PERSONAL COMPUTER WORLD

50p U.S.$2.00

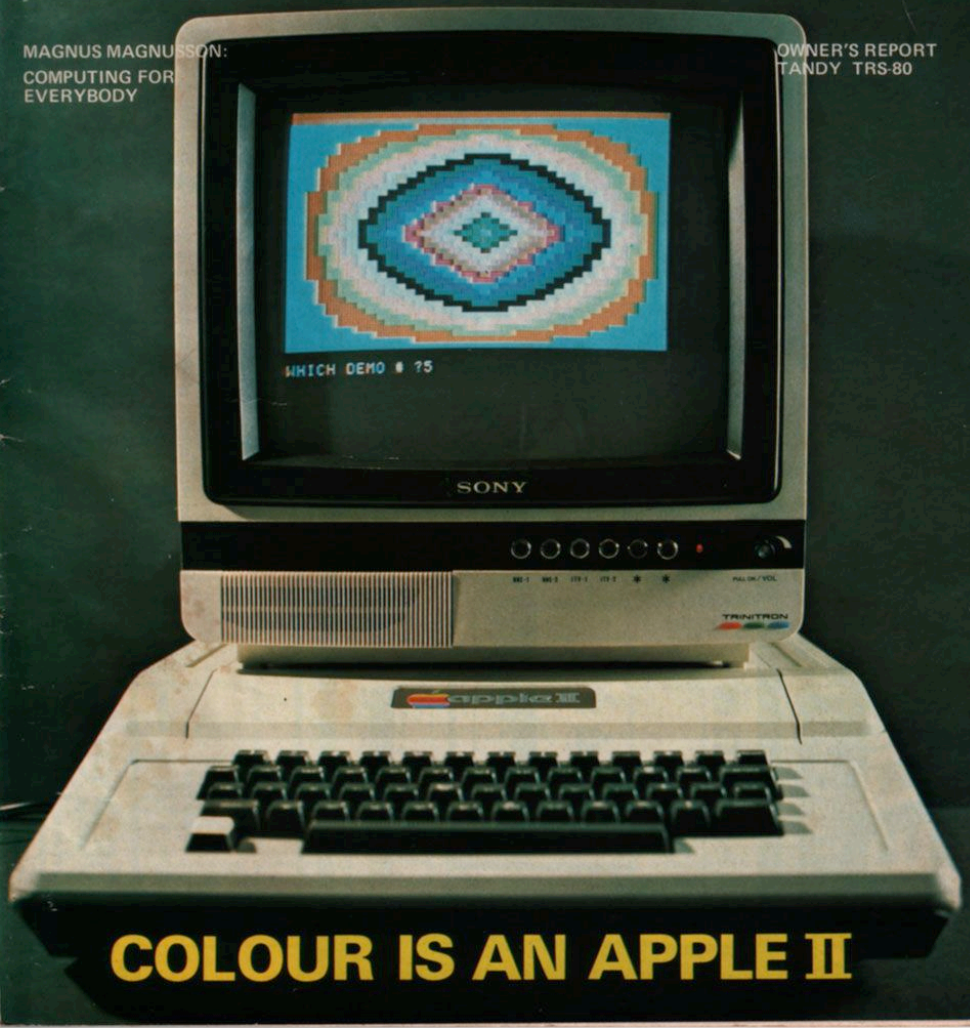
AUGUST 1978

- IT'S THE THOUGHT THAT COUNTS
- A COMPUTER OR A KIT
- TAKE IT OR LEAVE IT
- HANGMAN
- TOO HIGH A PRICE

Europe's first magazine for personal computers for home and business use

MAGNUS MAGNUSSON:
COMPUTING FOR EVERYBODY

OWNER'S REPORT
TANDY TRS-80



COLOUR IS AN APPLE II

COLOUR IS AN APPLE II

PCW REVIEW

JOHN COLL & CHARLES SWEETEN

The Apple II is a complete, assembled computer system. The system we had for evaluation loaned by Mike Sterland of Personal Computers Ltd., consists of a single box encasing the keyboard, the power supply, and the main board, together with a separate cassette recorder and a SONY portable colour TV set. There is also a carrying bag for the computer and two game paddles are included. All the necessary cables are present and attached.

