

MTR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 29  
 MTR - MAIN EXECUTIVE LOOP. 10:41:21 17-FEB-82

001.001	302 371 000	1214	JNZ	MTR.3	IF NOT PAST
001.004	076 007	1215	MVI	A,A,BEL	ELSE, DING ERROR
001.006	315 302 003	1217	CALL	WCC	
001.011	303 357 000	1218	JMP	MTR.2	TRY AGAIN
001.014	315 302 003	1220	CALL	WCC	WRITE CHARACTER BACK TO CONSOLE
001.017	176	1221	MOV	A,M	
001.020	043	1222	INX	H	GET MSB
001.021	146	1223	MOV	H,M	
001.022	157	1224	MOV	L,A	(H,L) = ROUTINE ADDRESS
001.023	351	1225	PCHL		GO TO ROUTINE

#### 1227 \*\* GETBND1 - CONTINUATION OF GETBND

001.024	041 003 040	1230	GETBND1	LXI	H,IOWRK+1
001.027	026 015	1231	MVI	D,A,CR	
001.031	315 023 015	1232	CALL	IOA	
001.034	052 002 040	1234	LHLD	IOWRK	
001.037	115	1235	MOV	C,L	
001.040	104	1236	MOV	B,H	
001.041	321	1237	POP	D	
001.042	341	1238	POP	H	
001.043	311	1239	RET		

#### 1241 \*\* VIEW - VIEW MEMORY BLOCKS

001.044	041 113 002	1245	VIEW	LXI	H,MSG,VIEW
001.047	315 100 006	1246	CALL	TYPMSG	
001.052	303 351 007	1247	JMP	VIEW3A	GET START IN DE, STOP IN BC
001.055	042 067 040	1249	VIEW1	SHLD	BLKICM
001.060	303 066 002	1250	JMP	VIEW2	SAVE START ADDRESS FOR ASCII STUFF

1253 \*\* SAE - STORE ABUSS AND EXIT.  
1254 \*

1255 \* ENTRY (HL) = ABUSS VALUE  
1256 \* EXIT TO (RET)  
1257 \* USES NONE

1258  
1259 ERRHI 1063A-\*

1260 ORG 1063A  
1261

001.063 042 024 040 SHLD ABUSS  
001.066 311 RET

1265 \*\* PIN - PORT IN  
1266 \*

1267 \* PIN INPUTS A BYTE FROM DISK  
1268 \*

1269 \* ENTRY: NONE  
1270 \*

1271 \* EXIT: (A) = INPUT BYTE FROM Z47  
1272 \*

1273 \* USE: AF  
1274

001.067 1275 PIN EQU \*

001.067 315 170 006 CALL IN.  
001.072 346 240 ANI S.DTR+S.DON

001.074 050 371 JR Z,PIN  
001.076 067 STC

001.077 360 RP  
001.100 303 150 006 JMP INL.

GET STATUS  
CHECK FOR DATA TERMINAL REQUEST  
IF NOT READY, WAIT  
IF NO S.DTR, MUST BE S.DON  
INPUT A BYTE FROM PORT

MTR90-1 - H/2-89 MONITOR  
MONITOR TASK SUBROUTINES.

#09.02.01.

Unix H8ASM V1.4.1 5-Jul-80  
10:41:23 17-FEB-82

Page 31

000.000	1284	ERRMI	1103A-*
001.103	1285	ORG	1103A
	1286	PCA	- PROGRAM COUNTER ALTER
	1287	*	
	1288	*	PCA INPUTS AND/OR DISPLAYS THE CURRENT USER PROGRAM VALUE AND ALLOWS
	1289	*	A NEW VALUE TO BE ENTERED OR RETAINS THE CURRENT VALUE IF
	1290	*	A CR IS TYPED
	1291	*	
	1292	*	ENTRY NONE
	1293	*	EXIT NONE
	1294	*	USES A,D,E,H,L,F
	1295		
	1296		
	1297	PCA	COMPLETE PC MESSAGE
001.103	041 214 006	LXI	H,MSG,PC
001.106	315 100 006	CALL	TYPMSC
001.111	076 012	MVI	A,10
001.113	315 052 003	CALL	LRA.
001.116	136	MOV	E,H
001.117	043	INX	H
001.120	126	MOV	D,M
001.121	353	XCHG	(H,L) = USER PC VALUE
	1305		
001.122	315 012 015	CALL	IROC
001.125	332 137 001	JC	PCAI
	1308		
001.130	315 064 015	CALL	IOA
001.133	315 012 015	CALL	IROC
001.136	320	RNC	ELSE, OUTPUT CURRENT VALUE SEE IF USER WANTS TO CHANGE IT NOW IF NO CHANGE, EXIT
	1311		
	1312		
	1313	*	ENTER NEW USER PC VALUE
	1314		
001.137	353	XCHG	(H,L) = ADDRESS OF USER PC VALUE
001.140	026 015	MVI	D,A,CR
001.142	315 023 015	CALL	IOA
001.145	311	RET	INPUT NEW ADDRESS EXIT
	1318		
	1320	**	G088 - GO TO USER ROUTINE FROM H88 MONITOR
	1321	*	
	1322	*	G088 WAITS FOR A CARRIAGE RETURN OR A NEW ADDRESS TERMINATED WITH
	1323	*	A CARRIAGE RETURN. IF NO ADDRESS IS ENTERED, G088 TRANSFERS
	1324	*	CONTROL TO THE ADDRESS SPECIFIED BY THE USER PC VALUE
	1325		
	1326		
	1327	G088	COMPLETE GO MESSAGE
001.146	041 165 006	LXI	H,MSG,GO
001.151	315 100 006	CALL	TYPMSC
001.154	315 012 015	CALL	IROC
001.157	322 177 001	JNC	G088.1
	1330		
	1331		
001.162	365	PUSH	PSW
001.163	076 012	MVI	A,10
001.165	315 052 003	CALL	LRA.
001.170	043	INX	H
001.171	361	POP	PSW
	1336		
			POINT TO MSB GET FIRST CHARACTER BACK

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 32  
 MONITOR TASK SUBROUTINES. 10:41:24 17-FEB-82 G088

001.172 026 015 1337 MVI D,A.CR END ADDRESS WITH A RETURN  
 001.174 315 023 015 1338 CALL IOA INPUT NEW GO ADDRESS  
 001.177 315 370 005 1339 G088.1 CALL WCR. ECHO RETURN  
 001.202 303 222 001 1340 JMP GO EXECUTE USER ROUTINE

1342 \*\* AUTO80 - AUTO 800T  
 1343 \*  
 1344 \* ENTRY: NONE  
 1345 \*  
 1346 \* EXIT: (SEE 'DEVICE' ROUTINE)  
 1347 \*  
 1348 \* USE: ALL  
 1349

001.205 257 1350 AUTO80 XRA A SET TO PRIMARY FLAG  
 001.206 315 273 002 1351 CALL DEVICE CHECK DEVICE INFORMATION  
 001.211 303 342 001 1352 JMP 800T GOTO 800T IT

000.006 1354 ERRMI 1222A-\*  
 001.222 1355 ORG 1222A  
 1356 \*\* GO - RETURN TO USER MODE  
 1357 \*  
 1358 \* ENTRY NONE  
 1359  
 000.000 1360 ERRNZ \*-1222A  
 1361  
 001.222 303 063 000 1362 GO JMP GO.

1364 \*\* SSTEP - SINGLE STEP INSTRUCTION.

1365 \*  
 1366 \* ENTRY NONE  
 1367  
 000.000 1368 ERRNZ \*-1225A  
 1369  
 001.225 1370 SSTEP EQU \* SINGLE STEP  
 001.225 363 1371 DI DISABLE INTERRUPTS UNTIL THE RIGHT TIME  
 001.226 072 011 040 1372 LDA CILFLG  
 001.231 356 020 1373 XRI CB.SSI CLEAR SINGLE STEP INHIBIT  
 001.233 323 360 1374 OUT OP.CTL PRIME SINGLE STEP INTERRUPT  
 001.235 062 011 040 1375 SSI STA CILFLG SET NEW FLAG VALUES  
 001.240 341 1376 POP H CLEAN STACK  
 001.241 303 172 000 1377 JMP INTXIT RETURN TO USER ROUTINE FOR STEP

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix HBASM V1.4.1 5-Jul-80 Page 33  
 MONITOR TASK SUBROUTINES. 10:41:25 17-FEB-82

	1379	**	STPRIN - SINGLE STEP RETURN
000.000	1380		
	1381		ERRNZ *-1244A
	1382		
001.244	1383	STPRIN EQU *	
001.244	1384	ORI CB.SSI	DISABLE SINGLE STEP INTERRUPTION
001.246	1385	OUT OP.CTL	TURN OFF SINGLE STEP ENABLE
040.011	1386	SET CTLFLG	
	1387	STAX D	
001.250	1388	ANI CB.MTL	SEE IF IN MONITOR MODE
001.253	1389	JNZ MTR	
001.256	1390	JMP UIVEC+3	TRANSFER TO USER'S ROUTINE

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASH V1.4.1 5-Jul-80 Page 34  
 NORMAL BOOT 10:41:26 17-FEB-82

