THE COMPUTER HOBBYIST

"The Computer Hobbyist" is a monthly newsletter totally dedicated to the use of micro-computers and associated devices as a hobby. Both software and hardware are discussed in feature articles. Circuit diagrams and program listings frequently supplement these articles. In addition to the features, each issue contains regular columns on surplus, letters from subscribers, and want ads free to subscribers. "The Computer Hobbyist" is offset printed on looseleaf 8 1/2" X 11" paper suitable for ring binding (except for the first three issues which were folded to half size) and is mailed first class.

The reverse side of this flyer contains a reprint of "A Graphics Display for the 8008, Part 1" by Hal Chamberlin from Volume 1 Number 1. This is typical of the feature articles found in "The Computer Hobbyist". If you would like to subscribe or order a sample copy, please use the blank provided at the bottom of this page.

For those interested in reprints of articles, we included in this flyer an index to articles by issue and spaces on the order blank for back issues.

VOLUME I NUMBER 1

1. A GRAPHICS DISPLAY FOR THE 8008 PART 1 - Fundamental concepts and programming technique for inexpensive vector display
2. SURPLUS SUMMARY - THE TELETYPE STORY - Survey of teleprinter equipment suitable for computer output
3. NOTES ON THE 8008 INSTRUCTION SET - Some simple tips for more effective 8008 programming not not mentioned in Intel manuals

VOLUME I NUMBER 2

1. THE 8080 IS HERE - MITS ALTAIR 8800 product line preview and and comments on the future of hobby computers
2. LOGIC SYMBOL CONVENTIONS or HOW TO READ TCH LOGIC DIAGRAMS Discussion of MIL-STD-806C logic symbols and logic design using the "dot convention"
3. A GRAPHICS DISPLAY FOR THE 8008 PART 2 - Description and diagram of digital and analog circuitry for display generator
4. INTERFACING A 5 LEVEL TELEPRINTER PART 1 - Description of simple current loop interface with common teleprinters

VOLUME I NUMBER 3

1. A GRAPHICS DISPLAY FOR THE 8008 PART 3 - Description and diagram for XYZ scope using common magnetic deflection CRT's and photographs of display
2. BOOK REVIEW - Critical review of MICROCOMPUTER DESIGN from Martin Reserach
3. CHESS BOARD DISPLAY - Description and listing of chessboard and chesspiece display program, fits in about 500 bytes
4. INTERFACING A 5 LEVEL TELEPRINTER PART 2 - Description and listing of print software that accepts ASCII input
5. A CHEAP MARK SENSE CARD READER - Description and diagram of circuit for reading penciled data from tab cards with surplus Western Union card reader

VOLUME I NUMBER 4

1. HUMAN INTERFACE YOUR GRAPHICS DISPLAY - Description, diagram, and driver software for a 4 variable proportional control input for interactive graphics
2. 8008 VS 8080 VS IMP-16 WHICH MICROPROCESSOR FOR YOU? Detailed comparison of strengths and weaknesses of microprocessors available to hobbyists

VOLUME I NUMBER 5

1. TCH AUDIO CASSETTE DATA RECORDING STANDARD PART 1 - Rationale and description of proposed data recording standard for recording on audio cassettes
2. RANDOM NUMBER GENERATOR - Description and listing of random number generator, sequence length over 2 billion, 40 bytes
3. TCH STANDARD AUDIO CASSETTE INTERFACE - Description and diagram of machine independent audio cassette interface conforming to TCH standard
4. DRAWING WITH YOUR POT CONTROLS - Description and listing of program for interactive drawing on graphics display

A GRAPHICS DISPLAY FOR THE 8008 Part 1 by Hal Chamberlin

A vector graphics display is probably the most flexible and most desirable peripheral you can interface to a personal computer system. With such a device you and your computer can plot graphs, plot mathematical curves, draw sound waves, display and edit music scores, display text including all kinds of special symbols, generate and maintain engineering drawings, play an infinite variety of games using the screen as a playing board, generate random kaliedoscopic patterns, etc. Part 1 will describe fundamental concepts and programming techniques of the display system. Part 2 will present the digital and analog circuitry of the display generator and interface. Part 3 will discuss large screen displays and give circuits for constructing a 12 inch screen display using a readily available surplus radar CRT.

Before going further, the limitations of this system should be mentioned. Since the display is generated and refreshed directly by the 8008 program, the slow speed of the 8008 will be evidenced by display flicker when a number of items are displayed. This can be allieviated to some extent by using a long persistance phosphor screen on the monitor, developing a tolerance to the flicker, using the faster 8008-1 chip, or a combination of the three. In particular, text will be limited to around 400 displayed characters. Nevertheless the usefullness and low cost of the display system will make constructing one a most worthwhile project.

