR)				
	0000H	2	FUNCTION	WORD 112* 121* 130* 134*				
	0002H	2	COUNT	WORD 113* 122* 131* 135*				
	0004H	2	XCEPT	WORD				
	0006H	2	ACTUAL	WORD 125				
	0008H	16	MESSAGE	BYTE ARRAY(16) 114 125 129 136* 137*				
22	0016H	2	THSEGPTR	POINTER 23 114 125 129				
23	0016H	4	THSEGTOKN	STRUCTURE AT INITIAL				
	0000H	2	BASE	SELECTOR 110* 115 120* 123 124* 132 133* 138				
	0002H	2	OFFSET	WORD				
25			THWRITE	LITERALLY '5' 112 130 134				
86	0000H	2	TIMELIMIT	WORD IN PROC (RQLOOKUPOBJECT) PARAMETER	88			
75	0000H	2	TIMELIMIT	WORD IN PROC (RQRECEIVEMESSAGE) PARAMETER	76			
4			TOKEN	LITERALLY 'SELECTOR' 10 21 27				
	0000H	476	TRANSMITTASK	PROCEDURE STACK=000CH				


```
48 0000H 2 XFERFLAG . . . . . WORD IN PROC (CQXFER) PARAMETER 50
6        XMITPORT . . . . . LITERALLY '0'          117
```

MODULE INFORMATION:

```
CODE AREA SIZE      = 01DCH    476D
CONSTANT AREA SIZE  = 0026H    38D
VARIABLE AREA SIZE  = 0028H    43D
MAXIMUM STACK SIZE  = 000CH    12D
295 LINES READ
0 PROGRAM WARNINGS
0 PROGRAM ERRORS
```

END OF PL/M-86 COMPILATION