1393	**	M800T	-	NORMAL BOOT
1394	*			
1395	*	M800T IS ENTERED WHEN USER TYPE 'BOOT' COMMAND FROM MONITOR.		
1396	*	IT WILL ACCEPT THE BOOT DEVICE AS WELL AS THE UNIT NUMBER FROM		
1397	*	CONSOLE AND GO TO THE BOOT CODE.		
1398	*			
1399	*	ENTRY: NONE		
1400	*			
1401	*	EXIT: (AIO.UNI) = UNIT NUMBER TO BOOT		
1402	*	(PRIM) = PORT ADDRESS OF THE BOOT DEVICE		
1403	*	(IMFG) = DEVICE TYPE, -1 IS I47; -0 IS H17		
1404	*			
1405	*	USED: ALL		
1406				
001.261	257	M800T	XRA A	SET Z FLAG TO PRIMARY DEVICE
001.262	315 273 002	1408 M800T0	CALL DEVICE	READ SWITCH TO DETERMINE BOOT DEVICE
001.265	315 262 003	1409 START1	CALL RCC	INPUT FROM KB
001.270	376 015	1410	CPI A,CR	IF INPUT IS CR
001.272	050 043	1411	JR Z,B00T0.	THEN TAKE IT AS DRIVE 0
001.274	315 133 016	1412	CALL B00T7	
001.277	070 007	1413	JR C,WRONG	
001.301	270	1414	CMP B	
001.302	070 044	1415	JR C,B00T5	IF WITHIN THE RANGE, BOOT IT!
001.304	010	1416	MI,EXAF	SAVE INPUT, CHECK PRIM OR SEC?
001.305	050 010	1417	JR Z,NB7	IF PRIMARY, CHECK 'S'
001.307	010	1418	DB MI,EXAF	RESTORE (Z) FLAG
001.310		1419	WRONG EQU *	
001.310	076 007	1420	HVI A,A,BEL	NOT THE CASES, BEEP!
001.312	315 302 003	1421	CALL WCC	
001.315	030 346	1422	JR START1	AND TRY AGAIN
1423				
001.317	010	1424 NB7	DB MI,EXAF	RESTORE INPUT & PRIM, SEC FLAG
001.320	346 137	1425	ANI 01011111B	MASK TO UPPER CASE LETTER
001.322	376 123	1426	CPI 'S'	CHECK THE USER LIKE TO BOOT FROM
001.324	040 362	1427	JR NZ,WRONG	BOOT SECONDARY DEVICE
1428				
1429	*	USER WISHES TO BOOT FROM SECONDARY DEVICE		
1430				
001.326		1431 BSEC	EQU *	
001.326	041 042 014	1432	LXI H,BSMSG	PRINT BOOT SECONDARY MESSAGE
001.331	315 100 006	1433	CALL TYPMSG	
001.334	074	1434	INR A	SET (Z)=0 FOR SECONDARY DEVICE
001.335	030 323	1435	JR NB00T0	
1436				
1437	*	SAVE THE AIO.UNI, CHECK IF THERE IS THE BOOT DEVICE AND GO!		
1438				
001.337	315 370 005	1439 B00T0.	CALL WCR.	PRINT CR FOR GOOD LOOKS
1440				
001.342	257	1441 B00T0	XRA A	TAKE CR OR AUTO BOOT AS DRIVE 0
001.343	061 200 042	1442	LXI SP,42200A	SET STACK FOR NO COMMAND LINE
001.346	030 010	1443	JR B00T6	
1444				
001.350	315 302 003	1445 B00T5	CALL WCC	PRINT UNIT NUMBER
001.353	326 060	1446	SUI '0'	MAKE IT BINARY
001.355	303 263 013	1447	JMP CCL	CHECK FOR COMMAND LINE
001.360	062 061 041	1448 B00T6	STA AIO.UNI	STORE THE UNIT #

MTR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 35  
NORMAL BOOT 10:41:29 17-FEB-82

001.363	174	1449	MOV	A,H	CHECK IF NO DEVICE AT ADDR. PORT
001.364	247	1450	ANA	A	
001.365	373	1451	EI		INSURE INTERRUPTS READY
001.366	312 171 002	1452	JZ	NODEV	NO DEVICE
001.371	351	1453	PCHL		JMP TO THE EXECUTION ROUTINE

```

1456 **      Z47      -      BOOT FROM Z47 DISK DRIVE
1457 *
1458 *      Z47 WILL LOAD DATA FROM DISK TRACK 0 SECTOR 0 THRU 9 TO
1459 *      USER FIRST AVAILABLE RAM LOCATION. IF THE BOOT IS SUCCEED,
1460 *      CONTROL PASS TO THAT LOCATION.
1461 *
1462 *      ENTRY: (AIO,UNI) = UNIT NUMBER TO BOOT
1463 *
1464 *      EXIT: NONE
1465 *
1466 *      USES: ALL
1467
001.372      EQU      *
1468      EQU      *
1469      LD      (STK),SP
001.372      DB      355Q,163Q
1470
001.374      DW      STK
1471
1472
1473      Z47A      EQU      *
001.376      EI
1474
001.377      LDA      AIO,UNI
002.002      RLC
1476      RLC
002.003      RLC
1477      RLC
002.004      RLC
1478      RLC
002.005      RLC
1479      RLC
002.006      RLC
1480      RLC
002.007      INR      A
1481      MOV      C,A
002.010      MOV      C,A
1482
1483      RESET      IF      .DEBUG
000.001      XRA      A
1484
1485      STA      DBFLG
1486
1487      ENDIF
1488
002.011      MVI      A,W.RES
002.013      CALL      Z47X
1489
1490      DO Z47 EXTENSION
1491
1492 *      READ BOOT CODE FROM Z47
1493
002.016      LXI      H,USERFHA
1494      BOOT DESTINATION
1495
000.001      IF      .DEBUG
1496      MVI      A,10
1497      STA      DBFLG
1498      ENDIF
1499
1500
002.021      CALL      ROBLCK
1501      READ A SECTOR FROM DISK
1502
000.001      IF      .DEBUG
1503      PUSH      PSW
1504      MVI      A,11
1505      STA      DBFLG
1506      POP      PSW
1507      ENDIF
1508
002.024      JC      NDEVEY
1509      IF READ ERROR
1510
1511

```



MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix M8ASM V1.4.1 5-Jul-80 Page 37  
 BOOT Z-47 DISK DRIVE 10:41:30 17-FEB-82

002.027 052 124 041 1512 LHL D STK  
 002.032 371 1513 SPHL RESTORE STACK  
 1514  
 002.033 303 201 016 1515 JMP EUC SET CLOCK AND ENTER USER CODE

1517 \*\* RETRY - RE-BOOT Z47  
 1518 \*  
 1519 \* RETRY IS ENTERED WHEN 3.5 SECONDS TIME OUT & BOOT Z47  
 1520 \* STILL NOT SUCCEED. IT RESTORE STACK & JUMP TO BOOT Z47 ROUTINE  
 1521 \*  
 1522 \* ENTRY: NONE  
 1523 \*  
 1524 \* EXIT: (HL) = (SP)  
 1525 \*  
 1526 \* USE: HL, SP  
 1527  
 002.036 052 124 041 1528 RETRY LHL D STK GET OLD STACK ADDRESS  
 002.041 371 1529 SPHL SET TO STACK POINTER  
 002.042 030 332 1530 JR Z47A RE-BOOT

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 38  
SUPPORT ROUTINES 10:41:31 17-FEB-82

```

1533 **      R.SDP      - SET DEVICE PARAMETER, ALLOW TO SET DRIVE 0, 1, AND 2.
1534 *      (MORE INFORMATION CAN BE FOUND IN H17 ROM CODE 36062A)
1535
002.044      1536 R.SDP      EQU      *
002.044      1537 MVI      A,ERPTCNT
002.046      062 264 040 1538 STA      D.OECNT
002.051      072 061 041 1539 LDA      A10.UNI
002.054      365          1540 PUSH     PSW
002.055      376 002      1541 CPI      2
002.057      070 002      1542 JR      C,R.SDP1
002.061      076 003      1543 MVI      A,3
002.063      303 073 036 1544 R.SDP1  JMP     SDP3

```

VIEW2 - CONTINUE \*VIEW\* COMMAND

```

1546 **      VIEW2 - CONTINUE *VIEW* COMMAND
1547 *
1548
002.066      176          1549 MOV      A,M
002.067      315 077 015 1550 CALL     T08
002.072      076 040      1551 MVI      A,' '
002.074      315 302 003 1552 CALL     MCC
002.077      315 363 007 1553 CALL     VIEW4
002.102      312 355 003 1554 JZ      VIEW9
002.105      315 340 003 1555 CALL     VIEW3.
002.110      303 000 006 1556 JMP      VIEW3
1557
002.113      151 145 167 1558 MSG.VEN D8

```

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASH V1.4.1 5-JUL-80 Page 39  
 MAKE NOISE ROUTINES 10:41:32 17-FEB-82

```

000.016      1561      ERRMI 2136A-*
002.136      1562      ORG   2136A
              1563      **   HORN - MAKE NOISE.
              1564      *
              1565      *   ENTRY (A) = (MILLISECOND COUNT)/2
              1566      *   EXIT  NONE
              1567      *   USES  A,F
              1568
              1569      ERRNZ *-2136A
              1570
              1571      ALARM EQU *
002.136      1572      JR     ALARM8      BRANCH TO A JUMP TO NOISE TO DING BELL
002.136      030 026
              1573
000.000      1574      ERRNZ *-2140A
              1575
002.140      1576      HORN  PUSH  PSW
002.141      076 200      MVI   A,CB-SPK      TURN ON SPEAKER
              1577
              1578
002.143      343      HRNO  XTHL      SAVE (HL), (H) = COUNT
              1579      HRNO  PUSH  D      SAVE (DE)
              1580      XCHG      (D) = LOOP COUNT
              1581
              1582      LXI   H,CTLFLG
              1583      XRA   M
              1584      MOV   E,M      (E) = OLD CTLFLG VALUE
              1585      MOV   H,A      TURN ON HORN
              1586      MVI   L,#TICCNT
              1587
              1588      MOV   A,D      (A) = CYCLE COUNT
              1589      ADD   M
              1590      HRN2  CMP   M      WAIT REQUIRED TICCOUNTS
              1591      JR     NZ,HRN2
              1592
002.163      303 045 006      JMP   HRNX      JUMP TO AN EXTENSION OF HORN SO ROOM
              1593      *   CAN BE MADE FOR A JUMP TO NIOSE
              1594
              1595
              1596
002.166      303 053 006      ALARM8 JMP   NOISE      SEND A BELL TO THE CONSOLE
              1597
  
```

Unix H8ASH V1.4.1 5-JUL-80 Page 40  
10:41:33 17-FEB-82

MTR90-1 - H/Z-89 MONITOR #09.02.01.  
NO DEVICE INSTALLED

```

1600 **      NODEV      - NO DEVICE AT THE UNIT USER INDICATE
1601 *
1602 *      NODEV IS ENTERED WHEN: 1. 15 SECONDS TIME OUT
1603 *      OR 2. NO DEVICE IS INDICATED ON SWITCH
1604 *      OR 3. USER HIT <DELETE> TO ABORT 800T
1605 *      OR 4. 800T ERROR
1606 *      IT WILL EXIT TO 'ERROR' ROUTINE AND MONITOR LOOP
1607 *
1608 *      ENTRY: NONE
1609 *
1610 *      EXIT: (A) = 0
1611 *
1612 *      USE:  AF, HL
1613
002.171      1614 NODEV EQU *
002.171      041 046 014 1615 LXI H,ERRMSG PRINT ERROR MESSAGE
002.174      315 100 006 1616 CALL TYPHSC
002.177      062 010 040 1617 NODEV1 STA *MFLAG STOP TIMER
002.202      323 177 1618 OUT DP,DC OFF DISK
002.204      303 322 000 1619 JMP ERROR BACK TO MONITOR LOOP