The fundamental reason for buying an APPLE II must be for its colour graphics, for its extra high resolution graphics package, for its extreme portability and for its high quality construction. It is designed with expansion and extension in mind and, for example, you may increase the amount of memory simply by plugging chips into the sockets on the single circuit board.

The main printed circuit board and switched mode power supply is a model of good design. 66 integrated circuits are neatly laid out, with clear printed identification on the upper side. There is space for expansion RAM on board, so one can easily put extra RAM's in place. I/O connectors are a row of sockets near the back of the board. The absence of keyways or polarizing pins is unfortunate. The main board carries a small loudspeaker; the keyboard is well constructed.

The add-on UHF modulator, plugged into port #7 is of much lower standard – but performed adequately. Modulating a carrier with detail as fine as the HI-RES graphics offers is stretching the system to its limit as can be seen in the photographs. We would expect a dramatic improvement with a colour monitor.

The price for the computer itself with 16K of RAM is £1,250 plus £35 for the carrying bag.

A maintenance contract is available at 7½% of purchase every year. The software is free with the system we tested so the price is £1,285 plus VAT. The complete cost, including £220 for a TV and £15 for a cassette player, is £1,520 plus VAT.

The box is made of strong 5 mm plastic and is light, the interior has plenty of room and is very neatly laid out. For its sheer convenience we would recommend the carrying bag which makes transporting it very easy and safe. The keyboard is a delight to use and sets a standard of touch which other system manufacturers, who are not looking for rock bottom prices, would do well to aim at. We would like the Reset Key not to have been next to Return and = though. It is nice to have a Repeat Key that works with everything, and the form of screen editing that is available is particularly useful. Back space does not delete on screen, but is interpreted by the software, and forward space acts as if the character left has been typed. So that one can move the cursor to an offending line and repeat forward space, amend the incorrect character(s) and repeat forward space to the end of the required line. BASIC would re-format embedded blanks (or lack of them) to a standard format.

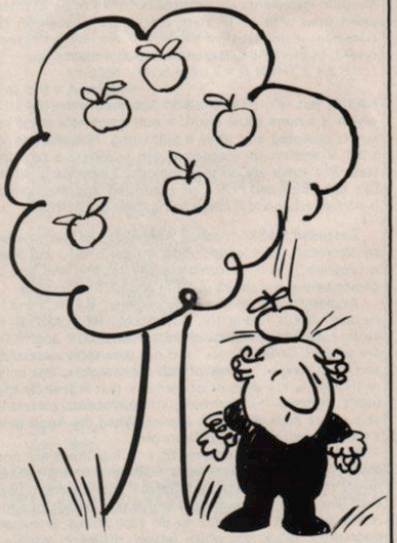
The display uses a UHF output and is 24 lines of 40 characters, with the standard ASCII set. It is memory mapped; that is to say that the contents of the screen are contained in a section of memory and the locations for each position of the screen are thus instantly addressable. This does give one obvious advantages in terms of speed and of course one can send the cursor to any position without having to wait. One can also have more than one page of text (or graphics) and can switch from one to another instantly.

Backing storage is cassette and the instructions for loading are very simple. The transfer rate is said to be 1500 bits per sec. which is five times as fast as many other cassette transfer rates. Applesoft extended BASIC takes 1½ mins. to load. Talking to Mike Sterland of Personal Computers Ltd., we gathered that APPLE are very much aware of the shortcomings of their cassette system as a backing store, notably that there is no way of finding the right file on a cassette except by listening to it. They are working on a floppy disk system rather than trying to improve the cassette handling. It will be interesting to see how APPLE cope with the file handling and to see whether they introduce an Editor and an Assembler.

Now, the prime reason that anyone buys an APPLE II must surely be for the colour graphics. In the low RES graphics mode, each character position of the screen is divided into two rectangles and each rectangle can be plotted in any of 15 brilliant colours plus black. Four lines of text can be retained at the bottom, which leaves a 40 by 40 grid for the colours. Very attractive pictures can be drawn and colourful games can be played.

