```
 5>POKE 23582,43
 6 LET restart=9710: GO TO 9700
 8 CLS : OUT 126,3
10 PRINT "Display Baud Rates"
12 PRINT "Baud Jumper Divisor"
14 PRINT "115200 15-16 16"
16 PRINT " 57600 13-14 16"
18 PRINT " 38400 11-12 16"
20 PRINT " 28800  9-10 16"
22 PRINT " 19200  7-8  16"
24 PRINT " 14400  5-6  16"
26 PRINT "  9600  3-4  16"
28 PRINT "  7200  1-2  16"
30 PRINT "  4800  7-8  64"
32 PRINT "  3600  5-6  64"
34 PRINT "  2400  3-4  64"
36 PRINT "  1800  1-2  64"
37 INPUT "Paused, enter to continue.";a$
38 CLS : PRINT "Select Clock Rate"
40 PRINT "Reset & Clock Divider Select"
42 PRINT "0...Divide by 1"
44 PRINT "1...Divide by 16"
46 PRINT "2...Divide by 64"
48 PRINT "3...Reset"
50 INPUT "Select Desired ";g
52  CLS : PRINT "Select Mode"
54 PRINT "Bits,Parity,Stop-Bits"
56 PRINT "0...7,E,2"
58 PRINT "1...7,O,2"
```

```
 60 PRINT "2...7,E,1"
 62 PRINT "3...7,O,1"
 64 PRINT "4...8, ,2"
 66 PRINT "5...8, ,1"
 68 PRINT "6...8,E,1"
 70 PRINT "7...8,O,1"
 72 INPUT "Select Desired ";i
 74 LET g=g+(i*4)
 76 CLS : PRINT "Select RTS and Interrupt"
 78 PRINT "0...RTS = Low, TX-Int Off"
 80 PRINT "1...RTS = Low, TX-Int On"
 82 PRINT "2...RTS = Hi,  TX-Int Off"
 84 PRINT "3...RTS = Low, Break, TX-Int Off"
 86 INPUT "Select Desired ",i
 88 LET g=g+(32*i)
 90 CLS : PRINT "Receiver Interrupt"
 92 INPUT "Enable Rcvr Interrupt Y/N ";j$
 94 IF j$="Y" OR j$="y" THEN  LET g=g+128
 96 OUT 126,3
 98 OUT 126,g
100 INPUT "Press Enter to Continue ";i$
199 GO TO restart
499 STOP
500 REM ******* Get System Vars
501 LET ps=(256*PEEK 23636)+PEEK 23635
502 LET se=(256*PEEK 23654)+PEEK 23653
503 LET vars=(256*PEEK 23628)+PEEK 23627
504 LET chars=(256*PEEK 23607)+PEEK 23606
505 LET Chans=(256*PEEK 23632)+PEEK 23631
```

```
 506  LET chansExt= chans+43
 510 LET output= (256*PEEK chansext+1)+ PEEK chansext
 511LET input=PEEK (chansext+2) + (256*PEEK chansext+3)
 550 PRINT "   Program Starts: ";ps
 551 PRINT "       Stack Ends: ";se
 552 PRINT "        Variables: ";vars
 553 PRINT "       Characters: ";chars
 558 PRINT "            Chans: ";chans
 559  PRINT "  Chans Extension: ";chansExt
 560 PRINT "   Output Routine: ";output
 561 PRINT "    Input Routine: ";input
 599 STOP
 800  REM ##### Get Formatted Hex
 801 REM Entry
 802 REM Decimal "d" value to convert
 803 REM Places "plcs" number of places
 804  REM Returns P$ 4 HEX Chars
 810  RANDOMIZE USR D2H: LET P$=H$
 820 IF LEN P$<plcs THEN  LET P$="0"+P$: GO TO 820
 830 RETURN
 900 REM Get Formatted Decimal
 901  REM Entry
 902 REM Decimal "d"
 903 REM Places "plcs"