A minimum display interface would provide an X position register, a Y position register, and a beam control bit. With these the display routine could move the beam around at will by loading the coordinates of the desired positions into the X and Y registers with OUT instructions. If the beam control bit had been set on previously, a line (often called a vector) would be drawn from one point to the next. Leaving the beam control off would allow positioning without drawing. Such an interface can, theoretically, draw anything with the appropriate programming.

Some details have to be considered however if the expected results are to be obtained. If the lines between end points are to be straight, it is essential that the X and Y registers change simultaneously. Also a special analog circuit known as a vector generator is usually necessary for uniform line brightness and minimum bandwidth requirements of the deflection amplifiers. Since a 4 bit X and Y resolution is not adequate, and the 8008 only outputs 8 bits at a time, some method of simultaneously updating X and Y will have to be found. Additionally, it would be nice to be able to avoid having a seperate OUT instruction to turn the beam on and off.

The method of display control chosen for this system utilizes four different OUT instructions. These four instructions will be given the symbolic addresses XMOVE, YMOVE, XSTOR, and YDRAW in the programming examples to follow. When a byte is sent to XMOVE or YMOVE, the beam immediately moves to the new position without being turned on. A byte sent to XSTOR is stored in an 8 bit buffer register only and does not affect the beam position. When a byte is sent to YDRAW, the Y position register is loaded from the output lines and simultaneously the X position register is loaded from the buffer register providing simultaneous update of X and Y. The beam is also turned on until the line is drawn and then is extinguished in order to avoid a bright dot at the end of the line.

The data for generating a display can be obtained in a number of ways. In cases where the pattern to be drawn is repetitive such as a gameboard, it is often advantageous to compute the display data in line. In plotting routines, the X axis would be computed but the Y axis would be taken from a table in memory. In random figures such as the outline of a state, both X and Y would have to come from a list in memory. The example draw routine in Appendix 1 draws the figure defined in a memory buffer. On entry, registers H and L should point to the first byte of the buffer which contains a count of X-Y pairs to be displayed before returning. Successive bytes are paired with X first. The routine moves to the first point and then draws until the last point is done and then returns. Several calls using different buffers are normally required to display a number of disjoint figures.

Character display for data readout or text editing is also possible using the basic interface. One would have a coordinate table in memory for each different character shape to be used. The display text routine would address the proper table according to the character codes fetched from the text buffer. Before character segments are drawn, their coordinates have to be added to the character position coordinates in order to position the character to the desired line and space. In a fast computer these steps may not limit the number of characters that can be displayed but with the 8008 only a couple dozen characters could be displayed with a reasonable refresh rate.

In order to increase symbol display capability, a minor deflection system can be added for about $20 worth of parts and one or two more OUT instructions. The minor system consists of another X and Y register, beam control, and analog adder to add the output of the minor system to the output of the major system. Now character shapes can be traced out with the minor system and the entire character positioned with one setting of the major system. An optional size register can be added to the minor system to determine the size of the characters.

The minor system used here has a 3 bit X value, a 3 bit Y value, and a beam bit, thus a character segment can fit into one byte. These are organized in the byte as 0B XXX YYY. The unused bit is mormally used by the display characters routine to signify the last segment in the character. X and Y are positive three bit numbers. As a result, the lower left corner of the character should be thought of as the origin. As an example, the sequence of minor deflection bytes for the letter "A" is 000B, 126B, 140B, 012B, 332B. The size register if used is one byte with the left 4 bits specifying X size and the right 4 bits specifying Y size. The size values can be thought of as binary fraction multipliers of the minor X and Y values. Thus a size of 1000 (.5) would be roughly 1/2 the maximum size of 111 (.9375). The maximum size is normally chosen to be 1/16 of the full screen dimension.

A naively written program loop for stepping through the stroke list of a character might be LAM, OUT, ORA, JTS, INL, JMP where H & L point to the strokes to be outputted and the assumption is made that the list is not split accross a memory page boundary. Using the above loop, 43 states are required for each stroke. The display character subroutine in Appendix 2 makes a few minor alterations to the basic scheme in order to increase the speed to 22 states per stroke. Being a subroutine, a conditional return that uses 3 states when unsuccessful can be used rather than a conditional jump that uses 9 to detect the end of list. An 11 state JMP can be eliminated if the loop is expanded into straight line code. Little additional memory is used since the most complex ASCII character (a "B") requires only 11 strokes. The ORA used to set the condition code can also be removed if the stroke list is a list of consecutive differences between stroke bytes. In this case, an ADM instruction is used to load the next strokefrom memory and simultaneously set the condition code. As a byproduct, the initial minor beam position can always be 0,0 thus eliminating a stroke from many characters such as A, B, D, etc. The program segments in Appendix 2 are part of a complete character display package developed by the author. The package occupies 512 bytes and contains everything necessary to display formatted text from an ASCII string in memory using the standard 64 character set.

