
SPOOL
MTCLIF.LST
05/04/82
15:56:36

SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.1 ASSEMBLY OF MODULE TCL_NET_MAN
 OBJECT MODULE PLACED IN :F1:MTCLIF.OBJ
 ASSEMBLER INVOKED BY: ASM86.26 :F1:MTCLIF.SRC PRINT(:F1:MTCLIF.LST) DEBUG

```

LOC  OBJ          LINE    SOURCE
                                1          NAME    TCL_NET_MAN
                                2 +1    $TITLE ('TCL - NETWORK MANAGEMENT ROUTINES')
                                3
                                4 +1    $include (:f1:cpyrt.dca)
=1      5          ;
=1      6          ;
=1      7          ;          /* Intel Corporation Proprietary Information.
=1      8          ;          This listing is supplied under the terms of a
=1      9          ;          license agreement with Intel Corporaton and
=1     10          ;          may not be copied nor disclosed except in
=1     11          ;          accordance with the terms of that agreement. */
                                12
                                13         CGROUP  GROUP  CODE
                                14         DGROUP  GROUP  DATA
                                15
----- 16         DATA  SEGMENT PUBLIC 'DATA'
                                17         EXTRN  LCIDVECTOR:WORD
                                18         EXTRN  TCLVERSION:WORD,MAXCDBS:BYTE,NUMCDBS:BYTE,DEFRETRANTODW:DWORD
                                19         EXTRN  DEFABORTTOHI:WORD,DEFPERSIST:WORD,TOTPKTSREJ:WORD
                                20         EXTRN  BADCHKSUM:WORD,TOTPKTSRETRAN:WORD,CURMAXCDBS:byte
                                21         EXTRN  LOCNET:WORD,LOCHOST:WORD,ONBDMAXCDBS:byte, SIZECDB:word
                                22
0000  ????????? 23         CDBPTR  DD      ?          ;RETURN FOR CDB POINTER
----- 24         DATA  ENDS
                                25
----- 26         CODE  SEGMENT PUBLIC 'CODE'
                                27         ASSUME  CS:CGROUP,DS:DGROUP
                                28         EXTRN  SETUPCDB:NEAR
                                29
0000  ----      R  30         DGRP   DW      DGROUP
                                31
                                32         ;;;      The Network Management routines are driven off a table with
                                33         ;          the following structure:
                                34         ;
                                35         ;          Byte 0:
                                36         ;
                                37         ;          BIT 5:      Set if the object is in the CDB
                                38         ;          Bit 4:      Set if the object is clearable
                                39         ;          Bit 3:      Set if the object is settable.
                                40         ;          Bits 0-2: The length in bytes of the object
                                41         ;
                                42         ;          Bytes 1,2:  The base address in DGROUP of a regular object,
                                43         ;          or the offset in the CDS of a CDB object.
                                44         ;
                                45         ;          In addition, the following constant defines the limits
                                46         ;          of the data structure:
                                47         ;
                                48         ;          LAST_OBJECT: The number of the last object in the table,
                                49         ;          starting from zero.
                                50