 904 REM Returns P$
 910 LET P$=STR$ d
 920 IF LEN P$<plcs THEN  LET P$=" "+P$:  GO TO 920
 930  RETURN
1000 REM **** Memory Dump Contd
```

```
1001 LET columns=15
1010 FOR n=addr TO al STEP columns+1
1015  LET P$=""
1020 LET d=n: GO SUB 900
1030 IF ah=1 THEN  GO SUB 800
1040 LET B$=P$
1050 LET plcs=4: FOR o=0 TO columns: LET taddr=n+o: LET d=PEEK taddr: GO SUB
900
1060 IF oh=1 THEN  GO SUB 800: LET P$="  "+P$(3 TO 4)
1070 LET B$=B$+P$
1080 NEXT o
1090 IF prt=1 THEN  PRINT B$
1100 IF lprt<> 0 THEN  PRINT #lprt,"",B$
1190 NEXT n
1199 STOP
2100 REM Main Loop Entry Point
2101 LET T$="X": LET Z$="": IF INKEY$ =" STOP " THEN  STOP
2103 IF INKEY$ ="^" THEN  INPUT X$: GO TO rst
2105  IF addr>al AND al<> 0 THEN  GO TO rst
2110 DIM Q$(45): LET op1=PEEK Addr: LET op2=PEEK (Addr+1): LET op3=PEEK (Addr+2)
: LET op4=PEEK (Addr+3)
2120 LET Q$(8 TO 10)=STR$ (op1)
2160 IF op1=203 THEN  GO TO 3000
2170 IF op1=221 OR op1=253 THEN  GO TO 4000
2180 IF op1=237 THEN  GO TO 5000
2190 IF op1=0 THEN  LET Q$(24 TO 36)="NOP": LET len=1: GO TO 8000
2200 REM Main Array Handler
2210 LET len=VAL M$(op1,13): IF len=1 THEN  LET Q$(24 TO 36)=M$(op1,1 TO 12): L
ET T$=M$(OP1,15): GO TO 6000
```

```
2218 LET Q$(12 TO 14)=STR$ (op2): LET ins=CODE M$(op1,14)
2220 LET T$=M$(OP1,15): IF len=2 THEN  LET q$(24 TO 36)=m$(op1,1 TO ins-1)+STR$
op2+m$(op1,ins+1 TO 12): GO TO 6000
2230 LET T$=M$(OP1,15): LET q$(16 TO 18)=STR$ op3: LET Q$(24 TO 36)=m$(op1,1 TO
ins-1)+STR$ (op3*256+op2)+m$(op1,ins+2 TO 12): GO TO 6000
2999 STOP
3000 REM CB Array Routine
3010 LET T$=C$(OP1,11): LET Q$(12 TO 14)=STR$ (op2): LET Q$(24 TO 36)=C$(op2+1,1
 TO 10): LET len=2: GO TO 6000
3999 STOP
4000 REM DD / FD Array Routine
4005  LET x=op3: IF x>127 THEN  LET x=x-256
4006 LET X$=STR$ (x+iy)
4010 LET Q$(12 TO 14)=STR$ (op2)
4070 IF op2=203 THEN  GO TO 4500
4080 FOR n=1 TO 39: IF CODE D$(n,16)=op2 THEN  GO TO 4100
4090 NEXT n: STOP
4100 LET ins2=CODE D$(n,17): LET len=VAL D$(n,13): LET ins=CODE D$(n,14)
4120 LET Q$(24 TO 36)=D$(n,1 TO 12): IF op1=253 AND ins2<> 15 THEN  LET Q$(23+in
s2)="Y"
4121 IF op1=253 AND ins2=15 THEN  LET Q$(29)="Y": LET Q$(32)="Y"
4125 LET T$=D$(N,15)
4190 IF len=2 THEN  GO TO 6000
4200 REM DD FD Len > 2
4210 LET Q$(16 TO 18)=STR$ (op3): IF len>3 THEN  GO TO 4300
4220 IF op3<128 THEN  LET Q$(23+INS TO 36)=STR$ (op3)+Q$(24+ins TO 36)
4230 IF op3>= 128 THEN  LET Q$(22+INS TO 36)=STR$ (op3-256)+Q$(24+ins TO 36)
4240 IF op1=253 THEN  LET Z$="("+X$+")"
4290 GO TO 6000
```

```
4300 REM FD DD len = 4