Somebody must produce Startrek in colour soon! But one of the restrictions is that alphanumerics cannot be mixed with the colour display so that any labelling of diagrams has to be done in the bottom four lines. The instructions are nice and simple in both versions of BASIC, and one of the advantages of the memory mapped VDU is that complete lines (PIXELS) can be drawn *instantaneously*. But let us look first at the software that is at present available on this machine.

The software consists of a monitor, integer BASIC and extended BASIC and High-resolution graphics. The machine language environment is good. The firmware



An Apple Newton could have done with.

monitor provides an extensive set of instructions and binary data can be simply stored on cassette. However, the absence of an Editor, in the accepted sense of that word; or of an Assembler, prevent anything but the simplest machine language programs from being written.

The monitor commands enable memory to be examined and changed. A block of memory can be moved, and the move command can be easily used to fill a block of memory with any bit pattern. One block of memory can be verified (compared) with another block. Input and output ports can be easily changed.

Memory can also be examined directly in mnemonics using the resident dis-assembler. Dis-assembly can be of single lines or 20 lines at a time. That is a *very* useful feature. In addition, a program can be single stepped or "traced", potentially a most useful facility – marred only by the fact that the print-out is so fast that it is quite impossible to read if more than one page is displayed. The same criticism applies to BASIC listings, though obviously the latter can be controlled with the LIST command.

This must be one of the most comprehensive monitors available for a micro. It is a pity that, with such an outstanding monitor, there is no Editor/Assembler available. When a good Editor/Assembler *is* available the machine will be very easy to use in that environment.

Integer BASIC is in ROM and can be entered via control B, or via control C if you wish to retain any BASIC program that is in memory. Integer BASIC supports numbers up to 32767 and has a useful function MOD, which reduces a number to the remainder after a division. For example, 17 MOD 5 gives 2. There is an automatic line numbering command which is useful, but there is no renumber command (however, the on-screen editing is a feature that makes the lack of renumbering much less serious than would otherwise be the case). Multiple statements are possible but need to be watched when used with an IF statement. Usually when an IF statement is not satisfied the rest of the line is not processed. In this BASIC it *is* processed. For example,

IF 2 = 3 THEN A = 3 : A = 5

will give A = 5 in this BASIC, but will give A = 3 in Apple extended BASIC, which is a more usual result. A null return to a string input is accepted and gives a null string. It is possible to LIST a section of a program and to delete a section. Naturally there are no trigonometric functions or such-like but PEEK and POKE are supported, and the manual is particularly helpful about using these generally.

Extended BASIC is called Applesoft and was written by Microsoft. It is loaded under integer BASIC and then a program is run to complete the loading and give a choice between graphics or REM and LET statements.

Applesoft is much like other extended BASIC's and it is only worth listing the differences. RENUMBER is again absent. The logical operators AND, OR and NOT are present. String "space" did not have to be allocated, and "full length" strings of 255 are available. The only restriction is the amount of memory that is free for the user's program. Furthermore, string matrices are available where each element is a string. When the Apple gets a disk this will be a very definite plus.

Incidentally, null returns to a string input are not accepted. Multiple statements with conditional statements work in the expected manner. There are also two excellent pages in the manual which give details of addresses to PEEK or POKE to do such things as cursor addressing, black on white letters, different margins, page lengths, and etc. There is a list here of various machine language subroutines that can be used for communicating with the game paddles and with the internal speaker.

There are some differences in language between the two versions of BASIC which we feel could be tidied up. For example, if one wants to list lines 300 through 400 one uses LIST 300, 400 in Integer BASIC and one uses LIST 300-400 in Applesoft. To enter a machine language subroutine one uses CALL 3000 or USR (3000) depending on which level of BASIC one is in. CON or CONT, TEX or TEXT; no, the three letter versions are not in the same level of BASIC. They are very minor differences but irritating. Slightly annoying also is the fact that a program that is syntactically correct in both versions of BASIC cannot be saved in one and loaded in the other. One feature that neither version has and that we would plead with producers to provide is the ability to remain on the same line after an input. This allows so much more freedom to produce an acceptable format for interactive data presentation.