```

LOC	OBJ	LINE	SOURCE
		51	
#		52	CTL RECORD CDB:1=0,CLR:1=0,SET:1=0,LEN:3=2
----		53	OBJ STRUC
0000		54	CTL_FLD DB ?
0001		55	LOC DW ?
----		56	OBJ ENDS
		57	
		58	
0006		59	LOCPORT EQU 6
0008		60	REMNET EQU 8
000A		61	RENHOST EQU 10
0010		62	REMPORT EQU 16
0016		63	REMCID EQU 22
0018		64	RETRANTO EQU 24
0014		65	ABORTTOHI EQU 20
0012		66	PERSIST EQU 18
0048		67	SEGMENTSSSENT EQU 72
004C		68	PKTSREJ EQU 76
004E		69	PKTSRETRAN EQU 78
		70	
		71	;
		72	; Object No 0 is a Special Object - the cidlist - its length is
		73	; twice maxcdbs in bytes. the length field is 0 but will be taken
		74	; care of by service routine.
		75	;
		76	
0002	01	E 77	TABLE OBJ <CTL<,,,1>,OFFSET DGROUP:LCIDVECTOR> ; 0
0003	0000		
0005	02	E 78	OBJ <CTL<,,,2>,OFFSET DGROUP:TCLVERSION> ; 1
0006	0000		
0008	01	E 79	OBJ <CTL<,,,1>,OFFSET DGROUP:MAXCDBS> ; 2
0009	0000		
000B	01	E 80	OBJ <CTL<,,,1>,OFFSET DGROUP:CURMAXCDBS> ; 3
000C	0000		
000E	01	E 81	OBJ <CTL<,,,1>,OFFSET DGROUP:ONBDMAXCDBS> ; 4
000F	0000		
0011	02	E 82	OBJ <CTL<,,,2>,OFFSET DGROUP:SIZECDB> ; 5
0012	0000		
0014	01	E 83	OBJ <CTL<,,,1>,OFFSET DGROUP:NUMCDBS> ; 6
0015	0000		
0017	0C	E 84	OBJ <CTL<,,1,4>,OFFSET DGROUP:DEFRETRANTODW>; 7
0018	0000		
001A	0A	E 85	OBJ <CTL<,,1,2>,OFFSET DGROUP:DEFABORTTOHI> ; 8
001B	0000		
001D	0A	E 86	OBJ <CTL<,,1,2>,OFFSET DGROUP:DEFPERSIST> ; 9
001E	0000		
0020	12	E 87	OBJ <CTL<,1,,2>,OFFSET DGROUP:TOTPKTSREJ> ; A = 10t
0021	0000		
0023	12	E 88	OBJ <CTL<,1,,2>,OFFSET DGROUP:BADCHKSUM> ; B = 11t
0024	0000		
0026	12	E 89	OBJ <CTL<,1,,2>,OFFSET DGROUP:TOTPKTSRETRAN>; C = 12t
0027	0000		
0029	02	E 90	OBJ <CTL<,,,2>,OFFSET DGROUP:LOCNET> ; D = 13t
002A	0000		
		91	

```

LOC  OBJ                LINE  SOURCE
002C  2C                92    OBJ    <CTL<1,,1,4>,RETRANTO>          ; E = 14t
002D  1800              93    OBJ    <CTL<1,,1,2>,ABORTTOHI>         ; F = 15t
002F  2A                94    OBJ    <CTL<1,,1,2>,PERSIST>          ; 10 = 16t
0030  1400              95    OBJ    <CTL<1,,,2>,SEGMENTSSENT>      ; 11 = 17t
0032  2A                96    OBJ    <CTL<1,1,,2>,PKTSREJ>         ; 12 = 18t
0033  1200              97    OBJ    <CTL<1,1,,2>,PKTSRETRAN>      ; 13 = 19t
0035  22                98    OBJ    <CTL<1,,,2>,LOCPORT>         ; 14 = 20t
0036  4800              99    OBJ    <CTL<1,,,2>,REMNET>          ; 15 = 21t
0038  32                100   OBJ    <CTL<1,,,6>,REMHOST>         ; 16 = 22t
0039  4C00              101   OBJ    <CTL<1,,,2>,REMPORT>        ; 17 = 23t
003B  32                102   OBJ    <CTL<1,,,2>,REMCID>         ; 18 = 24t
003C  4E00
003E  22
003F  0600
0041  22
0042  0800
0044  26
0045  0A00
0047  22
0048  1000
004A  22
004B  1600
0018  103
      104    LAST_OBJECT    EQU    24
      105
      106
      107
      108
      109    ;;;    TCL$READ (OBJECT, MODIFIER, VALUE$P) - READ NETWORK MANAGEMENT OBJECT.
      110    ;
      111    ;    PARAMETERS:
      112    ;    OBJECT = THE OBJECT NUMBER (WORD).
      113    ;    MODIFIER = WHICH COPY THE OF THE OBJECT (WORD, INGORED).
      114    ;    VALUE$P = POINTER TO A BUFFER TO RETURN THE VALUE IN (WORD).
      115    ;
      116    ;    RETURNS:
      117    ;    THE BUFFER POINTED TO BY VALUE$P CONTAINS THE CURRENT VALUE OF
      118    ;    THE OBJECT.
      119    ;    AX = THE LENGTH IN BYTES OF THE OBJECT.
      120    ;    AX = 0 IF ILLEGAL PARAMETER.
      121
      122
      123    PUBLIC    TCLREAD
004D  124    TCLREAD:
004D  E83700  125    CALL    SERVIC
0050  126    rtrn1:
0050  1E      127    PUSH    DS
0051  06      128    PUSH    ES
0052  1F      129    POP     DS
0053  07      130    POP     ES
0054  F3      131    REP     MOVSB
0055  A4
      132
0056  133    RTRN:    ; TCLread, TCLreadc and TCLset jump here to return
0056  8BC3  134    MOV     AX,BX