4310 LET Q$(20 TO 22)=STR$ (op4): IF op2<> 54 THEN  LET Q$(23+ins TO 36)=STR$ (o
p4*256+op3)+Q$(25+ins TO 36)
4315 IF op2=54 THEN  LET Q$(23+ins TO 36)=STR$ (op3)+"),"+STR$ (op4)
4320 LET T$=D$(N,15)
4390 GO TO 6000
4500 REM DD CB or FD CB Handler
4510 FOR n=40 TO 70: IF CODE D$(n,16)=op4 THEN  GO TO 4600
4520 NEXT n: STOP
4610 LET ins2=CODE D$(n,17): LET len=4: LET Q$(16 TO 18)=STR$ (op3)
4611 LET Q$(20 TO 22)=STR$ (op4): LET x=op3: IF x>127 THEN  LET x=x-256
4612 LET Q$(34 TO 38)=STR$ x+")"
4620 LET Q$(24 TO 33)=D$(n,1 TO 12): IF op1=253 THEN  LET Q$(23+ins2)="Y"
4625 LET T$=D$(N,15): IF op1=253 THEN  LET z$="("+X$+")"
4690 GO TO 6000
5000 REM ED Array Routine
5010 FOR n=1 TO 56: IF CODE E$(n,16)=op2 THEN  GO TO 5100
5020 NEXT n: STOP
5110 LET q$(24 TO 36)=e$(n,1 TO 12): LET len=VAL e$(n,13): LET Q$(12 TO 14)=STR$
 (op2)
5115 LET T$=E$(N,15)
5190 IF len=2 THEN  GO TO 6000
5210 LET ins=CODE e$(n,14): LET Q$(16 TO 18)=STR$ (op3): LET Q$(20 TO 22)=STR$ (
op4)
5220 LET q$(24 TO 36)=e$(n,1 TO ins-1)+STR$ (256*op4+op3)+e$(n,ins+2 TO 12): GO
TO 6000
6000 REM JP JR CALL & RST SUPPORT
6002 FOR n=1 TO len
6003 IF n=1 THEN  LET d=op1: LET q=39
```

```
6004 IF n=2 THEN  LET d=op2: LET q=40
6005 IF n=3 THEN  LET d=op3: LET q=41
6006 IF n=4 THEN  LET d=op4: LET q=42
6007 IF d>127 THEN  LET d=d-128
6008 IF d<32 THEN  LET d=32
6009 LET Q$(Q)=CHR$ (d): NEXT n
6010 LET N$=Q$(24 TO 26)
6020 IF N$="JP " THEN  GO TO 7200
6030 IF N$="JR " THEN  GO TO 7400
6040 IF N$="CAL" THEN  GO TO 7300
6050 IF N$="RST" THEN  LET Z$="  SOFTWARE RESTART!"
6060 IF N$<> "RET" THEN  GO TO 6099
6070 IF Q$(27)="I" THEN  LET Z$="  Int. "
6075 IF Q$(27)="N" THEN  LET Z$="  Non-MASK. Int. "
6080 IF Q$(28)>"A" THEN  LET Z$="  Cond. "
6090 LET Z$=Z$+"  RET"
6099 GO TO 8000
7199 STOP
7200 REM Jump Support
7210 LET TADDR=(256*OP3)+OP2
7215 LET X$=STR$ (taddr): IF ah=1 THEN  LET d=taddr: RANDOMIZE USR D2H: LET X$=H
$
7230 LET Z$="": IF Q$(27)="(" THEN  LET Z$="  Jump TO: "+Q$(27 TO 37)
7240 IF Q$(27)>"A" THEN  LET Z$="  Cond. "
7290 LET Z$=Z$+"Jump TO: "+X$: GO TO 8000
7300 REM CALL Routine
7310 LET taddr=(256*op3+op2): LET X$=STR$ (taddr): IF ah=1 THEN  LET d=taddr: RA
NDOMIZE USR D2H: LET X$=H$
7320 LET Z$="": IF Q$(29)>"A" THEN  LET Z$="  Cond. "
```

```
7390 LET Z$=Z$+"CALL TO: "+X$: GO TO 8000
7400 REM JR ROUTINE
7420 LET D=OP2: IF D>= 128 THEN  LET D=D-256
7440 LET taddr=D+2+ADDR+ofs
7445 LET X$=STR$ (taddr): IF ah=1 THEN  LET d=taddr: RANDOMIZE USR D2H: LET X$=H
$
7447 IF ah=1 THEN  IF LEN H$<4 THEN  LET H$="0"+H$: GO TO 7447
