

SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE ROMOBLST
 OBJECT MODULE PLACED IN :F1:OBJLST.OBJ
 INVOCATION LINE CONTROLS: DEBUG

LOC	OBJ	LINE	SOURCE
		1 +1	\$TITLE ('INITIALIZATION TIME CREATED OBJECTS')
		2 +1	\$NOLIST INCLUDE(:F1:SYSGEN.MAC)
		111	NAME ROMOBLST
		112	
		113 +1	\$INCLUDE(:F1:PROPA.LIT)
	=1	114	;
	=1	115	;
	=1	116	;
	=1	117	;
	=1	118	;
	=1	119	;
		120	;
		121	;
		122	;
		123	;
		124 +1	
		125 +1	CGROUP GROUP CODE
		126 +1	DGROUP GRUP DATA,STACKS
		127 +1	
		128	
		129 +1	
----		130 +1	CODE SEGMENT BYTE PUBLIC 'CODE'
		131 +2	PUBLIC STARTINGLIST
0000		132 +2	STARTINGLIST EQU \$
----		133 +1	CODE ENDS
		134 +1	
		135	
		136 +1	
----		137 +1	CODE SEGMENT BYTE PUBLIC 'CODE'
0000 GA		138 +1	DB 10
----		139 +1	CODE ENDS
		140 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
		141 +1	PUBLIC READYLIST,READYLISTSIZE
0000 (16		142 +2	READYLIST DW 16 DUP (?,?,?)
????			
????			
????			
)			
0010		143 +2	READYLISTSIZE EQU 16
----		144 +1	DATA ENDS
		145	
		146 +1	
		147 +2	EXTRN CQMIPTASK:FAR
----		148 +1	CODE SEGMENT BYTE PUBLIC 'CODE'
0001 02		149 +1	DB 2
0002 6000	R	150 +2	DW OFFSET DGROUP:PCB_CQMIPTASK
0004 0200		151 +2	DW 2
0006 0000	E	152 +2	DW OFFSET CGROUP:CQMIPTASK
0008 6000	R	153 +2	DW OFFSET DGROUP:STK_CQMIPTASK
----		154 +1	CODE ENDS

LOC	OBJ	LINE	SOURCE
----		155 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
0060	(7	156 +2	PCB_CQMIPTASK DW 7 DUP (?)
	????)
----		157 +1	DATA ENDS
----		158 +1	STACKS SEGMENT BYTE PUBLIC 'STACKS' ;CREATE STACK
0000	(48	159 +2	DW 3CH DUP (?)
	????)
	0060	160 +2	STK_CQMIPTASK EQU \$
----		161 +1	STACKS ENDS
		162 +1	
		163 +2	EXTRN CQBOOTTASK:FAR
----		164 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'
000A	02	165 +1	DB 2
000B	6E00	R 166 +2	DW OFFSET DGROUP:PCB_CQBOOTTASK
000D	0E00	167 +2	DW 14
000F	0000	E 168 +2	DW OFFSET CGROUP:CQBOOTTASK
0011	C000	R 169 +2	DW OFFSET DGROUP:STK_CQBOOTTASK
----		170 +1	CODE ENDS
----		171 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
006E	(7	172 +2	PCB_CQBOOTTASK DW 7 DUP (?)
	????)
----		173 +1	DATA ENDS
----		174 +1	STACKS SEGMENT BYTE PUBLIC 'STACKS' ;CREATE STACK
0060	(48	175 +2	DW 3CH DUP (?)
	????)
	0000	176 +2	STK_CQBOOTTASK EQU \$
----		177 +1	STACKS ENDS
		178 +1	
		179 +2	EXTRN IDLE:FAR
----		180 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'
0013	02	181 +1	DB 2
0014	7C00	R 182 +2	DW OFFSET DGROUP:PCB_IDLE
0016	0F00	183 +2	DW 15
0018	0000	E 184 +2	DW OFFSET CGROUP:IDLE
001A	FC00	R 185 +2	DW OFFSET DGROUP:STK_IDLE
----		186 +1	CODE ENDS
----		187 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
007C	(7	188 +2	PCB_IDLE DW 7 DUP (?)
	????)
----		189 +1	DATA ENDS
----		190 +1	STACKS SEGMENT BYTE PUBLIC 'STACKS' ;CREATE STACK
00C0	(30	191 +2	DW 1EH DUP (?)
	????)
	00FC	192 +2	STK_IDLE EQU \$
----		193 +1	STACKS ENDS
		194	
		195 +1	
		196 +2	PUBLIC CQBOOTCMDMB
----		197 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'