```

```

LOC  OBJ          LINE  SOURCE
0058 2E8E1E0000    R    135      MOV     DS,DGRP
005D C20800        136      RET     8
137
138
139
140
141      ;;;      TCL$READC (OBJECT, MODIFIER, VALUE$P) - READ AND CLEAR
142      ;          NETWORK MANAGEMENT OBJECT.
143      ;
144      ;      PARAMETERS:
145      ;          OBJECT = THE OBJECT NUMBER (WORD).
146      ;          MODIFIER = WHICH COPY THE OF THE OBJECT (WORD, IGNORED).
147      ;          VALUE$P = POINTER TO A BUFFER TO RETURN THE VALUE IN (WORD).
148      ;
149      ;      RETURNS:
150      ;          THE BUFFER POINTED TO BY VALUE$P CONTAINS THE CURRENT VALUE OF
151      ;          THE OBJECT.
152      ;          THE OBJECT HAS BEEN CLEARED TO 0 IF A CLEARABLE OBJECT.
153      ;          AX = THE LENGTH IN BYTES OF THE OBJECT.
154      ;          AX = 0 IF ILLEGAL PARAMETER.
155
156
157      PUBLIC  TCLREADC
0060      158      TCLREADC:
0060 E82400        159      CALL   SERVIC
0063 1E           160      PUSH  DS
0064 06           161      PUSH  ES
0065 1F           162      POP   DS
0066 07           163      POP   ES
0067 F6C210       164      TEST  DL,MASK CLR      ;TEST IF OBJECT IS CLEARABLE
006A 7409        165      JZ    RDC2             ;IF NOT CLEARABLE
006C A4           166      RDC1: MOVSB
006D C644FF00    167      MOV   BYTE PTR DS:[SI-1],0
0071 E2F9        168      LOOP RDC1
0073 EBE1        169      JMP   short rtrn
0075 F3           170      RDC2: REP   MOVSB
0076 A4
0077 EBDD        171      jmp   short rtrn
172
173
174
175      ;;;      TCL$SET (OBJECT, MODIFIER, VALUE$P) - SET NETWORK MANAGEMENT OBJECT.
176      ;
177      ;      PARAMETERS:
178      ;          OBJECT = THE OBJECT NUMBER (WORD).
179      ;          MODIFIER = WHICH COPY THE OF THE OBJECT (WORD, IGNORED).
180      ;          VALUE$P = POINTER TO BUFFER CONTAINING NEW VALUE FOR OBJECT (WORD).
181      ;
182      ;      RETURNS:
183      ;          AX = THE LENGTH IN BYTES OF THE OBJECT.
184      ;          AX = 0 IF ILLEGAL PARAMETER.
185
186
187      PUBLIC  TCLSET
0079 E80B00      188      TCLSET: CALL  SERVIC

```

```

LOC  OBJ          LINE    SOURCE
007C  F6C208      189      test    DL,mask set    ; Test if Object is Settable
007F  74CF         190      jz      rtn1         ; If not read current value and return
0081  87F7         191      XCHG   SI,DI
0083  F3            192      REP    MOVSB
0084  A4            193      jmp    short rtn
0085  EBCF         194
                    195
                    196
                    197      ;;    SERVIC - DATA INTERFACE SERVICE ROUTINE
                    198      ;
                    199      ;    RETURNS:
                    200      ;    (BX) = (CX) = OBJECT LENGTH.
                    201      ;    (ES:SI) = POINTER TO OBJECT.
                    202      ;    (DS:DI) = POINTER TO VALUE BUFFER.
                    203      ;    (DL)  = CTL BYTE FOR OBJECT.
                    204
                    205
----
0000          206      PARMS  STRUC
0002          207      BPSAVE DW      ?
0004          208      MYRET  DW      ?           ;MY RETURN ADDRESS
0006          209      HISRET DW      ?           ;CALLER'S RETURN ADDRESS
000A          210      VALUEP DD      ?           ;VALUE POINTER
000C          211      MODIF  DW      ?           ;MODIFIER
000C          212      OBJECT DW      ?           ;OBJECT NUMBER
----          213      PARMS  ENDS
                    214
                    215
0087  55           216      SERVIC: PUSH   BP           ;SET UP BP ACCORDING TO PLM CONVENTIONS
0088  83EC         217      MOV    BP,SP
                    218
008A  8B5E0C       219      MOV    BX,[BP].OBJECT
008D  83FB18       220      CMP    BX,LAST_OBJECT
0090  7F6C         221      JG     SER3           ;IF ILLEGAL OBJECT NUMBER
                    222
0092  83C3         223      MOV    AX,BX         ;MULTIPLY OBJECT NUMBER BY 3
0094  03DB         224      ADD    BX,BX         ;to give index in table.
0096  03D8         225      ADD    BX,AX
0098  2E8A970200   R  226      MOV    DL,TABLE[BX].CTL_FLD ;READ OBJECT LENGTH
                    227
009D  F6C220       228      TEST   DL,MASK CDB   ;CHECK IF CDB ITEM
00A0  7434         229      JZ     SER1         ;IF NOT
                    230
00A2  8B460A       231      MOV    AX,[BP].MODIF ;GET MODIFIER
00A5  0BC0         232      or    ax,ax         ;check if modifier = 0
00A7  7455         233      jz     ser3         ;0 is illegal - jump
00A9  BF0000       E  234      MOV    DI,OFFSET DGROUP:LCIDVECTOR
00AC  1E           235      PUSH  DS
00AD  07           236      POP   ES
00AE  33C9         237      XOR   CX,CX
00B0  8A0E0000     E  238      MOV    CL,MAXCDBS
00B4  FC           239      CLD
00B5  F2           240      REPNZ SCASW         ;SEARCH CID VECTOR
00B6  AF
00B7  7545         241      JNZ   SER3         ;IF NOT IN TABLE