```

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASH V1.4.1 5-Jul-80 Page 41  
 800T H-17 DISK DRIVE 10:41:33 17-FEB-82

```

1622 **      H17      - BOOT FROM H17 DISK SYSTEM
1623 *
1624 *      (THIS IS THE MODIFICATION OF THE H17 BOOT ROUTINE,
1625 *      MORE INFORMATION CAN BE FOUND ON H17 BOOT ROM 300000A)
1626 *      ENTRY: (A10.UNI) = THE UNIT TO BOOT
1627 *
1628 *      EXIT: NONE
1629 *
1630 *      USE: ALL
1631
002.207      EQU      *
002.207      LXI      R,8000A      SET THE COUNT TO MOVE IN CONSTANTS AND VECTORS
002.212      LXI      D,8000A      SET THE SOURCE ADDRESS
002.215      LXI      H,D.CON      SET THE DESTINATION ADDRESS
002.220      CALL     $MOVE      MOVE IT
1637
1638 **      SET ADDRESS FOR 'SET DEVICE PARAMETER' ROUTINE
1639 *      TO HANDLE DISK DRIVE 0, 1, AND 2.
1640
002.223      LXI      H,R.SDP      SET THIS ROM ROUTINE ADDRESS
002.226      SHLD     D.SDP      SET INTO RAM JUMP VECTOR
002.231      EI
1644
1645 *      WAIT TILL USER INSERT THE DISK AND CLOSE THE DOOR
1646 *      (TIMER INTERRUPT IS AFFECTED NOW)
1647
002.232      MVI      8,10      LOOK FOR SOME HOLE AND NO HOLE
002.234      CALL     R.SDP      SELECT UNIT & MOTOR ON
002.237      CALL     WNH      WAIT FOR NO HOLE
002.242      CALL     WHD      WAIT FOR HOLE
002.245      DJNZ     H17A
1653
1654 *      READ BOOT CODE
1655
002.247      CALL     H17X      H17 Extension Routine
002.252      LXI      D,USERFHA      SET THE LOAD LOCATION
002.255      LXI      8,9*256      LOAD 9 SECTORS
002.260      LXI      H,0      LOAD FROM TRACK 0 SECTOR 1
002.263      CALL     R.READ      READ DISK BOOT CODE
002.266      JR       C,NODEV      ERROR ON BOOT, BACK TO "H:"
002.270      JMP      EUC.      VECTORS ALREADY IN

```

DETERMINE BOOT WHICH DEVICE AT WHICH PORT

DETERMINE BOOT WHICH DEVICE AT WHICH PORT

DETERMINE BOOT WHICH DEVICE AT WHICH PORT

DETERMINE BOOT WHICH DEVICE AT WHICH PORT

DETERMINE BOOT WHICH DEVICE AT WHICH PORT

ENTRY: Z FLAG ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)

ENTRY: Z FLAG ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)

ENTRY: Z FLAG ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)

ENTRY: Z FLAG ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)

EXIT: HL = DEVICE BOOT EXECUTION ADDRESS

EXIT: HL = DEVICE BOOT EXECUTION ADDRESS

EXIT: HL = DEVICE BOOT EXECUTION ADDRESS

EXIT: HL = DEVICE BOOT EXECUTION ADDRESS

EXIT: HL = DEVICE BOOT EXECUTION ADDRESS

REG B = PRIMARY MAXI DRIVE NUMBER

REG B = PRIMARY MAXI DRIVE NUMBER

REG B = PRIMARY MAXI DRIVE NUMBER

REG B = PRIMARY MAXI DRIVE NUMBER

REG B = PRIMARY MAXI DRIVE NUMBER

IF Z47 = 0; H17 = 0; H37 = 0; H67 = 0; 4

IF Z47 = 0; H17 = 0; H37 = 0; H67 = 0; 4

IF Z47 = 0; H17 = 0; H37 = 0; H67 = 0; 4

IF Z47 = 0; H17 = 0; H37 = 0; H67 = 0; 4

IF Z47 = 0; H17 = 0; H37 = 0; H67 = 0; 4

(PRIM) = PRIMARY PORT ADDRESS

(PRIM) = PRIMARY PORT ADDRESS

(PRIM) = PRIMARY PORT ADDRESS

(PRIM) = PRIMARY PORT ADDRESS

(PRIM) = PRIMARY PORT ADDRESS

(TNEG) = SET UP FROM TABLE

(TNEG) = SET UP FROM TABLE

(TNEG) = SET UP FROM TABLE

(TNEG) = SET UP FROM TABLE

(TNEG) = SET UP FROM TABLE

ALL

ALL

ALL

ALL

MI-EXAF

MI-EXAF

MI-EXAF

MI-EXAF

SAVE Z FLAG

SAVE Z FLAG

SAVE Z FLAG

SAVE Z FLAG

SAVE Z FLAG

INITIAL VARIABLES

INITIAL VARIABLES

INITIAL VARIABLES

INITIAL VARIABLES

NO INTERRUPT

NO INTERRUPT

NO INTERRUPT

NO INTERRUPT

NO INTERRUPT

CLEAR H17 WORK RAM AREA

CLEAR H17 WORK RAM AREA

CLEAR H17 WORK RAM AREA

CLEAR H17 WORK RAM AREA

CLEAR H17 WORK RAM AREA

LENGTH TO CLEAR

LENGTH TO CLEAR

LENGTH TO CLEAR

LENGTH TO CLEAR

LENGTH TO CLEAR

OFF DISK

OFF DISK

OFF DISK

OFF DISK

OFF DISK

0 TIMER COUNTER

0 TIMER COUNTER

0 TIMER COUNTER

0 TIMER COUNTER

0 TIMER COUNTER

0.5 SECOND TIMER = 0

0.5 SECOND TIMER = 0

0.5 SECOND TIMER = 0

0.5 SECOND TIMER = 0

0.5 SECOND TIMER = 0

(A)=1

(A)=1

(A)=1

(A)=1

(A)=1

TIMER INTERRUPT MUST = 1

TIMER INTERRUPT MUST = 1

TIMER INTERRUPT MUST = 1

TIMER INTERRUPT MUST = 1

TIMER INTERRUPT MUST = 1

ALLOW TIMER INTERRUPT

ALLOW TIMER INTERRUPT

ALLOW TIMER INTERRUPT

ALLOW TIMER INTERRUPT

ALLOW TIMER INTERRUPT

SET ALL VECTOR TO EI/RET PROCESS

SET ALL VECTOR TO EI/RET PROCESS

SET ALL VECTOR TO EI/RET PROCESS

SET ALL VECTOR TO EI/RET PROCESS

SET ALL VECTOR TO EI/RET PROCESS

STORE LS BYTE

STORE LS BYTE

STORE LS BYTE

STORE LS BYTE

STORE LS BYTE

STORE HS BYTE

STORE HS BYTE

STORE HS BYTE

STORE HS BYTE

STORE HS BYTE

SET TIMER INTERRUPT VECTOR

SET TIMER INTERRUPT VECTOR

SET TIMER INTERRUPT VECTOR

SET TIMER INTERRUPT VECTOR

SET TIMER INTERRUPT VECTOR

DETERMINE BOOT DEVICE AND ITS INFORMATION

DETERMINE BOOT DEVICE AND ITS INFORMATION

DETERMINE BOOT DEVICE AND ITS INFORMATION

DETERMINE BOOT DEVICE AND ITS INFORMATION

DETERMINE BOOT DEVICE AND ITS INFORMATION

READ SWITCH DATA

READ SWITCH DATA

READ SWITCH DATA

READ SWITCH DATA

READ SWITCH DATA

DETERMINE WHICH TABLE IS PRIMARY

DETERMINE WHICH TABLE IS PRIMARY

DETERMINE WHICH TABLE IS PRIMARY

DETERMINE WHICH TABLE IS PRIMARY

DETERMINE WHICH TABLE IS PRIMARY

IF PORT 174 IS PRIMARY

IF PORT 174 IS PRIMARY

IF PORT 174 IS PRIMARY

IF PORT 174 IS PRIMARY

IF PORT 174 IS PRIMARY

PRIMARY DEVICE IS AT 170q

PRIMARY DEVICE IS AT 170q

PRIMARY DEVICE IS AT 170q

PRIMARY DEVICE IS AT 170q

PRIMARY DEVICE IS AT 170q

GET DEVICE SWITCHES

GET DEVICE SWITCHES

GET DEVICE SWITCHES

GET DEVICE SWITCHES

GET DEVICE SWITCHES

GET 'SD' FLAG

GET 'SD' FLAG

GET 'SD' FLAG

GET 'SD' FLAG

GET 'SD' FLAG

ASSUME PORT 174

ASSUME PORT 174

ASSUME PORT 174

ASSUME PORT 174

ASSUME PORT 174

IF WAS 174

IF WAS 174

IF WAS 174

IF WAS 174

IF WAS 174

OO PORT 170 STUFF

OO PORT 170 STUFF

OO PORT 170 STUFF

OO PORT 170 STUFF

OO PORT 170 STUFF

[illegible]

Unix M8ASH V1.4.1 5-Jul-80 Page 44  
10:41:37 17-FEB-82

MTR90-1 - H/Z-89 MONITOR #09.02.01.  
SUPPORT ROUTINES

000.007 1766 ERRHI 3047A-+  
003.047 1767 ORG 3047A  
1768 \*\* LRA - LOCATE REGISTER ADDRESS.  
1769 \*  
1770 \* ENTRY NONE.  
1771 \* EXIT  
1772 \* (A) = REGISTER INDEX  
1773 \* (H,L) = STORAGE ADDRESS  
1774 \* (D,E) = (0,A)  
1775 \* USES A,D,E,H,L,F

000.000 1776  
1777 ERRNZ \*-3047A  
1778

003.047 072 005 040 1779 LRA LDA REGI  
003.052 137 1780 LRA. MOV E,A  
003.053 026 000 1781 MVI D,0  
003.055 052 035 040 1782 LHL REGPTR  
003.060 031 1783 DAD D (DE) = (REGPTR)+(REGI)  
003.061 311 1784 RET

1786 \*\* IOA - INPUT OCTAL ADDRESS.  
1787 \*  
1788 \* ENTRY (H,L) = ADDRESS OF RECEPTION DOUBLE BYTE.  
1789 \* (D) = TERMINATING CHARACTER  
1790 \* EXIT NONE  
1791 \* USES A,D,E,H,L,F  
1792