The commands that Applesoft uses to control the colour graphics are PLTG (go to graphics), PLTCX (colour X), PLTH X1, X2, Y (draw horizontal line), PLTV Y1, Y2, X (draw vertical line), PLTP X, Y (plot a point), TEX (text mode). They are very easy to use and appear to have few vices. Text is limited to the lines at the bottom of the screen so that the graphics are used for drawing pictures rather than diagrams. Do not be misled into thinking that the Financial Times Index can be plotted on the screen in a useful way. First, you cannot label the vertical axis or the trend lines, and secondly, you can only store the data as a DATA statement in a BASIC program, which makes updating a manual process, which I imagine defeats the purpose of such a program.

But of course the scope for game playing is enormous. Most teenage boys can reveal the details of Battleships, but what about 5-a-side football with the pitch, players and ball in individual colours? Or golf with greens, darker shades for the less approved areas of the course, yellow for bunkers, and so on?

The system that we had contains 20K and with extended BASIC this gives 5K bytes of program space. With Integer BASIC there is more than 10K bytes available for user programs. Integer BASIC is also faster and thus it is more suitable for game playing on both counts. The Benchmark timings using the programs listed in the first issue of *PCW* are shown in Figure 1.

Benchmark	Integer BASIC	Extended (Applesoft)
BM1	1.5	1.3
BM2	3.2	8.5
BM3	7.3	16
BM4	7.2	17.8
BM5	8.9	19.1
BM6	18.6	28.6
BM7	28.2	44.8
BM8	–	10.7

Figure 1

Timings of Benchmark programs

These show that Integer BASIC is fast and that Applesoft is satisfactory. So there should be few problems with speed. Microsoft Co. wrote Applesoft and also the PET and Altair BASICs, which is why the timings are "rather similar".

We have said that one of the reasons for buying an Apple II would be for its high-resolution graphics and it is worth looking at this in detail. At this moment (May 1978), we are not aware of another complete machine in this price bracket that has this particular feature. High resolution graphics is a set of machine code sub-routines which allow the user to mark points and lines on a screen that is 280 points across by 160 points down.

Three slightly variable colours are available (green, violet, white) and also "black". The subroutines allow one to poke the co-ordinates of a point into three locations and CALL 3780 will plot the point on the screen. After another set of co-ordinates has been poked into the same locations, one has the option of joining the last two points by CALL 3786. The beauty of a real line on the screen has to be seen to be believed. Given enough points, or lines, a curve can be drawn which from a few feet away looks as smooth as one could wish for. What is more, it is in colour and different lines or curves can thus be distinguished on the same screen.

However, there is an oddity which users may be puzzled by, and that is that the colours do not come out consistently. It is possible to draw a horizontal line in any position but not a vertical line; vertical green lines can be drawn at odd co-ordinates only, and violet ones at even co-ordinates only. Every other green line (visible ones) is yellowish, and every other violet line is bluish. White vertical lines can be drawn at every position, but are in fact shades of violet, green, blue, green in that order. It is only if one fills in solid patches of a single colour that its true shade emerges, and that is not always what one wishes to do with high resolution graphics!

The disappearance of alternate vertical lines can result in moire fringe effects. Figure 2 gives the test program and Figure 3 shows the lack of vertical line at X = 11 with the colour set to violet. But one gets used to it and anyway no-one else has colour graphics like this at this sort of price.

```
 10  CALL 3072
 20  INPUT "COLOUR",C
 30  POKE 812,C MOD 256
 40  INPUT "X,Y:",X,Y
 50  CALL 3086
 60  X=XMOD279:POKE800,X MOD 256
 70  POKE 801,X/256:POKE 802,0:CALL 3761
 80  POKE 802,159:CALL 3786
 90  POKE 800,0: POKE 801,0
100  POKE 802,Y MOD 159:CALL3761
110  POKE 800,23:POKE 801,1:CALL3786
120  GOTO 40
```

Figure 2 High resolution line test

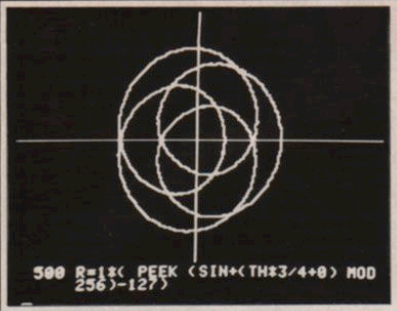
When I (C.S.) was first let loose on a computer it was a vast machine working in FORTRAN and after two days of my course, I discovered that it had a graph plotter on it. I spent the rest of the course drawing a four petal flower on a long stem. It seemed quite difficult, largely, I think, because I could not understand the manuals. Anyway, I have always wanted to do it on a colour TV.