LOC	OBJ	LINE	SOURCE
001C	06	198 +1	DB 6
001D	8A00	199 +2	DW OFFSET DGROUP:CQBOOTCMDMB
----		200 +1	CCODE ENDS
----		201 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
008A	(8	202 +2	CQBOOTCMDMB DW 8 DUP (?)
???)			
----		203 +1	DATA ENDS
----		204 +1	
----		205 +2	PUBLIC CQREMOTEWAITINGMB
001F	06	206 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'
0020	9A00	207 +1	DB 6
----		208 +2	DW OFFSET DGROUP:CQREMOTEWAITINGMB
----		209 +1	CCODE ENDS
----		210 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
009A	(8	211 +2	CQREMOTEWAITINGMB DW 8 DUP (?)
???)			
----		212 +1	DATA ENDS
----		213 +1	
----		214 +2	PUBLIC CQMIPSENDWTMBX
0022	06	215 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'
0023	AA00	216 +1	DB 6
----		217 +2	DW OFFSET DGROUP:CQMIPSENDWTMBX
----		218 +1	CCODE ENDS
----		219 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
00AA	(8	220 +2	CQMIPSENDWTMBX DW 8 DUP (?)
???)			
----		221 +1	DATA ENDS
----		222 +1	
----		223 +2	PUBLIC CQMIPREMOTEMBX
0025	06	224 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'
0026	BA00	225 +1	DB 6
----		226 +2	DW OFFSET DGROUP:CQMIPREMOTEMBX
----		227 +1	CCODE ENDS
----		228 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
00BA	(8	229 +2	CQMIPREMOTEMBX DW 8 DUP (?)
???)			
----		230 +1	DATA ENDS
----		231	
----		232 +1	
----		233 +1	CCODE SEGMENT BYTE PUBLIC 'CODE'
0028	04	234 +2	PUBLIC CQMIPUSEPERMIT
0029	CA00	235 +1	DB 4
002B	C100	236 +2	DW OFFSET DGROUP:CQMIPUSEPERMIT
----		237 +2	DW 1
----		238 +1	CCODE ENDS
----		239 +1	DATA SEGMENT BYTE PUBLIC 'DATA'
00CA	(4	240 +2	CQMIPUSEPERMIT DW 4 DUP (?)
???)			
----		241 +1	DATA ENDS
----		242	

```

LOC  OBJ                LINE  SOURCE
-----
                                243 +1
                                244 +2
                                245 +1   CODE   PUBLIC  CQWAITACB
                                246 +1   CODE   SEGMENT BYTE PUBLIC 'CODE'
002D  08                247 +1   DW      8
002E  ----                247 +2   DW      SEG DGROUP:CQWAITACB
0030  D200              R      248 +2   DW      OFFSET DGROUP:CQWAITACB
                                249 +1   CODE   ENDS
                                250 +1   DATA  SEGMENT BYTE PUBLIC 'DATA'
00D2  (8                251 +2   CQWAITACB DW      8 DUP (?)
      (???)
      )
-----
                                252 +1   DATA  ENDS
                                253 +1
                                254 +2   PUBLIC CQMIPSENDACB
                                255 +1   CODE   SEGMENT BYTE PUBLIC 'CODE'
0032  08                256 +1   DB      8
0033  ----                257 +2   DW      SEG DGROUP:CQMIPSENDACB
0035  E200              R      258 +2   DW      OFFSET DGROUP:CQMIPSENDACB
                                259 +1   CODE   ENDS
                                260 +1   DATA  SEGMENT BYTE PUBLIC 'DATA'
COE2  (8                261 +2   CQMIPSENDACB DW      8 DUP (?)
      (???)
      )
-----
                                262 +1   DATA  ENDS
                                263
                                264 +1
                                265 +1   CODE   SEGMENT BYTE PUBLIC 'CODE'
0037  00                266 +1   DB      0
                                267 +1   CODE   ENDS
                                268 +1
                                269
0038                270 +1   $TITLE('SPECIAL JUMP ROUTINES')
0038                271
0038                272   ;
0038                273   ; THIS MODULE CONTAINS VARIOUS ROUTINES TO CALL OR JUMP TO
0038                274   ; ADDRESSES/ROUTINES THAT ARE NOT ACCESSIBLE IN THE PLM-86
0038                275   ; COMPACT MODEL.
0038                276   ;
0038                277
0038                278   CODE   SEGMENT BYTE PUBLIC 'CODE'
0038                279   ASSUME  CS:CGROUP
0038                280
0038                281   ;
0038                282   ; THIS ROUTINE DOES A LONG JUMP TO THE SUPPLIED ADDRESS.
0038                283   ;
0038                284   PUBLIC  LONGGOTO
0038                285   LONGGOTO:
0038                286   LP      PROC   FAR
0038                287   POP      AX      ; GET RID OF SHORT CALL RETURN ADDRESS
0038                288   RET      ; RETURNS TO PARAMETER
0039  CB                289   LP      ENDP
                                290
                                291
                                292   ;
                                293   ; THIS ROUTINE DOES A SHORT CALL TO THE ADDRESS SUPPLIED ON THE

```

LOC	OBJ	LINE	SOURCE
		294	; STACK. IT MAINTAINS THE COMPACT MODEL AND MAY BE USED WHEN
		295	; PARAMETERS ARE TO BE PASSED.
		296	;
		297	PUBLIC SHORTCALL
003A		298	SHORTCALL:
003A 5B		299	POP BX ; RET ADR
003B 58		300	POP AX ; VALUE TO JUMP TO
003C 53		301	PUSH BX
003D 50		302	PUSH AX
003E C3		303	RET
		304	
----		305	CODE ENDS
		306	
		307	END

ASSEMBLY COMPLETE, NO ERRORS FOUND