7450 IF Q$(27)>"A" THEN  LET Z$="  Cond. "
7490 LET Z$=Z$+"Rel. Jump TO: "+X$: GO SUB 8000: GO TO 6800
7990 STOP
8000 REM Print Routine
8002 IF oh=0 THEN  GO TO 8008
8003 LET d=op1: RANDOMIZE USR D2H: LET Q$(8 TO 10)=H$(3 TO 4)
8004 IF len>1 THEN  LET d=op2: RANDOMIZE USR D2H: LET Q$(12 TO 14)=H$(3 TO 4)
8005 IF len>2 THEN  LET d=op3: RANDOMIZE USR D2H: LET Q$(16 TO 18)=H$(3 TO 4)
8006 IF len>3 THEN  LET d=op4: RANDOMIZE USR D2H: LET Q$(20 TO 22)=H$(3 TO 4)
8008 LET d=addr+ofs: LET Q$(1 TO 5)=STR$ (ADDR+ofs)
8010 IF ah<> 1 THEN  GO TO 8040
8020 RANDOMIZE USR D2H: LET X$=H$
8025 IF LEN X$<4 THEN  LET X$="0"+X$: GO TO 8025
8030 LET Q$(1 TO 5)=X$
8040 IF PRT=1 THEN  PRINT Q$(1 TO 32)'TAB 22;Q$(33 TO 42)
8050 IF LPRT<> 0 THEN  PRINT #LPRT;Q$;Z$
8099 LET addr=addr+len: GO TO 2100
8400 REM Replace Bytes
8410 LET D=PEEK addr: RANDOMIZE USR D2H: PRINT ADDR;"   ";H$,
8411 INPUT "HEX entry assumed, suffix T for DEC) New Byte: ";V$
8412 IF V$="" THEN  GO TO 8490
8414 IF V$="-" THEN  LET ADDR=ADDR-1: PRINT : GO TO 8400
```

```
8416 IF V$="@" THEN  INPUT "NEW ADDRESS: ";ADDR: PRINT : GO TO 8410
8420 IF V$="^" THEN  PRINT : LET X$="@": GO TO 8120
8425 IF V$(LEN V$)="T" THEN  LET D=VAL V$(1 TO LEN V$-1): GO TO 8488
8426 LET X$=STR$ D
8427 IF LEN X$<4 THEN  LET X$="0"+X$: GO TO 8427
8430 IF LEN V$<4 THEN  LET V$="0"+V$: GO TO 8430
8435 LET H$=V$: RANDOMIZE USR H2D
8488 POKE addr,D
8490 LET D=PEEK addr: PRINT H$: LET addr=addr+1: GO TO 8410
9399 STOP
9400 REM HEX DUMP
9402 LET t=0
9410 INPUT "Prepare Printer.              Press Enter When Ready.";X$
9412 LET X$=STR$ addr
9414 IF LEN X$<5 THEN  LET X$=" "+X$: GO TO 9414
9416 PRINT #3;X$;" ";
9420 FOR n=addr TO addr+15: LET d=PEEK n: RANDOMIZE USR D2H: LET X$=H$(3 TO 4)+"
 ": PRINT #3;X$;
9428 NEXT n: PRINT #3;"   ";
9430 FOR n=addr TO addr+15: LET d=PEEK n
9436 IF d>127 THEN  LET d=d-128
9438 IF d<32 THEN  LET d=32
9440 PRINT #3;CHR$ (d);
9450 NEXT n: PRINT #3;" "
9460 LET t=t+1: IF t=64 THEN  LET T=0: PRINT #4;CHR$ (13)
9470 IF addr<al THEN  LET addr=addr+16: GO TO 9412
9499 GO TO rst
9699 STOP
9700 REM INIT RUN
```

```
9702 BORDER 6: LET AH=1: LET OH=1: DIM S$(416): LET TRACE=0: RESTORE 9950: FOR N
=0 TO 6: READ S$(1+(N*64)): NEXT N
9704 FOR N=0 TO 5: READ S$(80+(N*64) TO 81+(N*64)): NEXT N
9706 LET FIX=0: LET OFS=0: LET SEL=0: LET LPRT=0: LET PRT=1
9710 CLS : PRINT "0    Go! (#11 Exits)"'
9711 PRINT "1";TAB 5;"Load Point: ";addr
9712 PRINT '"2    Offset Listing: ";: IF ofs=0 THEN  PRINT "N"
9713  IF ofs<> 0 THEN  PRINT ofs