000.000 1794 ERRNZ \*-3062A  
1795

003.062 303 166 005 1796 IOAO JMP IOA1  
003.065 000 1797 NOP RETAIN H8 ORG

1799 \*\* IOB - INPUT OCTAL BYTE.  
1800 \*  
1801 \* READ ONE OCTAL BYTE FROM THE KEYSER.  
1802 \*  
1803 \* ENTRY (H,L) = ADDRESS OF BYTE TO HOLD VALUE  
1804 \* (C) SET IF FIRST DIGIT IN (A)  
1805 \* EXIT NONE  
1806 \* USES A,D,E,H,L,F  
1807

000.000 1808  
1809 ERRNZ \*-3066A  
1810

003.066 066 000 1811 IOB0 MVI M,0 ZERO OUT OLD VALUE  
003.070 324 262 003 1812 IOB1 CMC RCC READ CONSOLE CHARACTER  
1813  
1814 \* SEE IF CHARACTER IS A VALID OCTAL VALUE  
1815 \*



MTR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 45  
 SUPPORT ROUTINES 10:41:38 17-FEB-82

```

003.073 376 060 1816 CPI '0' LESS THAN ZERO?
003.075 332 135 003 1817 JC I082 IF (A) < 0, SEE IF A TERMINATING CHARACTER
003.100 376 070 1818 CPI '8' GREATER THAN 7?
003.102 322 070 003 1819 JNC I081 IF TOO LARGE, TRY AGAIN
1820
1821 * HAVE AN OCTAL DIGIT
1822 *
003.105 315 302 003 1823 CALL WCC ECHO CHARACTER
003.110 346 007 1824 ANI 000001118 MASK FOR BINARY VALUE
003.112 137 1825 MOV E,A (E) = VALUE
003.113 176 1826 MOV A,M GET OLD VALUE
003.114 007 1827 RLC SHIFT 3
003.115 007 1828 RLC
003.116 007 1829 RLC
003.117 303 126 003 1830 JMP I081.5 JUMP AROUND AN H88/H89 TO H8 FAKE ROUTINE
1831
1832 ** FAKE OUT ROUTINE FOR CALLERS OF *00D* FROM THE H8 FRONT PANEL
1833
1834
000.000 1835 ERRNZ *-3122A
1836
003.122 043 1837 D00 INX H
003.123 043 1838 INX H
003.124 043 1839 INX H
003.125 311 1840 RET
1841
1842
1843 * CONTINUE
1844
003.126 346 370 1845 I081.5 ANI 111110008 TOSS OLD LSB DIGIT
003.130 263 1846 ORA E REPLACE WITH NEW VALUE
003.131 167 1847 MOV M,A M,A
003.132 303 070 003 1848 JMP I081 INPUT ANOTHER CHARACTER
1849
1850 * CHECK FOR A CARRIAGE RETURN TO TERMINATE BYTE
1851 *
003.135 376 015 1852 I082 CPI A,CR CARRIAGE RETURN?
003.137 310 1853 RZ RETURN IF CARRIAGE RETURN /JMT 790507/
003.140 257 1854 XRA A CLEAR CARRY /JMT 790507/
003.141 030 325 1855 JR I081 GET A NEW CHARACTER /JMT 790507/

```

## DYASC - DYNAMIC RAM ASCII OUTPUT TO CONSOLE

```

1858 **      DYASC      EQU      *
1859 *
1860 *      ENTRY      (A) = CHARACTER TO OUTPUT
1861 *      (IX) = RETURN ADDRESS
1862 *      EXIT      TO (IX)
1863 *      USES      A,C,F
1864
003.143      1865 DYASC      EQU      *
003.143      1866 *      EX      AF,AF
003.143      1867 *      DB      MI,EXAF
003.144      1868 DYASC1 IN      SC,ACE+UR,LSR  TERMINAL READY?
003.146      1869 *      ANI      UC,THE
003.150      1870 *      JZ      DYASC1      NOT YET.
003.150      1871 *
003.153      1872 *      EX      AF,AF
003.153      1873 *      DB      MI,EXAF
003.154      1874 *      OUT      SC,ACE+UR,THR
003.154      1875 *      JP      (IX)      RETURN TO CALLER
003.156      1876 *      DB      MI,JIYA,MI,JIY8

```

## DYBYT - DYNAMIC RAM BYTE OUTPUT

```

1878 **      DYBYT      JMP      DYBYTX
1879 *
1880 *      ENTRY      (A) = BYTE TO OUTPUT AS OCTAL
1881 *      (IX) = RETURN ADDRESS
1882 *      EXIT      TO (IX)
1883 *      USES      A,C,IY,F
1884
003.160      1885 DYBYT      JMP      DYBYTX
003.163      1886 DYBYT.1 ORI      '0'      MAKE ASCII
003.163      1887
003.165      1888 *      LD      IY,DYBYT.2
003.165      1889 *      DB      MI,LDYA,MI,LDYB
003.167      1890 *      DW      DYBYT.2
003.171      1891 *
003.171      1892 *      JMP      DYASC
003.171      1893
003.174      1894 DYBYT.2 MOV      A,C
003.175      1895 *      ANI      00111000B
003.177      1896 *      RRC
003.200      1897 *      RRC
003.201      1898 *      RRC
003.201      1899 *      ORI      '0'
003.202      1900
003.204      1901 *      LD      IY,DYBYT.4
003.204      1902 *      DB      MI,LDYA,MI,LDYB
003.206      1903 *      DW      DYBYT.4
003.210      1904
003.210      1905 *      JMP      DYASC
003.213      1906
003.213      1907 DYBYT.4 MOV      A,C
003.214      1908 *      ANI      00000111B
003.216      1909 *      ORI      '0'
003.216      1910
003.216      1911 *      LD      IY,DYBYT.6

```

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix HBASM V1.4.1 5-Jul-80 Page 47  
RAM TEST ROUTINES 10:41:40 17-FEB-82

003.220	375 041	1912	DB	MI.LDYA,MI.LDYB
003.222	227 003	1913	DB	DYBYT.6
		1914		
003.224	303 143 003	1915	JMP	DYASC
		1916		
003.227		1917	DYBYT.6 EQU *	
		1918	JP (IX)	
003.227	335 351	1919	DB	MI.JIXA,MI.JIXB

MSG.PAS - PASS MESSAGE FOR DYNAMIC RAM TEST

003.231	015 012 012	1921	**	MSG.PAS DB	A.CR,A.LF,A.LF
003.234	011 040 120	1922	*	DB	110,* Pass =*,110,*
003.250	000	1923		DB	0

MTR90-1 - H/Z-89 MONITOR  
RCK - READ CONSOLE KEYPAD

#09.02.01.

Unix H8ASH V1.4.1 5-Jul-80  
10:41:41 17-FEB-82

Page 48

000.007  
003.260

1929 ERRMI 3260A-\*  
1930 ORG 3260A  
1931 \*\* RCK - READ CONSOLE KEYPAD  
1932 \*  
1933 \*

1934 \* RCK IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE FRONT PANEL KEYPAD.  
1935 \* SINCE THE H88/89 DOES NOT HAVE A FRONT PANEL, THIS ROUTINE IS PROVIDED  
1936 \* ONLY TO MAINTAIN COMPATIBILITY WITH PAN-8.  
1937 \* RCK WILL IMMEDIATELY RETURN WITH A VALUE OF 0 (ZERO) IN THE ACCUMULATOR.

1938 \* ENTRY NONE  
1939 \* EXIT (A) = 0  
1940 \* USES A,F  
1941 \*

000.000 1942 \* RCK MUST HAVE SAME ENTRY AS RCK IN PAN-8  
1943 ERRNZ \*-3260A

003.260 1944  
1945 RCK EQU \*  
1946  
003.260 257 1947 XRA A  
003.261 311 1948 RET  
1949

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASH V1.4.1 5-Jul-80 Page 49  
 CONSOLE CHARACTER ROUTINES. 10:41:41 17-FEB-82

```

1953 **      RCC - READ CONSOLE CHARACTER.
1954 *
1955 *      RCC IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE.
1956 *      IF A RUBOUT/DELETE IS RECEIVED, EXIT IS TO *ERROR*.
1957 *
1958 *      ENTRY      NONE
1959 *      EXIT      TO ERROR - IF A DELETE OR RUBOUT IS ENCOUNTERED
1960 *      TO CALLER - WHEN A KEY IS HIT
1961 *      (A) = ASCII KEY VALUE
1962 *      USES      A,F
1963
1964
1965
1966 RCC      EQU      *
1967
003.262
003.262 333 355  IN      SC.ACE+UR.LSR  INPUT ACE LINE STATUS REGISTER
003.264 346 001  ANI      UC.DR        SEE IF THERE IS A DATA READY
003.266 050 372  JR       Z,RCC1
1970
1971
003.270 333 350  IN      SC.ACE+UR.RBR  ELSE, INPUT CHARACTER
003.272 346 177  ANI      0111111B     TOSS ANY PARITY
003.274 376 177  CPI      A,DEL      A,DEL
003.276 312 322 000 JZ       ERROR    IF RUBOUT, EXIT TO ERROR
1976
003.301 311      RET              ELSE, EXIT TO CALLER
1977

```

```

1979 **      WCC - WRITE CONSOLE CHARACTER
1980 *
1981 *      WRITE A CHARACTER TO THE CONSOLE UART PORT
1982 *
1983 *      ENTRY      (A) = ASCII CHARACTER TO OUTPUT
1984 *      EXIT      NONE
1985 *      USES      NONE
1986
1987
003.302 365      WCC      WCC      PUSH      PSW      SAVE CHARACTER
003.303 333 355  IN      SC.ACE+UR.LSR  INPUT ACE STATUS
003.305 346 040  ANI      UC.THE      SEE IF TRANSMITTER HOLDING REGISTER IS EMPTY
003.307 050 372  JR       Z,WCC1
1991
1992
003.311 361      POP      PSW      GET CHARACTER
003.312 323 350  OUT      SC.ACE+UR.THR  OUTPUT TO CONSOLE
003.314 311      RET

```

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASH V1.4.1 5-JUL-80 Page 50  
 CONSOLE CHARACTER ROUTINES. 10:41:42 17-FEB-82

1997 \*\* THE FOLLOWING IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!!