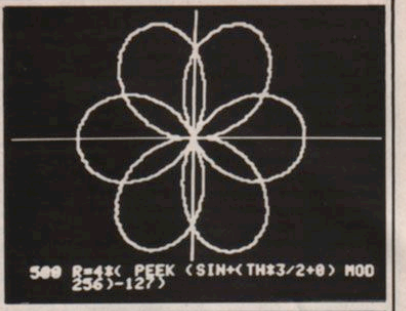
So, to demonstrate what the Apple high resolution graphics are capable of, I have written a polar graph program. The equations that I wanted to use need trigonometry and Integer BASIC naturally does not have such things. Happily there is a block of memory in the high resolution package which has a table of values of SIN(X) with the function evaluated between ϕ and 255 and with the values of X going from ϕ to 255. This table starts at 3840 and goes through to 4095 decimal. SIN=3840:PEEK(SIN + (TH)MOD256) gives SIN(TH) for positive TH with 256 units in a revolution. PEEK (SIN + (TH+64)MOD256) gives COS(TH).

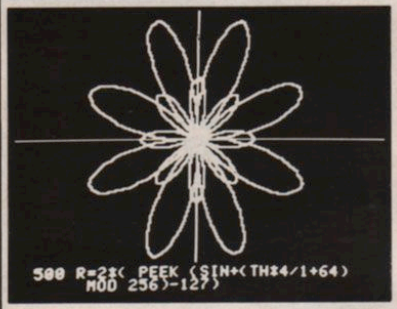
```
100  SIN=3840
110  POKE812,170
120  CALL3072
130  CALL3086
140  POKE800,0:POKE801,0
150  POKE802,80:CALL3761
160  POKE801,1:CALL3786
170  POKE800,123:POKE801,0
180  POKE802,0:CALL3761
190  POKE802,159:CALL3786

300  LIST500
310  PRINT"85VIOLET,170GREEN,255WHITE"
320  INPUT"COLOUR(85,170,255)",C
330  INPUT"NO.OF REVOLUTIONS",N
340  LIST500
350  POKE812,C
360  GR:B=0
400  FOR TH = 0 TO 256*N

500  R= 2*(PEEK(SIN+(TH*4/3+64)
     MOD256)-127)
510  R=R+256
550  IF ABS(R)>816THEN800
560  R=R*5/16
570  E=0

600  S=PEEK(SIN+(TH)MOD256)-127
610  C=PEEK(SIN+(TH+64)MOD256)-127
620  X=R*C/256+123
630  Y=R*S/256+80
640  IF X >279 THEN E=1
650  IF Y >159 THEN E=1
660  IF X<0 THEN E=1
670  IF Y<0 THEN E=1
680  IF E=1  THEN B=0
690  IF E=1  THEN 800

700  POKE800,XMOD256:POKE801,X/256
710  POKE802,Y
720  IF B=0 THEN CALL 3780
730  B=1
740  CALL3786

800  NEXT TH
900  END
```

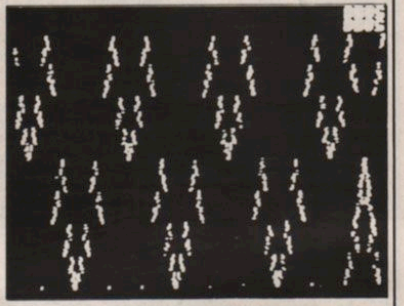
The program contains the equation of the picture at lines 500 and 510; these can easily be altered using the edit facility. From 600 to 700 converts the co-ordinates into x and y values and joins successive values up on the display.