9714 PRINT '"3    Original Origin: ";: IF ofs=0 THEN  PRINT "N/A"
9715 IF ofs<> 0 THEN  PRINT org
9716 PRINT '"4    Print Op Codes in: ";: IF oh=0 THEN  PRINT "DEC"
9717 IF oh=1 THEN  PRINT "HEX"
9718 PRINT '"5    Print Adresses in: ";: IF ah=0 THEN  PRINT "DEC"
9719 IF ah=1 THEN  PRINT "HEX"
9720 PRINT '"6    Print to Screen: ";: IF prt=1 THEN  PRINT "Yes"
9721 IF prt=0 THEN  PRINT "No"
9722 PRINT '"7    Line Print To: ";: IF lprt=0 THEN  PRINT "None"
9723 IF lprt=3 THEN  PRINT "Thermal"
9724 IF lprt=4 THEN  PRINT "Channel #4"
9726 PRINT '"8    Address Limit: ";al
9730 PRINT '"9    Memory Dump"
9735 PRINT '"10   Init 6850, Req'd for Chn #4"
9740 INPUT "Configuration Selection: ";cs: IF cs<0 OR cs>11 THEN  BEEP .5,21: GO
 TO 9740
9750 GO TO (9800+(10*cs))
9800 LET x$="": GO TO 2020
9810 INPUT "New Starting Address: ";X$: IF X$(LEN X$)="H" OR X$(LEN X$)="h" THE
N  LET X$=X$(1 TO LEN X$-1): GO TO 9815
9811 LET ADDR=VAL X$: GO TO RST
```

```
9815 IF LEN X$<4 THEN  LET X$="0"+X$: GO TO 9811
9817 LET H$=X$: RANDOMIZE USR H2D: LET addr=d: GO TO rst
9820 INPUT "Offset Listing Y or N";X$: IF X$="n" OR X$="N" THEN  LET ofs=0: GO T
O rst
9830 INPUT "Original Origin of Program: ";org: IF org<0 OR org>65535 THEN  BEEP
.5,21: GO TO 9830
9833 LET ofs=org-addr: GO TO rst
9840 LET oh=0: INPUT "Print Op Codes in [H]ex or [D]ec ";X$: IF X$="H" OR X$="h"
 THEN  LET oh=1: GO TO rst
9842 IF X$<> "D" AND X$<> "d" THEN  BEEP .5,21: GO TO 9840
9844 GO TO rst
9850 LET ah=0: INPUT "Print Addr in [H]ex or [D]ec ";X$: IF X$="H" OR X$="h" THE
N  LET ah=1: GO TO rst
9852 IF X$<> "D" AND X$<> "d" THEN  BEEP .5,21: GO TO 9850
9854 GO TO rst
9860 LET prt=1: INPUT "Print to Screen Y or N ";X$: IF X$="N" OR X$="n" THEN  LET prt=0: GO TO rst
9862 IF X$<> "Y" AND X$<> "y" THEN  BEEP .5,21: GO TO 9860
9864 GO TO rst
9870 LET lprt=4: INPUT "Print to [T]hermal/#[4]/[N]one";X$: IF X$="T" OR X$="t" THEN  LET lprt=3: GO TO rst
9872 IF X$="N" OR X$="n" THEN  LET lprt=0: GO TO rst
9874 IF X$<> "4" THEN  BEEP .5,21: GO TO 9870
9876 GO TO rst
9880 INPUT "Address Limit: ";AL: GO TO rst
9889 GO TO 9710
9890 REM    LPRINT "prt = "+STR$ prt: LPRINT "lprt = "+STR$ lprt: LPRINT " "
9895  GO TO 1000
9899  STOP
9900 GO TO 8
9910 LET org=0: LET I$="": LET d2h=26900: LET h2d=26930: DIM H$(4): LET D=0: LET
```

```
 addr=0: LET al=0: LET ofs=0: LET prt=0: LET lprt=4: LET oh=1: LET ah=0
9930 LET IY=23610
9950 DATA "A","B","C","D","E","H","L","BC","DE","HL","IX","IY","SP"
9998 STOP
9999 CLS : SAVE "Disembler" LINE 5: SAVE "Disembler" LINE 5
```