```

LOC	OBJ	LINE	SOURCE
		242	
00B9	33C0	243	XOR AX,AX
00BB	A00000	244	MOV AL,MAXCDBS ;COMPUTE INDEX = MAXCDBS - COUNT - 1
00BE	F9	245	STC
00BF	1BC1	246	SBB AX,CX
00C1	53	247	PUSH BX ;SAVE OFFSET INTO OBJECT TABLE
00C2	50	248	PUSH AX ;Param 1 : index in CDB table.
00C3	B80000	249	MOV AX,OFFSET DGROUP:CDBPTR
00C6	50	250	PUSH AX ;Param 2 : address of CDB pointer
00C7	E80000	251	CALL SETUPCDB ;GET POINTER TO CDB
00CA	5B	252	POP BX ;RESTORE OFFSET INTO OBJECT TABLE
00CB	C4360000	253	LES SI,CDBPTR
00CF	2E03B70300	254	ADD SI,TABLE[BX].LOC ;ADD OBJECT BASE ADDRESS
00D4	EB15	255	JMP short SER2
		256	
00D6	1E	257	SER1: PUSH DS ;FIND BASE OF NON-CDB OBJECT
00D7	07	258	POP ES
00D8	2E8B370300	259	MOV SI,TABLE[BX].LOC ;READ OBJECT BASE ADDRESS
		260	
00DD	0BD3	261	or bx,bx ; If BX =0 the object is CIDVECTOR
00DF	750A	262	jnz ser2 ; Pick Len from Table
		263	
00E1	33C9	264	xor cx,cx
00E3	8A0E0000	265	mov cl,maxcdb ; size of CIDVECTOR in words
00E7	0E1	266	shl cl,1 ; multiply by 2 to get length in bytes
		267	
00E9	EB0B	268	jmp short ser5
		269	
00EB	33C9	270	SER2: xor cx,cx
00ED	2E8A8F0200	271	MOV CL,table[bx].ctl_fld ;BX still has offset into object table
00F2	81E10700	272	AND CX,MASK LEN ;EXTRACT LENGTH FIELD OF "CTL"
		273	
00F6	8BD9	274	ser5: MOV BX,CX ;WANT TO RETURN LENGTH IN BX ALSO
00F8	C57E06	275	LDS DI,[BP].VALUEP ;LOAD VALUE POINTER INTO REGS
00FB	FC	276	CLD ;STRINGS INCREASING IN MEMORY ADDRESS
00FC	5D	277	POP BP
00FD	C3	278	RET
		279	
00FE	2E8E1E0000	280	SER3: MOV DS,DGRP
0103	33C0	281	XOR AX,AX ;ILLEGAL ITEM; RETURN 0 TO ORIGINAL CALLER
		282	
0105	5D	283	ser4: POP BP ;RESTORE BP
0106	5B	284	POP BX ;CLEAR LOCAL RETURN ADDRESS
0107	C20800	285	RET 8 ;RETURN TO CALLER'S CALLER
		286	
----		287	CODE ENDS
		288	END

ASSEMBLY COMPLETE, NO ERRORS FOUND