1998 \*  
 003.315 353 1999 DY9.5 XCHG SAVE ERROR ADDRESS  
 003.316 041 341 007 2000 LXI H,MSG.EQ OUTPUT " = "

2001  
 2002 \*  
 003.321 335 041 2003 LO IX,DY9.8 RETURN ADDRESS  
 003.323 330 003 2004 DB MI.LDXA,MI.LDXB  
 2005 DW DY9.8

003.325 303 265 007 2006 JHP DYM5G OUTPUT STRING

003.330 032 2008 DY9.8 LOAX 0 OUTPUT RAM CONTENTS

2009  
 2010 \*  
 003.331 335 041 2011 LO IX,DYHEM10 RETURN ADDRESS  
 003.333 252 013 2012 DB MI.LDXA,MI.LDXB  
 2013 DW DYHEM10

003.335 303 160 003 2014 JHP DY8YT

2016 \*\* VIEW3. - CONTINUATION OF \*VIEW\*

2017 \*  
 2018 \* SEE IF END OF BYTES  
 2019 \*  
 2020

003.340 043 2021 VIEW3. INX H BUMP POINTER  
 003.341 315 112 015 2022 CALL CHKRAD GET RADIX  
 003.344 076 360 2023 MVI A,11110000B ASSUME HEX  
 003.346 040 002 2024 JR NZ,VIEW3.A IF WAS HEX  
 003.350 076 370 2025 MVI A,11111000B  
 003.352 245 2026 VIEW3.A ANA L (A) = MASKED ADDR LSB  
 003.353 275 2027 CMP L SAME?  
 003.354 311 2028 RET LET CALLER DECIDE

2030 \*\* VIEW9 - DO THE ASCII

2031 \*  
 2032  
 003.355 052 067 040 2033 VIEW9 LHL D BLKICH RESTORE REGISTERS  
 003.360 303 171 010 2034 JMP VIEW5

MTR90-1 - H/Z-89 MONITOR  
CONSOLE CHARACTER ROUTINES.

#09.02.01.

Unix H8ASH V1.4.1 5-Jul-80  
10:41:43 17-FEB-82

Page 51

PRSRM

2037 \*\* IO ROUTINES TO BE COPIED INTO AND USED IN RAM.

2038 \*  
2039 \* MUST CONTINUE TO 3777A FOR PROPER COPY.

2040 \* THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL RAM

2041

000.006  
003.371

ERRMI 4000A-7-\*

ORG 4000A-7

2043

2044

003.371

003.371 001

003.372 000

003.373 000

003.374 000

003.375 000

003.376 012

003.377 311

000.000

2053

2054

2055

PRSRM

REFIND

CTFLG

\*MFLAG

DSPMOD

DSPROT

REGI

MI.RET

ERRNZ \*-4000A

2058	***	INITOX	EXTENSION OF INITO TO SUPPORT H88
2059			
004.000	076 002	INITOX MVI	A,H88B.CK
004.002	323 362	OUT	H88.CIL
2061			
2062			
2063	*	SET UP ACE FOR CONSOLE COMMUNICATIONS	
2064	*		
004.004	076 200	MVI	A,UC.DLA
004.006	323 353	OUT	SC.ACE+UR.LCR
004.010	041 101 004	LXI	H,BRTAB
004.013	333 362	IN	H88.SW
004.015	346 100	ANI	H88S.BK
004.017	017	RRC	
004.020	017	RRC	
004.021	017	RRC	
004.022	017	RRC	
004.023	017	RRC	
004.024	205	ADD	L
004.025	157	MOV	L,A
004.026	176	MOV	A,M
004.027	323 351	OUT	SC.ACE+UR.DLM
004.031	043	INX	H
004.032	176	MOV	A,M
004.033	323 350	OUT	SC.ACE+UR.DLL
004.035	076 003	MVI	A,UC.88M
004.037	323 353	OUT	SC.ACE+UR.LCR
004.041	076 000	MVI	A,0
004.043	323 351	OUT	SC.ACE+UR.IER
2086			
2087	*	WAIT A WHILE TO ALLOW THE CONSOLE RESET TO FINISH SO IT CAN	
2088	*	ACCEPT THE FIRST PROMPT	
2089	*		
004.045	303 113 016	JMP	INTOX0
004.050	015	INITOX1 DCR	C
004.051	040 375	JR	NZ,INITOX1
2093			
004.053	020 373	DJNZ	INITOX1
2095			
2096	*	INPUT SWITCH TO SEE IF TO BEGIN OPERATION OR MEMORY TEST	
2097	*		
004.055	333 362	IN	H88.SW
004.057	346 040	ANI	H88S.M
004.061	312 032 016	JZ	MEMORY.
2101			
2102	*	REPLACE WHAT WAS ORIGINALLY AT THE JUMP WHICH GOT US HERE	
2103	*		
004.064	021 371 003	LXI	D,PRSR0M
004.067	257	XRA	A
004.070	062 123 041	STA	AUT08
004.073	062 066 040	STA	DATA
004.076	303 003 000	JMP	INITO.0

(H,L) = BEGINNING OF BAUD RATE TABLE  
INPUT SWITCHES FOR DESIRED BAUD RATE  
MASK FOR BAUD RATE SWITCHES ONLY  
SHIFT FOR A \*2 FOR TABLE

ADD DISPLACEMENT FROM BEGINNING OF TABLE  
GET MSB OF DIVISOR  
GET LSB  
SET 8 BITS, 1 STOP BIT, NO PARITY  
SET NO INTERRUPTS  
OO OTHER STUFF FIRST

GET SWITCHES  
MASK FOR MEMORY TEST ONLY  
IF TO PERFORM MEMORY TESTS

(DE) = ROM COPY OF PRS CODE  
INITIAL AUTO BOOT FLAG  
INITIAL 3620 PORT DATA SAVE BYTE  
RETURN TO ORIGINAL CODE



MTR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASH V1.4.1 5-Jul-80 Page 53  
 H88/H89 ADDITIONAL ROUTINES BRTAB 10:41:45 17-FEB-82

# 2110 \*\* BRTAB - BAUD RATE DIVISOR TABLE

004.101	2111 *	2112 BRTAB	EQU	*	
	2113	2114 8R96	DB	0.12	9600 BAUD
004.101 000 014	2115 8R19.2	DB	0.6		19,200 BAUD
004.103 000 006	2116 *8R38.4	DB	0.3		38,400 BAUD
	2117 *8R56.0	DB	0.2		56,000 BAUD
	2118				

000.004	2119 •	SET	#7256		
000.000	2120	ERRNZ	BRTAB/256-	TABLE MUST BE IN ONE PAGE	

# 2122 \*\*\* SAVALLX - SAVALL EXTENSION TO MAKE ROOM FOR A JUMP TO THE NMI HANDLER

004.105	2124 SAVALLX	EQU	*	REPLACE OLD CODE
004.105 345	2125	PUSH	H	SET ON STACK AS 'REGISTER'
004.106 325	2126	PUSH	D	SET RETURN ADDRESS
004.107 021 011 040	2127	LXI	D,CTLFLG	
004.112 032	2128	LDAX	D	
004.113 303 151 000	2129	JMP	SAVALLR	RETURN TO OLD CODE

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 54  
H88/H89 NON MASKABLE INTERRUPT 10:41:46 17-FEB-82

2132	****	NMI - NON MASKABLE INTERRUPT	
2133	*		
2134	*	NMI IS USED AS THE TRAP FOR ALL ILLEGAL PORT REQUESTS	
2135	*		
2136	*	PORT ADDRESSES TRAPPED ARE:	
2137	*		
2138	*	IN 3600 FRONT PANEL KEYBOARD INPUT	
2139	*	OUT 3600 FRONT PANEL CONTROL	
2140	*	OUT 3610 FRONT PANEL DISPLAY CONTROL	
2141	*	IN/OUT 3720 CONSOLE DATA FOR AN 8251A	
2142	*	OUT 3730 CONSOLE CONTROL FOR AN 8251A	
2143	*		
2144	*		
2145	*	THESE PORT REQUESTS ARE RESPONDED TO AS FOLLOWS:	
2146	*		
2147	*	IN 3600 RETURNS WITH (A) = 3770 TO SHOW THAT	
2148	*	NO FRONT PANEL SWITCHES ARE PRESSED	
2149	*		
2150	*	OUT 3600 MOVES BIT 6 (CB.CLI) TO BIT 1, AND	
2151	*	BIT 4 (CB.SSI) INVERTED, TO BIT 0, AND	
2152	*	OUTPUTS THESE BITS TO PORT 3620 TO	
2153	*	CONTROL THE CLOCK AND SINGLE STEP INTERRUPTS	
2154	*		
2155	*	OUTPUTS TO 3610, 3720, AND 3730 JUST RETURN	
2156	*		
2157	*	INPUTS FROM 3610, 3720, AND 3730 RETURN WITH (A) = 0	
2158	*	TO INDICATE AN EMPTY BUSS	
2159	*		
2160	*		
2161	*	ENTRY NONE	
2162	*		
2163	*	EXIT NONE	
2164	*		
2165	*	USES (A) ONLY IF "FAKING" AN INPUT	
2166	*		
2167	*		
004.116	343	XTHL	GET RETURN ADDRESS FROM STACK
004.117	042 064 040	SHLD	SAVE FOR LATER USE
004.122	343	XTHL	PUT RETURN ADDRESS BACK ON STACK
2171	*		
004.123	345	PUSH H	SAVE REGISTERS
004.124	305	PUSH B	
004.125	365	PUSH PSW	
2174	*		
004.126	107	MOV B,A	SAVE (A) PRIOR TO I/O
004.127	052 064 040	LHLD	GET RETURN ADDRESS
004.132	053	DCX H	BACK UP TO PORT # WHICH GOT US HERE
004.133	176	MOV A,H	GET PORT #
2178	*		
2179	*		
004.134	376 360	CPI	PORT 360?
004.136	050 033	JR Z,NM11	IF PORT WAS 3600
2181	*		
2182	*		
2183	*	PORT REFERENCED WAS 3610, 3720, OR 3730	
2184	*		
2185	*		
004.140	376 361	CPI	3610
004.142	050 010	JR Z,NM10.5	MAKE SURE PORT IS LEGAL
2186	*		
2187	*		

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 55  
H88/H89 NON MASKABLE INTERRUPT 10:41:46 17-FEB-82