Lines 100 to 190 draw the axes and lines 300 to 400 control the colour and the number of revolutions needed to describe the curve. To draw one curve on top of another one merely starts at 300. To make solid flowers









one can join each point to the origin by adding the following lines:

```
750  POKE 800,123:POKE 801,0 : POKE 802,80 :
     CALL 3761
760  POKE 800,X MOD256: POKE 801,X/256:
     POKE 802,Y: CALL 3786
```

The photographs illustrate the remarkable resolution that is available. Figure 4 is obtained with 500 R = PEEK(SIN+(TH*3/4)MOD256)–127 and 4 revolutions. Figure 5 is obtained by a double plot; first 500 R=4*(PEEK(SIN+(TH*3/2)MOD256)–127) in white and with two revolutions; then 500 R=2*(PEEK(SIN+(TH*6+64)MOD256)–127) in violet with one revolution. Figure 6 is obtained with 500 R=4*(PEEK(SIN+(TH*4)MOD256)–127) in violet with one revolution; then follow with 500 R=2*(PEEK(SIN+(TH*4+64)MOD256)–127) in white with one revolution. Figure 7 shows the effect of filling in figure 6 and one can just make out the shot silk effect in the outer petals. Naturally it needs colour to do it full justice.

For those who would like to pursue this further I have listed a few shapes to try.

Circle	500 R=2*(PEEK(SIN+(TH+64)MOD256) –127)+256
Ellipse	500 R=10*(PEEK(SIN+(TH+64)MOD 256)/7 510 R=32767/(R+256)
Hyperbola	as for the Ellipse but divide line 500 by 3 rather than 7
Trisectrix	500 R=2*(PEEK(SIN+(TH+64)MOD256) –127)+128

The manuals were in some ways *excellent* and included, for example, a very detailed discussion of the subroutines available and a full listing of the monitor and board layout and circuit diagrams. But there are many errors and though the scope is comprehensive, it is very obviously a first edition and even includes some hand-written pages. A point worth mentioning to users is in the section on program editing with Applesoft on p.58 etc., where the example is wrong. The key 'left arrow' is not identical to escape B as stated. You must use escape B to make the example work and p. 56 is correct.

As an example of the locations which can be manipulated and the sections of resident code that can be utilised within one's own programs, here is a selection of what is given many times throughout the manuals.

POKE 32,L	Set left margin of TV display
POKE 33,W	Set the width of TV display
POKE 34,T	Set top margin line of TV display
POKE 35,B	Set bottom margin line of TV display
POKE 36,CH	Move cursor to horizontal position CH+1 relative to left margin
POKE 37,CV	Move cursor to vertical position CV T less than CV less than B
POKE 50,127	Set text mode to flash outputted characters
X = USR(–958)	Clear display from present cursor position to bottom of window and left of window
X = USR(–912)	Scroll window up one line and leave characters outside window unchanged
PEEK(–16384)	Examine keyboard input: if contents greater than 127 then a key has been hit whose value in ASCII is found from 127 AND the contents

The manual gives instructions for connecting peripherals via one of eight peripheral connectors near the back edge of the printed circuit board. Each of these has sixteen addresses associated with it, and all the signals required are fully described.

We mentioned earlier that it is easy to plug in more memory. What is also rather nice is that the manual tells you *exactly* how to do this. There are three memory select sockets, and these allow you to put in jumpers to place either 4K chips, 16K chips or both at any address

block you please up to 48K. There are a few restrictions naturally, but what we like is the approach that the consumer knows what he wants better than the manufacturer does, and here is one manufacturer who is anxious that we should get it.

All in all then the Apple II is a very promising machine which is nicely packaged, has minimal BASIC at the touch of a key, has very adequate extended BASIC, has colour graphics which are half way to being beautiful, has an excellent monitor, has well thought out expansion possibilities. The Apple would be even more of a temptation were its price slightly lower.

There is also considerable development to be done on the software side to make it comparable with what is available on many other systems; and the expansion possibilities will need to be catered for by the software.

We would like to say how much co-operation we received from Personal Computers Ltd and how much we appreciated their attitude – it bodes well for the future.

So for the moment, colour is an Apple II.