In conclusion it is felt that the interface described is adequate for general purpose graphic display applications within the limitations of the 8008. In the future, "The Computer Hobbyist" will publish a number of programs and routines that use this display interface.

APPENDIX 1

```
* EXAMPLE CALLING PROGRAM FOR DRAWING
* A TILTED SQUARE

SQUAR SHL TLTSQ      SET ADDRESS OF LIST
      CAL DRAW       DRAW SQUARE ONCE
       .             CHECK FOR IO DEVICE
       .             FINISHED, ETC.
      JMP SQUAR      LOOP FOR REFRESH



* EXAMPLE DRAW SUBROUTINE
* ENTER WITH ADDRESS OF BUFFER IN HL
* FIRST BUFFER BYTE IS COORDINATE COUNT
* SUCCEEDING BYTE PAIRS ARE COORDINATES
* X IS FRST IN BYTE PAIR

DRAW  LBM            GET COORD CONT
      CAL INHL       BUMP TO NEXT BYTE
      LAM            GET X OF FIRST
      OUT XMOV       COORD AND OUTPUT IT
      CAL INHL       BUMP TO NEXT BYTE
      LAM            GET Y OF FIRST
      OUT YMOV       COORD AND OUTPUT IT
DRAW1 DCB            DECREMENT COUNT
      RTZ            AND RETURN IF DONE
      CAL INHL       BUMP TO NEXT BYTE
      LAM            GET X COORD
      OUT XSTOR      AND STORE IN BUFFER
      CAL INHL       BUMP
      LAM            GET Y COORD
      OUT YDRAW      AND DRAW LINE
      JMP DRAW1      LOOP


* INCREMENT H AND L SUBROUTINE

INHL  INL            INCREMENT L
      RFZ            RETURN IF NO CARRY
      INH            INCREMENT H IF CARRY
      RET            RETURN


* TILTED SQUARE

TLTSQ DEF 5          NUMBER OF COORDINATES
      DEF -20,10     STARTING POINT
      DEF 0,80       CORNER COORDINATES
      DEF 80,60
      DEF 60,-20
      DEF -20,10
      APPENDIX 2

* PORTION OF DISPLAY TEXT ROUTINE
* TEXT ADDRESS IN DE

DTXT  LAB            POSITION BEAM TO
      OUT XMOV       CHAR LOCATION
      LAC
      OUT YMOV
      LHD            GET CHARACTER FROM
      LLE            ASCII SRING
      LAM
      INE            BUMP TEXT ADDRESS
      JFZ DTXT1
      IND
DTXT1 SUI '!'        TEST IF CONTROL CHAR
      JTS CTRL       JUMP IF SO
      LLA            ADDRESS THE STROKE
      LHI H(CHTB)    TABLE ACCORDING TO
      LLM            CHARACTER CODE
      CPI 'D'-'!'    INCREMENT H IF IN
      JTC DTXT2      SECOND PAGE OF STROKE
      INH            TABLE
      CAL DCHR       DISPLAY CHARACTER
      LAB            INCREMENT X POSITION
      ADI 8          SET FOR 32 CHAR/LINE
      LBA
      JMP DTXT+1
CTRL  .
      .

DCHR  XRA            SET MINOR TO 0,0
      OUT MINXY
      LAM            WAIT FOR SETTLE
      ORM            FETCH IRST STROKE
      OUT MINXY      TEST AND OUTPUT IT
      RTS            RETURN IF END OF
      INL            BUMP L
      ADM            FETCH AND TEST NEXT
      OUT MINXY      STROKE AND OUTPUT IT
      RTS            RETURN IF END OF LIST
      INL
      ADM
      OUT MINXY      REPEAT LAST 4
       .             INSTRUCTIONS 7 MORE
       .             TIMES
      RET


      ORG            ORG ON PAGE BUNDARY
CHTB  DEF L(ZEXC)    POINTER TABLE TO
      DEF L(ZDQUO)   STROKE TABLES
      DEF L(ZSHRP)
       .
       .

ZEXC  DEF 020B       EXCLAIMATION POINT
      DEF 120B-020B
      DEF 021B-12B
      DEF 362B-021B
ZDQUO DEF 014B       DOUBLE QUOTE
      DEF 116B-014B
       .
       .
```