004.144	376 372	2188	CPI	3720	
004.146	050 004	2189	JR	Z,NM10.5	
		2190			
004.150	376 373	2191	CPI	3730	
004.152	040 062	2192	JR	NZ,NM12.5	IF NONE OF THE ABOVE, EXIT
		2193			
004.154	053	2194	DCX	H	POINT TO IN/OUT INSTRUCTION
004.155	176	2195	MOV	A,M	SEE IF INPUT OR OUTPUT
004.156	376 323	2196	CPI	MI,OUT	
004.160	050 054	2197	JR	Z,NM12.5	IF OUTPUT, JUST EXIT
		2198			
004.162	376 333	2199	CPI	MI,IN	
004.164	040 050	2200	JR	NZ,NM12.5	IF NOT INPUT EITHER, ILLEGAL SO EXIT
		2201			
004.166	361	2202	POP	PSW	RESTORE FLAGS
004.167	076 000	2203	MVI	A,0	ELSE, RETURN LIKE AN EMPTY BUSS
004.171	030 044	2204	JR	NM13	EXIT
		2205			
004.173	053	2206	DCX	H	POINT TO IN/OUT INSTRUCTION
004.174	176	2207	MOV	A,M	GET I/O INSTRUCTION
004.175	376 333	2208	CPI	MI,IN	INPUT?
004.177	040 005	2209	JR	NZ,NM11.5	IF NOT "IN"
		2210			
004.201	361	2211	POP	PSW	RESTORE FLAGS
004.202	076 377	2212	MVI	A,11111111B	SHOW "NO KEYS PRESSED"
004.204	030 031	2213	JR	NM13	EXIT
		2214			
004.206	376 323	2215	CPI	MI,OUT	MAKE SURE INSTRUCTION IS AN "OUT"
004.210	040 024	2216	JR	NZ,NM12.5	IF NOT
		2217			
004.212	170	2218	MOV	A,B	GET OUTPUT DATA AGAIN
004.213	346 120	2219	ANI	CB,CLI+CB.SSI	MOVE CLOCK INFO TO BIT 1
004.215	017	2220	RRC		
004.216	017	2221	RRC		
004.217	017	2222	RRC		
004.220	017	2223	RRC		
004.221	017	2224	RRC		
004.222	070 001	2225	JR	C,NM12.2	
004.224	074	2226	INR	A	
004.225	041 066 040	2227	LXI	H,DATA	OR WITH THE BYTE IN RAM
004.230	266	2228	ORA	M	BEFORE OUTPUT IT
004.231	323 362	2229	OUT	H88-CTL	SET IN HARDWARE
004.233	346 374	2230	ANI	1111100B	
004.235	167	2231	MOV	M,A	
		2232			
004.236	361	2233	POP	PSW	RESTORE (A,F)
		2234			
004.237	301	2235	POP	B	
004.240	341	2236	POP	H	
		2237	* RETN		Z80 RETURN FROM NMI
004.241	355 105	2238	DB	3550,1050	

2242 \*\* ATB - AUTO 800T ROUTINE CONTINUE  
2243

004.243 167 2244 ATB MOV M,A SET AUTO 800T FLAG  
004.244 076 012 2245 MVI A,10 SET TO AUTO 800T ROUTINE  
004.246 315 052 003 2246 CALL LRA.  
004.251 021 205 001 2247 LXI D,AUTO80 SET AUTO 800T ROUTINE  
004.254 030 017 2248 JR 800TX

000.000 2250 ERRHI 4256A-\*  
004.256 2251 ORG 4256A

2252 \*\* 800T H-17 OR 247 ENTRY POINT FOR H88  
2253 \*

2254 \* ENTRY NONE  
2255 \*

2256 \* EXIT (DE) = NORMAL 800T ROUTINE ADDRESS  
2257 \*

2258 \* USES ALL  
2259

004.256 041 234 006 2260 800T LXI H,MSG.8T COMPLETE 800T MESSAGE  
004.261 315 100 006 2261 CALL TYPMSG

004.264 363 2262 DI

004.265 076 012 2263 MVI A,10

004.267 315 052 003 2264 CALL LRA. GET LOCATION OF USER PC

004.272 021 261 001 2265 LXI D,N800T SET ITS VALUE TO THE NORMAL 800T ROUTINE

004.275 163 2266 800TX MOV M,E

004.276 043 2267 INX H

004.277 162 2268 MOV M,D

004.300 373 2269 EI

004.301 303 063 000 2270 JMP GO. 00 IT

MIR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 57  
TIME OUT FOR BOOT 10:41:49 17-FEB-82

```

2273 **      TMOUT - BOOT CODE TIME OUT ROUTINE
2274 *
2275 *      TMOUT IS ENTERED FROM TIMER INTERRUPT EVER 100 MS. AND IT WILL
2276 *      EXIT: IF BOOT SUCCESS THEN TIMER OFF.
2277 *      IF 15 SECONDS TIME OUT AND BOOT IS NOT SUCCESS YES
2278 *      THEN ABORT BOOT Z47 & TO MONITOR LOOP
2279 *      IF < 15S & 3.5S THEN RE-BOOT
2280 *
2281 *      NOTE: Because the H37 and H67 run with interrupts disabled
2282 *      during portions of the code, they handle their own
2283 *      time outs.
2284 *
2285 *      ENTRY: (TMFG) = 1 IF THE TIME OUT IS FOR Z47
2286 *      = 0 IF THE TIME OUT IS FOR H17
2287 *      EXIT: NONE
2288 *
2289 *      USE: ALL (WHEN RETURN, ALL REGISTERS ARE RESTORED)
2290
004.304      EQU *
004.304      IN SC,ACE+UR,LSR      INPUT ACE LINE STATUS REGISTER
004.306      ANI UC,DR              SEE IF THERE IS A DATA READY
004.310      JR Z,TMOUT4           CHECK IF IT IS <DELETE>
2295
004.312      IN SC,ACE+UR,RBR      INPUT DATA FROM KB
004.314      ANI 01111111B        IS IT <DEL>?
004.316      CPI A,DEL            IF IT, ABORT THE BOOT
004.320      JZ NODEV             ELSE IGNORE THE INPUT
2300 *
004.323      LXI TMOUT4, LXI H,TMFG
004.326      MOV A,M
004.327      ANA A
004.330      DB MI,EXAF          SAVE Z FLAG
004.331      LDA TICCNT          GET LOW ORDER COUNTER
004.334      ANA A              SET ZERO FLAG
004.335      JR NZ,TMOUT2        NOT IN 0.5 SECOND
004.337      INX H              SET TO MYCNT
000.000      ERNZ MYCNT-TMFG-1  MYCNT MUST FOLLOW TMFG
004.340      INR M              INCREASE THE COUNT FOR 0.5 SECOND
004.341      MOV A,M
004.342      CPI 30             CHECK IF MORE THAN 15 SECONDS
004.344      JNC NDEV           NO DEVICE ?
004.347      SBI TMOUT1         IS IT 3.5 SECONDS?
004.351      JR C,TMOUT2        IF NOT, WAIT
004.353      JR NZ,TMOUT1       CHECK MORE
004.355      DB MI,EXAF
004.356      JNZ RETRY          IF IT IS Z47, THEN RE-BOOT
004.361      JR TMOUT3          IT IS H-17, CONTINUE IT CLOCK ROUTINE
004.363      DB MI,EXAF        CHECK IT IS Z47 OR H17
004.364      RNZ Z47, THEN RETURN
004.365      JMP CLOCK17       CONTINUE H17 CLOCK ROUTINE

```

000.000	2325	ERRM1	4370A-*
004.370	2326	ORG	4370A
	2327	SUBM -	SUBSTITUTE MEMORY
	2328	*	
	2329	*	SUBM INPUTS A MEMORY ADDRESS FROM THE CONSOLE AND THEN DISPLAYS
	2330	*	THAT ADDRESS AND ITS CONTENTS. IF A CARRIAGE RETURN IS THEN TYPED,
	2331	*	CONTROL RETURNS TO THE MONITOR. IF A SPACE IS TYPED, THE NEXT
	2332	*	MEMORY LOCATION AND CONTENTS ARE DISPLAYED. IF A MINUS SIGN IS
	2333	*	TYPED, THE PREVIOUS MEMORY LOCATION AND CONTENTS ARE DISPLAYED.
	2334	*	IF AN OCTAL CHARACTER IS TYPED, A BYTE IS ENTERED AND PLACED AT THE
	2335	*	CURRENT MEMORY LOCATION.
	2336	*	
	2337	*	
	2338	*	ENTRY NONE
	2339	*	EXIT NONE
	2340	*	USES A,E,H,L,F
	2341	*	
	2342	*	
004.370	041 201 006	2343	SUBM LXI H,MSG.SUB COMPLETE SUBSTITUTE MESSAGE
004.373	315 100 006	2344	CALL TYPMSG
004.376	315 012 015	2345	CALL IROC INPUT FIRST CHARACTER
005.001	320	2346	RNC IF A RETURN, EXIT
		2347	
005.002	041 003 040	2348	LXI H,IORR+1 ELSE, INPUT STARTING ADDRESS
005.005	026 015	2349	MVI D,A.CR ENDING WITH A RETURN
005.007	315 023 015	2350	CALL IOA
005.012	353	2351	XCHG (H,L) = INPUT ADDRESS
		2352	
005.013	315 064 015	2353	SUBM1 CALL TOA TYPE CRLF, ADDRESS, AND A SPACE
005.016	176	2354	MOV A,H GET MEMORY CONTENTS FOR DISPLAY
005.017	315 077 015	2355	CALL T08
005.022	076 040	2356	MVI A, ' ' SPACE
005.024	315 302 003	2357	CALL WCC
		2358	
005.027	315 051 015	2359	SUBM2 CALL IOC INPUT FIRST CHARACTER
005.032	322 075 005	2360	JNC SUBM7 IF FIRST CHARACTER IS OCTAL
		2361	
005.035	376 040	2362	CPI ' ' SPACE?
005.037	302 046 005	2363	JNZ SUBM4 IF NOT A SPACE
		2364	
005.042	043	2365	SUBM3 INX H POINT TO NEXT ADDRESS
005.043	303 013 005	2366	JMP SUBM1 DISPLAY NEXT
		2367	
005.046	376 055	2368	SUBM4 CPI ' - ' MINUS?
005.050	302 062 005	2369	JNZ SUBM6 IF NOT
		2370	
005.053	315 302 003	2371	SUBM5 CALL WCC ECHO HYPHEN
005.056	053	2372	DCX H POINT TO PREVIOUS ADDRESS
005.057	303 013 005	2373	JMP SUBM1 DISPLAY PREVIOUS
		2374	
005.062	376 015	2375	SUBM6 CPI A.CR RETURN?
005.064	310	2376	RZ IF RETURN, EXIT
		2377	
005.065	076 007	2378	MVI A,A.BEL ELSE, DING BELL
005.067	315 302 003	2379	CALL WCC
005.072	303 027 005	2380	JMP SUBM2 TRY AGAIN

MTR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 59  
 SUBSTITUTE MEMORY 10:41:53 17-FEB-82

```

2381      005.075 066 000      2382 SUBM7 MVI M,0      ZERO BYTE TO BE BUILT
2383
005.077 315 302 003      2384 SUBM8 CALL WCC      ECHO OCTAL CHARACTER
005.102 315 352 014      2385 CALL SUBM10
005.105 315 051 015      2386 SUBM9 CALL IOC      INPUT NEXT CHARACTER
005.110 322 077 005      2387 JNC SUBM8      IF OCTAL
2388
005.113 376 040      2389 CPI ' '      SPACE?
005.115 312 042 005      2390 JZ SUBM3      IF SPACE, DISPLAY NEXT BYTE
2391
005.120 376 055      2392 CPI '-'      MINUS?
005.122 312 053 005      2393 JZ SUBM5      IF MINUS, DISPLAY PREVIOUS
2394
005.125 376 015      2395 CPI A,CR      RETURN?
005.127 310      2396 RZ      IF RETURN, EXIT
2397
005.130 076 007      2398 MVI A,A,BEL      ELSE, DING BELL
005.132 315 302 003      2399 CALL WCC
005.135 303 105 005      2400 JMP SUBM9      TRY AGAIN
2401
2403 **      IROC - INPUT A RETURN OR AN OCTAL CHARACTER
2404 *
2405 *      IROC INPUTS A CHARACTER FROM THE CONSOLE AND WAITS UNTIL IT
2406 *      RECEIVES EITHER A VALID OCTAL CAHRACTER OR A CARRIAGE RETURN
2407 *
2408 *      ENTRY NONE
2409 *      EXIT (A) = INPUT CHARACTER
2410 *      'C' = SET IF CHARACTER IS OCTAL
2411 *      USES A,F
2412
2413
005.140 315 262 003      2414 IROC0 CALL RCC      INPUT CHARACTER
005.143 376 015      2415 CPI A,CR      RETURN?
005.145 310      2416 RZ      IF A CR
2417
005.146 376 060      2418 CPI '0'      < 0?
005.150 332 156 005      2419 JC IROC1      IF < OCTAL
2420
005.153 376 070      2421 CPI '8'      > 8?
005.155 330      2422 RC      IF OCTAL
2423
005.156 076 007      2424 IROC1 MVI A,A,BEL      ELSE, RING BELL
005.160 315 302 003      2425 CALL WCC
005.163 303 012 015      2426 JMP IROC      TRY AGAIN

```

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 60  
SUPPORT ROUTINES IOA1 10:41:55 17-FEB-82

2428	**	IOA1 - INPUT OCTAL ADDRESS
2429	*	
2430	*	IOA1 IS A CONTINUATION OF *IOA* AND INPUTS A SPLIT OCTAL ADDRESS
2431	*	WITHOUT REQUIRING LEADING ZEROS
2432	*	
2433	*	ENTRY (H,L) = ADDRESS + 1 WHERE INPUT ADDRESS IS TO BE PLACED
2434	*	(A) = FIRST OCTAL CHARACTER IF 'C' IS SET
2435	*	EXIT (O,E) = INPUT ADDRESS
2436	*	(A) = LAST INPUT CHARACTER
2437	*	USES A,D,E,H,L,F
2438	*	
2439	*	
005.166	305	IOA1 PUSH 8 SAVE (B,C)
005.167	102	MOV B,D (B) = TERMINATION CHARACTER
005.170	036 000	MVI E,0 CLEAR PSEUDO FLAGS
005.172	345	PUSH H SAVE ADDRESS WHERE INPUT IS TO BE PLACED
005.173	041 000 000	LXI H,0 SET NEW VALUE TO ZERO
005.176	324 262 003	CNC R,C IF CARRY SET, FIRST CHARACTER IS IN ACC
005.201	315 271 005	CALL IOC. CHECK VALIDITY
005.204	332 230 005	JC IOA3 IF < OCTAL
2448	*	
005.207	315 302 003	CALL MCC ECHO OCTAL CHARACTER
005.212	346 007	ANI 0000111B GET BINARY VALUE
005.214	365	PUSH PSH SAVE NEW CHARACTER VALUE
005.215	051	DAD H SHIFT THREE TO MAKE ROOM FOR NEW CHARACTER
005.216	051	DAD H
005.217	051	DAD H
005.220	365	PUSH PSH SAVE CARRY FROM DAD
005.221	321	POP D SAVE FLAG RESULT IN E
005.222	361	POP PSH RETURN NEW CHARACTER VALUE TO (A)
005.223	205	ADD L
005.224	157	MOV L,A
005.225	303 176 005	JMP IOA2 SEE IF MORE CHARACTERS
2461	*	
005.230	270	CMP 8 TERMINATING CHARACTER?
005.231	312 245 005	JZ IOA4 IF EQUAL
2464	*	
005.234	076 007	MVI A,A,BEL ELSE, DING BELL
005.236	315 302 003	CALL MCC
005.241	247	ANA A CLEAR CARRY
005.242	303 176 005	JMP IOA2
2470	*	
2471	*	END OF INPUT, PUT VALUE IN MEMORY AND EXIT
2472	*	
005.245	315 302 003	CALL MCC ECHO CHARACTER
005.250	127	MOV D,A LAST CHARACTER TO D
005.251	325	PUSH D
005.252	361	POP PSH (PSH) = RESULT OF DAD
005.253	174	MOV A,H MAKE (H) INTO SPLIT OCTAL
005.254	037	RAR
005.255	147	MOV H,A
005.256	172	MOV A,D RESTORE LAST INPUT CHARACTER
005.257	353	XCHG (O,E) = INPUT ADDRESS
005.260	341	POP H (H,L) = LOCATION TO PLACE THIS ADDRESS
005.261	162	MOV M,D
2483	*	



Unix H8ASM V1.4.1 5-Jul-80  
10:41:56 17-FEB-82

#09.02.01.

MTR90-1 - H/Z-89 MONITOR

## SUPPORT ROUTINES

IOA1

```

005.262 053      2484      DCX      H
005.263 163      2485      MOV      M,E
005.264 301      2486      POP      B
005.265 311      2487      RET

```

RESTORE (B,C)

## IOC - INPUT OCTAL CHARACTER

```

2489 **          IOC - INPUT OCTAL CHARACTER
2490 *
2491 *
2492 *          ENTRY      NONE
2493 *          EXIT      (A) = INPUT CHARACTER
2494 *          *C = SET IF CHARACTER NOT OCTAL
2495 *          *F
2496 *
2497 *          INPUT CHARACTER
2498 *          CALL      RCC
2499 *          CPI      '0'
2500 *          RC
2501 *
2502 *          CHARACTER > OCTAL?
2503 *          *C IF GREATER THAN
2504 *          RET

```

## TOA - TYPE OCTAL ADDRESS

TOA OUTPUTS TO THE CONSOLE A CRLF, THE SPECIFIED ADDRESS AND A SPACE

ENTRY (H,L) = ADDRESS TO BE DISPLAYED

EXIT NONE

USES A,B,C,F

HVI A,A,CR

CALL WCR.

MOV A,H

CALL T080

MOV A,L

CALL T080

HVI A,' '

JMP MCC

SPACE

```

2526 **      TOB - TYPE OCTAL BYTE
2527 *
2528 *      TOB OUTPUTS TO THE CONSOLE IN OCTAL, THE BYTE IN A
2529 *
2530 *
2531 *      ENTRY (A) - BYTE TO BE OUTPUT
2532 *      EXIT NONE
2533 *      USES A,F
2534
005.322 305      TOB0      PUSH B
005.323 006 002      MVI B,2
005.325 117      2537      NUMBER OF CHARACTERS - 1
005.326 247      2538      SAVE ORIGINAL BYTE
005.327 037      2539      ANA A
005.330 037      2540      RAR
005.331 037      2541      RAR
005.332 037      2542      RAR
005.333 037      2543      RAR
005.334 037      2544      RAR
005.335 346 007      2545      ANI 00000111B
005.337 366 060      2546      ORI 00110000B
005.341 315 302 003 2547      CALL MCC
005.344 171      2548      MOV A,C
005.345 005      2549      DCR B
005.346 302 332 005 2550      JNZ TOB1
005.351 346 007      2551      ANI 00000111B
005.353 366 060      2552      ORI 00110000B
005.355 301      2553      POP B
005.356 303 302 003 2554      JMP MCC
005.356 303 302 003 2555      JMP MCC

2557 **      MCR - WAIT FOR A CARRIAGE RETURN
2558 *
2559 *      MCR INPUTS CHARACTERS FROM THE CONSOLE UNTIL A CARRIAGE RETURN
2560 *      IS RECEIVED AND THEN ECHOS A CR LF
2561 *
2562 *
2563 *      ENTRY NONE
2564 *      EXIT NONE
2565 *      USES A,F
2566
005.361 315 262 003 2567      CALL RCC
005.364 376 015      2568      CPI A,CR
005.366 040 371      2569      JR NZ,MCR
005.370 315 302 003 2570      CALL MCC
005.373 076 012      2571      MVI A,A,LF
005.375 303 302 003 2572      JMP MCC
005.375 303 302 003 2573      JMP MCC
005.375 303 302 003 2574      JMP MCC
005.375 303 302 003 2575      JMP MCC

```

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 63  
SUPPORT ROUTINES 10:41:58 17-FEB-82

2576 \*\* VIEW3 - \*VEIW\* CONTINUATION

2577 \*

2578

006.000 302 066 002 2579 VIEW3 JNZ VIEW2 IF NOT END OF LINE  
006.003 315 355 003 2580 CALL VIEW9 END OF LINE, RESTORE ADDRESS  
006.006 076 015 2581 MVI A,A,CR  
006.010 303 355 007 2582 JMP VIEW3A. DO ASCII STUFF  
000.010 2583 EARM 6023A-\*  
006.023 2584 ORG 6023A

2586 \*\*\* DAT - DATA BYTE OUTPUT TO Z-47

2587 \*

2588 \* ENTRY: (A) = BYTE TO OUTPUT

2589 \*

2590 \* EXIT: (A) = BYTE TO OUTPUT

2591 \*

2592 \* (D) = S.DTR

2593 \*

2594 \* USE: AF, D

2595 \*

2596 \* EQU \*

2597 \* MVI D,S,DTR

2598 \* JR COM1

2599 \* EARM 6027A-\*

2600 \* ORG 6027A

SET MATCH CONDITION TO DATA TRANSFER  
REQUEST BIT

2601 \*\*\* COM - OUTPUT COMMAND BYTE TO Z-47

2602 \*

2603 \* ENTRY: (A) = COMMAND BYTE

2604 \*

2605 \* EXIT: (A) = COMMAND BYTE

2606 \*

2607 \* (D) = S.DON

2608 \*

2609 \* USE: AF, D

2610 \*

2611 \* EQU \*

2612 \* MVI D,S,DON

2613 \* PUSH PSW

2614 \* CALL IN.

2615 \* ANA D

2616 \* JR Z,WTDON1

2617 \* POP PSW

2618 \* JMP COM2

SET MATCH CONDITION TO DONE BIT

READ CONTROLLER STATUS REGISTER

GET MATCH BIT ONLY

IF NO MATCH, WAIT

CONTINUE \*COM\* ROUTINE

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 64  
 SUPPORT ROUTINES 10:41:59 17-FEB-82

000.001	2619	ERRMI	6045A--*
006.045	2620	ORG	6045A
	2621	**	HRNX - HORN EXTENSION ROUTINE
	2622	*	
	2623	*	THIS IS AN EXTENSION TO *HORN* TO MAKE ROOM FOR A JUMP
	2624		
006.045	056 011	HRNX	MVI L,#CILFLG
006.047	163	MOV	M,E
006.050	321	POP	D
006.051	341	POP	H
006.052	311	RET	
	2629		
	2631	**	NOISE - DING BELL ON CONSOLE
	2632	*	
	2633	*	THIS IS A MODIFICATION TO ALLOW THE H88/H89 TO USE THE CONSOLE BELL
	2634		
006.053	076 007	NOISE	MVI A,A,BEL
006.055	315 302 003	CALL	MCC
006.060	303 140 002	JMP	HORN
	2637		CONTINUE WITH NORMAL HORN DELAY
	2639	**	OUT. - OUTPUT BYTE TO Z-47
	2640	*	
	2641	*	ENTRY: (A) = OUTPUT BYTE
	2642	*	
	2643	*	EXIT: NONE
	2644	*	
	2645	*	USE: NONE
	2646		
006.063		OUT.	EQU *
006.063	305	PUSH	B
006.064	107	MOV	B,A
006.065	072 120 041	LDA	PRIM
006.070	117	MOV	C,A
006.071	170	MOV	A,B
	2652	OUT	(C),A
	2653	*	
006.072	355 171	DB	3550,171Q
006.074	301	POP	B
006.075	311	RET	
	2656		
	2658	ERRMI	6100A--*
000.002	2659	ORG	6100A
006.100	2660	**	TYPMSG - TYPE MESSAGE TO CONSOLE
	2661	*	
	2662	*	TYPMSG OUTPUTS AN ASCII MESSAGE FROM MEMORY TO THE CONSOLE
	2663	*	UNTIL A NULL IS SENSED
	2664	*	
	2665	*	ENTRY (H,L) = ADDRESS OF MESSAGE

Unix H8ASH V1.4.1 5-Jul-80 Page 65  
10:41:59 17-FEB-82

MIR90-1 - H/Z-89 MONITOR #09.02.01.  
SUPPORT ROUTINES

TYPMSG

2666 \* EXIT NONE  
2667 \* USES A,H,L,F  
2668

006.100 176 TYPMSG MOV A,M GET CHARACTER  
006.101 267 ORA A SEE IF A NULL  
006.102 310 RZ IF NULL, EXIT  
2673

006.103 315 302 003 2674 CALL WCC ELSE OUTPUT CHARACTER TO CONSOLE  
006.106 043 2675 INX H POINT TO NEXT CHARACTER  
006.107 030 367 2676 JR TYPMSG OUTPUT IT

2678 \*\* RDBLCK - INPUT A BLOCK FROM Z-47  
2679 \*

2680 \* RDBLCK READS IN A BLOCK FROM THE DISK CONTROLLER  
2681 \*

2682 \* ENTRY: HL = LOAD ADDRESS  
2683 \* C = SIDE/UNIT/SECTOR  
2684 \*

2685 \* EXIT: BLOCK IN READ IN MEMORY  
2686 \*

2687 \* USES: ALL  
2688 \*

006.111 076 007 2690 RDBLCK MVI A,DD-REAB  
006.113 315 027 006 2691 CALL COM SEND THE COMMAND  
006.116 257 023 006 2692 XRA A FOR TRACK 0  
006.117 315 023 006 2693 CALL DAT SEND IT TO DISK  
006.122 171 2694 MOV A,C LOAD SIDE/UNIT/SECTOR  
006.123 315 023 006 2695 CALL DAT SEND IT TO DISK  
2696

006.126 315 067 001 2697 RD2 GET STATUS  
006.131 332 104 010 2698 JC WDN JC SET IF S.DON  
2699

006.134 167 2700 MOV M,A  
006.135 043 2701 INX H  
006.136 030 366 2702 JR RD2 CONTINUE TRANSFER

2704 \*\* OUTL. - OUTPUT BYTE TO PORT (PRIM+1)  
2705 \*

2706 \* ENTRY: (A) = OUTPUT PORT  
2707 \*

2708 \* EXIT: NONE  
2709 \*

2710 \* USES: NONE  
2711

006.140 2712 OUTL. EQU \*

006.140 305 2713 PUSH B  
006.141 107 2714 MOV B,A  
006.142 072 120 041 2715 LDA PRIM SAVE THE OUTPUT DATA  
GET PORT ADDRESS

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 66  
SUPPORT ROUTINES 0011. 10:42:00 17-FEB-82

006.145 074 2716 INR A SET TO (PRIM+1)  
006.146 030 320 2717 JR OUT.1 GO TO OUTPUT ROUTINE

2719 \*\* INL. - INPUT BYTE FROM (PRIM+1) PORT

2720 \*  
2721 \* ENTRY: NONE  
2722 \*  
2723 \* EXIT: (A) = INPUT BYTE  
2724 \*  
2725 \* USE: A

006.150 2726 EQU \*  
006.150 305 2727 INL. EQU \*  
006.150 305 2728 PUSH B  
006.151 072 120 041 2729 LDA PRIM  
006.154 074 2730 INR A GET PORT ADDRESS  
006.155 247 2731 ANA A SET TO (PRIM+1)  
006.156 030 014 2732 JR IN.1 GO TO INPUT ROUTINE

000.005 2734 ERRMI 6165A-\*  
006.165 2735 ORG 6165A  
2736 \*\* MSG.GD - (G)D  
2737 \*  
2738 \* \*GD\*

006.165 157 040 000 2740 MSG.GD DB '0',0

2742 \*\* IN. - INPUT BYTE FROM PORT (PRIM)

2743 \*  
2744 \* ENTRY: NONE  
2745 \*  
2746 \* EXIT: (A) = INPUT BYTE  
2747 \*  
2748 \* USE: A

006.170 2749 EQU \*  
006.170 305 2750 IN. EQU \*  
006.170 305 2751 PUSH B  
006.171 072 120 041 2752 LDA PRIM GET PORT ADDRESS  
006.174 117 2753 IN.1 MOV C,A SET ADDR. TO REG C  
2754 \* IN A,(C)  
006.175 355 170 2755 DB 3550,1700 INPUT BYTE  
006.177 301 2756 POP B  
006.200 311 2757 RET

MTR90-1 - H/Z-89 MONITOR #09.02.01. Unix HBASM V1.4.1 5-Jul-80 Page 67  
SUPPORT ROUTINES 10:42:01 17-FEB-82 MSG.SUB

000.000 2759 ERRMI 6201A--\*  
006.201 2760 OKG 6201A  
2761 \*\* MSG.SUB - (S)UBSTITUTE  
2762 \*  
2763 \* "SUBSTITUTE"  
2764  
006.201 165 142 163 2765 MSG.SUB D8 'ubstitute ',0

2767 \*\* MSG.PC - (P)ROGRAM COUNTER  
2768 \*  
2769 \* "PROGRAM COUNTER"  
2770  
006.214 162 157 147 2771 MSG.PC D8 'rogram Counter ',0

2773 \*\* MSG.BT - (B)OOT  
2774 \*  
2775 \* "BOOT"  
2776  
006.234 157 157 164 2777 MSG.BT D8 'oot ',0

MTR90-1 - H/2-89 MONITOR #09.02.01. Unix H8ASM V1.4.1 5-Jul-80 Page 68  
 SPEED - ROTATIONAL SPEED TEST FOR H89 DISK DRIVE 10:42:02 17-FEB-82

2780 \*\*\* SPEED - ROTATIONAL SPEED TEST FOR 5.25 INCH DISK DRIVE  
 2781 \*  
 2782 \* \*SPEED\* IS USED ONLY FOR GROSS ADJUSTMENT OF DRIVE ROTATIONAL  
 2783 \* SPEED IF THE FIRST READ/WRITE TEST OF THE UNIT FAILS DURING SET UP.  
 2784 \*

2785 \* USE OF \*SPEED\* IS AS FOLLOWS:

2786 \*  
 2787 \* 1. ENTER \*GO AND THE ENTRY ADDRESS OF \*SPEED\*

2788 \* 2. ADJUST DRIVE SPEED UNTIL DATA AT DISPLAYED

2789 \* EQUALS 200

2790 \* A. IF SPEED < 200, TURN ADJUSTMENT CLOCKWISE

2791 \* B. IF SPEED > 200, TURN COUNTERCLOCKWISE

2792 \*  
 2793 \* THE ABOVE TEST ADJUSTS SY0:. TO ADJUST SY1:, USE H00S

2795 \*\* LABLE EQUIVALENCES

2796 \*

2797 \* I/O PORTS

2798 OP.DC EQU 177Q DRIVE CONTROL OUTPUT PORT

2799 IP.DS EQU 177Q DRIVE STATUS INPUT PORT

2801 \* MASKS

2802 \*

2803 DS.HOLE EQU 00000001B DRIVE STATUS SECTOR/INDEX HOLE

2805 \* CONSTANTS

2806 \*

2807 ONDR0 EQU 022Q TURN ON SY0:

000.001

000.022